Robert F. Kramer (SBN 181706)
rkramer@feinday.com
M. Elizabeth Day (SBN 177125)
eday@feinday.com
David Alberti (SBN 220625)
dalberti@feinday.com
Sal Lim (SBN 211836)
slim@feinday.com
Russell Tonkovich (SBN 233280)
rtonkovich@feinday.com
Marc Belloli (SBN 244290)
mbelloli@feinday.com
FEINBERG DAY KRAMER ALBERTI LIM TONKOVICH & BELLOLI LLP
577 Airport Boulevard, Suite 250
Burlingame, California 94010
Telephone: (650) 825-4300
Facsimile: (650) 460-8443

Attorneys for Plaintiff
Express Mobile, Inc.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

EXPRESS MOBILE, INC.,

Plaintiff,

v.

SALESFORCE.COM, INC.,

Defendant.

Case No. 3:20-cv-08461

COMPLAINT FOR PATENT INFRINGEMENT

DEMAND FOR JURY TRIAL

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Express Mobile, Inc. ("Express Mobile" or "Plaintiff"), by its attorneys, demands a trial by jury on all issues so triable and for its Complaint against Salesforce.com, Inc. ("Salesforce" or "Defendant") alleges the following:

NATURE OF THE ACTION

1. This action arises under 35 U.S.C. § 271 for Salesforce's infringement of Express Mobile's United States Patent Nos. 6,546,397 ("the '397 patent"), 7,594,168 ("the '168 patent"), 9,928,044 ("the '044 patent"), 9,471,287 ("the '287 patent") and 9,063,755 ("the '755 patent").

THE PARTIES

2. Plaintiff Express Mobile, Inc. is an inventor-owned corporation organized under the laws of the State of Delaware with a place of business at 38 Washington Street, Novato, California 94947.

3. Upon information and belief, Salesforce.com, Inc. is a corporation organized and existing under the laws of the State of Delaware, with a place of business at Salesforce Tower 415 Mission Street, 3rd Floor San Francisco, California 94105. Salesforce may be served through its registered agent for service at CT Corporation System 818 West Seventh St., Suite 930 Los Angeles, California 90017.

4. Upon information and belief, Salesforce claims to be the world's #1 customer relationship management (CRM) platform. More than 150,000 companies use Salesforce CRM to grow their businesses by strengthening customer relationships and better understanding their customers' needs by better managing customer information and interactions.

JURISDICTION AND VENUE

5. This Court has jurisdiction over the subject matter of this action pursuant to 28 U.S.C. §§ 1331 and 1338(a).

6. Upon information and belief, jurisdiction and venue for this action are proper in the Northern District of California.

7. This Court has personal jurisdiction over Salesforce because Salesforce has purposefully availed itself of the rights and benefits of the laws of this State and this Judicial District. Upon information and belief, Salesforce resides in the Northern District of California by maintaining a regular and established place of business at Salesforce Tower 415 Mission Street, 3rd Floor San Francisco, California 94105.

8. This Court also has personal jurisdiction over Salesforce because Salesforce has done and is doing substantial business in this Judicial District, both generally and, upon information and belief, with respect to the allegations in this complaint, including Salesforce's one or more acts of infringement in this Judicial District.

9. Venue is proper in this Judicial District under 28 U.S.C. §§ 1391(b) and (c) and § 1400(b). Salesforce has committed acts of infringement through, for example, performing a method to allow users to produce Internet websites in the Northern District of California and has a regular and established place of business in this District. Salesforce's office in San Francisco is a physical place in the District, it is an established location where Salesforce's business has been carried out for several years, and Salesforce publicly advertises its presence in the District.

INTRADISTRICT ASSIGNMENTS

10. Pursuant to Local Rule 3-2 (c), this case involves intellectual property rights and is subject to assignment on a district wide basis.

PATENTS IN SUIT

11. Express Mobile is the lawful owner of all rights, title, and interest in the '397 patent titled "Browser Based Web Site Generation Tool and Run Time Engine," including the right to sue and to recover for infringement thereof. The '397 patent was duly and legally issued on April 8, 2003, naming Steven H. Rempell as the inventor. A true and correct copy of the '397 patent is attached as Exhibit A.

12. The inventions of the '397 patent solve technical problems related to website creation and generation. For example, the inventions enable the creation of websites through browser-based visual editing tools such as selectable settings panels which describe website

elements, with one or more settings corresponding to commands. These features are exclusively implemented utilizing computer technology including a virtual machine.

13. The claims of the '397 patent do not merely recite the performance of some pre-Internet business practice on the Internet. Instead, the claims of the '397 patent recite inventive concepts that are rooted in computerized website creation technology, and overcome problems specifically arising in the realm of computerized website creation technologies.

14. The claims of the '397 patent recite inventions that are not merely the routine or conventional use of website creation systems and methods. Instead, the inventions teach a browser-based website creation system and method in which the user-selected settings representing website elements are stored in a database, and in which said stored information is retrieved to generate said website.

15. The technology claimed in the '397 patent does not preempt all ways of using website or web page authoring tools nor any other well-known prior art technology.

16. Accordingly, each claim of the '397 patent recites a combination of elements sufficient to ensure that the claim amounts to significantly more than a patent on an ineligible concept.

17. Plaintiff is the lawful owner of all rights, title, and interest in United States Patent No. 7,594,168 titled "Browser Based Web Site Generation Tool and Run Time Engine," including the right to sue and to recover for infringement thereof. The '168 patent was duly and legally issued on September 22, 2009, naming Steven H. Rempell as the inventor. A true and correct copy of the '168 patent is attached as Exhibit B.

18. The inventions of the '168 patent solve technical problems related to website creation and generation. For example, the inventions enable the creation of websites through browser-based build tools and a user interface. The inventions greatly improve the productivity of the designer utilizing an innovative implementation for styles. These features are implemented utilizing computer technology.

19. The claims of the '168 patent do not merely recite the performance of some pre-Internet business practice on the Internet. Instead, the claims of the '168 patent recite inventive

concepts that are rooted in computerized website creation technology and overcome problems specifically arising in the realm of computerized website creation technologies.

20. The claims of the '168 patent recite inventions that are not merely the routine or conventional use of website creation systems and methods. Instead, the inventions teach a browser-based website creation system including a server comprising a build engine configured to create and apply styles to, for example, a website with web pages comprised of objects.

21. The technology claimed in the '168 patent does not preempt all ways of using website or webpage authoring tools nor any other well-known or prior art technology.

22. Accordingly, each claim of the '168 patent recites a combination of elements sufficient to ensure that the claim amounts to significantly more than a patent on an ineligible concept.

23. In *Express Mobile v. KTree Computer Solutions*, a case filed in the Eastern District of Texas, the defendant, KTree Computer Solutions, brought a Motion for Judgment on the Pleadings asserting that the '397 patent and the '168 patent is invalid as claiming abstract subject matter under 35 U.S.C. § 101. (C.A. 2:17-00128; Dkt. 9, 17, 22-27). The briefing associated with the motion is incorporated by reference into this Complaint.

24. After considering the respective pleadings, Magistrate Judge Payne recommended denial of KTree's motion, without prejudice, holding that "the claims appear to address a problem particular to the internet: dynamically generating websites and displaying web pages based on stored user-selected settings" and further stating "the asserted claims do not bear all of the hallmarks of claims that have been invalidated on the pleadings by other courts in the past. For example, the claims are not merely do-it-on-a-computer claims." (Dkt. 29, attached as Exhibit C.) No objection was filed to the Magistrate Judge's report and recommendation and the decision therefore became final.

25. In *Express Mobile v. Pantheon Systems, Inc*., a case filed in the Northern District of California, the defendant, Pantheon Systems, Inc., brought a Motion to Dismiss Plaintiff's First Amended Complaint asserting that the '397 patent and the '168 patent were directed to the abstract idea of creating and displaying webpages based upon information from a user with no further

inventive concept and purportedly ineligible for patenting under 35 U.S.C. § 101. (Case No. 3:18-CV-04688-RS; Dkt. 26, 32 and 34). The briefing associated with the motion is incorporated by reference into this Complaint.

26. In *Express Mobile v. Code and Theory LLC*, a case filed in the Northern District of California, the defendant, Code and Theory LLC, brought a Motion to Dismiss Plaintiff's Complaint asserting that the '397 patent and the '168 patent are not subject matter eligible under 35 U.S.C. § 101 as a matter of law. (Case No. 3:18-CV-04679-RS; Dkt. 35, 40 and 41). The briefing associated with the motion is incorporated by reference into this Complaint.

27. After a hearing and a consideration of the respective pleadings, Hon. Richard Seeborg denied both motions holding that:

- "The patents here are directed at a purportedly revolutionary technological solution to a technological problem—how to create webpages for the internet in a manner that permits 'what you see is what you get' editing, and a number of other alleged improvements over the then-existing methodologies." *Id*. at 5.
- The claims of the '397 and '168 patents are "directed to a specific improvement to the way computers operate," and "it simply cannot be said on the present record that the claims are drawn so broadly as to be divorced from the potentially patent-eligible purported technological improvements described in the specification." *Id*. at 5-6. (Case No. 3:18-CV-04679-RS; Dkt.45; Case No. 3:18-CV-04688-RS Dkt.40; attached as Exhibit D.)

28. In Case Nos. 1:18-CV-01173-RGA and 1:18-CV-01175-RGA, infringement actions filed by Plaintiff in the District of Delaware, the respective defendants in those actions, Dreamhost LLC and Hostway Services, Inc., brought Motions to Dismiss claims of the '397 and '168 patents on the basis of invalidity under 35 U.S.C. § 101. (Case No. 1:18-CV-01173-RGA D.I. 14, D.I. 18-21 and 24 Case No. 1:18-CV-01175-RGA D.I. 17-19 and 23). The briefing associated with the motion is incorporated by reference into this Complaint.

29. After consideration of the respective pleadings, Judge Andrews denied both motions in a joint order, pointing to factual allegations of inventiveness identified by the

Plaintiff, and an expert declaration explaining inventiveness of the claims, noting that such factual issues preclude a finding of invalidity on a motion to dismiss. (Case No. 1:18-CV-01173-RGA D.I. 43; Case No. 1:18-CV-01175-RGA D.I. 42; attached as Exhibit E.)

30. Plaintiff is the lawful owner of all rights, title, and interest in United States Patent No. 9,928,044 titled "Systems and Methods for Programming Mobile Devices," including the right to sue and to recover for infringement thereof. The '044 patent was duly and legally issued on March 27, 2018, naming Steven H. Rempell, David Chrobak and Ken Brown as the inventors. A true and correct copy of the '044 patent is attached as Exhibit F.

31. The inventions of the '044 patent solve technical problems associated with methods and systems for displaying content on displays of devices by providing more efficient ways of generating, storing and retrieving code for displaying content, for example, dynamic content, uniformly across different kinds of devices. For example, the inventions of the '044 patent allow a data-efficient and flexible association between a symbolic name with a User Interface ("UI") object (e.g., a UI object for a widget) corresponding to a web component of a web service, that is manually or automatically selected. The symbolic name has a data format type corresponding to a subclass of UI objects that support the data format type of the symbolic name and is only available to UI objects that support the data format of the symbolic name. Information representative of the defined UI object can be stored in a database and subsequently retrieved from the database to build an application consisting of at least a portion of the database using a player, which uses the information to generate one or more web pages for display across different kinds of devices (e.g., PC, mobile or tablet; or different browsers, operating systems and applications, including for example both native and browser-based applications.)

32. The claims of the '044 patent do not merely recite the performance of some pre-Internet business practice on the Internet. Instead, the claims of the '044 patent recite inventive concepts that are rooted in the computerized, data-efficient definition, selection, storage and generation of user defined object attributes (e.g., a UI object for a widget) on displays for different types of devices, such as PC, mobile or tablet or different browsers, and applications. Such features are specifically grounded in, and overcome problems with data efficiency and

flexibility specifically arising in, the realm of computerized content generation and display technologies, and are not well-understood, routine and conventional elements.

33. For example, the claimed inventions of the '044 patent recite innovative, technical improvements that select and associate symbolic names with defined UI objects (e.g., UI objects for a widget) corresponding to web components of web services based on, for example, data format type, storing information representative of such settings in a database, and building applications, which together with players, generate uniform, data-efficient content, such as dynamic content, for display across different types of devices.

34. The technology claimed in the '044 patent does not preempt all ways for the computerized generation of code for a display of a device nor any other well-known or prior art technology. For example, the specific, innovative technical improvements do not preempt well-known methods of generating code for a display of a device by programming in HTML or JavaScript code.

35. Accordingly, each claim of the '044 patent thus recites a combination of elements sufficient to ensure that the claim amounts to significantly more than a patent on an ineligible concept.

36. Plaintiff is the lawful owner of all rights, title, and interest in United States Patent No. 9,471,287 titled "Systems and Methods for Integrating Widgets on Mobile Devices," including the right to sue and to recover for infringement thereof. The '287 patent was duly and legally issued on October 18, 2016, naming Steven H. Rempell, David Chrobak and Ken Brown as the inventors. A true and correct copy of the '287 patent is attached as Exhibit G.

37. The inventions of the '287 patent solve technical problems associated with methods and systems for displaying content on displays of devices by providing more efficient ways of generating code for uniformly displaying content, for example dynamic content, across different kinds of devices. For example, the inventions of the '287 patent allow a data-efficient and flexible association between a symbolic name and a UI object (e.g., a UI object for a widget) corresponding to a web component of a web service, that is defined for presentation on a display of a device. The defined UI object can be selected by a user of an authoring tool or

automatically selected by a system based on a web component selected by the user. Further, the symbolic name has a data format type corresponding to a subclass of UI objects that support the data format type of the symbolic name. A device-independent application including the symbolic name is then produced and provided to the device together with a device-platform-dependent player. Such operations provide a user-friendly platform allowing the UI object to be efficiently defined and uniformly displayed across different kinds of devices (e.g., PC, mobile or tablet; or different browsers, operating systems, and applications, including for example both native and browser-based applications).

38. The claims of the '287 patent do not recite merely the performance of a known business practice on the Internet. Instead, the claims of the '287 patent recite inventive concepts grounded in the computerized, data-efficient definition and generation of object attributes (e.g., a UI object for a widget) on displays for different types of devices, such as PC, tablet, or mobile devices, or different browsers and applications. Such features are specifically grounded in, and overcome problems with data efficiency and flexibility specifically arising in, the realm of computerized content generation and display technologies, and are not well-understood, routine, and conventional elements.

39. For example, the claimed inventions of the '287 patent recite innovative, technical improvements that associate symbolic names with UI objects (e.g., UI objects for a widget) corresponding to web components of web services that are manually or automatically selected, and defined based on, for example, data format type, and produce device-independent applications including those symbolic names, together with device-dependent players, to provide uniform, data-efficient server-based content display across different types of devices.

40. The technology claimed in the '287 patent does not preempt all ways for the computerized generation of code for a display of a device nor any other well-known or prior art technology. For example, the specific, innovative technical improvements do not preempt well-known methods of generating code for a display of a device by programming in HTML or JavaScript code.

41. Each claim of the ’287 patent thus recites a combination of elements sufficient to ensure that the claim amounts to significantly more than a patent on an ineligible concept.

42. Accordingly, each claim of the ’287 patent recites a combination of elements sufficient to ensure that the claim amounts to significantly more than a patent on an ineligible concept.

43. Plaintiff is the lawful owner of all rights, title, and interest in United States Patent No. 9,063,755 titled “Systems and Methods for Presenting Information on Mobile Devices,” including the right to sue and to recover for infringement thereof. The ’755 patent was duly and legally issued on June 23, 2015, naming Steven H. Rempell, David Chrobak and Ken Brown as the inventors. A true and correct copy of the ’755 patent is attached as Exhibit H.

44. The inventions of the ’755 patent utilize inventive concepts to solve technical problems associated with methods and systems for displaying content on displays of devices, providing more efficient ways of generating code for uniformly displaying content, for example dynamic content, across different kinds of devices. For example, the inventions of the ’755 patent allow a data-efficient and flexible association between a symbolic name and a UI object (e.g., a UI object for a widget), corresponding to a web component of a web service, that is defined for presentation on a display of a device. A device-independent application including the symbolic name is produced and provided to the device, together with a device-platform-dependent player.

45. The claimed inventions of the ’755 patent allow the UI object to be efficiently displayed across different kinds of devices (e.g., PC, mobile or tablet; or different browsers, operating systems, and applications, including both native and browser-based applications), as opposed to, for example, programming directly in HTML or JavaScript code. In turn, a user can enter an input value to the UI object, and obtain an output value based on a web service associated with the UI object, the input value and output value also being communicated through symbolic names to provide an additional level of efficiency.

46. The claims of the ’755 patent do not recite merely the performance of a known business practice on the Internet. Instead, the claims of the ’755 patent recite inventive concepts

concerning the computerized, data-efficient generation of server-based content (e.g., a UI object for a widget) on displays for different types of devices, such as PC, tablet, or mobile devices, or different browsers and applications. For example, the claims of the '755 utilize symbolic name associations and provide device-independent applications including those symbolic names, together with device-platform-dependent players, to devices. Further, input values and output values for the defined content are also communicated as symbolic names. Such features are specifically grounded in, and overcome problems with data efficiency and flexibility specifically arising in, the realm of computerized content generation and display technologies, and are not well-understood, routine, and conventional elements.

47. For example, the claimed inventions of the '755 patent recite innovative, technical improvements that associate symbolic names with defined UI objects (e.g., UI objects for a widget) corresponding to web components of web services, and produce device-independent applications including those symbolic names, together with device-platform-dependent players, to provide uniform, data-efficient content, such as dynamic content, for display across different types of devices.

48. The technology claimed in the '755 patent does not preempt all ways for the computerized generation of code for a display of a device, nor any other well-known or prior art technology. For example, the specific, innovative technical improvements claimed in the '755 patent do not preempt well-known methods of generating code for a display of a device by programming in HTML or JavaScript code.

49. Each claim of the '755 patent thus recites a combination of elements sufficient to ensure that the claim amounts to significantly more than a patent on an ineligible concept. Accordingly, each claim of the '755 patent recites a combination of elements sufficient to ensure that the claim amounts to significantly more than a patent on an ineligible concept.

BACKGROUND

50. Plaintiff Express Mobile is a leader in the business of developing mobile app and web site design and creation platforms, and has intellectual property including U.S. patents relating to certain tools useful in the field. Express Mobile is managed by individuals with

decades of technology and business experience. The Chairman of the Board and CTO of Express Mobile, Steve Rempell, is the inventor of Express Mobile's patent portfolio. Mr. Rempell has over 50 years of experience in technology companies, with much of that work focused on web-based technologies and applications.

51. Defendant Salesforce is a well-known company that claims to be the world's #1 customer relationship management (CRM) platform. More than 150,000 companies use Salesforce CRM to grow their businesses by strengthening customer relationships and better understanding their customers' needs by better managing customer information and interactions. Salesforce has grown rapidly and now generates billions of dollars of revenue per year.

COUNT I - INFRINGEMENT OF U.S. PATENT NO. 6,546,397

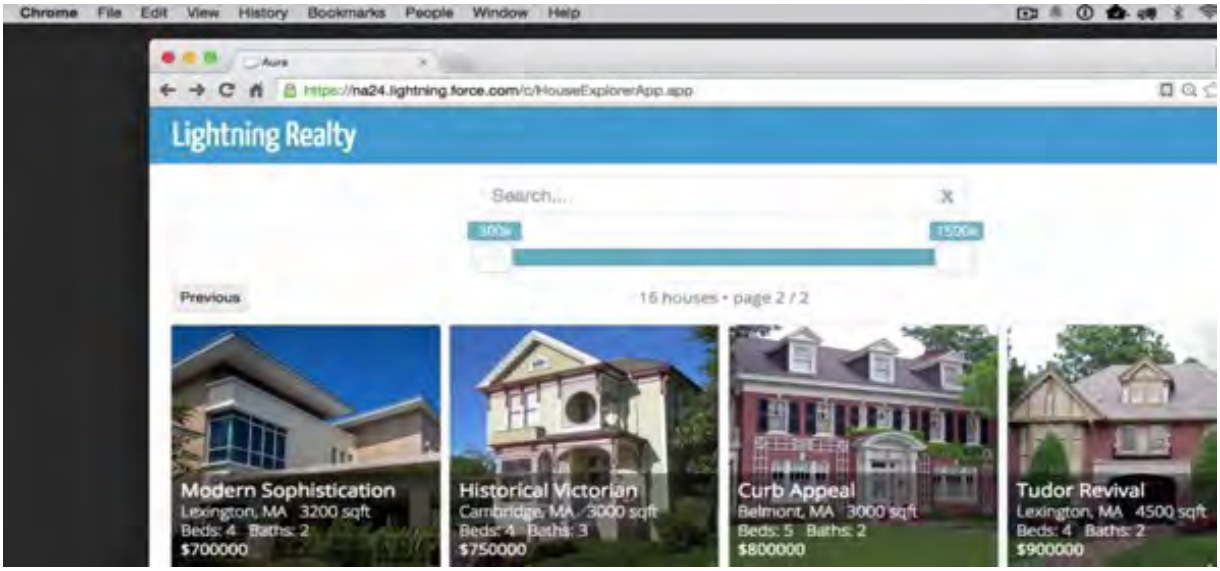
52. Plaintiff incorporates by reference the allegations contained in paragraphs 1 to 51 above.

53. Defendant Salesforce has performed a method to allow users to produce Internet websites which infringes, either literally or under the doctrine of equivalents, one or more claims of the '397 patent in violation of 35 U.S.C. § 271(a).

54. Upon information and belief, Salesforce directly infringed at least claim 1 of the '397 patent through its Lightning App Builder website-building tools that are configured for use with one of a number of browsers on a user's computer (the "Accused Instrumentality") that provided browser-based website authoring tools in which the user-selected settings representing website elements are stored in a database and in which the stored information is retrieved to generate the website.

55. Salesforce has infringed at least claim 1 of the '397 patent by having performed a method to allow users to produce Interest websites on and for computers having a browser and a virtual machine capable of generating displays.

56. The Accused Instrumentality enabled a user to produce a website through a browser on the user's computer interacting with Salesforce's server facilities, hosted by Salesforce.com. By way of example, an enterprise (e.g., Lightning Realty) creates an account @ Salesforce.com and builds a Home Page using the Lightning App Builder.



Source: https://www.youtube.com/watch?v=KxamMxgzjxw

57. This resulted in a home page with a unique and stable URL (i.e., na24.lightning.force.com/c/HouseExplorerApp.app) as well as a set of one or more related web pages, which can include blogs, widgets and plug-ins.

58. The user can then add different types of pages.

> **Lightning App Builder**
> The Lightning App Builder is a point-and-click tool that makes it easy to create custom pages for the Salesforce mobile app and Lightning Experience, giving your users what they need all in one place. The Lightning App Builder is also a one-stop shop for configuring Lightning apps.
> You can access the Lightning App Builder from Setup by entering Lightning App Builder in the Quick Find box and then selecting **Lightning App Builder**.
>
> With the Lightning App Builder, you can build:
> - Single-page apps that drill down into standard pages
> - Dashboard-style apps, such as apps to track top sales prospects or key leads for the quarter
> - "Point" apps to solve a particular task, such as an expense app for users to enter expenses and monitor expenses they've submitted
> - Custom record pages for your objects, tailored to the needs of your users
> - Custom Home pages containing the components and features that your users use most
>
> But that's not all. When you edit a Lightning app from the App Manager in Setup, you're brought into the Lightning App Builder to manage the app's settings. You can update the app's branding, navigation, app options, and manage the Lightning pages assigned to that app all inside the Lightning App Builder.

(https://help.salesforce.com/articleView?id=lightning_app_builder_overview.htm&type=5).

59. The Accused Instrumentality is configured to operate with browser engines to interpret and execute JavaScript and HTML to render web pages on a computer. For example, Internet Explorer relies on Trident (code name for MSHTML) (which now includes the Chakra JavaScript Engine, see http://en.wikipedia.org/wiki/MSHTML); Safari and Chrome rely on Webkit (which includes WebCore and JavaScript Core, see http://en.wikipedia.org/wiki/WebKit); Firefox relies on Gecko (which includes Spidermonkey, see http://www.mozilla.org/projects/technologies.html); Opera currently relies on Presto (which in turn relies on Carakan) and will soon move to WebKit. (See http://en.wikipedia.org/wiki/Presto_layout_engine).

Lightning Experience and Lightning-Based Features

The following table describes minimum browser version requirements for using Lightning components within various features that are built with our next-generation user interface platform.

	Microsoft® Internet Explorer®	Microsoft® Edge	Google Chrome™	Mozilla® Firefox®	Apple® Safari®
Lightning Experience	IE 11 (EOL December 21, 2020)[1]	Windows 10	Latest	Latest	10.x+
Salesforce Console in Lightning Experience	N/A	Windows 10	Latest	Latest	10.x+
Lightning Communities	IE 11	Windows 10	Latest	Latest	10.x+
Lightning for Outlook (Client)	IE 11	N/A	N/A	N/A	N/A
Lightning for Outlook (Web)	IE 11	Windows 10	Latest	Latest	10.x+
Standalone Lightning App (my.app)	IE 11	Windows 10	Latest	Latest	10.x+
Lightning Out	IE 9+	Windows 10	Latest	Latest	10.x+

Source: https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/intro_browsers.htm

60. All of these well-known modern browsers rely on engines that fit the definition of a virtual machine, which interpret and execute JavaScript and HTML to render web pages on a computer. These include, but are not limited to, Javascript engines such as Chrome V8

(Chrome), SpiderMonkey (Firefox), JavaScriptCore (Safari), Chakra (Edge), as well as browser engines such as Webkit (Safari), Gecko (Safari) and Blink (Chrome).

61. The Accused Instrumentality presented a viewable menu having a user selectable panel of settings describing elements on a website, the panel of settings being presented through a browser on a computer adapted to accept one or more of the selectable settings in the panel as inputs therefrom, and where at least one of the user selectable settings in the panel corresponds to commands to a virtual machine.

62. After Logging into Salesforce, the Accused Instrumentality was accessed from the Salesforce setup page, by entering "Lightning App Builder" in the "Quick Find" box and then selecting the Accused Instrumentality. An app page was built using the following steps:

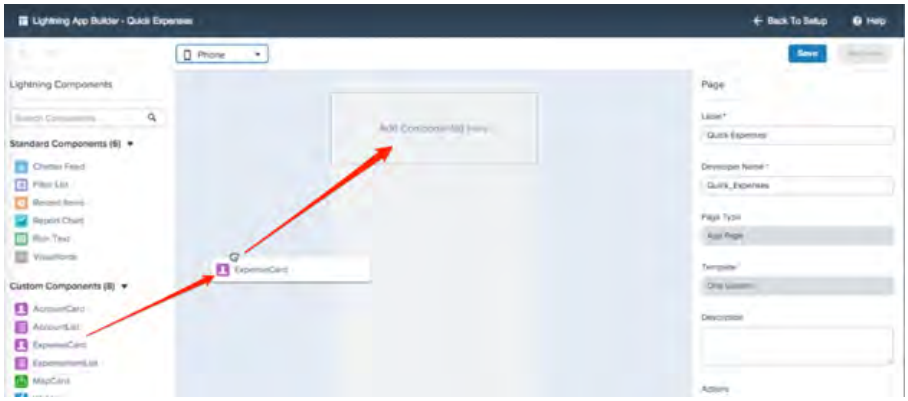
> **Create an App Home Page with the Lightning App Builder**
> Create a home page for an app that you can add directly into the Salesforce mobile app navigation list and into Lightning apps. Your users can then easily access the objects and items that are most important in that app.
>
> 1. From Setup, enter App Builder in the Quick Find box, then select **Lightning App Builder**.

Screenshot - Salesforce Lightning Experience Set-up page." https://youtu.be/XMK5Huuviow " 1:03/4:29

> 2. Click New.
> 3. Select App Page and then click Next…"

Screenshot - Create new Lightning page. https://youtu.be/XMK5Huuviow " 1:05/4:29

4. Give your app page a label, and then click **Next**. The label can be up to 80 characters.
5. Select a page template and click **Finish**.

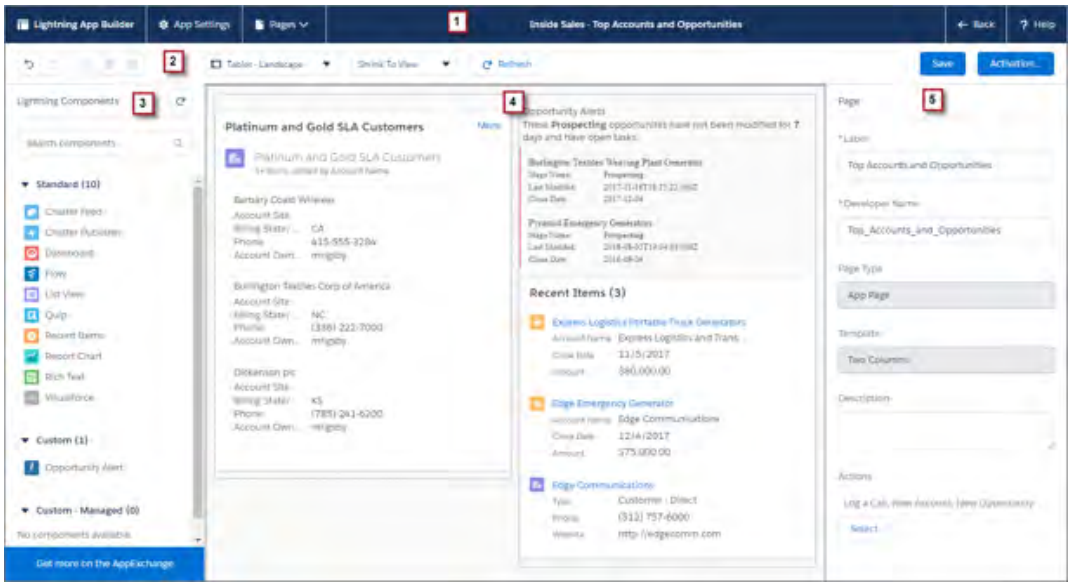


Screenshot - App Builder Drag and drop WYSIWYG UI. https://youtu.be/XMK5Huuviow 2:02/4:29

6. Drag the components that you want onto the page.
7. Click each component on the page to configure its properties."

Source: "https://help.salesforce.com/articleView?id=lightning_app_builder_create_app_page.htm&type=5," and "https://youtu.be/XMK5Huuviow".

63. The Accused Instrumentality presented a viewable menu having a user selectable panel of settings describing elements on a website. For example, upon choosing a template from the Dropdown shown above, the browser opens a webpage containing the Accused Instrumentality User Interface.



UI Graphic from

https://trailhead.salesforce.com/modules/lightning_app_builder/units/lightning_app_builder_intro

The Lightning App Builder's user interface makes creating Lightning pages easy. Here's a breakdown of the parts of the tool.

Header (1)

When you're working on a Lightning page, the header shows you its label, and contains a Pages list where you can see the last 10 pages that you modified. You can also return to Setup without saving or to view more help for the Lightning App Builder.

If you're editing an app, the header also shows the app name and contains an App Settings tab where you can configure the app's options such as branding, navigation, and the utility bar. In an app context, the Pages list shows all active Lightning pages associated with the current app, including record and app pages assigned to the app, and all Home pages that are assigned to a profile.

Toolbar (2)

Use the buttons in the toolbar to cut, copy, and paste page content; and to undo, redo, save, or activate your Lightning page. You can also view your page in different formats, refresh the canvas, or adjust the canvas size to fit your view.

Lightning Components Pane (3)

The components pane contains all standard and custom Lightning components that are supported for your Lightning page. Click and drag a component to add it to the page.

Lightning Page Canvas (4)

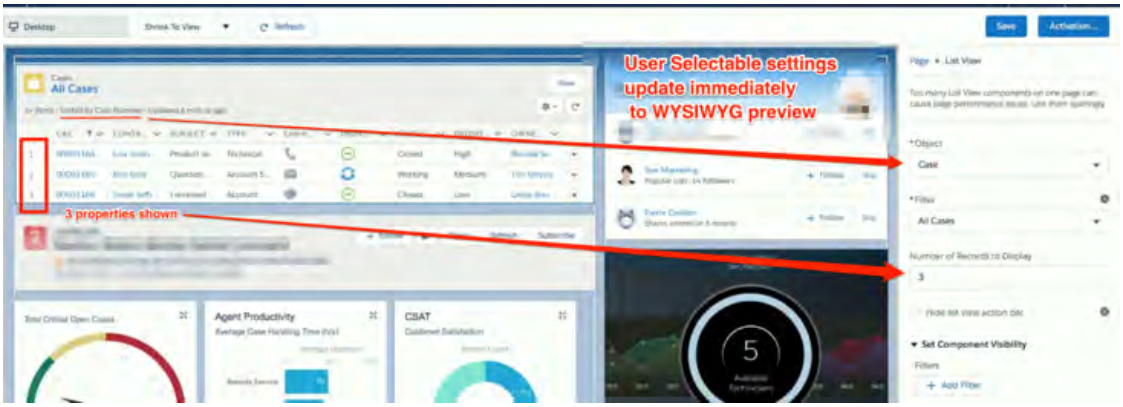
The canvas area is where you build your page. Drag components to reorder them on the page.

Properties Pane (5)

Depending on what you select on the page, the properties pane shows either the overall page properties or the properties of the component that you've selected. Click Page in the breadcrumb to access the page properties when viewing a component.

Source: https://trailhead.salesforce.com/modules/lightning_app_builder/units/lightning_app_builder_intro

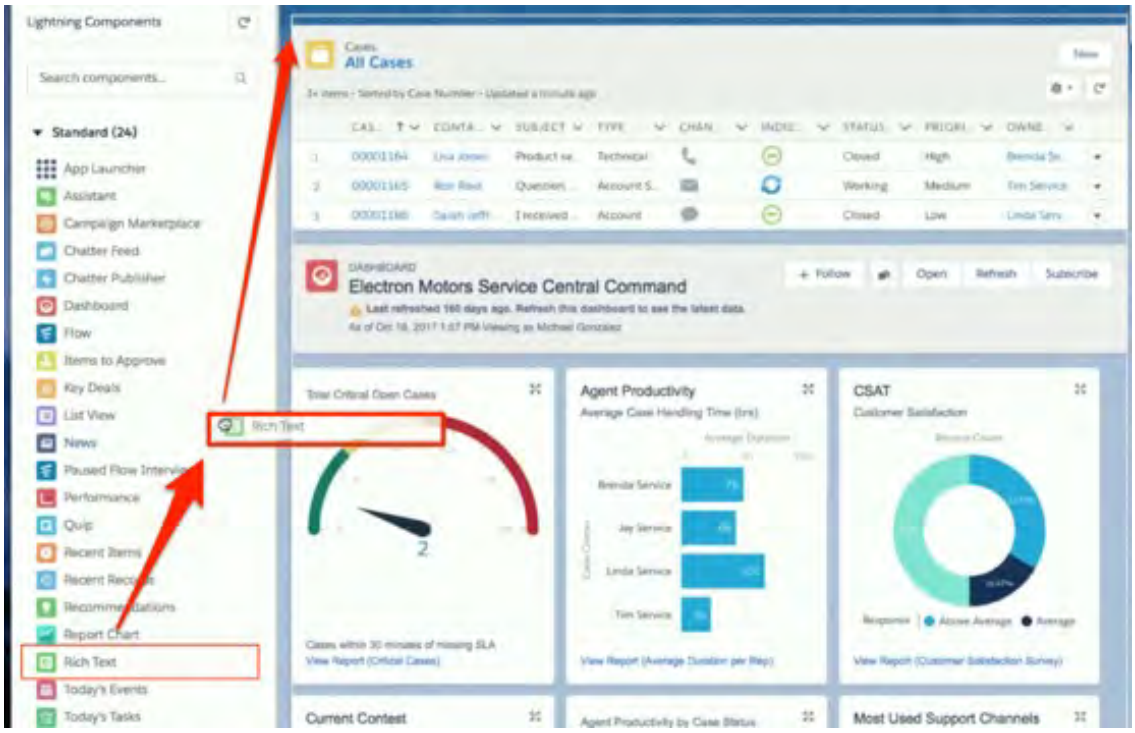
64. In the figure below, the Properties Pane (panel) to the right contained exemplary user selectable settings for that list view. The user selected settings "Object: Case" and "Number of Records to Display: 3" are reflected in the Lightning Page Canvas to the left.



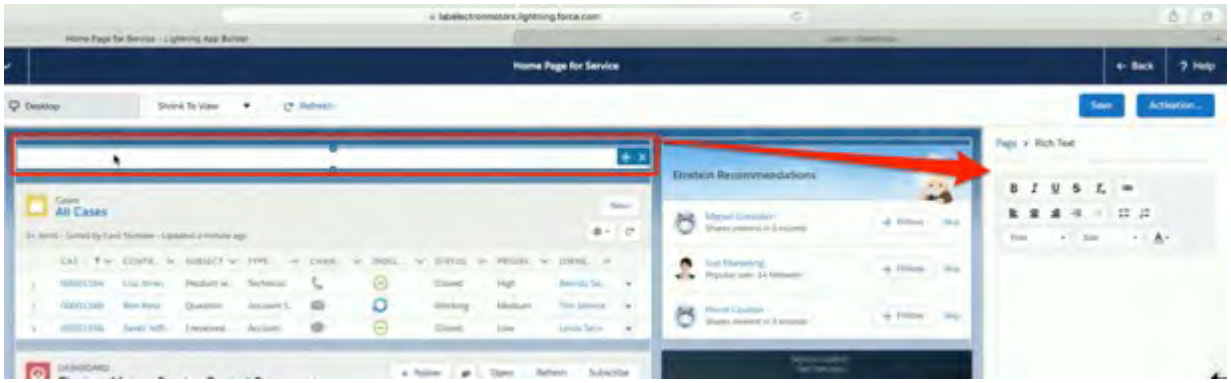
Screenshot - Panel of user selectable settings asper https://youtu.be/dQTa4sijMqY 06:23 / 17:14

65. Because the compatible browsers all rely on browser engines (which are process virtual machines) to generate and display the user-authored web pages on a particular computer, the settings that a user-author selects through the Salesforce visual editor necessarily correspond to virtual machine commands. For example, JavaScript and HTML code are used by the browser's engine to generate and display a Salesforce web page for a user viewing the web page on a computer, the author's selections being reflected in the displayed page.

66. The Accused Instrumentality generated a display in accordance with one or more user selected settings substantially contemporaneously with the selection thereof. This is performed, for example, using the Salesforce visual editing tool through the browser. Exemplary screens below show the selection of the Rich Text component from the left-hand Components Pane and its placement above the list component in the preview window. The properties panel changes to reflect the Rich Text Component's user selectable settings.



Source: https://youtu.be/dQTa4sijMqY 06:27 / 17:14



Source: https://youtu.be/dQTa4sijMqY 06:31 / 17:14

67. In the exemplary figure below, the user selected settings in the Properties Pane, the text "Reminder: Manager Training in HQ. Please register" and "Bold" style are immediately reflected in the Page Canvas. Once the user selects options such as image alignment, text color and paragraph styles through the WYSIWYG editor, the display immediately updates to reflect the selected option. The result from the selections is illustrated in the Lightning Page Canvas shown in the center of the App Builder Window.

Source: https://youtu.be/dQTa4sijMqY 06:43 / 17:14

68. User selected settings were displayed substantially contemporaneously with the selection thereof. In the example above, the bold option has been selected from the options available in the Rich Text properties panel and applied to the user defined text as soon as the button is pressed.

69. The Accused Instrumentality stored information representative of the one or more user selected settings in a database. For example, the Accused Instrumentality built Lightning apps which were built and hosted on the Salesforce server.

> **Lightning App Builder**
> The Lightning App Builder is a point-and-click tool that makes it easy to create custom pages for the Salesforce mobile app and Lightning Experience, giving your users what they need all in one place. The Lightning App Builder is also a one-stop shop for configuring Lightning apps.
> You can access the Lightning App Builder from Setup by entering Lightning App Builder in the Quick Find box and then selecting **Lightning App Builder**.
>
> With the Lightning App Builder, you can build:
> - Single-page apps that drill down into standard pages
> - Dashboard-style apps, such as apps to track top sales prospects or key leads for the quarter
> - "Point" apps to solve a particular task, such as an expense app for users to enter expenses and monitor expenses they've submitted
> - Custom record pages for your objects, tailored to the needs of your users
> - Custom Home pages containing the components and features that your users use most
>
> But that's not all. When you edit a Lightning app from the App Manager in Setup, you're brought into the Lightning App Builder to manage the app's settings. You can update the app's branding, navigation, app options, and manage the Lightning pages assigned to that app all inside the Lightning App Builder.

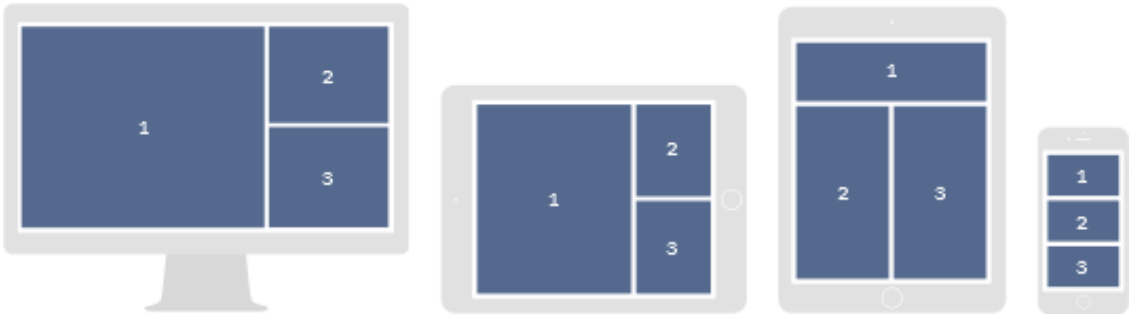
Source: https://help.salesforce.com/articleView?id=lightning_page_overview.htm&type=5

Lightning Pages.

A Lightning page is a custom layout that lets you design pages for use in the Salesforce mobile app or Lightning Experience.

Lightning pages occupy a middle ground between page layouts and Visualforce pages. Like a page layout, Lightning pages allow you to add custom items to a page. However, these items, instead of being fields or Visualforce components, are Lightning components, which allow much more flexibility.

The structure of a Lightning page adapts for the device it's viewed on. The template you choose when creating the page controls how it displays on a given device. The Lightning page's template divides the page into regions.



Lightning pages are built using Lightning components—compact, configurable, and reusable elements that you can drop into regions of the page in the Lightning App Builder.

Source: https://help.salesforce.com/articleView?id=lightning_page_overview.htm&type=5

70. As shown below, in the Salesforce Developer website is a description of Lightning Components that explains of what they are composed, how data was stored and received within them, the data being representative of one or more user settings in a database.

A Lightning component is a combination of markup, JavaScript, and CSS. You first create a component bundle.

1. In the Developer Console, select **File | New | Lightning Component**.
2. For the component name, enter `MyContactList`.
3. Check **Lightning Record Page** and then click **Submit**.
4. Add a reference to the Apex controller, `controller="MyContactListController"`, on the `aura:component` tag. The `aura:component` tag should now look like this:

```
<aura:component controller="MyContactListController"
implements="flexipage:availableForRecordHome,force:hasRecordId" access="global" >
```

5. Add the following code to the component on line 2:

```
<aura:attribute name="recordId" type="Id" />

<aura:attribute name="Account" type="Account" />

<aura:attribute name="Contacts" type="Contact" />

<aura:attribute name="Columns" type="List" />

<force:recordData aura:id="accountRecord"

                  recordId="{!v.recordId}"

                  targetFields="{!v.Account}"

                  layoutType="FULL"

                  />

<lightning:card iconName="standard:contact" title="{! 'Contact List for ' +
v.Account.Name}">

    <!-- Contact list goes here -->

</lightning:card>
```

An `aura:attribute` is what we use to store data within a Lightning component, in this case, the id of the current Account record page. The component is also using Lightning Data Service, `force:recordData`, to retrieve and store the fields of the current record in the attribute named `Account`, so that we can display the name of the Account as the title of the component. The Lightning Framework contains UI building blocks known as Base Lightning Components. `lightning:card` is a base component for displaying data. The title of the card is pulling the name of the account from the aura attribute `Account`.

6. Select **File | Save**.

Source: https://trailhead.salesforce.com/projects/quickstart-lightning-components/steps/quickstart-lightning-components3

71. The Accused Instrumentality generated a website at least in part by retrieving the information representative of the one or more user selected settings stored in the database. In particular, the author-user can then generate and view the created web page. On initial request App components are rendered to produce HTML DOM elements and sent to populate the web page in the browser.

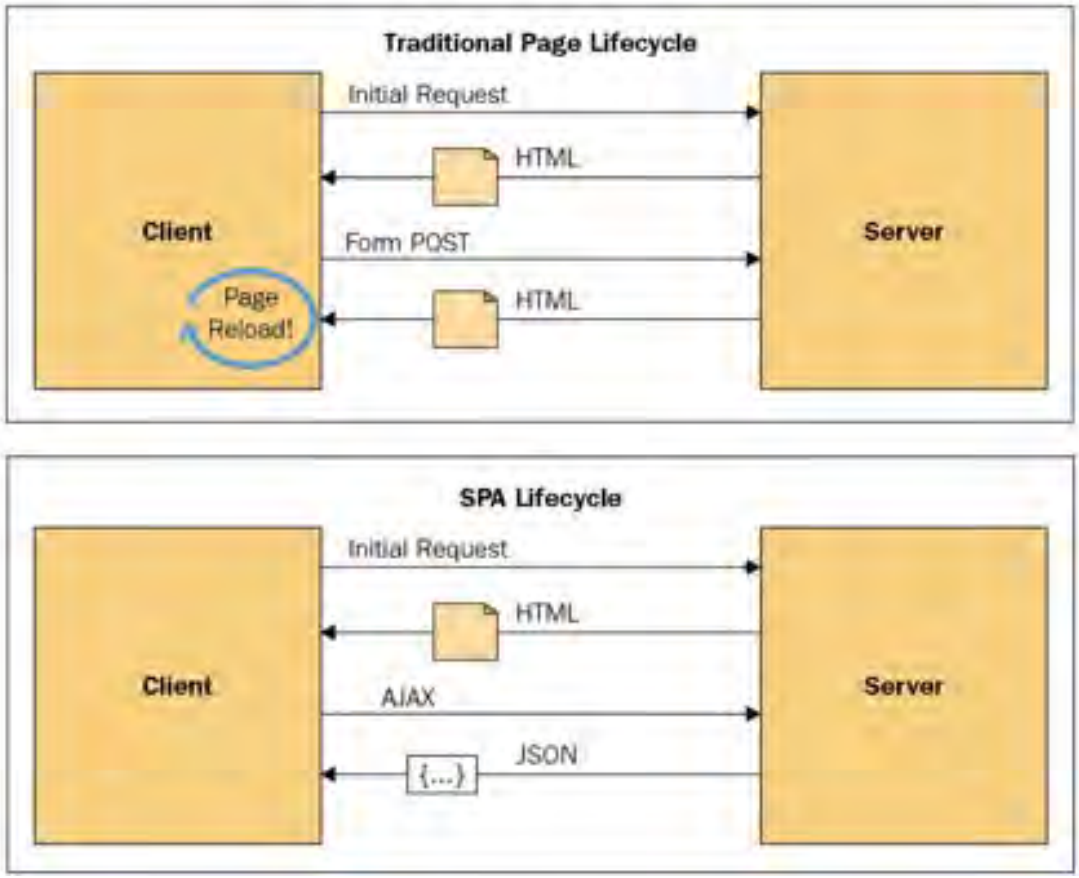
Lightning Component architecture

Before we start building Lightning Components, let's understand the architectural differences between how Lightning Components and Salesforce Classic Visualforce work.

In a traditional Classic Salesforce UI, every time a user clicks a link, an HTML page is loaded from the server. Even for small interactions, the whole web page or tab needs to be reloaded.

Lightning Experience is based on the concept of a **Single-Page Application** (**SPA**). In SPAs, once the page is loaded, for any other subsequent request of data from the server, an Ajax callout is made and the page is re-rendered with new data. The aim is to avoid the whole page refreshing.

The following diagram helps to visualize the difference between a single-page architecture and a traditional web page rendering mechanism:



Source:
https://www.packtpub.com/mapt/book/application_development/9781787124677/1/ch01lvl1sec11/lightning-component-architecture

72. The Lightning Component Frame work was built on the Aura Framework using metadata to build user interfaces in Lightning pages.

"What is Aura?
Aura is an open-source UI framework built by Salesforce for developing dynamic web apps for mobile and desktop devices. You can use open-source Aura to build apps independent of Salesforce.
Salesforce uses Aura to build apps, such as Lightning Experience and the Salesforce app. Salesforce customers use the Lightning Component framework to build Lightning components. The Lightning Component framework is built on the open-source Aura framework. However, the open-source Aura framework has some features that aren't available to Salesforce customers.

Aura's architecture uses two key building blocks: components and events.
Components are the self-contained and reusable units of an app. They represent a reusable section of the UI, and can range in granularity from a single line of text to an entire app.
The framework includes a set of prebuilt components in the aura and ui namespaces. You can assemble and configure components to form new components in an app. Components are rendered to produce HTML DOM elements within the browser.
Components communicate with other components by firing and handling events. The robust event model enables you to develop loosely coupled components. Once you define the events that interact with your components, your team can work on the components in parallel to quickly build a powerful app."

Source: https://github.com/forcedotcom/aura

73. As shown in the excerpt below, Lightning Data Service, (force:recordData), retrieved and stored the fields of current records from the server-side database. Aura:attribute was used to store data within the Lightning component and lightning:card was used to Display the data. The data was refreshed from the database server via Ajax requests and JSON Responses through the Component controller.

"A Lightning component is a combination of markup, JavaScript, and CSS. You first create a component bundle.

```
<aura:component controller="MyContactListController"
implements="flexipage:availableForRecordHome,force:hasRecordId" access="global" >
```

7. Add the following code to the component on line 2:

```
<aura:attribute name="recordId" type="Id" />
<aura:attribute name="Account" type="Account" />
<aura:attribute name="Contacts" type="Contact" />
<aura:attribute name="Columns" type="List" />
<force:recordData aura:id="accountRecord"
                  recordId="{!v.recordId}"
                  targetFields="{!v.Account}"
                  layoutType="FULL"
                  />
<lightning:card iconName="standard:contact" title="{! 'Contact List for ' +
v.Account.Name}">
    <!-- Contact list goes here -->
</lightning:card>
```

An `aura:attribute` is what we use to store data within a Lightning component, in this case, the id of the current Account record page. The component is also using Lightning Data Service, `force:recordData`, to retrieve and store the fields of the current record in the attribute named `Account`, so that we can display the name of the Account as the title of the component. The Lightning Framework contains UI building blocks known as Base Lightning Components. `lightning:card` is a base component for displaying data. The title of the card is pulling the name of the account from the aura attribute `Account`.

6. Select **File | Save**.

Source: https://trailhead.salesforce.com/projects/quickstart-lightning-components/steps/quickstart-lightning-components3

74. The Accused Instrumentality built one or more web pages to generate the website from at least a portion of the database and at least one run time file, where the at least one run time file utilized information stored in the database to generate virtual machine commands for the display of at least a portion of the one or more web pages. The basic building block of a Lightning Page, which is a Lighting Component, is made of a combination of Markup (e.g., HTML), CSS, and Java Script, including Runtime files. The Runtime files provided virtual machine commands for display of the one or more pages when they were read and used by the applicable browser's engine to render the one or more pages. A user, whether a software developer, a website designer at Salesforce or a third-party utilizing the Accused Instrumentality, could view the finalized

website in a browser outside of the website authoring environment to verify the website conforms to the intended design.

75. Upon information and belief, Salesforce was made aware of the '397 patent and its infringement thereof at least as early as December 20, 2018 when Express Mobile provided notice of Saleforce's infringement of the '397 patent to Amy Weaver, who, at the time was, President, Legal & Corporate Affairs, General Counsel and Secretary. Upon information and belief, since at least the time Salesforce received notice, Salesforce induced others to infringe at least one claim of the '397 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Salesforce's clients, customers, and end users, whose use of the Accused Instrumentalities constitutes direct infringement of at least one claim of the '397 patent. In particular, Saleforce's actions that aided and abetted others such as customers and end users to infringe included advertising and distributing the Accused Instrumentalities and providing instruction materials, training, and services regarding the Accused Instrumentalities. *See, e.g.* www.salesforce.com, help.salesforce.com, https://trailhead.salesforce.com, and related subdomains. Upon information and belief, Salesforce engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Salesforce has had actual knowledge of the '397 patent and knowledge that its acts were inducing infringement of the '397 patent since at least the date Salesforce received notice that such activities infringed the '397 patent.

76. Upon information and belief, Salesforce was liable as a contributory infringer of the '397 patent under 35 U.S.C. § 271(c) by offering to sell, selling and importing into the United States website or web page authoring tools to be especially made or adapted for use in an infringement of the '397 patent. The Accused Instrumentalities are a material component for use in practicing the '397 patent and are specifically made and are not a staple article of commerce suitable for substantial non-infringing use.

77. Upon information and belief, since the date of its receipt of notice, Salesforce's infringement of the '397 patent was willful and intentional under the standard announced in *Halo Elecs., Inc. v. Pulse Elecs., Inc.,* 136 S.Ct. 1923, 195 L.Ed 2d 278 (2016). Since at least December

20, 2018, Salesforce willfully infringed the '397 patent by refusing to take a license and continuing to make, use, test, sell, license, and/or offer for sale/license the '397 Accused Instrumentality. Salesforce has been aware that it infringes the '397 patent since at least December 20, 2018 and instead of taking a license, Salesforce has opted to make the business decision to "efficiently infringe" the '397 patent. In doing so, Salesforce willfully infringed the '397 Patent.

78. Salesforce's infringement has damaged Express Mobile.

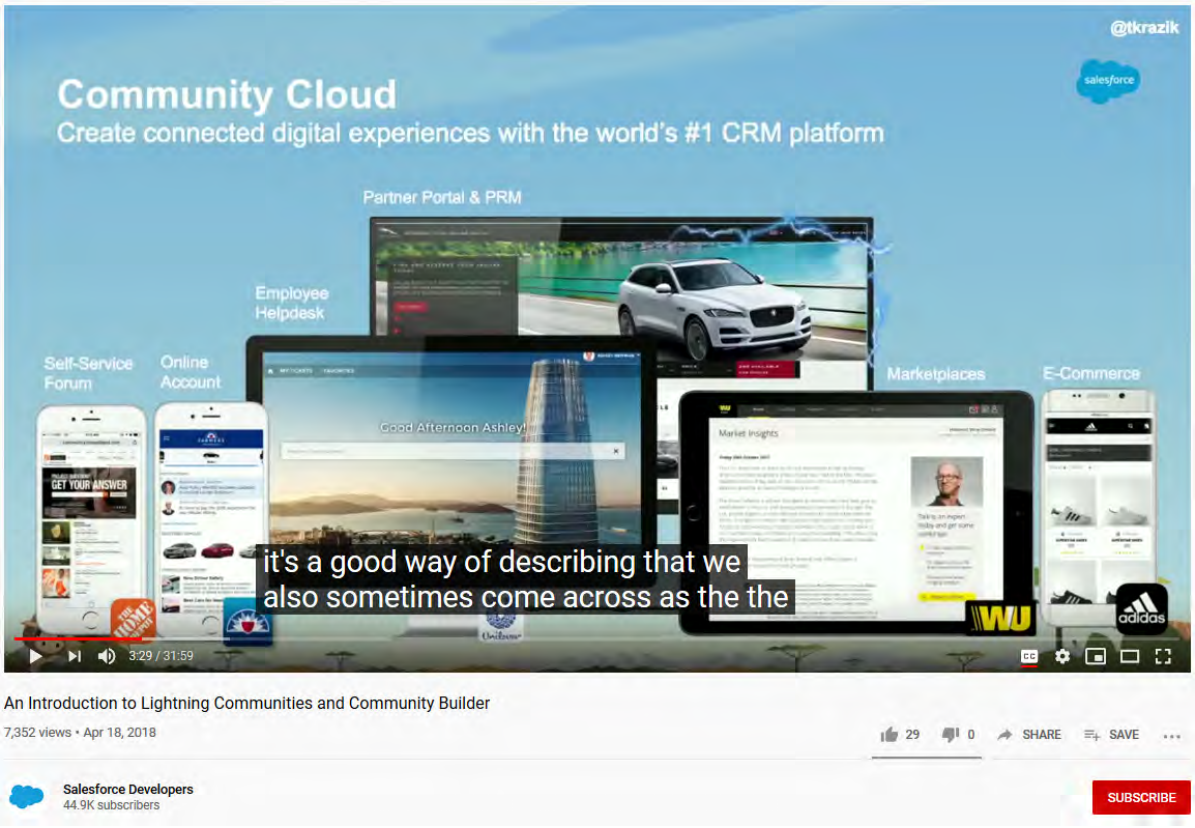
COUNT II - INFRINGEMENT OF U.S. PATENT NO. 7,594,168

79. Plaintiff incorporates by reference the allegations contained in paragraphs 1 to 78 above.

80. Salesforce manufactured, sold, offered to sell, used and/or provided and continues to manufacture, sell, offer for sale, use and/or provide its Community Cloud, which includes the Lightning Community Builder when employed with any number of computers and servers (the "Accused Instrumentality") that infringes, either literally or under the doctrine of equivalents, one or more claims of the '168 patent in violation of 35 U.S.C. § 271(a).

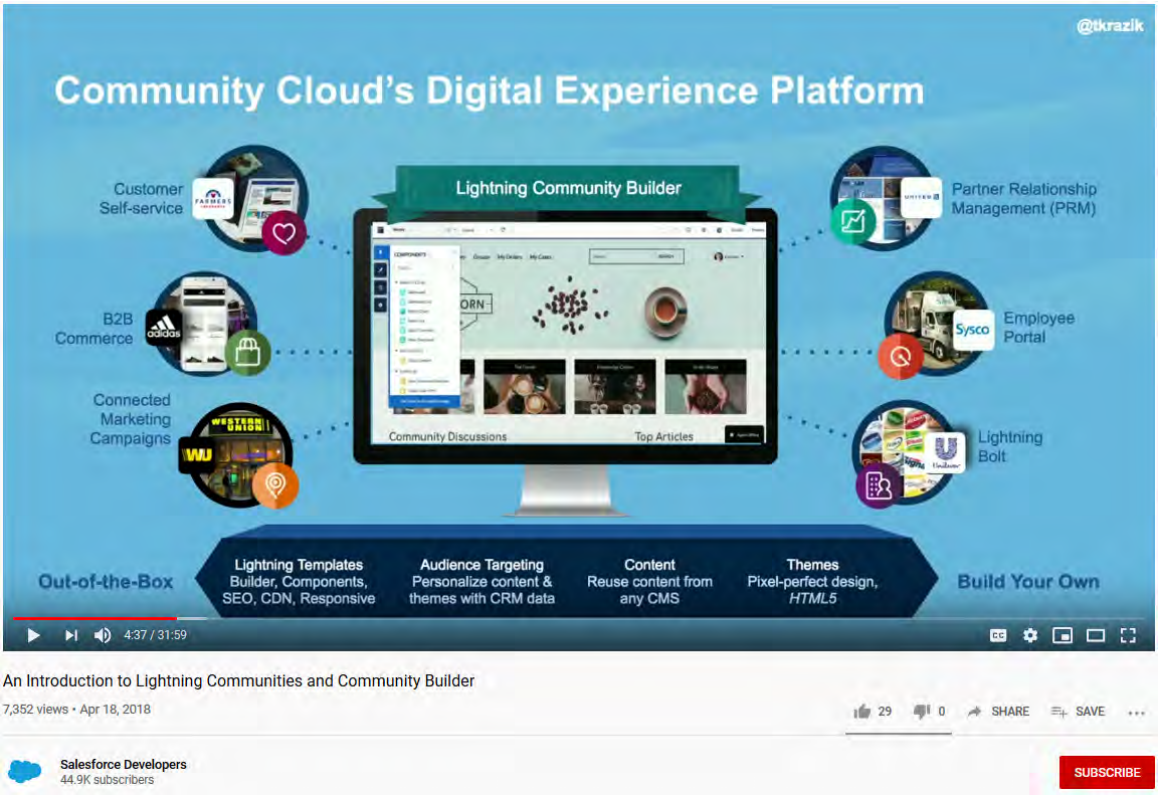
81. Upon information and belief, Salesforce has directly infringed at least claim 1 of the '168 patent by making and providing its users with a system for assembling a web site.

82. The Accused Instrumentality is configured for use with one of a number of browsers on a user's computer. The Accused Instrumentality performs a method to allow users to produce Internet websites on and for computers having a browser and a virtual machine capable of generating displays.

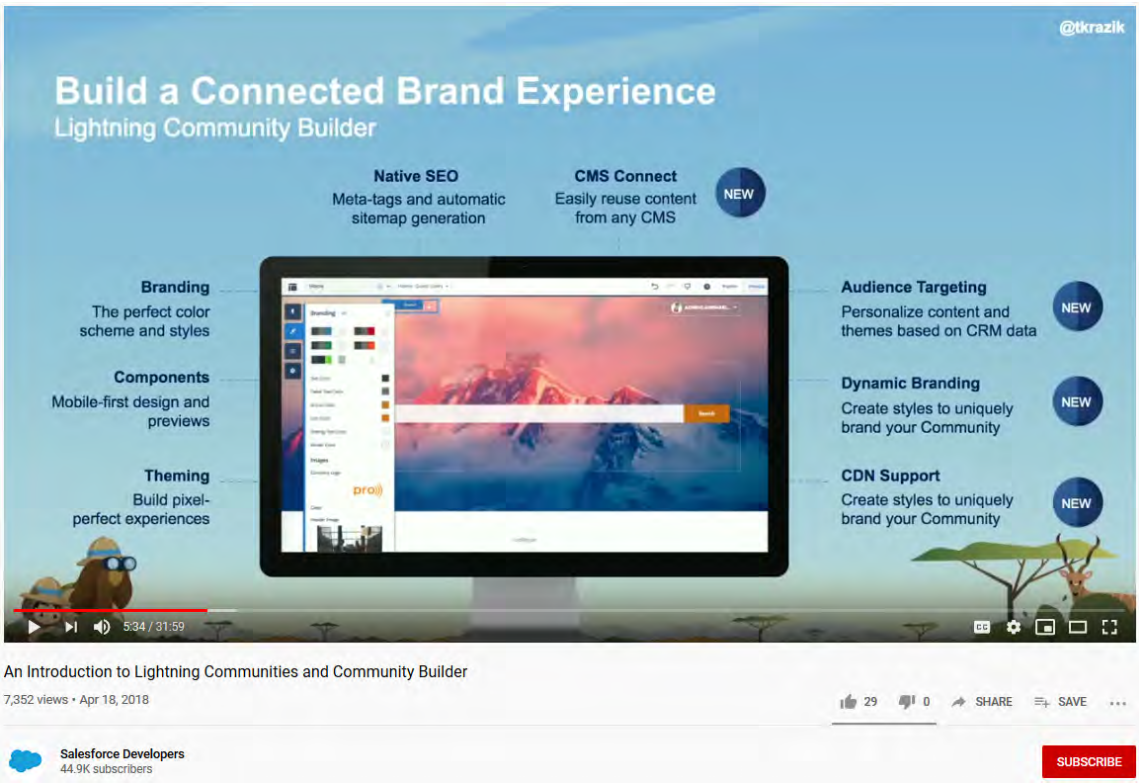


Source: https://www.youtube.com/watch?v=yBz0F9b3KLY

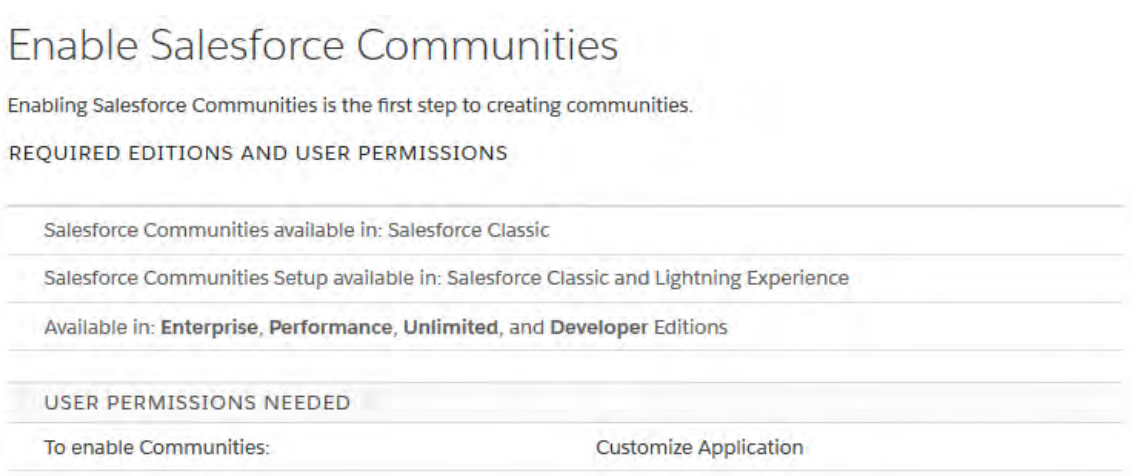
83. The Accused Instrumentality enables a user to produce a website through a browser on the user's computer interacting with the Salesforce's server facilities, hosted by Salesforce.com. For example, an enterprise may create an account on Salesforce.com and build a web site using the Lightning Community Builder.



Source: https://www.youtube.com/watch?v=yBz0F9b3KLY



Source: https://www.youtube.com/watch?v=yBz0F9b3KLY



Enable Salesforce Communities

Enabling Salesforce Communities is the first step to creating communities.

REQUIRED EDITIONS AND USER PERMISSIONS

Salesforce Communities available in: Salesforce Classic

Salesforce Communities Setup available in: Salesforce Classic and Lightning Experience

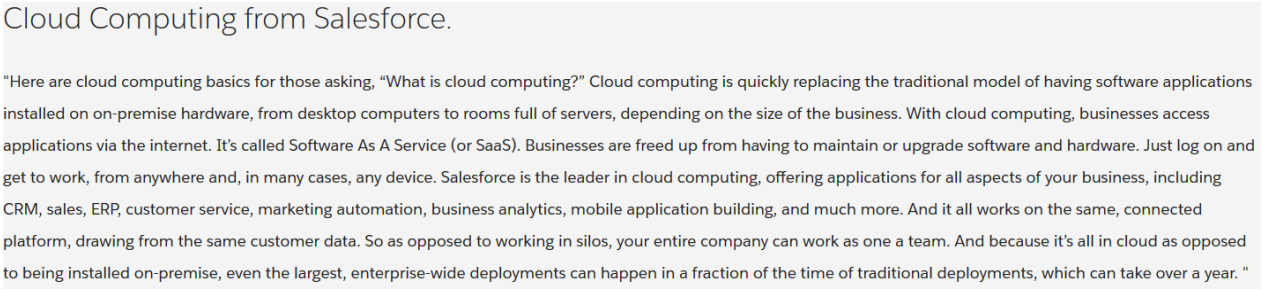
Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

USER PERMISSIONS NEEDED

To enable Communities:	Customize Application

Source: https://help.salesforce.com/articleView?id=networks_enable.htm&type=5

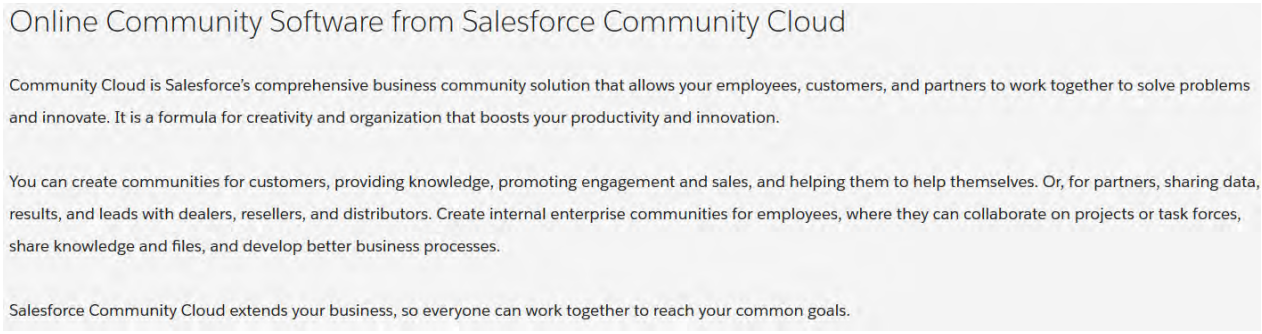
84. The Accused Instrumentality includes a server comprising a build engine. Salesforce is a cloud computing platform, which means that its users use their browsers to create and edit their websites using a build engine on Salesforce servers.



Cloud Computing from Salesforce.

"Here are cloud computing basics for those asking, "What is cloud computing?" Cloud computing is quickly replacing the traditional model of having software applications installed on on-premise hardware, from desktop computers to rooms full of servers, depending on the size of the business. With cloud computing, businesses access applications via the internet. It's called Software As A Service (or SaaS). Businesses are freed up from having to maintain or upgrade software and hardware. Just log on and get to work, from anywhere and, in many cases, any device. Salesforce is the leader in cloud computing, offering applications for all aspects of your business, including CRM, sales, ERP, customer service, marketing automation, business analytics, mobile application building, and much more. And it all works on the same, connected platform, drawing from the same customer data. So as opposed to working in silos, your entire company can work as one a team. And because it's all in cloud as opposed to being installed on-premise, even the largest, enterprise-wide deployments can happen in a fraction of the time of traditional deployments, which can take over a year. "

Source: https://www.salesforce.com/ap/cloudcomputing/



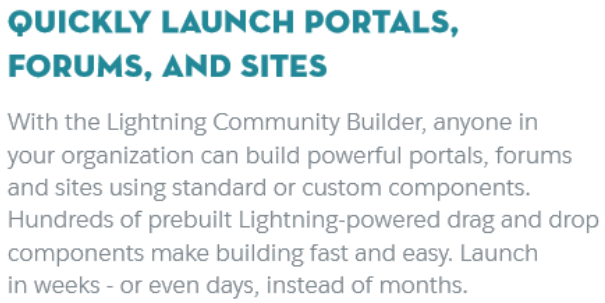
Online Community Software from Salesforce Community Cloud

Community Cloud is Salesforce's comprehensive business community solution that allows your employees, customers, and partners to work together to solve problems and innovate. It is a formula for creativity and organization that boosts your productivity and innovation.

You can create communities for customers, providing knowledge, promoting engagement and sales, and helping them to help themselves. Or, for partners, sharing data, results, and leads with dealers, resellers, and distributors. Create internal enterprise communities for employees, where they can collaborate on projects or task forces, share knowledge and files, and develop better business processes.

Salesforce Community Cloud extends your business, so everyone can work together to reach your common goals.

Source: https://www.salesforce.com/products/community-cloud/features/



Source:
https://c1.sfdcstatic.com/content/dam/web/en_us/www/documents/datasheets/community-cloud-datasheet.pdf

Browser Support for Communities

Browser support for communities varies by the device on which you access the community and the community template.

REQUIRED EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

Available in: **Enterprise**, **Performance**, **Unlimited**, and **Developer** Editions

Salesforce Tabs + Visualforce

Salesforce Tabs + Visualforce communities are supported on all desktop browsers that Salesforce supports. These communities are also accessible from the Salesforce app.

Lightning Communities

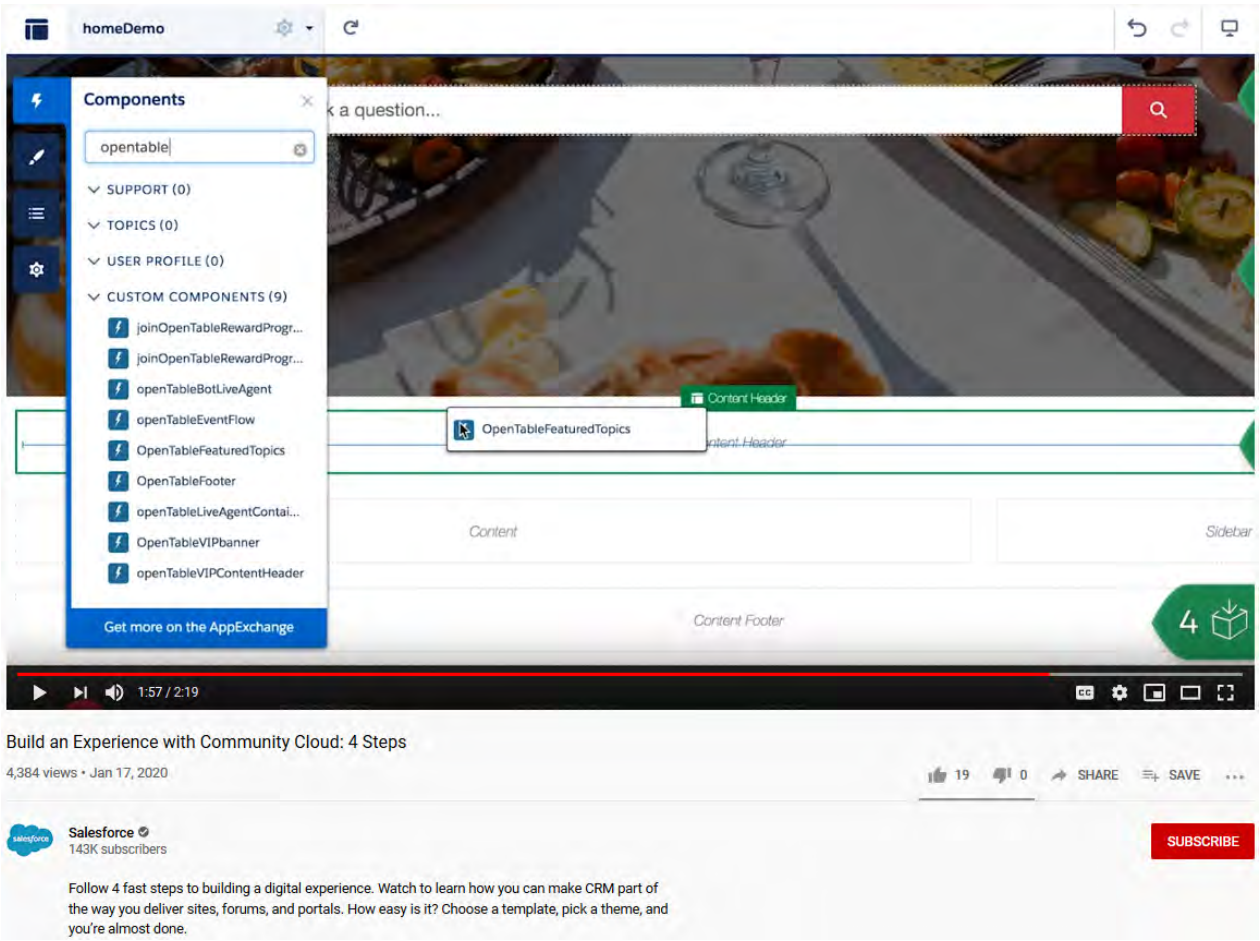
Lightning communities, such as Customer Service, Partner Central, and Help Center, are supported on the following browsers.

On Desktop Computers

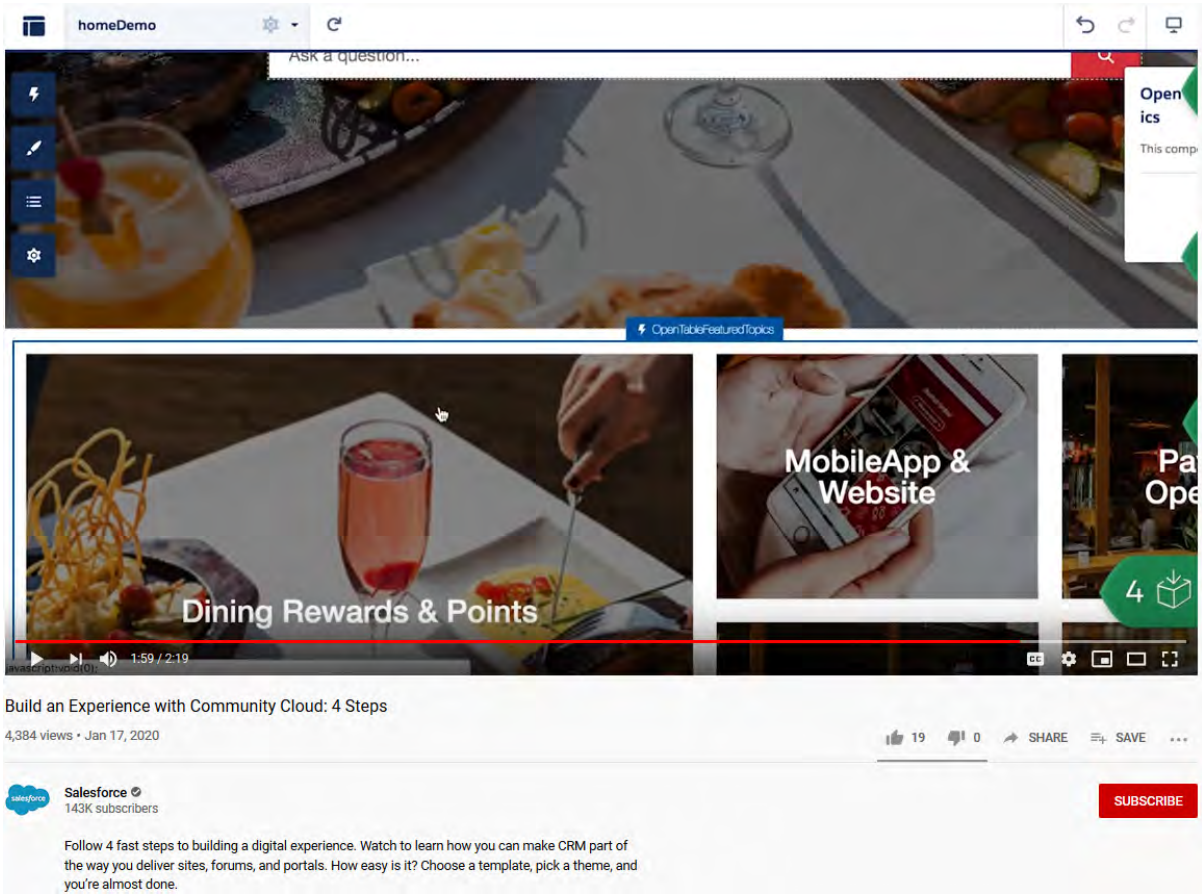
PLATFORMS	APPLE SAFARI (LATEST VERSION)	GOOGLE CHROME™ (LATEST VERSION)	MICROSOFT® EDGE (WINDOWS® 10 ONLY)	MICROSOFT® INTERNET EXPLORER® 11	MOZILLA® FIREFOX® (LATEST VERSION)
Apple® Mac OS® Desktop	✓	✓			✓
Microsoft® Windows® Desktop		✓	✓	✓(1)	✓

Source:
https://help.salesforce.com/articleView?id=networks_communities_browser_support.htm&type=5

85. The Accused Instrumentality accepts user input to create a web site, the web site comprising a plurality of web pages, each web page comprising a plurality of objects, and accept user input to associate a style with objects of the plurality of web pages. The Accused Instrumentality accepts user input to create a web site. For example, users may drag and drop components on to their web sites to build them.

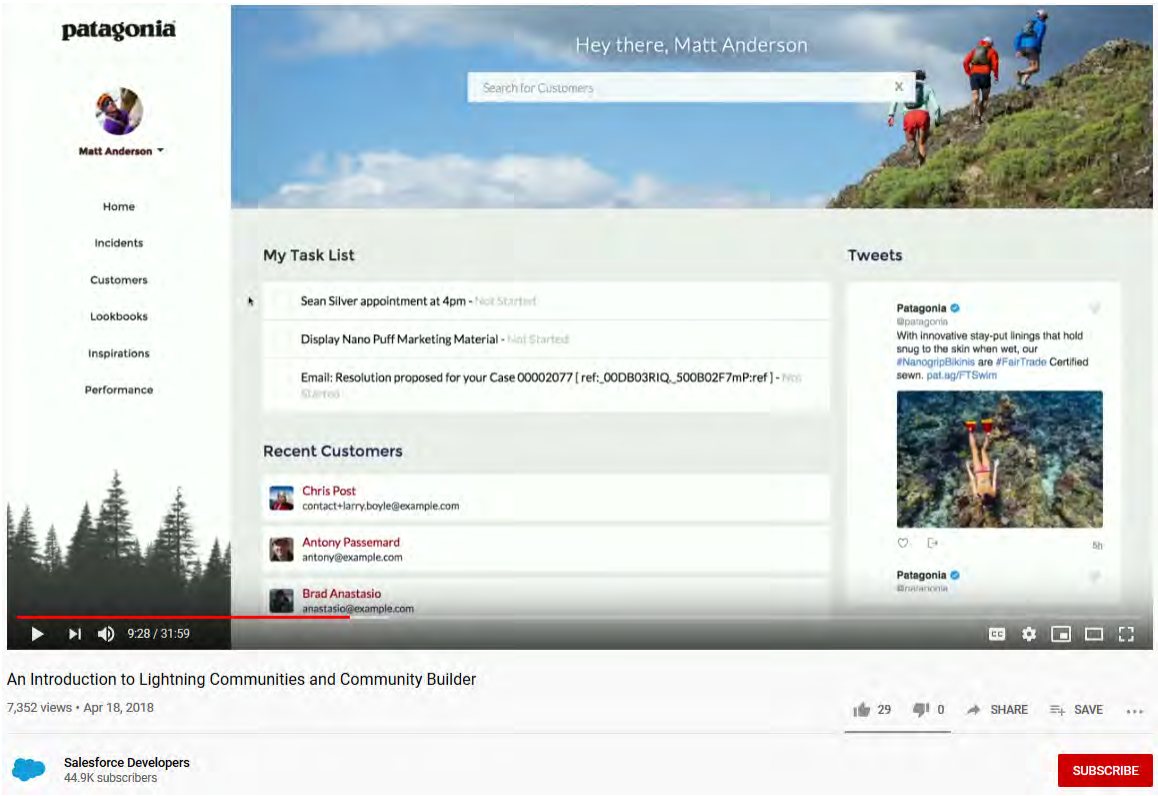


Source: https://www.youtube.com/watch?v=cIzoxQ_bWaM

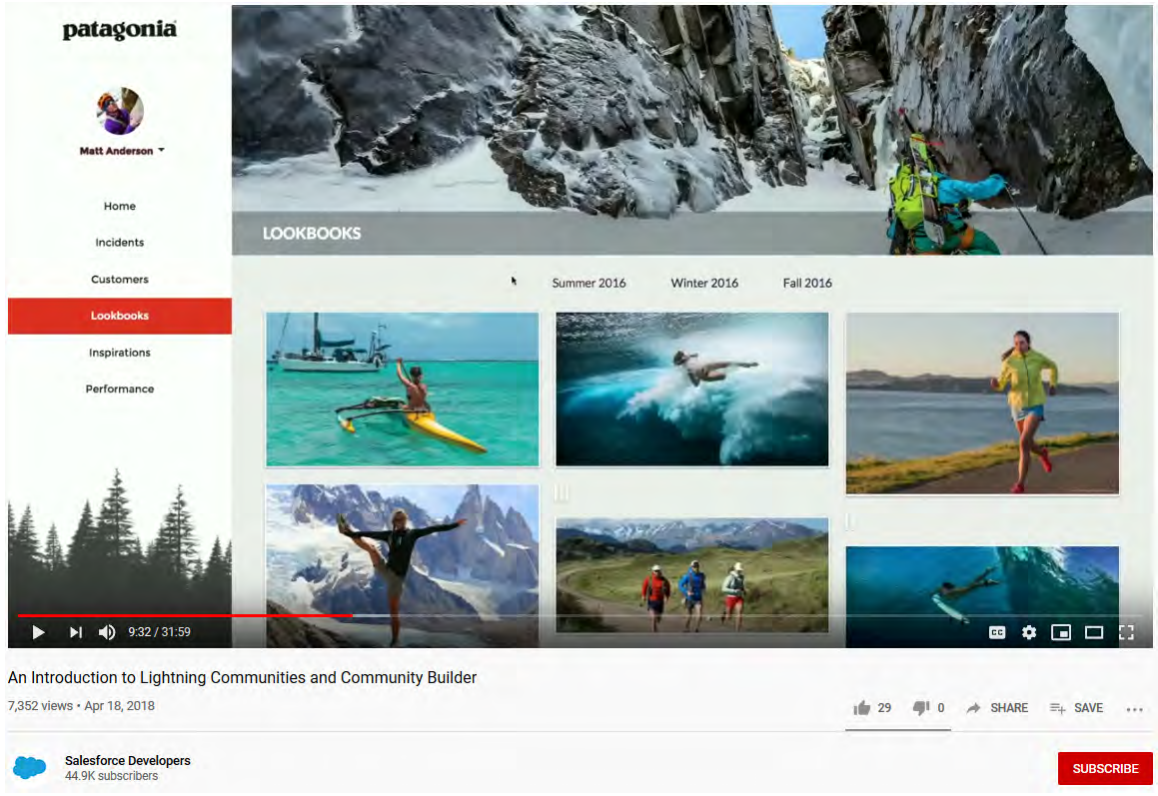


Source: https://www.youtube.com/watch?v=cIzoxQ_bWaM

86. Salesforce shows the Patagonia web site as an example of a site built with the Accused Instrumentality. The Patagonia site includes a plurality of pages, and each web page includes a plurality of objects, such as the navigation buttons on the left side, such as "Home," "Incidents," "Customers," "Lookbooks," "Inspirations," and "Performance."

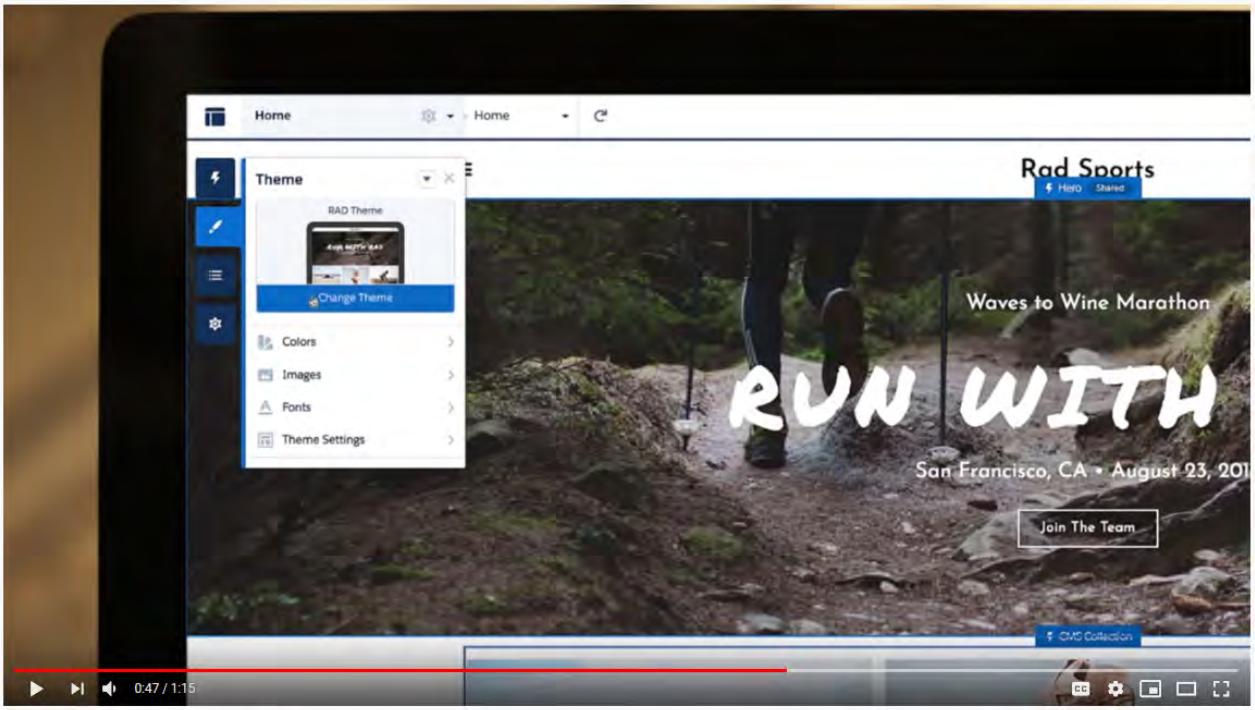


Source: https://www.youtube.com/watch?v=yBz0F9b3KLY

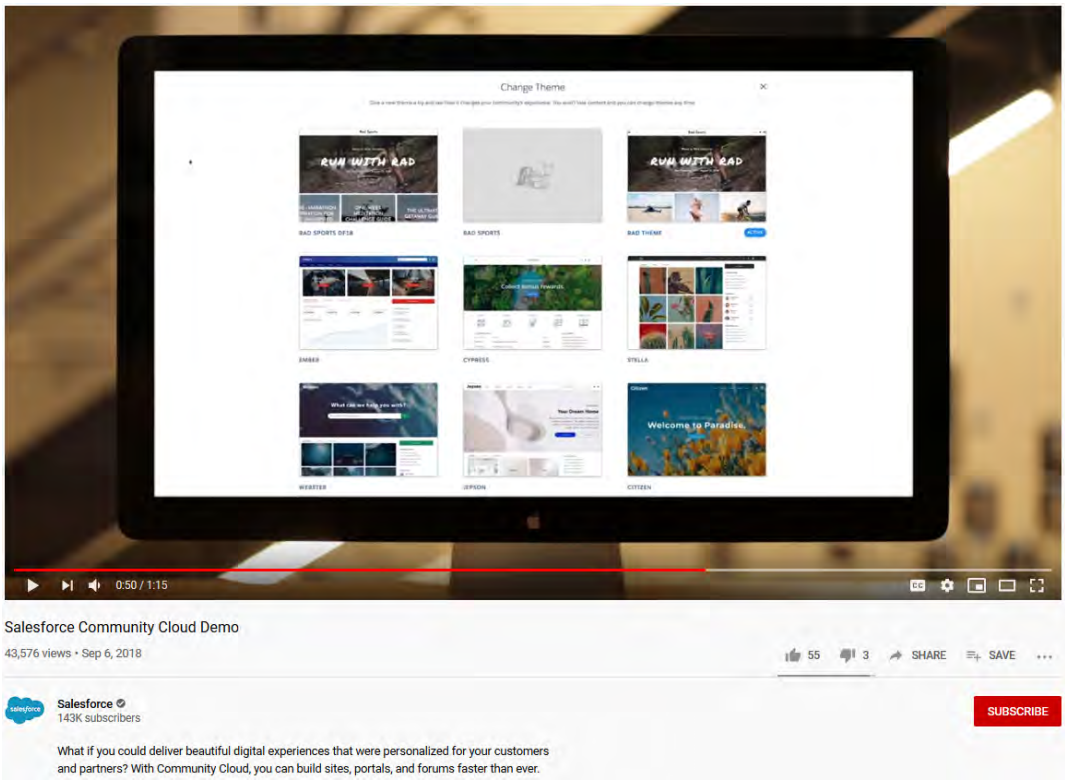


Source: https://www.youtube.com/watch?v=yBz0F9b3KLY

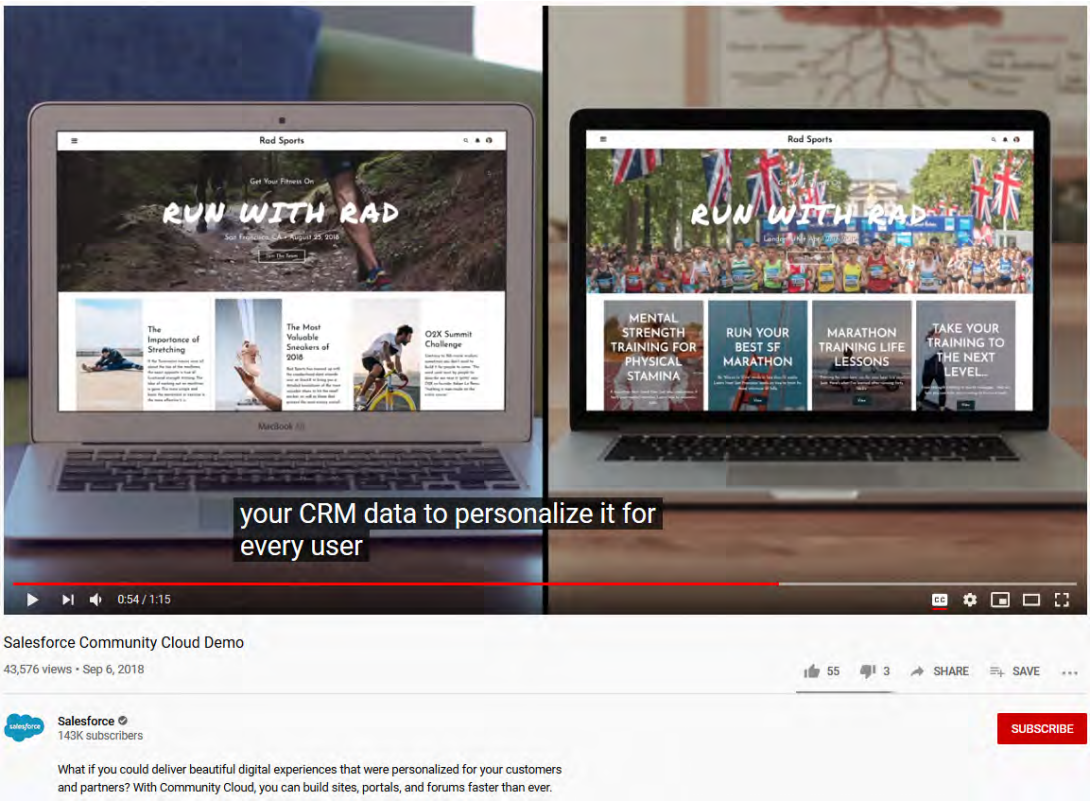
87. The Accused Instrumentality accepts user input to associate a style with objects of the plurality of web pages. For example, Salesforce advertises that its web pages may have their theme changed, which will change the style of objects, including being able to "personalize it for every user."



Source: https://www.youtube.com/watch?v=twjgE019p3o

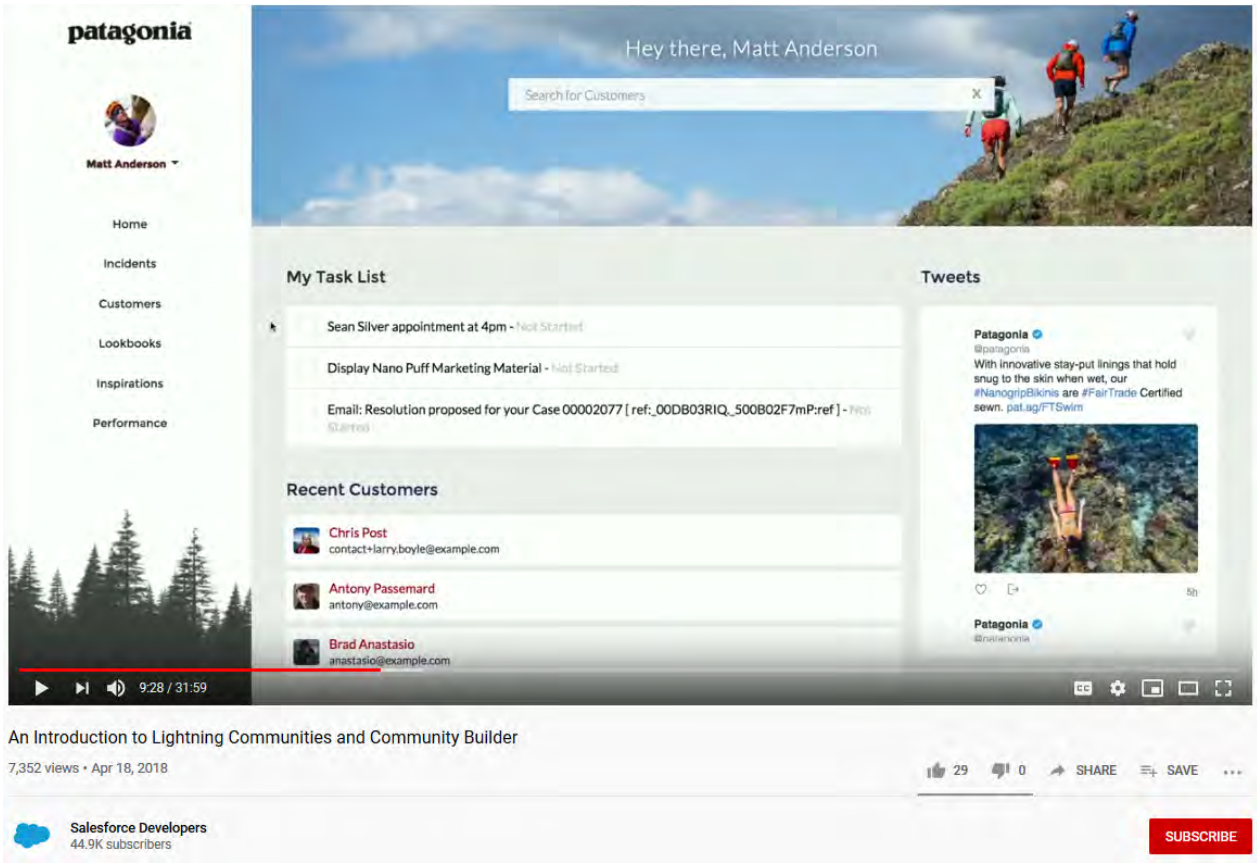


Source: https://www.youtube.com/watch?v=twjgE019p3o

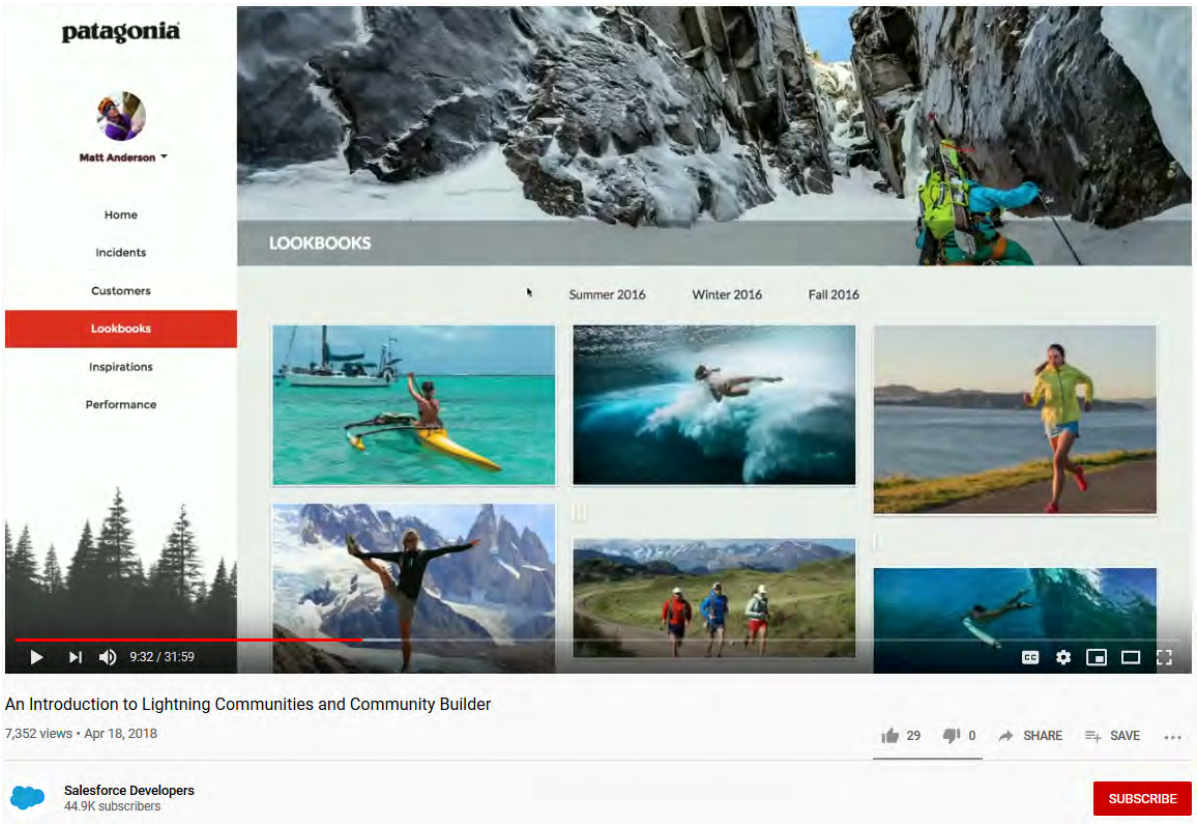


Source: https://www.youtube.com/watch?v=twjgE019p3o

88. In Community Cloud sites created by the Accused Instrumentality, each web page comprises at least one button object or at least one image object. For example, in the "Patagonia" example, in the Lightning Communities and Community Builder video, button objects are visible on the left side of both web pages, such as "Home," "Incidents," "Customers," "Lookbooks," "Inspirations," and "Performance."

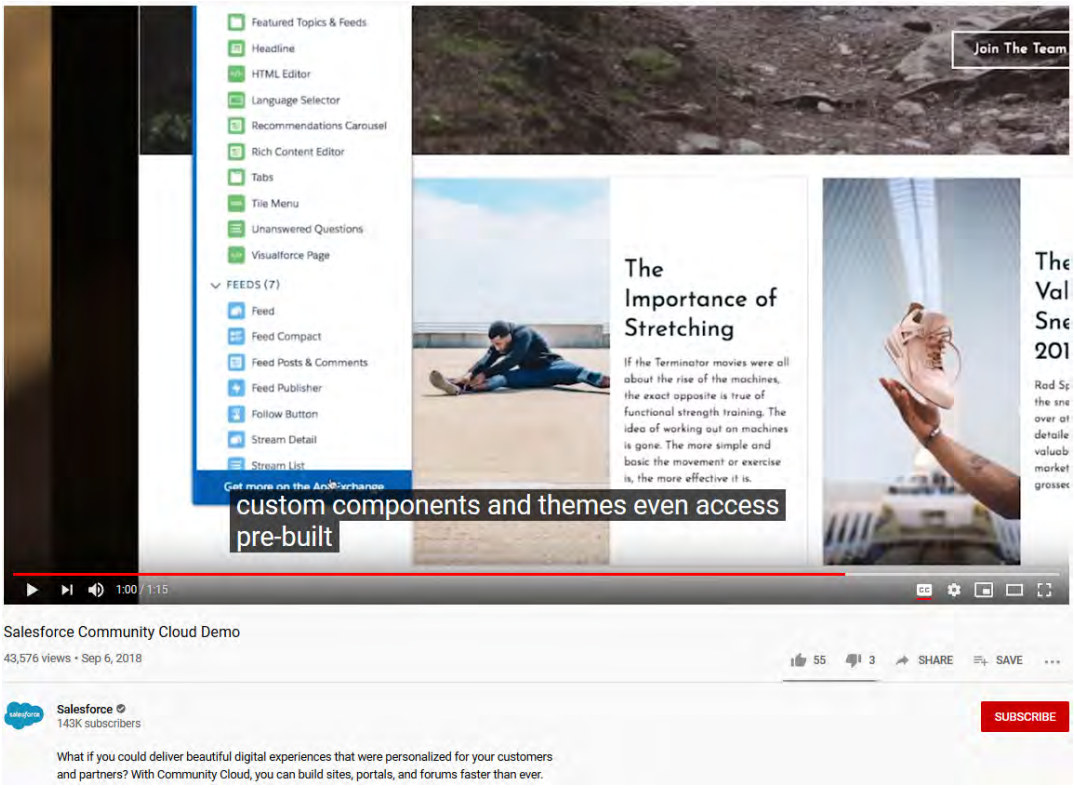


Source: https://www.youtube.com/watch?v=yBz0F9b3KLY

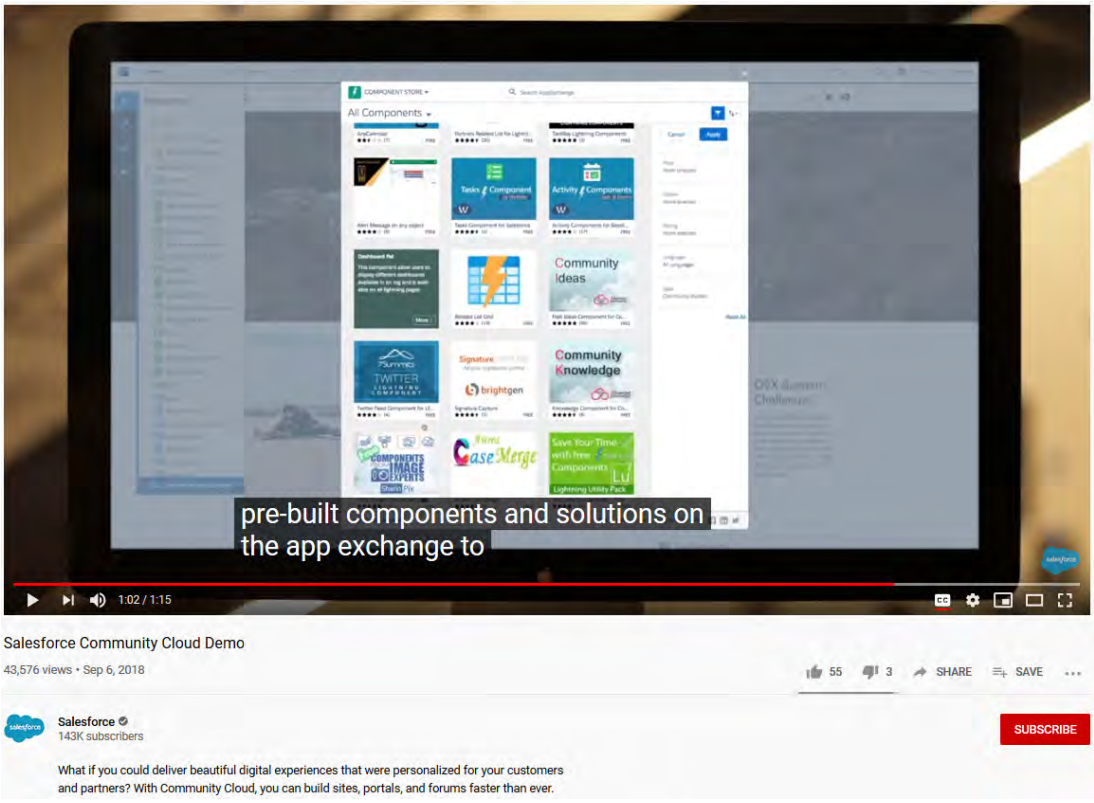


Source: https://www.youtube.com/watch?v=yBz0F9b3KLY

89. The Accused Instrumentality is configured to create sites including at least one button object or at least one image object that is associated with a style that includes values defining transformations and time lines for the at least one button object or at least one image object. The Accused Instrumentality allows users to change the theme of their websites. In addition, Salesforce advertises that users may create their own custom components, or use pre-built components on its "app exchange."

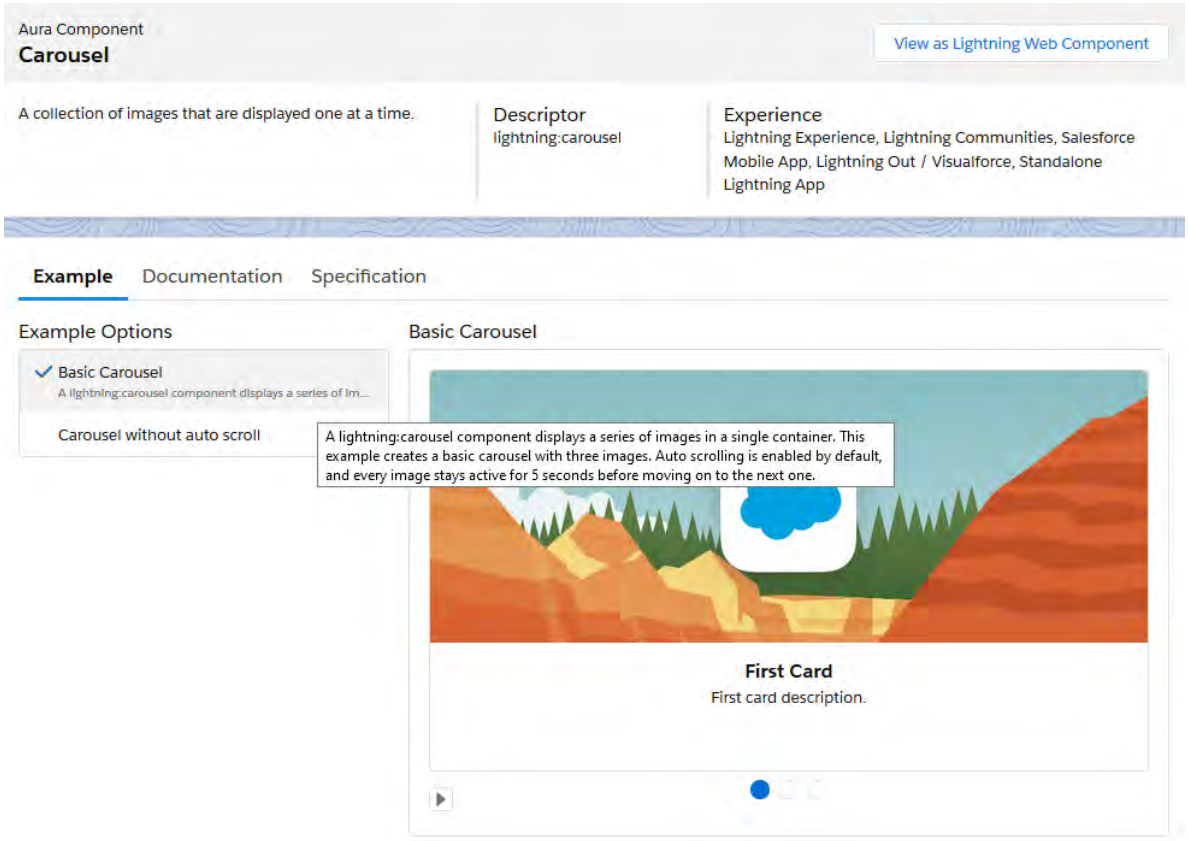


Source: https://www.youtube.com/watch?v=twjgE019p3o



Source: https://www.youtube.com/watch?v=twjgE019p3o

90. These components include objects that are able to be added to a user's site with explicit transformations and timelines, such as the "Carousel" component, which describes that "auto scrolling is enabled by default, and every image stays active for 5 seconds before moving on to the next one."



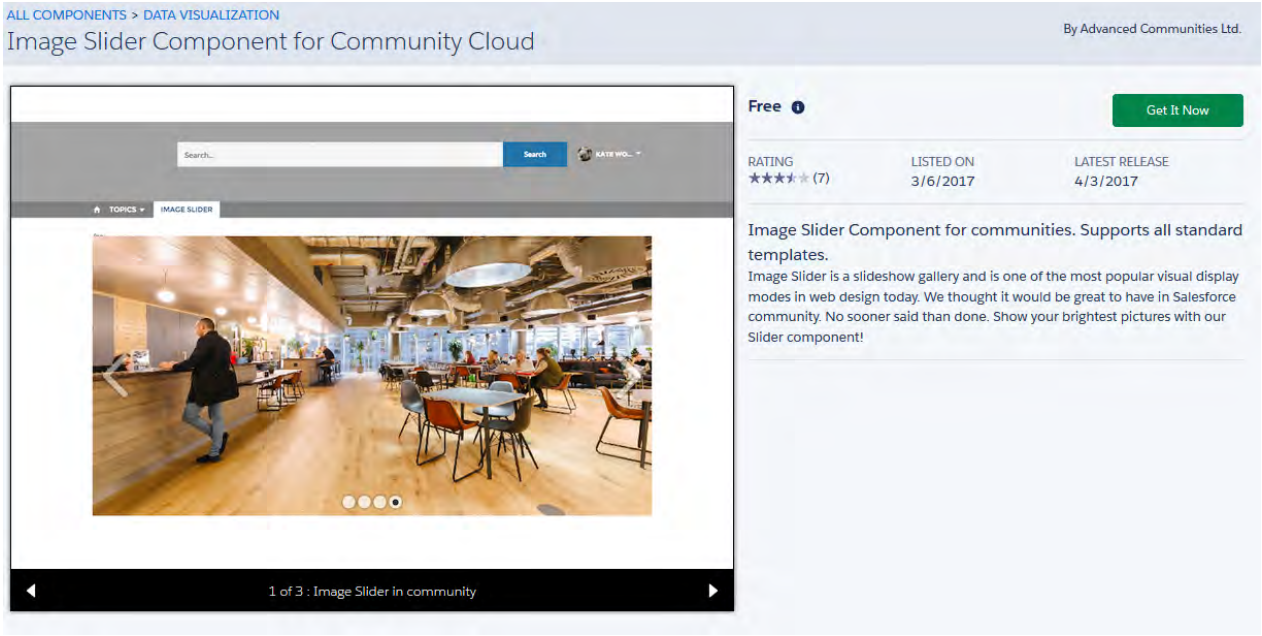
Source: https://developer.salesforce.com/docs/component-library/bundle/lightning:carousel/example#lightningcomponentdemo:exampleCarousel

91. This object has a style that defines the transformations and timelines for this object, such as the variables "disableAutoRefresh," "disableAutoScroll," and "scrollDuration," which affect how images in the carousel are displayed.

disableAutoRefresh	boolean	global	false	Specifies whether the carousel should stop looping from the beginning after the last item is displayed. The default value is false.
disableAutoScroll	boolean	global	false	Specifies whether auto scroll is disabled. The default value is false.
scrollDuration	integer	global	5	The auto scroll duration. The default is 5 seconds, after that the next image is displayed.

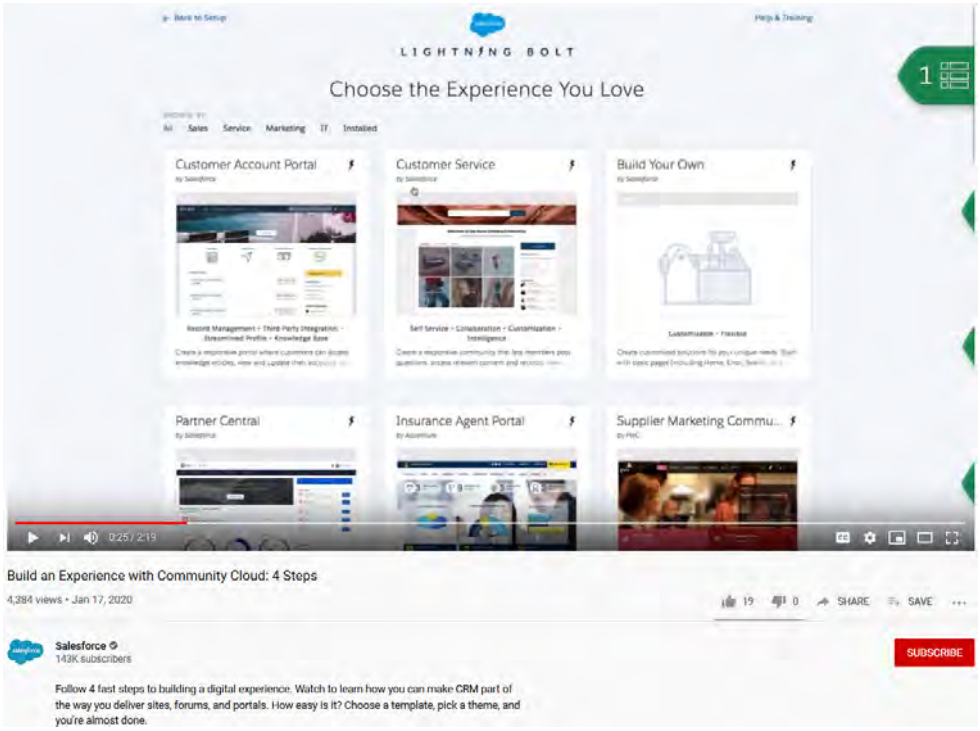
Source: https://developer.salesforce.com/docs/component-library/bundle/lightning:carousel/specification

92. More components that feature transformations and timelines are also available from the "App Exchange," such as an "Image Slider Component."



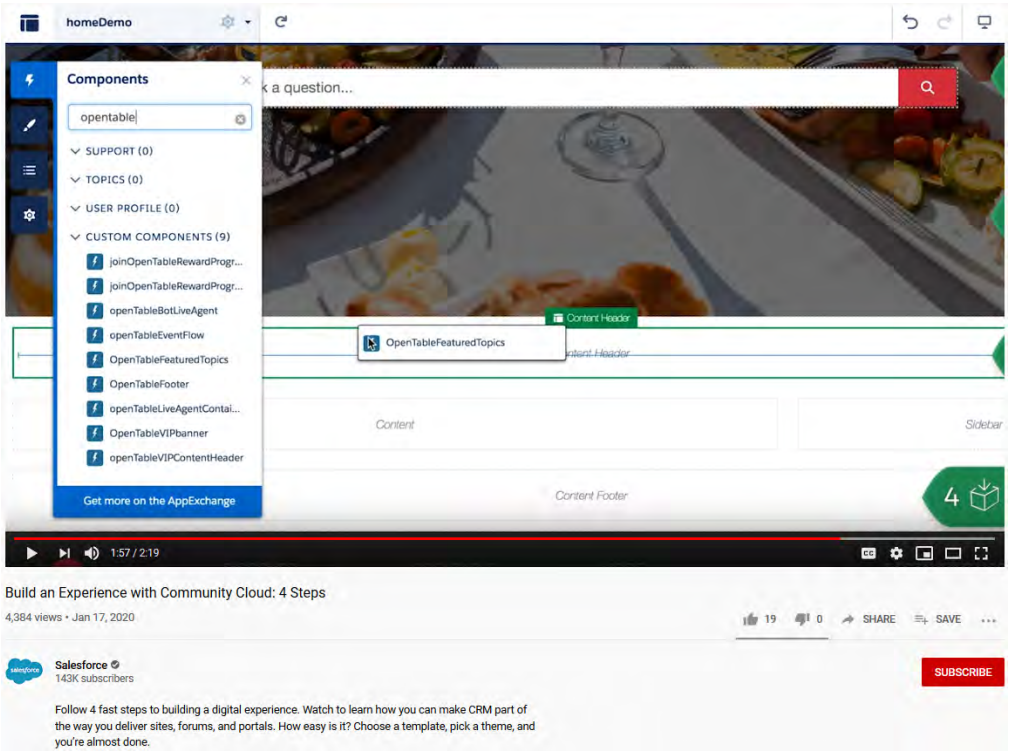
Source:
https://appexchange.salesforce.com/appxListingDetail?listingId=a0N3A00000EOLfBUAX

93. Each web page is defined entirely by each of the plurality of objects comprising that web page and the style associated with the object. For example, one option for creating web sites is to use "Lightning Bolt" templates comprised of objects from the Salesforce libraries.



Source: https://www.youtube.com/watch?v=cIzoxQ_bWaM

94. Users may also drag and drop objects to create the web site.



Source: https://www.youtube.com/watch?v=cIzoxQ_bWaM

95. Using the Document Object Model, modern browsers parse the HTML code that comprises web pages into objects.

96. The Accused Instrumentality produces a database with a multidimensional array comprising the objects that comprise the web site including data defining, for each object, the object style, an object number, and an indication of the web page that each object is part of. For example, Salesforce describes its Community Cloud as a "window into CRM," which is the database and backend technologies that support the website.

> The Document Object Model (DOM) connects web pages to scripts or programming languages by representing the structure of a document—such as the HTML representing a web page—in memory. Usually, that means JavaScript, although modeling HTML, SVG, or XML documents as objects are not part of the core JavaScript language, as such.
>
> The DOM represents a document with a logical tree. Each branch of the tree ends in a node, and each node contains objects. DOM methods allow programmatic access to the tree. With them, you can change the document's structure, style, or content.

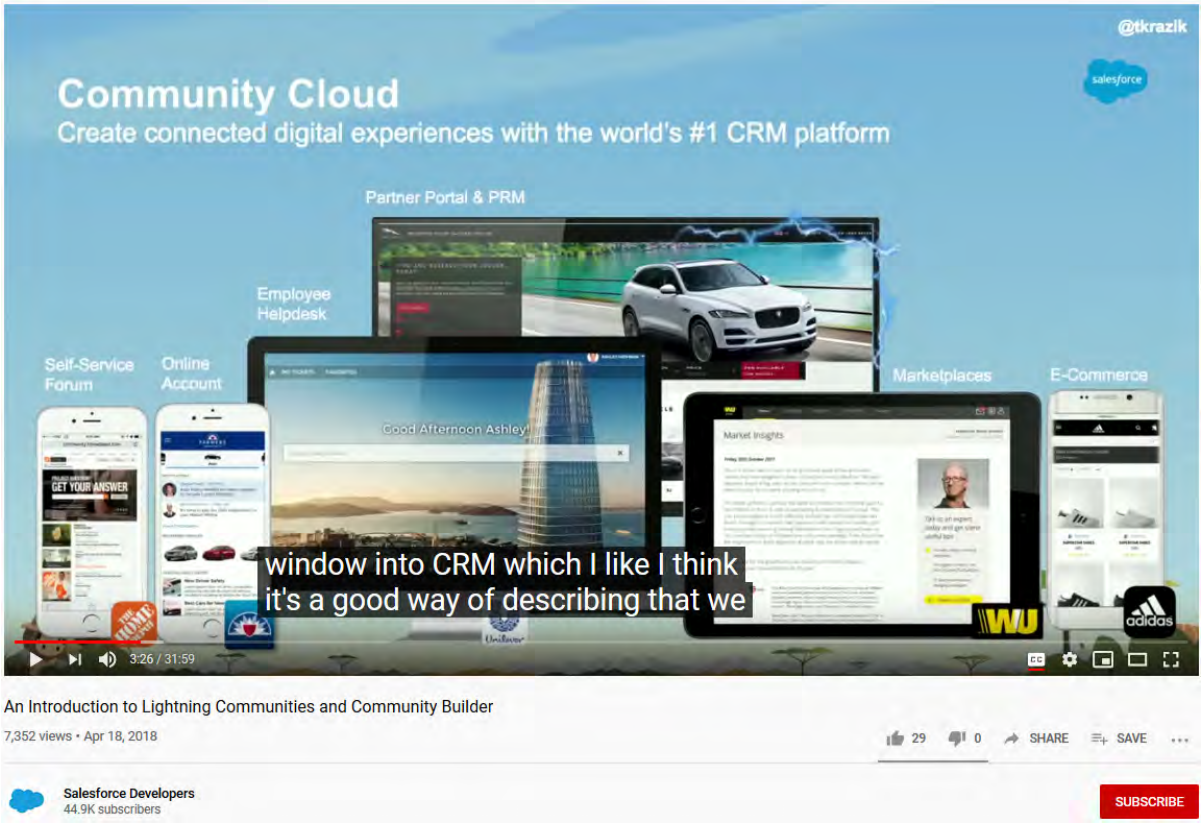
https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model

The HTML DOM is a standard **object** model and **programming interface** for HTML. It defines:

- The HTML elements as objects
- The properties of all HTML elements
- The methods to access all HTML elements
- The **events** for all HTML elements

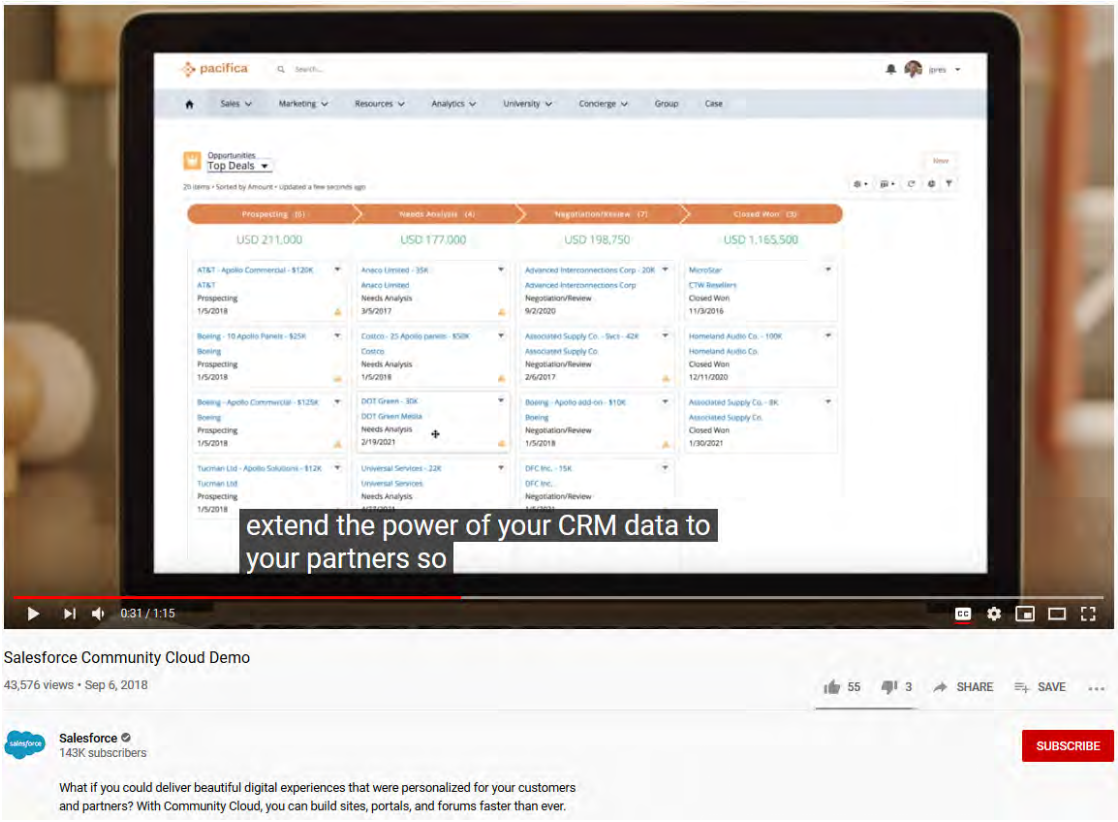
In other words: The HTML DOM is a standard for how to get, change, add, or delete HTML elements.

https://www.w3schools.com/js/js_htmldom.asp

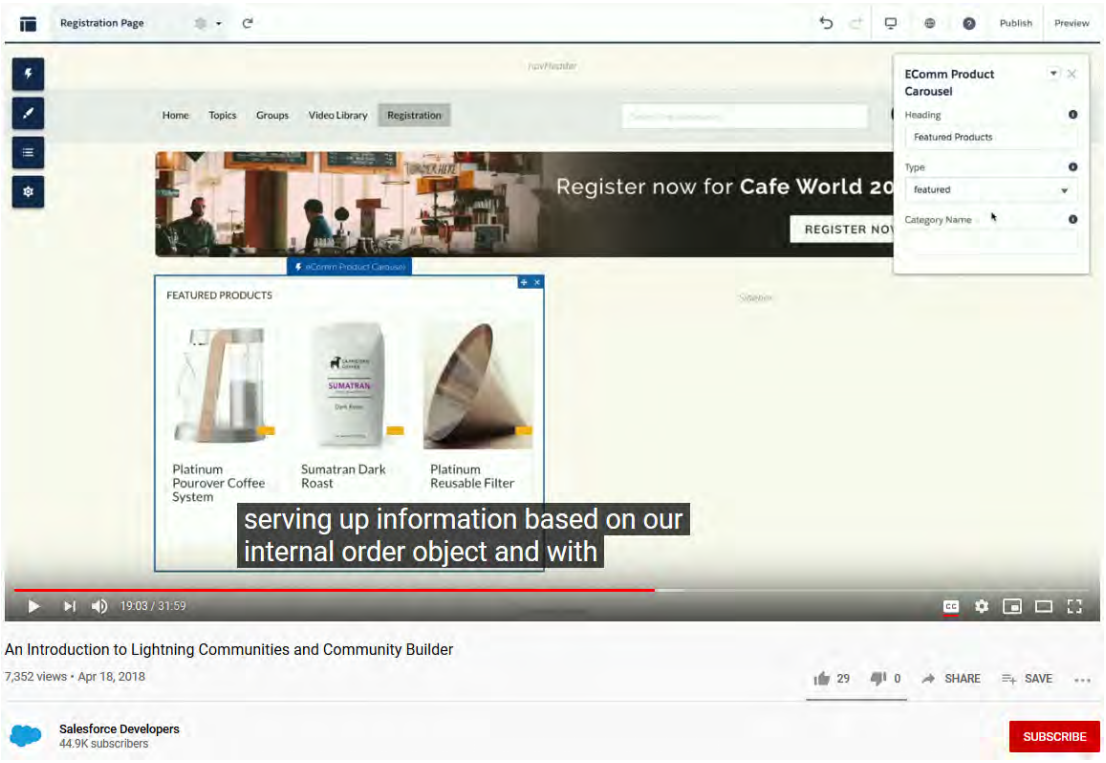


Source: https://www.youtube.com/watch?v=yBz0F9b3KLY

97. The data that populates objects is saved within Salesforce's Database.com and Oracle databases (*see, e.g.* https://www.computerworld.com/article/2514750/salesforce-com-unveils-database-com.html) because it can "extend the power of your CRM data to your partners" as shown in the Community Cloud Demo, below, and relies on information such as "internal order objects" to populate the objects, as shown in the "An Introduction to Lightning Communities and Community Builder."



Source: https://www.youtube.com/watch?v=twjgE019p3o



Source: https://www.youtube.com/watch?v=yBz0F9b3KLY

98. The settings for the styles and themes are also saved in Salesforce databases. For example, code for the style of components such as the Carousel is saved within Salesforce, including the settings the user has chosen, such as whether AutoRefresh is enabled (whether the carousel will continue once it has reached the end of its images).

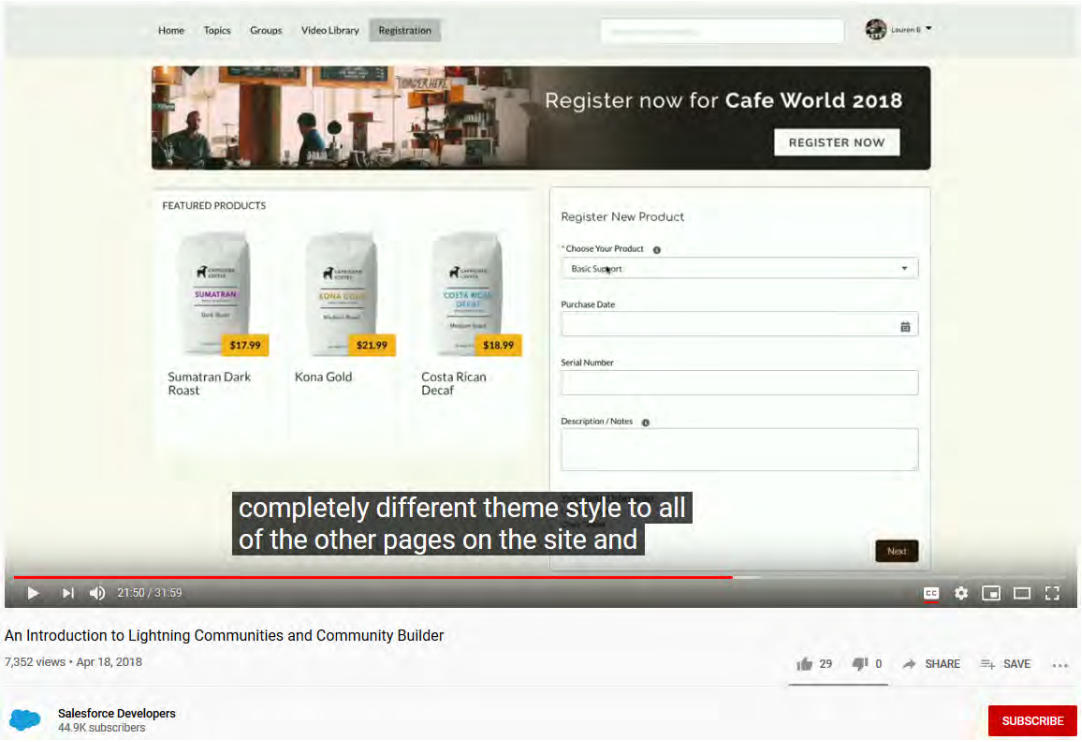
Code

Component

```
<aura:component>
    <lightning:carousel disableAutoRefresh="false" disableAutoScroll="false">
        <lightning:carouselImage
            src = "/docs/component-library/app/images/examples/carousel-01.png"
            header = "First Card"
            description = "First card description."
            alternativeText = "First card accessible description."
            href = "https://www.salesforce.com">
        </lightning:carouselImage>
        <lightning:carouselImage
            src = "/docs/component-library/app/images/examples/carousel-02.png"
            header = "Second Card"
            description = "Second card description."
            alternativeText = "Second card accessible description."
            href = "https://www.salesforce.com">
        </lightning:carouselImage>
        <lightning:carouselImage
            src = "/docs/component-library/app/images/examples/carousel-03.png"
            header = "Third Card"
            description = "Third card description."
            alternativeText = "Third card accessible description."
            href = "https://www.salesforce.com">
        </lightning:carouselImage>
    </lightning:carousel>
</aura:component>
```

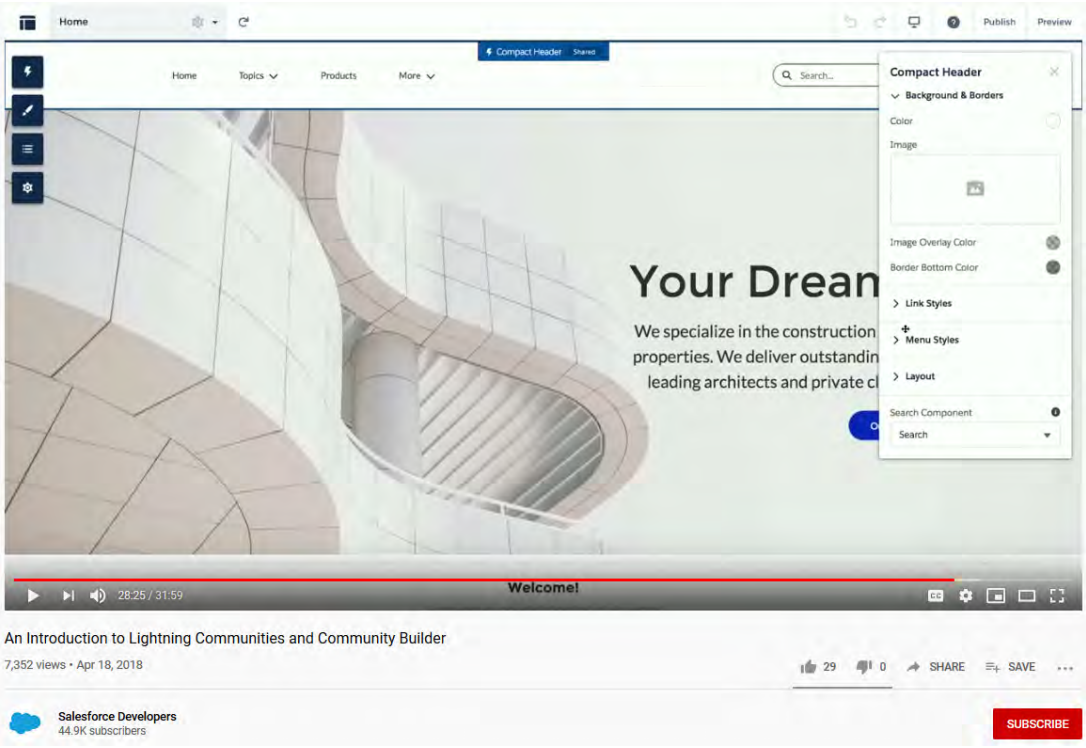
Source: https://developer.salesforce.com/docs/component-library/bundle/lightning:carousel/example

99. Styles are saved in relation to objects and pages. Salesforce describes that users can apply one theme to one page, and "can apply a completely different theme style to all of the other pages on the site."



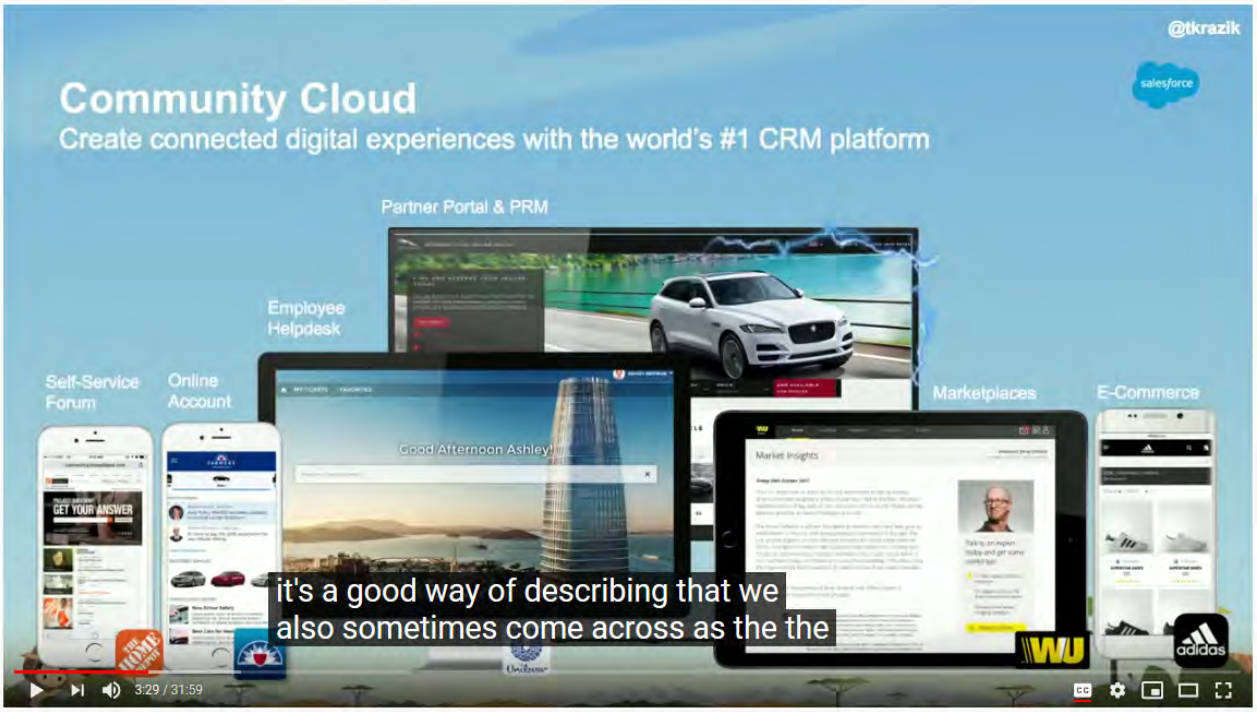
Source: https://www.youtube.com/watch?v=yBz0F9b3KLY

100. When a user edits an object, those edits are saved in Salesforce's databases.



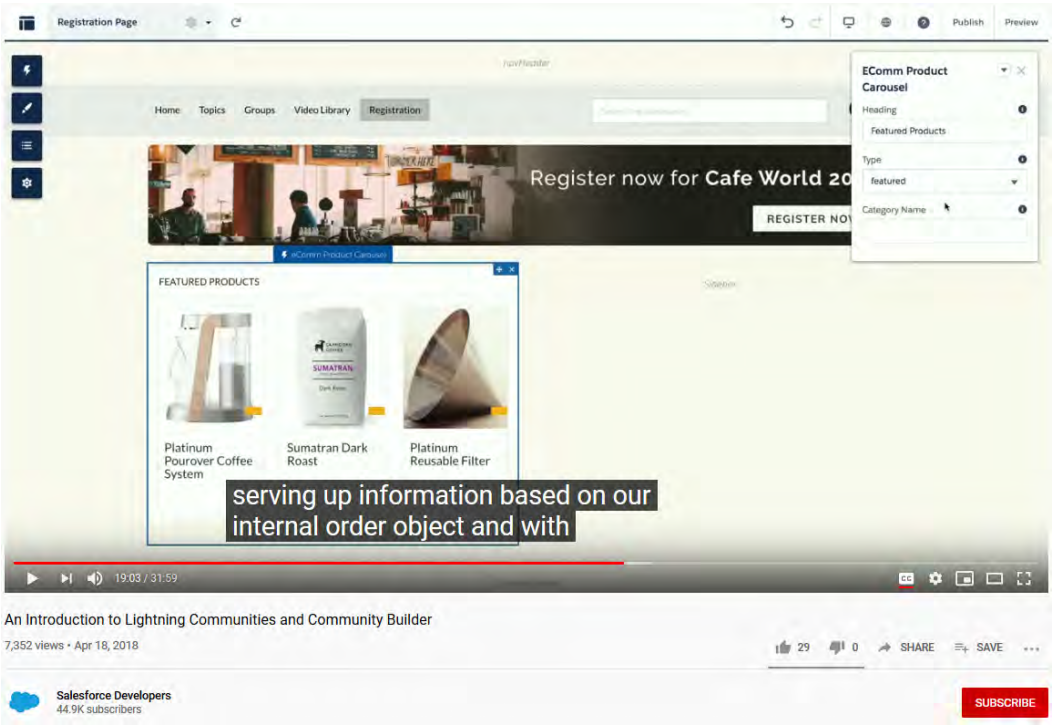
Source: https://www.youtube.com/watch?v=yBz0F9b3KLY

101. The Accused Instrumentality provides the database to a server accessible to a web browser. For example, Salesforce's servers host websites for its users over the Internet that include web pages created using the Accused Instrumentality. These websites are accessible to web visitors through a web browser.



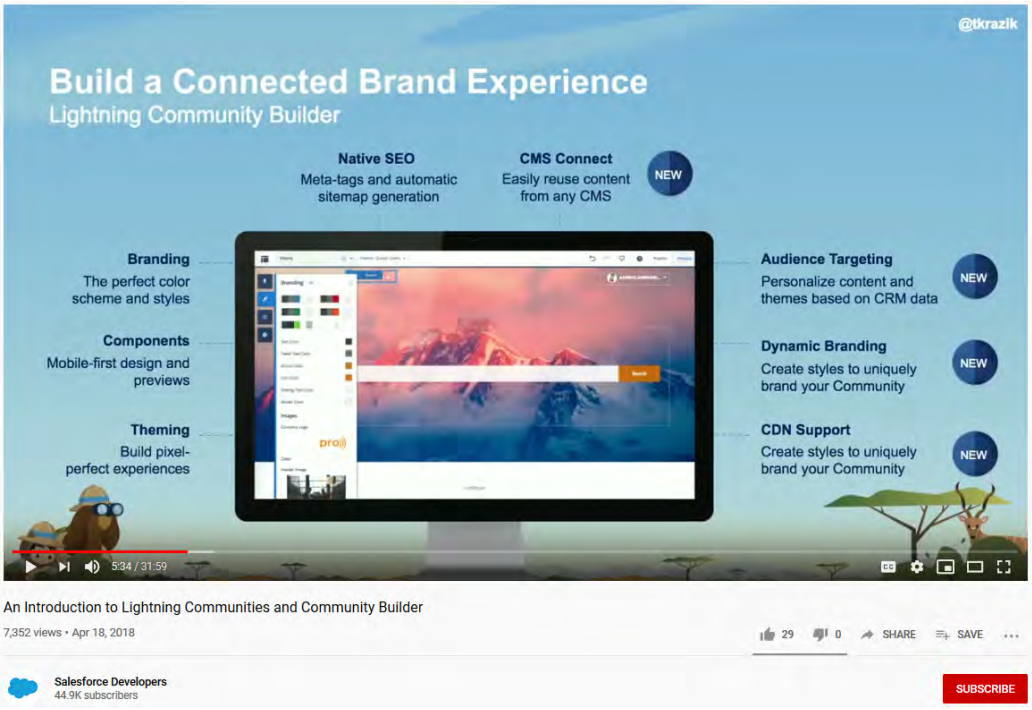
Source: https://www.youtube.com/watch?v=yBz0F9b3KLY

102. The database produced by the Accused Instrumentality enables a web browser with access to a runtime engine to generate the web-site from the objects and style data extracted from the provided database. Salesforce describes that its pages are generated by exposing data from its databases, such as its "internal order objects."



Source: https://www.youtube.com/watch?v=yBz0F9b3KLY

103. Salesforce also describes how its audience targeting functions can "personalize content and themes based on CRM data."



Source: https://www.youtube.com/watch?v=yBz0F9b3KLY

104. Modern web browsers all include a runtime engine for generating web sites. Modern web browsers rely on browser engines, to interpret and execute JavaScript and HTML to render web pages on a computer. Internet Explorer has relied on Trident (code name for MSHTML) (which has included the Chakra JavaScript Engine, see http://en.wikipedia.org/wiki/MSHTML); Edge relies on EdgeHTML (which also includes the Chakra JavaScript Engine, see https://docs.microsoft.com/en-us/microsoft-edge/dev-guide) Safari and Chrome rely on Webkit (which includes WebCore and JavaScript Core, see http://en.wikipedia.org/wiki/WebKit); Firefox relies on Gecko (which includes Spidermonkey, see http://www.mozilla.org/projects/technologies.html). The browser constructs the website according to the Document Object Model. Using the Document Object Model, modern browsers parse the HTML code that comprises web pages into objects.

Internet Explorer has relied on Trident (code name for MSHTML) (which has included the Chakra JavaScript Engine, see http://en.wikipedia.org/wiki/MSHTML); Edge relies on EdgeHTML (which also includes the Chakra JavaScript Engine, see https://docs.microsoft.com/en-us/microsoft-edge/dev-guide) Safari and Chrome rely on Webkit (which includes WebCore and JavaScript Core, see http://en.wikipedia.org/wiki/WebKit); Firefox relies on Gecko (which includes Spidermonkey, see http://www.mozilla.org/projects/technologies.html).

As shown above, the browser constructs the website according to the Document Object Model. Using the Document Object Model, modern browsers parse the HTML code that comprises web pages into objects.

> The Document Object Model (DOM) connects web pages to scripts or programming languages by representing the structure of a document—such as the HTML representing a web page—in memory. Usually, that means JavaScript, although modeling HTML, SVG, or XML documents as objects are not part of the core JavaScript language, as such.
>
> The DOM represents a document with a logical tree. Each branch of the tree ends in a node, and each node contains objects. DOM methods allow programmatic access to the tree. With them, you can change the document's structure, style, or content.

https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model

The HTML DOM is a standard **object** model and **programming interface** for HTML. It defines:

- The HTML elements as objects
- The properties of all HTML elements
- The methods to access all HTML elements
- The **events** for all HTML elements

In other words: The HTML DOM is a standard for how to get, change, add, or delete HTML elements.

https://www.w3schools.com/js/js_htmldom.asp

105. Salesforce was made aware of the '168 patent and its infringement thereof at least as early as November 30, 2020 when Express Mobile provided notice of Salesforce's infringement of the '168 patent to Marc Benioff, Chair and Chief Executive Officer and Amy Weaver, President and Chief Legal Officer of Salesforce. Since at least the time Salesforce received notice, Salesforce has induced others to infringe at least one claim of the '168 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Salesforce's clients, customers, and end users, whose use of the Accused Instrumentality constitutes direct infringement of at least one claim of the '168 patent. In particular, Salesforce's actions that aid and abet others such as customers and end users to infringe include advertising and distributing the Accused Instrumentality and providing instruction materials, training, and services regarding the Accused Instrumentality. *See, e.g.* www.salesforce.com, help.salesforce.com, https://trailhead.salesforce.com, including all related domains and subdomains. Salesforce has engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Salesforce has had actual knowledge of the '168 patent and knowledge that its acts were inducing infringement of the '168 patent since at least the date Salesforce received notice that such activities infringed the '168 patent.

106. Salesforce is liable as a contributory infringer of the '168 patent under 35 U.S.C. § 271(c) by offering to sell, selling and importing into the United States website or web page authoring tools to be especially made or adapted for use in an infringement of the '168 patent. The Accused Instrumentality is a material component for use in practicing the '168 patent, is specifically made and is not a staple article of commerce suitable for substantial non-infringing use.

107. Salesforce's infringement has damaged and continues to damage and injure Express Mobile.

COUNT III - INFRINGEMENT OF U.S. PATENT NO. 9,928,044

108. Plaintiff incorporates by reference the allegations contained in paragraphs 1 to 107 above.

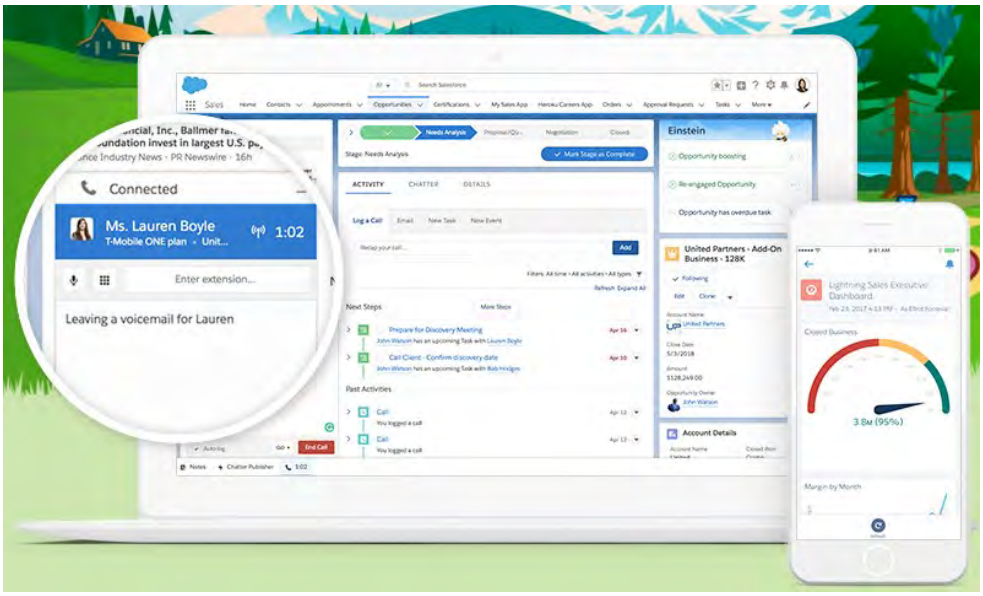
109. Salesforce has manufactured, sold, offered to sell, used and/or provided and continues to manufacture, sell, offer for sale, use and/or provide the Lightning App Builder (the "Accused Instrumentality") that infringes, either literally or under the doctrine of equivalents, one or more claims of the '044 patent in violation of 35 U.S.C. § 271(a).

110. Upon information and belief, Salesforce has directly infringed at least claim 1 of the '044 patent through the Accused Instrumentality that generates code to provide content on a display of a device.

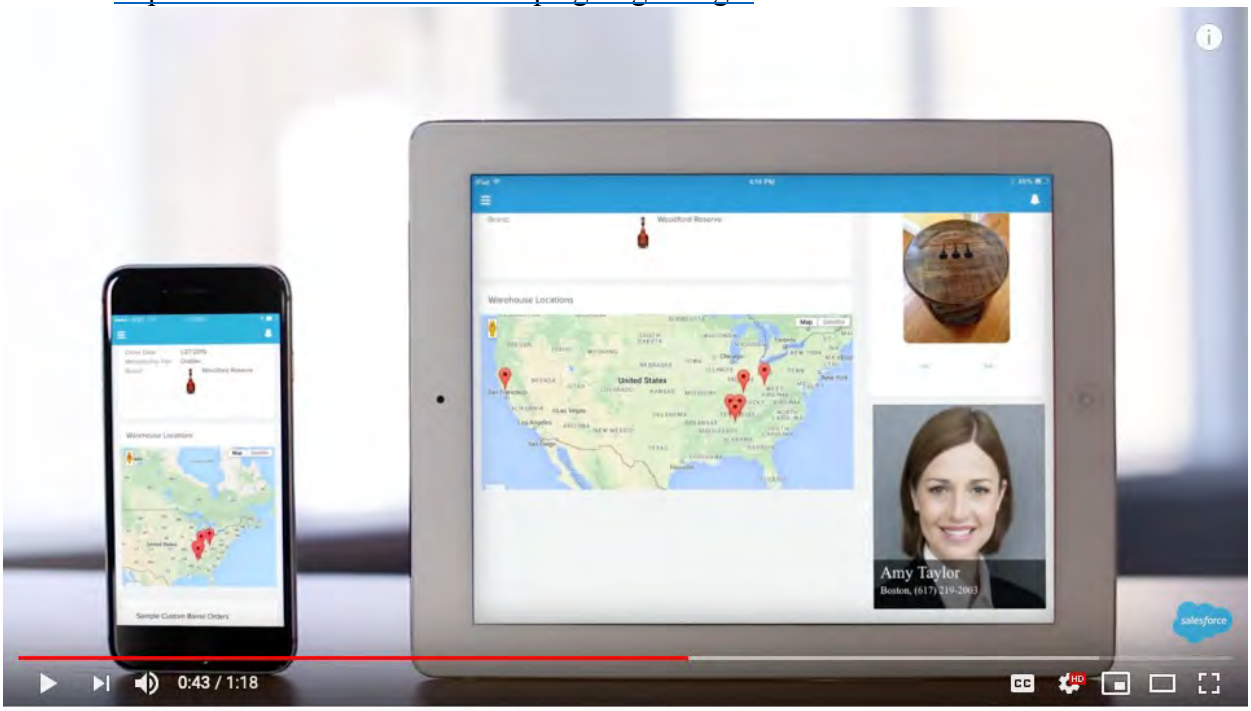
111. The Accused Instrumentality is a system for generating code to provide content on the display of a device for each of its users.

Source: https://www.salesforce.com/form/demo/platform-lightning-demo-sem/

112. The Accused Instrumentality includes memory to store user selections and other information relating to a user's web site, web pages, and the web components or elements that are intended to be displayed.



Source: https://www.salesforce.com/campaign/lightning/#



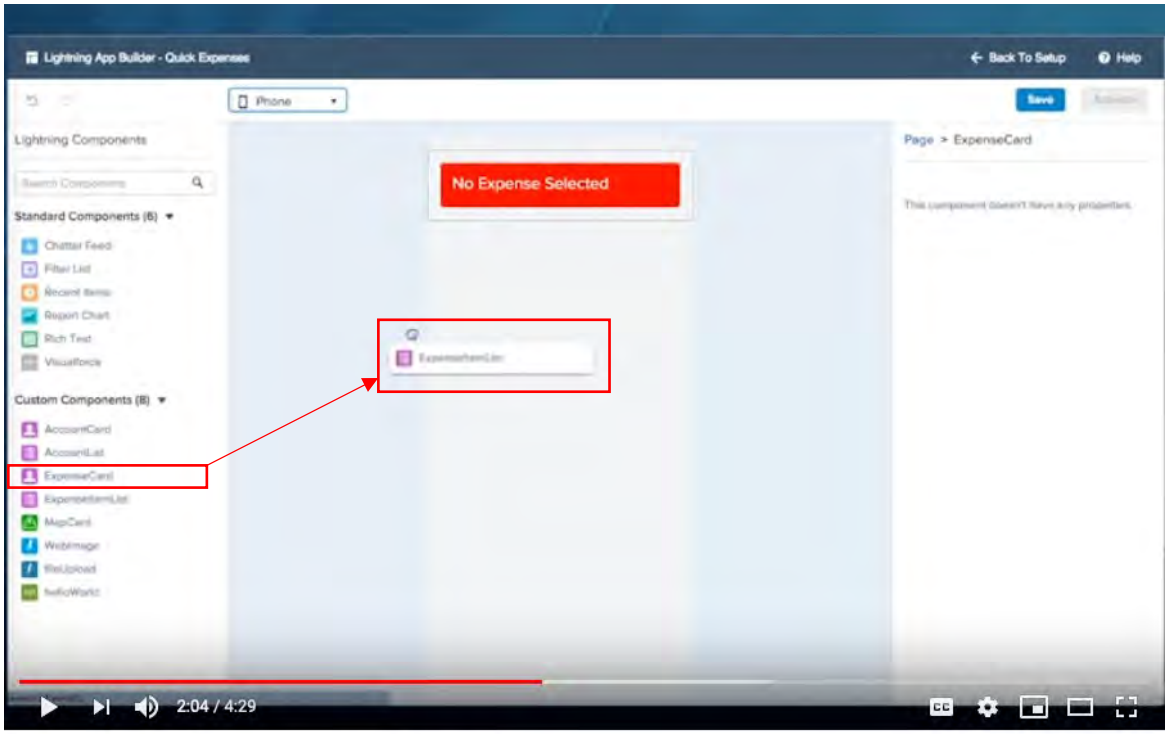
Salesforce Platform: Lightning App Builder Demo

Source: https://www.youtube.com/watch?v=PKW4b_Hk9Hw 0:43/1:18

113. The Accused Instrumentality contains symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent

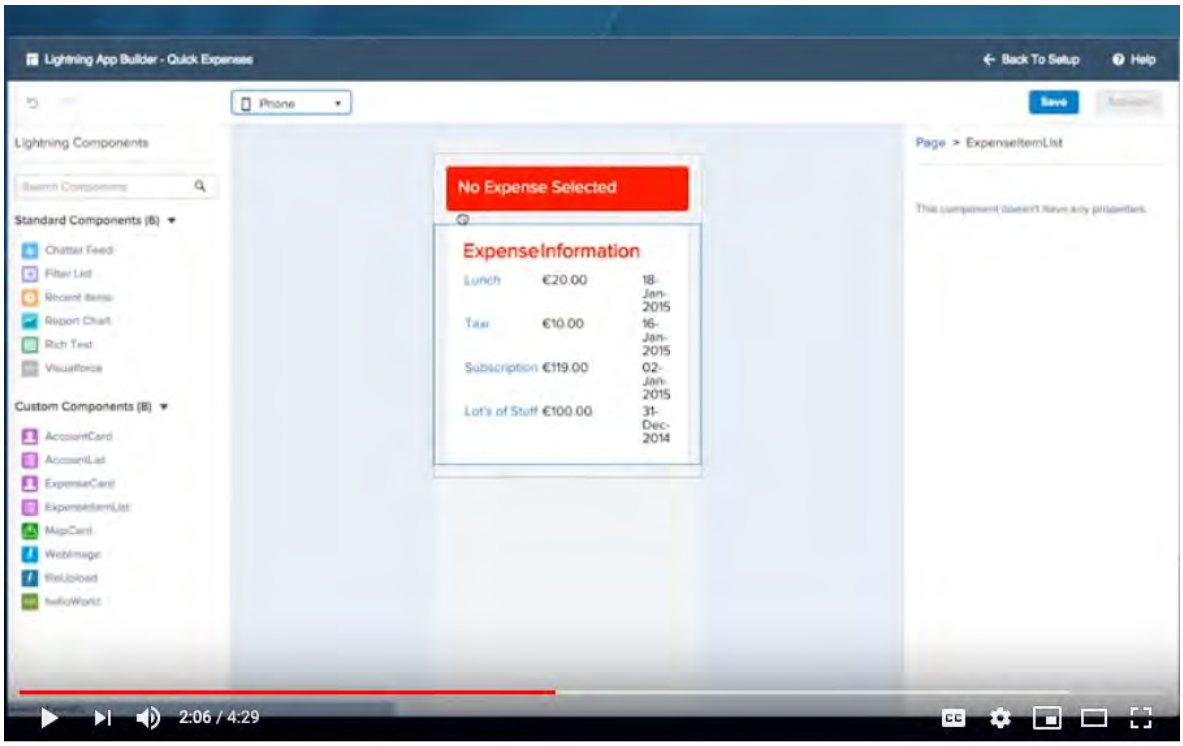
address or pointer to an output value accessible to the web service, where each symbolic name has an associated data format class type corresponding to a subclass of User Interface (UI) objects that support the data format type of the symbolic name, and where each symbolic name has a preferred UI object.

114. The Accused Instrumentality contains symbolic names as the names of its components. For example, a symbolic name is "ExpenseItemList." The "ExpenseItemList" symbolic name corresponds to the ExpenseItemList web component.



Lightning Intro Screencasts: Lightning App Builder

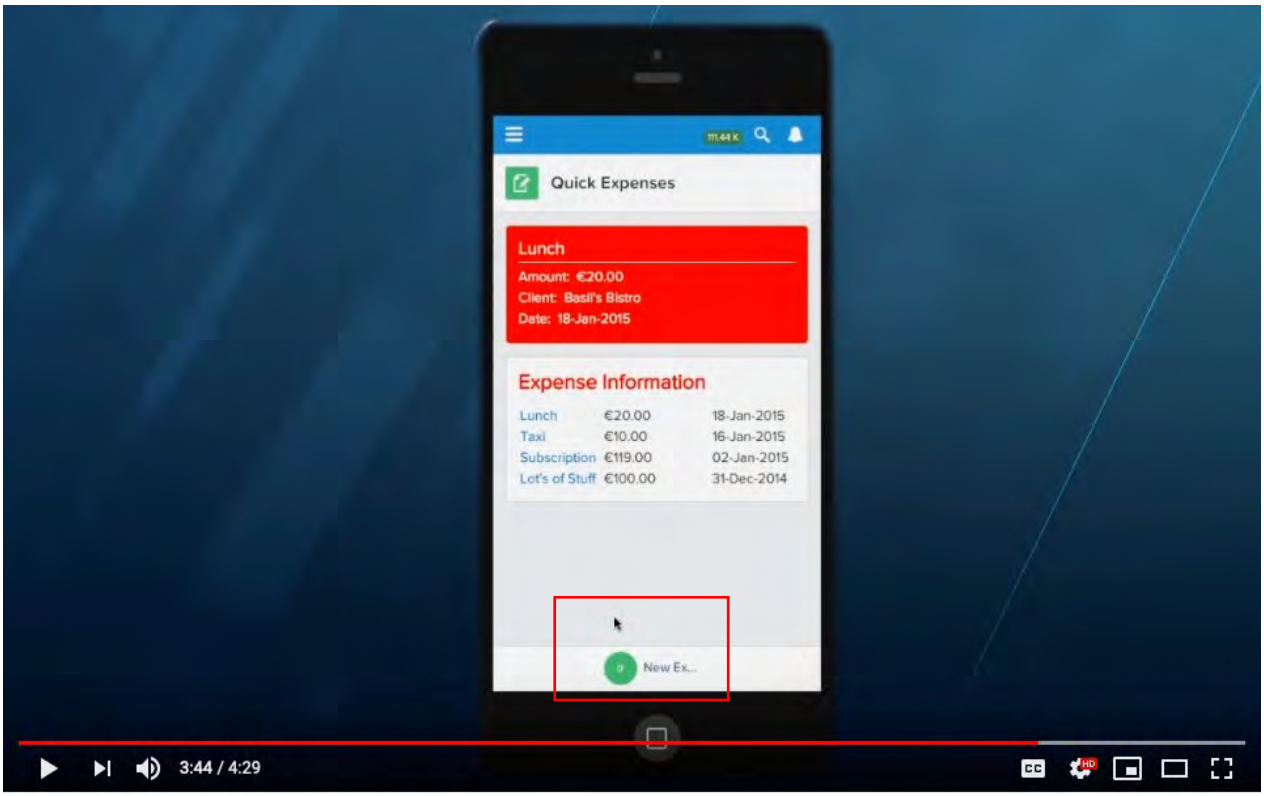
Source: https://youtu.be/XMK5Huuviow 2:04/4:29



Lightning Intro Screencasts: Lightning App Builder

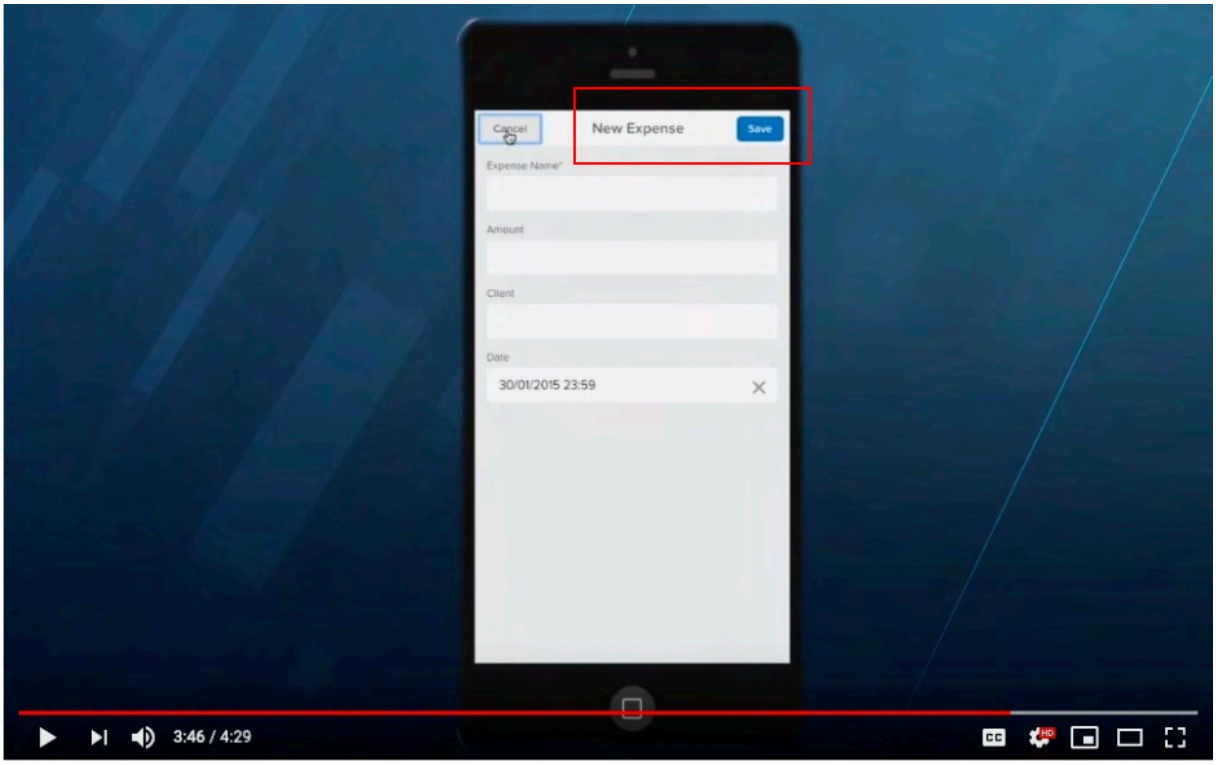
Source: https://youtu.be/XMK5Huuviow 2:06/4:29

115. The "ExpenseItemList" web component widget is related to a set of inputs and outputs of a web service obtainable over a network, as seen by the data populated in the widget, such as the first line item, "Lunch, €20.00, 18-Jan-2015." Users of the app can add additional Expense Items as inputs, and they will be reflected in the Expense Item List as outputs.



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 3:44/4:29



Source: https://youtu.be/XMK5Huuviow 3:46/4:29

116. These web components require either a web browser or the Salesforce mobile app to run. Salesforce publishes a list of supported browsers for the Accused Instrumentality.

	MICROSOFT® INTERNET EXPLORER®	MICROSOFT® EDGE	GOOGLE CHROME™	MOZILLA® FIREFOX®	APPLE® SAFARI®
Lightning Experience	IE 11 (EOL December 31, 2020)	Latest	Latest	Latest	12.x+
Lightning Communities	IE 11	Latest	Latest	Latest	12.x+
Special setup considerations?	No	No	No	No	No
Limitations?	Yes	Yes	No	Yes	Yes

Source: https://help.salesforce.com/articleView?id=getstart_browsers_sfx.htm

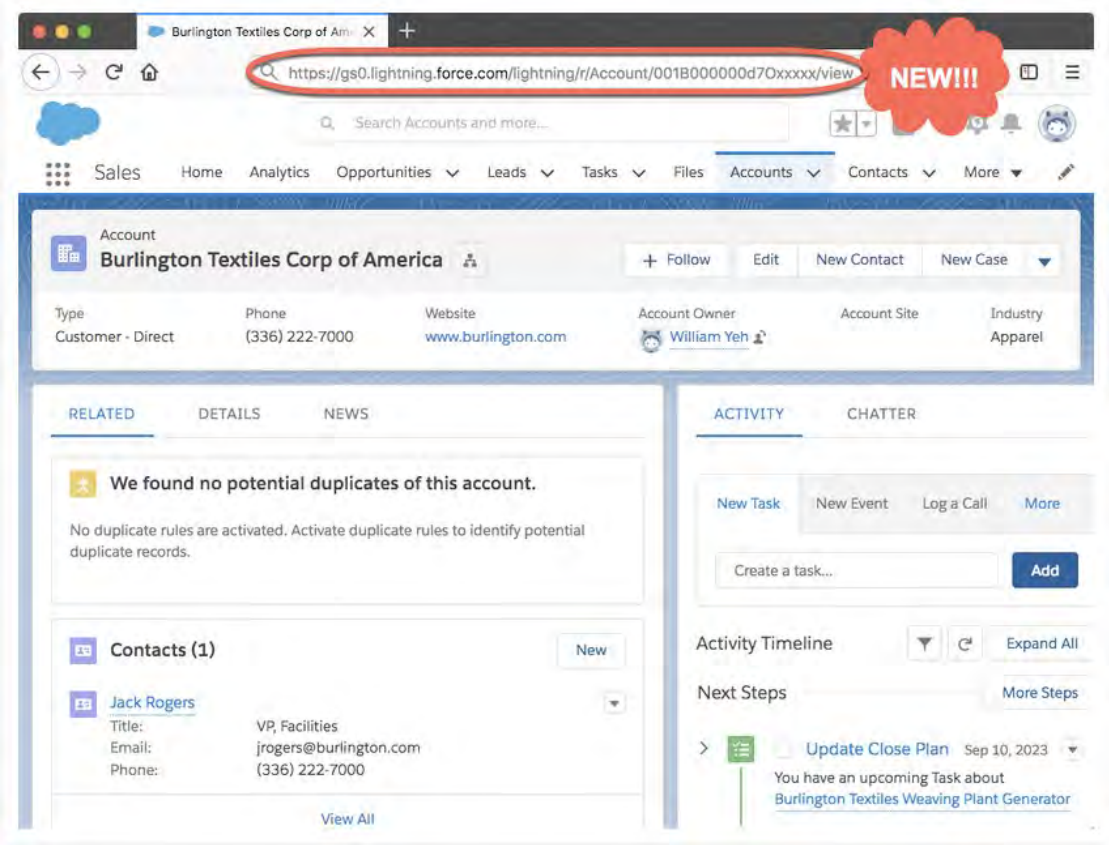
117. The Accused Instrumentality stores an address of the web service in computer memory. For example, the address of the webservice in Salesforce Lightning is embedded in the Component that is dragged from the AppStore and seamlessly placed in the Lightning page in the app builder. For custom webservices, "MyDomain" is deployed.

Your custom Lightning components don't automatically work on Lightning pages or in the Lightning App Builder. To make a custom component usable in both, you need to:

1. Configure the component and its component bundle so that they're compatible with the Lightning App Builder and Lightning pages.
2. Deploy My Domain in your org. When you deploy My Domain, references and links to Lightning resources are in the format `https://<myDomain>.lightning.force.com`.

Source:
https://help.salesforce.com/articleView?id=lightning_page_components_custom.htm&type=5

118. As another example, the Accused Instrumentality's web pages such as the "Burlington Textiles Corp of America" served to users contain addresses. These addresses are stored in computer memory on at least Salesforce's servers. On information and belief, the mobile app also contains the address of the web service.



Source: https://developer.salesforce.com/blogs/2018/01/heres-need-know-new-url-format-lightning.html

Example of new URLs

Original format: Account Home URL

https://<lightning.domain.com>/one/one.app/#/sObject/Account/home

New format: Account Home URL

https://<lightning.domain.com>/lightning/o/Account/home

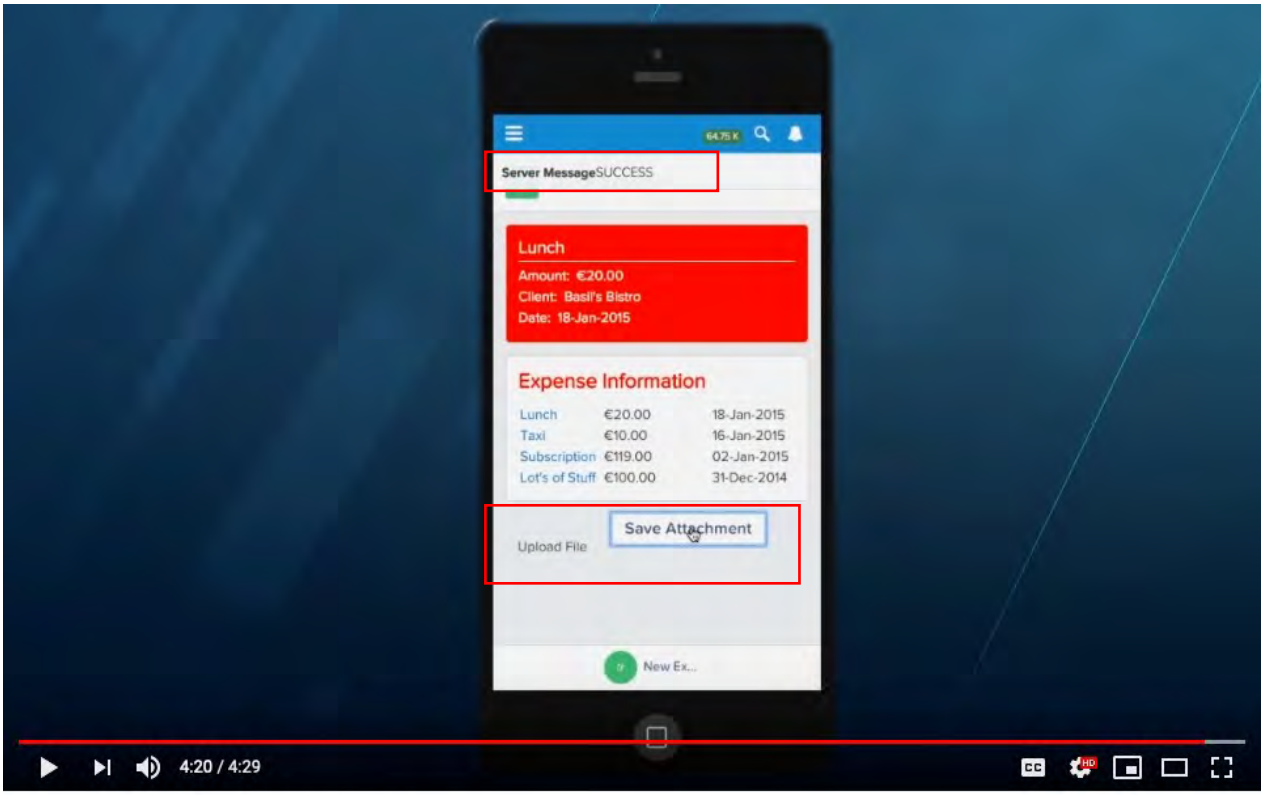
Original format: Account Record View

https://<lightning.domain.com>/one/one.app#/sObject/<recordID>/view

New format: Account Record View

https://<lightning.domain.com>/lightning/r/Account/<recordID>/view

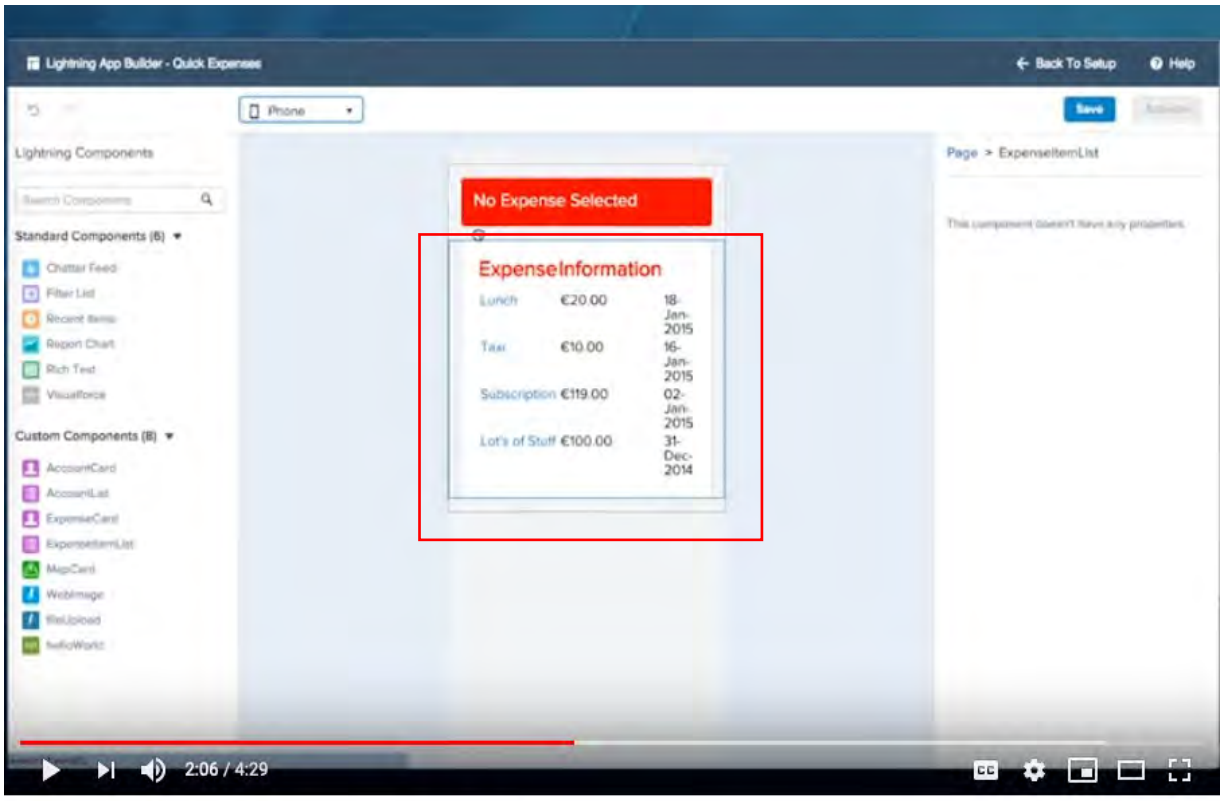
Source: https://developer.salesforce.com/blogs/2018/01/heres-need-know-new-url-format-lightning.html



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 4:20/4:29

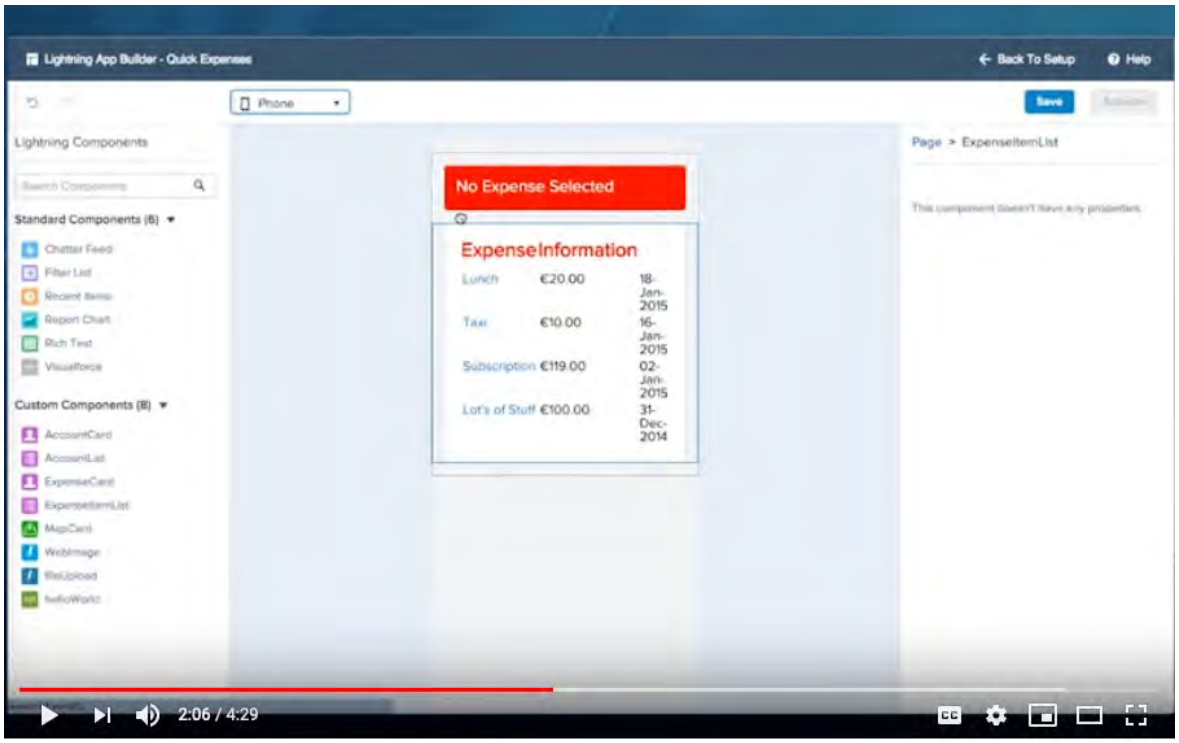
119. The Accused Instrumentality contains an authoring tool configured to define a UI object for presentation on the display. For example, the Accused Instrumentality's tool configures an ExpenseItemList object for presentation on the display, titled Expense Information (hereafter "Expense Information").



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 2:06/4:29

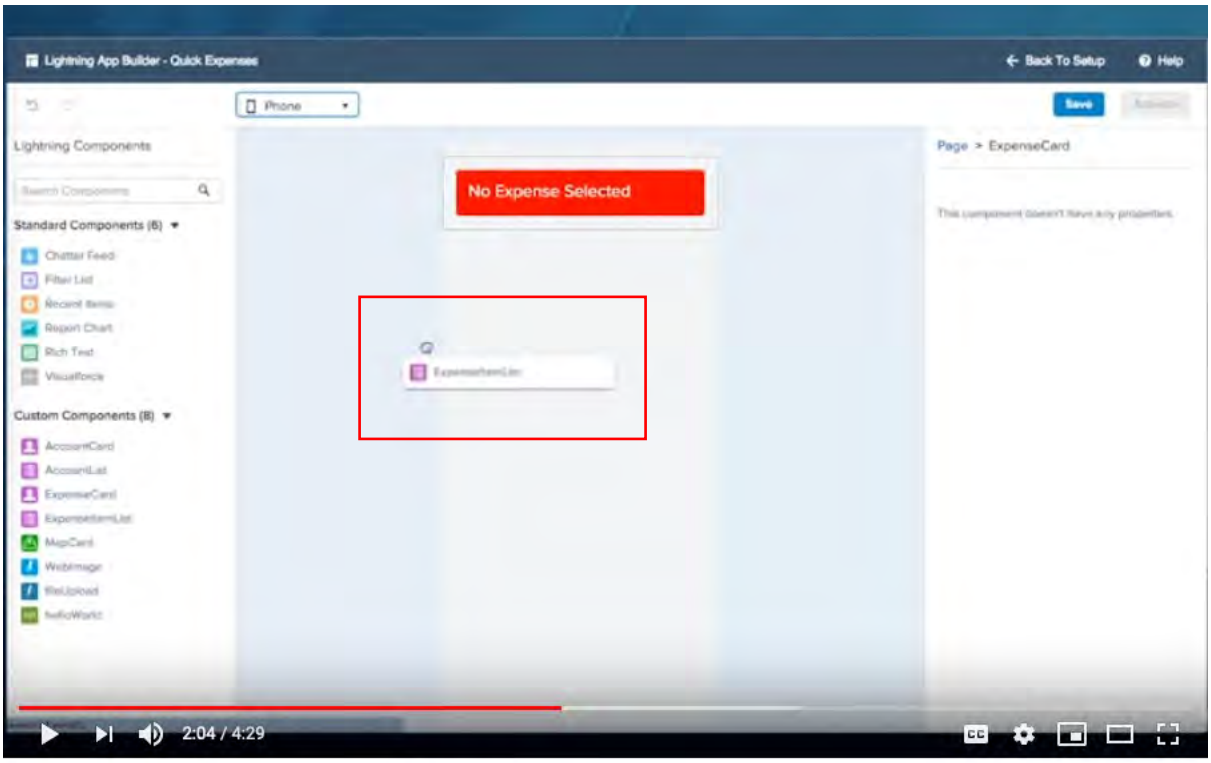
120. The UI object corresponds to a web component included in the computer memory selected from a group consisting of an input of the web service and an output of the web service, where each defined UI object is either: 1) selected by a user of the authoring tool; or 2) automatically selected by the system as the preferred UI object corresponding to the symbolic name of the web component selected by the user of the authoring tool. The defined Expense Information UI object in the center of the screen corresponds to the ExpenseItemList web component on the left side of the screen. The available web components are listed next to the ExpenseItemList web components and are stored in memory on Salesforce servers.



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 2:06/4:29

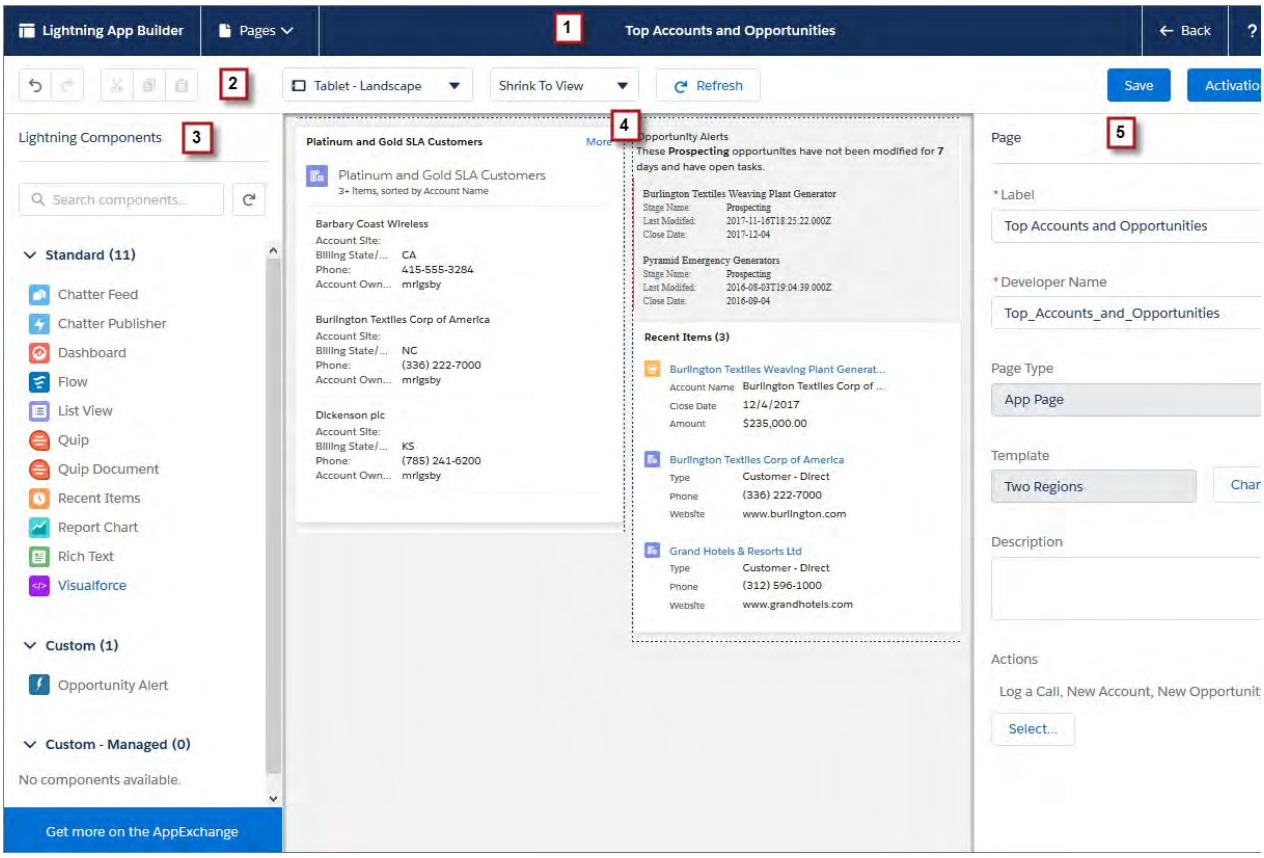
121. The defined Expense Information UI object is selected by the user. Since the UI object is a Custom Component, it contains information selected by the user.



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 2:04/4:29

122. Other defined UI objects, such as Standard Components, might be automatically selected by the system as the preferred UI object corresponding to the symbolic name of the web component selected by the user of the authoring tool.



Source:
https://trailhead.salesforce.com/content/learn/modules/lightning_app_builder/lightning_app_builde r_intro

Lightning Components Pane (3)

The components pane contains all standard and custom Lightning components that are supported for your Lightning page. Click and drag a component to add it to the page.

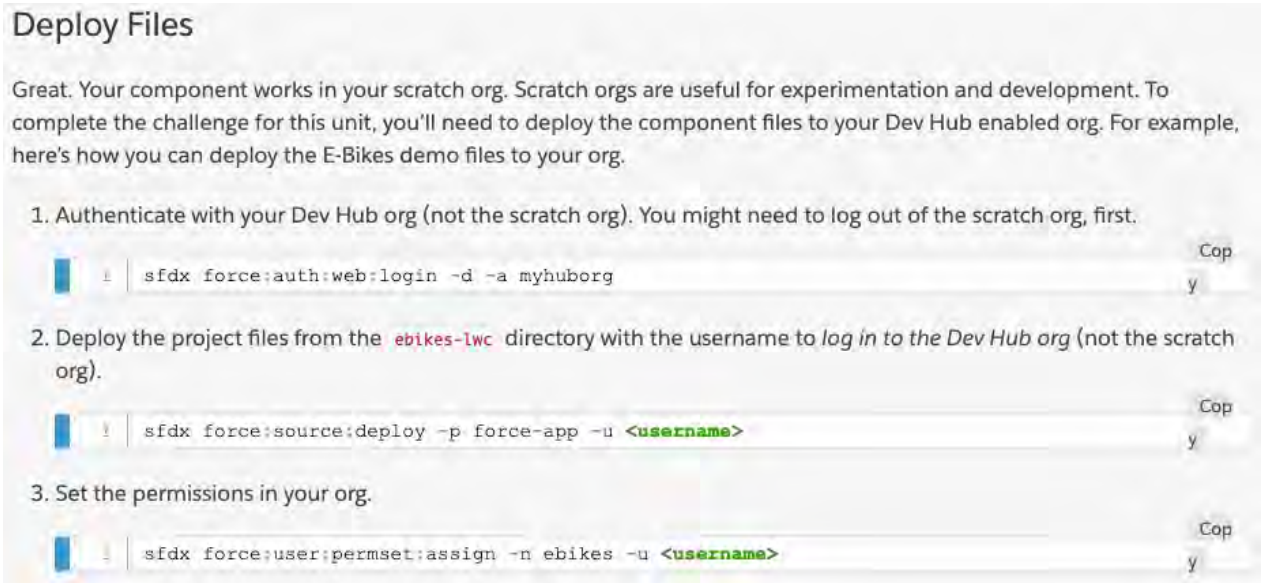
Tip

If you have a lot of custom components, enter text in the search field to easily find the one you need. You can access third-party custom components on the AppExchange using the button at the bottom of the pane.

Lightning Page Canvas (4)

The canvas area is where you build your page. Drag components to reorder them on the page.

Source:
https://trailhead.salesforce.com/content/learn/modules/lightning_app_builder/lightning_app_builde r_intro



Deploy Files

Great. Your component works in your scratch org. Scratch orgs are useful for experimentation and development. To complete the challenge for this unit, you'll need to deploy the component files to your Dev Hub enabled org. For example, here's how you can deploy the E-Bikes demo files to your org.

1. Authenticate with your Dev Hub org (not the scratch org). You might need to log out of the scratch org, first.

```
sfdx force:auth:web:login -d -a myhuborg
```

2. Deploy the project files from the `ebikes-lwc` directory with the username to *log in to the Dev Hub org* (not the scratch org).

```
sfdx force:source:deploy -p force-app -u <username>
```

3. Set the permissions in your org.

```
sfdx force:user:permset:assign -n ebikes -u <username>
```

Source: https://trailhead.salesforce.com/content/learn/modules/lightning-web-components-basics/push- lightning-web-component-files

Lightning pages support these components:

Standard Components

Standard components are Lightning components built by Salesforce.

Custom Components

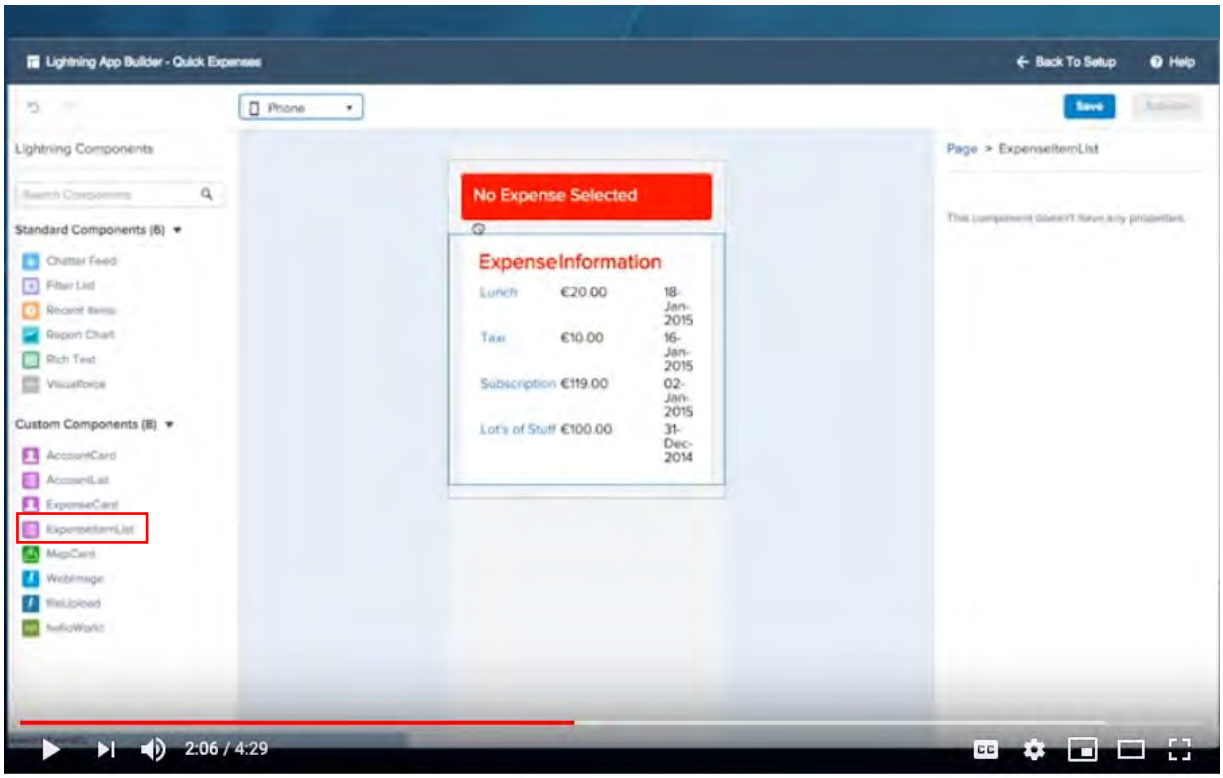
Custom components are Lightning components that you or someone else have created. You can configure custom Lightning components to work in Lightning App Builder.

Third-Party Components on AppExchange

The AppExchange provides a marketplace for Lightning components. You can find packages containing components already configured and ready to use in the Lightning App Builder.

Source: https://trailhead.salesforce.com/content/learn/modules/lightning_app_builder/lightning_app_builde r_intro

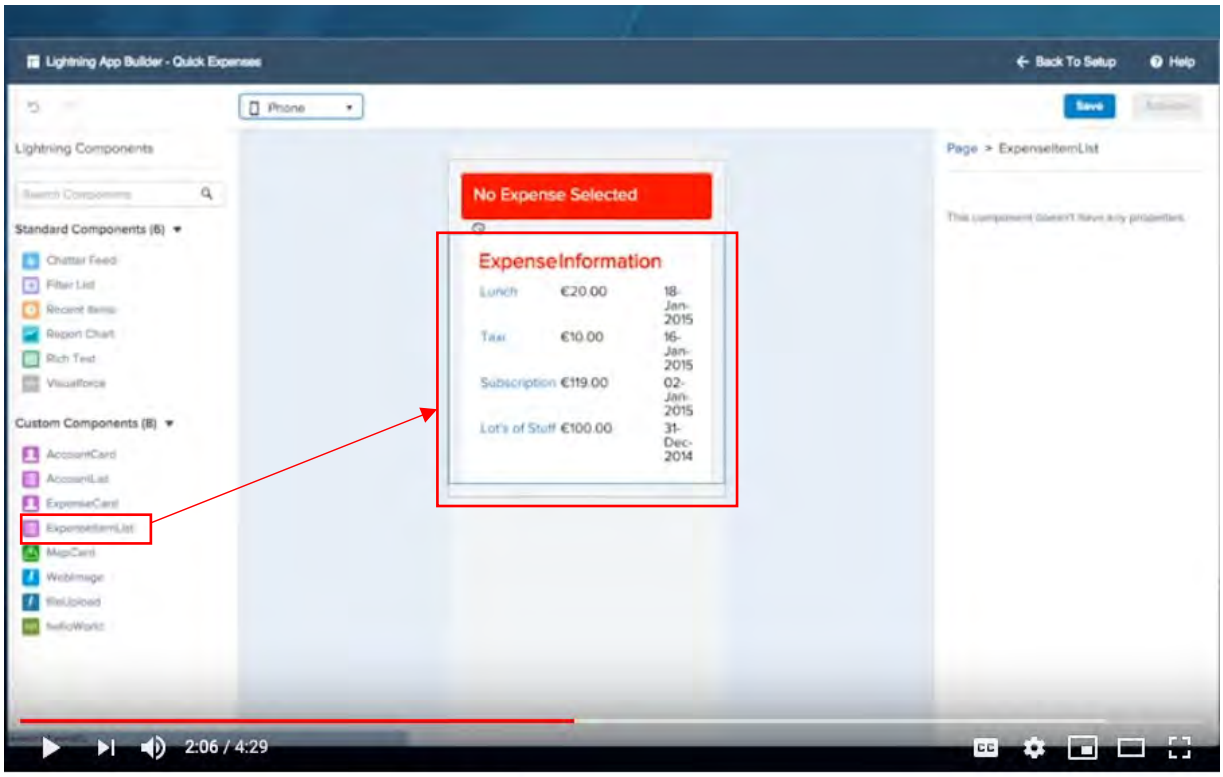
123. The authoring tool in the Accused Instrumentality is configured to access the computer memory to select the symbolic name corresponding to the web component of the defined UI object. For example, the Accused Instrumentality accesses computer memory to select the name of the "ExpenseItemList" component of the defined Expense Information UI object.



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 2:06/4:29

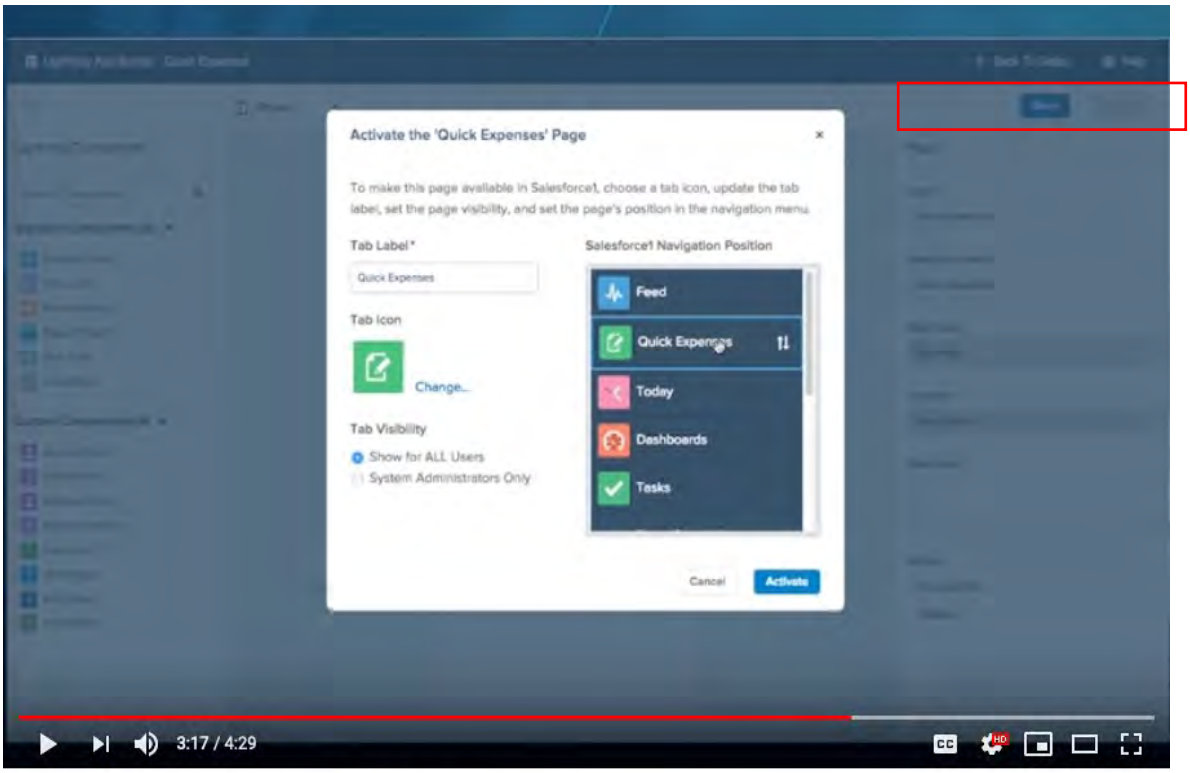
124. The authoring tool in the Accused Instrumentality is configured to associate the selected symbolic name with the defined UI object, where the selected symbolic name is only available to UI objects that support the defined data format associated with that symbolic name. For example, the authoring tool associates the "ExpenseItemList" symbolic name with the defined UI object, where the symbolic name is only available to UI objects that support the data of the ExpenseItemList.



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 2:06/4:29

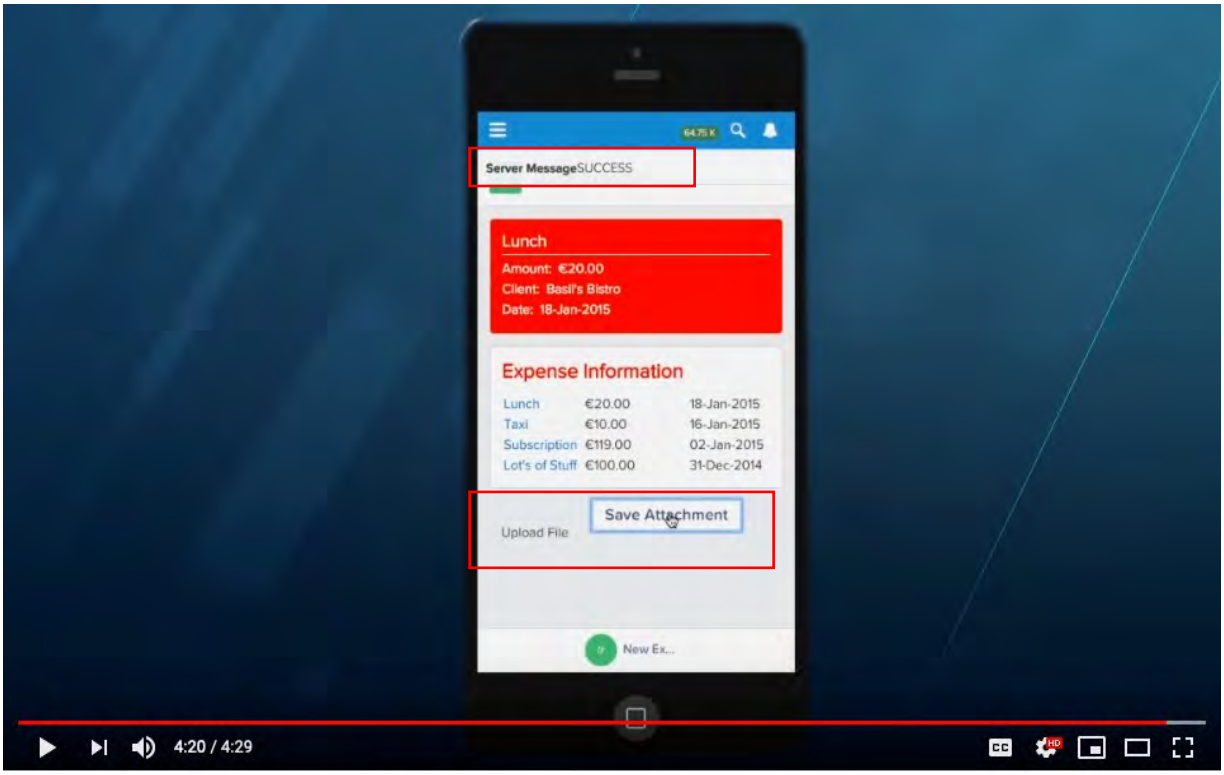
125. The authoring tool in the Accused Instrumentality is configured to store information representative of the defined UI object and related settings in a database. For example, once the user clicks "save" and "activate" on the authoring tool, the application is saved and built.



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 3:17/4:29

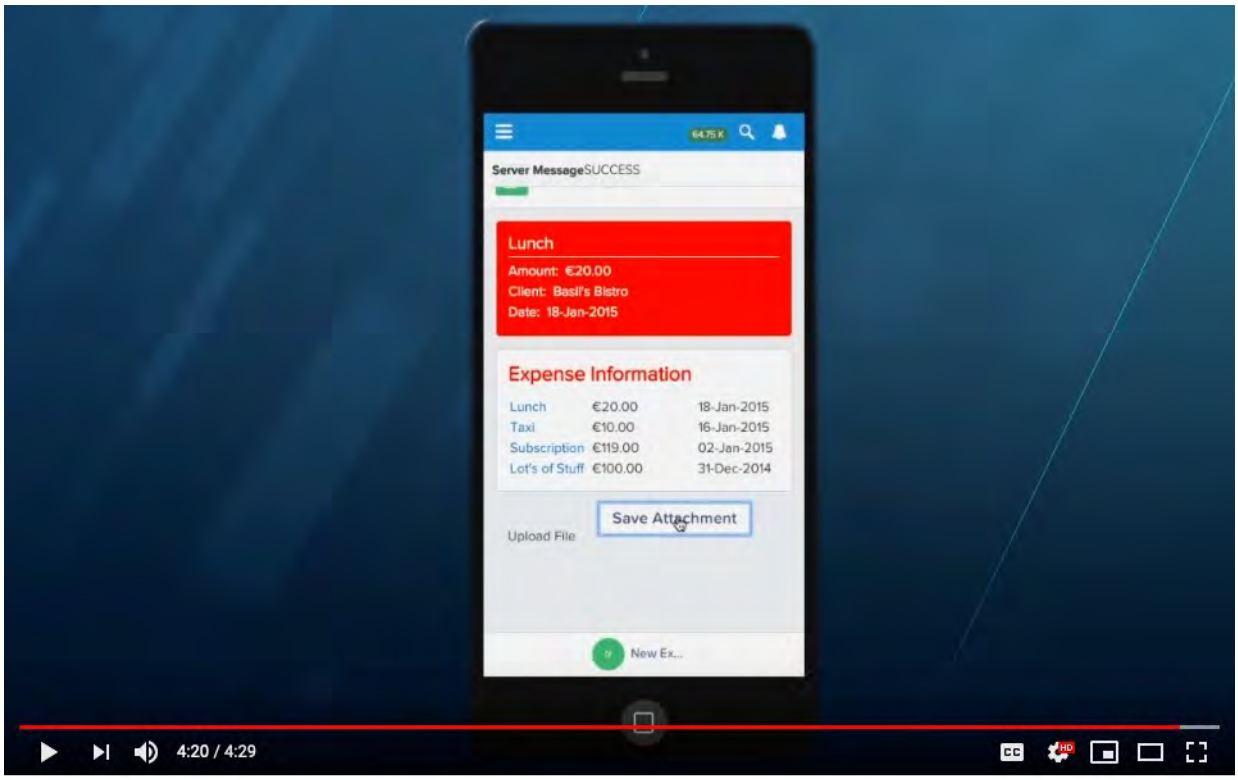
126. The information representative of the defined UI object is saved in a database on Salesforce's servers. Users using either the web browser or mobile app communicate with the server and information is stored in Salesforce's databases.



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 4:20/4:29

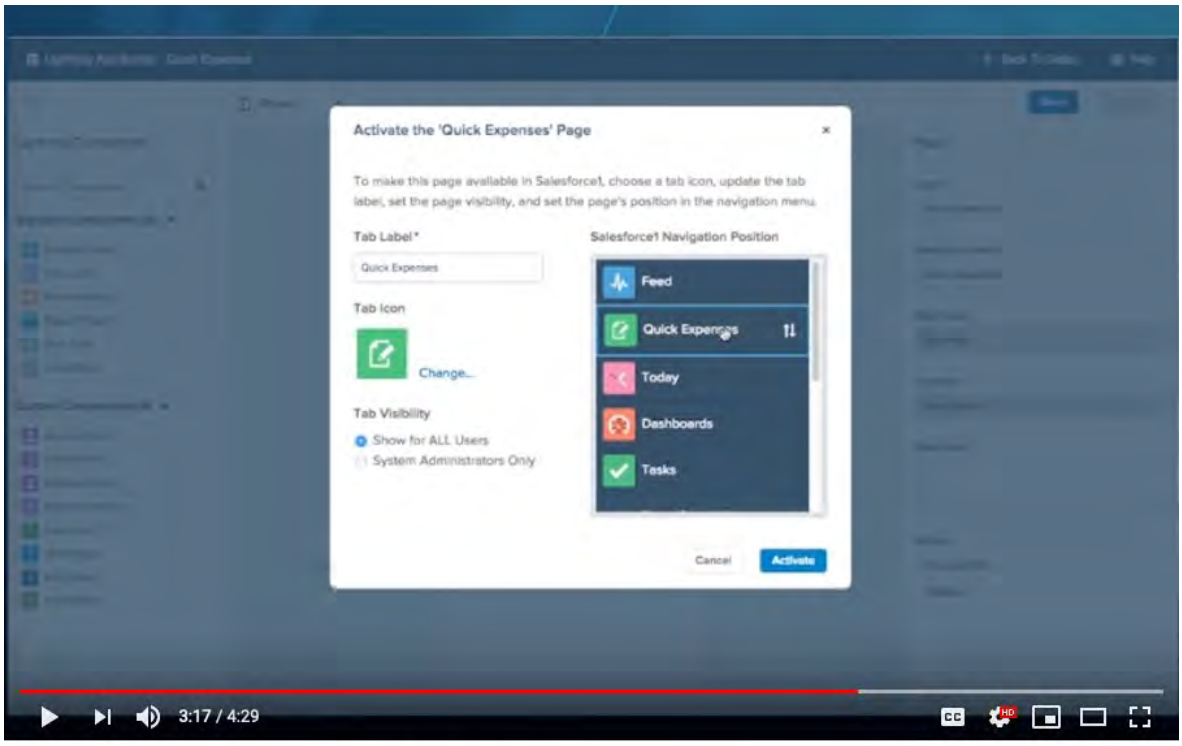
127. The authoring tool in the Accused Instrumentality is configured to retrieve the information representative of the one or more the UI object settings stored in the database. For example, the UI object Expense Information, with its object settings such as the expenses shown, is retrieved from the database on Salesforce servers for the purpose of displaying the information on the web site or app.



Lightning Intro Screencasts: Lightning App Builder

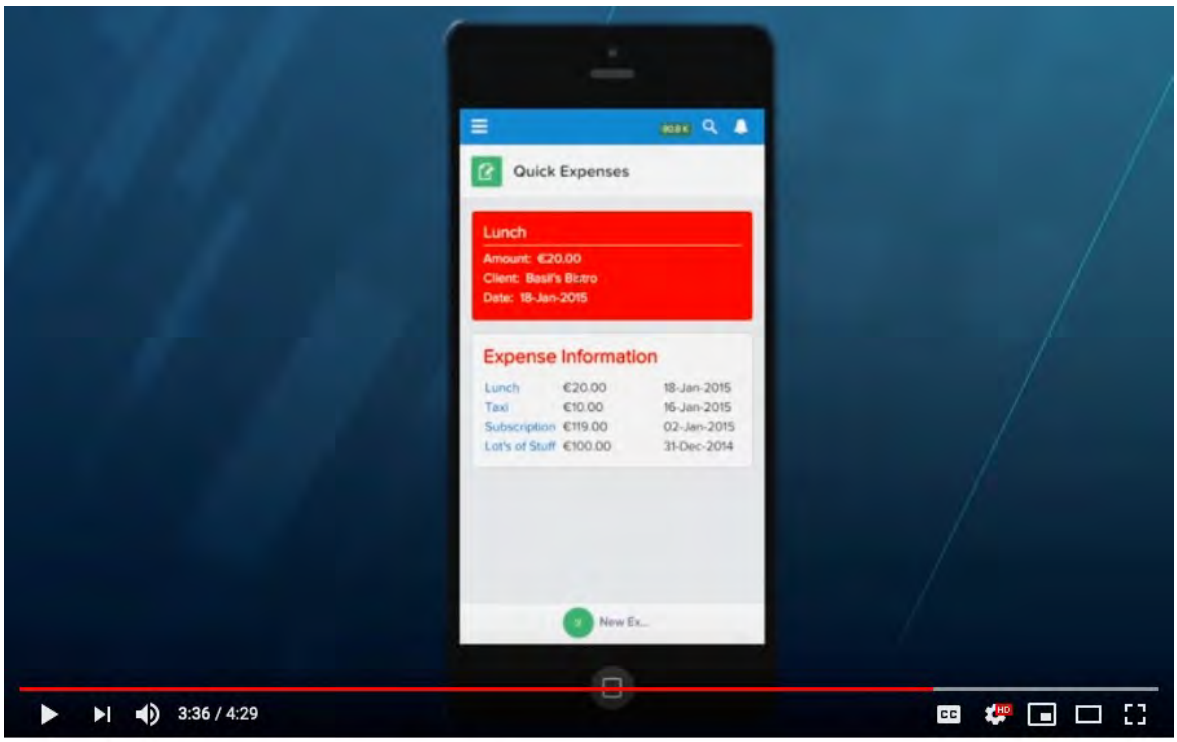
Source: https://youtu.be/XMK5Huuviow 4:20/4:29

128. The authoring tool in the Accused Instrumentality is configured to build an application consisting of one or more web page views from at least a portion of the database utilizing at least one player, where the player utilizes information stored in the database to generate for the display of at least a portion of the one or more web pages when the application and player are provided to the device and executed on the device. For example, once the user clicks "save" and "activate" on the authoring tool, the application is saved and built. The authoring tool builds and delivers code that uses information stored in the database that is independent of the device that it is displayed on.



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 3:17/4:29



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 3:36/4:29

129. The Salesforce phone app shown consists of one or more web page views from the Salesforce server, utilizing the player embedded in the mobile app and where the app information from the Salesforce server generates the information shown, such as the Expense Information. Similarly, the app can be accessed through a web browser on a standard computer, subject to the requirements of using a supported browser. The listed web browsers rely on browser engines to interpret and execute JavaScript and HTML to render web pages on a computer. Internet Explorer has relied on Trident (code name for MSHTML) (which has included the Chakra JavaScript Engine, see http://en.wikipedia.org/wiki/MSHTML); Safari and Chrome rely on Webkit (which includes WebCore and JavaScript Core, see http://en.wikipedia.org/wiki/WebKit); Firefox relies on Gecko (which includes Spidermonkey, see http://www.mozilla.org/projects/technologies.html).

Supported Browsers for Lightning Experience

See the supported browsers and limitations for Lightning Experience.

Where: Lightning Experience is available in the Essentials, Group, Professional, Enterprise, Performance, Unlimited, and Developer editions.

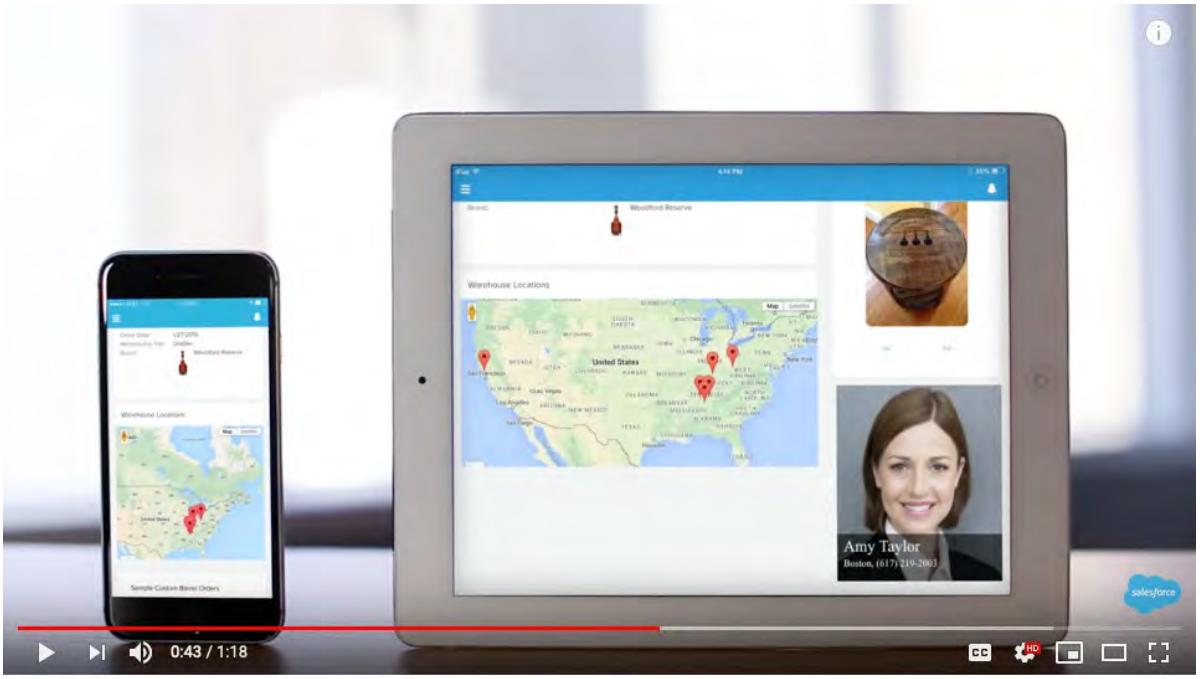
NOTE

- Salesforce doesn't support non-browser applications that embed WebView or similar controls to render content for Salesforce Classic, Lightning Experience, or Salesforce mobile web. Examples of approaches that embed this type of control include Salesforce Mobile SDK, Microsoft's WebBrowser Control, Electron's embedded Chromium browser, iOS's UIWebView and WKWebView, and Android's WebView.
- You can't access Lightning Experience in a mobile browser. Instead, we recommend using the Salesforce app when you're working on a mobile device. For a list of supported mobile browsers on Salesforce, see Requirements for the Salesforce App.

	MICROSOFT® INTERNET EXPLORER®	MICROSOFT® EDGE	GOOGLE CHROME™	MOZILLA® FIREFOX®	APPLE® SAFARI®
Lightning Experience	IE 11 (EOL December 31, 2020)	Latest	Latest	Latest	12.x+
Lightning Communities	IE 11	Latest	Latest	Latest	12.x+
Special setup considerations?	No	No	No	No	No
Limitations?	Yes	Yes	No	Yes	Yes

Source: https://help.salesforce.com/articleView?id=getstart_browsers_sfx.htm

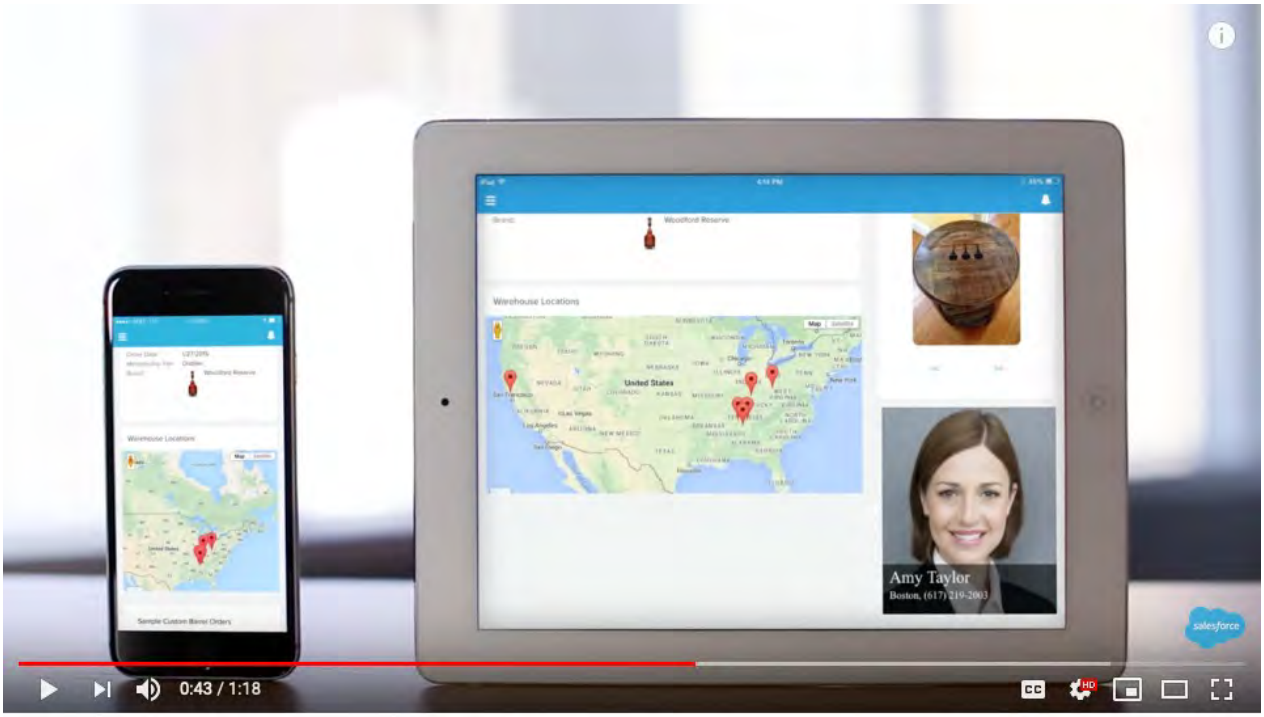
130. An additional example of a display of web page views on both a desktop browser and mobile device is shown below.



Salesforce Platform: Lightning App Builder Demo

Source: https://www.youtube.com/watch?v=PKW4b_Hk9Hw 0:43/1:18

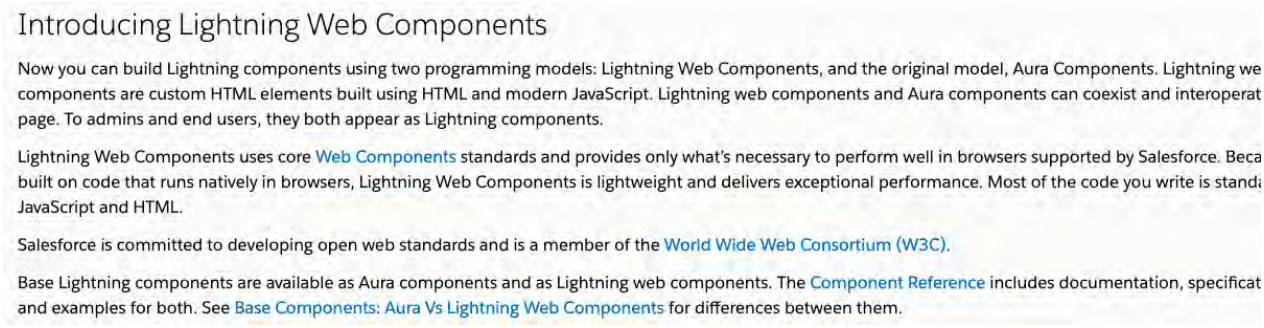
131. The application and player are provided to the device and executed on the device. Apps created with the Accused Instrumentality are available on both desktop and mobile devices. In both the mobile app and desktop experience, both the application and player are provided to a user's device and executed on the device.



Salesforce Platform: Lightning App Builder Demo

Source: https://www.youtube.com/watch?v=PKW4b_Hk9Hw 0:43/1:18

132. On desktop devices, the application and player are embedded in standard web browsers.



Introducing Lightning Web Components

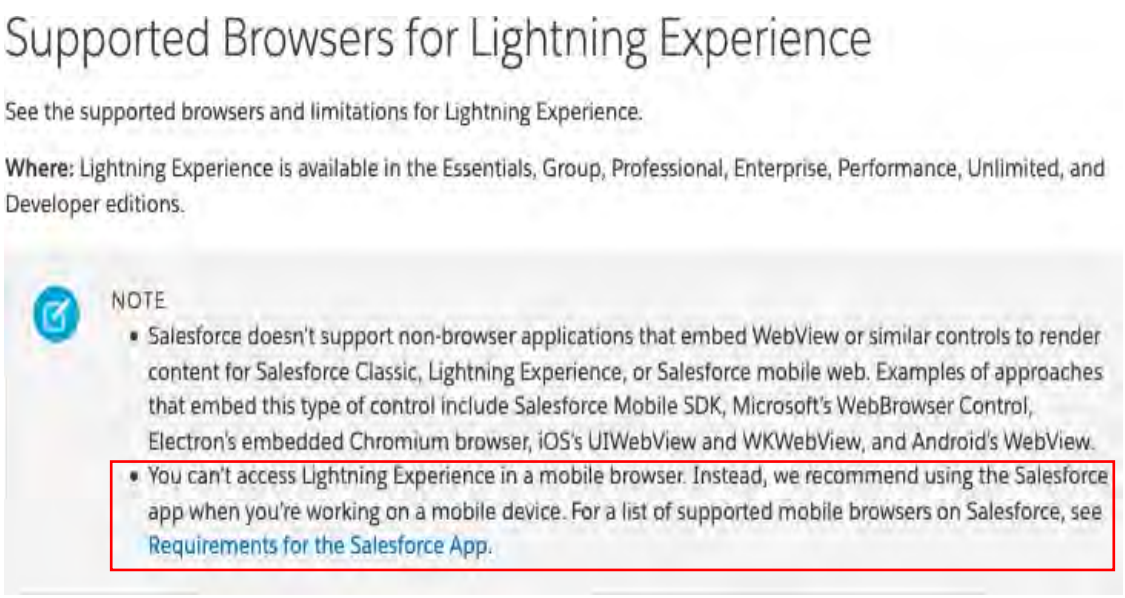
Now you can build Lightning components using two programming models: Lightning Web Components, and the original model, Aura Components. Lightning we components are custom HTML elements built using HTML and modern JavaScript. Lightning web components and Aura components can coexist and interoperat page. To admins and end users, they both appear as Lightning components.

Lightning Web Components uses core Web Components standards and provides only what's necessary to perform well in browsers supported by Salesforce. Beca built on code that runs natively in browsers, Lightning Web Components is lightweight and delivers exceptional performance. Most of the code you write is standa JavaScript and HTML.

Salesforce is committed to developing open web standards and is a member of the World Wide Web Consortium (W3C).

Base Lightning components are available as Aura components and as Lightning web components. The Component Reference includes documentation, specificat and examples for both. See Base Components: Aura Vs Lightning Web Components for differences between them.

Source: https://developer.salesforce.com/docs/component-library/documentation/lwc



Supported Browsers for Lightning Experience

See the supported browsers and limitations for Lightning Experience.

Where: Lightning Experience is available in the Essentials, Group, Professional, Enterprise, Performance, Unlimited, and Developer editions.

NOTE

- Salesforce doesn't support non-browser applications that embed WebView or similar controls to render content for Salesforce Classic, Lightning Experience, or Salesforce mobile web. Examples of approaches that embed this type of control include Salesforce Mobile SDK, Microsoft's WebBrowser Control, Electron's embedded Chromium browser, iOS's UIWebView and WKWebView, and Android's WebView.
- You can't access Lightning Experience in a mobile browser. Instead, we recommend using the Salesforce app when you're working on a mobile device. For a list of supported mobile browsers on Salesforce, see Requirements for the Salesforce App.

Source: https://help.salesforce.com/articleView?id=getstart_browsers_sfx.htm

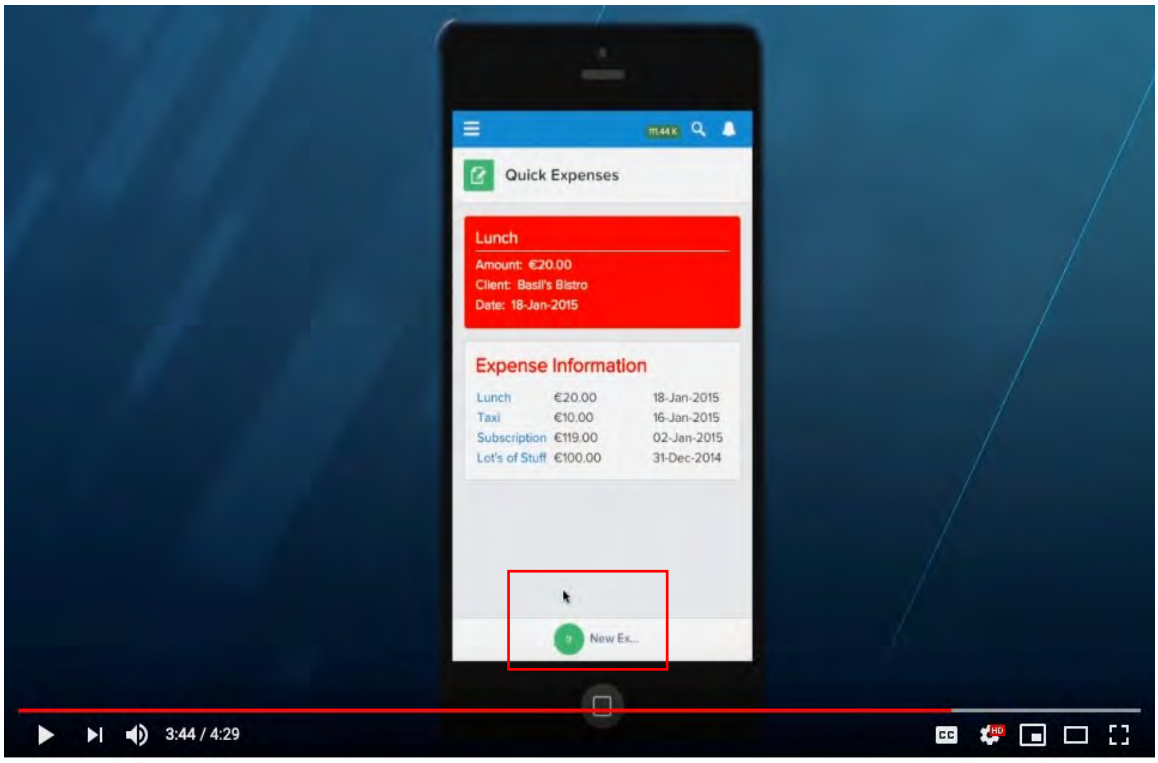
Welcome to the Salesforce Mobile App

The Salesforce app is an enterprise-class mobile app that provides your users with instant access to your company's CRM data from a phone or tablet. Here are some of the reasons why the app is so awesome.

- **The mobile app is included with every Salesforce license.** Yup, you heard us correctly–it's free. Procrastinating on your mobile rollout is basically like setting piles of money on fire.
- **The app is plug-and-play**, which means users just download it from the App Store or Google Play and go. It works out of the box with no setup required. It's lightning fast, seriously. In the time it took you to read this paragraph, you could have already installed the app and logged in.
- **The app is cross platform**, so it runs on Android and iOS operating systems. Like, automatically–without you having to do any development work.
- **The app has offline capabilities.** Your mobile users won't be affected by capricious cellular signals, FAA regulations, subway commutes, or bunker-style buildings.
- **It isn't just an app. It's a platform.** Because the app is powered by the Salesforce platform, it's infinitely customizable. You can use point-and-click tools to make it your own.

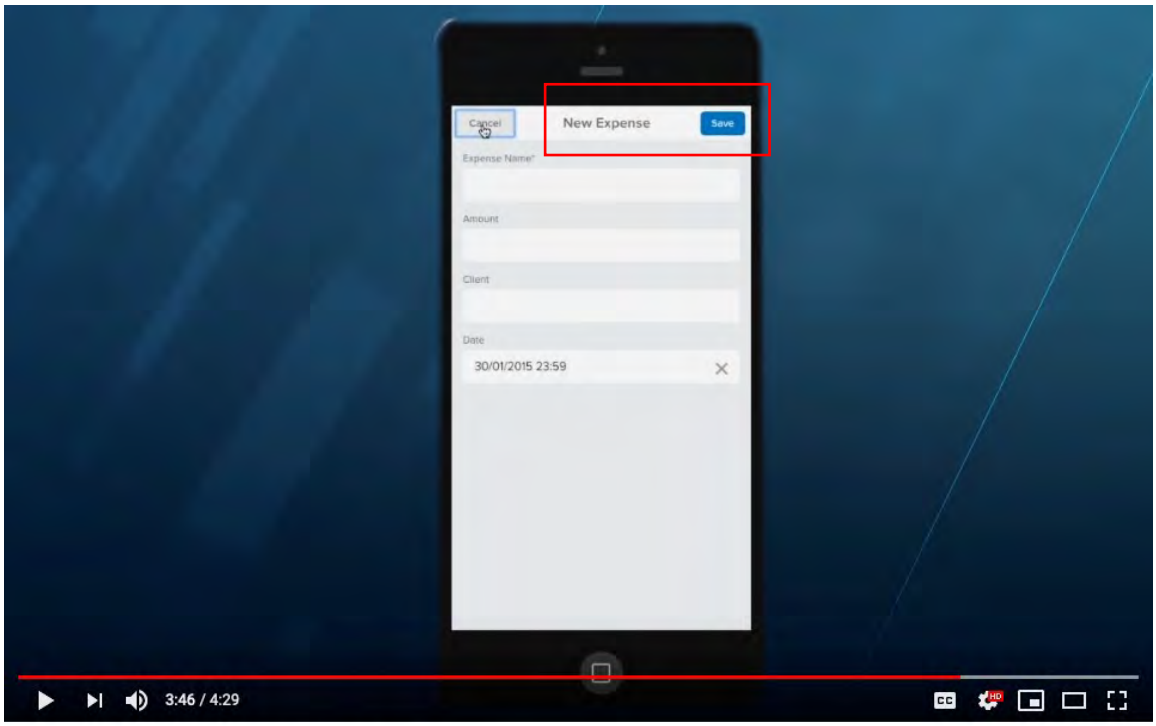
Source: https://trailhead.salesforce.com/content/learn/modules/salesforce1_mobile_app/salesforce1_mobile_app_intro

133. The user of the device provides one or more input values associated with an input symbolic name to an input of the defined UI object. For example, users of the app can add additional Expense Items.



Lightning Intro Screencasts: Lightning App Builder

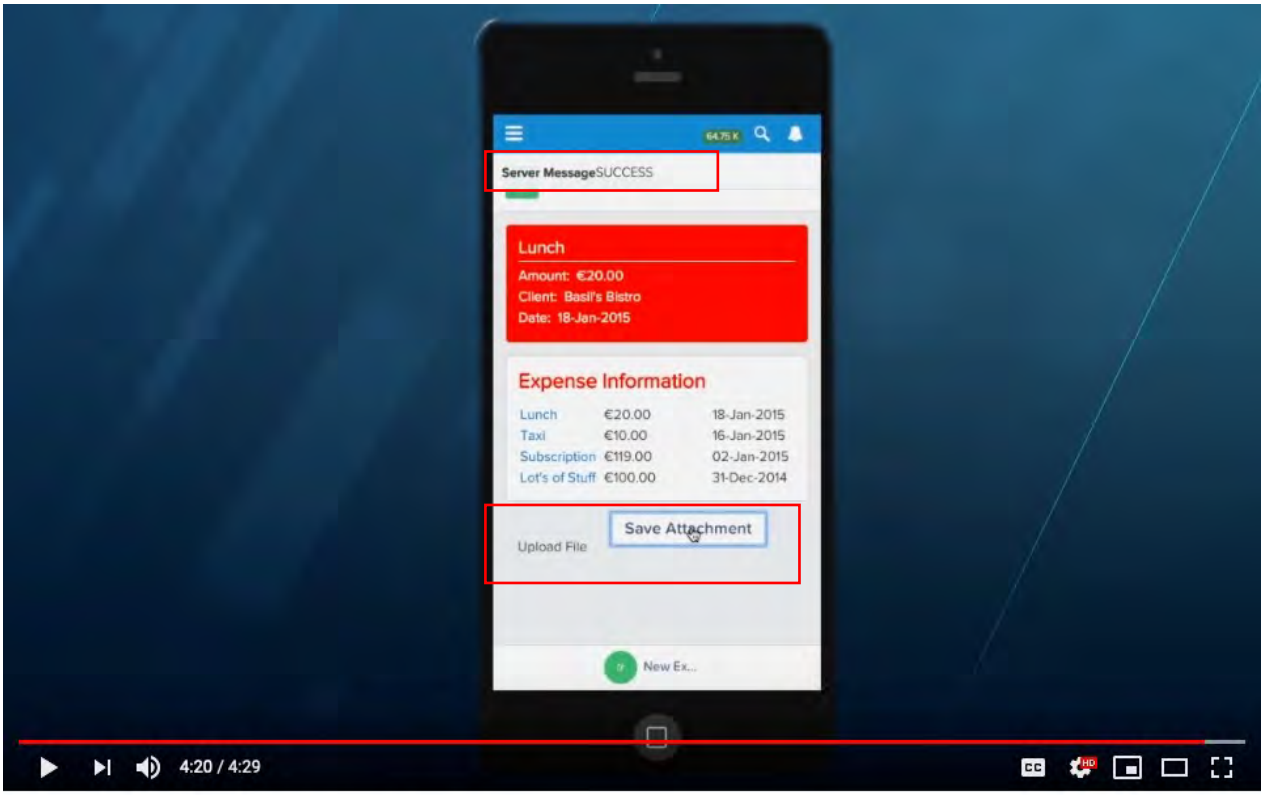
Source: https://youtu.be/XMK5Huuviow 3:44/4:29



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 3:46/4:29

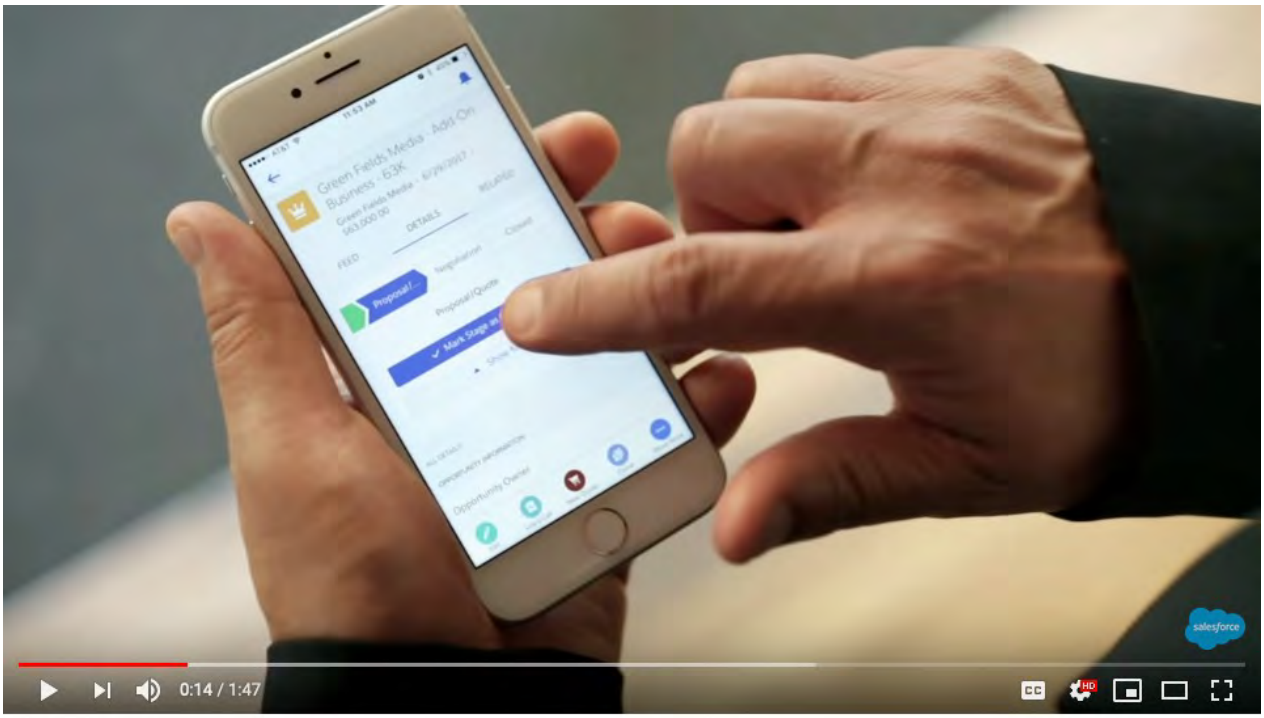
134. The device provides the user provided one or more input values and corresponding input symbolic name to the web service. For example, the device provides the new expenses to the Salesforce web service.



Lightning Intro Screencasts: Lightning App Builder

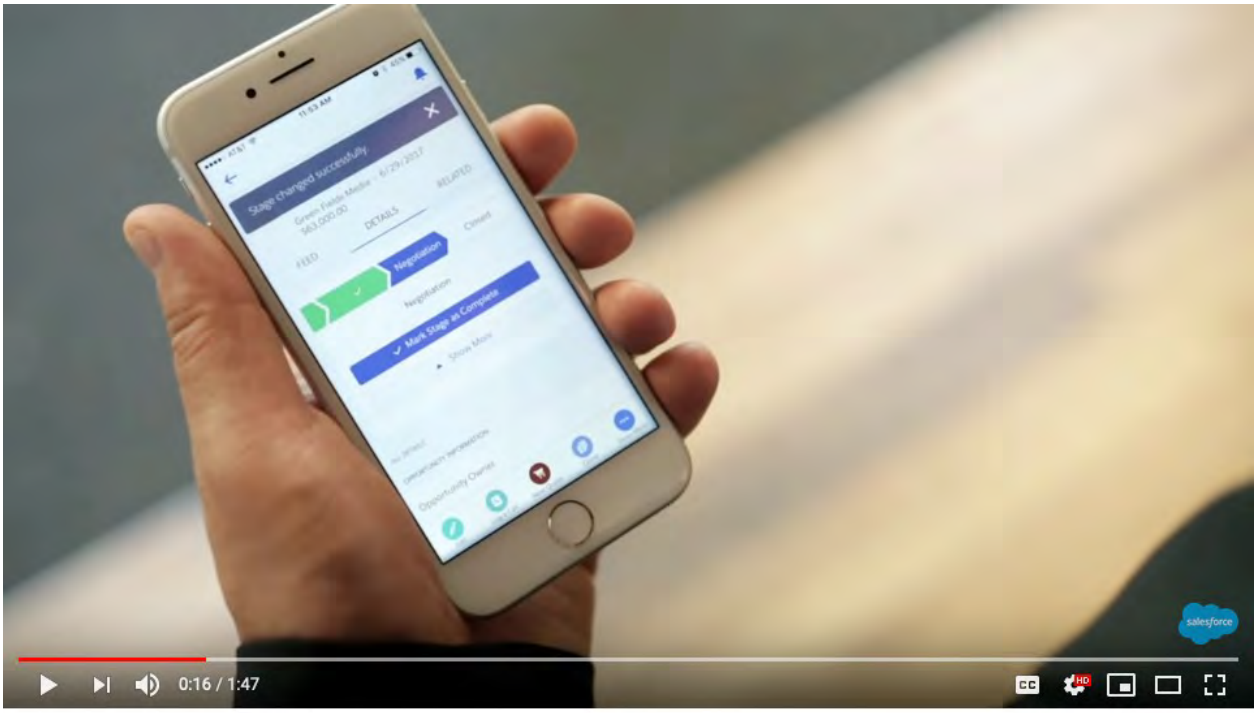
Source: https://youtu.be/XMK5Huuviow 4:20/4:29

135. As another example, a user can mark items as complete, which will communicate to the server.



Introduction to Salesforce Lightning

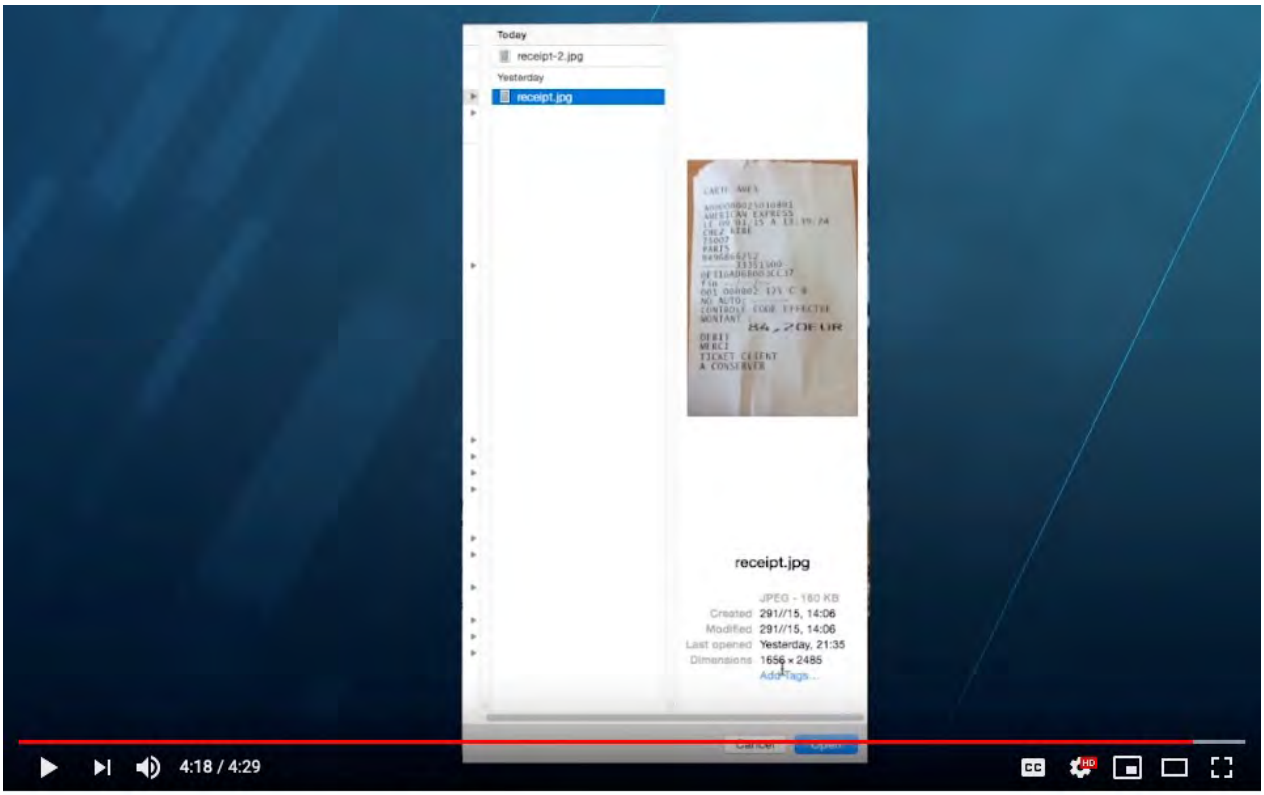
Source: https://www.youtube.com/watch?v=s_sHeac3gfo 0:14/1:47



Introduction to Salesforce Lightning

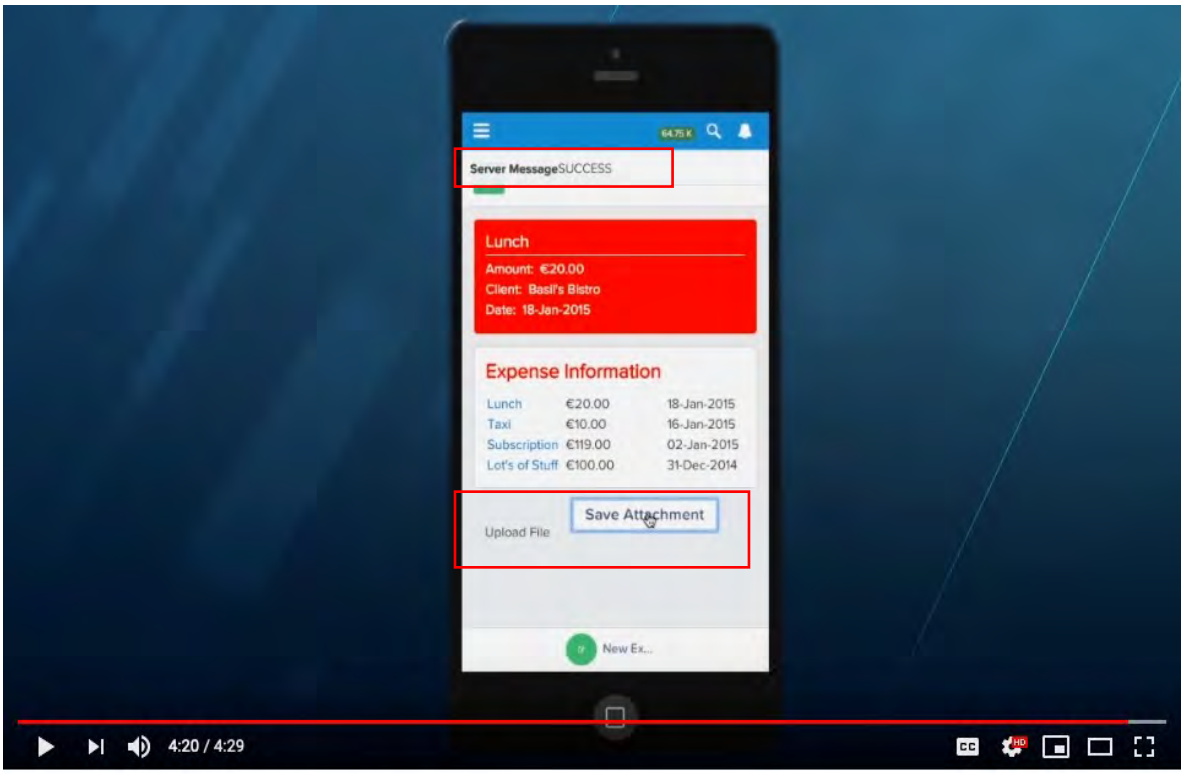
Source: https://www.youtube.com/watch?v=s_sHeac3gfo 0:16/1:47

136. The web service utilizes the input symbolic name and the user provided one or more input values for generating one or more output values having an associated output symbolic name, and the player receives the output symbolic name and corresponding one or more output values and provides instructions for the display of the device to present an output value in the defined UI object. For example, the web service will use the user's input values for generating output values and return data to the player for generating a confirmation "SUCCESS" message after the user uploaded a receipt.



Lightning Intro Screencasts: Lightning App Builder

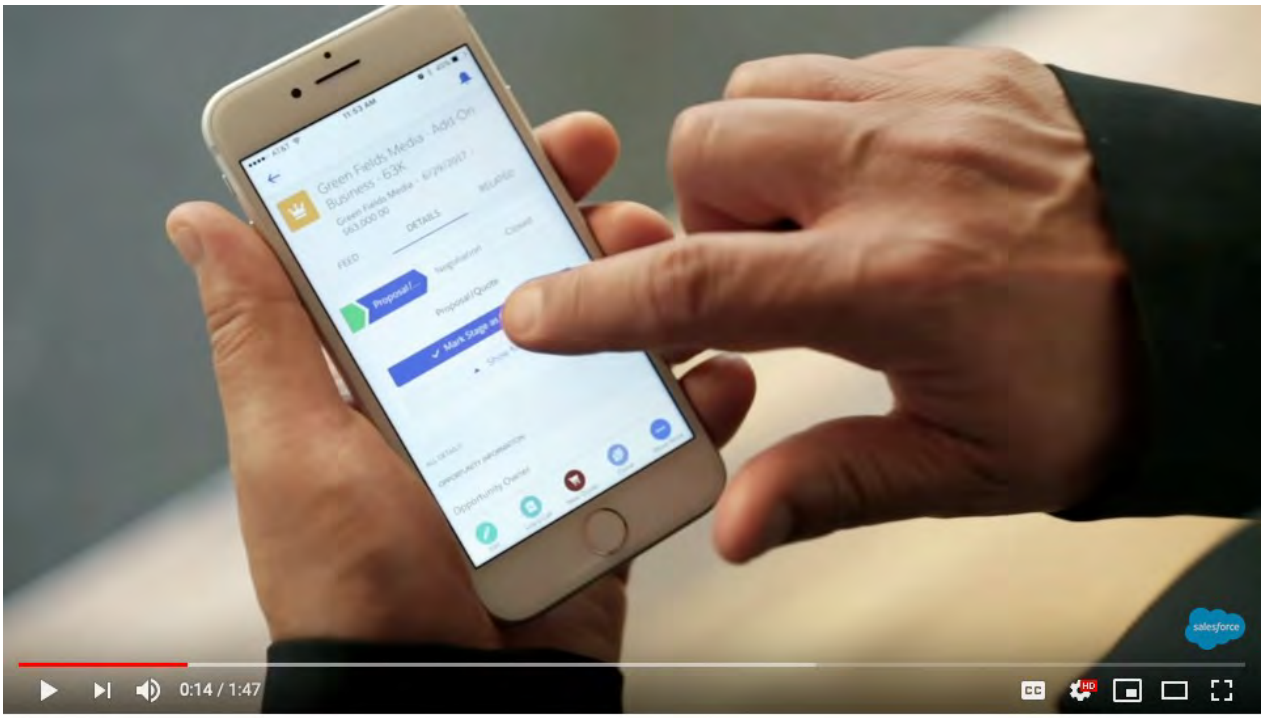
Source: https://youtu.be/XMK5Huuviow 4:18/4:29



Lightning Intro Screencasts: Lightning App Builder

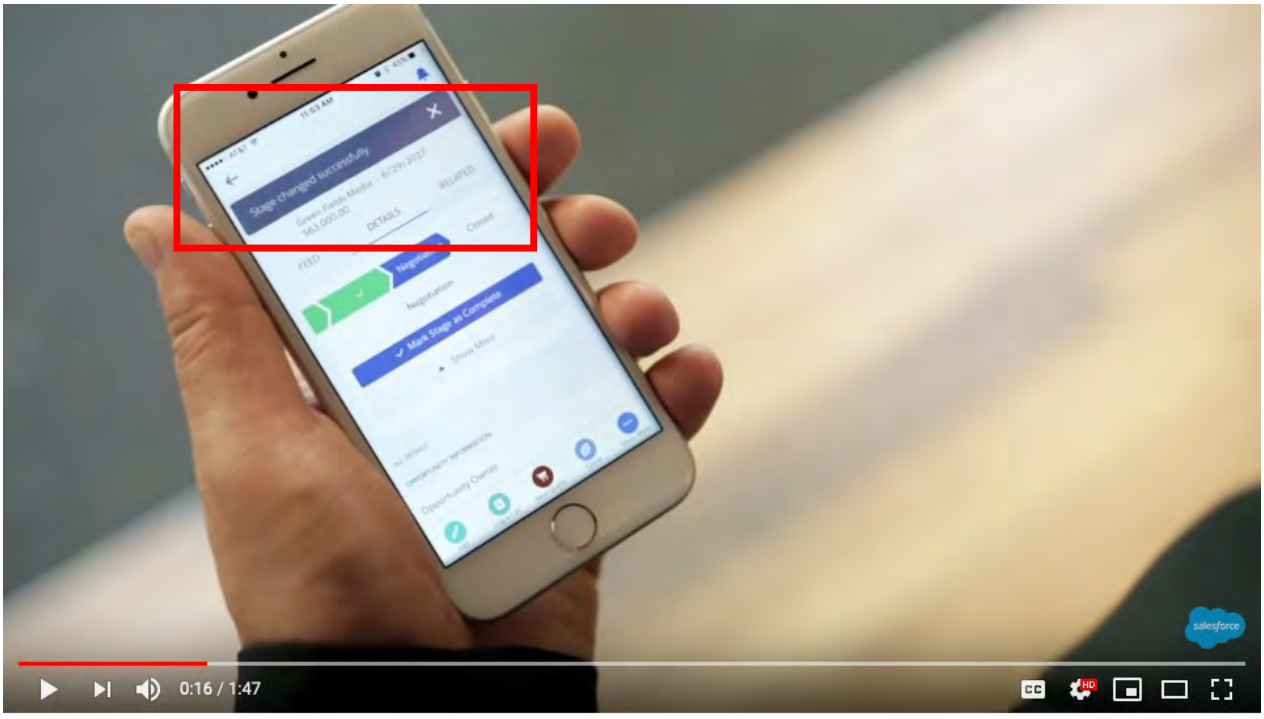
Source: https://youtu.be/XMK5Huuviow 4:20/4:29

137. As another example, the web service will utilize the user's input values for generating output values and return data to the player for generating a confirmation "Stage changed successfully" message.



Introduction to Salesforce Lightning

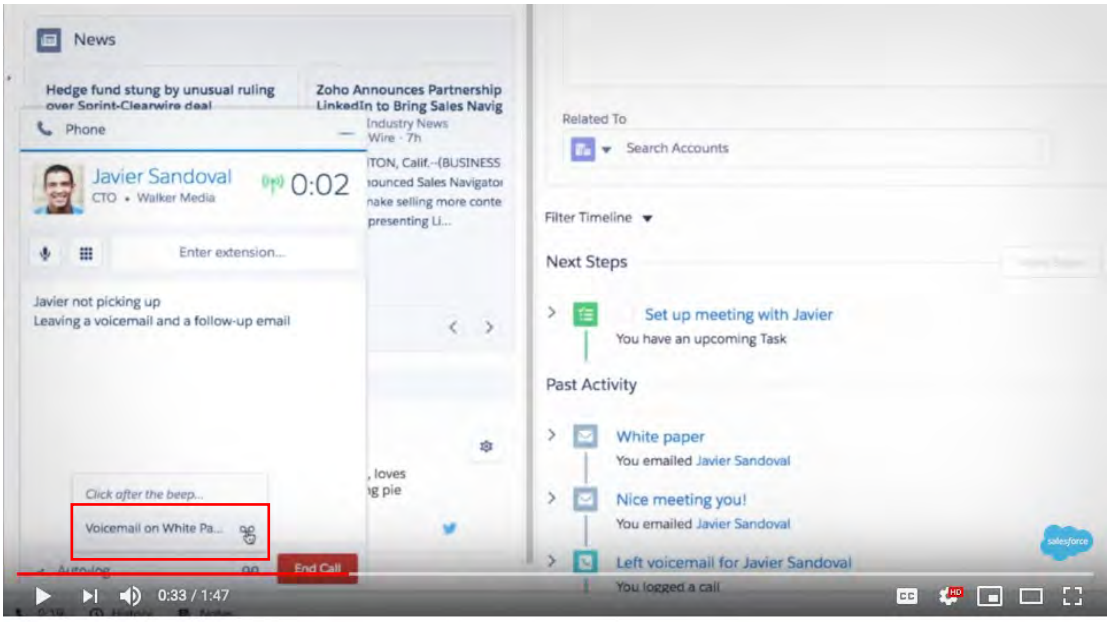
Source: https://www.youtube.com/watch?v=s_sHeac3gfo 0:14/1:47



Introduction to Salesforce Lightning

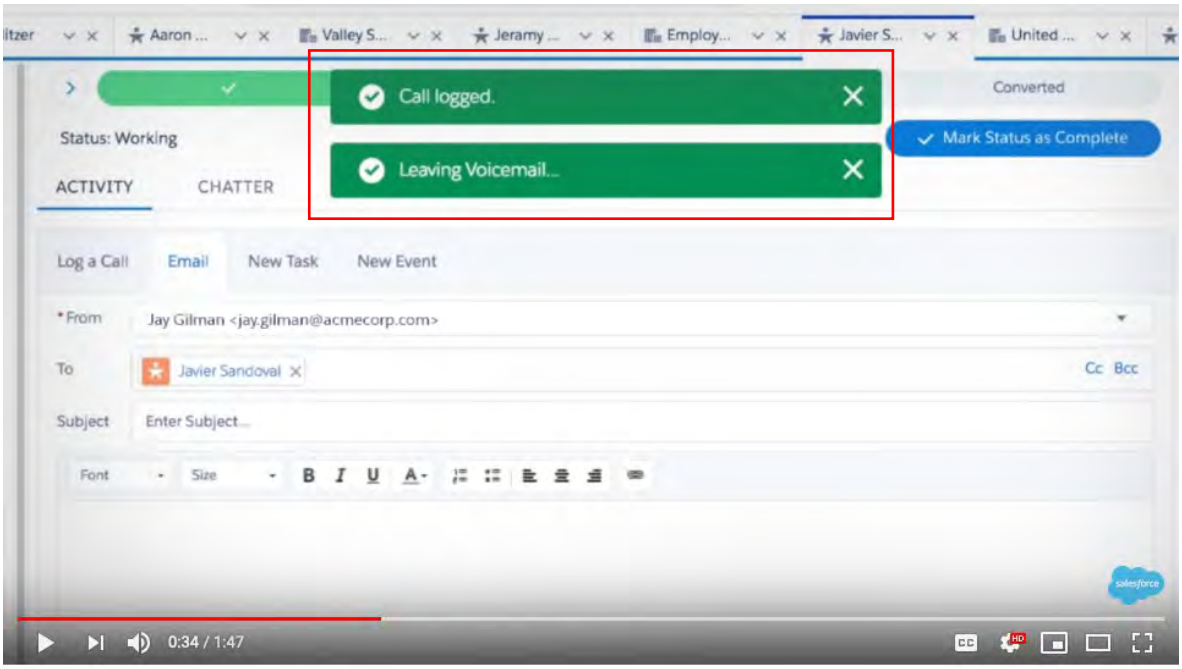
Source: https://www.youtube.com/watch?v=s_sHeac3gfo 0:16/1:47

138. As another example, the web service will use the user's input values for generating output values and return data to the player to verify that a call placed through the system was logged with output symbolic name "Call logged."



Introduction to Salesforce Lightning

Source: https://www.youtube.com/watch?v=s_sHeac3gfo 0:34/1:47



Introduction to Salesforce Lightning

Source: https://www.youtube.com/watch?v=s_sHeac3gfo 0:34/1:47

139. Salesforce was made aware of the '044 patent and its infringement thereof at least as early as November 30, 2020 when Express Mobile provided notice of Salesforce's infringement of the '044 patent to Marc Benioff, Chair and Chief Executive Officer and Amy Weaver, President and Chief Legal Officer of Salesforce. Since at least the time Salesforce received notice, Salesforce has induced others to infringe at least one claim of the '044 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Salesforce's clients, customers, and end users, whose use of the Accused Instrumentality constitutes direct infringement of at least one claim of the '044 patent. In particular, Salesforce's actions that aid and abet others such as customers and end users to infringe include advertising and distributing the Accused Instrumentality and providing instruction materials, training, and services regarding the Accused Instrumentality. *See e.g.*, www.salesforce.com, help.salesforce.com, https://trailhead.salesforce.com, including all related domains and subdomains. Salesforce has engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Salesforce has had actual knowledge of the '044 patent and knowledge that its acts were inducing infringement of the '044 patent since at least the date Salesforce received notice that such activities infringed the '044 patent.

140. Salesforce is liable as a contributory infringer of the '044 patent under 35 U.S.C. § 271(c) by offering to sell, selling and importing into the United States website or web page authoring tools to be especially made or adapted for use in an infringement of the '044 patent. The Accused Instrumentality is a material component for use in practicing the '044 patent, is specifically made and is not a staple article of commerce suitable for substantial non-infringing use.

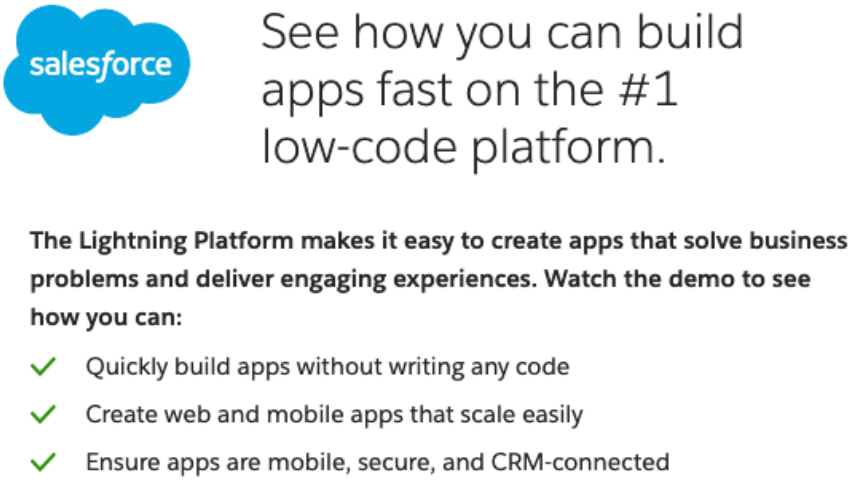
141. Saleforce's infringement has damaged and continues to damage and injure Express Mobile.

COUNT IV - INFRINGEMENT OF U.S. PATENT NO. 9,471,287

142. Plaintiff incorporates by reference the allegations contained in paragraphs 1 to 141 above.

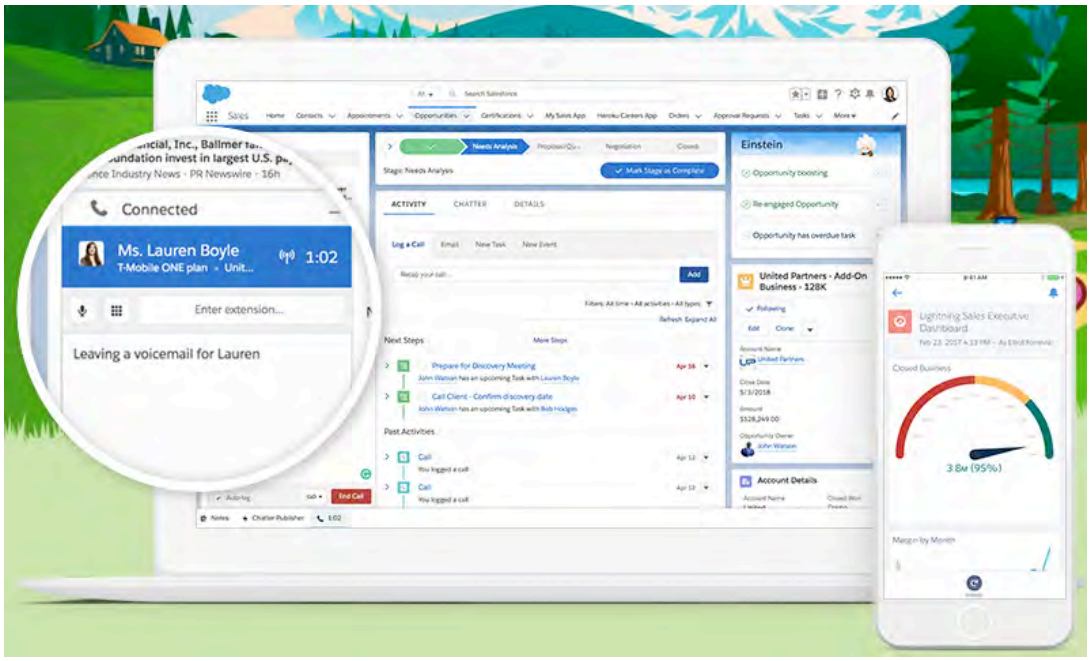
143. Salesforce has manufactured, sold, offered to sell, used and/or provided and continues to manufacture, sell, offer for sale, use and/or provide the Lightning App Builder which is a system for generating code to provide content on a display of a device (the "Accused Instrumentality") which infringes, either literally or under the doctrine of equivalents, one or more claims of the '287 patent in violation of 35 U.S.C. § 271(a).

144. The Accused Instrumentality is a system for generating code to provide content on a display of a device.

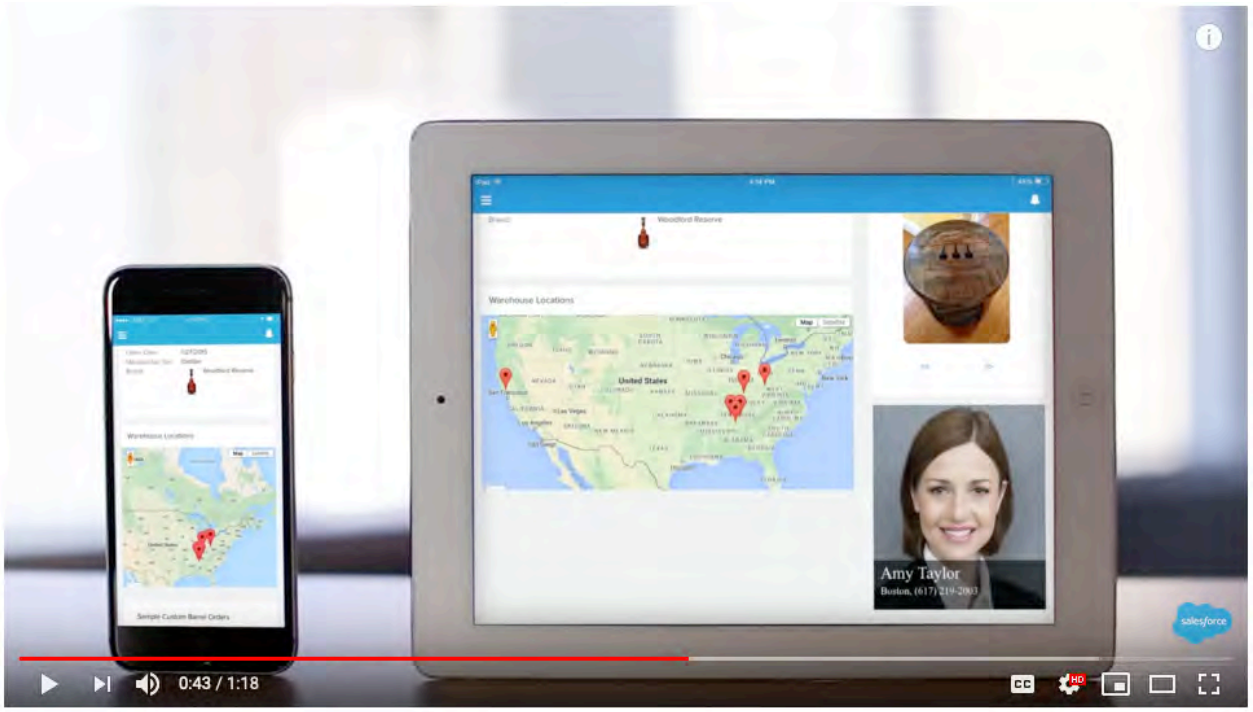


Source: https://www.salesforce.com/form/demo/platform-lightning-demo-sem/

145. The Accused Instrumentality stores items in computer memory in order to serve web pages and apps to devices.



Source: https://www.salesforce.com/campaign/lightning/#

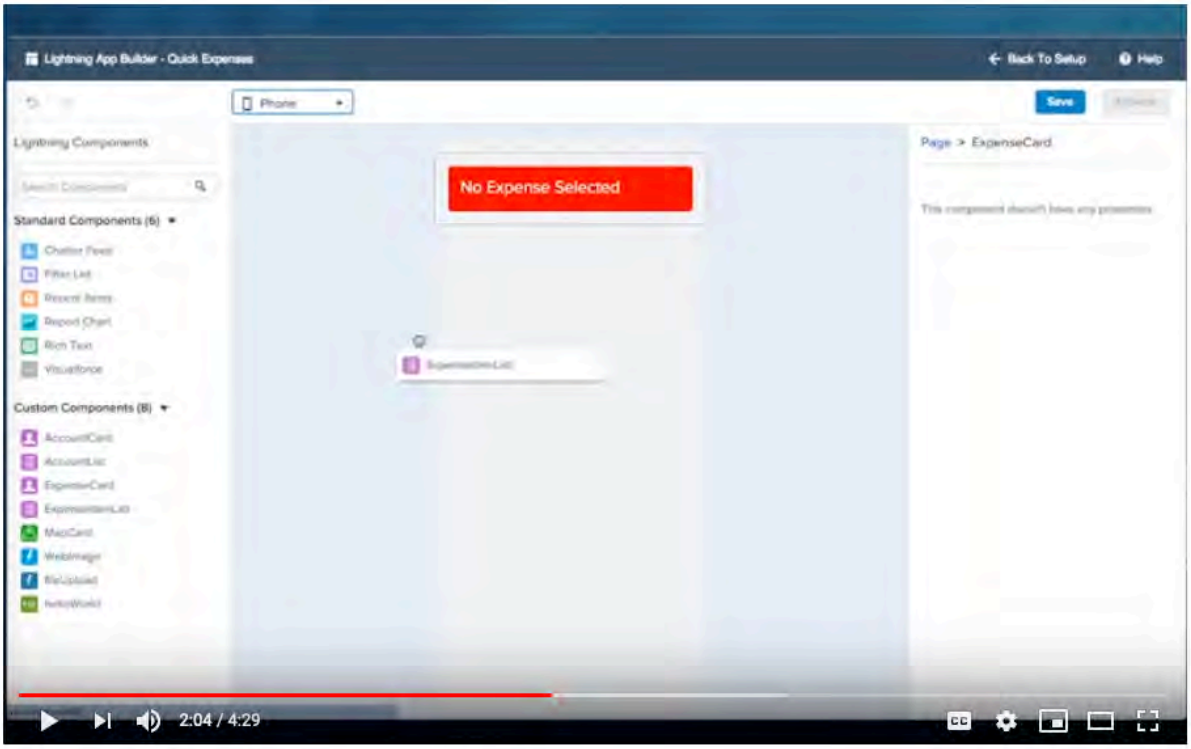


Salesforce Platform: Lightning App Builder Demo

Source: https://www.youtube.com/watch?v=PKW4b_Hk9Hw 0:43/1:18

146. The Accused Instrumentality contains symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over

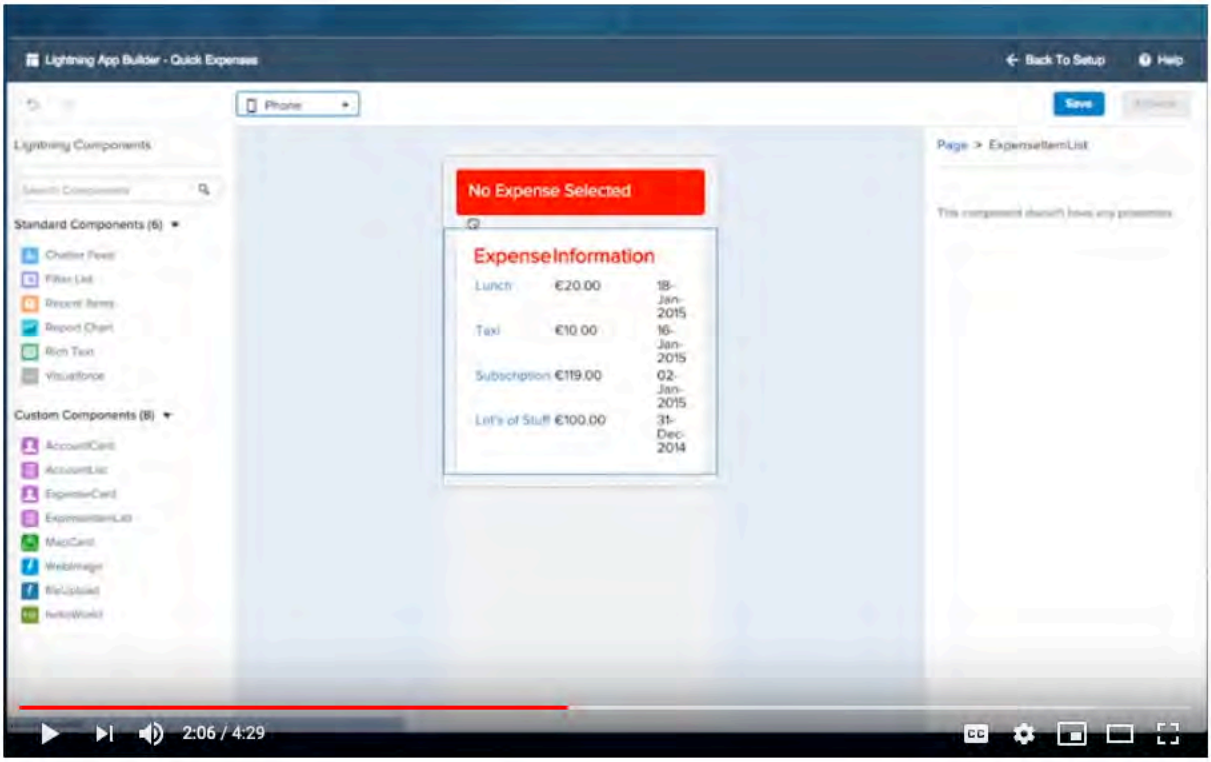
a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service, where each symbolic name has an associated data format class type corresponding to a subclass of User Interface (UI) objects that support the data format type of the symbolic name, and where each symbolic name has a preferred UI object. The Accused Instrumentality contains symbolic names as the names of its components. For example, the symbolic name "ExpenseItemList" is seen below.



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 2:04/4:29

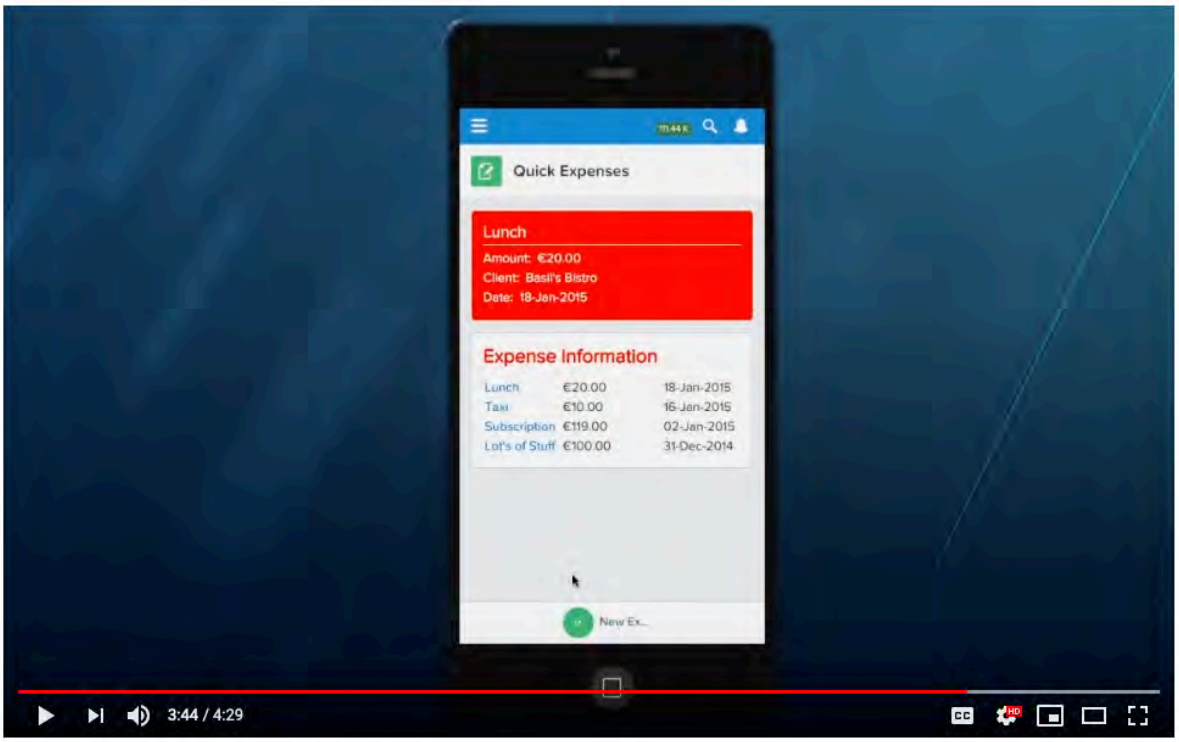
147. The "ExpenseItemList" symbolic name corresponds to the ExpenseItemList web component.



Lightning Intro Screencasts: Lightning App Builder

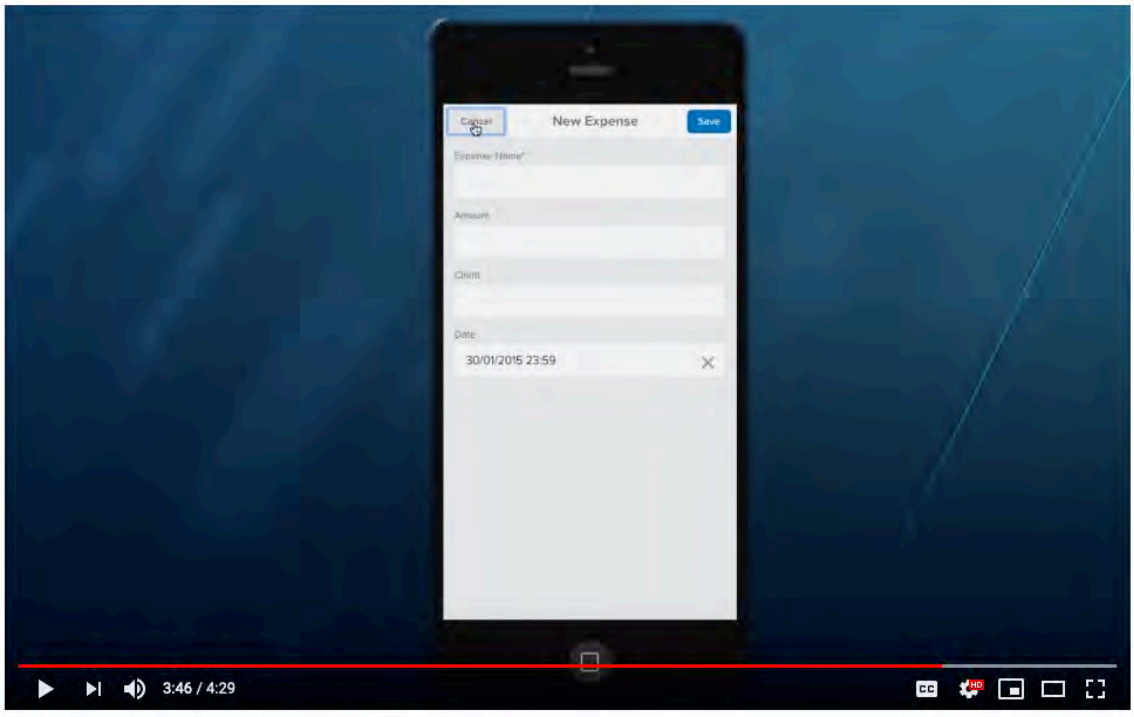
Source: https://youtu.be/XMK5Huuviow 2:06/4:29

148. The "ExpenseItemList" web component widget is related to a set of inputs and outputs of a web service obtainable over a network, as seen by the data populated in the widget, such as the first line item, "Lunch, €20.00, 18-Jan-2015." Users of the app can add additional Expense Items as inputs, and they will be reflected in the Expense Item List as outputs.



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 3:44/4:29



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 3:46/4:29

149. These web components require either a web browser or the Salesforce mobile app to run. Salesforce publishes a list of supported browsers for the Accused Instrumentality.

	MICROSOFT® INTERNET EXPLORER®	MICROSOFT® EDGE	GOOGLE CHROME™	MOZILLA® FIREFOX®	APPLE® SAFARI®
Lightning Experience	IE 11 (EOL December 31, 2020)	Latest	Latest	Latest	12.x+
Lightning Communities	IE 11	Latest	Latest	Latest	12.x+
Special setup considerations?	No	No	No	No	No
Limitations?	Yes	Yes	No	Yes	Yes

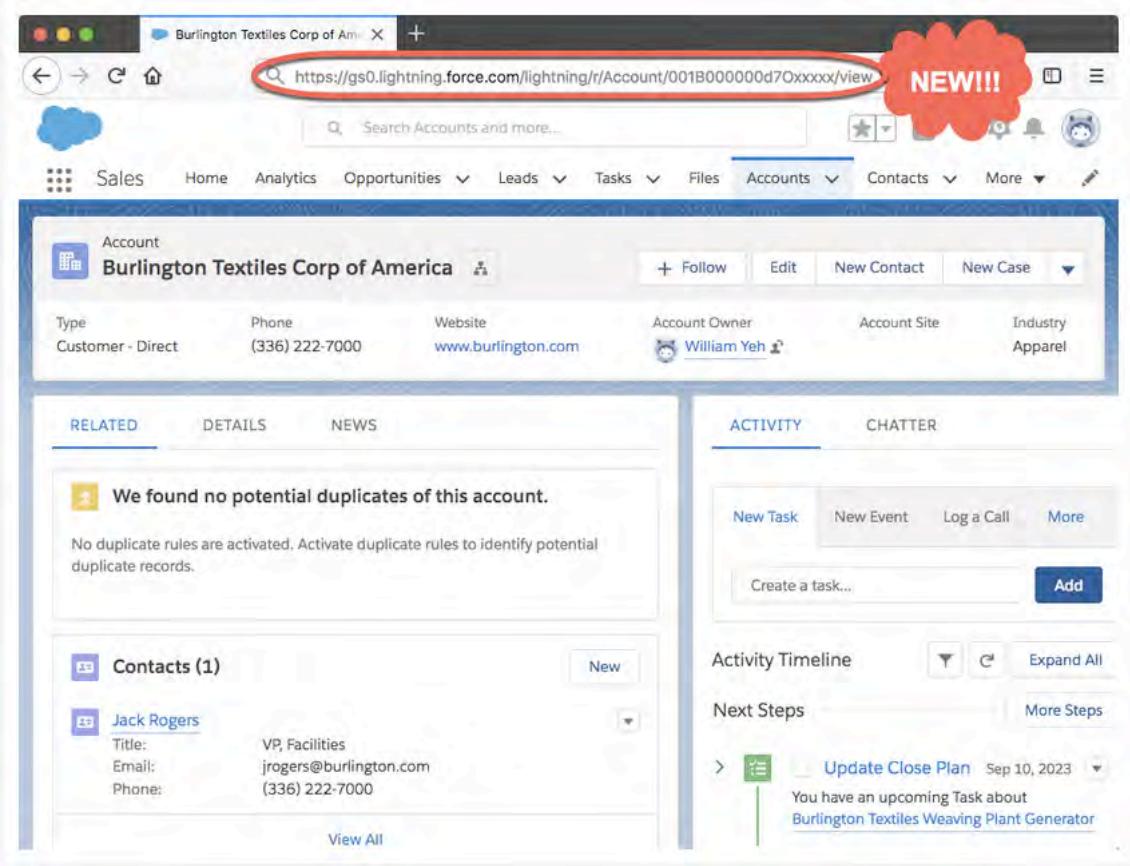
Source: https://help.salesforce.com/articleView?id=getstart_browsers_sfx.htm

150. The address of the web service in the Accused Instrumentality is embedded in the Component that is dragged from the AppStore and seamlessly placed in the Accused Instrumentality's pages in the app builder.

> Your custom Lightning components don't automatically work on Lightning pages or in the Lightning App Builder. To make a custom component usable in both, you need to:
>
> 1. Configure the component and its component bundle so that they're compatible with the Lightning App Builder and Lightning pages.
> 2. Deploy My Domain in your org. When you deploy My Domain, references and links to Lightning resources are in the format `https://<myDomain>.lightning.force.com`.

https://help.salesforce.com/articleView?id=lightning_page_components_custom.htm&type=5

151. The Accused Instrumentality stores an address of the web service and stores it in computer memory. For example, the Accused Instrumentality's web pages such as the "Burlington Textiles Corp of America" served to users contain addresses. These addresses are stored in computer memory on at least Salesforce's servers. On information and belief, the mobile app also contains the address of the web service.

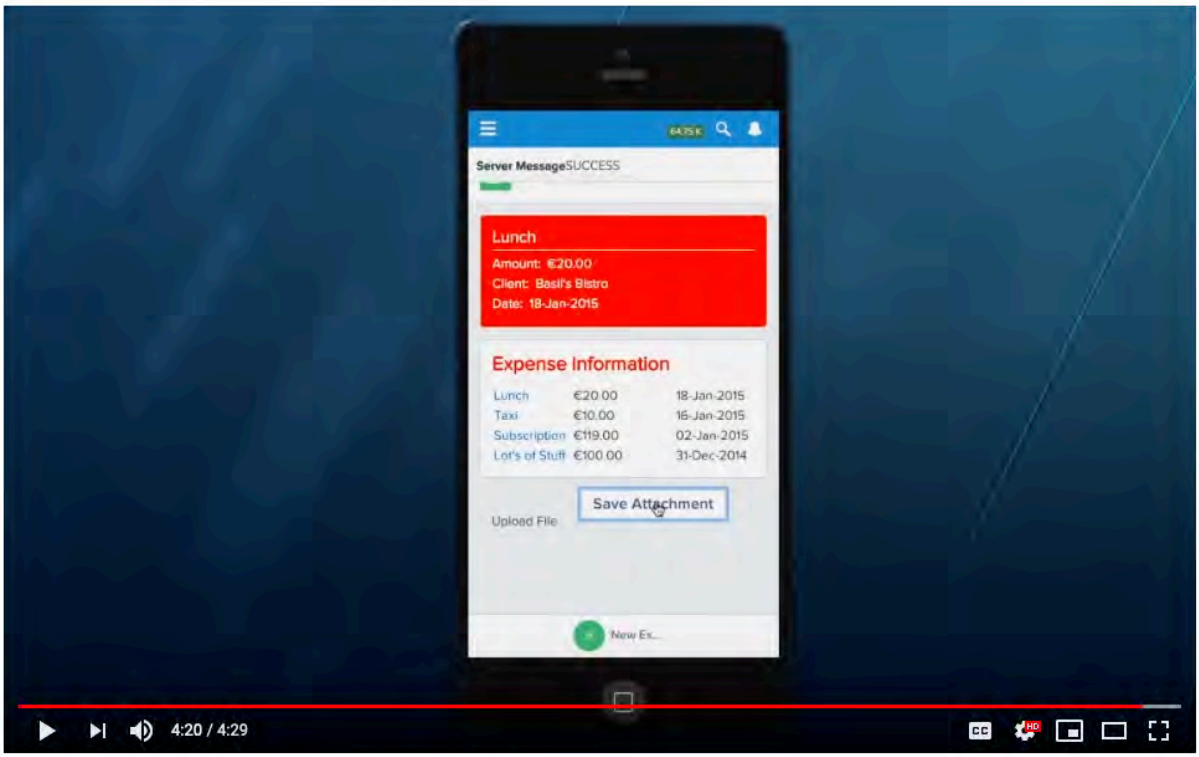


Source:https://developer.salesforce.com/blogs/2018/01/heres-need-know-new-url-format-lightning.html

Example of new URLs
Original format: Account Home URL
https://<lightning.domain.com>/one/one.app/#/sObject/Account/home
New format: Account Home URL
https://<lightning.domain.com>/lightning/o/Account/home

Original format: Account Record View
https://<lightning.domain.com>/one/one.app#/sObject/<recordID>/view
New format: Account Record View
https://<lightning.domain.com>/lightning/r/Account/<recordID>/view

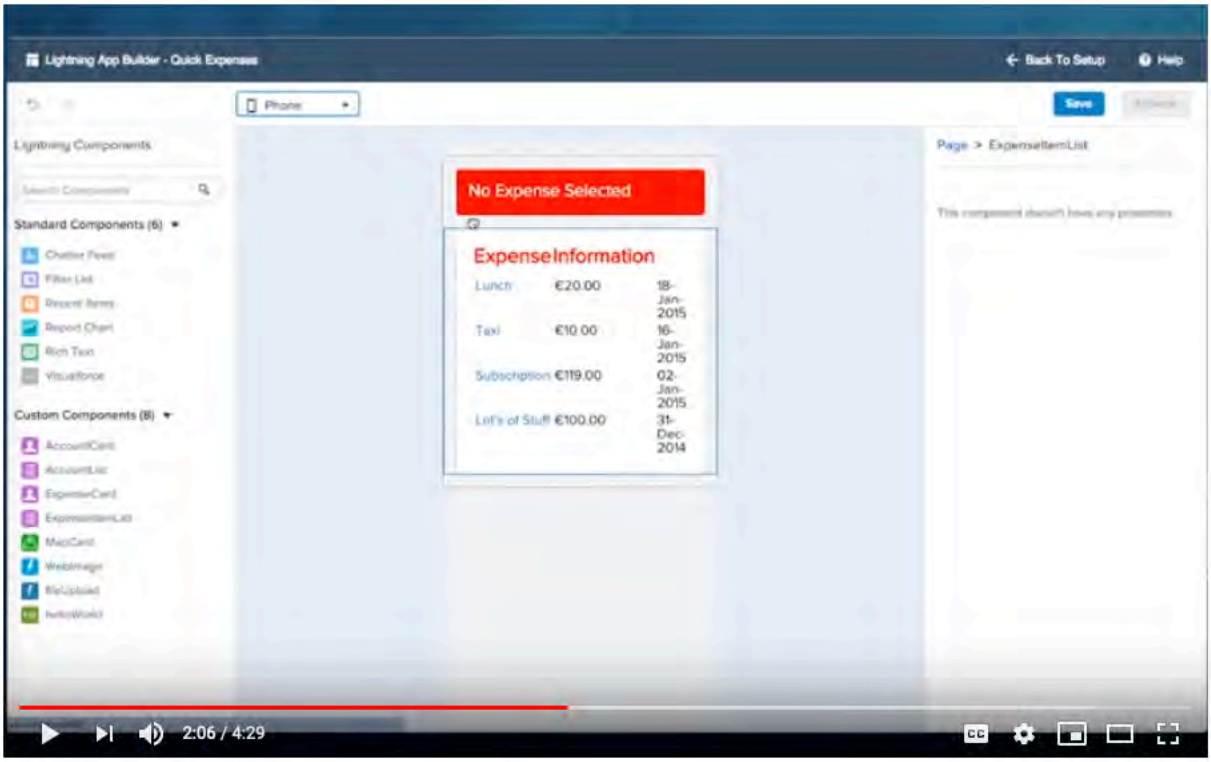
Source: https://developer.salesforce.com/blogs/2018/01/heres-need-know-new-url-format-lightning.html



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 4:20/4:29

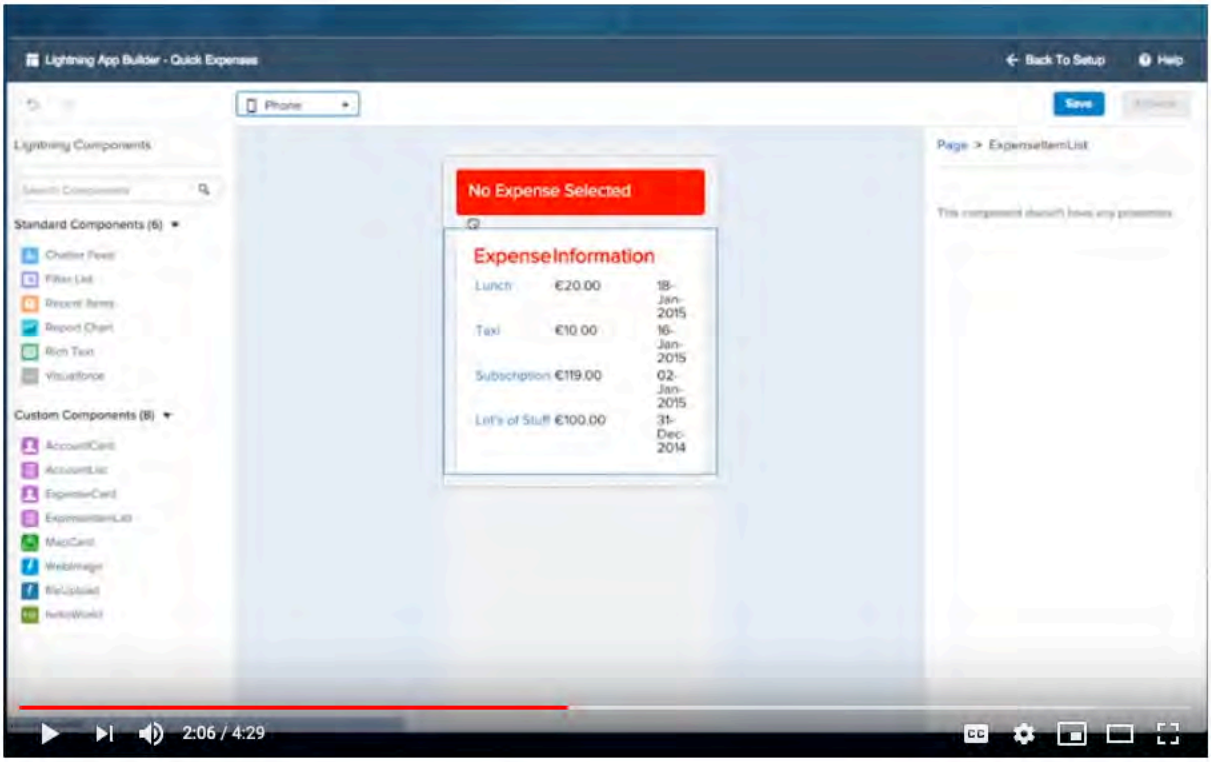
152. The Accused Instrumentality contains an authoring tool configured to define a UI object for presentation on the display. For example, the Lightning App Builder tool configures an ExpenseItemList object for presentation on the display, titled Expense Information (hereafter "Expense Information").



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 2:06/4:29

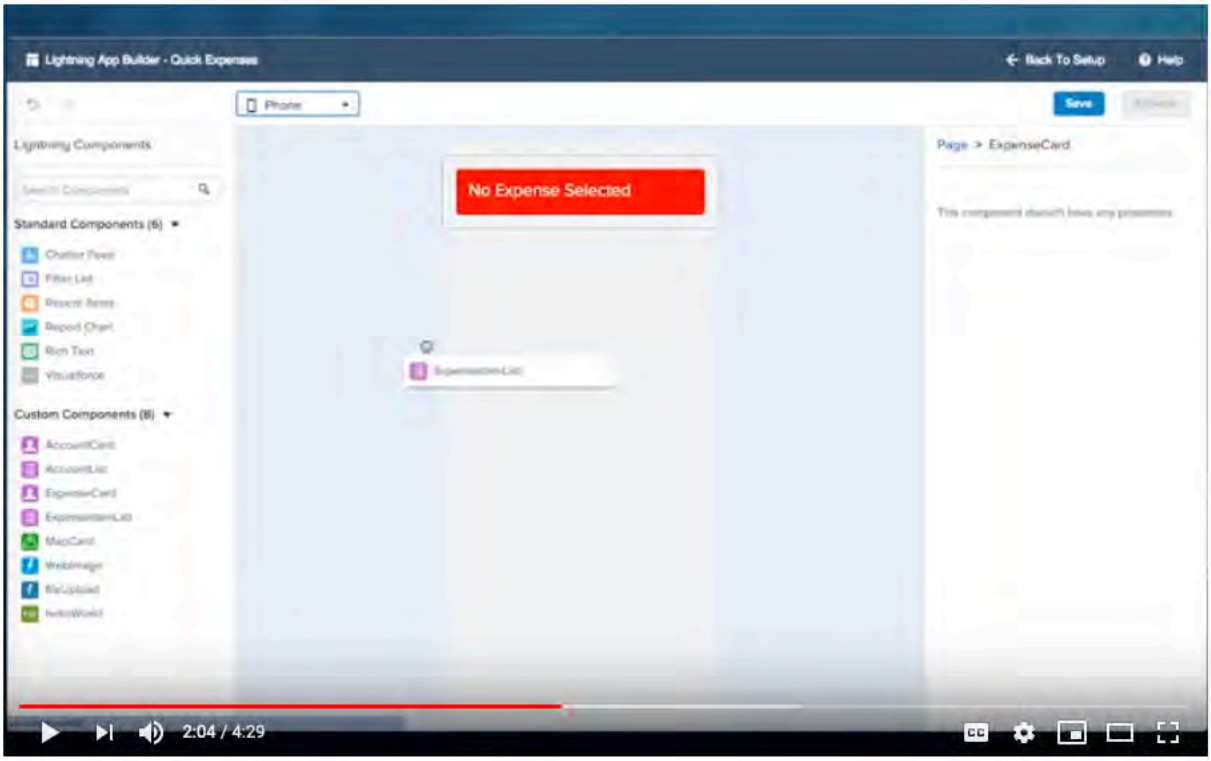
153. The UI object corresponds to a web component included in the computer memory selected from a group consisting of an input of the web service and an output of the web service, where each defined UI object is either: 1) selected by a user of the authoring tool; or 2) automatically selected by the system as the preferred UI object corresponding to the symbolic name of the web component selected by the user of the authoring tool. The defined Expense Information UI object in the center of the screen corresponds to the ExpenseItemList web component on the left side of the screen. The available web components are listed next to the ExpenseItemList web component and are stored in memory on Salesforce servers.



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 2:06/4:29

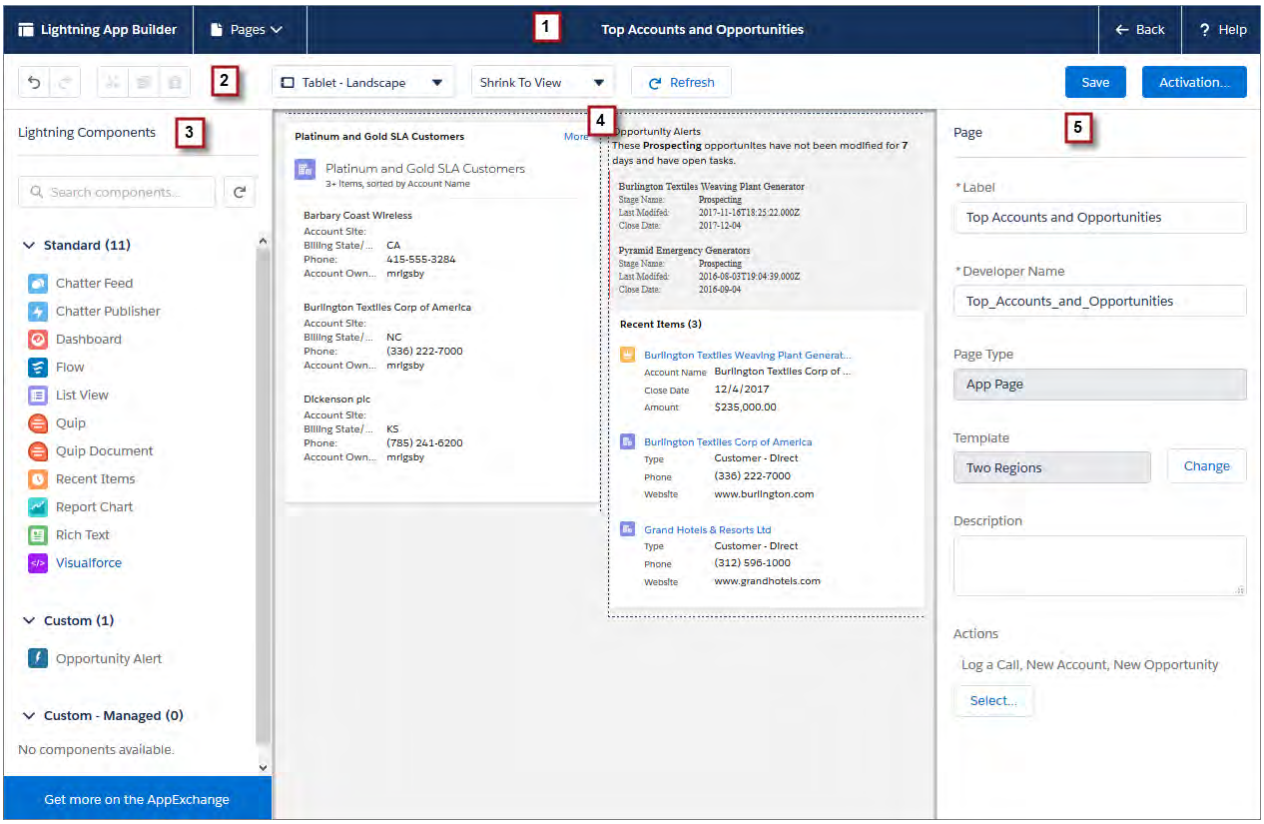
154. The defined Expense Information UI object is selected by the user. Since the UI object is a Custom Component, it contains information selected by the user.



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 2:04/4:29

155. Other defined UI objects, such as Standard Components, might be automatically selected by the system as the preferred UI object corresponding to the symbolic name of the web component selected by the user of the authoring tool.



Source:
https://trailhead.salesforce.com/content/learn/modules/lightning_app_builder/lightning_app_builder_intro

Lightning Components Pane (3)

The components pane contains all standard and custom Lightning components that are supported for your Lightning page. Click and drag a component to add it to the page.

Tip

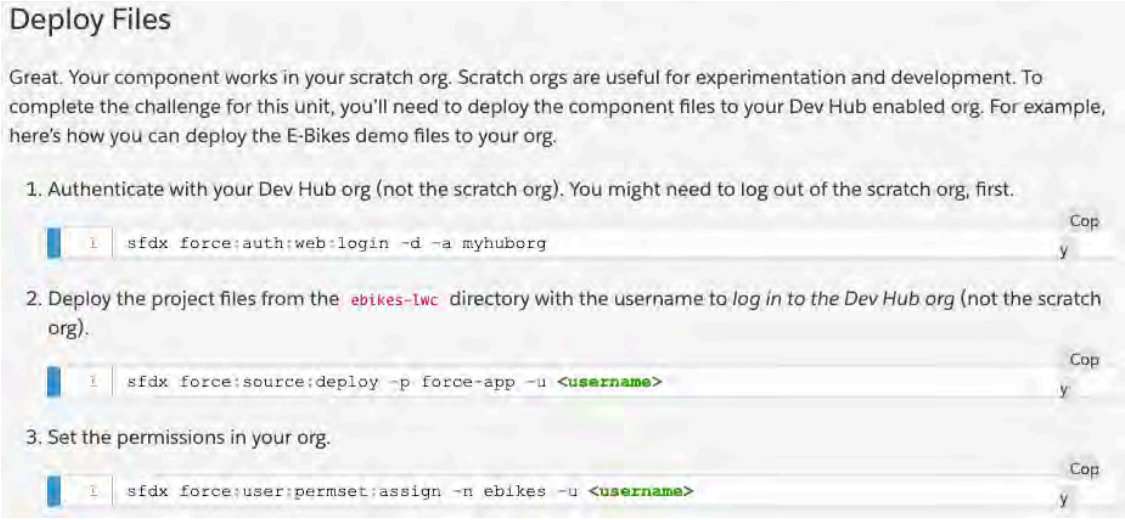
If you have a lot of custom components, enter text in the search field to easily find the one you need. You can access third-party custom components on the AppExchange using the button at the bottom of the pane.

Lightning Page Canvas (4)

The canvas area is where you build your page. Drag components to reorder them on the page.

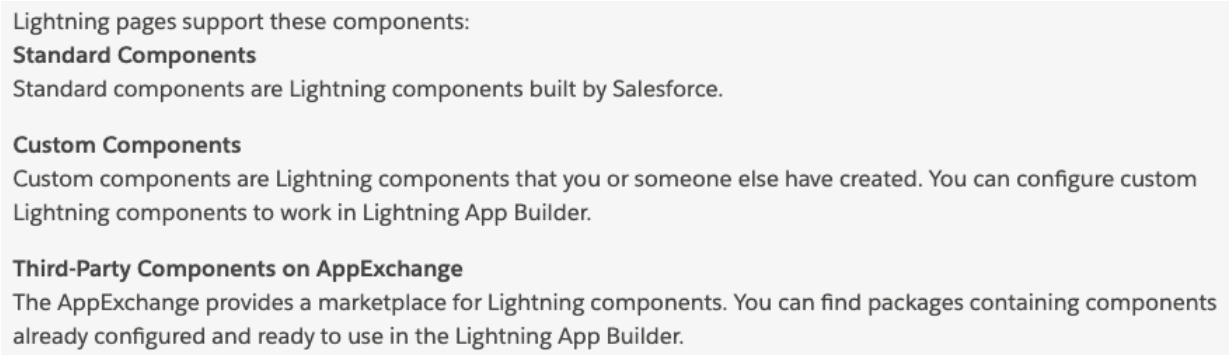
Source:
https://trailhead.salesforce.com/content/learn/modules/lightning_app_builder/lightning_app_builder_intro

156. Custom Components are uploaded and saved on Salesforce servers.



Source: https://trailhead.salesforce.com/content/learn/modules/lightning-web-components-basics/push-lightning-web-component-files

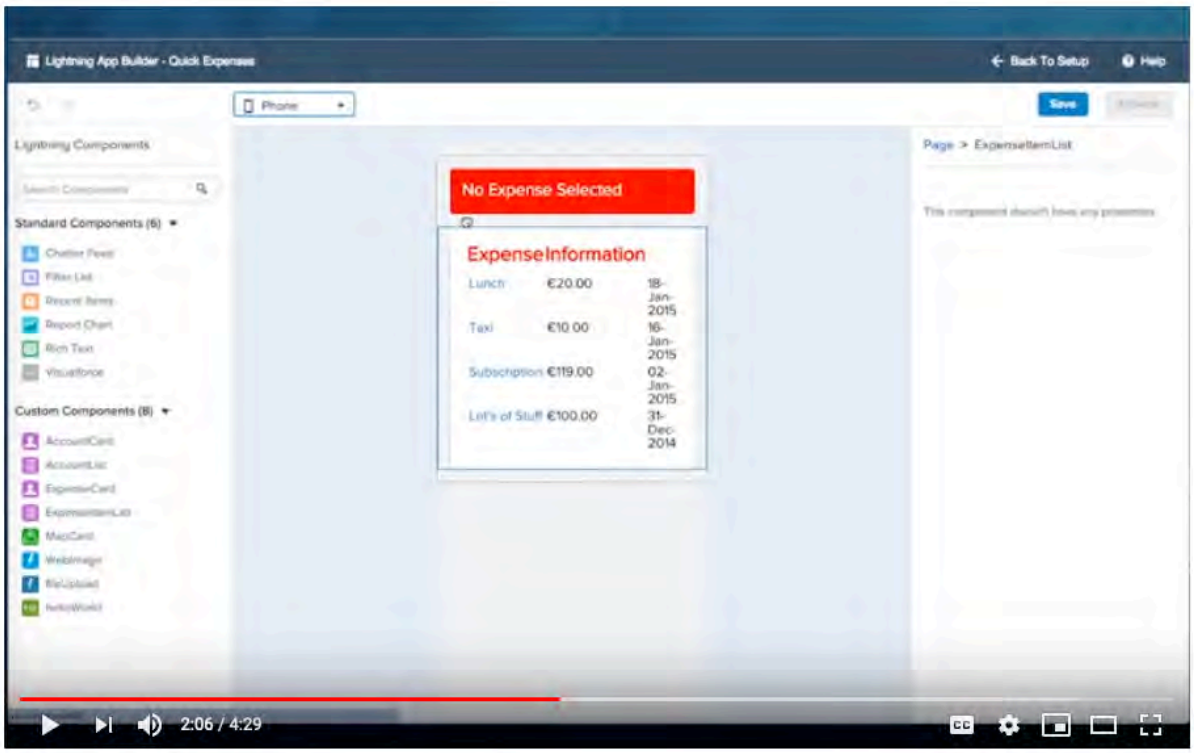
157. Standard Components and third-party created Components are both stored on Salesforce servers. Standard Components are available on all projects, and third-party created Components can be added through Salesforce's "AppExchange."



Source: https://trailhead.salesforce.com/content/learn/modules/lightning_app_builder/lightning_app_builder_intro

158. The authoring tool in the Accused Instrumentality is configured to associate the selected symbolic name with the defined UI object, where the selected symbolic name is only available to UI objects that support the defined data format associated with that symbolic name. For example, the authoring tool associates the "ExpenseItemList" symbolic name with the defined

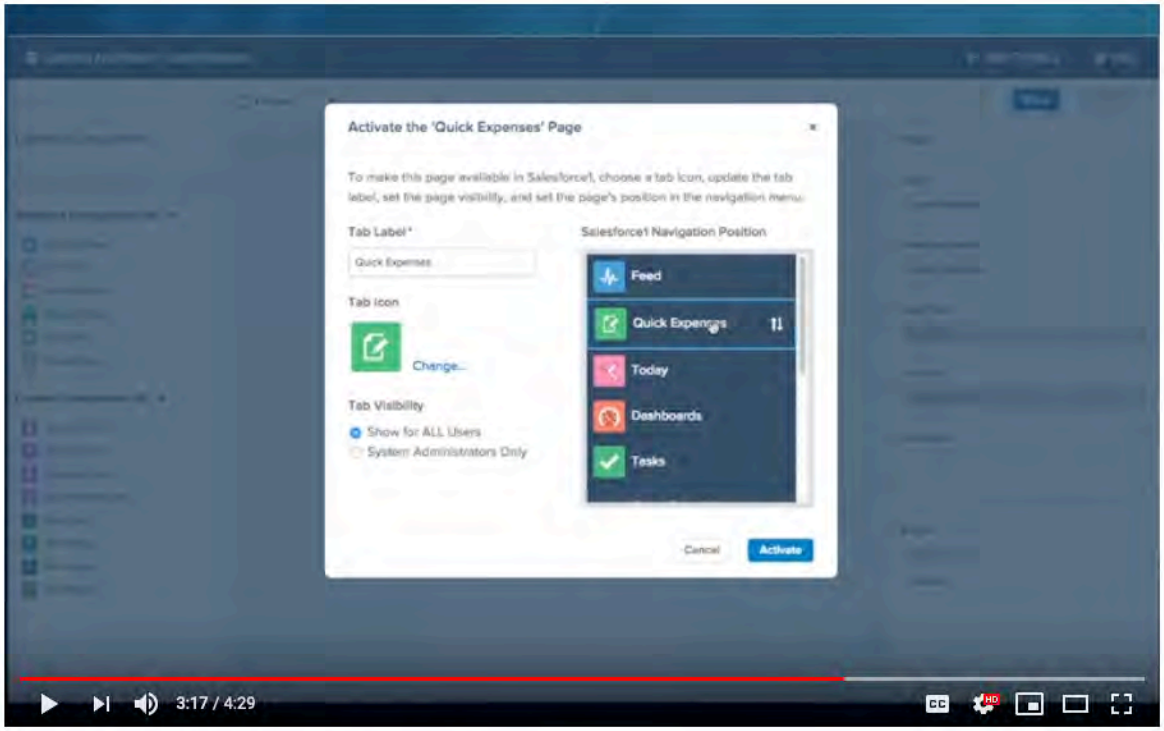
UI object, where the symbolic name is only available to UI objects that support the data of the ExpenseItemList.



Lightning Intro Screencasts: Lightning App Builder

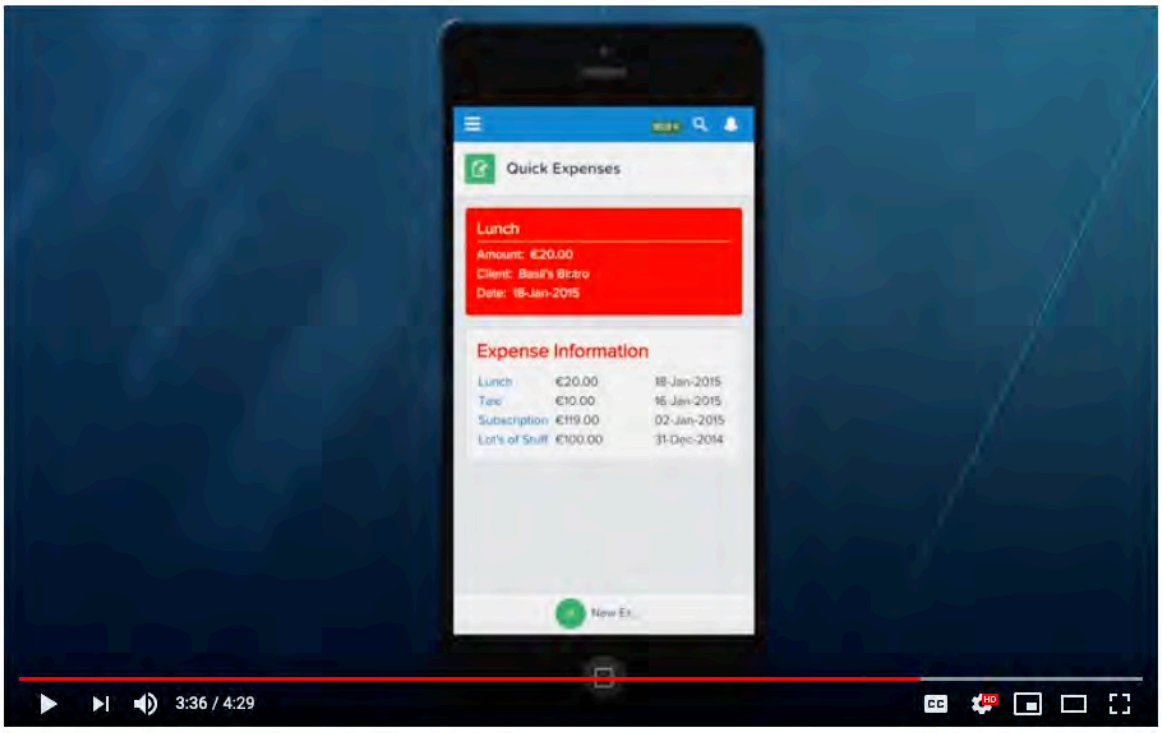
Source: https://youtu.be/XMK5Huuviow 2:06/4:29

159. The authoring tool in the Accused Instrumentality is configured to build an application consisting of one or more web page views from at least a portion of the database utilizing at least one player, where the player utilizes information stored in the database to generate for the display of at least a portion of the one or more web pages and the application and player are provided to the device and executed on the device. For example, once the user clicks "save" and "activate" on the authoring tool, the application is saved and built. The authoring tool builds and delivers application code that uses information stored in the database that is independent of the device that it is displayed on.



Lightning Intro Screencasts: Lightning App Builder

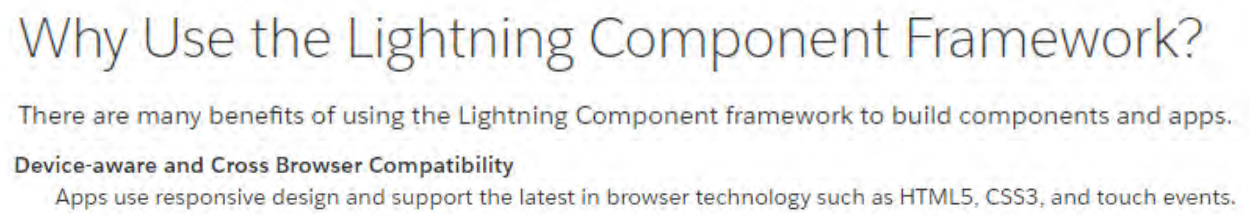
Source: https://www.youtube.com/watch?v=XMK5Huuviow 3:17/4:29



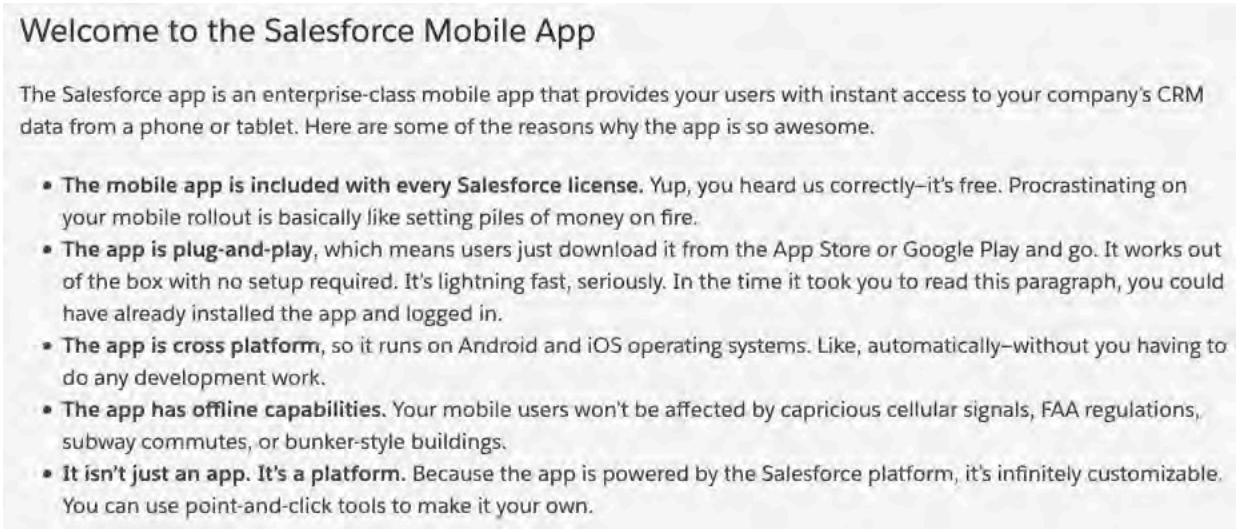
Lightning Intro Screencasts: Lightning App Builder

Source: https://help.salesforce.com/articleView?id=aura_features.htm&type=5

160. The Salesforce phone app shown consists of one or more web page views from the Salesforce server, utilizing the player embedded in the mobile app, where the app information from the Salesforce server is used to generate the information shown, such as the Expense Information. The player consists of device-dependent code or code that is specific to the operating system, programming language, or platform of the client device (e.g., "device-aware").



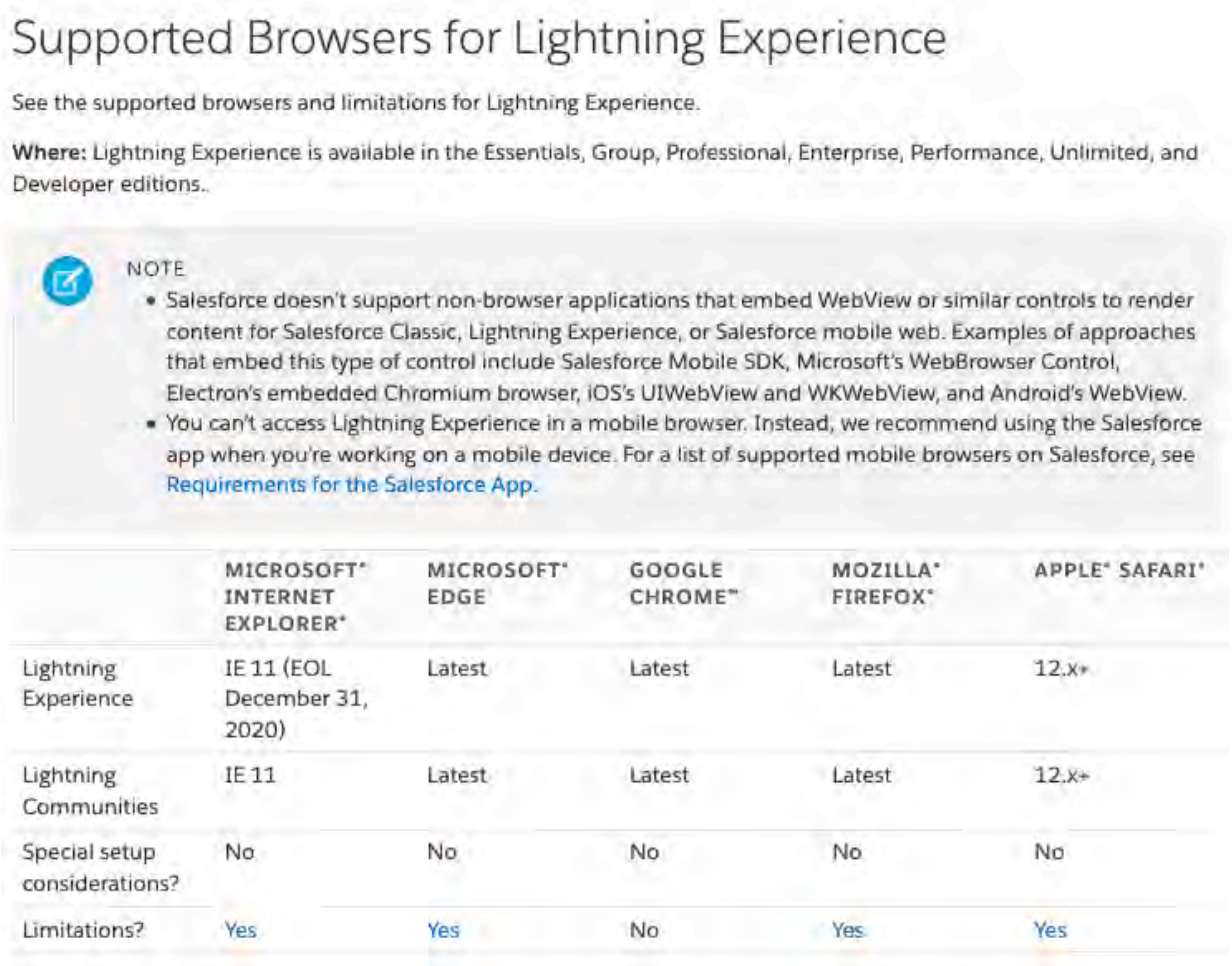
Source: https://help.salesforce.com/articleView?id=aura_features.htm&type=5



Source: https://trailhead.salesforce.com/content/learn/modules/salesforce1_mobile_app/salesforce1_mobile_app_intro

161. Similarly, the app can be accessed through a web browser on a standard computer, subject to the requirements of using a supported browser. The listed web browsers rely on browser engines to interpret and execute JavaScript and HTML to render web pages on a computer. Internet Explorer has relied on Trident (code name for MSHTML) (which has included the Chakra JavaScript Engine, see http://en.wikipedia.org/wiki/MSHTML); Safari and Chrome rely on Webkit (which includes WebCore and JavaScript Core, see http://en.wikipedia.org/wiki/WebKit);

Firefox relies on Gecko (which includes Spidermonkey, see http://www.mozilla.org/projects/technologies.html).



Supported Browsers for Lightning Experience

See the supported browsers and limitations for Lightning Experience.

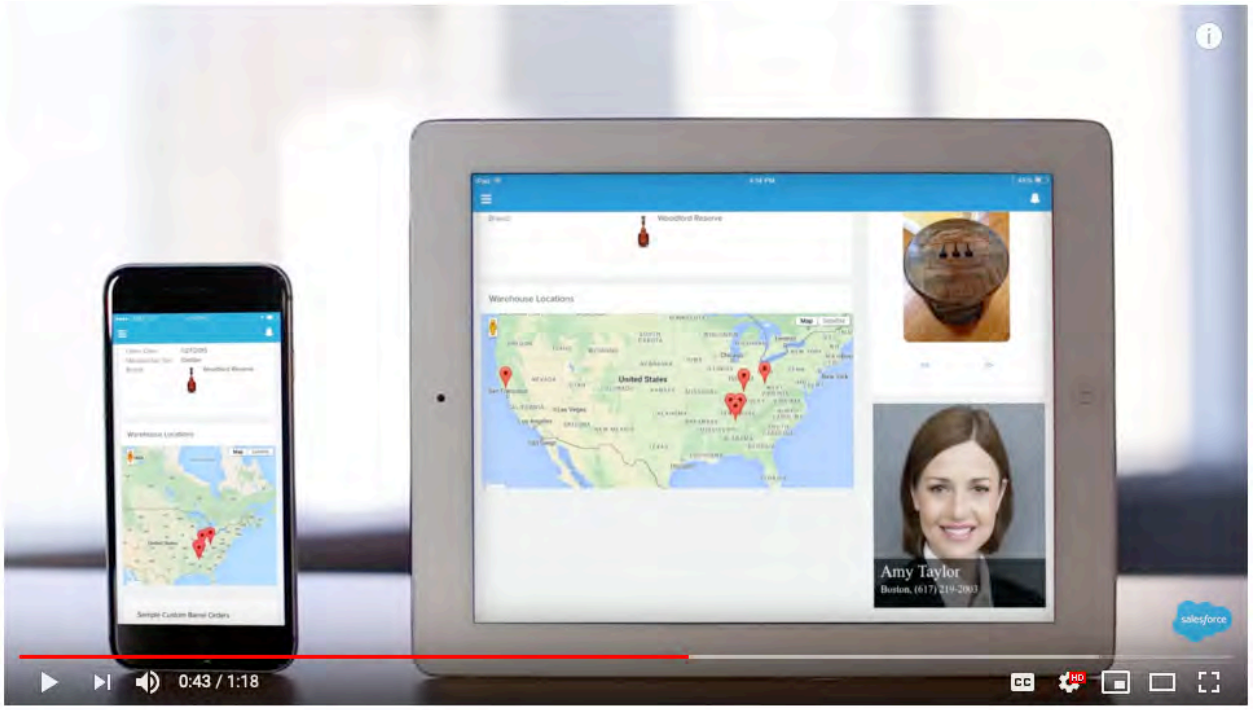
Where: Lightning Experience is available in the Essentials, Group, Professional, Enterprise, Performance, Unlimited, and Developer editions.

NOTE

- Salesforce doesn't support non-browser applications that embed WebView or similar controls to render content for Salesforce Classic, Lightning Experience, or Salesforce mobile web. Examples of approaches that embed this type of control include Salesforce Mobile SDK, Microsoft's WebBrowser Control, Electron's embedded Chromium browser, iOS's UIWebView and WKWebView, and Android's WebView.
- You can't access Lightning Experience in a mobile browser. Instead, we recommend using the Salesforce app when you're working on a mobile device. For a list of supported mobile browsers on Salesforce, see Requirements for the Salesforce App.

	MICROSOFT® INTERNET EXPLORER®	MICROSOFT® EDGE	GOOGLE CHROME™	MOZILLA® FIREFOX®	APPLE® SAFARI®
Lightning Experience	IE 11 (EOL December 31, 2020)	Latest	Latest	Latest	12.x+
Lightning Communities	IE 11	Latest	Latest	Latest	12.x+
Special setup considerations?	No	No	No	No	No
Limitations?	Yes	Yes	No	Yes	Yes

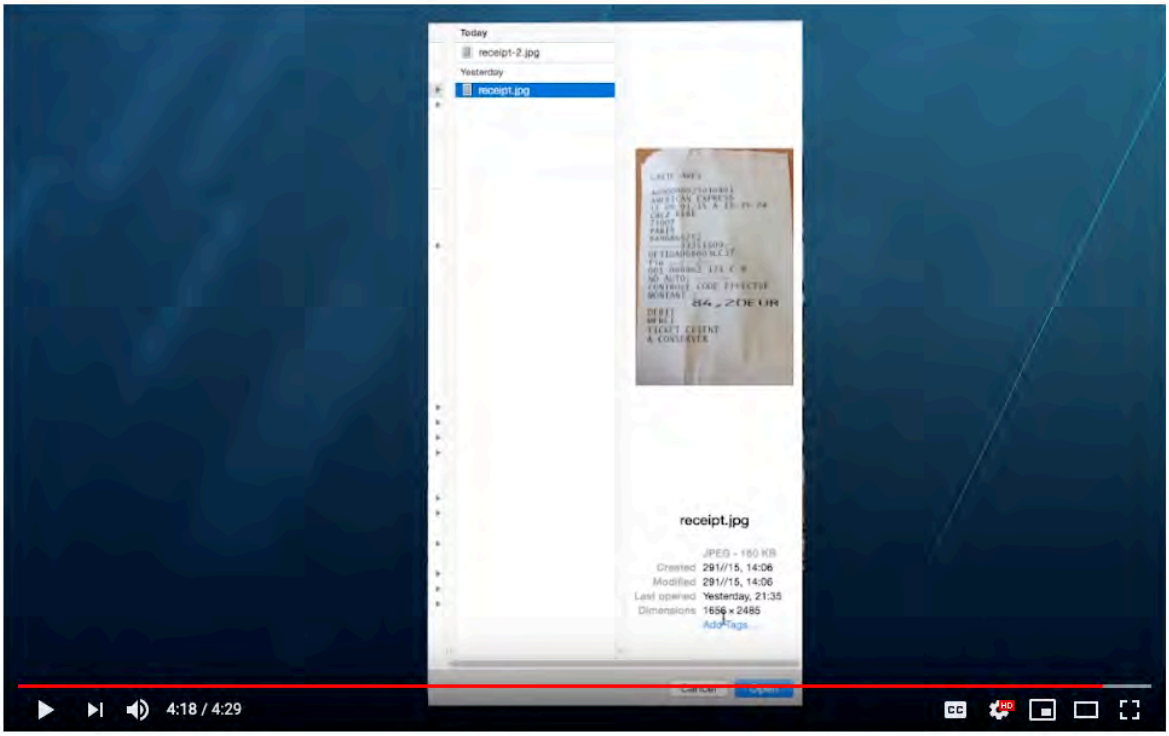
Source: https://help.salesforce.com/articleView?id=getstart_browsers_sfx.htm



Salesforce Platform: Lightning App Builder Demo

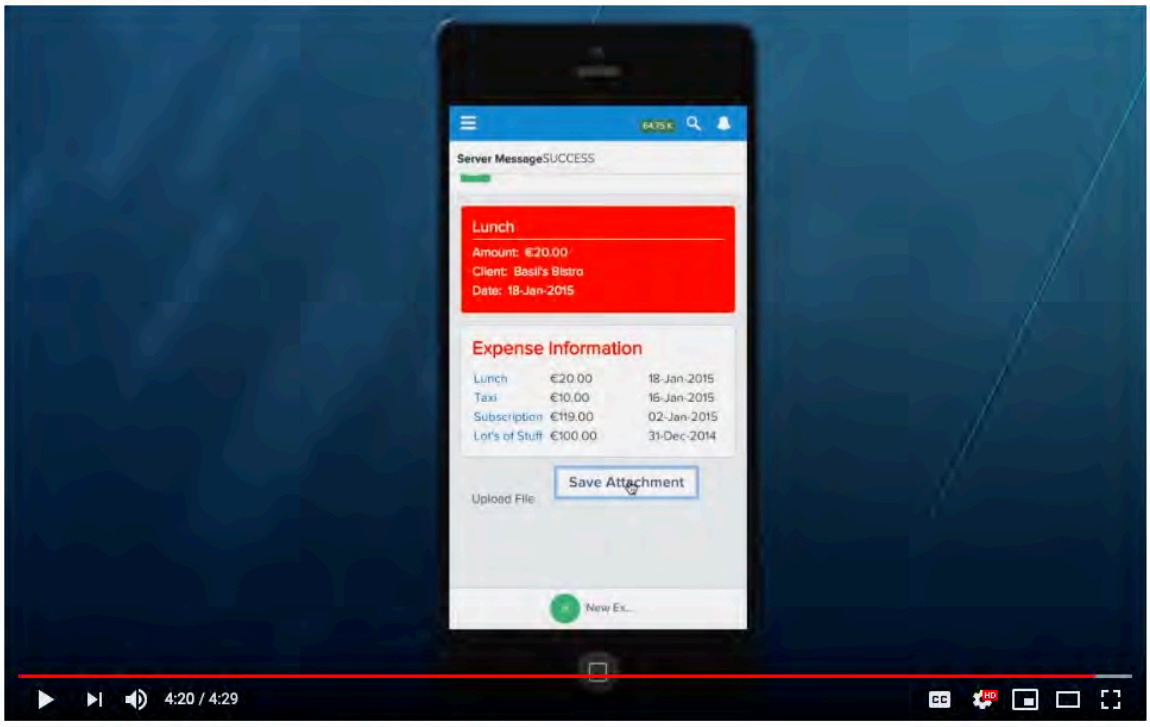
Source: https://www.youtube.com/watch?v=PKW4b_Hk9Hw 0:43/1:18

162. When the user of the device provides one or more input values associated with an input symbolic name to an input of the defined UI object. The device provides the user provided one or more input values and corresponding input symbolic name to the Salesforce Lightning web service. The web service utilizes the input symbolic name and the user provided one or more input values for generating one or more output values having an associated output symbolic name. The player receives the output symbolic name and corresponding one or more output values and provides instructions for the display of the device to present an output value in the defined UI object. For example, the web service will use the user's input values for generating output values and return data to the player for generating a confirmation "SUCCESS" message after the user has uploaded a receipt.



Lightning Intro Screencasts: Lightning App Builder

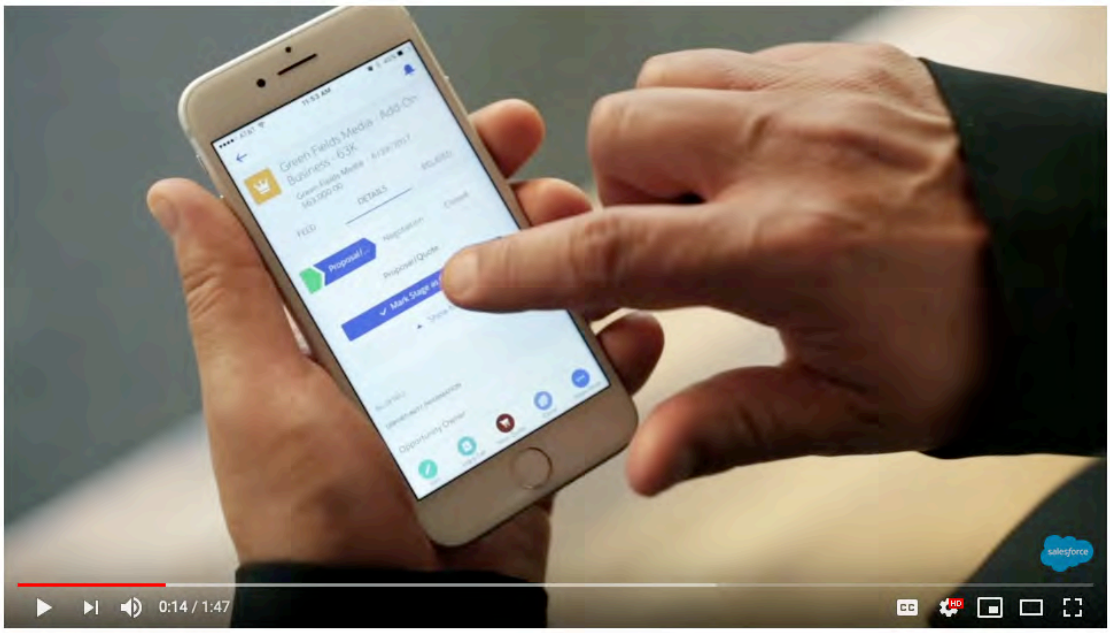
Source: https://youtu.be/XMK5Huuviow 4:18/4:29



Lightning Intro Screencasts: Lightning App Builder

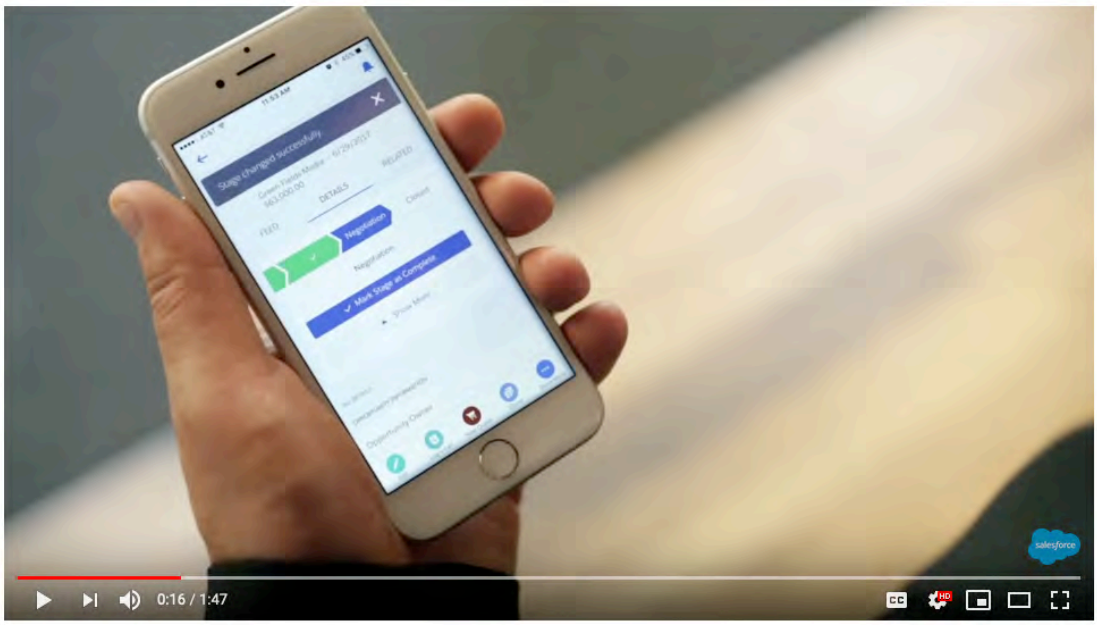
Source: https://youtu.be/XMK5Huuviow 4:20/4:29

163. As another example, the web service will utilize the user's input values for generating output values and return data to the player for generating a confirmation "Stage changed successfully" message.



Introduction to Salesforce Lightning

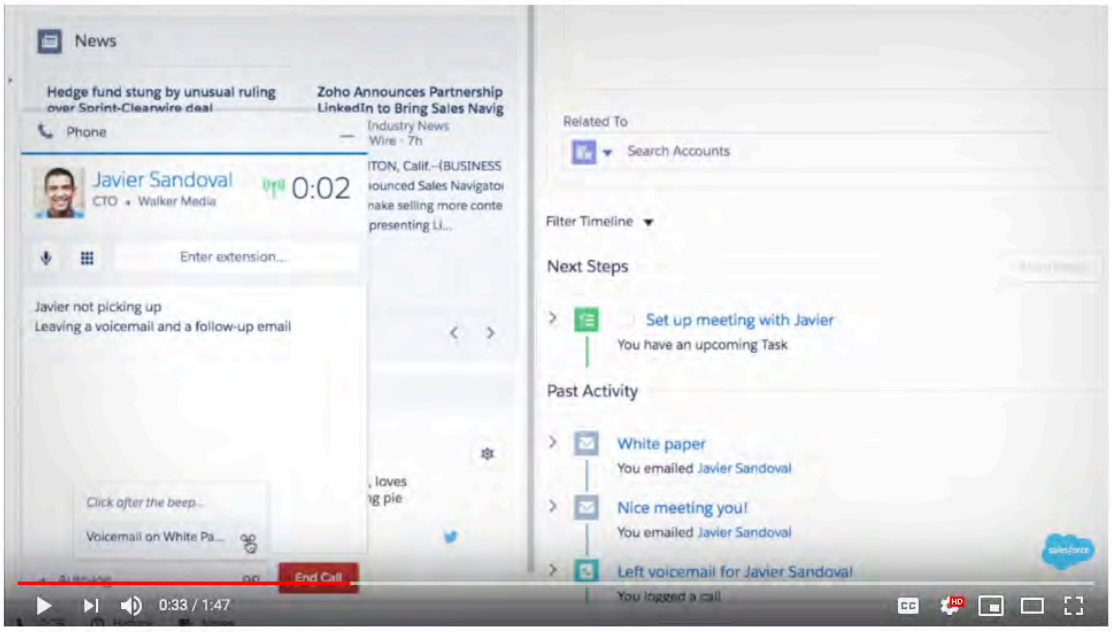
Source: https://www.youtube.com/watch?v=s_sHeac3gfo 0:14/1:47



Introduction to Salesforce Lightning

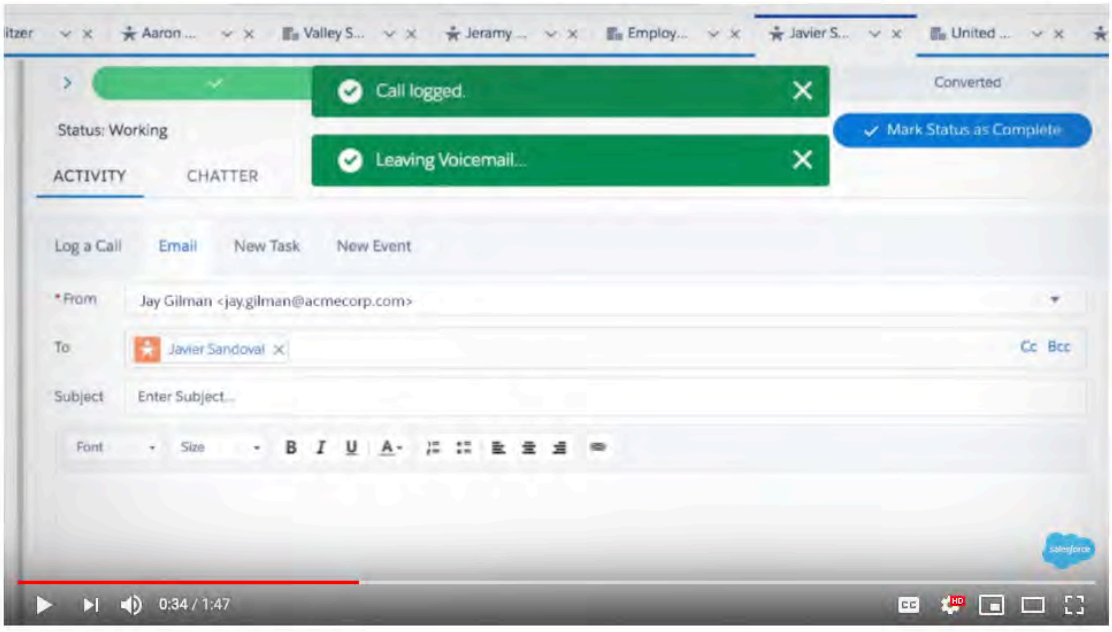
Source: https://www.youtube.com/watch?v=s_sHeac3gfo 0:16/1:47

164. As another example, the web service will use the user's input values for generating output values and return data to the player to verify that a call placed through the system was logged with output symbolic name "Call logged."



Introduction to Salesforce Lightning

Source: https://www.youtube.com/watch?v=s_sHeac3gfo 0:34/1:47



Introduction to Salesforce Lightning

Source: https://www.youtube.com/watch?v=s_sHeac3gfo 0:34/1:47

165. Salesforce was made aware of the '287 patent and its infringement thereof at least as early as November 30, 2020 when Express Mobile provided notice of Salesforce's infringement of the '287 patent to Marc Benioff, Chair and Chief Executive Officer and Amy Weaver, President and Chief Legal Officer of Salesforce. Since at least the time Salesforce received notice, Salesforce has induced others to infringe at least one claim of the '287 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Salesforce's clients, customers, and end users, whose use of the Accused Instrumentality constitutes direct infringement of at least one claim of the '287 patent. In particular, Salesforce's actions that aid and abet others such as customers and end users to infringe include advertising and distributing the Accused Instrumentality and providing instruction materials, training, and services regarding the Accused Instrumentality. *See e.g.*, www.salesforce.com, help.salesforce.com, https://trailhead.salesforce.com , including all related domains and subdomains. Salesforce has engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Salesforce has had actual knowledge of the '287 patent and knowledge that its acts were inducing infringement of the '287 patent since at least the date Salesforce received notice that such activities infringed the '287 patent.

166. Salesforce is liable as a contributory infringer of the '287 patent under 35 U.S.C. § 271(c) by offering to sell, selling and importing into the United States website or web page authoring tools to be especially made or adapted for use in an infringement of the '287 patent. The Accused Instrumentality is a material component for use in practicing the '287 patent, is specifically made and is not a staple article of commerce suitable for substantial non-infringing use.

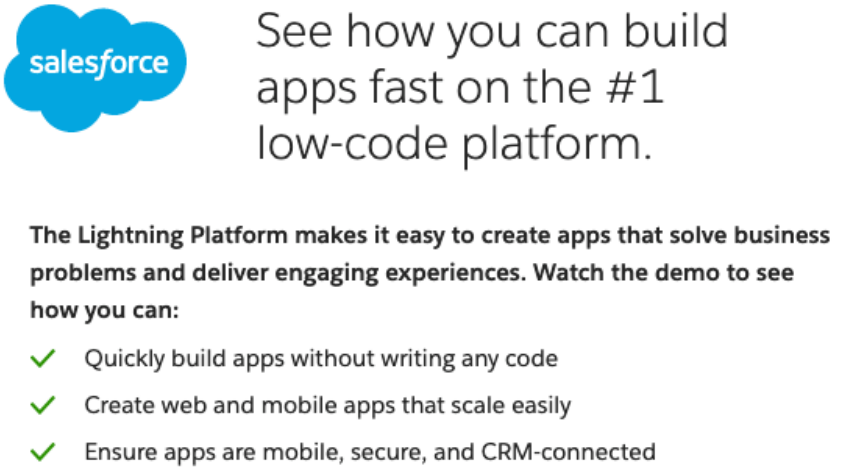
167. Salesforce's infringement has damaged and continues to damage and injure Express Mobile.

COUNT V - INFRINGEMENT OF U.S. PATENT NO. 9,063,755

168. Plaintiff incorporates by reference the allegations contained in paragraphs 1 to 167 above.

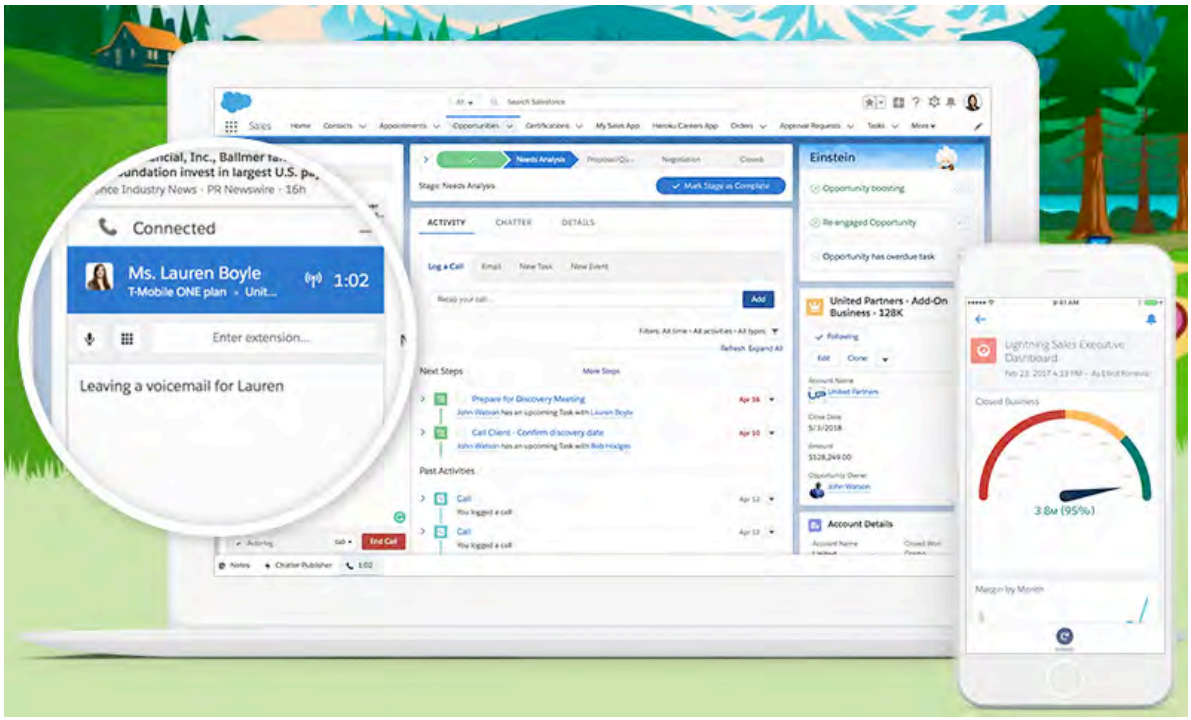
169. Salesforce has manufactured, sold, offered to sell, used and/or provided and continues to manufacture, sell, offer for sale, use and/or provide the Lightning App Builder which is a system for generating code to provide content on a display of a device (the "Accused Instrumentality") which infringes, either literally or under the doctrine of equivalents, one or more claims of the '755 patent in violation of 35 U.S.C. § 271(a).

170. The Accused Instrumentality is a website and app building tool that allows users to "quickly build apps without writing any code" and "create web and mobile apps."

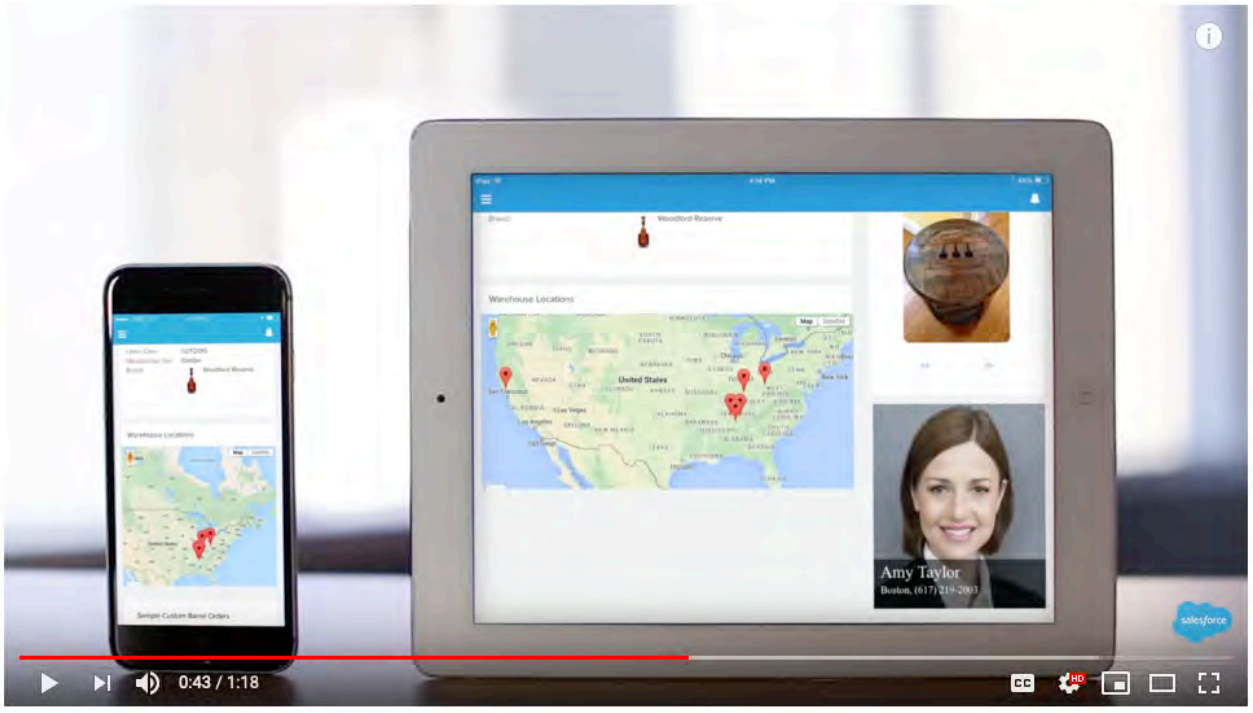


Source: https://www.salesforce.com/form/demo/platform-lightning-demo-sem/

171. The Accused Instrumentality stores items in computer memory in order to serve web pages and apps to devices.

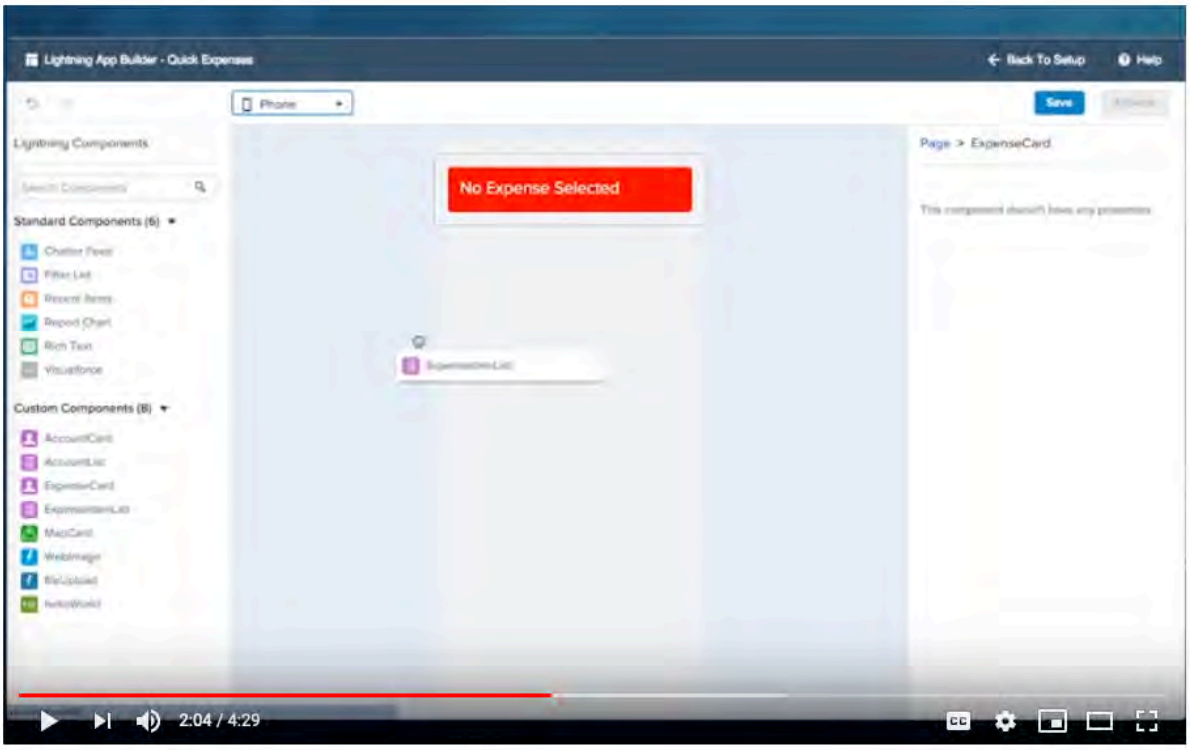


Source: Salesforce Lightning: The Future of Sales and CRM – Salesforce.com https://www.salesforce.com/campaign/lightning/#



Source: Salesforce Platform: Lightning App Builder Demo. https://www.youtube.com/watch?v=PKW4b_Hk9Hw 0:43/1:18

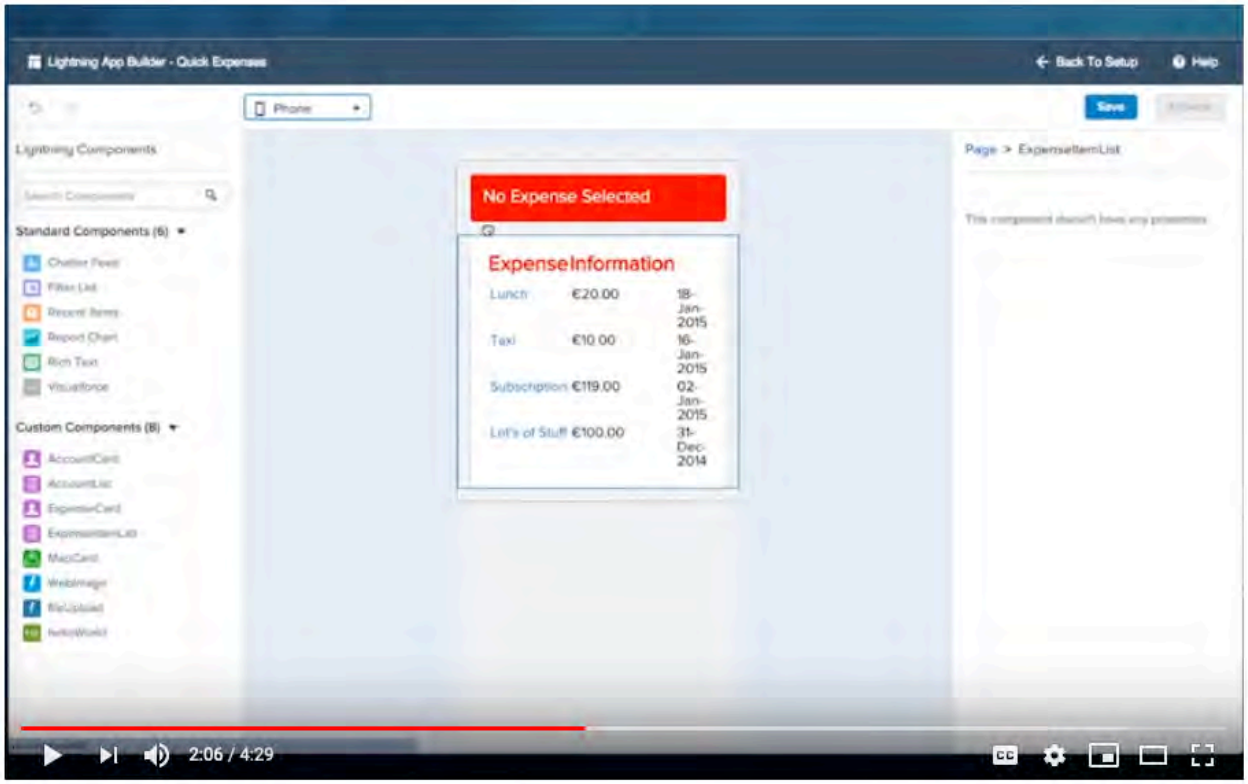
172. The Accused Instrumentality contains symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service, where each symbolic name has an associated data format class type corresponding to a subclass of User Interface (UI) objects that support the data format type of the symbolic name, and where each symbolic name has a preferred UI object. The Accused Instrumentality contains symbolic names as the names of its components. For example, the symbolic name "ExpenseItemList" is seen below.



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 2:04/4:29

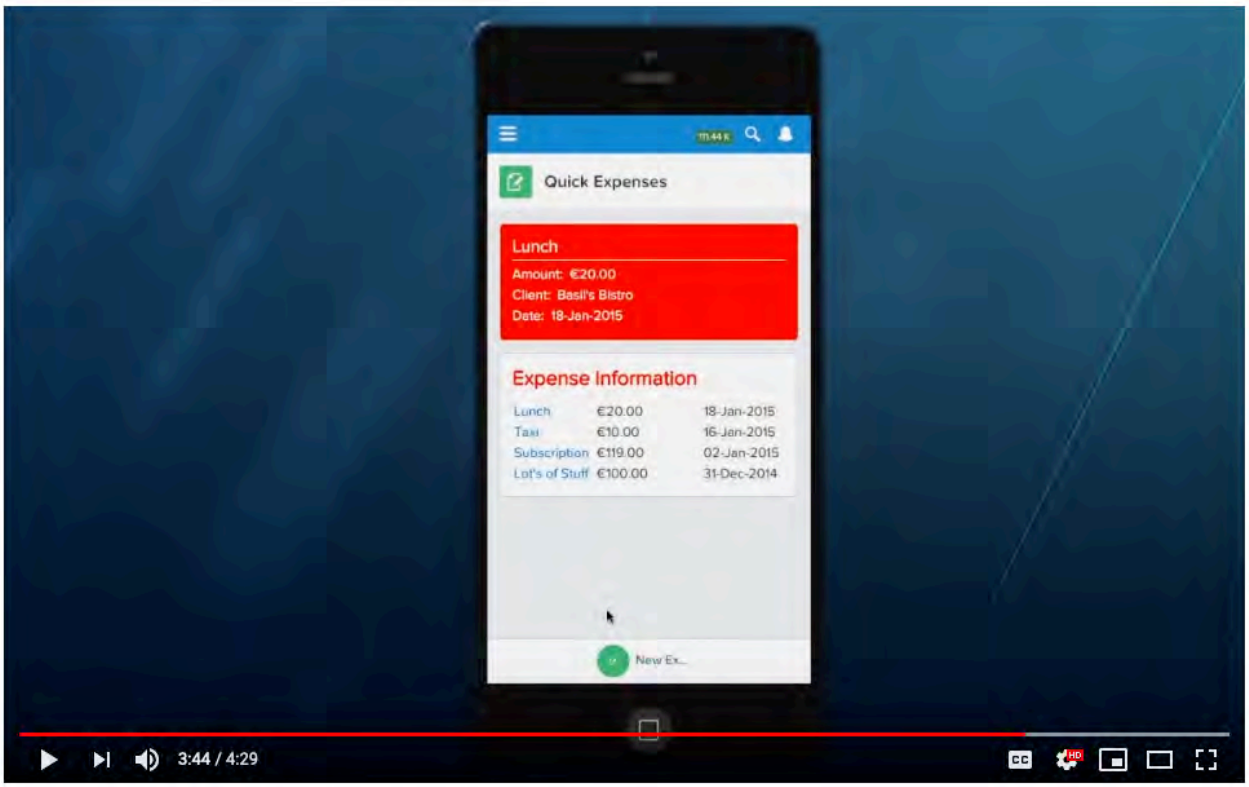
173. The "ExpenseItemList" symbolic name corresponds to the ExpenseItemList web component.



Lightning Intro Screencasts: Lightning App Builder

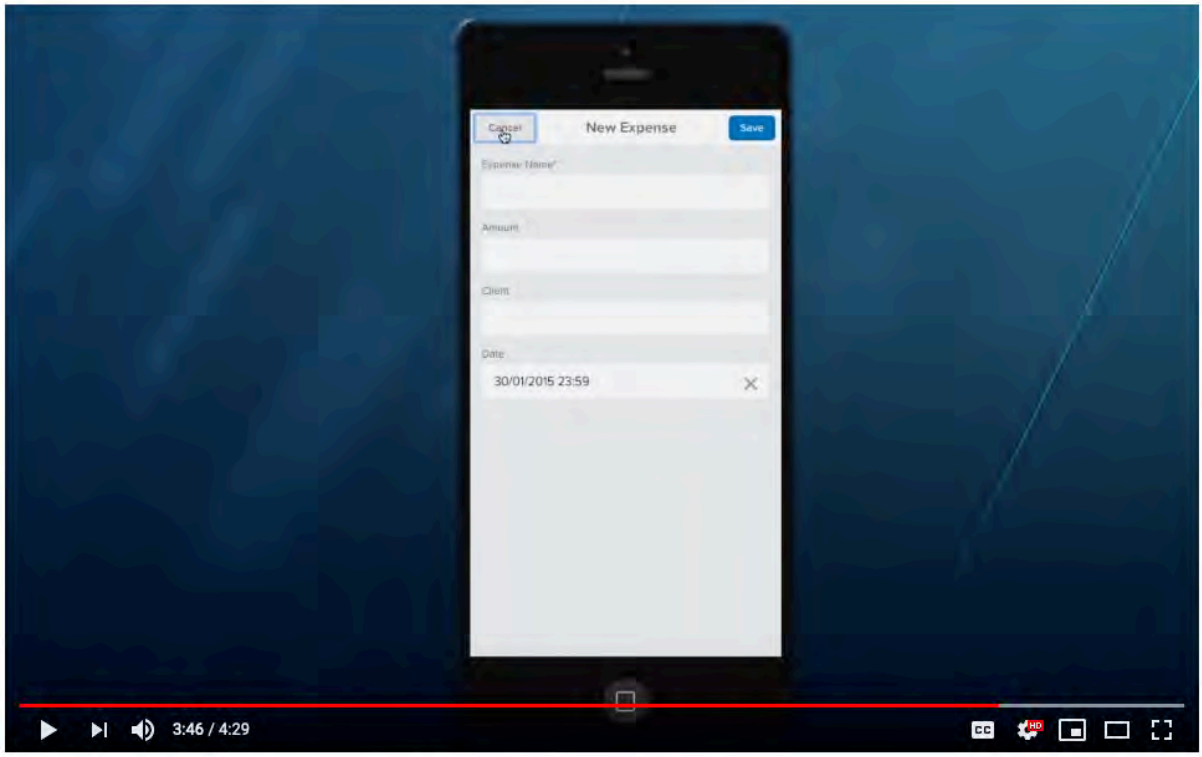
Source: https://youtu.be/XMK5Huuviow 2:06/4:29

174. The "ExpenseItemList" web component widget is related to a set of inputs and outputs of a web service obtainable over a network, as seen by the data populated in the widget, such as the first line item, "Lunch, €20.00, 18-Jan-2015." Users of the app can add additional Expense Items as inputs, and they will be reflected in the Expense Item List as outputs.



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 3:44/4:29



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 3:46/4:29

175. These web components require either a web browser or the Salesforce mobile app to run. Salesforce publishes a list of supported browsers for the Accused Instrumentality.

	MICROSOFT® INTERNET EXPLORER®	MICROSOFT® EDGE	GOOGLE CHROME™	MOZILLA® FIREFOX®	APPLE® SAFARI®
Lightning Experience	IE 11 (EOL December 31, 2020)	Latest	Latest	Latest	12.x+
Lightning Communities	IE 11	Latest	Latest	Latest	12.x+
Special setup considerations?	No	No	No	No	No
Limitations?	Yes	Yes	No	Yes	Yes

Source: https://help.salesforce.com/articleView?id=getstart_browsers_sfx.htm

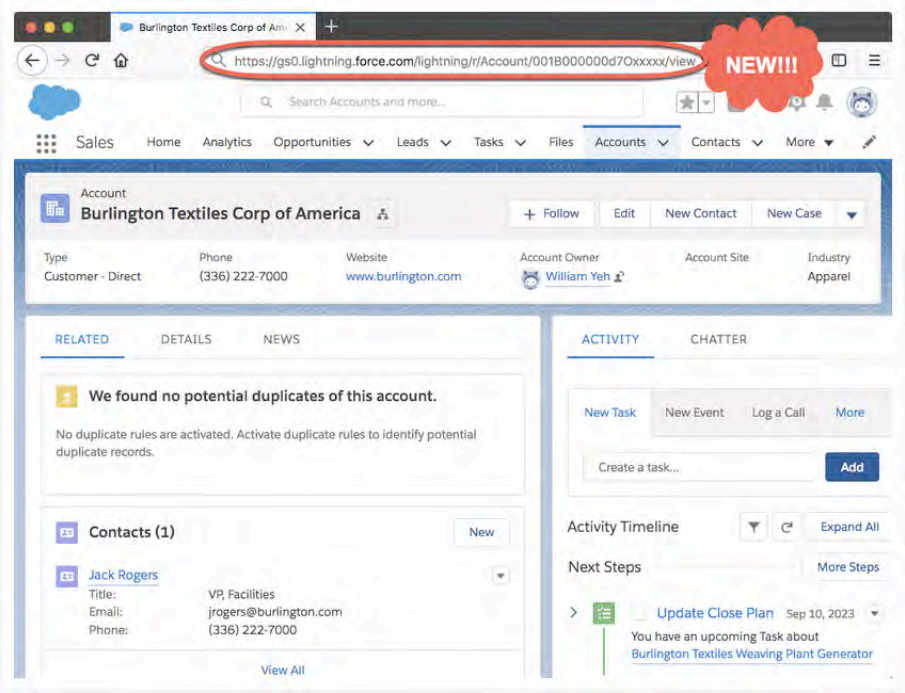
176. The address of the web service in the Accused Instrumentality is embedded in the Component that is dragged from the AppStore and seamlessly placed in the Accused Instrumentality's page in the app builder.

Your custom Lightning components don't automatically work on Lightning pages or in the Lightning App Builder. To make a custom component usable in both, you need to:

1. Configure the component and its component bundle so that they're compatible with the Lightning App Builder and Lightning pages.
2. Deploy My Domain in your org. When you deploy My Domain, references and links to Lightning resources are in the format `https://<myDomain>.lightning.force.com`.

https://help.salesforce.com/articleView?id=lightning_page_components_custom.htm&type=5

177. The Accused Instrumentality stores an address of the web service and stores it in computer memory. For example, Lightning web pages such as the "Burlington Textiles Corp of America" served to users contain addresses. These addresses are stored in computer memory on at least Salesforce's servers. On information and belief, the mobile app also contains the address of the web service.

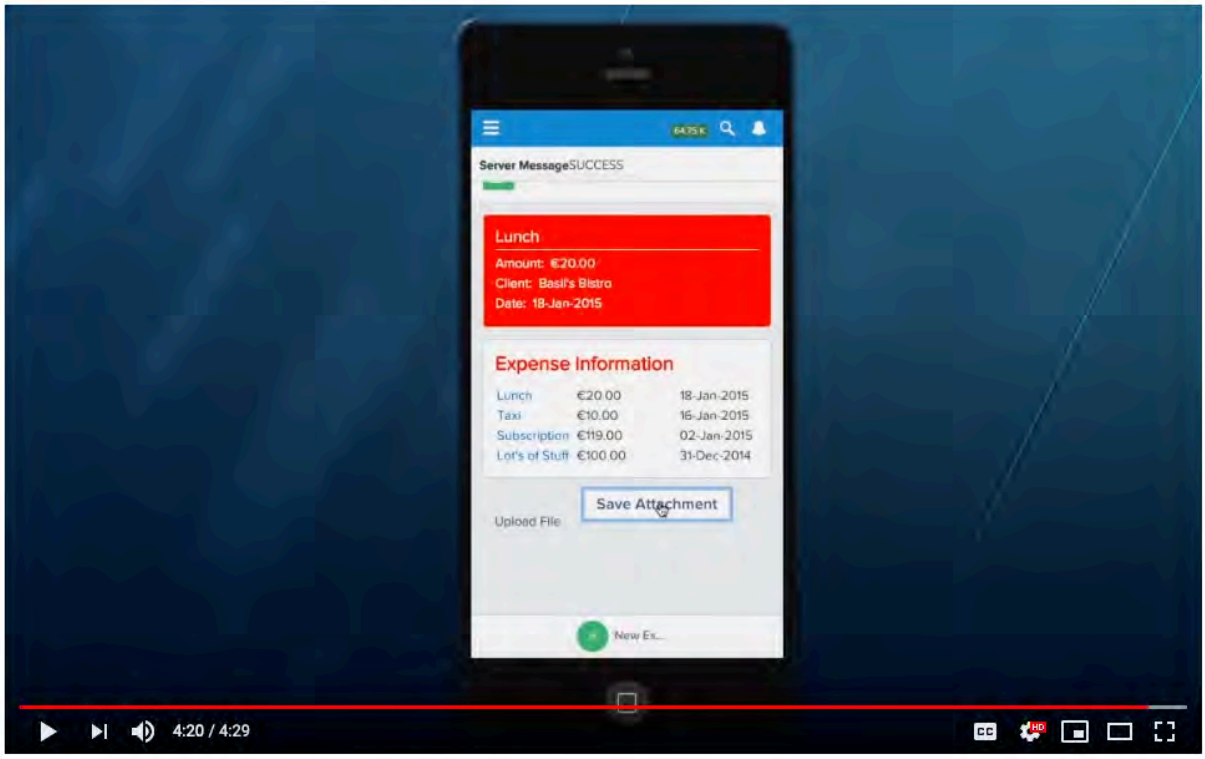


Source: https://developer.salesforce.com/blogs/2018/01/heres-need-know-new-url-format-lightning.html

Example of new URLs
Original format: Account Home URL
https://<lightning.domain.com>/one/one.app/#/sObject/Account/home
New format: Account Home URL
https://<lightning.domain.com>/lightning/o/Account/home

Original format: Account Record View
https://<lightning.domain.com>/one/one.app#/sObject/<recordID>/view
New format: Account Record View
https://<lightning.domain.com>/lightning/r/Account/<recordID>/view

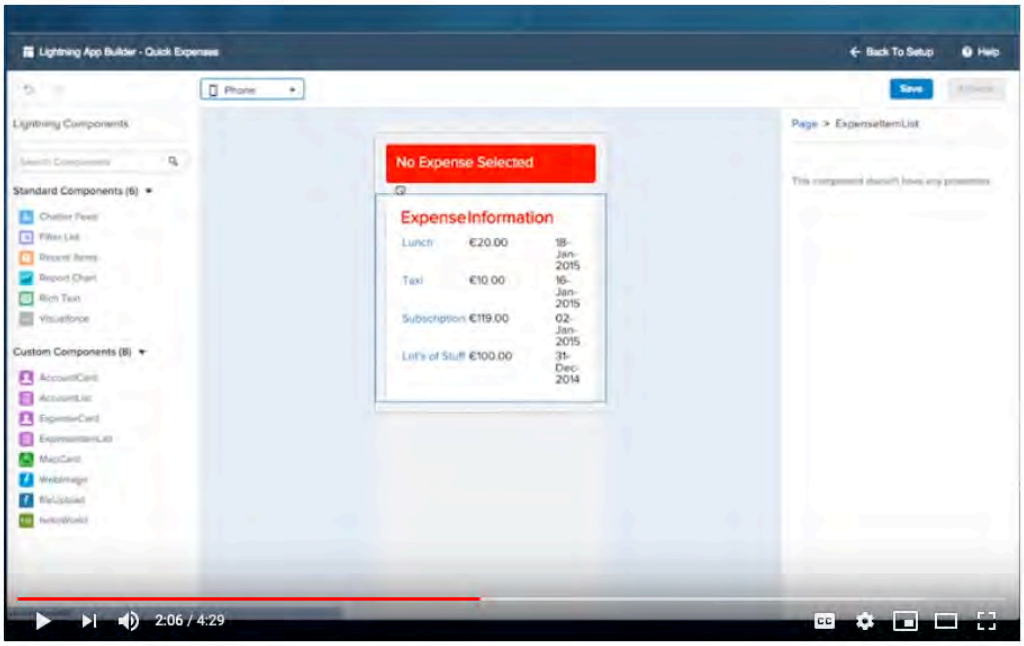
Source: https://developer.salesforce.com/blogs/2018/01/heres-need-know-new-url-format-lightning.html



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 4:20/4:29

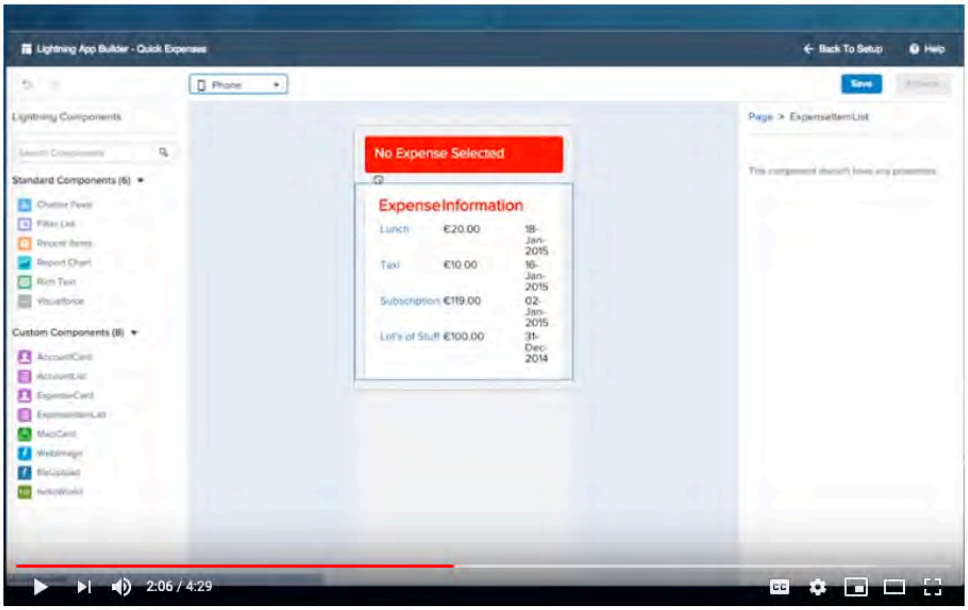
178. The Accused Instrumentality contains an authoring tool configured to define a UI object for presentation on the display. The defined UI object corresponds to the web component included in the computer memory selected from a group consisting of an input of the web service and an output of the web service. For example, the Lightning App Builder tool configures an ExpenseItemList UI object for presentation on the display, titled Expense Information (hereafter "Expense Information").



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 2:06/4:29

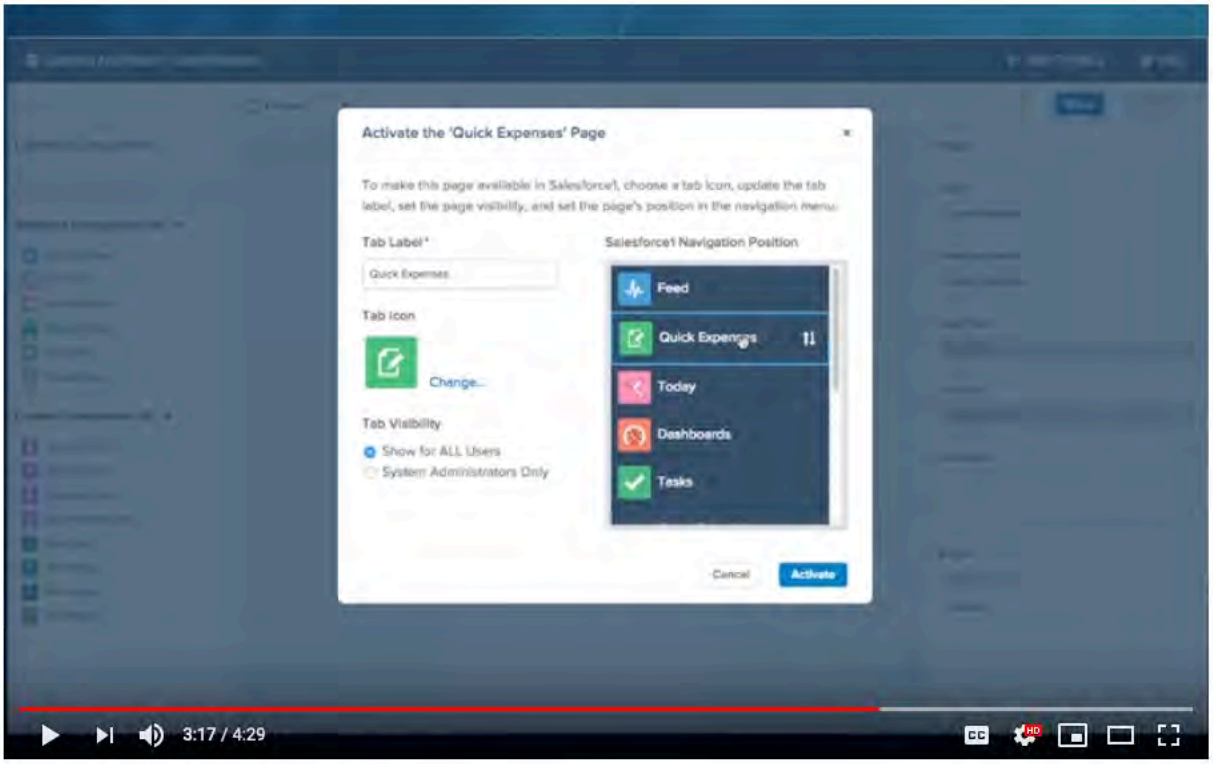
179. The Accused Instrumentality accesses the computer memory to select the symbolic name corresponding to the web component of the defined UI object (e.g., ExpenseItemList) and associates the selected symbolic name with the defined UI object.



Lightning Intro Screencasts: Lightning App Builder

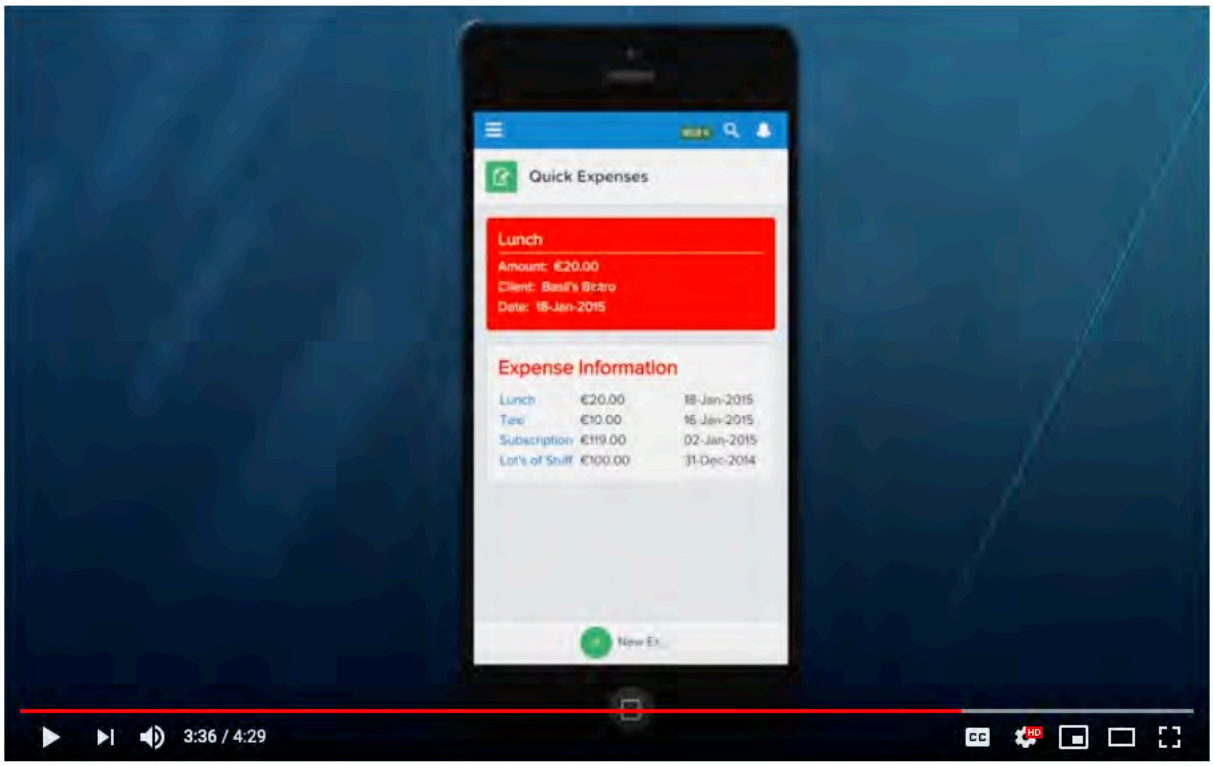
Source: https://youtu.be/XMK5Huuviow 2:06/4:29

180. The Accused Instrumentality produces an application which consists of device-independent code and further includes the selected symbolic name of the defined UI object. The tool also produces a player consisting of device-dependent code, or code that is specific to the operating system, programming language, or platform of the client device. For example, once the user clicks "save" and "activate" on the authoring tool, the application is saved and built. The authoring tool builds and delivers the application code that uses information stored in the database that is independent of the device that it is displayed on.



Lightning Intro Screencasts: Lightning App Builder

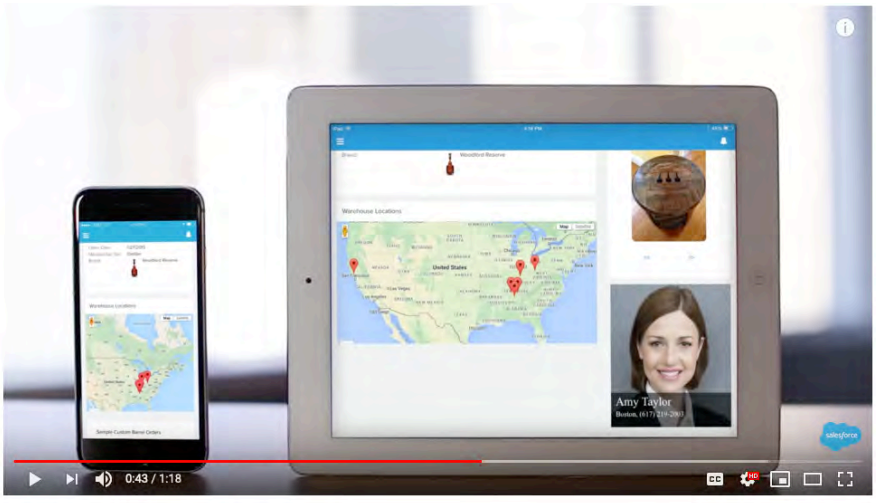
Source:https://www.youtube.com/watch?v=XMK5Huuviow 3:17/4:29



Lightning Intro Screencasts: Lightning App Builder

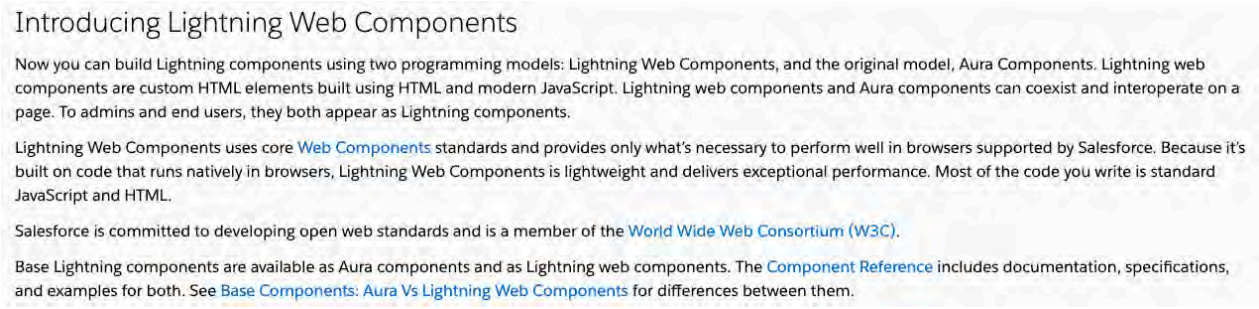
Source: https://www.youtube.com/watch?v=XMK5Huuviow 3:36/4:29

181. Apps created with the Accused Instrumentality are available on both desktop and mobile devices. In both the mobile app and desktop experience, both the application and the player are provided to a user's device and executed on the device.



Salesforce Platform: Lightning App Builder Demo

Source: https://www.youtube.com/watch?v=PKW4b_Hk9Hw 0:43/1:18



Introducing Lightning Web Components

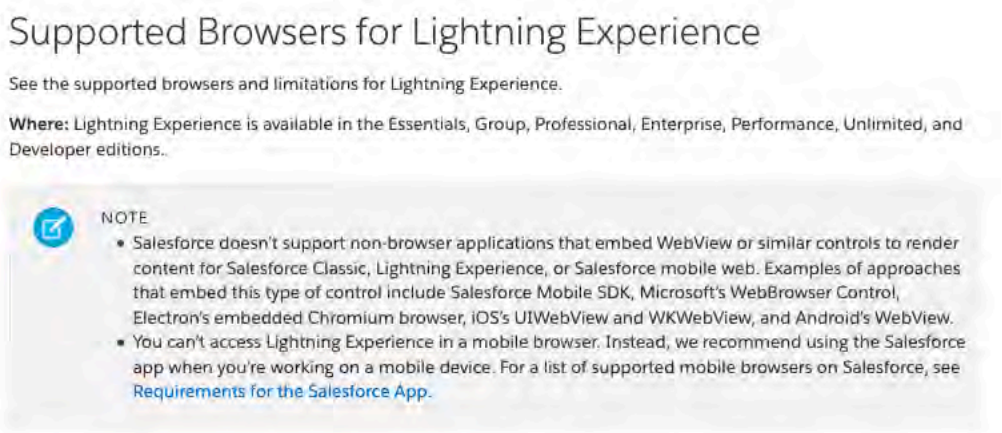
Now you can build Lightning components using two programming models: Lightning Web Components, and the original model, Aura Components. Lightning web components are custom HTML elements built using HTML and modern JavaScript. Lightning web components and Aura components can coexist and interoperate on a page. To admins and end users, they both appear as Lightning components.

Lightning Web Components uses core Web Components standards and provides only what's necessary to perform well in browsers supported by Salesforce. Because it's built on code that runs natively in browsers, Lightning Web Components is lightweight and delivers exceptional performance. Most of the code you write is standard JavaScript and HTML.

Salesforce is committed to developing open web standards and is a member of the World Wide Web Consortium (W3C).

Base Lightning components are available as Aura components and as Lightning web components. The Component Reference includes documentation, specifications, and examples for both. See Base Components: Aura Vs Lightning Web Components for differences between them.

Source:
https://developer.salesforce.com/docs/component-library/documentation/lwc



Supported Browsers for Lightning Experience

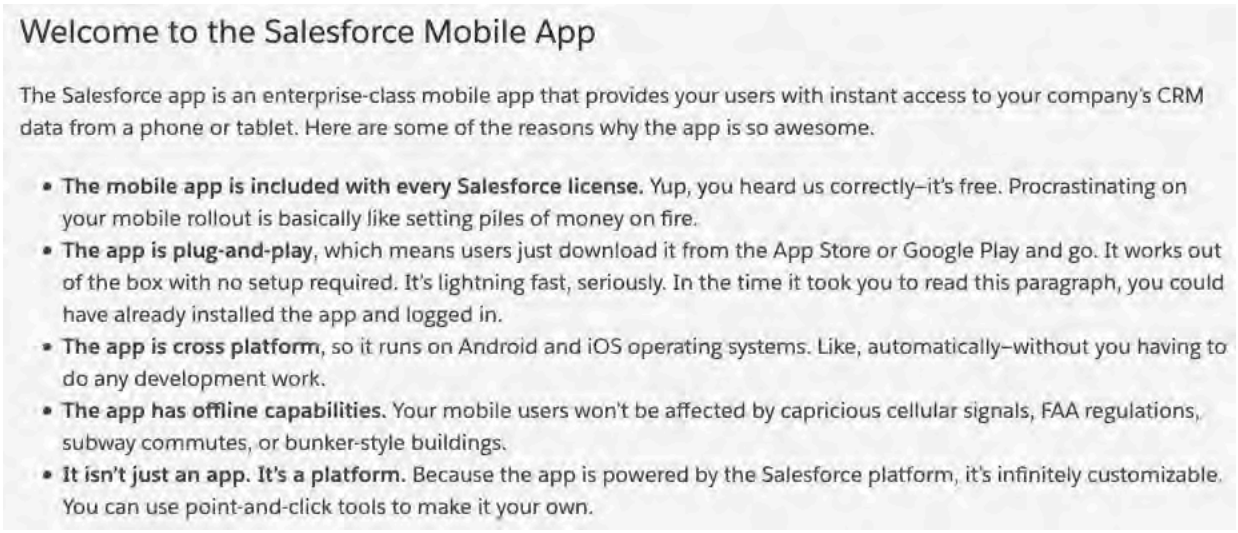
See the supported browsers and limitations for Lightning Experience.

Where: Lightning Experience is available in the Essentials, Group, Professional, Enterprise, Performance, Unlimited, and Developer editions.

NOTE

- Salesforce doesn't support non-browser applications that embed WebView or similar controls to render content for Salesforce Classic, Lightning Experience, or Salesforce mobile web. Examples of approaches that embed this type of control include Salesforce Mobile SDK, Microsoft's WebBrowser Control, Electron's embedded Chromium browser, iOS's UIWebView and WKWebView, and Android's WebView.
- You can't access Lightning Experience in a mobile browser. Instead, we recommend using the Salesforce app when you're working on a mobile device. For a list of supported mobile browsers on Salesforce, see Requirements for the Salesforce App.

Source: https://help.salesforce.com/articleView?id=getstart_browsers_sfx.htm



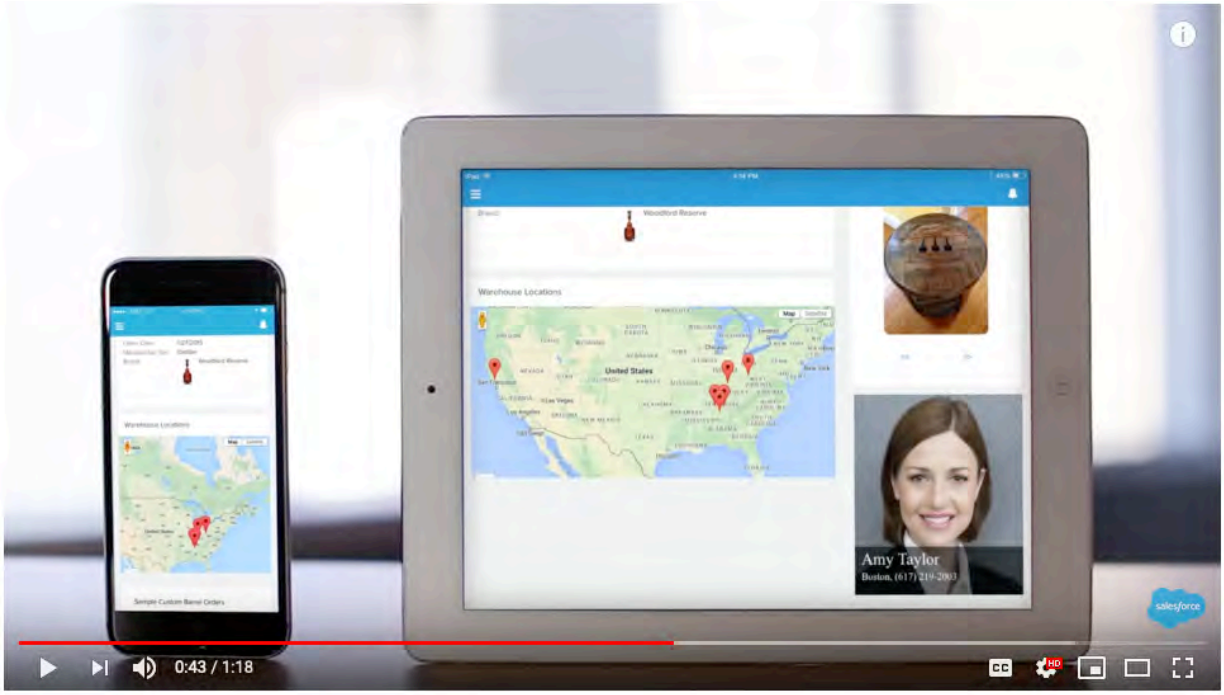
Welcome to the Salesforce Mobile App

The Salesforce app is an enterprise-class mobile app that provides your users with instant access to your company's CRM data from a phone or tablet. Here are some of the reasons why the app is so awesome.

- **The mobile app is included with every Salesforce license.** Yup, you heard us correctly–it's free. Procrastinating on your mobile rollout is basically like setting piles of money on fire.
- **The app is plug-and-play**, which means users just download it from the App Store or Google Play and go. It works out of the box with no setup required. It's lightning fast, seriously. In the time it took you to read this paragraph, you could have already installed the app and logged in.
- **The app is cross platform**, so it runs on Android and iOS operating systems. Like, automatically–without you having to do any development work.
- **The app has offline capabilities.** Your mobile users won't be affected by capricious cellular signals, FAA regulations, subway commutes, or bunker-style buildings.
- **It isn't just an app. It's a platform.** Because the app is powered by the Salesforce platform, it's infinitely customizable. You can use point-and-click tools to make it your own.

Source:
https://trailhead.salesforce.com/content/learn/modules/salesforce1_mobile_app/salesforce1_mobile_app_intro

182. The application and player are provided to the device and executed on the device. Apps created with the Accused Instrumentality are available on both desktop and mobile devices. In both the mobile app and desktop experience, both the application and player are provided to a user's device and executed on the device.

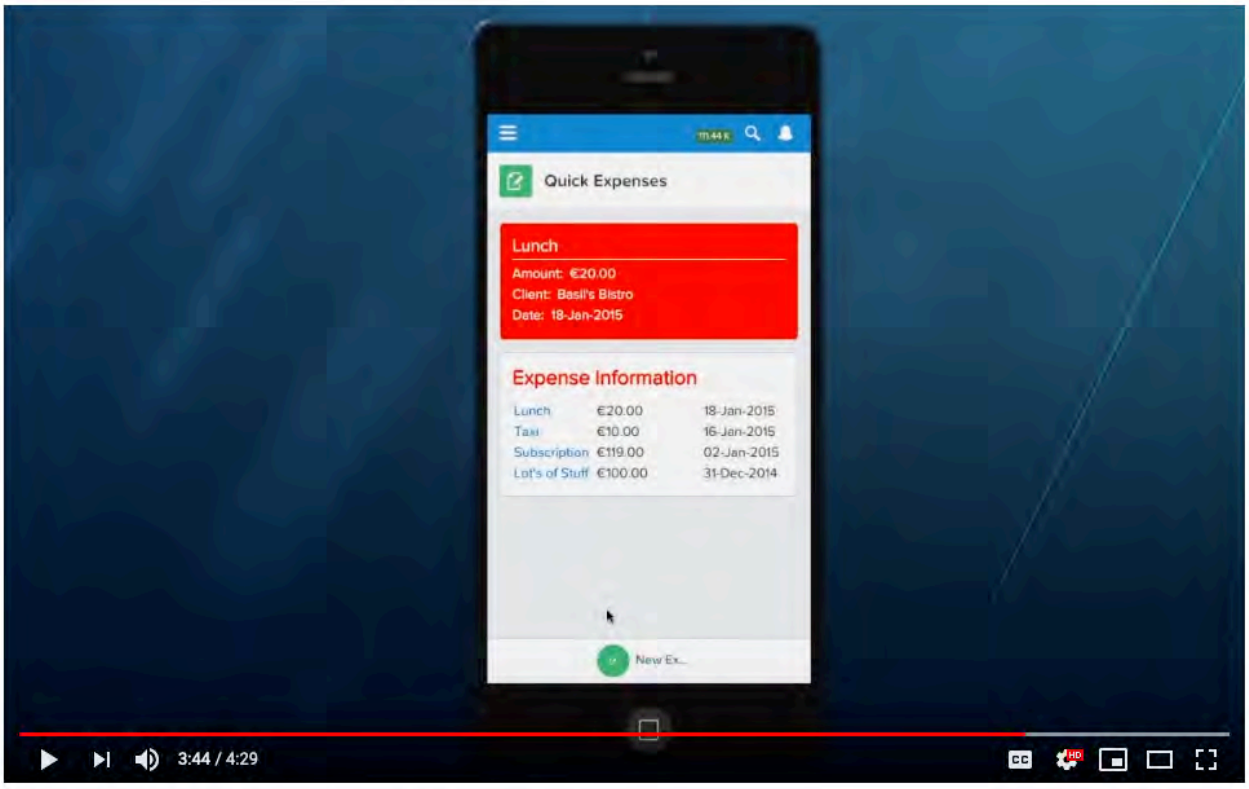


Salesforce Platform: Lightning App Builder Demo

Source: https://www.youtube.com/watch?v=PKW4b_Hk9Hw 0:43/1:18

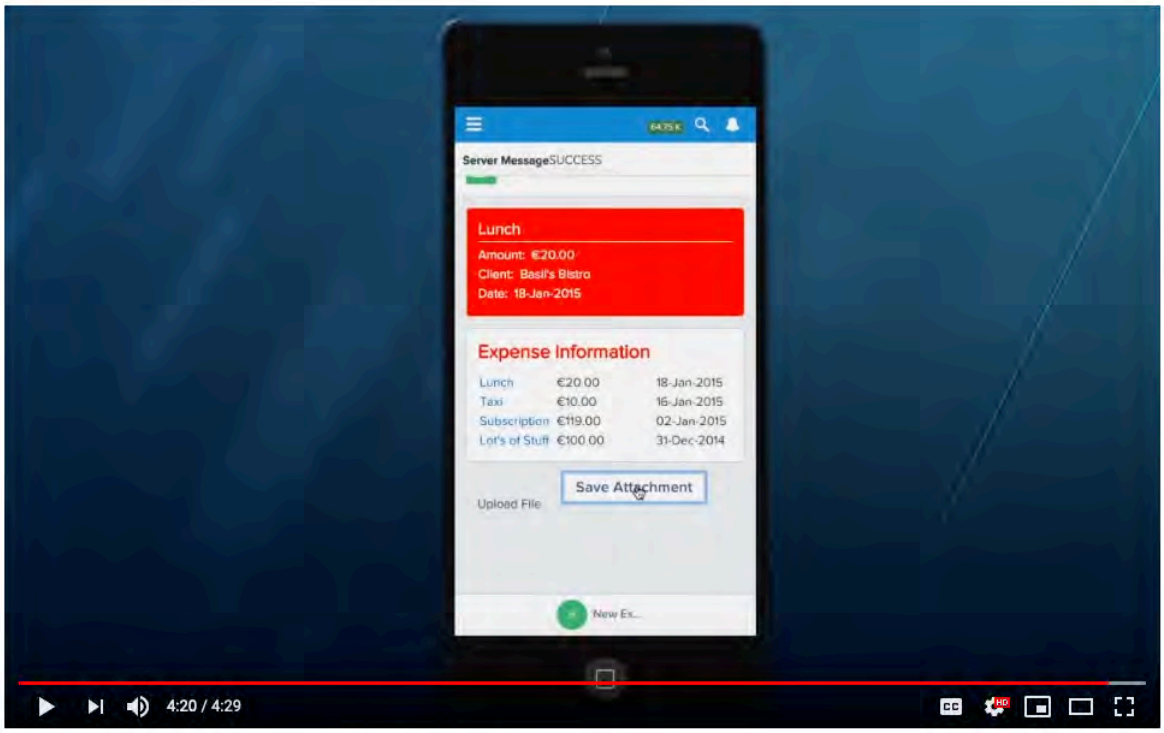
183. The application and player are provided to and executed on the device and the user provides one or more input values associated with an input symbolic name to an input of a defined UI object. The device provides the user provided one or more input values and corresponding input symbolic name to the web service. The web service utilizes the input symbolic name and user provided one or more input values for generating one or more output values having an associated symbolic name. The player receives the output symbolic name and corresponding one or more output values and provides instructions for a display of the device to present an output value in the defined UI object. For example, users of the app can add additional Expense Items. The web service will use the user's input values for generating output values and return data to the player for generating a confirmation "SUCCESS" message after the user has uploaded a receipt.

The device provides the input new expenses to the Salesforce web service for generating one or more output values having an associated symbolic name. The player receives the output symbolic name and corresponding output value(s) and provides instructions for a display of the device to present an output value in the defined UI object.



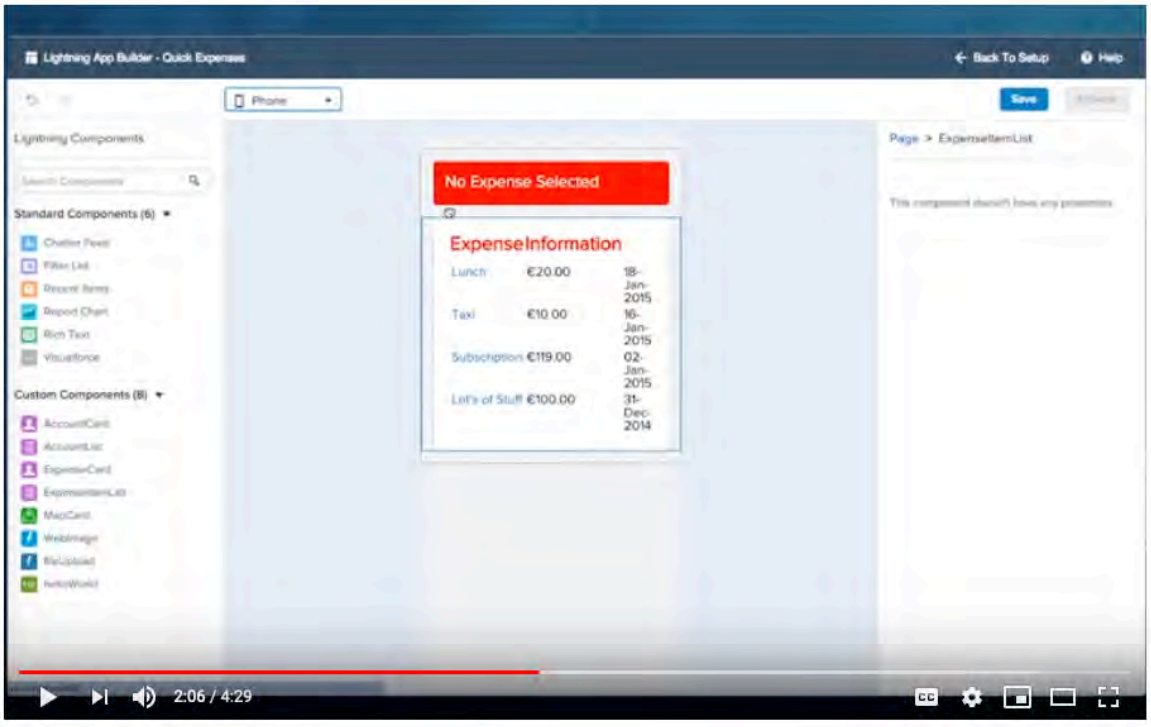
Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 3:44/4:29



Lightning Intro Screencasts: Lightning App Builder

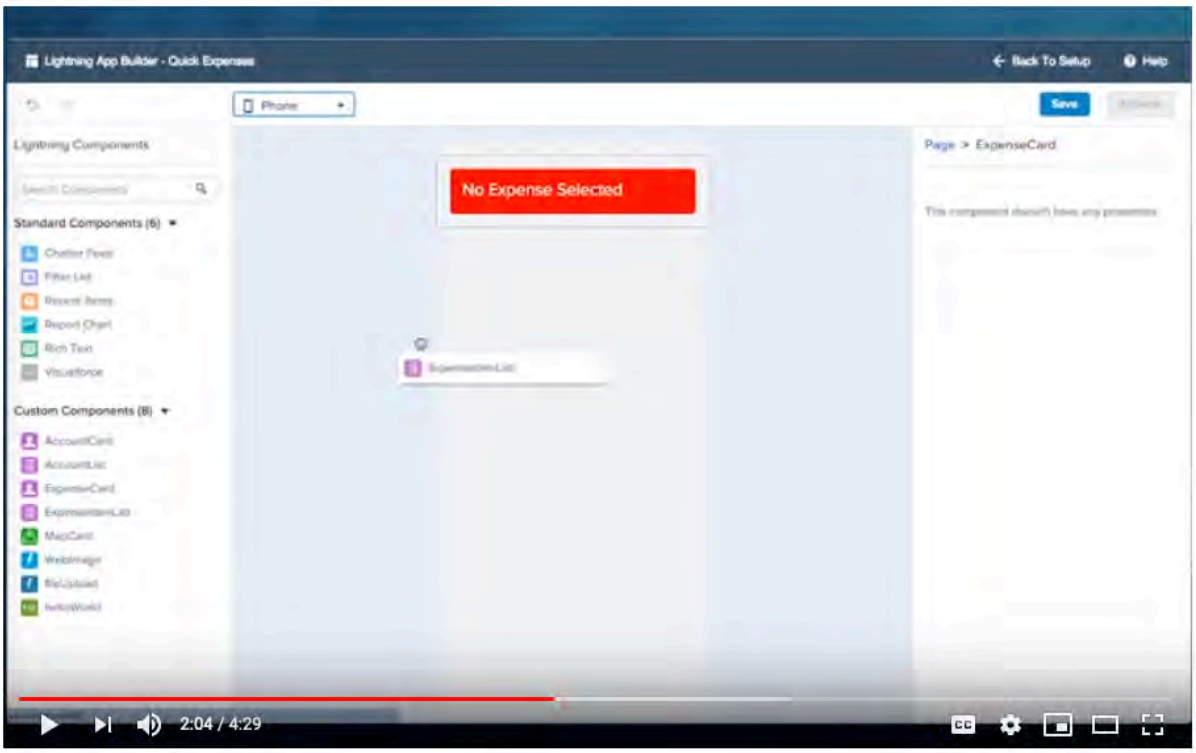
Source: https://youtu.be/XMK5Huuviow 4:20/4:29



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 2:06/4:29

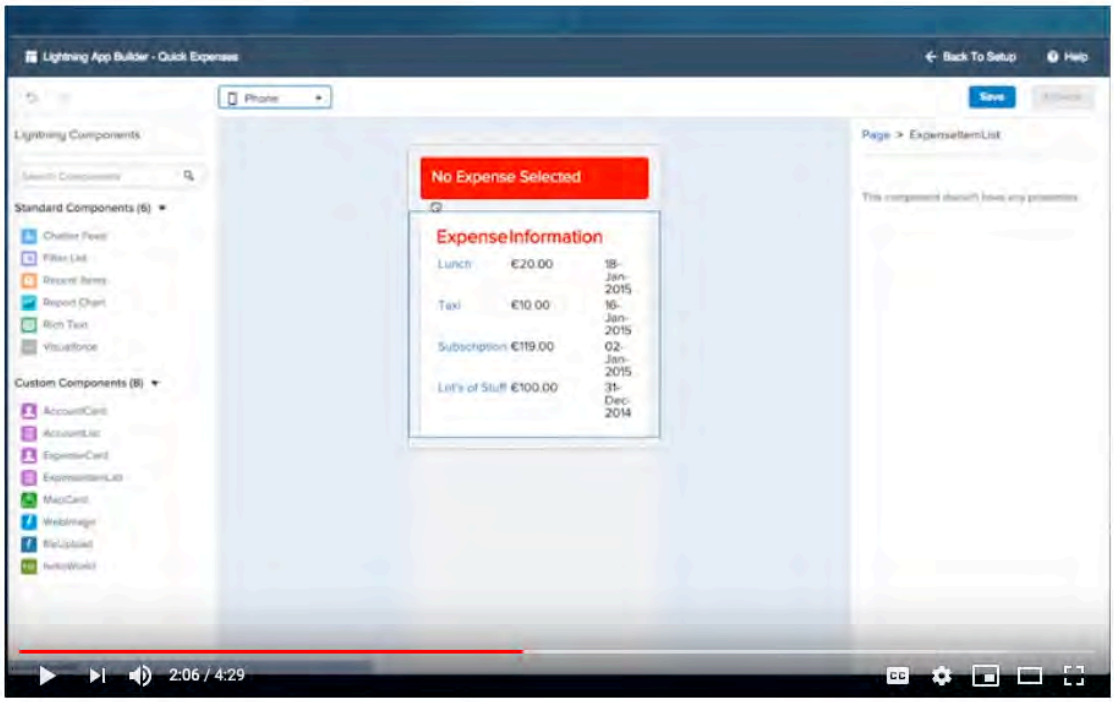
184. The defined Expense Information UI object is selected by the user. Since the UI object is a Custom Component, it contains information selected by the user.



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 2:04/4:29

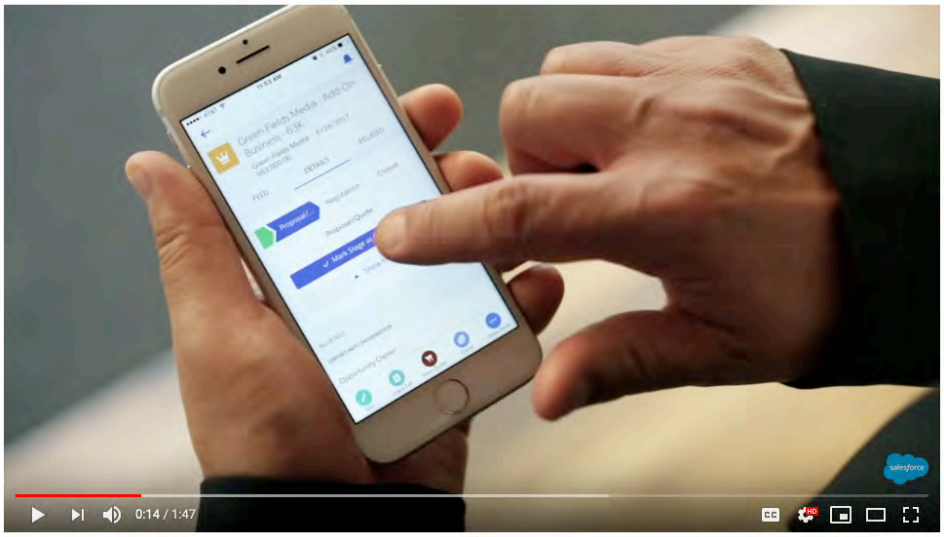
185. The authoring tool in the Accused Instrumentality is configured to associate the selected symbolic name with the defined UI object, where the selected symbolic name is only available to UI objects that support the defined data format associated with that symbolic name. For example, the authoring tool associates the "ExpenseItemList" symbolic name with the defined UI object, where the symbolic name is only available to UI objects that support the data of the ExpenseItemList.



Lightning Intro Screencasts: Lightning App Builder

Source: https://youtu.be/XMK5Huuviow 2:06/4:29

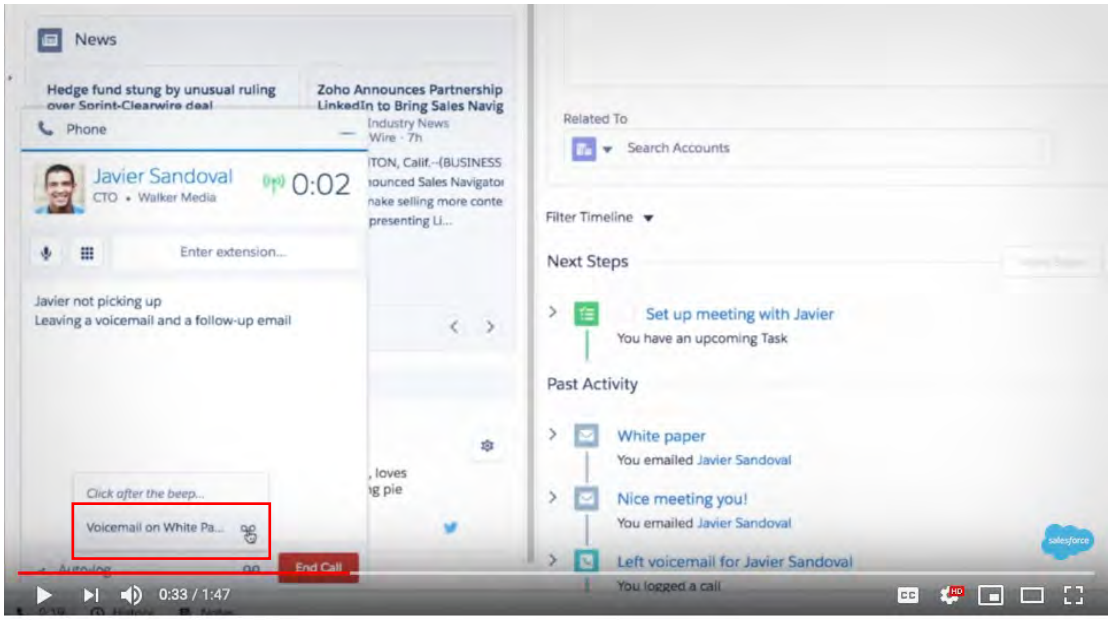
186. As another example, the web service will utilize the user's input values for generating output values and return data to the player for generating a confirmation "Stage changed successfully" message.



Introduction to Salesforce Lightning

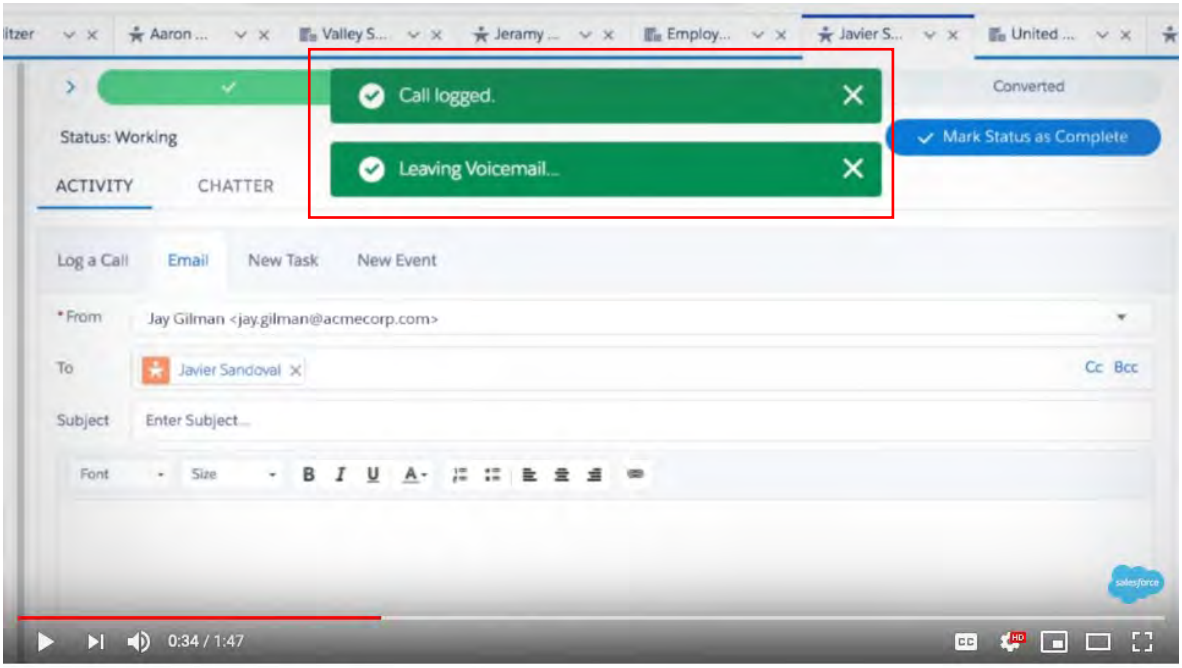
Source: https://www.youtube.com/watch?v=s_sHeac3gfo 0:14/1:47

187. As another example, the web service will use the user's input values for generating output values and return data to the player to verify that a call placed through the system was logged with output symbolic name "Call logged."



Introduction to Salesforce Lightning

Source: https://www.youtube.com/watch?v=s_sHeac3gfo 0:34/1:47



Introduction to Salesforce Lightning

Source: https://www.youtube.com/watch?v=s_sHeac3gfo 0:34/1:47

188. Salesforce was made aware of the '755 patent and its infringement thereof at least as early as November 30, 2020 when Express Mobile provided notice of Salesforce's infringement of the '755 patent to Marc Benioff, Chair and Chief Executive Officer and Amy Weaver, President and Chief Legal Officer of Salesforce. Since at least the time Salesforce received notice, Salesforce has induced others to infringe at least one claim of the '755 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Salesforce's clients, customers, and end users, whose use of the Accused Instrumentality constitutes direct infringement of at least one claim of the '755 patent. In particular, Salesforce's actions that aid and abet others such as customers and end users to infringe include advertising and distributing the Accused Instrumentality and providing instruction materials, training, and services regarding the Accused Instrumentality. *See e.g.*, www.salesforce.com, help.salesforce.com, https://trailhead.salesforce.com, including all related domains and subdomains. Salesforce has engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Salesforce has had actual knowledge of the '755 patent and knowledge that its acts were inducing infringement of the '755 patent since at least the date Salesforce received notice that such activities infringed the '755 patent.

189. Salesforce is liable as a contributory infringer of the '755 patent under 35 U.S.C. § 271(c) by offering to sell, selling and importing into the United States website or web page authoring tools to be especially made or adapted for use in an infringement of the '755 patent. The Accused Instrumentality is a material component for use in practicing the '755 patent, is specifically made and is not a staple article of commerce suitable for substantial non-infringing use.

190. Salesforce's infringement has damaged and continues to damage and injure Express Mobile.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff requests that the Court enter judgment for Plaintiff and against Defendant as follows:

191. That U.S. Patent No. 6,546,397 be judged valid, enforceable, and infringed by Defendant;

192. That U.S. Patent No. 7,594,168 be judged valid, enforceable, and infringed by Defendant

193. That U.S. Patent No. 9,928,044 be judged valid, enforceable, and infringed by Defendant;

194. That U.S. Patent No. 9,471,287 be judged valid, enforceable, and infringed by Defendant;

195. That U.S. Patent No. 9,063,755 be judged valid, enforceable, and infringed by Defendant;

196. That Plaintiff be awarded judgment against Defendant for damages together with interests and costs fixed by the Court including an accounting of all infringements and/or damages not presented at trial;

197. That the Court declare this an exceptional case and award Plaintiff its attorneys' fees, as provided by 35 U.S.C. § 285; and

198. That Plaintiff be awarded such other and further relief as this Court may deem just and proper.

JURY DEMAND

Plaintiff respectfully requests a jury trial on all issues so triable.

Dated: December 1, 2020

Respectfully submitted,

By: */s/ Robert Kramer*
Robert Kramer

FEINBERG DAY KRAMER ALBERTI LIM TONKOVICH & BELLOLI LLP

Attorneys for Express Mobile, Inc.