Seth W. Wiener (SBN 203747)
seth@sethwienerlaw.com
LAW OFFICES OF SETH WIENER
9107 Wilshire Blvd., Suite 450
Beverly Hills, CA 90210
Telephone: (925) 487-5607

Jeffrey Francis Craft (SBN 147186)
jcraft@devlinlawfirm.com
DEVLIN LAW FIRM LLC
1731 Fox Springs Circle,
Newbury Park, CA 91320

Timothy Devlin (*pro hac vice* to be filed)
tdevlin@devlinlawfirm.com
Robert Kiddie (*pro hac vice* to be filed)
rkiddie@devlinlawfirm.com
DEVLIN LAW FIRM LLC
1526 Gilpin Avenue
Wilmington, DE 19806
Telephone: (302) 449-9010
Facsimile: (302) 353-4251

Attorneys for Express Mobile, Inc.

UNITED STATES DISTRICT COURT

CENTRAL DISTRICT OF CALIFORNIA

WESTERN DIVISION

EXPRESS MOBILE, INC., Plaintiff, vs. POSSIBLE WORLDWIDE, LLC Defendant.	Case No.: 2:19-cv-5110 **COMPLAINT FOR PATENT INFRINGEMENT** DEMAND FOR JURY TRIAL

Plaintiff Express Mobile, Inc. ("Express Mobile" or "Plaintiff"), for its Complaint against Defendant Possible Worldwide, LLC ("Possible" or "Defendant") alleges the following:

NATURE OF THE ACTION

1. This is an action for patent infringement arising under the Patent Laws of the United States, 35 U.S.C. § 1 *et seq*.

THE PARTIES

2. Plaintiff is a corporation organized under the laws of the State of Delaware with a place of business at 3415 Custer Rd. Suite 104, Plano, TX 75023.

3. Upon information and belief, Possible is a limited liability company organized and existing under the laws of California, with a place of business at 12180 Millennium Drive, Suite 420, Playa Vista, CA 90094 and can be served through its registered agent, Corporate Creations Network Inc., 4640 Admiralty Way, 5th Floor, Marina del Rey, CA 90292.

4. Upon information and belief, Possible sells and offers to sell products and services throughout the United States, including in this judicial district, and introduces products and services into the stream of commerce and that incorporate infringing technology knowing that they would be sold in this judicial district and elsewhere in the United States.

JURISDICTION AND VENUE

5. This is an action for patent infringement arising under the Patent Laws of the United States, Title 35 of the United States Code.

6. This Court has subject matter jurisdiction under 28 U.S.C. §§ 1331 and 1338(a).

7. Venue is proper in this judicial district under 28 U.S.C. §1400(b). On information and belief, Defendant incorporated in California, has committed acts of infringement in this District, and has a regular and established place of business within this District.

8. On information and belief, Defendant is subject to this Court's general and specific personal jurisdiction because Defendant has sufficient minimum contacts within the State of California and this District, pursuant to due process and/or the California Long Arm Statute because Defendant purposefully availed itself of the privileges of conducting business in the State of California and in this District, because Defendant regularly conducts and solicits business within the State of California and within this District, and because Plaintiff's causes of action arise directly from each of Defendant's business contacts and other activities in the State of California and this District.

COUNT I – INFRINGEMENT OF U.S. Patent No. 6,546,397

9. The allegations set forth in the foregoing paragraphs 1 through 8 are incorporated into this First Claim for Relief.

10. On April 8, 2003, U.S. Patent No. 6,546,397 ("the '397 patent"), entitled "*Browser Based Web Site Generation Tool and Run Time Engine*," was duly and legally issued by the United States Patent and Trademark Office. A true and correct copy of the '397 patent is attached as Exhibit A.

11. The inventions of the '397 patent resolve 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, which features are exclusively implemented utilizing computer technology including a virtual machine.

12. The claims of the '397 patent do not merely recite the performance of some business practice known from the pre-Internet world along with the requirement to perform it on the Internet. Instead, the claims of the '397 patent recite one or more inventive concepts that are rooted in computerized website creation technology, and overcome problems specifically arising in the realm of computerized website creation technologies.

13. The claims of the '397 patent recite an invention that is not merely the routine or conventional use of website creation systems and methods. Instead, the invention describes 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.

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

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

16. In C.A. 2:17-00128, a case filed in the Eastern District of Texas, the defendant in that action, KTree Computer Solutions brought a Motion for Judgment on the Pleadings asserting that the '397 patent, along with U.S. Patent No. 7,594,168 (asserted in Count II below) were invalid as claiming abstract subject matter under 35 U.S.C. § 101. (C.A. 2:17-00128 Dkt. 9.) Subsequent briefing included Plaintiff's Response and related Declarations and Exhibits (C.A. 2:17-00128 Dkt. 17, 22-24), KTree's Reply (C.A. 2:17-00128 Dkt. 25), and Plaintiff's Sur-Reply and related Declarations and Exhibits (C.A. 2:17-00128 Dkt. 26-27). Each of those filings is incorporated by reference into this Complaint.

17. After a consideration of 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." (C.A. 2:17-00128 Dkt. 29 attached hereto as Exhibit B.) Judge Payne's report and recommendation is incorporated by reference

into this Complaint. No objection was filed to the Magistrate Judge's report and recommendation and the decision therefore became final.

18. In C.A. Nos. 3:18-cv-04679 and 3:18-04688, both of which were filed in the Northern District of California, the respective defendant in each of those actions brought a Motion to Dismiss asserting that the '397 patent, along with U.S. Patent No. 7,594,168 (asserted in Count II below) were invalid as claiming abstract subject matter under 35 U.S.C. § 101. The § 101 briefing in each of those cases is incorporated by reference into this Complaint.

19. After consideration of the respective pleadings and oral argument, Judge Richard Seeborg issued orders denying each respective motion to dismiss drawing a comparison between the asserted Express Mobile patents with those patents asserted in *Enfish, LLC v. Microsoft Corp.*, 822 F.3d 1327 (Fed. Cir. 2016). (C.A. 3:18 -04679 Dkt. 45, attached hereto as Exhibit C, and C.A. 3:18-04688 Dkt. 40, attached hereto as Exhibit D.)

20. Plaintiff is the assignee and owner of the right, title and interest in and to the '397 patent, including the right to assert all causes of action arising under said patents and the right to any remedies for infringement of them.

21. Upon information and belief, Defendant has and continues to directly infringe at least claims 1-6, 8-11, 14-15, 17, 20, 24-25, 35, and 37 of the '397 patent by using a browser-based website and/or web page authoring tool 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 (the "Accused Instrumentalities"). The Accused Instrumentalities include but are not limited to the website building tools used and/or provided by Defendant, such as, for example Drupal, Joomla, and/or Shopify. *See, e.g.*, https://www.possible.com/about.

22. On information and belief, Defendant is a for-profit organization with revenues of approximately $600 million U.S.D. per year. Moreover, Defendant, its employees and/or agents utilize the Accused Instrumentalities in the building and/or

hosting of websites for Defendant's customers, leading to direct or indirect revenues and profit. As one example of indirect profit, entities such as Defendant will frequently offer website building and/or hosting services at reduced pricing as an inducement to attract customers, who then purchase additional products or services. On information and belief, without the availability of infringing tools such as the Accused Instrumentalities, Defendant would be at a disadvantage in the marketplace and would generate less revenue overall.

23. In particular, claim 1 of the '397 patent generally recites a method enabling production of websites on and for computers with browsers and virtual machines, by presenting, through a browser, a selectable settings menu describing elements, such setting(s) corresponding to commands to the virtual machine; generating a display in accordance with selected settings; storing information regarding selected settings in a database; generating a website at least in part by retrieving said information; and building web page(s) to generate said website and a run time file, where the run time file uses the stored information to generate virtual machine commands for the display of at least a portion of web page(s).

24. The Accused Instrumentalities infringe claim 1 of the '397 patent through a combination of features which collectively practice each limitation of claim 1. By way of example, modern internet browsers such as Microsoft Internet Explorer, Mozilla's Firefox, Apple Safari, Google Chrome, and Opera include virtual machines within the meaning of the '397 patent. (*See, e.g.*, http://developer.telerik.com/featured/a-guide-to-javascript-engines-for-idiots/; http://dictionary.reference.com/browse/virtual+machine?s=t). The Accused Instrumentalities support the use of the latest versions of Internet Explorer 11 or later, Microsoft Edge, latest–1, Firefox latest, latest–1, Chrome latest, latest–1, Safari latest, latest–1 (Mac OS), Safari Mobile for iPad 2, iPad Mini, iPad with Retina Display (iOS 7 or later), for desktop site, Safari Mobile for iPhone 4 or later; iOS 7 or later, for mobile site, Chrome for mobile latest–1 (Android 4 or later) for mobile site, where

latest–1 means one major version earlier than the latest released version. (*See, e.g.,* https://www.drupal.org/docs/8/system-requirements/browser-; https://docs.joomla.org/Joomla_Browser_Support.) All of these browsers rely on browser engines comprising virtual machines to interpret and execute JavaScript and HTML to render web pages on a computer.

25. By way of further example, the Accused Instrumentalities enable users to produce websites through browsers on users' computers via interaction with an Internet server. For example, in order to add a new page to a user's website, the user logs in and then a server of the Accused Instrumentalities initiates presentation to the user through a browser of a website-builder tool. From the interface—sometimes referred to as a dashboard—of the Accused Instrumentalities, the user can navigate and add elements and element properties commensurate with a new page. A display is generated in accordance with one or more user selected settings substantially contemporaneously with the selection thereof. This is performed, for example, using a visual editing tool through a browser. The WYSIWYG interface for selecting center alignment of an image can also be accessed, and then the user can select various options such as a font and paragraph styles. After the user selects options such as image/text alignment or font and paragraph styles through the WYSIWYG editor, the display immediately updates to reflect the selected option. Furthermore, when images are uploaded by a user, those images are displayed in approximately 0-2 seconds depending on file size and bandwidth.

26. Data is stored in a database, including information corresponding to user selected settings such as, for example, the selections of text color. Other user selections are also stored including, for example, the layout, image filenames, thumbnails, and paragraph margin settings for defining the alignment of an image location. The Accused Instrumentalities build one or more web pages to generate a website from at least a portion of a database and at least one run time file, where at

least one run time file utilizes information stored in said database to generate virtual machine commands for the display of at least a portion of said one or more web pages.

27. At run time, at least some of these files use information stored in the database to generate the HTML for the final rendered HTML page. This HTML represents virtual machine commands for display of the page because it is read and used by the applicable browser's engine, including a virtual machine, in order to render the page. On information and belief, the Accused Instrumentalities further rely on the browser engine's component JavaScript engine to either display a portion of the page directly, or generate HTML to be executed for display by the main layout engine.

28. Additionally, the "PHP code," including the PHP template files, can be viewed in the file directory for the Accused Instrumentalities, and this directory includes various other runtime files (including other PHP files, JavaScript files, PHTML, and/or XML). It follows that a user will view the finalized website developed with said tools in a browser outside of the website authoring environment to verify the website conforms to the intended design. *See, e.g.,* https://techterms.com/definition/runtime.

29. The presence of the above referenced elements are demonstrated, by way of example, by reference to publicly available information. Regarding Drupal, *see, e.g.,* https://www.drupal.org/home; https://www.drupal.org/docs/8/system-requirements/browser-requirements; https://www.drupal.org/project/ckeditor; https://www.drupal.org/docs/8/core/modules/ckeditor/overview; https://dev.acquia.com/blog/tutorial-drupal-8-wysiwyg-inline-and-responsive-images/10/03/2016/9821; Angela Byron, *Ultimate Guide to Drupal 8* at 4 (2016); https://www.drupal.org/docs/7/understanding-drupal/technology-stack; https://www.drupal.org/docs/8/system-requirements/web-server; https://www.drupal.org/docs/8/core/modules/rest/overview; https://www.drupal.org/docs/8/core/modules/serialization/overview; https://www.drupal.org/docs/8/understanding-drupal-8/overview;

https://www.drupal.org/docs/develop/standards/css/css-architecture-for-drupal-8. Regarding Joomla, *see, e.g.,* https://docs.joomla.org/Joomla_Browser_Support; https://www.joomlart.com/documentation/other/joomla-3-and-joomla-2-5-system-requirement; https://showcase.joomla.org/; https://docs.joomla.org/Editor_form_field_type; https://developer.joomla.org/coding-standards/html.html; https://developer.joomla.org/coding-standards/css.html; https://developer.joomla.org/coding-standards/javascript.html; https://docs.joomla.org/Generating_JSON_output; https://api.joomla.org/cms-3/classes/Joomla.CMS.Input.Json.html; https://docs.joomla.org/How_do_you_assign_a_module_to_specific_pages%3F; https://docs.joomla.org/Where_are_the_web_pages%3F.

30. Claim 2 of the ’397 patent generally recites an apparatus for producing websites on and for computers having a browser and a virtual machine, said apparatus comprising an interface to present a settings menu which describes elements, said panel presented through a browser, where the selectable setting(s) corresponds to commands to the virtual machine; a browser to generate a display in accordance with selected setting(s); a database for storing information regarding selected settings; and a build tool having run time file(s) for generating web page(s) and using stored information to generate commands to the virtual machine for generating at least a portion of web page(s).

31. The Accused Instrumentalities infringe claim 2 of the ’397 patent through a combination of features which collectively practice each limitation of claim 2. By way of example, modern internet browsers such as Microsoft Internet Explorer, Mozilla’s Firefox, Apple Safari, Google Chrome, and Opera include virtual machines within the meaning of the ’397 patent. (*See, e.g.*, http://developer.telerik.com/featured/a-guide-to-javascript-engines-for-idiots/; http://dictionary.reference.com/browse/virtual+machine?s=t). The Accused Instrumentalities support the use of the latest versions of Internet Explorer 11 or later,

Microsoft Edge, latest–1, Firefox latest, latest–1, Chrome latest, latest–1, Safari latest, latest–1 (Mac OS), Safari Mobile for iPad 2, iPad Mini, iPad with Retina Display (iOS 7 or later), for desktop site, Safari Mobile for iPhone 4 or later; iOS 7 or later, for mobile site, Chrome for mobile latest–1 (Android 4 or later) for mobile site, where *latest–1* means one major version earlier than the latest released version. (*See, e.g.,* https://www.drupal.org/docs/8/system-requirements/browser-requirements; https://docs.joomla.org/Joomla_Browser_Support.) All of these browsers rely on browser engines comprising virtual machines to interpret and execute JavaScript and HTML to render web pages on a computer.

32. By way of further example, the Accused Instrumentalities enable users to produce websites through browsers on users' computers via interaction with an Internet server. For example, in order to add a new page to a user's website, the user logs in and then a server of the Accused Instrumentalities initiates presentation to the user through a browser of a website-builder tool. From the interface—sometimes referred to as a dashboard—of the Accused Instrumentalities, the user can navigate and add elements and element properties commensurate with a new page. A display is generated in accordance with one or more user selected settings substantially contemporaneously with the selection thereof. This is performed, for example, using a visual editing tool through a browser. The WYSIWYG interface for selecting center alignment of an image can also be accessed, and then the user can select various options such as a font and paragraph styles. After the user selects options such as image/text alignment or font and paragraph styles through the WYSIWYG editor, the display immediately updates to reflect the selected option. Furthermore, when images are uploaded by a user, those images are displayed in approximately 0-2 seconds depending on file size and bandwidth.

33. Data is stored in a database, including information corresponding to user selected settings such as, for example, the selections of text color. Other user selections are also stored including, for example, the layout, image filenames,

thumbnails, and paragraph margin settings for defining the alignment of an image location. The Accused Instrumentalities build one or more web pages to generate a website from at least a portion of a database and at least one run time file, where at least one run time file utilizes information stored in said database to generate virtual machine commands for the display of at least a portion of said one or more web pages.

34. At run time, at least some of these files use information stored in the database to generate the HTML for the final rendered HTML page. This HTML represents virtual machine commands for display of the page because it is read and used by the applicable browser's engine, including a virtual machine, in order to render the page. On information and belief, the Accused Instrumentalities further rely on the browser engine's component JavaScript engine to either display a portion of the page directly, or generate HTML to be executed for display by the main layout engine.

35. Additionally, the "PHP code," including the PHP template files, can be viewed in the file directory for the Accused Instrumentalities, and this directory includes various other runtime files (including other PHP files, JavaScript files, PHTML, and XML). *See, e.g.,* https://techterms.com/definition/runtime.

36. It follows that a user will view the finalized website developed with said tools in a browser outside of the website authoring environment to verify the website conforms to the intended design.

37. The presence of the above referenced elements are demonstrated, by way of example, by reference to publicly available information. Regarding Drupal, *see, e.g.,* https://www.drupal.org/home; https://www.drupal.org/docs/8/system-requirements/browser-requirements; https://www.drupal.org/project/ckeditor; https://www.drupal.org/docs/8/core/modules/ckeditor/overview; https://dev.acquia.com/blog/tutorial-drupal-8-wysiwyg-inline-and-responsive-images/10/03/2016/9821; Angela Byron, *Ultimate Guide to Drupal 8* at 4 (2016); https://www.drupal.org/docs/7/understanding-drupal/technology-stack; https://www.drupal.org/docs/8/system-requirements/web-server;

https://www.drupal.org/docs/8/core/modules/rest/overview; https://www.drupal.org/docs/8/core/modules/serialization/overview; https://www.drupal.org/docs/8/understanding-drupal-8/overview; https://www.drupal.org/docs/develop/standards/css/css-architecture-for-drupal-8. Regarding Joomla, *see, e.g.,* https://docs.joomla.org/Joomla_Browser_Support; https://www.joomlart.com/documentation/other/joomla-3-and-joomla-2-5-system-requirement; https://showcase.joomla.org/; https://docs.joomla.org/Editor_form_field_type; https://developer.joomla.org/coding-standards/html.html; https://developer.joomla.org/coding-standards/css.html; https://developer.joomla.org/coding-standards/javascript.html; https://docs.joomla.org/Generating_JSON_output; https://api.joomla.org/cms-3/classes/Joomla.CMS.Input.Json.html; https://docs.joomla.org/How_do_you_assign_a_module_to_specific_pages%3F; https://docs.joomla.org/Where_are_the_web_pages%3F.

38. Claim 3 of the ’397 patent recites the apparatus of claim 2, wherein the database is a multi-dimensional array structured database.

39. The Accused Instrumentalities infringe claim 3 of the ’397 patent through, by way of example, patent through a combination of features which collectively practice each limitation of claim 3.

40. By way of example, the JSON strings that are used to generate, in part, field capabilities originate from the database and therefore reflect the database structure and contents showing, on information and belief, the implementation of a multidimensional array structured database. By way of further evidence, the JSON strings show that there are dimensions for various parameters. *See, e.g.*, https://www.drupal.org/files/issues/Field.png; https://api.drupal.org/api/drupal/core%21modules%21field%21field.module/group/field/8.3.x; https://code.tutsplus.com/.

41. Claim 4 of the '397 patent recites the apparatus of claim 3, wherein the representative information is Boolean data, numeric data, string data or multi-dimensional arrays of various multimedia objects.

42. The Accused Instrumentalities infringe claim 4 of the '397 patent through a combination of features that practice the limitations of Claim 4. *See, e.g.*, https://www.drupal.org/docs/8/api/entity-api/defining-and-using-content-entity-field-definitions.

43. Claim 5 of the '397 patent recites the apparatus of claim 4, wherein said elements include multimedia objects selected from the group consisting of a color, a font, an image, an audio clip, a video clip, a text area and a URL.

44. The Accused Instrumentalities infringe claim 5 of the '397 patent through a combination of features that practice the limitations of Claim 5.

45. By way of example, the Accused Instrumentalities include various multimedia objects selected from a group contained within a WYSIWYG Editor. Examples include color, font, an image, a video, a text area and a URL as they appear in the WYSIWYG Editor. The multimedia objects created in the WYSIWYG editor are stored in the database and appear as HTML scripted text in the database. Text and vector objects can be selected and colored by selecting them or "click and dragging" over them in the WYSIWYG editor. A color may also be selected from the color dropdowns on the control bar of the Editor. This color is saved to the database; as part of the HTML of the description record. Moreover, text objects may be assigned a font by making such a selection or "click and dragging" over them in the WYSIWYG editor. A font can then be selected from the font dropdown on the control bar of the Editor. This font selection is thereafter saved to the database as part of the HTML of the description record. Selecting the Image button in the WYSIWYG editor opens a tabbed panel where the user designates source, title, format, size, etc. The image file is uploaded to the server and the file's location and style are saved and posted to the database as part of the HTML of the description record. Furthermore, videos are

created by clicking on the Media module, which opens a tabbed panel where the user designates URL, format, size, etc. The video's URL and style elements are saved to the database as part of the HTML of the description record. A text area may also be selected for creation by clicking in the frame of the WYSIWYG Editor and typing. The text and its style are saved to the database as part of the HTML of the description record. After entering text into the WYSIWYG editor's text area, a URL assigned by clicking and dragging over the text object you wish to link, and then selecting the "chain" link button from the control bar; which opens a tabbed panel where the user can designate the URL, target, etc. The text and its style are saved to the database as part of the HTML of the description record.

46. The presence of the above referenced elements are demonstrated, by way of example, by reference to publicly available information. *See, e.g.*, https://www.drupal.org/project/ckeditor; https://www.drupal.org/docs/8/core/modules/ckeditor/overview; https://dev.acquia.com/blog/tutorial-drupal-8-wysiwyg-inline-and-responsive-images/10/03/2016/9821; Angela Byron, *Ultimate Guide to Drupal 8* at 4 (2016); https://dev.acquia.com/blog/tutorial-drupal-8-wysiwyg-inline-and-responsive-images/10/03/2016/9821; https://www.drupal.org/docs/8/core/modules/media/overview; https://www.drupal.org/project/media_entity.

47. Claim 6 of the '397 patent recites the apparatus of claim 2, wherein said elements are selected from the group consisting of a button, an image, a paragraph, a frame, a table, a form and a vector object.

48. The Accused Instrumentalities infringe claim 6 of the '397 patent through a combination of features that practice the limitations of Claim 6.

49. By way of example, the Accused Instrumentalities include various user selectable menus where various elements can be placed on a web page. Those various user selectable menus are used to place elements selected from the group consisting of

a button, an image, a paragraph, a frame, a table, a form and a vector object. The cells of a table and maps would reside in a frame, and that, dividers, maps and the lines in tables would be, at least in part, vector objects.

50. The presence of the above referenced elements are demonstrated, by way of example, by reference to publicly available information. *See, e.g.*, https://www.drupal.org/project/ckeditor; https://www.drupal.org/docs/8/core/modules/ckeditor/overview; https://dev.acquia.com/blog/tutorial-drupal-8-wysiwyg-inline-and-responsive-images/10/03/2016/9821; Angela Byron, *Ultimate Guide to Drupal 8* at 4 (2016); https://www.drupal.org/docs/8/core/modules/custom_block/overview; https://www.drupal.org/docs/8/core/modules/contact/overview.

51. Claim 9 recites the apparatus of claim 2, wherein said elements include a button or an images, wherein said selectable settings include the selection of an element style, and wherein said build tool includes means for storing information representative of selected style in a database.

52. The Accused Instrumentalities infringe claim 9 of the ’397 patent through a combination of features which collectively practice each limitation of claim 9. *See, e.g.,* https://www.drupal.org/project/ckeditor; https://www.drupal.org/docs/8/core/modules/ckeditor/overview; https://dev.acquia.com/blog/tutorial-drupal-8-wysiwyg-inline-and-responsive-images/10/03/2016/9821; https://www.drupal.org/docs/8/core/modules/image/working-with-images; https://www.drupal.org/docs/7/understanding-drupal/technology-stack; https://www.drupal.org/docs/8/system-requirements/web-server; https://www.drupal.org/docs/8/core/modules/rest/overview; https://www.drupal.org/docs/8/core/modules/serialization/overview; https://www.drupal.org/docs/8/understanding-drupal-8/overview; https://www.drupal.org/docs/develop/standards/css/css-architecture-for-drupal-8.

53. Claim 10 recites the apparatus of claim 9, wherein said elements are described by multiple object states.

54. The Accused Instrumentalities infringe claim 10 of the '397 patent through a combination of features which collectively practice each limitation of claim 10. For example, buttons can have multiple object states. *See, e.g.,* https://www.drupal.org/docs/8/core/themes/seven-theme.

55. Claim 11 recites the apparatus of claim 9, wherein said elements are described by a transformation or a timelines of said selected styles.

56. The Accused Instrumentalities infringe claim 11 of the '397 patent through a combination of features which collectively practice each limitation of claim 11. By way of example, the Accused Instrumentalities support CSS architecture. *See, e.g.,* https://www.drupal.org/docs/develop/standards/css/css-architecture-for-drupal-8; *see also, e.g.,* http://demos.dojotoolkit.org/demos/css3/demo.html.

57. Claim 14 recites the apparatus of claim 2, wherein said elements include buttons or images, wherein said description of elements is a transition or a timeline which is selected according to input from a mouse, and wherein said build tool includes means for storing information representative of said selected description of elements in said database.

58. The Accused Instrumentalities infringe claim 14 of the '397 patent through a combination of features which collectively practice each limitation of claim 14.

59. By way of example, the Accused Instrumentalities include various CSS libraries that are used extensively for adding transformations and timelines to selected elements. *See, e.g.*, https://www.drupal.org/docs/develop/standards/css/css-architecture-for-drupal-8; http://demos.dojotoolkit.org/demos/css3/demo.html.

60. Claim 15 recites the apparatus of claim 14, wherein at least one of said description of elements is a timeline or an animation.

61. The Accused Instrumentalities infringe claim 15 of the '397 patent through a combination of features which collectively practice each limitation of claim 15.

62. By way of example, the Accused Instrumentalities enable descriptions of elements describing CSS animations. *See, e.g.*, https://www.drupal.org/docs/develop/standards/css/css-architecture-for-drupal-8; http://demos.dojotoolkit.org/demos/css3/demo.html.

63. Claim 17 recites the apparatus of claim 2, wherein one or more of said elements is a button or an image, wherein said description of elements is a transition, an animation or a timeline, and wherein said build engine includes means to synchronize said description of said one or more elements.

64. The Accused Instrumentalities infringe claim 17 of the '397 patent through a combination of features which collectively practice each limitation of claim 17.

65. By way of example, the Accused Instrumentalities enable the definition of certain parent elements and child element in certain classes, which can include the URL for an image, in the database, and that also can be zoomed. The parent element can include two transform styles, such as one for the parent and one for its lens.

66. Claim 20 recites the apparatus of claim 2, wherein at least one of said elements is a child button or a child object, wherein said description of said elements is a timeline, a transition or an animation, and wherein said build engine includes means for defining said description of said element.

67. The Accused Instrumentalities infringe claim 20 of the '397 patent through a combination of features which collectively practice each limitation of claim 20.

68. By way of example, the Accused Instrumentalities enable the description of elements as timelines or transition. Moreover, the build engine includes the means

for defining said description of said element through a choice of menu items through the design tab of the Product Information Admin Panel.

69. Claim 24 recites the apparatus of claim 2, wherein said run time files include one compressed website specific, customized run time engine program file and one compressed website specific, customized run time engine library file.

70. The Accused Instrumentalities infringe claim 24 of the '397 patent through a combination of features which collectively practice each limitation of claim 24.

71. By way of example, the Accused Instrumentalities include two customized runtime files, an HTML file and a second unique CSS file. *See, e.g.*, https://www.drupal.org/docs/7/understanding-drupal/technology-stack; https://www.drupal.org/docs/8/system-requirements/web-server; https://www.drupal.org/docs/8/core/modules/rest/overview; https://www.drupal.org/docs/8/core/modules/serialization/overview; https://www.drupal.org/docs/8/understanding-drupal-8/overview; https://www.drupal.org/docs/develop/standards/css/css-architecture-for-drupal-8.

72. Claim 25 recites the apparatus of claim 24, wherein said run time files include a dynamic web page scaling mechanism, whereby each of said one or more generated web pages is scaled for viewing on said display.

73. The Accused Instrumentalities infringe claim 25 of the '397 patent through a combination of features which collectively practice each limitation of claim 25.

74. By way of example, the Accused Instrumentalities enable rescaling of a web page to the size of the particular screen that is being used. *See, e.g.*, https://www.drupal.org/docs/8/mobile/responsive-web-design; https://www.drupal.org/docs/8/mobile/web-based-mobile-apps.

75. Claim 35 of the '397 patent generally recites the apparatus of claim 2, wherein the build tool includes dynamic resizing means operable to redefine a size of a web page upon being display.

76. The Accused Instrumentalities infringe claim 35 of the '397 patent through a combination of features which collectively practice each limitation of claim 35.

77. By way of example, the Accused Instrumentalities enable dynamic resizing upon display to a different device and screen. For example, the Accused Instrumentalities include "Responsive Web Design." Responsive Web Design refers to web design that changes formatting and lay-out to respond to different devices, screen sizes and browser capabilities. The Accused Instrumentalities therefore enable the creation of web pages that may be viewed with resizing means operable to redefine a size of a web page upon being displayed. *See, e.g.,* http://www.w3schools.com/html/html_responsive.asp; https://www.drupal.org/docs/8/mobile/responsive-web-design; https://www.drupal.org/docs/8/mobile/web-based-mobile-apps.

78. Claim 37 of the '397 patent generally recites [a]n apparatus for producing websites with web page(s) on and for a computer with a browser and a virtual machine, the apparatus comprising: an interface for building a website through control of website elements, being operable through the browser on to: present a selectable settings menu, accept settings, and generate the display in accordance with an assembly of settings contemporaneously with the acceptance thereof, at least one setting being operable to generate said display through commands to said virtual machine; an internal database associated with the interface for storing information representative of one or more of assembly of settings for controlling elements of the website; and a build tool to construct web page(s) of the website having: an external database containing data corresponding to the information stored in the internal database, and one or more run time files, where said run time files use information

stored in the external database to generate virtual machine commands for the display of at least a portion of one or more web pages.

79. The Accused Instrumentalities infringe claim 37 of the '397 patent through a combination of features which collectively practice each limitation of claim 37. By way of example, modern internet browsers such as Microsoft Internet Explorer, Mozilla's Firefox, Apple Safari, Google Chrome, and Opera include virtual machines within the meaning of the '397 patent. (*See, e.g.*, http://developer.telerik.com/featured/a-guide-to-javascript-engines-for-idiots/; http://dictionary.reference.com/browse/virtual+machine?s=t). The Accused Instrumentalities support the use of the latest versions of Internet Explorer 11 or later, Microsoft Edge, latest–1, Firefox latest, latest–1, Chrome latest, latest–1, Safari latest, latest–1 (Mac OS), Safari Mobile for iPad 2, iPad Mini, iPad with Retina Display (iOS 7 or later), for desktop site, Safari Mobile for iPhone 4 or later; iOS 7 or later, for mobile site, Chrome for mobile latest–1 (Android 4 or later) for mobile site, where *latest–1* means one major version earlier than the latest released version. (*See* https://www.drupal.org/docs/8/system-requirements/browser-requirements; https://docs.joomla.org/Joomla_Browser_Support.)

80. By way of example, the Accused Instrumentalities include various multimedia objects selected from a group contained within a WYSIWYG Editor. Examples include color, font, an image, a video, a text area and a URL as they appear in the WYSIWYG Editor. The multimedia objects created in the WYSIWYG editor are stored in the database and appear as HTML scripted text in the database. Text and vector objects can be selected and colored by selecting them or "clicking and dragging" over them in the WYSIWYG editor. A color may also be selected from the color dropdowns on the control bar of the Editor. This color is saved to the database; as part of the HTML of the description record. Moreover, text objects may be assigned a font by making such a selection or "click and dragging" over them in the WYSIWYG editor. A font can then be selected from the font dropdown on the

control bar of the Editor. This font selection is thereafter saved to the database as part of the HTML of the description record. Selecting the Image button in the WYSIWYG editor opens a tabbed panel where the user designates source, title, format, size, etc. The image file is uploaded to the server and the file's location and style are saved and posted to the database as part of the HTML of the description record. Furthermore, videos are created by clicking on the Media module, which opens a tabbed panel where the user designates URL, format, size, etc. The video's URL and style elements are saved to the database as part of the HTML of the description record. A text area may also be selected for creation by clicking in the frame of the WYSIWYG Editor and typing. The text and its style are saved to the database as part of the HTML of the description record. After entering text into the WYSIWYG editor's text area, a URL assigned by clicking and dragging over the text object you wish to link, and then selecting the "chain" link button from the control bar; which opens a tabbed panel where the user can designate the URL, target, etc. The text and its style are saved to the database as part of the HTML of the description record.

81. Furthermore, the Accused Instrumentalities enable data from the client-side form referenced to be stored in a server-side database.

82. The presence of the above referenced elements are demonstrated, by way of example, by reference to publicly available information. *See, e.g.,* https://www.drupal.org/home; https://www.drupal.org/docs/8/system-requirements/browser-requirements; https://www.drupal.org/project/ckeditor; https://www.drupal.org/docs/8/core/modules/ckeditor/overview; https://dev.acquia.com/blog/tutorial-drupal-8-wysiwyg-inline-and-responsive-images/10/03/2016/9821; Angela Byron, *Ultimate Guide to Drupal 8* at 4 (2016); https://www.drupal.org/project/save_draft; https://www.drupal.org/docs/7/understanding-drupal/technology-stack; https://www.drupal.org/docs/8/system-requirements/web-server; https://www.drupal.org/docs/8/core/modules/rest/overview;

https://www.drupal.org/docs/8/core/modules/serialization/overview; https://www.drupal.org/docs/8/understanding-drupal-8/overview; https://www.drupal.org/docs/develop/standards/css/css-architecture-for-drupal-8;. Regarding Joomla, *see, e.g.,* https://docs.joomla.org/Joomla_Browser_Support; https://www.joomlart.com/documentation/other/joomla-3-and-joomla-2-5-system-requirement; https://showcase.joomla.org/; https://docs.joomla.org/Editor_form_field_type; https://developer.joomla.org/coding-standards/html.html; https://developer.joomla.org/coding-standards/css.html; https://developer.joomla.org/coding-standards/javascript.html; https://docs.joomla.org/Generating_JSON_output; https://api.joomla.org/cms-3/classes/Joomla.CMS.Input.Json.html; https://docs.joomla.org/How_do_you_assign_a_module_to_specific_pages%3F; https://docs.joomla.org/Where_are_the_web_pages%3F.

83. Upon information and belief, these Accused Instrumentalities are used, marketed, provided to, and or used by or for each of Defendant's partners, clients, customers, and/or end users across the country and in this District.

84. Defendant was made aware of the '397 patent and its infringement thereof at least as early as the filing of this Complaint.

85. Since the date of the filing of this Complaint, Defendant's infringement of the '397 patent has been willful.

86. Plaintiff has not sold any product nor offered a service within the scope of any claim of the '397 patent. In addition, prior to August 12, 2015, no license to the '397 patent had been granted.

87. Plaintiff has been harmed by Defendant's infringing activities.

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

88. The allegations set forth in the foregoing paragraphs 1 through 87 are incorporated into this Second Claim for Relief.

89. On September 22, 2009, U.S. Patent No. 7,594,168 entitled *Browser Based Web Site Generation Tool and Run Time Engine* was duly and legally issued by the United States Patent and Trademark Office. A true and correct copy of the '168 patent is attached as Exhibit E.

90. The inventions of the '168 patent resolve 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, which features are exclusively implemented utilizing computer technology.

91. The claims of the '168 patent do not merely recite the performance of some business practice known from the pre-Internet world along with the requirement to perform it on the Internet. Instead, the claims of the '168 patent recite one or more inventive concepts that are rooted in computerized website creation technology, and overcome problems specifically arising in the realm of computerized website creation technologies.

92. The claims of the '168 patent recite an invention that is not merely the routine or conventional use of website creation systems and methods. Instead, the invention describes 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.

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

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

95. As noted above and incorporated into this Second Claim for Relief, defendants in other cases in which the '397 and '168 patents were asserted, asserted

that the '397 and '168 patents were invalid under 35 U.S.C. § 101. Those motions and related Orders are discussed above.

96. Plaintiff is the assignee and owner of the right, title and interest in and to the '168 patent, including the right to assert all causes of action arising under said patents and the right to any remedies for infringement of them.

97. Upon information and belief, Defendant has and continues to directly infringe at least claims 1-6 of the '168 patent by using a browser-based website and/or web page authoring tool in which the user-selected settings representing website elements are stored in a database, and retrieval of said information to generate said website (the "Accused Instrumentalities"). The Accused Instrumentalities include but are not limited website building tools used and/or provided by Defendant, such as, for example Drupal, Joomla, and/or Shopify. *See, e.g.*, https://www.possible.com/about.

98. In particular, claim 1 of the '168 patent generally recites a system for assembling a website comprising a server with a build engine, the website comprising web pages with objects (one button or one image object), the server accepting user input to associate a style with objects, wherein a button or image object is associated with a style that includes values defining transformations and time lines; wherein each web page is defined entirely by the objects and the style associated with the object, produce a database with a multidimensional array comprising the objects that comprise the website including data defining the object style, number, and an indication of the web page that each object is part of, and provide the database to a server accessible to web browser; wherein the database is produced such that a web browser with access to a runtime engine is configured to generate the website from the objects and style data extracted from the provided database.

99. The Accused Instrumentalities infringe claim 1 of the '168 patent through a combination of features which collectively practice each limitation of claim 1. (*See, e.g.*, https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model, http://www.w3schools.com/js/js_htmldom.asp.)

100. Further, by way of example, the JSON strings that are used by the Accused Instrumentalities to generate, in part, element formatting originate from the database and therefore reflect the database structure and contents showing, on information and belief, the implementation of a multidimensional array structured database comprising the objects that comprise the web site. By way of further evidence, the JSON strings show that there are dimensions for the pages, for arrays of columns, for arrays of sections, and for arrays of modules generated using the Accused Instrumentalities. *See, e.g.*, https://api.drupal.org/api/drupal/core%21modules%21field%21field.module/group/field/8.3.x

101. Further, the Accused Instrumentalities enable the storing in the database of data defining each object such as object styles, an object number, and an indication of the which page each object is a part of. For example, a user can select a theme style for a body title on a specific page. The CSS database file is thereafter saved to the server, reflecting the selected font, size, and the object and page to which it applies.

102. By way of example, for the completed web site, the Accused Instrumentalities include runtime files, such as, for example HTML CSS files. *See, e.g.*, https://www.drupal.org/home; https://www.drupal.org/docs/8/system-requirements/browser-requirements; https://www.drupal.org/project/ckeditor; https://www.drupal.org/docs/8/core/modules/ckeditor/overview; https://dev.acquia.com/blog/tutorial-drupal-8-wysiwyg-inline-and-responsive-images/10/03/2016/9821; Angela Byron, *Ultimate Guide to Drupal 8* at 4 (2016); https://www.drupal.org/docs/7/understanding-drupal/technology-stack; https://www.drupal.org/docs/8/system-requirements/web-server; https://www.drupal.org/docs/8/core/modules/rest/overview; https://www.drupal.org/docs/8/core/modules/serialization/overview; https://www.drupal.org/docs/8/understanding-drupal-8/overview;

https://www.drupal.org/docs/develop/standards/css/css-architecture-for-drupal-8; https://www.drupal.org/docs/8/core/modules/media/overview; https://www.drupal.org/project/media_entity; https://www.drupal.org/docs/8/core/modules/image/working-with-images; http://demos.dojotoolkit.org/demos/css3/demo.html; https://www.drupal.org/files/issues/Field.png; https://api.drupal.org/api/drupal/core%21modules%21field%21field.module/group/field/8.3.x.; https://www.drupal.org/project/save_draft.

103. Claim 2 of the '168 patent generally recites the system of claim 1, wherein one of said plurality of objects is a child, and wherein the build engine is configured to accept user input to associate a style with child button and child image objects.

104. The Accused Instrumentalities infringe claim 2 of the '168 patent through a combination of features which collectively practice each limitation of claim 2.

105. By way of example from Express Mobile's investigatory use of the Accused Instrumentalities, users of the Accused Instrumentalities are able to access child element images by clicking on a parent element which unlock additional images related to a product being viewed.

106. Claim 3 of the '168 patent generally recites the system of claim 2, wherein at least one of said styles includes values defining timelines for child button and child image objects.

107. The Accused Instrumentalities infringe claim 3 of the '168 patent through a combination of features which collectively practice each limitation of claim 3.

108. By way of example, the Accused Instrumentalities incorporate various CSS libraries, and CSS-animations and CSS-transitions are used extensively for

adding transformations and timelines to selected elements. On information and belief, this includes timelines for child buttons and child image objects.

109. Claim 4 of the '168 patent generally recites the system of claim 1, wherein at least one of said styles includes settings for multiple object states.

110. The Accused Instrumentalities infringe claim 4 of the '168 patent through a combination of features which collectively practice each limitation of claim 4.

111. By way of example, the Accused Instrumentalities enable the ability to define a hover state, so that an element, including a button, has defined styles. *See, e.g.*, https://www.drupal.org/docs/8/core/themes/seven-theme.

112. Claim 5 of the '168 patent generally recites the system of claim 1, further including file size reduction means for reducing total size of files generated by said build engine to a size between 12k and 50k.

113. On information and belief, the Accused Instrumentalities infringe claim 5 of the '168 patent through a combination of features which collectively practice each limitation of claim 5.

114. Claim 6 of the '168 patent generally recites the system of claim 1, where said data is stored as one or more of a Boolean an integer, a string, a floating point variables, or a URL.

115. The Accused Instrumentalities infringe claim 6 of the '168 patent through a combination of features which collectively practice each limitation of claim 6. A review of the API documentation behind websites created using the Accused Instrumentalities reveals data that is stored as one or more of a Boolean, an integer, or a string. *See, e.g.*, https://www.drupal.org/docs/8/api/entity-api/defining-and-using-content-entity-field-definitions.

116. Upon information and belief, these Accused Instrumentalities are used, marketed, provided to, and or used by or for each of Defendant's partners, clients, customers, and/or end users across the country and in this District.

117. Defendant was made aware of the '168 patent and its infringement thereof at least as early as the filing of this Complaint.

118. Since the date of the filing of this Complaint, Defendant's infringement of the '168 patent has been willful.

119. Plaintiff has not sold any product nor offered a service within the scope of any claim of the '168 patent. In addition, prior to August 12, 2015, no license to the '168 patent had been granted.

120. Plaintiff has been harmed by Defendant's infringing activities.

COUNT III – INFRINGEMENT OF U.S. PATENT NO. 9,471,287

121. The allegations set forth in the foregoing paragraphs 1 through 120 are incorporated into this Third Claim for Relief.

122. On October 18, 2016, U.S. Patent No. 9,471,287 ("the '287 patent"), entitled "*Systems and Methods for Integrating Widgets on Mobile Devices*," was duly and legally issued by the United States Patent and Trademark Office. A true and correct copy of the '287 patent is attached as Exhibit F.

123. The inventions of the '287 patent resolve technical problems related to generating content on a display of a device, such as the display of a mobile device. For example, the inventions of the '287 patent feature a registry and an authoring tool or Player configured to define a User Interface ("UI") object for display on the device, where the UI object corresponds to a web component. Each UI object is either: 1) selected by a user or 2) automatically selected by the system as a preferred UI object corresponding to a symbolic name of the web component and used to produce an Application, where the Application is a device-independent code; and a Player, where the Player is a device-dependent code. The Application and Player enable 1) the device to provide one or more input values and corresponding input symbolic name to the web service and 2) the web service to utilize the input symbolic name and the user provided one or more input values to generate one or more output values having an associated output symbolic name, while 3) the Player receives the output symbolic

name and corresponding one or more output values and provide instructions for the display of the device to present an output value in the defined UI object. These features are exclusively implemented utilizing computer technology.

124. The claims of the '287 patent do not merely recite the performance of some business practice known from the pre-Internet world along with the requirement to perform it on the Internet. Instead, the claims of the '287 patent recite one or more inventive concepts that are rooted in the computerized generation of content on a display of a device, such as a mobile device, and overcome problems specifically arising in the realm of computerized display content generation technologies.

125. The claims of the '287 patent recite inventions that are not merely the routine or conventional use of systems and methods for the computerized generation of content on a display of a device. Instead, the inventions feature systems for use with devices and methods of using the systems with authoring tools or Players specific to each device and Applications that are device independent.

126. The technology claimed in the '287 patent does not preempt all ways for the computerized generation of content on a display of a device, such as a mobile device, nor preempt the use of all authoring tools or Players for the computerized generation of content on a display of a device, such as a mobile devices, nor preempt any other well-known or prior art technology.

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

128. Plaintiff is the assignee and owner of the right, title and interest in and to the '287 patent, including the right to assert all causes of action arising under the patents and the right to any remedies for infringement of them.

129. Upon information and belief, Plaintiffs has and continues to directly infringe at least claims 1 and 15 of the '287 patent by a system and method which includes a registry and an authoring tool or Player configured to define a User

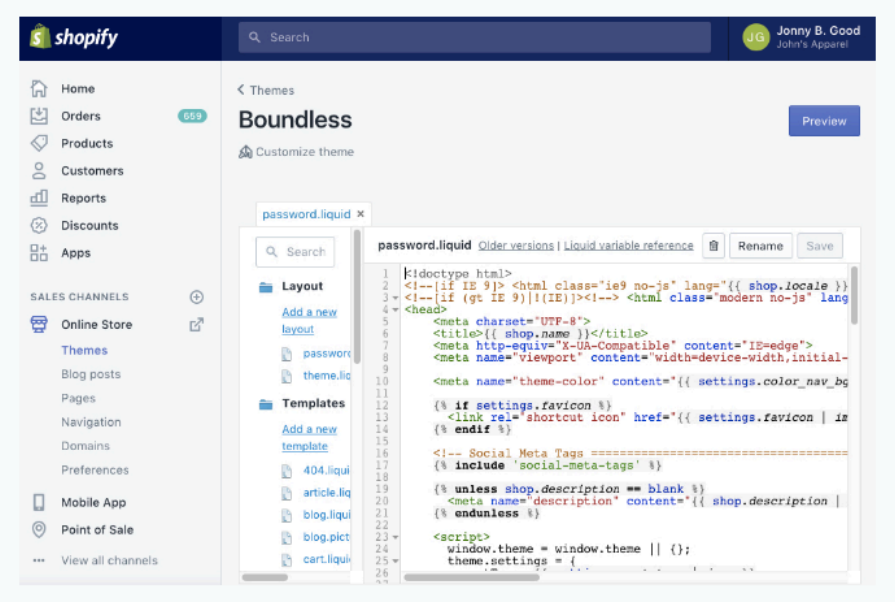
Interface ("UI") object for display on the device, where the UI object corresponds to a web component. Each UI object is either: 1) selected by a user or 2) automatically selected by the system as a preferred UI object corresponding to a symbolic name of the web component and used to produce an Application, where the Application is a device-independent code and a Player, where the Player is a device-dependent code. The Application and Player enable 1) the device to provide one or more input values and corresponding input symbolic name to the web service and 2) the web service to utilize the input symbolic name and the user provided one or more input values to generate one or more output values having an associated output symbolic name, while 3) 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 (the "Accused Instrumentalities"). The Accused Instrumentalities include the Shopify Theme Editor, based on the Ruby on Rails platform, that enables the functionality described above. *See, e.g.*, https://www.possible.com/about; https://medium.com/@chris.chimen/build-shopify-app-with-ruby-on-rails-for-beginer-part-1-40471da7d607.

130. In particular, claim 1 of the '287 patent recites 1 a system for generating code to provide content on a display of a device, the system comprising: computer memory storing a registry of: a) 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 has a preferred UI object, and b) an address of the web service; an authoring tool configured to: define a (UI) object for presentation on the display, where the defined UI object corresponds to a web component included in the registry 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, access the computer memory to select the symbolic name corresponding to the web component of the defined UI object, 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, and produce an Application including the selected symbolic name of the defined UI object, where the Application is a device-independent code; and a Player, where the Player is a device-dependent code, wherein, when the Application and Player are provided to the device and executed on the device, and 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, 1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service, 2) 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, 3) 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.

131. The Accused Instrumentalities infringe claim 1 of the ’287 patent through a combination of features which collectively practice each limitation of claim 1. By way of example, as shown in the screen shot below, the Accused Instrumentalities feature a system for generating code, including HTML, CSS, and JSON code, to provide content on a display of a device.

Edit HTML/CSS page



The **Edit HTML/CSS page** can be accessed through the **Themes** page of your Shopify Admin. It allows you to see all of your theme templates and assets, and directly make changes to them.

(https://help.shopify.com/themes/development/getting-started/choosing-an-editor).

132. The Accused Instrumentalities feature a computer memory provided by the Shopify servers and their MySQL database. By way of example, the Accused Instrumentalities' Ruby on Rails platform utilizes JSON strings extensively as part of its API, which necessarily require servers and databases. (See, https://developer.Ruby on Rails.org/rest-api/.)

133. The computer memory stores a registry of a) 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. In particular, the Accused Instrumentalities' MySQL database 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

by the formatting of the symbolic names in conjunction with the Shopify Theme Editor, widget, and plugin authoring tools.

134. Furthermore, each symbolic name has an associated data format class type corresponding to a subclass of defined UI objects *i.e.*, element/UI components, that supports the data format type of the symbolic name, and has a preferred UI object as evidenced by the JSON formatting of the name in conjunction with the Shopify Theme Editor and widget authoring tools. JSON names are strings that only represent the symbolic names that are bound both to a web service input and/or output and to a UI object. All JSON names in the name/value pairs are character strings. The Shopify Theme Editor includes elements for defining the layout for placement of the defined UI objects. Widgets, plug-ins and other elements correspond to the de3fined UI objects and are the product of the JSON formatting.

135. The computer memory also stores b) an address of the web service. Because the Shopify Theme Editor contains web services, it contains the corresponding addresses for the web services.

136. The Accused Instrumentalities feature an authoring tool in the form of the Shopify Theme Editor, widget, and plug-in authoring tools.

137. The authoring tool is configured to define a UI object for presentation on the display, where the defined UI object corresponds to a web component included in the registry selected from a group consisting of an input of the web service and an output of the web service. The Shopify Theme Editor's drag and drop component capabilities define the presence of a (UI) object for presentation on a display, where the defined UI object corresponds to a web component included in the registry (Shopify Server and mySQLdatabase) selected from a group consisting of an input of the web service and an output of the web service.

138. Each defined UI object automatically selected by the system as the preferred UI object corresponding to the symbollic name of the web component selected by the user of the authoring tool. The Accused Instrumentalities' UI objects

are 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, i.e., a UI object selected by a user is automatically selected.

139. The Accused Instrumentalities' authoring tool is configured to access the computer memory to select the symbolic name corresponding to the web component of the defined UI object by a JSON formatted element.

140. The Accused Instrumentalities' authoring tool is also configured to associate the selected symbolic name with the defined UI object, i.e., the JSON formatted element, where the selected symbolic name is only available to UI objects that support the defined data format associated with the element associated with that symbolic na me, i.e., JSON string. JSON names are strings that only represent the symbolic names that are bound both to a web service input and/or output and to a UI object. All JSON names in the name/value pairs are character strings. When the Shopify Theme Editor makes a UI element request a JSON request is sent to the Server and a JSON data response is provided to the Interface.

141. The Ruby on Rails authoring tool is configured to produce an Application including the selected symbolic name of the defined UI object, where the Application is a device-independent with its API and "hybrid" capabilities.

142. The Accused Instrumentalities authoring tool is further configured to produce a Player, where the Player is a device-dependent code. The Shopify Theme Editor contains a Player in the form of a run time player. The Accused Instrumentality produces a device dependent file, which is wrapped inside the runtime file, *i.e*., a file, including a run time engine, that is downloaded or created when a browser is pointed to a web page or website,. In order for a site to display on different devices through a browser or through responsive capabilities, there is device dependent code. *See, e.g.*, *X Commerce, Inc. v Express Mobile, Inc.*, Case No 17-cv-02605-RS, NDCA, DKT 79-8;

143. The Accused Instrumentalities feature a system where the Application and Player are provided to the device and executed on the device and 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. Because the Accused Instrumentalities incorporate a system that includes Ruby on Rails, when a user of the device provides one or more input values associated with an input symbolic name, using JSON formatting characteristics, to an input of the defined UI object, the device provides the user provided one or more input values and corresponding input symbolic name, using JSON formatting characteristics, to the web service.

144. The Accused Instrumentalities feature a system where the device provides the user provided one or more input values and corresponding input symbolic name to the web service. Because the Accused Instrumentalities incorporate a system that includes the Ruby on Rails platform, 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 defined UI object output value corresponds to the output symbolic name based on its JSON formatting characteristics.

145. The Accused Instrumentalities feature a system where 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. Because of the JSON formatting, the output values having an associated output symbolic name.

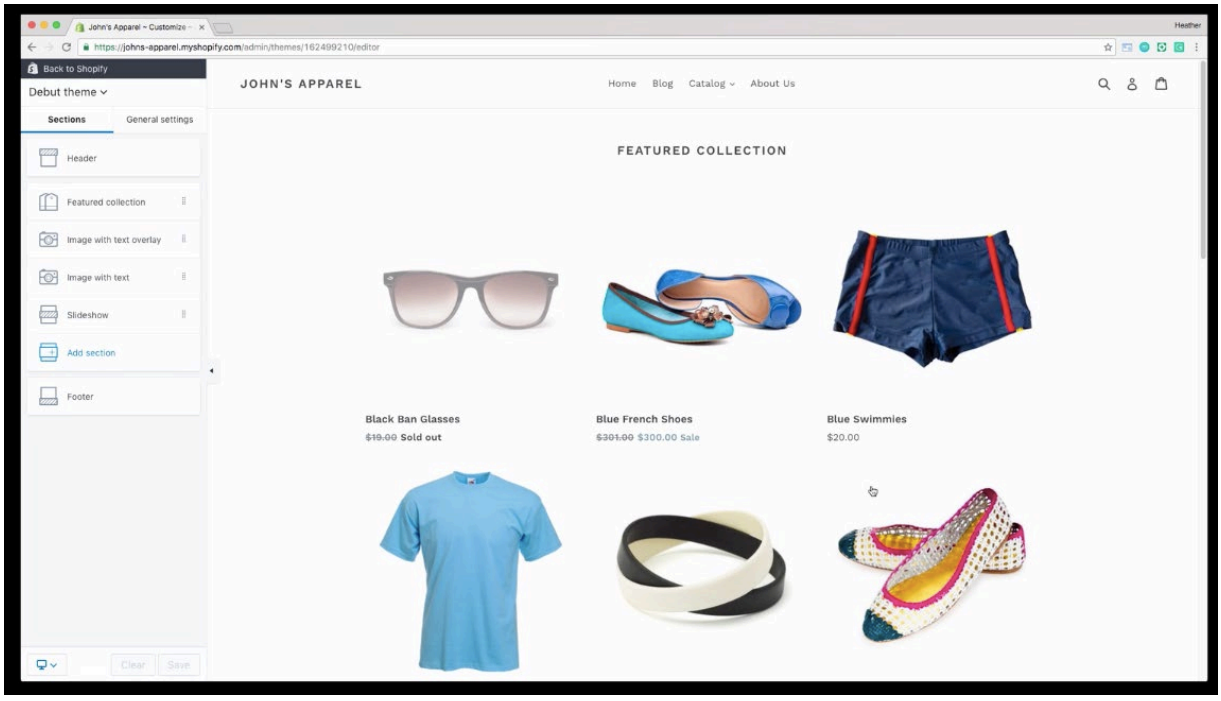
146. The Accused Instrumentalities feature a system where the Player receives the output symbolic name and corresponding one or more output values and provides instructions for the display of the device to present an output value in the defined UI object. The runtime player within Ruby on Rails receives the output name, output value, and provides instructions for a display as shown by the fact that the defined UI object are ultimately rendered

147. The presence of the above referenced features is demonstrated, by way of example, by reference to publicly available information, including https://www.shopify.com/; https://www.shopify.com/website/hosting; https://help.shopify.com/manual/using-themes/change-the-layout; https://help.shopify.com/manual/using-themes/change-the-layout/using-theme-presets https://help.shopify.com/manual/using-themes/troubleshooting/fix-64-kilobyte-limit-errors; https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html; https://www.slideshare.net/jduff/how-shopify-scales-rails-20443485; http://techstacks.io/shopify; https://help.shopify.com/manual/intro-to-shopify/pricing-plans/plan-features; https://help.shopify.com/api/reference; https://help.shopify.com/api/storefront-api https://help.shopify.com/api/sdks/shopify-apps/modifying-online-store/use-javascript-responsibly; https://help.shopify.com/api/reference/product; https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html; http://techstacks.io/shopify;l https://help.shopify.com/api/reference; https://help.shopify.com/manual/apps; https://api.rubyonrails.org/ https://guides.rubyonrails.org/rails_application_templates.html; https://teamtreehouse.com/community/ruby-on-rails-with-html-templates-2; https://guides.railsgirls.com/design; https://guides.rubyonrails.org/plugins.html; https://www.rubydoc.info/gems/dashing-rails; https://github.com/Shopify/dashing/wiki/Additional-Widgets; https://help.shopify.com/themes/customization/communication/add-contact-form; and https://help.shopify.com/manual/using-themes/change-the-layout/add-video.

148. Claim 15 of the ’287 patent recites a method of displaying content on a display of a device having a Player, where the Player is a device-dependent code, the method comprising: defining a user interface (UI) object for presentation on the display, where the UI object corresponds to a web component included in a registry of

one or more web components selected from a group consisting of an input of a web service and an output of the web service, where each web component includes a plurality of symbolic names of inputs and outputs associated with each web service, and where the registry includes: a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of the 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, and b) an address of the web service, and where each defined UI object is either:1) selected by a user of an authoring tool; 2) automatically selected by a system as a preferred UI object corresponding to a symbolic name of the web component selected by the user of the authoring tool.

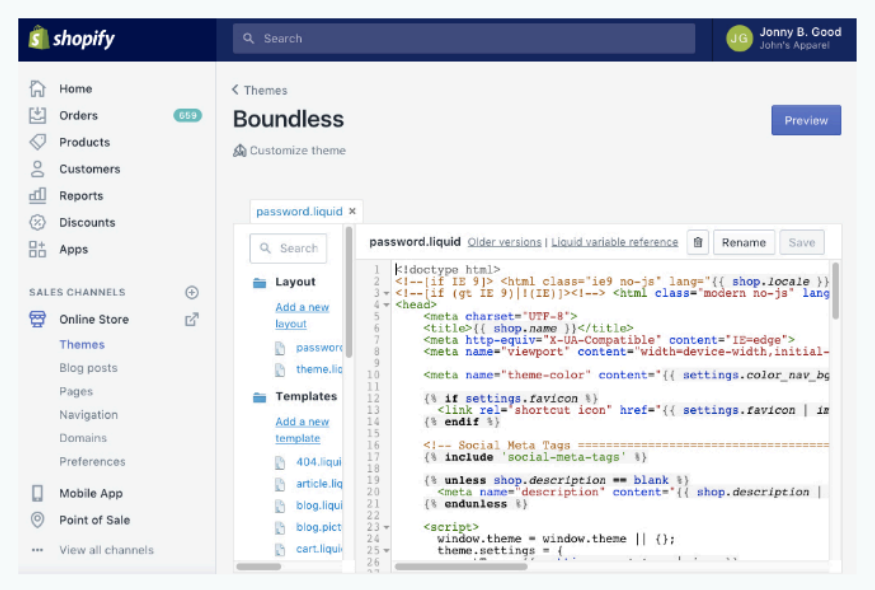
149. The Accused Instrumentalities infringe claim 15 of the '287 patent through a combination of features which collectively practice each limitation of claim 15. By way of example, the method is for displaying content on a display of a device as shown in the screen shot below.



(https://help.shopify.com/manual/using-themes/change-the-layout).

150. The Accused Instrumentalities feature a Player, where the Player is a device-dependent code. For example, Shopify, for Mobile Apps, produces a device dependent file, which is wrapped inside a runtime file. The HTML, CSS, Java, and JSON coding capabilities are s in the screen shot below.

Edit HTML/CSS page



The **Edit HTML/CSS page** can be accessed through the **Themes** page of your Shopify Admin. It allows you to see all of your theme templates and assets, and directly make changes to them.

(https://help.shopify.com/themes/development/getting-started/choosing-an-editor).

151. The Accused Instrumentalities include defining a user interface (UI) object for presentation on the display, where the defined UI object corresponds to a web component included in the registry of one or more web components, and where the web component is selected from a group consisting of an input of a web service and an output of the web service. TheShopify Theme Editor defines a UI object for presentation on display, where the UI object corresponds to a web component included in the non-volatile computer memory selected from a group consisting of an input of a web service and an output of the web service by JSON data formatting. JSON names are strings that only represent the symbolic names that are bound both to

a web service input and/or output and to a UI object. All JSON names in the name/value pairs are character strings. When the Ruby on Rails interface makes a UI element request a JSON request is sent to the Server and a JSON data response is provided to the Interface.

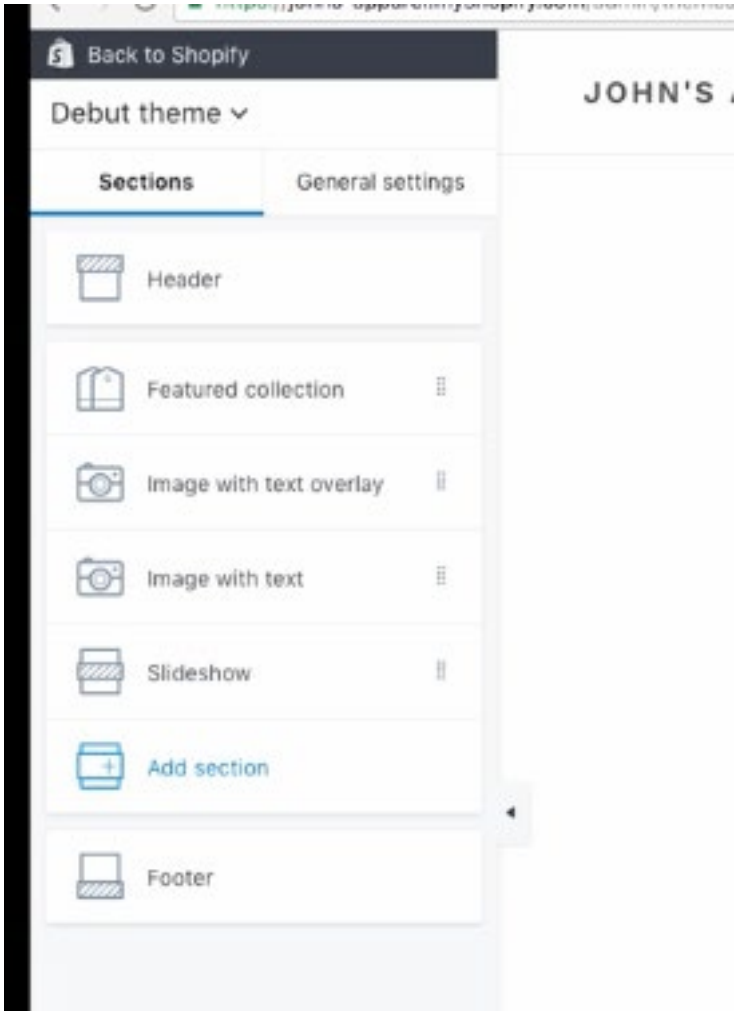
152. Each web component includes a plurality of symbolic names of inputs and outputs associated with each web service. The plurality of symbolic names of inputs and outputs associated with each web service is a feature of their JSON formatting characteristics. Each symbolic name has an associated data format class type corresponding to a subclass of UI objects that supports the data format type of the symbolic name, and has a preferred UI object as demonstrated by the presence of JSON formatting in conjunction with the Shopify Theme Editor and widget capabilities.

153. The registry includes: a) 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. The registry and Ruby on Rails MySQL database contain 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 as demonstrated by the formatting in conjunction with the Shopify Theme Editor, widget, and plugin authoring tools. JSON names are strings that only represent the symbolic names that are bound both to a web service input and/or output and to a UI object. All JSON names in the name/value pairs are character strings. When the Ruby on Rails interface makes a UI element request a JSON request is sent to the Server and a JSON data response is provided to the Interface.

154. The registry also includes b) an address of the web service. Because the Shopify Theme Editor contains web services, it contains the corresponding web addresses.

155. Each defined UI object is either: 1) selected by a user of an 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.

156. The Accused Instrumentalities include an authoring tool in the form of the Theme Editor shown the screen shot below



(https://help.shopify.com/manual/using-themes/change-the-layout).

157. The Shopify Theme Editor's UI objects are 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, i.e., a UI object selected by a user is automatically selected.

158. The Accused Instrumentalities include selecting the symbolic name from the web component (i.e Shopify Theme EditorWidget or Plug-in) corresponding to the defined UI object, where the selected symbolic name has an associated data format class type corresponding to a subclass of UI objects that support the data format type of the symbolic name and has the preferred UI object. The Shopify Theme Editor accesses its memory to select the symbolic name corresponding to the web component of the defined UI object (as evidenced by JSON data formatting), associate the selected symbolic name with the defined UI object (the JSON element corresponding to an element), where the selected symbolic name is only available to UI objects that support the defined data format associated with that symbolic name (the element associated with at JSON string). Additionally, the preferred UI object is the selected UI object. JSON names are strings that only represent the symbolic names that are bound both to a web service input and/or output and to a UI object. All JSON names in the name/value pairs are character strings. When the Ruby on Rails interface makes a UI element request a JSON request is sent to the Server and a JSON data response is provided to the Interface.

159. The Accused Instrumentalities include associating the selected symbolic name with the defined UI object.

160. The Accused Instrumentalities include producing an Application including the selected symbolic name of the defined UI object, where the Application is a device-independent code. Ruby on Rails produces an Application included in the symbolic name of the defined UI object. The inependent code is a arises from the API and “hybrid” capabilities.

161. The Accused Instrumentalities feature a system where the Application and Player are provided to the device and executed on the device and 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. Because the Accused Instrumentalities incorporate a system that includes the Shopify Theme Editor, when a user of the

device provides one or more input values associated with an input symbolic name, using JSON formatting characteristics, to an input of the defined UI object, the device provides the user provided one or more input values and corresponding input symbolic name, using JSON formatting characteristics, to the web service.

162. The Accused Instrumentalities feature a system where the device provides the user provided one or more input values and corresponding input symbolic name to the web service. Because the Accused Instrumentalities incorporate a system that includes Ruby on Rails, 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 defined UI object output value corresponds to the output symbolic name based on its JSON formatting characteristics.

163. The Accused Instrumentalities feature a system where 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. Because of the JSON formatting, the output values having an associated output symbolic name.

164. The Accused Instrumentalities feature a system where 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. The runtime player within Ruby on Rails receives the output name, output value, and provides instructions for a display as shown by the fact that the defined UI object are ultimately rendered.

The presence of the above referenced features is demonstrated, by way of example, by reference to publicly available information including: https://www.shopify.com/; https://www.shopify.com/website/hosting; https://help.shopify.com/manual/using-themes/change-the-layout; https://www.shopify.com/online; https://help.shopify.com/api/sdks/shopify-apps/apps-on-shopify-mobile;

https://developers.shopify.com/mobile-buy-sdk; thttps://help.shopify.com/manual/using-themes/change-the-layout/using-theme-presets https://help.shopify.com/manual/using-themes/troubleshooting/fix-64-kilobyte-limit-errors; https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html; https://www.slideshare.net/jduff/how-shopify-scales-rails-20443485; http://techstacks.io/shopify; https://help.shopify.com/manual/intro-to-shopify/pricing-plans/plan-features; https://help.shopify.com/api/reference; https://help.shopify.com/api/storefront-api https://help.shopify.com/api/sdks/shopify-apps/modifying-online-store/use-javascript-responsibly; https://help.shopify.com/api/reference/product; https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html; http://techstacks.io/shopify;l https://help.shopify.com/api/reference; https://help.shopify.com/manual/apps; https://api.rubyonrails.org/

165. https://guides.rubyonrails.org/rails_application_templates.html; https://teamtreehouse.com/community/ruby-on-rails-with-html-templates-2; https://guides.railsgirls.com/design; https://guides.rubyonrails.org/plugins.html; https://www.rubydoc.info/gems/dashing-rails; https://github.com/Shopify/dashing/wiki/Additional-Widgets; https://help.shopify.com/themes/customization/communication/add-contact-form; and https://help.shopify.com/manual/using-themes/change-the-layout/add-video.

166. Upon information and belief, these Accused Instrumentalities are used, marketed, provided to, and or used by or for each of Defendant's partners, clients, customers, and/or end users across the country and in this District.

167. Defendant was made aware of the '287 patent and its infringement thereof at least as early as the filing of this Complaint.

168. Since the date of the filing of this Complaint, Defendant's infringement of the '287 patent has been willful.

169. Within the past six years, Plaintiff has not sold any product nor offered a service within the scope of any claim of the '287 patent. In addition, prior to August 12, 2015, no license to the '287 patent had been granted.

170. Plaintiff has been harmed by Defendant's infringing activities.

COUNT IV – INFRINGEMENT OF U.S. PATENT NO. 9,063,755

171. The allegations set forth in the foregoing paragraphs 1 through 170 are incorporated into this Third Claim for Relief.

172. On June 23, 2015, U.S. Patent No. 9,063,755 ("the '755 patent"), entitled "Systems and methods for presenting information on mobile devices," was duly and legally issued by the United States Patent and Trademark Office. A true and correct copy of the '755 patent is attached as Exhibit H.

173. The inventions of the '755 patent resolve technical problems related to a system a system for generating code to provide content on a display of a device. The system includes a computer memory and an authoring tool, where the computer memory stores a registry of: a) symbolic names required for evoking one or more web components related to a web service and b) an address of the web service. The authoring tool is configured to: define a (UI) object for presentation on the display, where the defined UI object corresponds to a web component included in the registry selected from a group consisting of an input of the web service and an output of the web service; access the computer memory to select the symbolic name corresponding to the web component of the defined UI object, associate the selected symbolic name with the defined UI object; produce an Application including the selected symbolic name of the defined UI object, where the Application is a device-independent code; and produce a Player, where the Player is a device-dependent code, such that 1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service, 2) 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 3) the Player receives the

output symbolic name and corresponding one or more output values and provides instructions. These features are exclusively implemented utilizing computer technology.

174. The claims of the '755 patent do not merely recite the performance of some business practice known from the pre-Internet world along with the requirement to perform it on the Internet. Instead, the claims of the '755 patent recite one or more inventive concepts that are rooted in the computerized generation of content on a display of a device, such as a mobile device, and overcome problems specifically arising in the realm of computerized display content generation technologies.

175. The claims of the '755 patent recite inventions that are not merely the routine or conventional use of systems and methods for the computerized generation of content on a display of a device. Instead, the inventions feature systems for use with devices and methods of using the systems with authoring tools or Players specific to each device and Applications that are device independent.

176. The technology claimed in the '755 patent does not preempt all ways for the computerized generation of code for a display of a device, such as a mobile device, nor preempt the use of all authoring tools or Players for the computerized generation of content on a display of a device, such as a mobile devices, nor preempt any other well-known or prior art technology.

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

178. Plaintiff is the assignee and owner of the right, title and interest in and to the '755 patent, including the right to assert all causes of action arising under the patents and the right to any remedies for infringement of them.

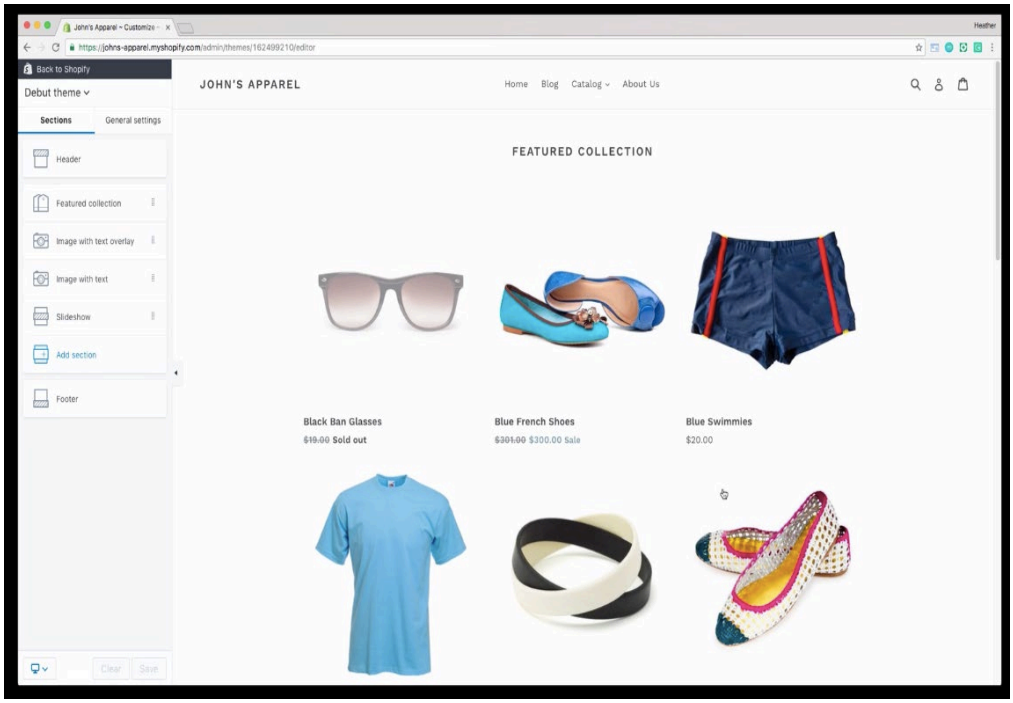
179. Upon information and belief, Plaintiffs has and continues to directly infringe at least claims 1 and 15 of the '755 patent by a system a system for generating code to provide content on a display of a device. The system includes a computer

memory and an authoring tool, The computer memory stores a registry of: a) 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 has a preferred UI object, and b) an address of the web service. The authoring tool is configured to: define a (UI) object for presentation on the display, where the defined UI object corresponds to a web component included in the registry selected from a group consisting of an input of the web service and an output of the web service; access the computer memory to select the symbolic name corresponding to the web component of the defined UI object, associate the selected symbolic name with the defined UI object; produce an Application including the selected symbolic name of the defined UI object, where the Application is a device-independent code; and produce a Player, where the Player is a device-dependent code, such that when the Application and Player are provided to the device and executed on the device, and 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, 1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service, 2) 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, 3) the Player receives the output symbolic name and corresponding one or more output values and provides instructions. (The "Accused Instrumentalities"). The Accused Instrumentalities include the Shopify Theme Editor, based on the Ruby on Rails platform, that enables the functionality described above. *See, e.g.*, https://www.possible.com/about;

https://medium.com/@chris.chimen/build-shopify-app-with-ruby-on-rails-for-beginer-part-1-40471da7d607.

180. In particular, claim 1 of the '755 patent recites 1 a system for generating code to provide content on a display of a device, said system comprising: computer memory storing a registry of: a) 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, and b) the address of the web service; an authoring tool configured to: define a user interface (UI) object for presentation on the display, where said UI object corresponds to the web component included in said registry selected from the group consisting of an input of the web service and an output of the web service, access said computer memory to select the symbolic name corresponding to the web component of the defined UI object, associate the selected symbolic name with the defined UI object, produce an Application including the selected symbolic name of the defined UI object, where said Application is a device-independent code, and produce a Player, where said Player is a device-dependent code; such that, when the Application and Player are provided to the device and executed on the device, and when a user of the device provides one or more input values associated with an input symbolic name to an input of defined UI object, 1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service, 2) 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, 3) said 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.

181. The Accused Instrumentalities infringe claim 1 of the '755 patent through a combination of features which collectively practice each limitation of claim 1. By way of example, as shown in the screen shot below, the Accused Instrumentalities feature a method of displaying content of a device.



(https://help.shopify.com/manual/using-themes/change-the-layout).

182. The Accused Instrumentalities feature a registry of one or more web components related to inputs and outputs of a web service obtainable over a network. The registry is provided by the Shopify servers and their MySQL database. By way of example, the Accused Instrumentalities' Ruby on Rails platform utilizes JSON strings extensively as part of its API, which necessarily require servers and databases. (See, https://developer.Ruby on Rails.org/rest-api/.)

183. Each web component includes a plurality of symbolic names of inputs and outputs associated with each web service.

184. Furthermore, the registry 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. In particular, the Accused Instrumentalities' MySQL database 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 by the formatting of the symbolic names in conjunction with the Shopify Theme Editor, widget, and plugin authoring tools.

185. The computer memory also stores b) an address of the web service. Because the Shopify Theme Editor contains web services, it contains the corresponding addresses for the web services.

186. The Accused Instrumentalities feature an authoring tool in the form of the Shopify Theme Editor, widget, and plug-in authoring tools.

187. The authoring tool is configured to define a UI object for presentation on the display, where the defined UI object corresponds to a web component included in the registry selected from a group consisting of an input of the web service and an output of the web service. The Shopify Theme Editor's drag and drop component capabilities define the presence of a (UI) object for presentation on a display, where the defined UI object corresponds to a web component included in the registry (Shopify Server and mySQLdatabase) selected from a group consisting of an input of the web service and an output of the web service.

188. The Accused Instrumentalities' authoring tool is configured to access the computer memory to select the symbolic name corresponding to the web component of the defined UI object by a JSON formatted element.

189. The Accused Instrumentalities' authoring tool is also configured to associate the selected symbolic name with the defined UI object, i.e., the JSON formatted element, where the selected symbolic name is only available to UI objects that support the defined data format associated with the element associated with that symbolic na me, i.e., JSON string. JSON names are strings that only represent the symbolic names that are bound both to a web service input and/or output and to a UI

object. All JSON names in the name/value pairs are character strings. When the Shopify Theme Editor makes a UI element request a JSON request is sent to the Server and a JSON data response is provided to the Interface.

190. The Accused Instrumentalities authoring tool is configured to produce an Application including the selected symbolic name of the defined UI object, where the Application is a device-independent. The Accused Instrumentalities' Application is comprised of the mySQLdatabase, including all the user selectable settings, as augmented by the web component definition The application, because it contains user selected settings, and is represented by Boolean, numbers and String primitives, is device independent, and stored in the mySQL database for each application.

191. The Accused Instrumentalities authoring tool is further configured to produce a Player, where the Player is a device-dependent code. The Shopify Theme Editor contains a Player in the form of a run time player. The Accused Instrumentality produces a device dependent file, which is wrapped inside the run time file, i.e., *i.e*., a file, including a run time engine, that is downloaded or created when a browser is pointed to a web page or website,. In order for a site to display on different devices through a browser or through responsive capabilities, there is device dependent code. *See, e.g.*, *X Commerce, Inc. v Express Mobile, Inc.*, Case No 17-cv-02605-RS, NDCA, DKT 79-5.

192. The Accused Instrumentalities feature a system where the Application and Player are provided to the device and executed on the device and 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. Because the Accused Instrumentalities incorporate a system that includes Ruby on Rails, when a user of the device provides one or more input values associated with an input symbolic name, using JSON formatting characteristics, to an input of the defined UI object, the device provides the user provided one or more input values and corresponding input symbolic name, using JSON formatting characteristics, to the web service.

193. The Accused Instrumentalities feature a system where the device provides the user provided one or more input values and corresponding input symbolic name to the web service. Because the Accused Instrumentalities incorporate a system that includes the Ruby on Rails platform, 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 defined UI object output value corresponds to the output symbolic name based on its JSON formatting characteristics.

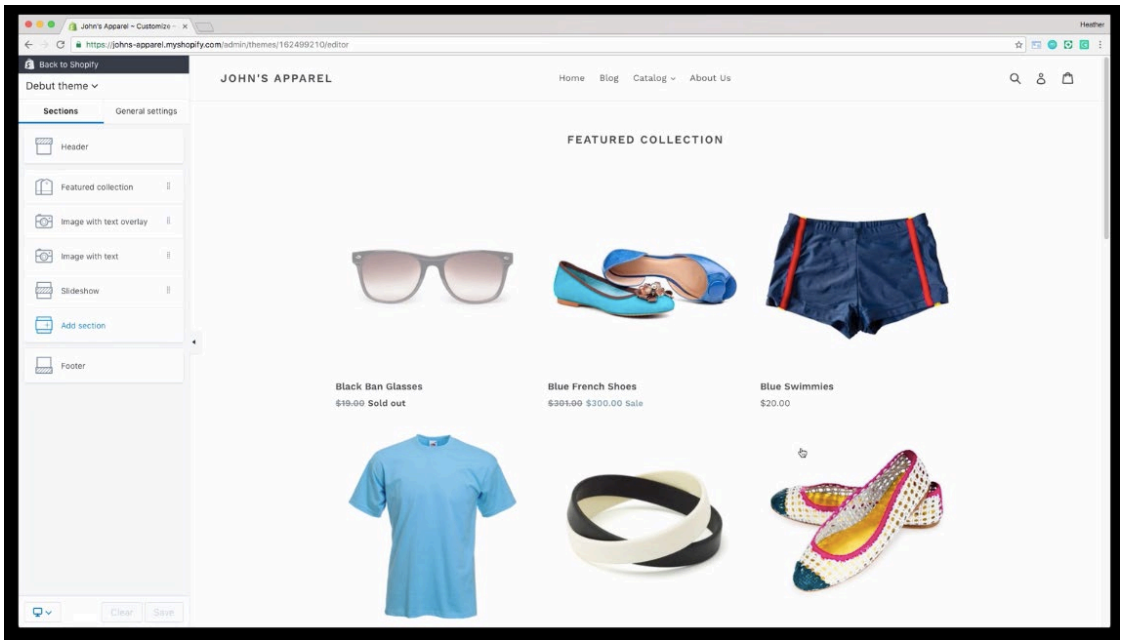
194. The Accused Instrumentalities feature a system where the web service utilizes the input symbolic name and the user provided one or more input values for generating one or more output values having an associated output symbolic name. Because of the JSON formatting, the output values having an associated output symbolic name.

195. The Accused Instrumentalities feature a system where 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. The runtime player within Ruby on Rails receives the output name, output value, and provides instructions for a display as shown by the fact that the defined UI object are ultimately rendered.

196. The presence of the above referenced features is demonstrated, by way of example, by reference to publicly available information, including https://www.shopify.com/; https://www.shopify.com/website/hosting; https://help.shopify.com/manual/using-themes/change-the-layout; https://help.shopify.com/manual/using-themes/change-the-layout/using-theme-presets https://help.shopify.com/manual/using-themes/troubleshooting/fix-64-kilobyte-limit-errors; https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html; https://www.slideshare.net/jduff/how-shopify-scales-rails-20443485; http://techstacks.io/shopify; https://help.shopify.com/manual/intro-to-shopify/pricing-

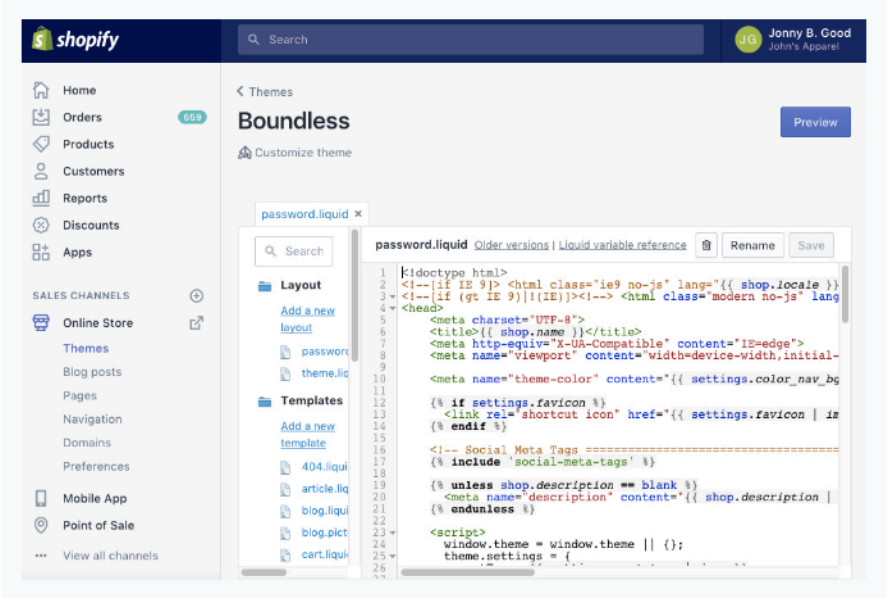
plans/plan-features; https://help.shopify.com/api/reference; https://help.shopify.com/api/storefront-api https://help.shopify.com/api/sdks/shopify-apps/modifying-online-store/use-javascript-responsibly; https://help.shopify.com/api/reference/product; https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html; http://techstacks.io/shopify;l https://help.shopify.com/api/reference; https://help.shopify.com/manual/apps; https://api.rubyonrails.org/; https://guides.rubyonrails.org/rails_application_templates.html; https://teamtreehouse.com/community/ruby-on-rails-with-html-templates-2; https://guides.railsgirls.com/design; https://guides.rubyonrails.org/plugins.html; https://www.rubydoc.info/gems/dashing-rails; https://github.com/Shopify/dashing/wiki/Additional-Widgets; https://help.shopify.com/themes/customization/communication/add-contact-form; and https://help.shopify.com/manual/using-themes/change-the-layout/add-video.

197. The Accused Instrumentalities infringe claim 12 of the ’755 patent through a combination of features which collectively practice each limitation of claim 12. By way of example, the method is for displaying content on a display of a device as shown in the screen shot below.



https://help.shopify.com/manual/using-themes/change-the-layout).

Edit HTML/CSS page



The **Edit HTML/CSS page** can be accessed through the **Themes** page of your Shopify Admin. It allows you to see all of your theme templates and assets, and directly make changes to them.

(https://help.shopify.com/themes/development/getting-started/choosing-an-editor).

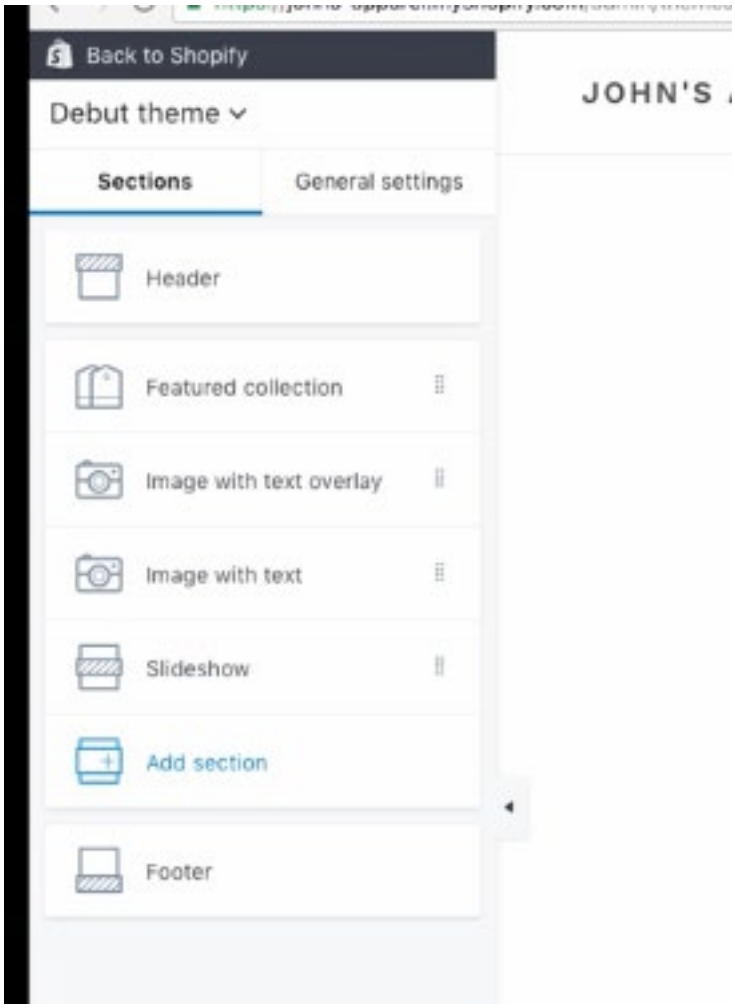
198. The Accused Instrumentalities feature a registry of one or more web components, inputs and outputs of a web service obtainable over a network and an output of the web service by JSON data formatting. JSON names are strings that only represent the symbolic names that are bound both to a web service input and/or output and to a UI object. All JSON names in the name/value pairs are character strings. When the Ruby on Rails interface makes a UI element request a JSON request is sent to the Server and a JSON data response is provided to the Interface.

199. Each web component includes a plurality of symbolic names of inputs and outputs associated with each web service. The plurality of symbolic names of inputs and outputs associated with each web service is a feature of their JSON formatting characteristics. Each symbolic name has an associated data format class type corresponding to a subclass of UI objects that supports the data format type of the symbolic name, and has a preferred UI object as demonstrated by the presence of JSON formatting in conjunction with the Shopify Theme Editor and widget capabilities.

200. The registry includes: a) 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. The registry and Ruby on Rails MySQL database contain 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 as demonstrated by the formatting in conjunction with the Shopify Theme Editor, widget, and plugin authoring tools. JSON names are strings that only represent the symbolic names that are bound both to a web service input and/or output and to a UI object. All JSON names in the name/value pairs are character strings. When the Ruby on Rails interface makes a UI element request a JSON request is sent to the Server and a JSON data response is provided to the Interface.

201. The registry also includes b) an address of the web service. Because the Shopify Theme Editor contains web services, it contains the corresponding web addresses.

202. Accused Instrumentalities define a user interface (UI) object for presentation on the display, where said UI object corresponds to a web component included in said registry selected from the group consisting of an input of the web service and an output of the web service. The Accused Instrumentalities contain an authoring tool in the form of the Shopify Theme Editor shown in the screen shot below.



(https://help.shopify.com/manual/using-themes/change-the-layout).

The ShopifyTheme Editor's drag and drop component capabilities define the presence of a (UI) object for presentation on a display, where the defined UI object

corresponds to a web component included in the registry selected from a group consisting of an input of the web service and an output of the web service.

203. The Accused Instrumentalities include selecting the symbolic name from the web component (i.e Shopify Theme EditorWidget or Plug-in) corresponding to the defined UI object, where the selected symbolic name has an associated data format class type corresponding to a subclass of UI objects that support the data format type of the symbolic name and has the preferred UI object. The Shopify Theme Editor accesses its memory to select the symbolic name corresponding to the web component of the defined UI object (as evidenced by JSON data formatting), associate the selected symbolic name with the defined UI object (the JSON element corresponding to an element), where the selected symbolic name is only available to UI objects that support the defined data format associated with that symbolic name (the element associated with at JSON string). Additionally, the preferred UI object is the selected UI object. JSON names are strings that only represent the symbolic names that are bound both to a web service input and/or output and to a UI object. All JSON names in the name/value pairs are character strings. When the Ruby on Rails interface makes a UI element request a JSON request is sent to the Server and a JSON data response is provided to the Interface.

204. The Accused Instrumentalities include associating the selected symbolic name with the defined UI object.

205. The Accused Instrumentalities include producing an Application including the selected symbolic name of the defined UI object, where the Application is a device-independent code. The Application is comprised of the mySQL database that includes all the user selectable settings, augmented by the web component definition. The application, because it contains user selected settings, and is represented by Boolean, numbers and String primitives, is device independent, and stored in a database for each application.

206. The Accused Instrumentalities also include producing a Player, where the Player is a device-dependent code. The Shopify Theme Editor contains a Player in the form of a runtime player.

207. The Accused Instrumentalities feature a system where the Application and Player are provided to the device and executed on the device and 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. Because the Accused Instrumentalities incorporate a system that includes the Shopify Theme Editor, when a user of the device provides one or more input values associated with an input symbolic name, using JSON formatting characteristics, to an input of the defined UI object, the device provides the user provided one or more input values and corresponding input symbolic name, using JSON formatting characteristics, to the web service.

208. The Accused Instrumentalities feature a system where the device provides the user provided one or more input values and corresponding input symbolic name to the web service. Because the Accused Instrumentalities incorporate a system that includes Ruby on Rails, 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 defined UI object output value corresponds to the output symbolic name based on its JSON formatting characteristics.

209. The Accused Instrumentalities feature a system where 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. Because of the JSON formatting, the output values having an associated output symbolic name.

210. The Accused Instrumentalities feature a system where 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. The runtime player within Ruby on Rails receives the output name, output value, and provides instructions for a display as shown by the fact that the defined UI object are ultimately rendered.

211. The presence of the above referenced features is demonstrated, by way of example, by reference to publicly available information includig. https://www.shopify.com/; https://www.shopify.com/website/hosting; https://help.shopify.com/themes/development/getting-started/choosing-an-editor; https://help.shopify.com/manual/using-themes/change-the-layout; https://www.shopify.com/online; https://help.shopify.com/api/sdks/shopify-apps/apps-on-shopify-mobile; https://developers.shopify.com/mobile-buy-sdk; thttps://help.shopify.com/manual/using-themes/change-the-layout/using-theme-presets https://help.shopify.com/manual/using-themes/troubleshooting/fix-64-kilobyte-limit-errors; https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html; https://www.slideshare.neasst/jduff/how-shopify-scales-rails-20443485; http://techstacks.io/shopify; https://help.shopify.com/manual/intro-to-shopify/pricing-plans/plan-features; https://help.shopify.com/api/reference; https://help.shopify.com/api/storefront-api https://help.shopify.com/api/sdks/shopify-apps/modifying-online-store/use-javascript-responsibly; https://help.shopify.com/api/reference/product; https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html; http://techstacks.io/shopify;l https://help.shopify.com/api/reference; https://help.shopify.com/manual/apps; https://api.rubyonrails.org/; https://guides.rubyonrails.org/rails_application_templates.html; https://teamtreehouse.com/community/ruby-on-rails-with-html-templates-2; https://guides.railsgirls.com/design; https://guides.rubyonrails.org/plugins.html; https://www.rubydoc.info/gems/dashing-rails; https://github.com/Shopify/dashing/wiki/Additional-Widgets;

https://help.shopify.com/themes/customization/communication/add-contact-form; and https://help.shopify.com/manual/using-themes/change-the-layout/add-video.

213. Upon information and belief, these Accused Instrumentalities are used, marketed, provided to, and or used by or for each of Defendant's partners, clients, customers, and/or end users across the country and in this District.

214. Defendant was made aware of the '755 patent and its infringement thereof at least as early as the filing of this Complaint.

215. Since the date of the filing of this Complaint, Defendant's infringement of the '755 patent has been willful.

216. Within the past six years, Plaintiff has not sold any product nor offered a service within the scope of any claim of the '755 patent. In addition, prior to August 12, 2015, no license to the '755 patent had been granted.

217. Plaintiff has been harmed by Defendant's infringing activities.

JURY DEMAND

Pursuant to Rule 38 of the Federal Rules of Civil Procedure, Plaintiff demands a trial by jury on all issues triable as such.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff demands judgment for itself and against Defendant as follows:

A. An adjudication that Defendant has infringed the '397, '168, '287, and '755 patents;

B. An award of damages to be paid by Defendant adequate to compensate Plaintiff for Defendant's past infringement of the '397, '168, '287, and '755 patents, and any continuing or future infringement through the date such judgment is entered, including interest, costs, expenses and an accounting of all infringing acts including, but not limited to, those acts not presented at trial;

C. A declaration that this case is exceptional under 35 U.S.C. § 285, and an award of Plaintiff's reasonable attorneys' fees; and

D. An award to Plaintiff of such further relief at law or in equity as the Court deems just and proper.

Dated: June 12, 2019

By: */s/ Jeffrey Francis Craft*

Jeffrey Francis Craft (SBN 147186)
jcraft@devlinlawfirm.com
DEVLIN LAW FIRM LLC
1731 Fox Springs Circle,
Newbury Park, CA 91320

Attorneys for Plaintiff Express Mobile, Inc.