Jeffrey Francis Craft SBN 147186
jcraft@devlinlawfirm.com
DEVLIN LAW FIRM LLC
2069 Cold Canyon Road
Calabasas, CA 91302
Telephone: (302) 449-9010
Facsimile: (302) 353-4251

Timothy Devlin (*pro hac vice* to be filed)
tdevlin@devlinlawfirm.com
Robert Dean Kiddie, Jr. (*pro hac vice* to be filed)
rkiddie@devlinlawfirm.com
DEVLIN LAW FIRM LLC
1306 N. Broom St., 1st Floor
Wilmington, DE 19806
Telephone: (302) 449-9010
Facsimile: (302) 353-4251

Seth Wiener SBN 203747
seth@sethwienerlaw.com
LAW OFFICES OF SETH WIENER
609 Karina Court
San Ramon, CA 94582
Telephone: (925) 487-5607

Attorneys for Express Mobile, Inc.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

EXPRESS MOBILE, INC., Plaintiff, vs. HAPPIEST MINDS TECHNOLOGIES PVT. LTD., Defendant.	Case No.: 3:18-cv-04683-RS **FIRST AMENDED COMPLAINT FOR PATENT INFRINGEMENT** DEMAND FOR JURY TRIAL

Plaintiff Express Mobile, Inc. ("Express Mobile" or "Plaintiff"), for its First Amended Complaint against Defendant Happiest Minds Technologies Pvt. Ltd., ("HMT" 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, HMT is a company organized and existing under the laws of India, with a place of business at 2051 Junction Ave., Suite 208, San Jose, CA 95131 and can be served through its registered agent, Incorporating Services, Ltd., 7801 Folsom Blvd., Suite 202, Sacramento, CA 95826.

4. Upon information and belief, HMT sells and offers to sell products and services throughout the United States, including in this judicial district, and introduces products and services that 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 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." (Dkt. 29, attached hereto as Exhibit C.) No objection was filed to the Magistrate Judge's report and recommendation and the decision therefore became final.

18. 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.

19. Upon information and belief, Defendant has and continues to directly infringe at least claims 1-6, 8-11, 14-15, 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 and/or Wordpress. *See, e.g.*, https://www.happiestminds.com/services/technology-services/.

20. On information and belief, Defendant is a for-profit organization with revenues of approximately $500 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.

21. 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).

22. 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-requirements; http://themeforest.net/category/wordpress.) 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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/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

Wordpress, *see, e.g.,* http://themeforest.net/category/wordpress; http://codex.wordpress.org/Templates; http://codex.wordpress.org/Template_Hierarchy; http://codex.wordpress.org/Function_Reference/the_title; http://codex.wordpress.org/Function_Reference/the_content; https://www.wpbeginner.com/glossary/database/; https://codex.wordpress.org/Pages; http://codex.wordpress.org/Templates; http://codex.wordpress.org/Template_Tags/get_the_title; and http://codex.wordpress.org/Query_Overview.

28. 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).

29. 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; http://themeforest.net/category/wordpress.) 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.

30. 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.

31. 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.

32. 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.

33. 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.

34. 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.

35. 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/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 Wordpress, *see, e.g.,* http://themeforest.net/category/wordpress; http://codex.wordpress.org/Templates; http://codex.wordpress.org/Template_Hierarchy; http://codex.wordpress.org/Function_Reference/the_title; http://codex.wordpress.org/Function_Reference/the_content; https://www.wpbeginner.com/glossary/database/; https://codex.wordpress.org/Pages;

http://codex.wordpress.org/Templates; http://codex.wordpress.org/Template_Tags/get_the_title; and http://codex.wordpress.org/Query_Overview.

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

37. 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.

38. 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. Regarding Drupal, *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. Regarding Wordpress, *see, e.g.,* https://code.tutsplus.com/; https://wordpress.stackexchange.com/questions/43302/wordpress-settings-api-and-option-array-structure.

39. 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.

40. The Accused Instrumentalities infringe claim 4 of the '397 patent through a combination of features that practice the limitations of Claim 4. Regarding Drupal, *see, e.g.*, https://www.drupal.org/docs/8/api/entity-api/defining-and-using-content-entity-field-definitions. Regarding Wordpress, *see, e.g.,* https://wordpress.stackexchange.com/questions/43302/wordpress-settings-api-and-option-array-structure.

41. 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.

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

43. 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.

44. 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.

45. 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.

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

47. 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.

48. 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.

49. Claim 8 recites the apparatus of claim 2, wherein said elements include one or more objects on a web page, and wherein said description of elements are a transition or an animation of at least one of said elements on a web page.

50. The Accused Instrumentalities infringe claim 8 of the '397 patent through a combination of features which collectively practice each limitation of claim 8. *See, e.g.,* https://wordpress.org/plugins/animate-everything/.

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. Regarding Drupal *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. Regarding Wordpress, *see, e.g.,* https://www.wpbeginner.com/wp-tutorials/how-to-add-custom-styles-to-wordpress-visual-editor/.

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. Regarding Drupal *see, e.g.,* https://www.drupal.org/docs/8/core/themes/seven-theme. Regarding Wordpress, *see, e.g.,* https://wordpress.org/plugins/animate-everything/.

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. Regarding Drupal, *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. Regarding Wordpress, *see, e.g.,* https://wordpress.org/plugins/animate-everything/.

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 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.

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

65. 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.

66. 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.

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

68. 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.

69. 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.

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

71. 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.

72. 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.

73. 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; http://themeforest.net/category/wordpress.)

74. 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.

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

76. 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/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 Wordpress, *see, e.g.,* http://themeforest.net/category/wordpress; http://codex.wordpress.org/Templates; http://codex.wordpress.org/Template_Hierarchy; http://codex.wordpress.org/Function_Reference/the_title; http://codex.wordpress.org/Function_Reference/the_content; http://codex.wordpress.org/Template_Tags/get_the_title; http://codex.wordpress.org/Query_Overview; https://www.wpbeginner.com/glossary/database/; and https://codex.wordpress.org/Pages.

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

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

79. Plaintiff has been harmed by Defendant’s infringing activities.

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

80. The allegations set forth in the foregoing paragraphs 1 through 79 are incorporated into this Second Claim for Relief.

81. 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 B.

82. 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.

83. 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.

84. 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.

85. 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.

86. 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.

87. As noted above and incorporated into this Second Claim for Relief, a defendant in another case in which the '397 and '168 patents were asserted, asserted that the '397 and '168 patents were invalid under 35 U.S.C. § 101. That motion and related Order are discussed above.

88. 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.

89. Upon information and belief, Defendant has and continues to directly infringe at least claims 1, 4, and 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. *See, e.g.*, https://www.happiestminds.com/solutions/drupal/.

90. 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.

91. The Accused Instrumentalities infringe claim 1 of the '168 patent through a combination of features which collectively practice each limitation of claim 1.

92. 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://www.drupal.org/files/issues/Field.png; https://api.drupal.org/api/drupal/core%21modules%21field%21field.module/group/field/8.3.x

93. 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.

94. 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.

95. 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.

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

97. 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.

98. 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.

99. 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.

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

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

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

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

103. The allegations set forth in the foregoing paragraphs 1 through 102 are incorporated into this Third Claim for Relief.

104. 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 D.

105. 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-e 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.

106. 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.

107. 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.

108. 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.

109. 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.

110. 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.

111. Upon information and belief, Defendant has and continues to directly infringe at least claims 1-5, 11, 12, 15-19, 25 and 26 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 platforms that enable the functionality described above and include but are not limited to, for example, WordPress. *See, e.g.*, https://www.happiestminds.com/services/technology-services/. For example, Defendant provides comprehensive implementation of a number of product solutions including, but not limited to the WordPress platform.

112. 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.

113. The Accused Instrumentalities infringe claim 1 of the ’287 patent through a combination of features 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, the Accused Instrumentalities feature a system for generating code to provide content on a display of a device. The system includes a server hosting the WordPress platform, which provides WordPress’s WYSIWYG visual effects editor, and which is accessed through a WordPress-compatible browser. WordPress’s WYSIWYG visual effects editor generates code, such as JavaScript or HTML code, for such options as defining title, text, images, videos and paragraph styles, while the browser displays the resulting content as a WordPress webpage on a display of a device.

114. For example, on information and belief, WordPress uses a variety of databases in its technology stack including MySQL. Data from the wp_options table for the website header and the data from the wp_posts table for the “WordPress Info” web page extracted directly from the Bitnami WordPress server-side database using MySQL Workbench. The stored data in the wp_options table

includes the website's url, the website's title (blogname), the website's tagline (blogdescription), and the active template (style sheet).

option_id	option_name	option_value
1	siteurl	http://localhost/wordpress
2	blogname	Express Mobile
3	blogdescription	SELECTED USERS OF WORDPRESS
44	template	the-fundamentals-of-graphic-design

The stored data in the wp_ posts table for the "WordPress Info" web page includes information corresponding to user selected settings such as, for example, the color red ("#ff0000") for "manages 22%". Other user selections shown in stored database data below include, for ex-ample, the image filename for the image (http://localhost/wordpress/wp-content/uploads/2013/03/icon21.png), the image alignment (class="wp-image-24 aligncenter"), and a selected paragraph style (h3) for "Heading 3".

The stored data in the wp_ posts table for the "WordPress Info" web page includes information corresponding to user selected settings such as, for example, the color red ("#ff0000") for "manages 22%". Other user selections shown in stored database data below include, for ex-ample, the image filename for the image (http://localhost/wordpress/wp-content/uploads/2013/03/icon21.png), the image alignment (class="wp-image-24 aligncenter"), and a selected paragraph style (h3) for "Heading 3".

ID	post_content	post_title	post_type
23	WordPress is used by over 14.7% of Alexa Internet's "top 1 million" websites and as of August, 2011, believe it or not, manages 22% of all new websites. WordPress is currently the most popular blogging system in use on the Web. As of December 2011, WordPress version 3.0 had been downloaded over 65 million times. 	WordPress Info	page

WordPress's HTML, CSS, Java, and JSON coding capabilities further are shown, e.g., by http://codex.wordpress.org/Templates, https://codex.wordpress.org/CSS , https://developer.wordpress.org/rest-api/.

115. The Accused Instrumentalities feature a computer memory provided by WordPress MySQL database functionality on the WordPress server. By way of example, WordPress utilizes JSON strings extensively as part of its API, which necessarily require servers and databases. (See, https://developer.wordpress.org/rest-api/.)

116. 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. The WordPress 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 WordPress's WYSIWYG visual effects editor, widget, and plugin authoring tools.

117. 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 WordPress' WYSIWYG visual effects 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. WordPress' WYSIWYG visual effects editor includes elements for defining the layout for placement of the defined UI objects. Widgets, plug-ins and other elements correspond to the defined UI objects and are the product of the JSON formatting. (See, for example, https://codex.wordpress.org/WordPress_Lessons#Template_Files, https://codex.wordpress.org/WordPress_Widgets, https://codex.wordpress.org/Plugin_Resources, and https://codex.wordpress.org/Plugin_API .)

118. The computer memory also stores b) an address of the web service. Because WordPress contains web services, it contains the corresponding addresses for the web services. (See, for example, https://developer.wordpress.org/rest-api/, https://codex.wordpress.org/WordPress_Widgets, https://codex.wordpress.org/Plugin_Resources, and https://codex.wordpress.org/Plugin_API .)

119. The Accused Instrumentalities feature an authoring tool in the form of WordPress's WYSIWYG visual effects editor, widget, and plug-in authoring tools. (See, for example, https://codex.wordpress.org/WordPress_Widgets, https://codex.wordpress.org/Plugin_Resources, https://codex.wordpress.org/Plugin_API .)

120. 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. WordPress's WYSIWYG visual effects editor and widget authoring tools define the presence of a defined UI object for presentation on a display and the defined UI object corresponds to a web component included in the computer memory selected from a group consisting of an input of the web service and an output of the web service.

121. 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. WordPress'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. When a Widget is selected in the WordPress Widget selection list steps 1 to 5 under "Displaying Widgets" , the widget UI will automatically displayed in the Web Page Sidebar. (See, for example, https://codex.wordpress.org/WordPress_Widgets, https://codex.wordpress.org/Plugin_Resources, https://codex.wordpress.org/Plugin_API .)

122. The 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.

123. The 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 name, 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 WordPress Editor makes a UI element request a JSON request is sent to the Server and a JSON data response is provided to the Interface. (See, for example, https://developer.wordpress.org/rest-api/, https://codex.wordpress.org/WordPress_Widgets, https://codex.wordpress.org/Plugin_Resources, https://codex.wordpress.org/Plugin_API.)

124. The WordPress authoring tool is configured to produce an Application including the selected symbolic name of the defined UI object, (see, for example, https://developer.wordpress.org/rest-api/reference/, https://codex.wordpress.org/Widgets_API, https://codex.wordpress.org/Plugin_Resources, and https://codex.wordpress.org/Plugin_API), where the Application is a device-independent with its API and "responsive" capabilities. (See, for

example, https://torquemag.io/2017/08/make-wordpress-website-mobile-friendly/ and https://torquemag.io/2017/08/make-wordpress-website-mobile-friendly/)

125. The WordPress authoring tool is further configured to produce a Player (see, for example, https://codex.wordpress.org/CSS, https://codex.wordpress.org/Using_Javascript, https://developer.wordpress.org/rest-api/), where the Player is a device-dependent code. WordPress contains a Player in the form of a runtime player. The Accused Instrumentality produces a device dependent file, which is wrapped inside the runtime file. In order for a site to display on different devices through a browser or through responsive capabilities, there is device dependent code (see, for example, https://torquemag.io/2017/08/make-wordpress-website-mobile-friendly/ and https://wordpress.org/themes/ultra/...)

126. 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 WordPress, 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. (See, for example, https://developer.wordpress.org/rest-api/, https://developer.wordpress.org/rest-api/reference/, https://codex.wordpress.org/Widgets_API, https://codex.wordpress.org/Plugin_Resources, and https://codex.wordpress.org/Plugin_API .)

127. 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 WordPress, 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. (See, for example, https://developer.wordpress.org/rest-api/,

https://developer.wordpress.org/rest-api/reference/, https://codex.wordpress.org/Widgets_API, https://codex.wordpress.org/Plugin_Resources, and https://codex.wordpress.org/Plugin_API .)

128. 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. (See, for example, https://developer.wordpress.org/rest-api/, https://developer.wordpress.org/rest-api/reference/, https://codex.wordpress.org/Widgets_API, https://codex.wordpress.org/Plugin_Resources, and https://codex.wordpress.org/Plugin_API .)

129. 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 WordPress 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. (See, for example, https://developer.wordpress.org/rest-api/, https://developer.wordpress.org/rest-api/reference/, https://codex.wordpress.org/Widgets_API, https://codex.wordpress.org/Plugin_Resources, and https://codex.wordpress.org/Plugin_API .)

130. The presence of the above referenced features is demonstrated, by way of example, by reference to publicly available information. Regarding WordPress, see, e.g., http://themeforest.net/category/wordpress; http://codex.wordpress.org/Templates; http://codex.wordpress.org/Template_Hierarchy; http://codex.wordpress.org/Function_Reference/the_title; http://codex.wordpress.org/Function_Reference/the_content; https://www.wpbeginner.com/glossary/database/; https://codex.wordpress.org/Pages; http://codex.wordpress.org/Templates; http://codex.wordpress.org/Template_Tags/get_the_title; and http://codex.wordpress.org/Query_Overview. .

131. Claim 2 of the ’287 patent recites a system for generating code to provide content on a display of a device that includes all the elements of claim 1, additionally where the registry includes definitions of input and output related to the web service.

132. The Accused Instrumentalities infringe claim 2 of the ’287 patent through a combination of features which collectively practice each limitation of claim 1. By way of example, the registry includes definitions of input and output related to a web service as evidenced by WordPress’s JSON formatting characteristics of the defined UI objects. (See, for example, https://developer.wordpress.org/rest-api/; https://developer.wordpress.org/rest-api/reference/; https://developer.wordpress.org/rest-api/reference/posts/ https://developer.wordpress.org/rest-api/reference/posts/#schema-title https://codex.wordpress.org/Widgets_API; https://codex.wordpress.org/Plugin_Resources; and https://codex.wordpress.org/Plugin_API; https://codex.wordpress.org/WordPress_Widgets)

133. Claim 3 of the ’287 patent recites a system for generating code to provide content on a display of a device that includes all the elements of claim 1, additionally where the web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.

134. The Accused Instrumentalities infringe claim 3 of the ’287 patent through a combination of features which collectively practice each limitation of claim 3. By way of example, the Accused Instrumentalities feature web components additionally including web chat, Reuters RSS feed, Calendar image, and map image widgets.

135. Claim 4 of the ’287 patent recites a system for generating code to provide content on a display of a device that includes all the elements of claim 1, additionally where the defined UI object is an input field for a chat.

136. The Accused Instrumentalities infringe claim 4 of the ’287 patent through a combination of features which collectively practice each limitation of claim 4. By way of example, the Accused Instrumentalities additionally feature a defined UI object that is an input field for a chat.

137. Claim 5 of the ’287 patent recites a system for generating code to provide content on a display of a device that includes all the elements of claim 1, additionally the defined UI object is an input field for a web service.

138. The Accused Instrumentalities infringe claim 5 of the '287 patent through a combination of features which collectively practice each limitation of claim 1. By way of example, the Accused Instrumentalities additionally feature a defined UI object that is an input field for a web service.

139. Claim 11 of the '287 patent recites a system for generating code to provide content on a display of a device that includes all the elements of claim 1, additionally where the code is provided over the network.

140. The Accused Instrumentalities infringe claim 11 of the '287 patent through a combination of features which collectively practice each limitation of claim 11. WordPress sends all files over a network using a variety of databases in its technology stack including MySQL. These backend capabilities provided the code over a network. By way of example, data from the wp_options table for the website header and from the wp_posts table for the "WordPress Info" web page are extracted directly from the Bitnami WordPress server-side database using MySQL Workbench. The stored data in the wp_options table includes the website's url, the website's title (blogname), the website's tagline (blogdescription), and the active template (style sheet).

option_id	option_name	option_value
1	siteurl	http://localhost/wordpress
2	blogname	Express Mobile
3	blogdescription	SELECTED USERS OF WORDPRESS
44	template	the-fundamentals-of-graphic-design

The stored data in the wp_ posts table for the "WordPress Info" web page includes information corresponding to user selected settings such as, for example, the color red ("#ff0000") for "manages 22%". Other user selections shown in stored database data below include, for ex-ample, the image filename for the image (http://localhost/wordpress/wp-content/uploads/2013/03/icon21.png), the image alignment (class="wp-image-24 aligncenter"), and a selected paragraph style (h3) for "Heading 3".

ID	post_content	post_title	post_type
23	WordPress is used by over 14.7% of Alexa Internet's "top 1 million" websites and as of August, 2011, believe it or not, manages 22% of all new websites. WordPress is currently the most popular blogging system in use on the Web. As of December 2011, WordPress version 3.0 had been downloaded over 65 million times. 	WordPress Info	page

WordPress's HTML, CSS, Java, and JSON coding capabilities further are shown, for example. by http://codex.wordpress.org/Templates, https://codex.wordpress.org/CSS , https://developer.wordpress.org/rest-api/ and https://codex.wordpress.org/Widgets_API.).

141. Claim 12 of the '287 patent recites a system for generating code to provide content on a display of a device that includes all the elements of claim 1, additionally where the defined UI object corresponds to a widget.

142. The Accused Instrumentalities infringe claim 12 of the '287 patent through a combination of features which collectively practice each limitation of claim 1. By way of example, the Accused Instrumentalities' incorporation of WordPress includes widgets. (See, for example., https://codex.wordpress.org/Widgets_API.)

143. 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.

144. 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 and the Accused Instrumentalities include a WordPress-compatible browser which content as a WordPress webpage on a display of a device.

145. The Accused Instrumentalities feature a Player, where the Player is a device-dependent code. The device has a Player (see, for example, https://codex.wordpress.org/CSS, https://codex.wordpress.org/Using_Javascript, https://developer.wordpress.org/rest-api/) in the form of a runtime player that is a device dependent code. WordPress produces a device dependent file, which is wrapped inside a runtime file. In order for a site to display on different devices through a browser or through responsive capabilities, there is device dependent code. (See, for example, https://torquemag.io/2017/08/make-wordpress-website-mobile-friendly/; https://wordpress.org/themes/ultra/.)

146. 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, where the web component is selected from a group consisting of an input of a web service and an output of the web service. WordPress 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. (See, for example, https://developer.wordpress.org/rest-api/, https://developer.wordpress.org/rest-api/reference/,

https://codex.wordpress.org/WordPress_Widgets , https://codex.wordpress.org/Plugin_API , and https://codex.wordpress.org/Plugin_Resources.) 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 WordPress interface makes a UI element request a JSON request is sent to the Server and a JSON data response is provided to the Interface. (See, for example, https://developer.wordpress.org/rest-api/, https://codex.wordpress.org/WordPress_Widgets, https://codex.wordpress.org/Plugin_Resources, https://codex.wordpress.org/Plugin_API.)

147. 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 WordPress' WYSIWYG and widget capabilities. (See, for example, https://developer.wordpress.org/rest-api/, https://developer.wordpress.org/rest-api/, https://codex.wordpress.org/WordPress_Widgets , https://codex.wordpress.org/Plugin_API , and https://codex.wordpress.org/Plugin_Resources.)

148. 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 WordPress 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 WordPress's WYSIWYG visual effects 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 WordPress interface makes a UI element request a JSON request is sent to the Server and a JSON data response is provided to the Interface. (See, for example,

https://developer.wordpress.org/rest-api/, https://codex.wordpress.org/WordPress_Widgets, https://codex.wordpress.org/Plugin_Resources, https://codex.wordpress.org/Plugin_API.)

149. The registry also includes b) an address of the web service. Because WordPress contains web services, it contains the corresponding web addresses. Because WordPress contains web services, both as a library of Widgets (see https://codex.wordpress.org/Widgets_API) and Plug-ins (see https://codex.wordpress.org/Plugin_API*)*, it contains the corresponding addresses of the web services. All web services are represented as a wsdl (Web Service Description Language) and wsdls have URLs to point to the internet location that receives the web service's inputs and returns the web service's outputs (See https://www.soapui.org/soap-and-wsdl/working-with-wsdls.html).

150. 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. WordPress'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. When a Widget is selected in the WordPress Widget selection list steps 1 to 5 under "Displaying Widgets" , the widget UI will automatically displayed in the Web Page Sidebar. (See, for example, https://codex.wordpress.org/WordPress_Widgets, https://codex.wordpress.org/Plugin_Resources, https://codex.wordpress.org/Plugin_API .)

151. The Accused Instrumentalities include selecting the symbolic name from the web component (i.e. WordPress Widget 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. WordPress 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). (See, for example, https://developer.wordpress.org/rest-api/, https://developer.wordpress.org/rest-api/reference/,

https://codex.wordpress.org/WordPress_Widgets , https://codex.wordpress.org/Plugin_API , and https://codex.wordpress.org/Plugin_Resources.)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 WordPress interface makes a UI element request a JSON request is sent to the Server and a JSON data response is provided to the Interface. (See, for example, https://developer.wordpress.org/rest-api/, https://codex.wordpress.org/WordPress_Widgets, https://codex.wordpress.org/Plugin_Resources, https://codex.wordpress.org/Plugin_API.)

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

153. 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. WordPress produces an Application included in the symbolic name of the defined UI object. (See, for example, https://developer.wordpress.org/rest-api/, https://developer.wordpress.org/rest-api/reference/, https://codex.wordpress.org/WordPress_Widgets , https://codex.wordpress.org/Plugin_API , and https://codex.wordpress.org/Plugin_Resources.)

154. 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 WordPress, 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. (See, for example, https://developer.wordpress.org/rest-api/, https://developer.wordpress.org/rest-api/reference/, https://codex.wordpress.org/Widgets_API, https://codex.wordpress.org/Plugin_Resources, and https://codex.wordpress.org/Plugin_API .)

155. 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 WordPress, 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. (See, for example, https://developer.wordpress.org/rest-api/, https://developer.wordpress.org/rest-api/reference/, https://codex.wordpress.org/Widgets_API, https://codex.wordpress.org/Plugin_Resources, and https://codex.wordpress.org/Plugin_API .)

156. 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. (See, for example, https://developer.wordpress.org/rest-api/, https://developer.wordpress.org/rest-api/reference/, https://codex.wordpress.org/Widgets_API, https://codex.wordpress.org/Plugin_Resources, and https://codex.wordpress.org/Plugin_API .)

157. 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 WordPress 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. (See, for example, https://developer.wordpress.org/rest-api/, https://developer.wordpress.org/rest-api/reference/, https://codex.wordpress.org/Widgets_API, https://codex.wordpress.org/Plugin_Resources, and https://codex.wordpress.org/Plugin_API .)

158. The presence of the above referenced features is demonstrated, by way of example, by reference to publicly available information. Regarding WordPress, *see, e.g.,* http://themeforest.net/category/wordpress; http://codex.wordpress.org/Templates; http://codex.wordpress.org/Template_Hierarchy; http://codex.wordpress.org/Function_Reference/the_title; http://codex.wordpress.org/Function_Reference/the_content;

https://www.wpbeginner.com/glossary/database/; https://codex.wordpress.org/Pages; http://codex.wordpress.org/Templates; http://codex.wordpress.org/Template_Tags/get_the_title; and http://codex.wordpress.org/Query_Overview.

159. Claim 16 of the '287 patent recites a method of displaying content on a display that includes all the elements of claim 15, additionally where the registry includes definitions of input and output related to the web service.

160. The Accused Instrumentalities infringe claim 16 of the '287 patent through a combination of features which collectively practice each limitation of claim 16. By way of example, the Accused Instrumentalities include definitions of input and output related to a web service as based on their JSON formatting characteristics and the defined UI object functionality. See, e.g., https://developer.wordpress.org/rest-api/; https://developer.wordpress.org/rest-api/reference/; https://developer.wordpress.org/rest-api/reference/posts/ https://developer.wordpress.org/rest-api/reference/posts/#schema-title https://codex.wordpress.org/Widgets_API; https://codex.wordpress.org/Plugin_Resources; and https://codex.wordpress.org/Plugin_API; https://codex.wordpress.org/WordPress_Widgets.

161. Claim 17 of the '287 patent recites a method of displaying content on a display that includes all the elements of claim 15, additionally where the web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.

162. The Accused Instrumentalities infringe claim 17 of the '287 patent through a combination of features which collectively practice each limitation of claim 17. By way of example, the Accused Instrumentalities feature web components including web chat, Reuters RSS feed, Calendar image, and map image widgets.

163. Claim 18 of the '287 patent recites a method of displaying content on a display that includes all the elements of claim 15, additionally where the defined UI object is an input field for a chat.

164. The Accused Instrumentalities infringe claim 18 of the '287 patent through a combination of features which collectively practice each limitation of claim 18. By way of example, the Accused Instrumentalities additionally feature an UI object that is an input field for a chat.

165. Claim 19 of the '287 patent recites a method of displaying content on a display that includes all the elements of claim 15, additionally where the UI object is an input field for a web service.

166. The Accused Instrumentalities infringe claim 19 of the '287 patent through a combination of features which collectively practice each limitation of claim 19. By way of example, the Accused Instrumentalities additionally feature a UI object that is an input field for a web service.

167. Claim 25 of the '287 patent recites a method of displaying content on a display that includes all the elements of claim 15, additionally where the method incudes providing the Application and Player over the network.

168. The Accused Instrumentalities infringe claim 25 of the '287 patent through a combination of features which collectively practice each limitation of claim 25. WordPress sends all files over a network using a variety of databases in its technology stack including MySQL. These backend capabilities provided the code over a network. By way of example, data from the wp_options table for the website header and from the wp_posts table for the "WordPress Info" web page are extracted directly from the Bitnami WordPress server-side database using MySQL Workbench. The stored data in the wp_options table includes the website's url, the website's title (blogname), the website's tagline (blogdescription), and the active template (style sheet).

option_id	option_name	option_value
1	siteurl	http://localhost/wordpress
2	blogname	Express Mobile
3	blogdescription	SELECTED USERS OF WORDPRESS
44	template	the-fundamentals-of-graphic-design

The stored data in the wp_ posts table for the "WordPress Info" web page includes information corresponding to user selected settings such as, for example, the color red ("#ff0000") for "manages 22%". Other user selections shown in stored database data below include, for ex-ample, the image filename for the image (http://localhost/wordpress/wp-content/uploads/2013/03/icon21.png), the

image alignment (class="wp-image-24 aligncenter"), and a selected paragraph style (h3) for "Heading 3".

ID	post_content	post_title	post_type
23	WordPress is used by over 14.7% of Alexa Internet's "top 1 million" websites and as of August, 2011, believe it or not, manages 22% of all new websites. WordPress is currently the most popular blogging system in use on the Web. As of December 2011, WordPress version 3.0 had been downloaded over 65 million times. 	WordPress Info	page

WordPress's HTML, CSS, Java, and JSON coding capabilities further are shown, for example. by http://codex.wordpress.org/Templates, https://codex.wordpress.org/CSS , https://developer.wordpress.org/rest-api/ and https://codex.wordpress.org/Widgets_API.).

169. Claim 26 of the '287 patent recites a method of displaying content on a display that includes all the elements of claim 15, additionally where the UI object corresponds to a widget.

170. The Accused Instrumentalities infringe claim 26 of the '287 patent through a combination of features which collectively practice each limitation of claim 26. By way of example, the Accused Instrumentalities employ WordPress which includes widgets. (See, for example., https://codex.wordpress.org/Widgets_API.)

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

172. Since the date of the filing of the original Complaint in this action, Defendant's infringement of the '287 patent has been willful.

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

COUNT IV – INFRINGEMENT OF U.S. PATENT NO. 9,928,044

174. The allegations set forth in the foregoing paragraphs 1 through 173 are incorporated into this Fourth Claim for Relief.

175. The allegations set forth in the foregoing paragraphs 1 through 138 are incorporated into this Fourth Claim for Relief.

176. On March 27, 2018, U.S. Patent No. 9,928,044 ("the '044 patent"), entitled "*Systems and Methods for Programming Mobile Devices*," was duly and legally issued by the United States Patent and Trademark Office. A true and correct copy of the '044 patent is attached as Exhibit E.

177. The inventions of the '044 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 feature a computer memory and an authoring tool or Player configured to define a User Interface ("UI") object for display on the device, where the defined UI object corresponds to a web component and where 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. Additionally, the computer memory and the authoring tool or Player are configured to build an Application consisting of one or more web page views to provide for the display of at least a portion of one or more of the web pages. These features are exclusively implemented utilizing computer technology.

178. The claims of the '044 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 '044 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.

179. The claims of the '044 patent recite an invention that is 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 invention describes systems for use with devices with authoring tools or Players specific to each device and Applications that are device independent.

180. The technology claimed in the '044 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.

181. Accordingly, each claim of the '044 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.

182. Plaintiff is the assignee and owner of the right, title and interest in and to the '044 patent, including the right to assert all causes of action arising under the patents and the right to any remedies for infringement of them.

183. Upon information and belief, Defendant has and continues to directly infringe at least claims 1-5, 11, 12, 15-19, 25 and 26 of the '044 patent by a system which includes a computer memory 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 and where 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. Additionally, the computer memory and the authoring tool or Player are configured to build an Application consisting of one or more web page views to provide for the display of at least a portion of one or more of the web pages (the "Accused Instrumentalities"). The Accused Instrumentalities include platforms that enable the functionality described above and include but are not limited to, for example, WordPress. *See, e.g.*, https://www.happiestminds.com/services/technology-services/. For example, Defendant provides comprehensive implementation of a number of product solutions including, but not limited to the WordPress platform.

184. In particular, claim 1 of the '044 patent recites a system for generating code to provide content on a display of a device, the system comprising: computer memory storing: 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 where each symbolic name 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 computer memory selected from a group consisting of an input of the web service and an output of the web service, where each defined UI object is either: 1) selected by a user of the authoring tool; or 2) automatically selected by the system as the preferred UI object corresponding to the symbolic name of the web component selected by the user of the authoring tool, 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, store information representative of the defined UI object and related settings in a database; retrieve the information representative of the one or more the UI object settings stored in the database; and build an Application consisting of one or more web page views from at least a portion of the database utilizing at least one Player, where the Player utilizes information stored in the database to generate for the display of at least a portion of the one or more web pages, 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, the device provides the user provided one or more input values and corresponding input symbolic name to the web service, the web service utilizes the input symbolic name and the user provided one or more input values for generating one or more output values having an associated output symbolic name, and the Player receives the output symbolic name and corresponding one or more output values and provides instructions for the display of the device to present an output value in the defined UI object.

185. The Accused Instrumentalities infringe claim 1 of the ’044 patent through a combination of features which collectively practice each limitation of claim 1. By way of example, the Accused Instrumentalities feature a system for generating code to provide content on a display of a device. The system includes a WordPress server, which provides WordPress’s WYSIWYG visual

effects editor and a WordPress-compatible browser. WordPress's WYSIWYG visual effects editor generates code, such as JavaScript or HTML code for such as options for defining title, text, images, videos and paragraph styles, while the browser displays the resulting content as a WordPress webpage on a display of a device, such as a computer display.

186. For example, on information and belief, WordPress uses a variety of databases in its technology stack including MySQL. Data from the wp_options table for the website header and the data from the wp_posts table for the "WordPress Info" web page extracted directly from the Bitnami WordPress server-side database using MySQL Workbench. The stored data in the wp_options table includes the website's url, the website's title (blogname), the website's tagline (blogdescription), and the active template (style sheet).

187. The Accused Instrumentalities feature a computer memory provided by WordPress MySQL database functionality on the WordPress server. By way of example, WordPress utilizes JSON strings extensively as part of its API, which necessarily require servers and databases. (See, https://developer.wordpress.org/rest-api/.)

188. The computer memory stores 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 WordPress 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 as demonstrated by the formatting in conjunction with WordPress's WYSIWYG visual effects 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. WordPress' WYSIWYG visual effects editor includes elements for defining the layout for placement of the defined UI objects. Widgets, plug-ins and other elements correspond to the defined UI objects and are the product of the JSON formatting. (See, for example, https://codex.wordpress.org/WordPress_Lessons#Template_Files,

https://codex.wordpress.org/WordPress_Widgets, https://codex.wordpress.org/Plugin_Resources, and https://codex.wordpress.org/Plugin_API .)

189. Furthermore, 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 characteristics in conjunction with WordPress' WYSIWYG visual effects 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. WordPress' WYSIWYG visual effects editor includes elements for defining the layout for placement of the defined UI objects. Widgets, plug-ins and other elements correspond to the defined UI objects and are the product of the JSON formatting. (See, for example, https://codex.wordpress.org/WordPress_Lessons#Template_Files, https://codex.wordpress.org/WordPress_Widgets, https://codex.wordpress.org/Plugin_Resources, and https://codex.wordpress.org/Plugin_API .)

190. The computer memory also stores b) an address of the web service. Because WordPress contains web services, it contains the corresponding addresses for the web services. (See, for example, https://developer.wordpress.org/rest-api/, https://codex.wordpress.org/WordPress_Widgets, https://codex.wordpress.org/Plugin_Resources, and https://codex.wordpress.org/Plugin_API .)

191. The Accused Instrumentalities feature an authoring tool in the form of WordPress's WYSIWYG visual effects editor, widget, and plug-in authoring tools. (See, for example, https://codex.wordpress.org/WordPress_Widgets, https://codex.wordpress.org/Plugin_Resources, https://codex.wordpress.org/Plugin_API .)

192. 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 computer memory selected from a group consisting of an input of the web service and an output of the web service. WordPress's WYSIWYG visual effects editor and widget authoring tools define the presence of a UI object for presentation on a display and the defined UI object corresponds to a web component

included in the computer memory selected from a group consisting of an input of the web service and an output of the web service.

193. 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. WordPress'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. When a Widget is selected in the WordPress Widget selection list (See https://codex.wordpress.org/WordPress_Widgets) steps 1 to 5 under "Displaying Widgets" , the widget UI will automatically displayed in the Web Page Sidebar. (See, for example, https://codex.wordpress.org/WordPress_Widgets, https://codex.wordpress.org/Plugin_Resources, https://codex.wordpress.org/Plugin_API .)

194. The authoring tool is configured to access the computer memory to select the symbolic name corresponding to the web component of the defined UI object based on its JSON formatting characteristics.

195. The 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 name, 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 WordPress Editor makes a UI element request a JSON request is sent to the Server and a JSON data response is provided to the Interface. (See, for example, https://developer.wordpress.org/rest-api/, https://codex.wordpress.org/WordPress_Widgets, https://codex.wordpress.org/Plugin_Resources, https://codex.wordpress.org/Plugin_API.)

196. The authoring tool is configured to store information representative of the defined UI object and related settings in a database. For example, WordPress's computer memory is configured

to store information representative of defined UI objects. (See, for example, https://developer.wordpress.org/rest-api/ and https://codex.wordpress.org/Widgets_API.)

197. The authoring tool is also configured to retrieve the information representative of the one or more the UI object settings stored in the database based on the JSON strings. (See, for example, https://developer.wordpress.org/rest-api/, https://developer.wordpress.org/rest-api/reference/ , https://codex.wordpress.org/Plugin_Resources, https://codex.wordpress.org/Plugin_API , and https://codex.wordpress.org/Widgets_API.)

198. The WordPress authoring tool is configured to build an Application consisting of one or more web page views from at least a portion of the database utilizing at least one Player, where the Player utilizes information stored in the database to generate for the display of at least a portion of the one or more web pages. WordPress builds an Application including the symbolic name of the defined UI object. (See, for example, https://codex.wordpress.org/Widgets_API, https://codex.wordpress.org/Plugin_Resources , https://codex.wordpress.org/Plugin_API , https://developer.wordpress.org/rest-api/ , https://developer.wordpress.org/rest-api/reference/.) It is a feature of the Accused Instrumentalities that WordPress also contains at least one Player in the form of a runtime player, such that the Application and the Player are provided to the device and executed on the device. (See, for example, https://codex.wordpress.org/CSS , https://codex.wordpress.org/Using_Javascript and https://developer.wordpress.org/rest-api/)

199. The Accused Instrumentalities feature a system where the Application and Player are provided to the device and executed on the device.

200. When the user of the device provides one or more input values associated with an input symbolic name to an input of the defined UI object the device provides the user provided one or more input values and corresponding input symbolic name to the web service. Because the Accused Instrumentalities incorporate a system that includes WordPress, 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. (See, for example, https://developer.wordpress.org/rest-api/,

https://developer.wordpress.org/rest-api/reference/, https://codex.wordpress.org/Widgets_API, https://codex.wordpress.org/Plugin_Resources, and https://codex.wordpress.org/Plugin_API .)

201. 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. (See, for example, https://developer.wordpress.org/rest-api/, https://developer.wordpress.org/rest-api/reference/, https://codex.wordpress.org/Widgets_API, https://codex.wordpress.org/Plugin_Resources, and https://codex.wordpress.org/Plugin_API .)

202. 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 WordPress 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. (See, for example, https://developer.wordpress.org/rest-api/, https://developer.wordpress.org/rest-api/reference/, https://codex.wordpress.org/Widgets_API, https://codex.wordpress.org/Plugin_Resources, and https://codex.wordpress.org/Plugin_API .)

203. The presence of the above referenced features is demonstrated, by way of example, by reference to publicly available information. Regarding WordPress, *see, e.g.,* http://themeforest.net/category/wordpress; http://codex.wordpress.org/Templates; http://codex.wordpress.org/Template_Hierarchy; http://codex.wordpress.org/Function_Reference/the_title; http://codex.wordpress.org/Function_Reference/the_content; https://www.wpbeginner.com/glossary/database/; https://codex.wordpress.org/Pages; http://codex.wordpress.org/Templates; http://codex.wordpress.org/Template_Tags/get_the_title; and http://codex.wordpress.org/Query_Overview. .

204. Claim 2 of the ’044 patent recites a system for generating code to provide content on a display of a device that includes all the elements of claim 1, additionally where the system stores

information in a registry, and wherein the registry includes definitions of input and output related to the web service.

205. The Accused Instrumentalities infringe claim 2 of the '044 patent through a combination of features which collectively practice each limitation of claim 2. By way of example, the registry includes definitions of input and output related to a web service as evidenced by WordPress's JSON formatting characteristics of the defined UI objects. (See, for example, https://developer.wordpress.org/rest-api/; https://developer.wordpress.org/rest-api/reference/; https://developer.wordpress.org/rest-api/reference/posts/ https://developer.wordpress.org/rest-api/reference/posts/#schema-title https://codex.wordpress.org/Widgets_API; https://codex.wordpress.org/Plugin_Resources; and https://codex.wordpress.org/Plugin_API; https://codex.wordpress.org/WordPress_Widgets)

206. Claim 3 of the '044 patent recites a system for generating code to provide content on a display of a device that includes all the elements of claim 1, additionally where the web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.

207. The Accused Instrumentalities infringe claim 3 of the '044 patent through a combination of features which collectively practice each limitation of claim 3. By way of example, the Accused Instrumentalities feature web components additionally including web chat, Reuters RSS feed, Calendar image, and map image widgets.

208. Claim 4 of the '044 patent recites a system for generating code to provide content on a display of a device that includes all the elements of claim 1, additionally where the UI object is an input field for a chat.

209. The Accused Instrumentalities infringe claim 4 of the '044 patent through a combination of features which collectively practice each limitation of claim 4. By way of example, the Accused Instrumentalities additionally feature an UI object that is an input field for a chat.

210. Claim 5 of the '044 patent recites a system for generating code to provide content on a display of a device that includes all the elements of claim 1, additionally where the system stores information in a registry, and wherein the registry includes definitions of input and output related to the web service.

211. The Accused Instrumentalities infringe claim 5 of the '044 patent through a combination of features which collectively practice each limitation of claim 5. By way of example, the Accused Instrumentalities additionally feature a defined UI object that is an input field for a web service.

212. Claim 11 of the '044 patent recites a system for generating code to provide content on a display of a device that includes all the elements of claim 1, additionally where the code is provided over the network.

213. The Accused Instrumentalities infringe claim 11 of the '044 patent through a combination of features which collectively practice each limitation of claim 11. WordPress sends all files over a network using a variety of databases in its technology stack including MySQL. These backend capabilities provided the code over a network. By way of example, data from the wp_options table for the website header and from the wp_posts table for the "WordPress Info" web page are extracted directly from the Bitnami WordPress server-side database using MySQL Workbench. The stored data in the wp_options table includes the website's url, the website's title (blogname), the website's tagline (blogdescription), and the active template (style sheet).

option_id	option_name	option_value
1	siteurl	http://localhost/wordpress
2	blogname	Express Mobile
3	blogdescription	SELECTED USERS OF WORDPRESS
44	template	the-fundamentals-of-graphic-design

The stored data in the wp_ posts table for the "WordPress Info" web page includes information corresponding to user selected settings such as, for example, the color red ("#ff0000") for "manages 22%". Other user selections shown in stored database data below include, for ex-ample, the image filename for the image (http://localhost/wordpress/wp-content/uploads/2013/03/icon21.png), the image alignment (class="wp-image-24 aligncenter"), and a selected paragraph style (h3) for "Heading 3".

ID	post_content	post_title	post_type
23	WordPress is used by over 14.7% of Alexa Internet's "top 1 million" websites and as of August, 2011, believe it or not, manages 22% of all new websites. WordPress is currently the most popular blogging system in use on the Web. As of December 2011, WordPress version 3.0 had been downloaded over 65 million times. 	WordPress Info	page

214. WordPress's HTML, CSS, Java, and JSON coding capabilities further are shown, for example. by http://codex.wordpress.org/Templates, https://codex.wordpress.org/CSS , https://developer.wordpress.org/rest-api/ and https://codex.wordpress.org/Widgets_API.).

215. Claim 12 of the '287 patent recites a system for generating code to provide content on a display of a device that includes all the elements of claim 1, additionally where the defined UI object corresponds to a widget.

216. The Accused Instrumentalities infringe claim 12 of the '044 patent through a combination of features which collectively practice each limitation of claim 12. By way of example, the Accused instrumentalities' WordPress includes widgets. . See, e.g., https://codex.wordpress.org/Widgets.

217. The Accused Instrumentalities infringe claim 15 of the '044 patent through a combination of features which collectively practice each limitation of claim 15. By way of example, the Accused Instrumentalities feature a method of displaying content on a display of a device having a Player, in the form of a runtime player and a non-volatile computer memory storing the WordPress MySQL database functionality on the device. The non-volatile computer memory stores 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 WordPress 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 as demonstrated by the formatting in conjunction with WordPress's WYSIWYG visual effects editor, widget, and plugin authoring tools.

218. Furthermore, 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 WordPress' WYSIWYG visual effects editor and widget authoring tools.

219. The computer memory also stores an address of the web service. Because WordPress contains web services, both as a library of Widgets (see https://codex.wordpress.org/Widgets_API) and Plug-ins (see https://codex.wordpress.org/Plugin_API), it contains the corresponding addresses of the web services. All web services are represented as a wsdl (Web Service Description Language) and wsdls have URLs to point to the internet location that receives the web service's inputs and returns the web service's outputs (See https://www.soapui.org/soap-and-wsdl/working-with-wsdls.html).

220. The Accused Instrumentalities include defining a UI object for presentation on the display, where the UI object corresponds to a web component included in the computer memory, where the web component is selected from a group consisting of an input of a web service and an output of the web service. WordPress defines a user interface object, *i.e.*, an element/UI component, 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 (as evidenced by JSON data formatting)..

221. 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. When a Widget is selected in the

WordPress Widget selection list (See https://codex.wordpress.org/WordPress_Widgets) steps 1 to 5 under "Displaying Widgets" , the widget UI will automatically displayed in the Web Page Sidebar. WordPress contains an authoring tool in the form of the WYSIWYG visual effects editor, widgets, and plug-in authoring tools.

222. The Accused Instrumentalities include selecting the symbolic name corresponding to the web component (i.e. WordPress Widget or Plug-in) of the defined UI object and associating 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. WordPress accesses the non-volatile 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).

223. The Accused Instrumentalities also include retrieving the information representative of the one or more the UI object settings stored in the database based on the JSON formatting characteristics.

224. The Accused Instrumentalities include building an Application consisting of one or more web page views from at least a portion of the database utilizing the Player, where the Player utilizes information stored in the database to generate for the display of at least a portion of the one or more web pages. WordPress builds an Application included in the symbolic name of the defined UI object.

225. With the Accused Instrumentalities when the Application and Player are provided to the device and executed on the device 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, and 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. For example, in WordPress, a user of a device provides an input value associated with an input symbolic name to an input of a defined UI object, such as utilizing an element, plug-in, or widget. The element, plug-in, or widget is associated with symbolic name based on their JSON formatting characteristics. 1) The element input value corresponds to the input symbolic name based on its JSON formatting characteristics 2) The element/UI component output value corresponds to the output symbolic name via JSON. 3) The runtime player within WordPress receives the output name, output value, and provides instructions for a display as shown by the fact that the defied UI object is displayed.

226. The presence of the above referenced features is demonstrated, by way of example, by reference to publicly available information. Regarding WordPress, see, e.g., http://themeforest.net/category/wordpress; http://codex.wordpress.org/Templates; http://codex.wordpress.org/Template_Hierarchy; http://codex.wordpress.org/Function_Reference/the_title; http://codex.wordpress.org/Function_Reference/the_content; https://www.wpbeginner.com/glossary/database/; https://codex.wordpress.org/Pagesl http://codex.wordpress.org/Templates; http://codex.wordpress.org/Template_Tags/get_the_title; and http://codex.wordpress.org/Query_Overview.

227. Claim 16 of the ’044 patent recites a method of displaying content on a display that includes all the elements of claim 15, additionally where the method stores information in a registry, and the registry includes definitions of input and output related to the web service.

228. The Accused Instrumentalities infringe claim 16 of the ’044 patent through a combination of features which collectively practice each limitation of claim 16. By way of example, the Accused Instrumentalities feature storing information in a WordPress registry that includes definitions of input and output related to a web service as evidenced by WordPress’s JSON formatting characteristics and the UI object functionality. See, e.g., https://developer.wordpress.org/rest-api/; https://developer.wordpress.org/rest-api/reference/;

https://developer.wordpress.org/rest-api/reference/posts/ https://developer.wordpress.org/rest-api/reference/posts/#schema-title https://codex.wordpress.org/Widgets_API; https://codex.wordpress.org/Plugin_Resources; and https://codex.wordpress.org/Plugin_API; https://codex.wordpress.org/WordPress_Widgets.

229. Claim 17 of the ’044 patent recites a method of displaying content on a display that includes all the elements of claim 15, additionally where the method additionally where the web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.

230. The Accused Instrumentalities infringe claim 17 of the ’044 patent through a combination of features which collectively practice each limitation of claim 17. By way of example, the Accused Instrumentalities feature web components additionally including web chat, Reuters RSS feed, Calendar image, and map image widgets.

231. Claim 18 of the ’044 patent recites a method of displaying content on a display that includes all the elements of claim 15, additionally where the UI object is an input field for a chat.

232. The Accused Instrumentalities infringe claim 18 of the ’044 patent through a combination of features which collectively practice each limitation of claim 18. By way of example, the Accused Instrumentalities additionally feature an UI object that is an input field for a chat.

233. Claim 19 of the ’044 patent recites a method of displaying content on a display that includes all the elements of claim 15, additionally where the UI object is an input field for a web service.

234. The Accused Instrumentalities infringe claim 19 of the ’044 patent through a combination of features which collectively practice each limitation of claim 19. By way of example, the Accused Instrumentalities additionally feature a UI object that is an input field for a web service.

235. Claim 25 of the ’044 patent recites a method of displaying content on a display that includes all the elements of claim 15, additionally where the method incudes providing the code over the network.

236. The Accused Instrumentalities infringe claim 25 of the ’044 patent through a combination of features which collectively practice each limitation of claim 25. By way of example,

WordPress uses a variety of databases including MySQL. These backend capabilities demonstrate that the code is provided to a user over a network. See, e.g., http://codex.wordpress.org/Templates.

237. Claim 26 of the '044 patent recites a method of displaying content on a display that includes all the elements of claim 15, additionally where the UI object corresponds to a widget.

238. The Accused Instrumentalities infringe claim 26 of the '044 patent through a combination of features which collectively practice each limitation of claim 26. By way of example, the Accused Instrumentalities employ WordPress which includes widgets. See, e.g., https://codex.wordpress.org/Widgets_API.

239. Defendant was made aware of the '044 patent and its infringement thereof at least as early as the filing of the original Complaint in this action.

240. Since the date of the filing of the original Complaint in this action, Defendant's infringement of the '044 patent has been willful.

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

242. Defendant was made aware of the '044 patent and its infringement thereof at least as early as the filing of the original Complaint in this action.

243. Since the date of the filing of the original Complaint in this action, Defendant's infringement of the '044 patent has been willful.

244. 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 '044 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 '044 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: October 25, 2018

DEVLIN LAW FIRM LLC

By: */s/ Jeffrey Francis Craft*

Jeffrey Francis Craft
jcraft@devlinlawfirm.com
California Bar No. 147186
2069 Cold Canyon Road
Calabasas, California 91302

Robert Kiddie (*pro hac vice* to be filed)
Texas Bar No. 24060092
rkiddie@devlinlawfirm.com
1306 N. Broom St., 1st Floor
Wilmington, Delaware 19806
Telephone: (302) 449-9010
Facsimile: (302) 353-4251

Attorneys for Plaintiff Express Mobile, Inc.