IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

SHOPIFY INC. AND SHOPIFY
(USA), INC.

Plaintiffs/Counterclaim
Defendants,

v.

EXPRESS MOBILE, INC.

Defendant/Counterclaim
Plaintiff.

Civil Action No. 1:19-cv-00439-RGA

JURY TRIAL DEMANDED

DEFENDANT EXPRESS MOBILE, INC.'S FIRST AMENDED ANSWER, AFFIRMATIVE DEFENSES, AND COUNTERCLAIMS TO PLAINTIFFS' COMPLAINT

Defendant Express Mobile, Inc. ("Defendant") hereby provides Defendant's First Amended Answer and Counterclaims to Plaintiffs' Complaint for Declaratory Judgement (the "Complaint") as follows:

NATURE OF THE ACTION

1. Admitted.

THE PARTIES

2. Defendant lacks knowledge or information sufficient to form a belief as to the truth of the allegations of Paragraph 2, and therefore denies the same.

3. Defendant lacks knowledge or information sufficient to form a belief as to the truth of the allegations of Paragraph 2, and therefore denies the same.

4. Admitted.

5. Admitted.

JURISDICTION AND VENUE

6. This paragraph contains legal conclusions to which no response is required. To

the extent that a response is required, Defendant admits that the Complaint purports to state action that arises under the Declaratory Judgment Act, 28 U.S.C. §§ 2201–2202, and under the patent laws of the United States, 35 U.S.C. §§ 1 et seq.

7. This paragraph contains legal conclusions to which no response is required. To the extent that a response is required, Defendant admits that the Court has subject matter jurisdiction over this action under 28 U.S.C. §§1331, 1338(a), and 2201(a).

8. This paragraph contains legal conclusions to which no response is required. To the extent that a response is required, Defendant admits that an immediate, real, and justiciable controversy exists between Shopify and Express Mobile as to whether Shopify is infringing or has infringed the Patents-in-Suit. The remaining allegations of paragraph 8 are denied.

9. This paragraph contains legal conclusions to which no response is required. To the extent that a response is required, Defendant admits that Express Mobile is subject to general personal jurisdiction in this district because it is a Delaware corporation and thus resides and is at home in the District of Delaware.

10. This paragraph contains legal conclusions to which no response is required. To the extent that a response is required, Defendant admits that venue is proper in this Court pursuant to 28 U.S.C. § 1391(b)–(c).

THE PATENTS-IN-SUIT

11. Admitted.

12. Admitted.

13. Admitted.

14. Admitted.

15. Admitted.

16. Admitted.

17. Admitted.

18. Admitted.

EXPRESS MOBILE'S ENFORCEMENT OF THE DECLARATORY JUDGMENT PATENTS AND THREATS AGAINST SHOPIFY

19. Defendant does not understand what is meant by "industry-wide litigation campaign" and, therefore, denies same. Defendant admits the remaining allegations of Par. 19.

20. Admitted.

21. Admitted.

22. Admitted.

23. Admitted.

24. Admitted.

25. Admitted.

26. Admitted.

COUNT I

DECLARATORY JUDGMENT OF NON-INFRINGEMENT OF THE '397 PATENT

27. The allegations set forth in the foregoing paragraphs 1-26 are hereby realleged and incorporated herein by reference.

28. Admitted.

29. Denied.

30. Denied.

31. Admitted.

32. Admitted.

COUNT II

DECLARATORY JUDGMENT OF NON-INFRINGEMENT OF THE ’168 PATENT

33. The allegations set forth in the foregoing paragraphs 1-32 are hereby realleged and incorporated herein by reference.

34. Admitted.

35. Denied.

36. Denied.

37. Denied.

38. Admitted.

39. Admitted.

COUNT III

DECLARATORY JUDGMENT OF NON-INFRINGEMENT OF THE ’755 PATENT

40. The allegations set forth in the foregoing paragraphs 1-39 are hereby realleged and incorporated herein by reference.

41. Admitted.

42. Denied.

43. Denied.

44. Admitted.

45. Admitted.

COUNT IV

DECLARATORY JUDGMENT OF NON-INFRINGEMENT OF THE ’287 PATENT

46. The allegations set forth in the foregoing paragraphs 1-45 are hereby realleged and incorporated herein by reference.

47. Admitted.

48. Denied.

49. Denied.

50. Admitted.

51. Admitted.

PRAYER FOR RELIEF

Defendant denies any and all remaining allegations not specifically admitted and denies that Plaintiffs are entitled to the relief requested in paragraphs a through f of their Request for Relief, or to any relief at all. To the extent that any statement in Plaintiffs' Request for Relief is considered to contain factual allegations that require a response, Defendant denies each and every such allegation. Defendant denies each and every allegation of non-infringement as alleged in the Complaint.

JURY DEMAND

This paragraph contains a statement to which no response is required. To the extent that a response is required, Defendant admits that Plaintiffs demand a trial by jury on all triable issued raised in the Complaint.

AFFIRMATIVE DEFENSES

Defendant asserts the following affirmative defenses. Defendant reserves the right to assert additional affirmative defenses as they become known through discovery.

FIRST AFFIRMATIVE DEFENSE

Plaintiffs' Complaint fails to state a claim upon which relief can be granted.

SECOND AFFIRMATIVE DEFENSE

Defendant reserves the right to assert any other legal or equitable defenses to which it is entitled.

COUNTERCLAIMS

Counterclaim-Plaintiff/Defendant Express Mobile, Inc. ("Express Mobile") alleges counterclaims against Counterclaim-Defendants/Plaintiffs Shopify, Inc. and Shopify (USA), Inc.'s (collectively "Plaintiffs'") as follows:

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. Express Mobile is a corporation organized under the laws of the State of Delaware with a place of business at38 Washington Street, Novato CA 94947.

3. Plaintiff Shopify Inc. alleges it is a corporation organized and existing under the laws of Canada, with a principal place of business at 150 Elgin St., 8th Floor, Ottawa, Ontario, Canada.

4. Plaintiff Shopify (USA), Inc. is a corporation organized and existing under the laws of Delaware, with a place of business at 33 New Montgomery St., Suite 750, San Francisco, California. Shopify (USA), Inc. is a wholly-owned subsidiary of Shopify, Inc. Shopify (USA), Inc. was formed in 2016 and based on the acquisition of a company known as Kit, which developed and still maintains a marketing chatbot.

JURISDICTION AND VENUE

5. This Court has jurisdiction over these Counterclaims under 28 U.S.C. §§ 1331 1338(a), 2201, and 2202.

6. Plaintiffs are subject to personal jurisdiction in this district arising out of its purposeful acts and/or transactions directed toward this district, including without limitation Plaintiffs' bringing of this declaratory judgment action in this district.

7. These Counterclaims for patent infringement arise under the Patent Laws of the United States, Title 35 of the United States Code.

8. Venue is proper in this judicial district under 28 U.S.C. §1400(b).

FIRST COUNTERCLAIM – INFRINGEMENT OF U.S. PATENT NO. 6,546,397

9. The allegations set forth in the foregoing paragraphs 1 through 8 of these Counterclaims are incorporated into this First Counterclaim 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 In Case No. 3:18-CV-04679-RS, a case filed in the Northern District of California, the defendant in that action, Code and Theory LLC. brought a Motion to Dismiss Plaintiff’s Complaint asserting that the ’397 patent, along with U.S. Patent No. 7,594,168 (asserted in Count II below) ae not subject matter eligible under 35 U.S.C. § 101 as a matter of law. (Case No. 3:18-CV-04679-RS Dkt.35) Subsequent briefing included Plaintiff Express Mobile, Inc.’s Opposition to Defendant Code and Theory LLC’s Motion to Dismiss Plaintiff’s Complaint (Case No. 3:18-CV-04679-RS Dkt.40), and Motion to Dismiss Plaintiff’s Complaint [sic] (Case No. 3:18-CV-04679-RS Dkt.41), Each of those filings is incorporated by reference into this Complaint.

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 E.) No objection was filed to the Magistrate Judge's report and recommendation and the decision therefore became final.

18. In Case No. 3:18-CV-04688-RS, a case filed in the Northern District of California, the defendant in that action, Pantheon Systems, Inc. brought a Motion to Dismiss Counts I and II of Plaintiff's First Amended Complaint asserting that the '397 patent, along with U.S. Patent No. 7,594,168 (asserted in Count II below) were directed to the abstract idea of creating and displaying webpages based upon information from a user with no further inventive concept and purportedly ineligible for patenting under 35 U.S.C. § 101 ("§ 101"). (Case No. 3:18-CV-04688-RS Dkt.26) Subsequent briefing included Plaintiff's Answering Brief in Opposition of Defendant's Motion to Dismiss (Case No. 3:18-CV-04688-RS Dkt.32), and Reply in Support of Defendant's Motion to Dismiss Counts I and II of Plaintiff's First Amended Complaint (Case No. 3:18-CV-04688-RS Dkt.34), Each of those filings is incorporated by reference into this Complaint.

19. The Court heard the motions together and after a consideration of the respective pleadings, Hon. Richard Seeborg denied both motions holding "it simply cannot be said on the present record that the claims are drawn so broadly as to be divorced from the potentially patent-

eligible purported technological improvements described in the specification" and further stating "The patents here are directed at a purportedly revolutionary technological solution to a technological problem—how to create webpages for the internet in a manner that permits "what you see is what you get" editing, and a number of other alleged improvements over the then-existing methodologies." (Case No. 3:18-CV-04679-RS Dkt.45; Case No. 3:18-CV-04688-RS Dkt.40; attached hereto as Exhibit F.)

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

21. Upon information and belief, Plaintiffs have and continue to directly infringe at least claims 1, 2 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 by Plaintiffs, such as, for example, all versions of Plaintiffs' eCommerce software.

22. On information and belief, Plaintiffs are for-profit organizations. Shopify, Inc. has revenues of approximately one billion U.S.D. per year. Moreover, Plaintiffs, their employees and/or agents utilize the Accused Instrumentalities in the building of websites for Plaintiffs' customers, leading to direct or indirect revenues and profit. As one example of indirect profit, entities such as Plaintiffs will frequently offer website building 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, Plaintiffs would be at a disadvantage in the marketplace and would generate less revenue overall.

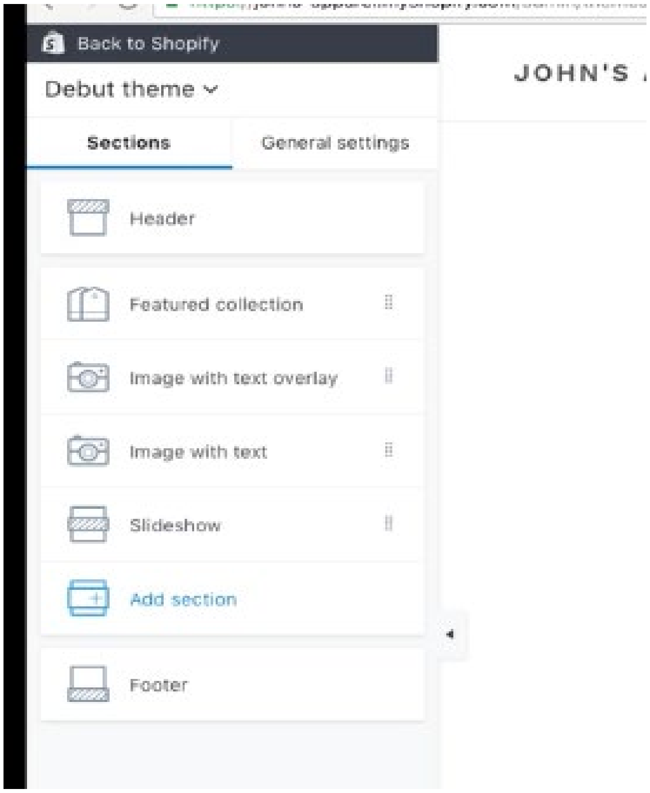
23. In particular, claim 1 of the ’397 patent generally recites a method enabling production of websites on and for computers with browsers and virtual machines, *i.e.*, an abstract machine that is not built in hardware but is emulated in software, such as a Java Virtual Machine. *See, e.g.*, *X Commerce, Inc. v Express Mobile, Inc.*, Case No 17-cv-02605-RS, NDCA, DKT 79-5; http://developer.telerik.com/featured/a-guide-to-javascript-engines-for-idiots/; http://dictionary.reference.com/browse/virtual+machine?s=t.

24. The method includes 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, *i.e.*, one or more file, including a run time engine, that are downloaded or created when a browser is pointed to a web page or website, 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). *See, e.g.*, *X Commerce, Inc. v Express Mobile, Inc.*, Case No 17-cv-02605-RS, NDCA, DKT 79-8.

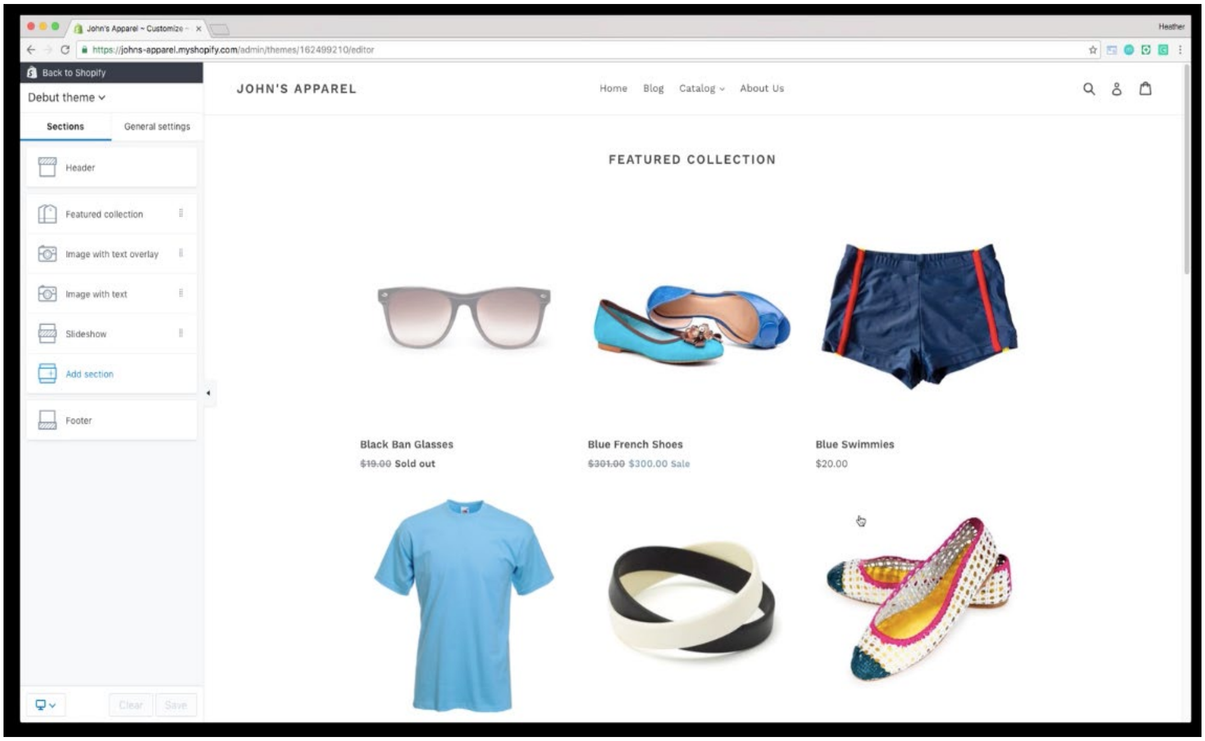
25. The Accused Instrumentalities infringe claim 1 of the ’397 patent through a combination of features which collectively practice each limitation of claim 1. By way of example, modern internet browsers such as Microsoft Internet Explorer, Mozilla’s Firefox, Apple Safari, Google Chrome, and Opera include virtual machines capable of generating a display.

26. The Accused Instrumentalities support the use of browsers including Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Edge, Opera, Apple Safari for iOS and Google Chrome for Android, (*See* https://help.shopify.com/manual/intro-to-shopify/shopify-admin/supported-browsers.) 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.

27. 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, the Shopify Theme Editor (i.e. website building tools used by Plaintiffs), shown below, presents through a browser on a computer a viewable menu having a user selectable panel of settings describing elements on a Website and the Shopify Theme Editor is adapted to accept one or more selectable settings in the panels.



28. A display is generated in accordance with one or more user selected settings substantially contemporaneously with the selection thereof. A browser displayed the Shopify site that is shown below, where the elements are added on the fly.



29. The display is generated, for example, using a visual editing tool through a browser. The Shopify Theme Editor 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 Shopify Theme Editor, the display immediately updates to reflect the selected option.

30. Data is stored including information corresponding to user selected settings such as, for example, the selections of text color. Other user selections are also stored in including, for example, the layout, image filenames, thumbnails, and paragraph margin settings for defining the alignment of an image location.

31. 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, the at least one run time file generates 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, which includes an abstract machine that is not built in hardware but is emulated in software, in order to render the page HTML.

33. 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.

34. Additionally, the "Ruby code," including the Ruby template files, can be viewed in the file directory for the Accused Instrumentalities, and this directory includes various other run time files (including other Ruby files, JavaScript files, PHTML, and 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.

35. The presence of the above referenced elements is demonstrated, by way of example, by reference to publicly available information. S*ee*, e.g., https://www.shopify.com/website/hosting; https://help.shopify.com/api/storefront-api; https://help.shopify.com/api/reference; https://help.shopify.com/api/reference/page; https://help.shopify.com/api/storefront-api; https://help.shopify.com/api/reference/product; https://help.shopify.com/manual/intro-to-shopify/shopify-admin/supported-browsers;

https://help.shopify.com/manual/using-themes/change-the-layout; https://help.shopify.com/manual/using-themes/change-the-layout/using-theme-presets; https://help.shopify.com/manual/intro-to-shopify/shopify-admin/supported-browsers; https://help.shopify.com/manual/products/product-variant-images; https://help.shopify.com/manual/using-themes/troubleshooting/fix-64-kilobyte-limit-errors; http://techstacks.io/shopify; https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html; https://www.slideshare.net/jduff/how-shopify-scales-rails-20443485; https://help.shopify.com/manual/intro-to-shopify/pricing-plans/plan-features; https://help.shopify.com/manual/intro-to-shopify/pricing-plans/plan-features; https://help.shopify.com/api/sdks/shopify-apps/modifying-online-store/use-javascript-responsibly; https://help.shopify.com/themes/development/getting-started/choosing-an-editor; https://help.shopify.com/manual/products/product-variant-images; https://engineering.shopify.com/blogs/engineering/e-commerce-at-scale-inside-shopifys-tech-stack; https://engineering.shopify.com/blogs/engineering/e-commerce-at-scale-inside-shopifys-tech-stack .

36. Claim 2 of the ’397 patent generally recites an apparatus for producing websites on and for computers having a browser and a virtual machine; *i.e.*, an abstract machine that is not built in hardware but is emulated in software, such as a Java Virtual Machine. *See, e.g.*, *X Commerce, Inc v Express Mobile, Inc.*, Case No 17-cv-02605-RS, NDCA, DKT 79-5; http://developer.telerik.com/featured/a-guide-to-javascript-engines-for-idiots/; http://dictionary.reference.com/browse/virtual+machine?s=t.

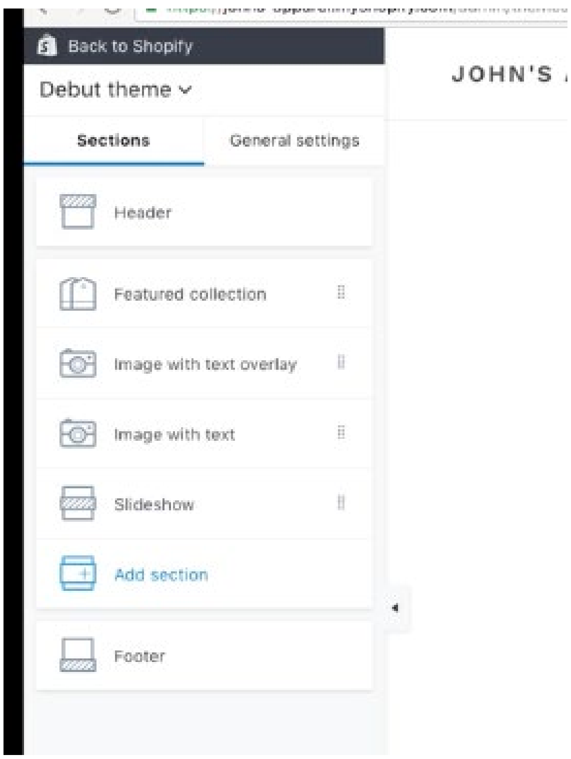
37. The apparatus comprises an interface to present a settings menu which describes elements, said panel presented through a browser, where the selectable setting(s) corresponds to

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).

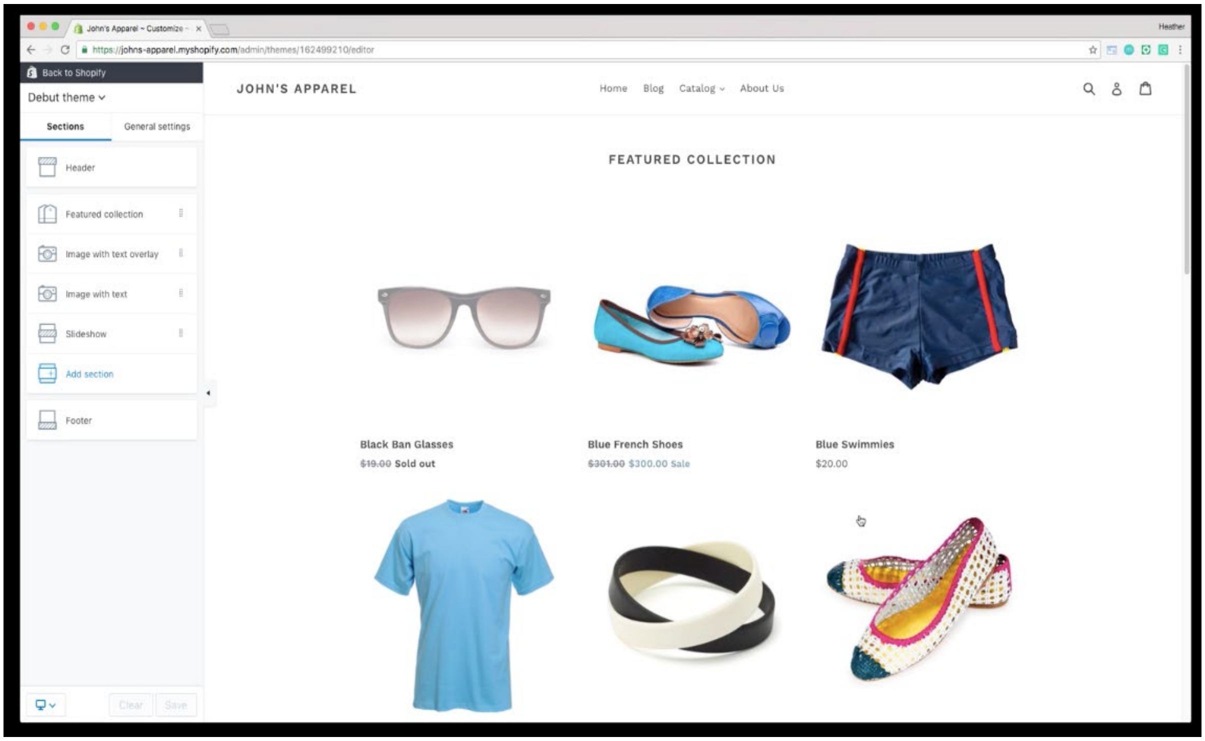
38. The Accused Instrumentalities infringe claim 2 of the '397 patent through a combination of features which collectively practice each limitation of claim 2. By way of example, modern internet browsers such as Microsoft Internet Explorer, Mozilla's Firefox, Apple Safari, Google Chrome, and Opera include virtual machines capable of generating a display.

39. The Accused Instrumentalities support the use of browsers including Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Edge, Opera, Apple Safari for iOS and Google Chrome for Android, (*See* https://help.shopify.com/manual/intro-to-shopify/shopify-admin/supported-browsers.) 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.

40. 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, the Shopify Theme Editor, shown below, presents through a browser on a computer a viewable menu having a user selectable panel of settings describing elements on a Website and the Shopify Theme Editor is adapted to accept one or more selectable settings in the panels.



41. A display is generated in accordance with one or more user selected settings substantially contemporaneously with the selection thereof. A browser displayed the Shopify site that is shown below, where the elements are added on the fly.



42. The display is generated, for example, using a visual editing tool through a browser. The Shopify Theme Editor 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 Shopify Theme Editor, the display immediately updates to reflect the selected option.

43. Data is stored including information corresponding to user selected settings such as, for example, the selections of text color. Other user selections are also stored in including, for example, the layout, image filenames, thumbnails, and paragraph margin settings for defining the alignment of an image location.

44. 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.

45. 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 an abstract machine that is not built in hardware but is emulated in software, in order to render the page.

46. 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.

47. Additionally, the "Ruby code," including the Ruby template files, can be viewed in the file directory for the Accused Instrumentalities, and this directory includes various other run time files (including other Ruby files, JavaScript files, PHTML, and 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.

48. The presence of the above referenced elements is demonstrated, by way of example, by reference to publicly available information. S*ee*, e.g., https://www.shopify.com/website/hosting; https://help.shopify.com/api/storefront-api; https://help.shopify.com/api/reference; https://help.shopify.com/api/reference/page; https://help.shopify.com/api/storefront-api; https://help.shopify.com/api/reference/product; https://help.shopify.com/manual/intro-to-shopify/shopify-admin/supported-browsers; https://help.shopify.com/manual/using-themes/change-the-layout; https://help.shopify.com/manual/using-themes/change-the-layout/using-theme-presets; https://help.shopify.com/manual/intro-to-shopify/shopify-admin/supported-browsers; https://help.shopify.com/manual/products/product-variant-images;

https://help.shopify.com/manual/using-themes/troubleshooting/fix-64-kilobyte-limit-errors; http://techstacks.io/shopify; https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html; https://www.slideshare.net/jduff/how-shopify-scales-rails-20443485; https://help.shopify.com/manual/intro-to-shopify/pricing-plans/plan-features; https://help.shopify.com/manual/intro-to-shopify/pricing-plans/plan-features; https://help.shopify.com/api/sdks/shopify-apps/modifying-online-store/use-javascript-responsibly; https://help.shopify.com/themes/development/getting-started/choosing-an-editor; https://help.shopify.com/manual/products/product-variant-images.

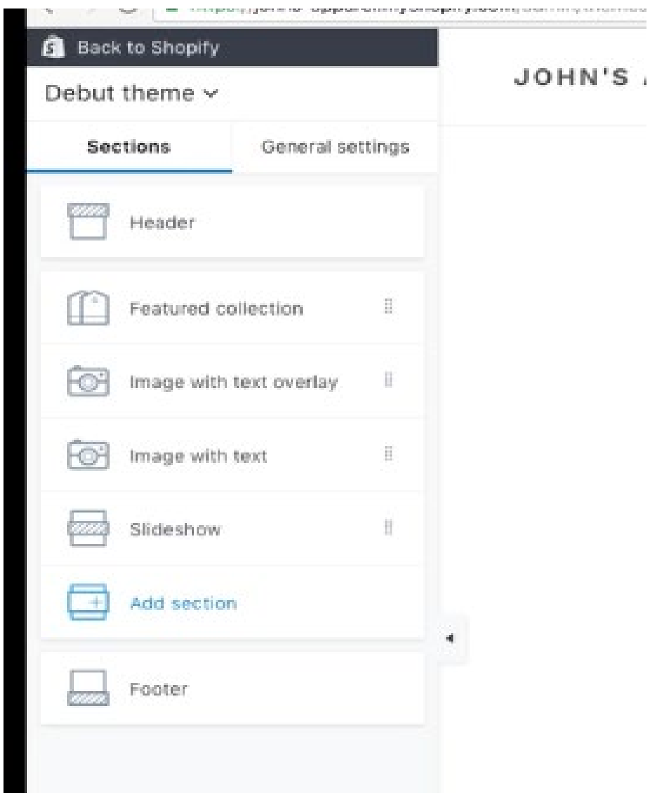
49. Claim 37 of the ’397 patent generally recites [a]n apparatus for producing websites with web page(s) on and for a computer with a browser and a virtual machine, *i.e.*, an abstract machine that is not built in hardware but is emulated in software, such as a Java Virtual Machine. *See, e.g.*, *X Commerce, Inc v Express Mobile, Inc.*, Case No 17-cv-02605-RS, NDCA, DKT 79-5; http://developer.telerik.com/featured/a-guide-to-javascript-engines-for-idiots/; http://dictionary.reference.com/browse/virtual+machine?s=t; https://engineering.shopify.com/blogs/engineering/e-commerce-at-scale-inside-shopifys-tech-stack; https://engineering.shopify.com/blogs/engineering/e-commerce-at-scale-inside-shopifys-tech-stack.

50. The apparatus comprises: 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.

51. 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 capable of generating a display.

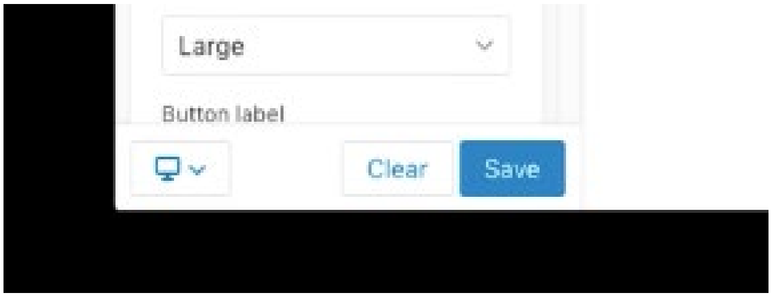
52. The Accused Instrumentalities support the use of browsers including Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Edge, Opera, Apple Safari for iOS and Google Chrome for Android, (*See* https://help.shopify.com/manual/intro-to-shopify/shopify-admin/supported-browsers.) All of these browsers rely on browser engines comprising virtual machines to interpret and execute JavaScript and HTML to render web pages on a computer. By way of further example, the Accused Instrumentalities enables a user to produce a website through a browser on the user’s computer interacting with the server facility hosted by Shopify. For example, the Shopify Theme Editor, shown below, shows an interface operable through a browser on a computer configured for building a Web site through control of the website elements. The interface is operable to present a viewable menu having a user selectable panel of settings describing elements on a Website and to accept multiple user selectable panel of settings.



53. The interface is operable to generate the display in accordance with the assembly of settings contemporaneously with the acceptance thereof. The display is generated, 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.

54. On information and belief, the Accused Instrumentalities further rely on at least one of the user selectable settings of the panel of settings being operable to generate said display through commands to said virtual machine. For example, the browser engine's component JavaScript engine is operable to either generate the display directly, or generate HTML to be executed for display.

55. The Accused Instrumentalities include an internal database associated with the interface for storing information representative of one or more of said assembly of settings for controlling elements of the website. As shown below the Accused Instrumentalities feature a "Save" command that will save the internal database settings to an external database.



Furthermore, the Accused Instrumentalities update the internal database as part of their architecture and its drag and drop interface.

56. The Accused Instrumentalities include a build tool to construct one or more web pages of said Web. In particular, the Accused Instrumentalities build webpages with the Shopify Theme Editor.

57. The build tool includes an external database containing data corresponding to the information stored in said internal database. Shopify stores the internal database settings into as external database using the "Save" feature, in an external database.

58. The build tool additionally includes one or more run time files. 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.

59. Additionally, the "Ruby code," including the Ruby template files, can be viewed in the file directory for the Accused Instrumentalities, and this directory includes various other run time files (including other Ruby files, JavaScript files, PHTML, and 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.

60. The presence of the above referenced elements is demonstrated, by way of example, by reference to publicly available information. S*ee,* e.g., https://www.shopify.com/website/hosting; https://help.shopify.com/api/storefront-api; https://help.shopify.com/api/reference; https://help.shopify.com/api/reference/page; https://help.shopify.com/api/storefront-api; https://help.shopify.com/api/reference/product; https://help.shopify.com/manual/intro-to-shopify/shopify-admin/supported-browsers; https://help.shopify.com/manual/using-themes/change-the-layout; https://help.shopify.com/manual/using-themes/change-the-layout/using-theme-presets; https://help.shopify.com/manual/intro-to-shopify/shopify-admin/supported-browsers; https://help.shopify.com/manual/products/product-variant-images; https://help.shopify.com/manual/using-themes/troubleshooting/fix-64-kilobyte-limit-errors; http://techstacks.io/shopify; https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html; https://www.slideshare.net/jduff/how-shopify-scales-rails-20443485; https://help.shopify.com/manual/intro-to-shopify/pricing-plans/plan-features; https://help.shopify.com/manual/intro-to-shopify/pricing-plans/plan-features; https://help.shopify.com/api/sdks/shopify-apps/modifying-online-store/use-javascript-responsibly; https://help.shopify.com/themes/development/getting-started/choosing-an-editor; https://help.shopify.com/manual/products/product-variant-images; https://engineering.shopify.com/blogs/engineering/e-commerce-at-scale-inside-shopifys-tech-stack; https://engineering.shopify.com/blogs/engineering/e-commerce-at-scale-inside-shopifys-tech-stack.

61. On information and belief, Accused Instrumentalities are used, marketed, provided to, and/or used by or for Plaintiffs' partners, clients, customers and end users across the country and in this district.

62. On information and belief, Plaintiffs were made aware of the '397 patent and its infringement thereof at least as early as the filing by Express Mobile of the first patent infringement suits against Plaintiffs' Solution Partners on January 15, 2016.

63. Upon information and belief, since at least the time Plaintiffs received notice, Plaintiffs have induced and continues to induce others to infringe at least one claim of the '397 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Plaintiffs' partners, clients, customers, and end users, whose use of the Accused Instrumentalities constitutes direct infringement of at least one claim of the '397 patent.

64. In particular, Plaintiffs' actions that aid and abet others such as customers, clients, partners, developers, and end users to infringe include advertising and distributing the Accused Instrumentalities and providing instruction materials, training, and services regarding the Accused Instrumentalities. Plaintiffs actively encourage the adoption of the Accused Instrumentalities and provide multiple support sites for the vast network of developers working with the Accused Instrumentalities and also offer courses for training and certification courses to Plaintiffs' customers, partners, and developers. (*See, e.g.,* https://community. shopify.com/; https://shopify.com/support; https://u. shopify.com/about.). On information and belief, Plaintiffs have engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Plaintiffs have had actual knowledge of the '397 patent and

knowledge that their acts were inducing infringement of the ’397 patent since at least the date Plaintiffs received notice that such activities infringed the ’397 patent.

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

66. Upon information and belief, since the date of its receipt of notice, Plaintiffs’ infringement of the ’397 patent has been willful.

67. Express Mobile has been harmed by Plaintiffs’ infringing activities.

SECOND COUNTERCLAIM – INFRINGEMENT OF U.S. PATENT NO. 7,594,168

68. The allegations set forth in the foregoing paragraphs 1 through 67 are incorporated into this Second Claim for Relief.

69. 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.

70. 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.

71. 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.

72. 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.

73. 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.

74. 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.

75. Express Mobile 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.

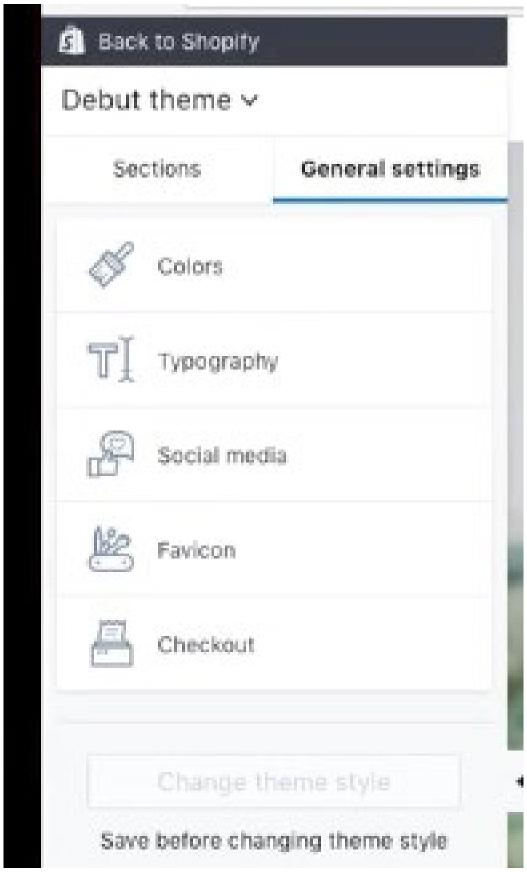
76. Upon information and belief, Plaintiffs has and continues to directly infringe at least claim 1 of the '168 patent by using a browser-based website and/or web page authoring tool in which the user-selected settings representing website elements are stored in a database, and retrieval of said information to generate said website (the "Accused Instrumentalities"). The Accused Instrumentalities include but are not limited to Plaintiffs' website building tools.

77. 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 run time engine file that is executed at run time that facilitates retrieval of information from the database and generates commands to display a web page or website is configured to generate the website from the objects and style data extracted from the provided database. *See, e.g.*, *X Commerce, Inc. v Express Mobile, Inc.*, Case No 17-cv-02605-RS, NDCA, DKT 79-7;

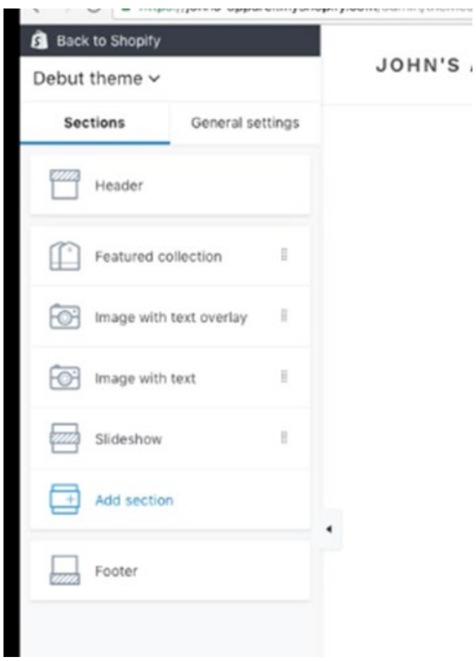
78. The Accused Instrumentalities infringe claim 1 of the ’168 patent through a combination of features which collectively practice each limitation of claim 1.

79. By way of example, the Accused Instrumentalities comprise plaintiffs’ Shopify Theme Editor, shown below, as part of a system for assembling a Web site.



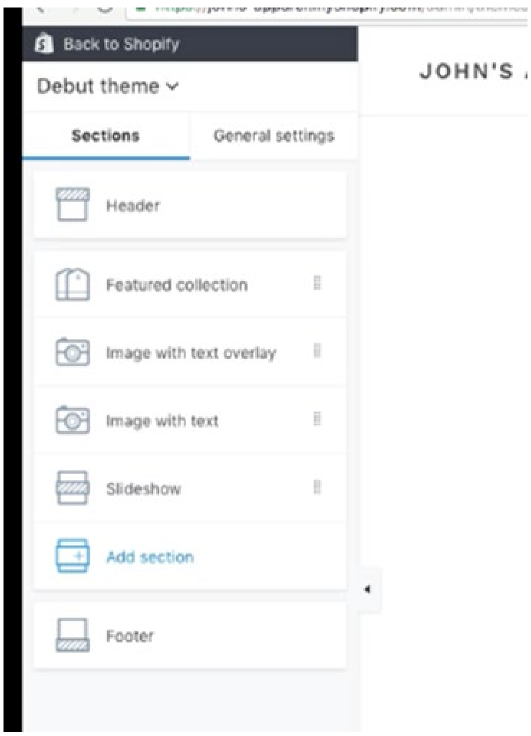
80. The Accused Instrumentalities include a server comprising a build engine, the build engine comprising the Shopify Theme Editor, its mySQL database and its JSON functionality.

81. The build engine is configured to accept user input to create a website comprising a plurality of web pages, where each web page comprising a plurality of objects. For example, as shown in the screen shot below, the Shopify Theme Editor is configured to accept user input and create a website.

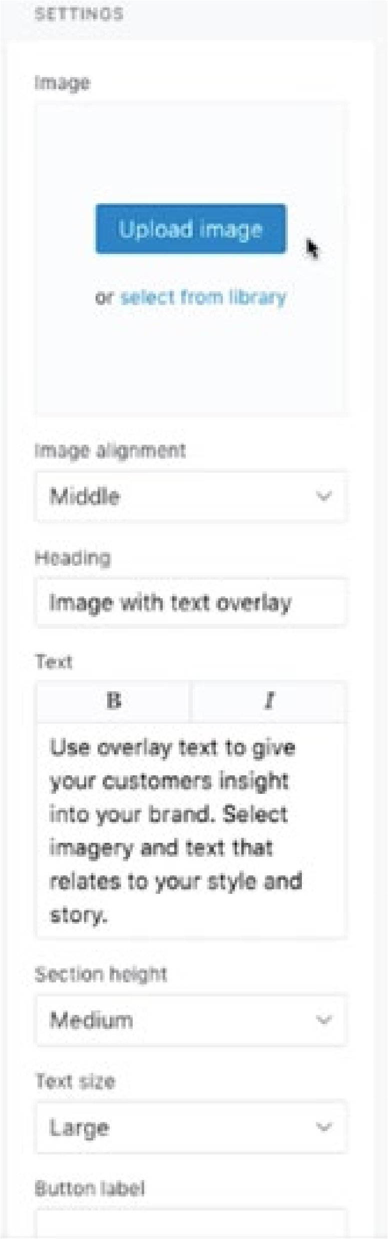


(https://help.shopify.com/manual/using-themes/change-the-layout)

As shown in the screen shot below, the website is capable of comprising a plurality of web pages, each web page comprising a plurality of objects, including, but not limited to, multimedia objects, such as "image."



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



SETTINGS

Image

Image alignment: Middle

Heading: Image with text overlay

Text:

Use overlay text to give your customers insight into your brand. Select imagery and text that relates to your style and story.

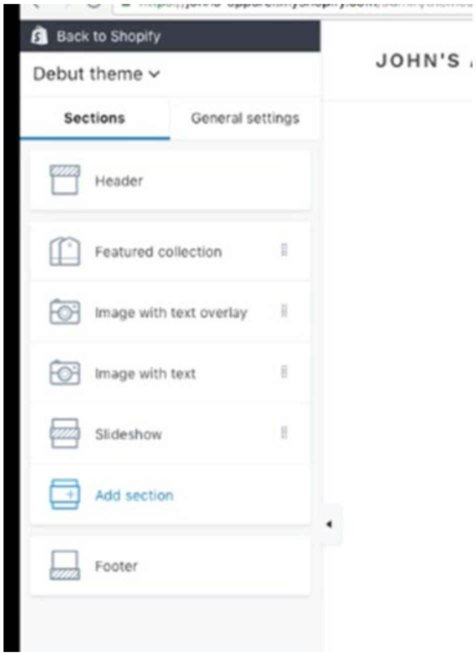
Section height: Medium

Text size: Large

Button label

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

82. Additionally, the Accused Instrumentalities' build engine is configured to accept user input to associate a style with objects of the plurality of web pages. As shown in the screen shots below, by using the Shopify Theme Editor, a user can input fields to associate a style, such as "color," with an object. A new style can be created by selecting the various options below.



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

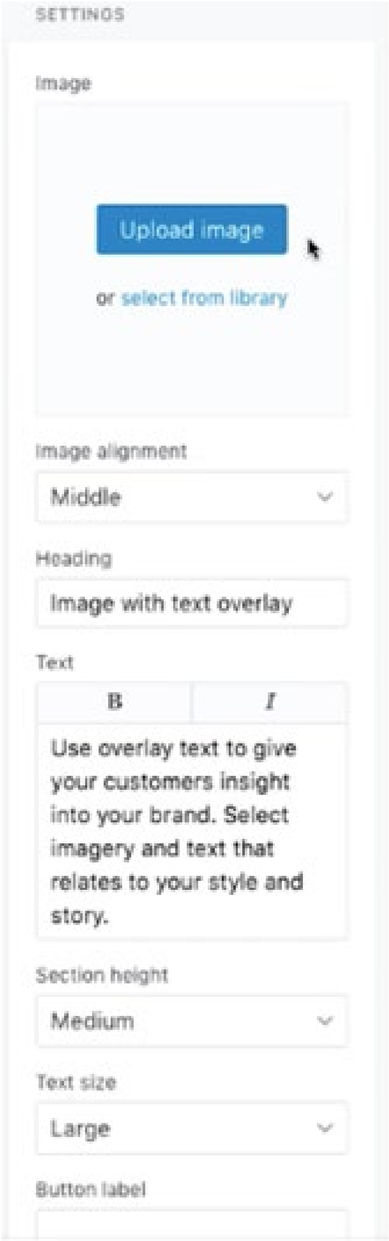


Image alignment
Middle

Heading
Image with text overlay

Text
Use overlay text to give your customers insight into your brand. Select imagery and text that relates to your style and story.

Section height
Medium

Text size
Large

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

83. Each web page can comprise at least one button object or at least one image object, as shown in the screen shots below.

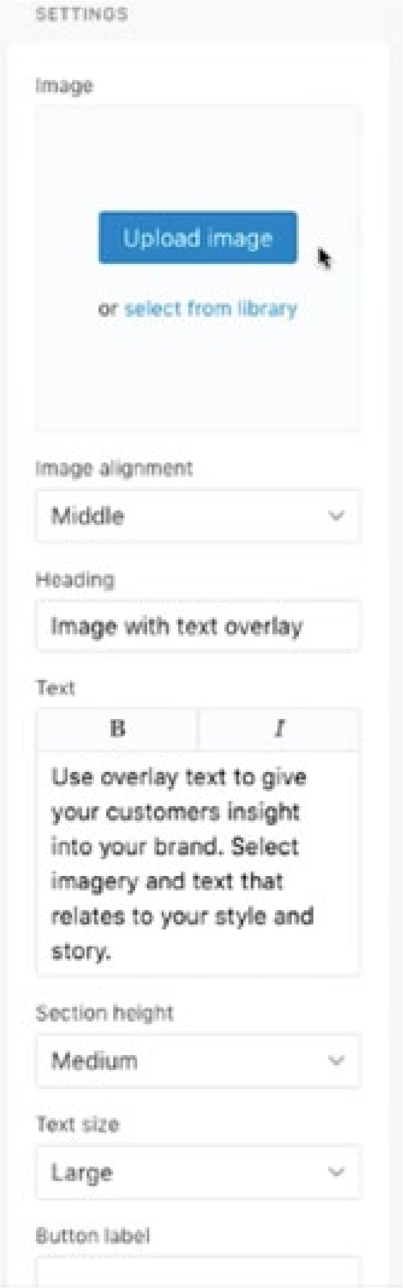


Image alignment
Middle

Heading
Image with text overlay

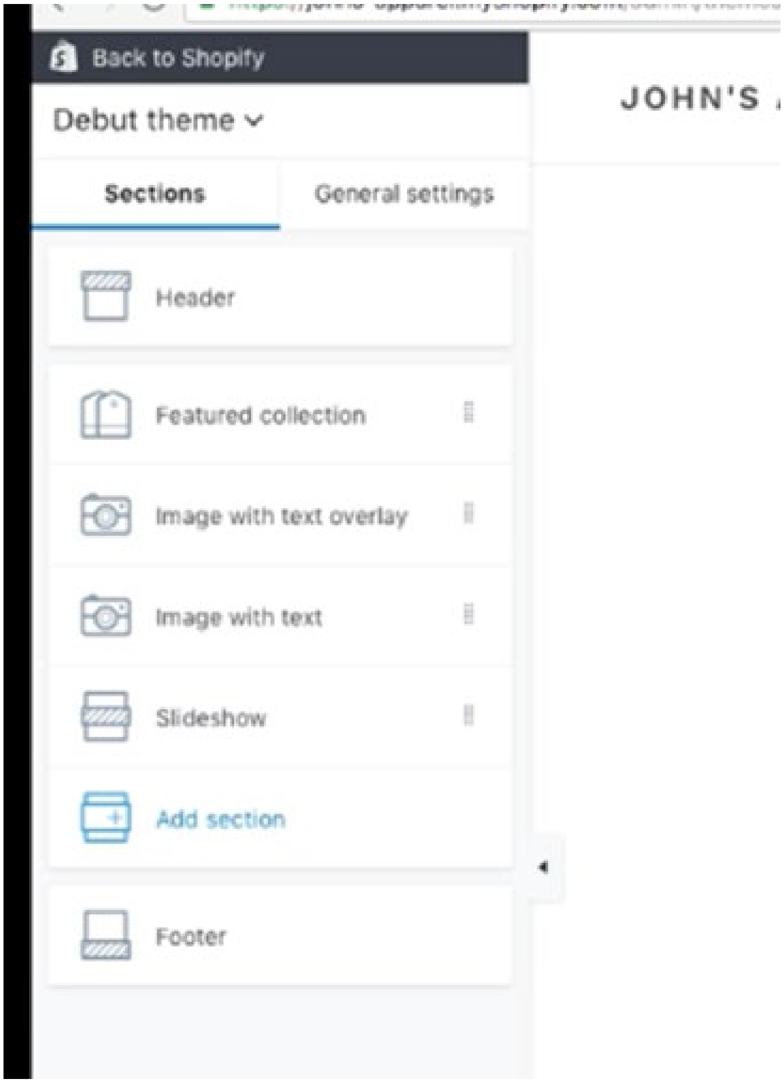
Text
Use overlay text to give your customers insight into your brand. Select imagery and text that relates to your style and story.

Section height
Medium

Text size
Large

Button label

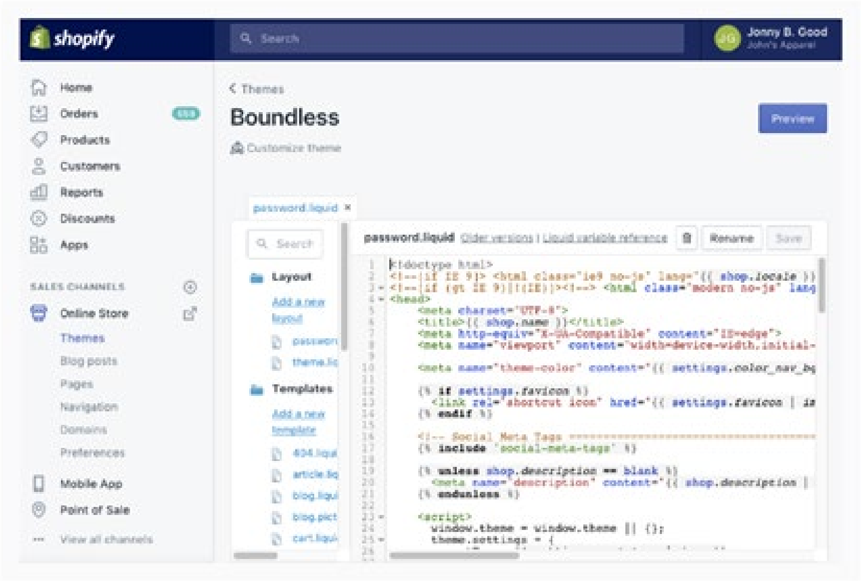
(https://help.shopify.com/manual/using-themes/change-the-layout).



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

84. Additionally, the at least one button object or at least one image object is associated with a style that includes values defining transformations and time lines for the at least one button object or at least one image object. Buttons and images are associated with the animation styles through CSS as shown in the screen shots below. These animations define transformations and time lines as shown, for example, with “Bounce.”

Edit HTML/CSS page

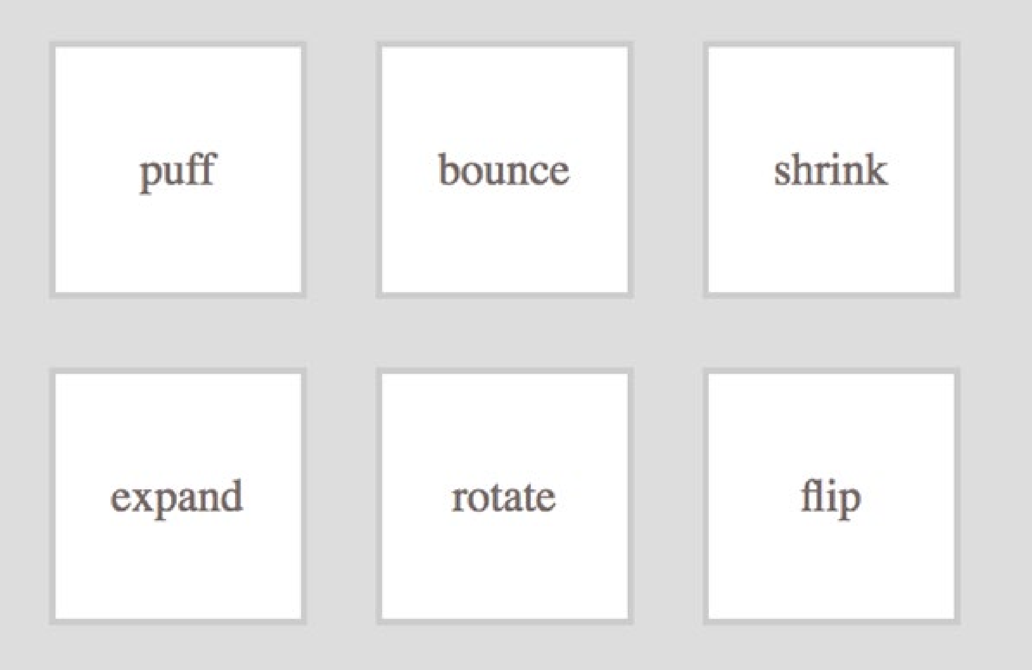


```
<!doctype html>
<!--[if IE 9]> <html class="ie9 no-js" lang="{{ shop.locale }}
<!--[if (gt IE 9)|!(IE)]><!--> <html class="modern no-js" lang
<head>
    <meta charset="UTF-8">
    <title>{{ shop.name }}</title>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-

    <meta name="theme-color" content="{{ settings.color_nav_bg

    {% if settings.favicon %}
      <link rel="shortcut icon" href="{{ settings.favicon | im
    {% endif %}

    <!-- Social Meta Tags ==================================
    {% include 'social-meta-tags' %}

    {% unless shop.description == blank %}
      <meta name="description" content="{{ shop.description |
    {% endunless %}

    <script>
      window.theme = window.theme || {};
      theme.settings = {
```

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

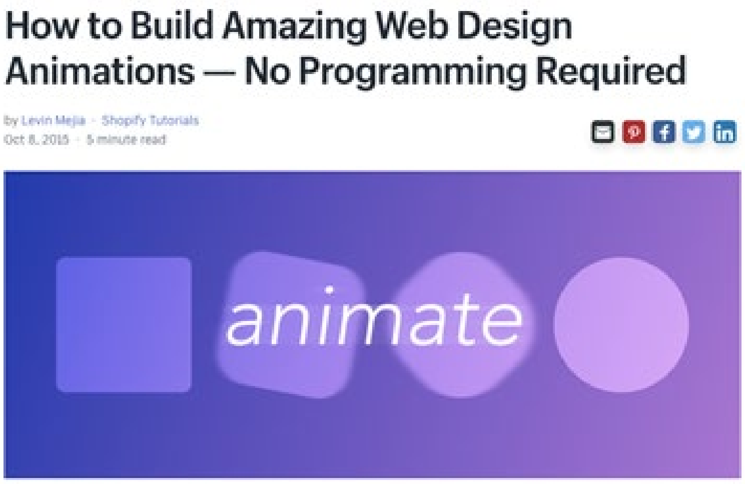
(https://help.shopify.com/themes/development/getting-started/choosing-an-editor).

Various Shopify Theme Editor CSS3 animations are shown in the screen shots below.



http://demos.dojotoolkit.org/demos/css3/demo.html

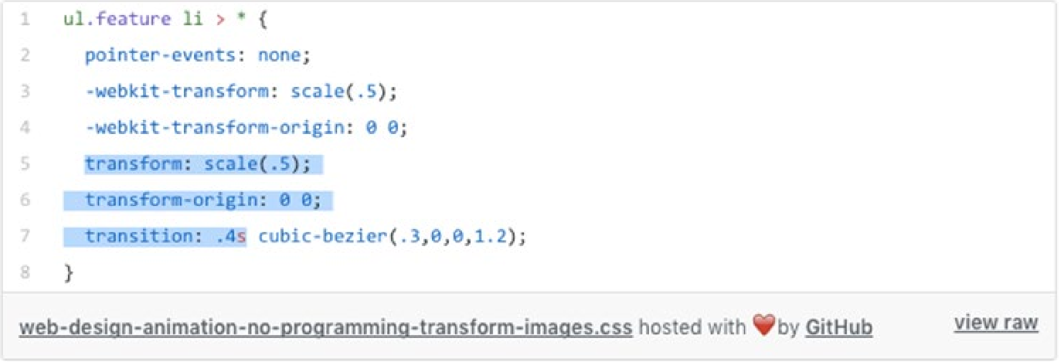
The Shopify Theme Editor incorporates animations that necessarily includes transformations and timelines as shown in the screen shots below.



How to Build Amazing Web Design Animations — No Programming Required

by Levin Mejia · Shopify Tutorials
Oct 8, 2015 · 5 minute read

Initial Transform:



```
ul.feature li > * {
  pointer-events: none;
  -webkit-transform: scale(.5);
  -webkit-transform-origin: 0 0;
  transform: scale(.5);
  transform-origin: 0 0;
  transition: .4s cubic-bezier(.3,0,0,1.2);
}
```

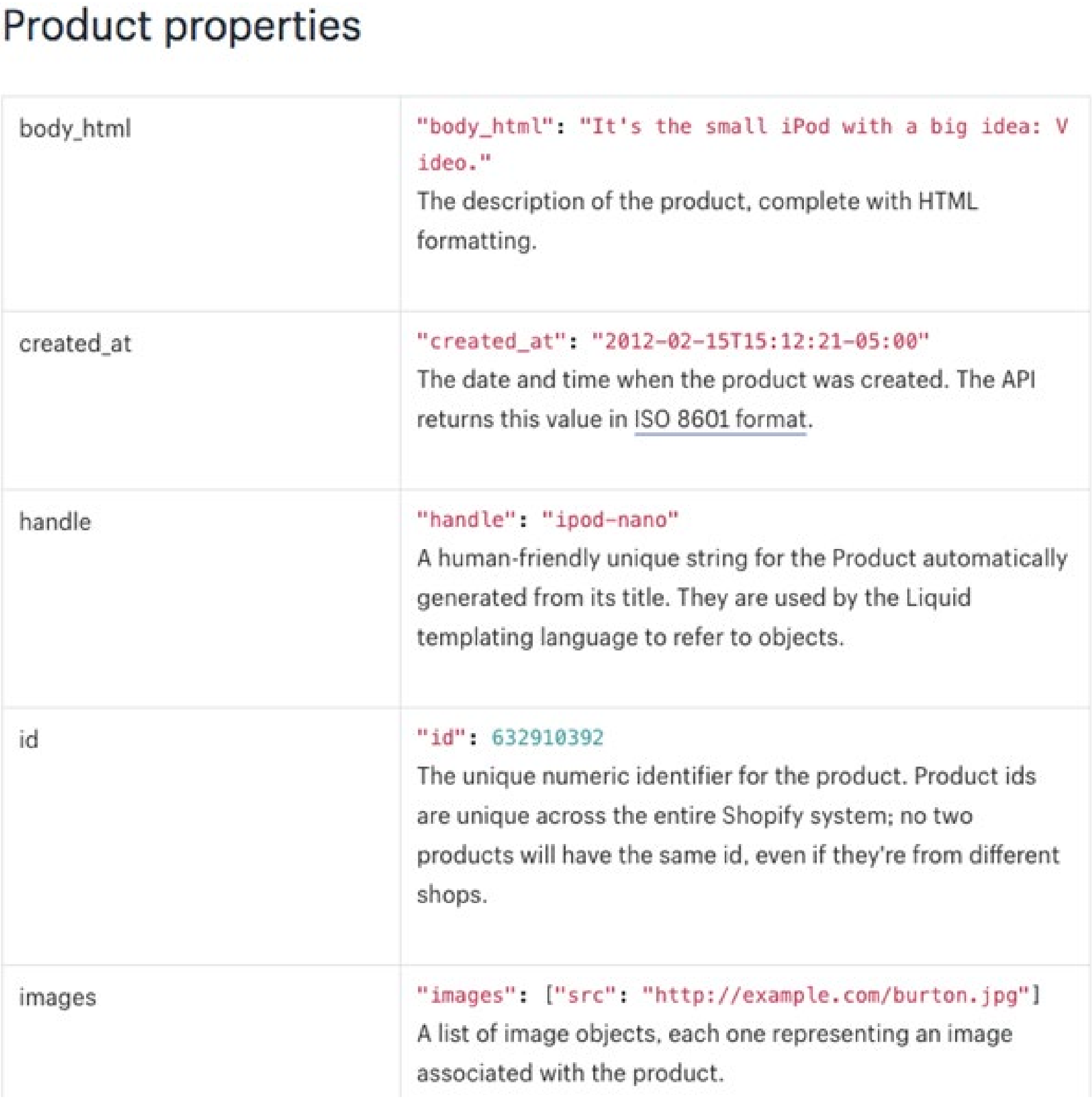
web-design-animation-no-programming-transform-images.css hosted with ❤ by GitHub

(Https://www.shopify.com/partners/blog/68949059-how-to-build-amazing-web-design-animations-no-programming-required).

85. Each Accused Instrumentalities' web page is defined entirely by each of the plurality of objects comprising that web page and the style associated with the object.

86. The Accused Instrumentalities are configured to produce a database with a multidimensional array comprising the objects that comprise the web site. The database that includes a multi-dimensional array comprising objects that comprise the website as shown in the

screen shot below. For example, Shopify has parameters for "Product" including "body" (one dimension), created at (the second dimension), and images (third dimension), thus making this parameter a multidimensional array.



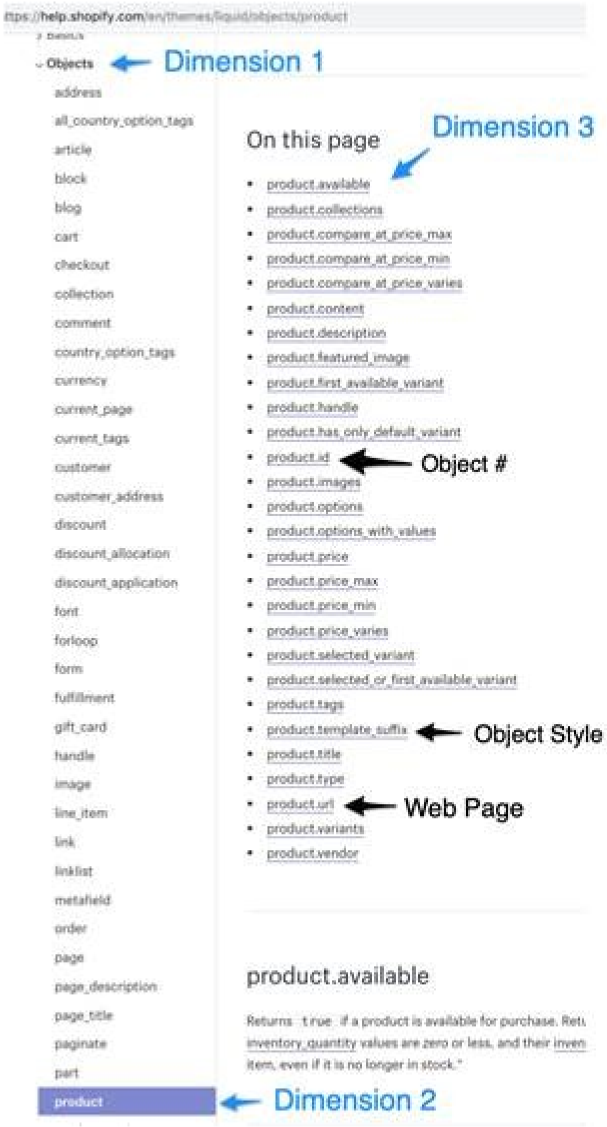
Product properties

body_html	`"body_html": "It's the small iPod with a big idea: Video."` The description of the product, complete with HTML formatting.
created_at	`"created_at": "2012-02-15T15:12:21-05:00"` The date and time when the product was created. The API returns this value in ISO 8601 format.
handle	`"handle": "ipod-nano"` A human-friendly unique string for the Product automatically generated from its title. They are used by the Liquid templating language to refer to objects.
id	`"id": 632910392` The unique numeric identifier for the product. Product ids are unique across the entire Shopify system; no two products will have the same id, even if they're from different shops.
images	`"images": ["src": "http://example.com/burton.jpg"]` A list of image objects, each one representing an image associated with the product.

(https://help.shopify.com/api/reference/product).

87. The Accused Instrumentalities include data defining, for each object, the object style, an object number, and an indication of the web page that each object is part. The screen shot below illustrates the Shopify Theme Editors' capability to include data defining, for each

object, the object style, an object number, and an indication of the web page that each object is part.



product.available

Returns true if a product is available for purchase. Retu
inventory_quantity values are zero or less, and their inven
item, even if it is no longer in stock."

(Https://help.shopify.com/en/themes/liquid/objects/product).

88. The Accused Instrumentalities are configured to provide the database to a server accessible to a web browser. The interaction between the mySQL database and the Shopify Server provides the database to a server accessible to a web browser.

89. The Accused Instrumentalities' database is produced such that a web browser with access to a runtime engine is configured to generate the web-site from the objects and style data extracted from the provided database. The Accused Instrumentalities' runtime files, such as the CSS, Java, JSON files, are the files that are executed when a given Shopify site is being utilized.

90. On information and belief, Accused Instrumentalities are used, marketed, provided to, and/or used by or for Plaintiffs' partners, clients, customers and end users across the country and in this district.

91. On information and belief, Plaintiffs were made aware of the '168 patent and its infringement thereof at least as early as the filing by Express Mobile of the first patent infringement suits against Plaintiffs' Solution Partners on January 15, 2016.

92. Upon information and belief, since at least the time Plaintiffs received notice, Plaintiffs have induced and continue to induce others to infringe at least one claim of the '168 patent under 35 U.S.C. § 271(b) by, among other things, and with specific intent or willful blindness, actively aiding and abetting others to infringe, including but not limited to Plaintiffs' partners, clients, customers, and end users, whose use of the Accused Instrumentalities constitutes direct infringement of at least one claim of the '168 patent.

93. In particular, Plaintiffs' actions that aid and abet others such as its customers, clients, partners, developers, and end users to infringe include advertising and distributing the Accused Instrumentalities and providing instruction materials, training, and services regarding the Accused Instrumentalities. Plaintiffs actively encourage the adoption of the Accused Instrumentalities and provide multiple support sites for the vast network of developers working with the Accused Instrumentalities and also offer courses for training and certification courses to

Plaintiffs' customers, partners, and developers. (*See, e.g.,* https://community.shopify.com/; https://shopify.com/support; https://u.shopify.com/about.). On information and belief, Plaintiffs have engaged in such actions with specific intent to cause infringement or with willful blindness to the resulting infringement because Plaintiffs have had actual knowledge of the '168 patent and knowledge that their acts were inducing infringement of the '168 patent since at least the date Plaintiffs received notice that such activities infringed the '168 patent.

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

95. Upon information and belief, since the date of its receipt of notice, Plaintiffs' infringement of the '168 patent has been willful.

96. Express Mobile has been harmed by Plaintiffs' infringing activities.

THIRD COUNTERCLAIM – INFRINGEMENT OF U.S. PATENT NO. 9,063,755

97. The allegations set forth in the foregoing paragraphs 1 through 96 are incorporated into this Third Claim for Relief.

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

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

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

device, and overcome problems specifically arising in the realm of computerized display content generation technologies.

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

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

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

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

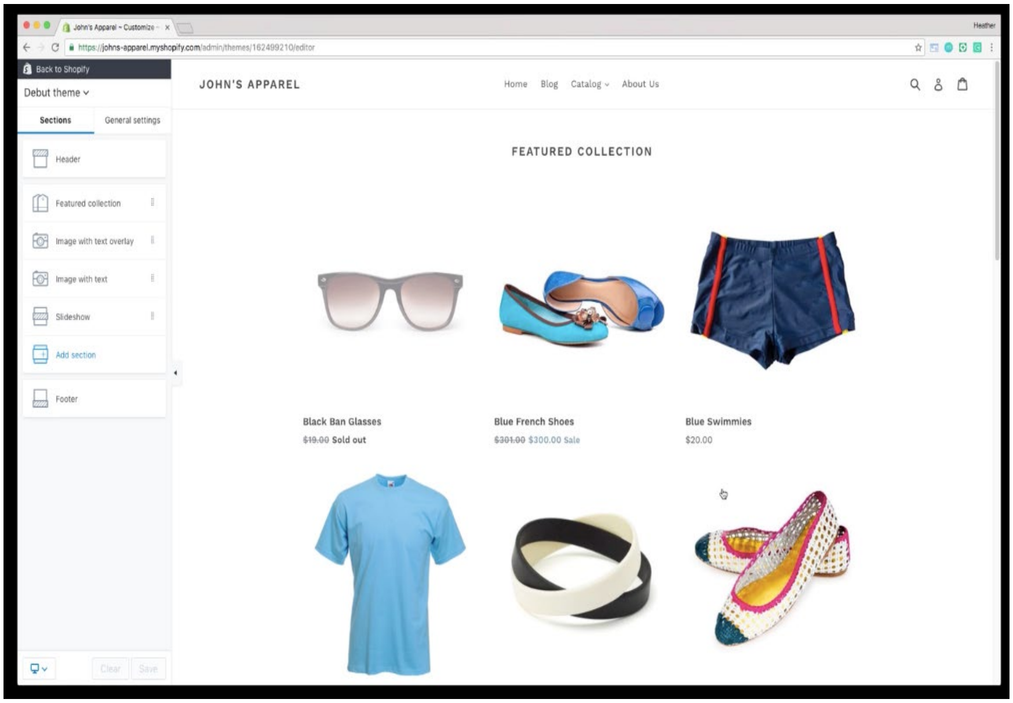
105. Upon information and belief, Plaintiffs have and continue to directly infringe at least claims 1 and 15 of the ’755 patent by a system for generating code to provide content on a display of a device. The system includes a computer memory and an authoring tool, The computer memory stores a registry of: a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent

address or pointer to an output value accessible to the web service, where each symbolic name has an associated data format class type corresponding to a subclass of User Interface (UI) objects that support the data format type of the symbolic name, and has a preferred UI object, and b) an address of the web service. The authoring tool is configured to: define a (UI) object for presentation on the display, where the defined UI object corresponds to a web component included in the registry selected from a group consisting of an input of the web service and an output of the web service; access the computer memory to select the symbolic name corresponding to the web component of the defined UI object, associate the selected symbolic name with the defined UI object; produce an Application including the selected symbolic name of the defined UI object, where the Application is a device-independent code; and produce a Player, where the Player is a device-dependent code, such that when the Application and Player are provided to the device and executed on the device, and when the user of the device provides one or more input values associated with an input symbolic name to an input of the defined UI object, 1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service, 2) the web service utilizes the input symbolic name and the user provided one or more input values for generating one or more output values having an associated output symbolic name, 3) the Player receives the output symbolic name and corresponding one or more output values and provides instructions. (The "Accused Instrumentalities"). The Accused Instrumentalities include the Shopify Theme Editor, based on the Ruby on Rails platform, that enables the functionality described above. See, for example, https://medium.com/@chris.chimen/build-shopify-app-with-ruby-on-rails-for-beginer-part-1-40471da7d607

106. In particular, claim 1 of the ’755 patent recites 1 a system for generating code to provide content on a display of a device, said system comprising: computer memory storing a registry of: a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service, and b) the address of the web service; an authoring tool configured to: define a user interface (UI) object for presentation on the display, where said UI object corresponds to the web component included in said registry selected from the group consisting of an input of the web service and an output of the web service, access said computer memory to select the symbolic name corresponding to the web component of the defined UI object, associate the selected symbolic name with the defined UI object, produce an Application including the selected symbolic name of the defined UI object, where said Application is a device-independent code, and produce a Player, where said Player is a device-dependent code; such that, when the Application and Player are provided to the device and executed on the device, and when a user of the device provides one or more input values associated with an input symbolic name to an input of defined UI object, 1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service, 2) the web service utilizes the input symbolic name and the user provided one or more input values for generating one or more output values having an associated output symbolic name, 3) said Player receives the output symbolic name and corresponding one or more output values and provides instructions for a display of the device to present an output value in the defined UI object.

107. The Accused Instrumentalities infringe claim 1 of the ’755 patent through a combination of features which collectively practice each limitation of claim 1. By way of

example, as shown in the screen shot below, the Accused Instrumentalities feature a method of displaying content of a device.



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

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

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

110. Furthermore, the registry contains symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable

over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service. In particular, the Accused Instrumentalities' MySQL database contains symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network by the formatting of the symbolic names in conjunction with the Shopify Theme Editor, widget, and plugin authoring tools.

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

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

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

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

115. The Accused Instrumentalities' authoring tool is also configured to associate the selected symbolic name with the defined UI object, i.e., the JSON formatted element, where the

selected symbolic name is only available to UI objects that support the defined data format associated with the element associated with that symbolic 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 Shopify Theme Editor makes a UI element request a JSON request is sent to the Server and a JSON data response is provided to the Interface.

116. The Accused Instrumentalities' authoring tool is configured to produce an Application including the selected symbolic name of the defined UI object, where the Application is a device-independent code. The Accused Instrumentalities' Application includes the user selectable settings, as augmented by the web component definition. The application, because it contains user selected settings, and is represented by Boolean, numbers and String primitives, is device independent.

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

118. The Accused Instrumentalities feature a system where the Application and Player are provided to the device and executed on the device and when the user of the device provides one or more input values associated with an input symbolic name to an input of the defined UI

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

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

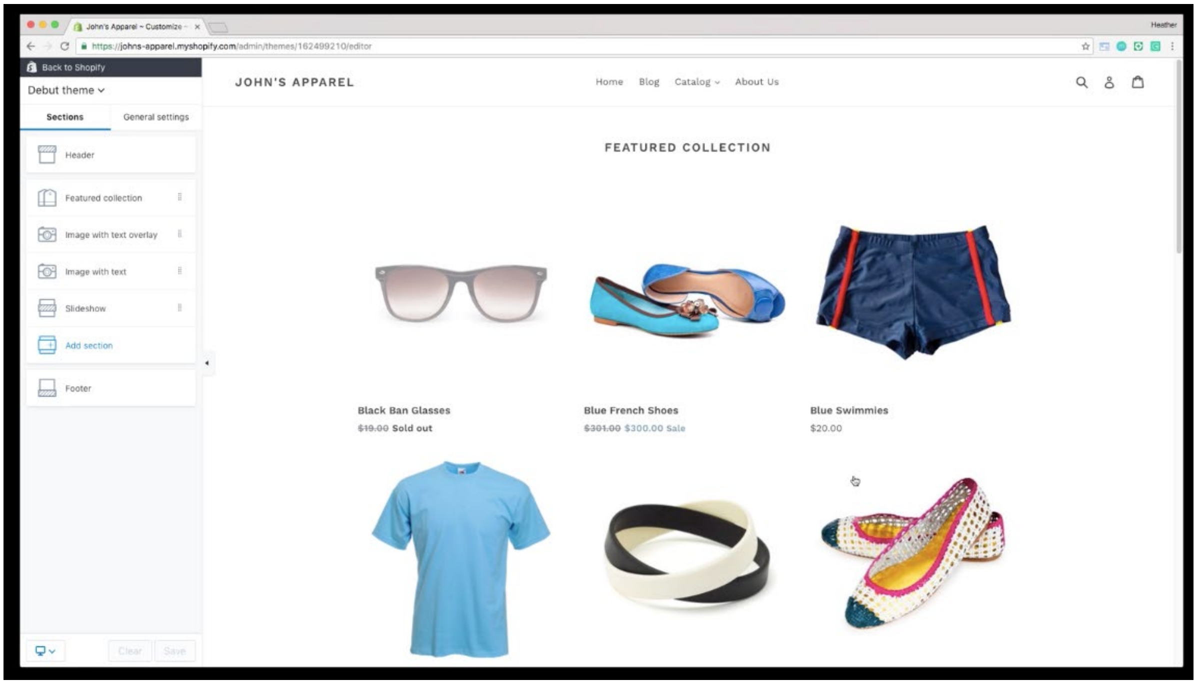
120. The Accused Instrumentalities feature a system where the web service utilizes the input symbolic name and the user provided one or more input values for generating one or more output values having an associated output symbolic name. Because of the JSON formatting, the output values having an associated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

121. 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 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.

122. The presence of the above referenced features is demonstrated, by way of example, by reference to publicly available information, including https://www.shopify.com/;

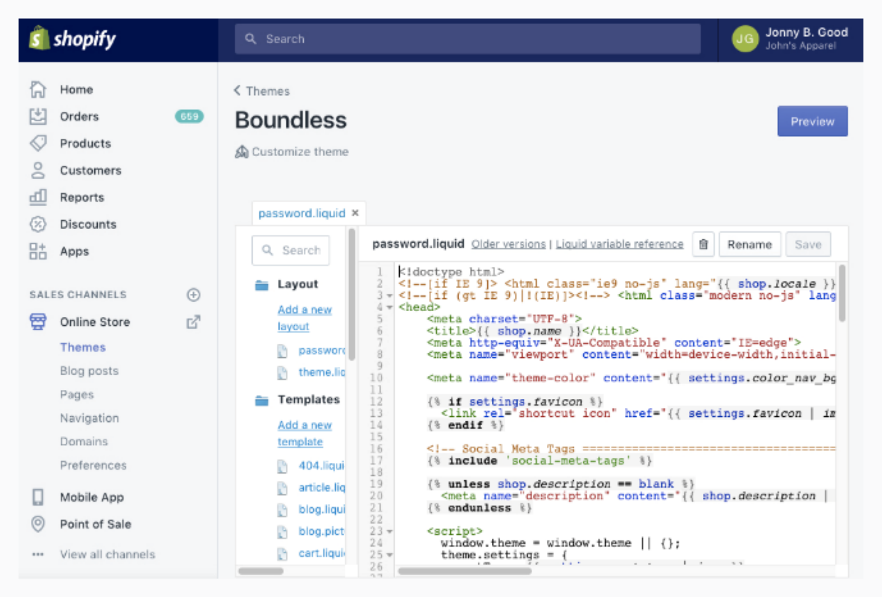
https://www.shopify.com/website/hosting; https://help.shopify.com/manual/using-themes/change-the-layout; https://help.shopify.com/manual/using-themes/change-the-layout/using-theme-presets https://help.shopify.com/manual/using-themes/troubleshooting/fix-64-kilobyte-limit-errors; https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html;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

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



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

Edit HTML/CSS page



```
<!doctype html>
<!--[if IE 9]> <html class="ie9 no-js" lang="{{ shop.locale }}
<!--[if (gt IE 9)|!(IE)]><!--> <html class="modern no-js" lang
<head>
    <meta charset="UTF-8">
    <title>{{ shop.name }}</title>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-

    <meta name="theme-color" content="{{ settings.color_nav_bg

    {% if settings.favicon %}
      <link rel="shortcut icon" href="{{ settings.favicon | im
    {% endif %}

    <!-- Social Meta Tags ================================
    {% include 'social-meta-tags' %}

    {% unless shop.description == blank %}
      <meta name="description" content="{{ shop.description |
    {% endunless %}

    <script>
      window.theme = window.theme || {};
      theme.settings = {
```

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

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

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

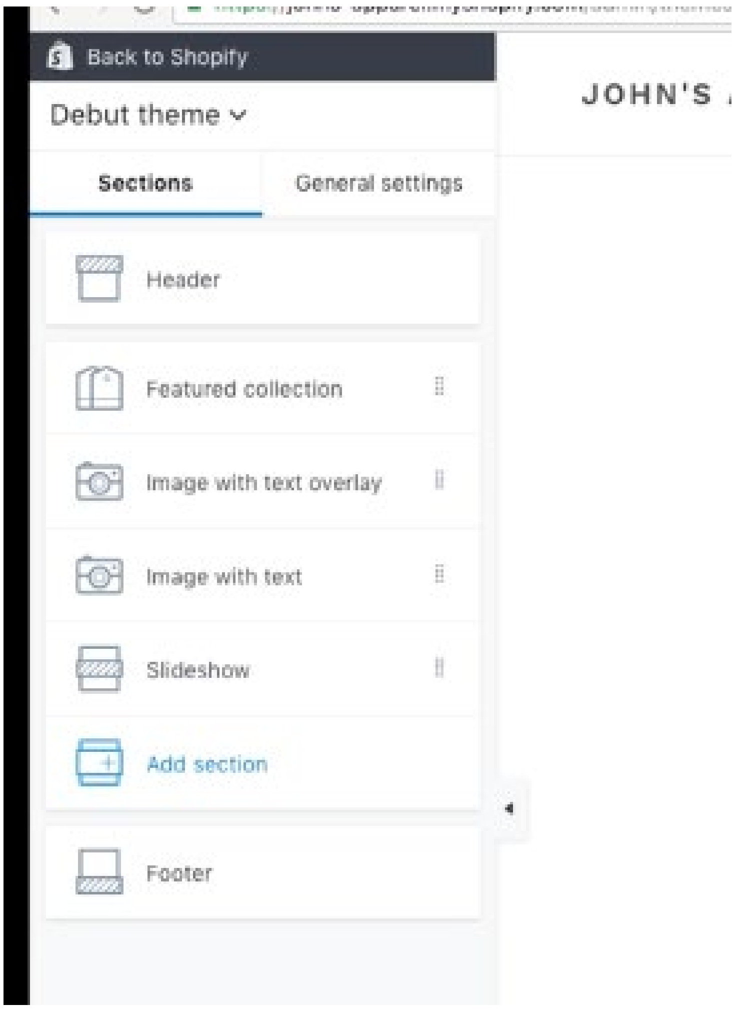
125. Each web component includes a plurality of symbolic names of inputs and outputs associated with each web service. The plurality of symbolic names of inputs and outputs associated with each web service is a feature of their JSON formatting characteristics. Each symbolic name has an associated data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

126. The registry includes: a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service. The registry and Ruby on Rails MySQL database contain symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network as demonstrated by the formatting in conjunction with the Shopify Theme Editor, widget, and plugin authoring tools. JSON names are strings that only represent the symbolic names that are bound both to a web service input and/or output and to a UI object. All JSON

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

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

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



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

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

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

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

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

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

133. The Accused Instrumentalities feature a system where the Application and Player are provided to the device and executed on the device and when the user of the device provides

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

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

135. 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.

136. 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 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.

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

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

FOURTH COUNTERCLAIM – INFRINGEMENT OF U.S. PATENT NO. 9,471,287

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

140. 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.

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

the defined UI object. These features are exclusively implemented utilizing computer technology.

142. 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.

143. 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.

144. 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.

145. 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.

146. Express Mobile 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.

147. Upon information and belief, Plaintiffs have and continue to directly infringe at least claims 1 and 15 of the ’287 patent by a system and method which includes a registry and an authoring tool or Player configured to define a User Interface (“UI”) object for display on the device, where the UI object corresponds to a web component. Each UI object is either: 1) selected by a user or 2) automatically selected by the system as a preferred UI object corresponding to a symbolic name of the web component and used to produce an Application, where the Application is a device-independent code and a Player, where the Player is a device-dependent code. The Application and Player enable 1) the device to provide one or more input values and corresponding input symbolic name to the web service and 2) the web service to utilize the input symbolic name and the user provided one or more input values to generate one or more output values having an associated output symbolic name, while 3) the Player receives the output symbolic name and corresponding one or more output values and provides instructions for the display of the device to present an output value in the defined UI object (the “Accused Instrumentalities”). The Accused Instrumentalities include the Shopify Theme Editor, based on the Ruby on Rails platform, that enables the functionality described above. See, for example, https://medium.com/@chris.chimen/build-shopify-app-with-ruby-on-rails-for-beginer-part-1-40471da7d607

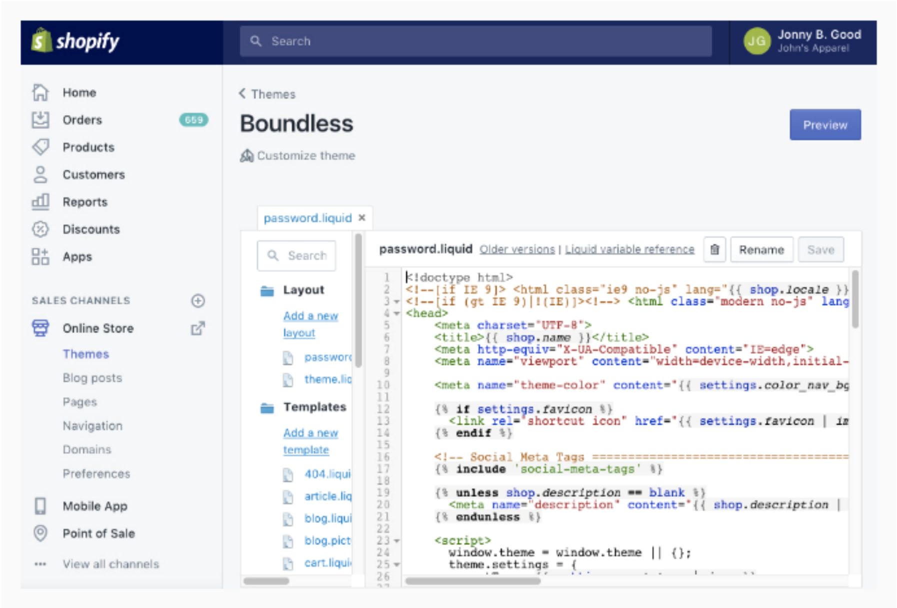
148. In particular, claim 1 of the ’287 patent recites 1 a system for generating code to provide content on a display of a device, the system comprising: computer memory storing a registry of: a) symbolic names required for evoking one or more web components each related to

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.

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

Edit HTML/CSS page

1.



```
<!doctype html>
<!--[if IE 9]> <html class="ie9 no-js" lang="{{ shop.locale }}
<!--[if (gt IE 9)|!(IE)]><!--> <html class="modern no-js" lang
<head>
    <meta charset="UTF-8">
    <title>{{ shop.name }}</title>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-

    <meta name="theme-color" content="{{ settings.color_nav_bg

    {% if settings.favicon %}
      <link rel="shortcut icon" href="{{ settings.favicon | im
    {% endif %}

    <!-- Social Meta Tags ====================================
    {% include 'social-meta-tags' %}

    {% unless shop.description == blank %}
      <meta name="description" content="{{ shop.description |
    {% endunless %}

    <script>
      window.theme = window.theme || {};
      theme.settings = {
```

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

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

150. The Accused Instrumentalities feature a computer memory provided by the Shopify servers and their MySQL database. By way of example, the Accused Instrumentalities’

Ruby on Rails platform utilizes JSON strings extensively as part of its API, which necessarily require servers and databases. (See, https://developer.Ruby on Rails.org/rest-api/.)

151. The computer memory stores a registry of a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service. In particular, the Accused Instrumentalities' MySQL database contains symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network by the formatting of the symbolic names in conjunction with the Shopify Theme Editor, widget, and plugin authoring tools.

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

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

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

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

156. Each defined UI object automatically selected by the system as the preferred UI object corresponding to the symbolic name of the web component selected by the user of the authoring tool. The Accused Instrumentalities' UI objects are automatically selected by the system as the preferred UI object corresponding to the symbolic name of the web component selected by the user of the authoring tool, i.e., a UI object selected by a user is automatically selected.

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

158. The Accused Instrumentalities' authoring tool is also configured to associate the selected symbolic name with the defined UI object, i.e., the JSON formatted element, where the selected symbolic name is only available to UI objects that support the defined data format associated with the element associated with that symbolic 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 Shopify Theme Editor makes a UI element request a JSON request is sent to the Server and a JSON data response is provided to the Interface.

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

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

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

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

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

164. The Accused Instrumentalities feature a system where the Player receives the output symbolic name and corresponding one or more output values and provides instructions for the display of the device to present an output value in the defined UI object. The runtime player 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

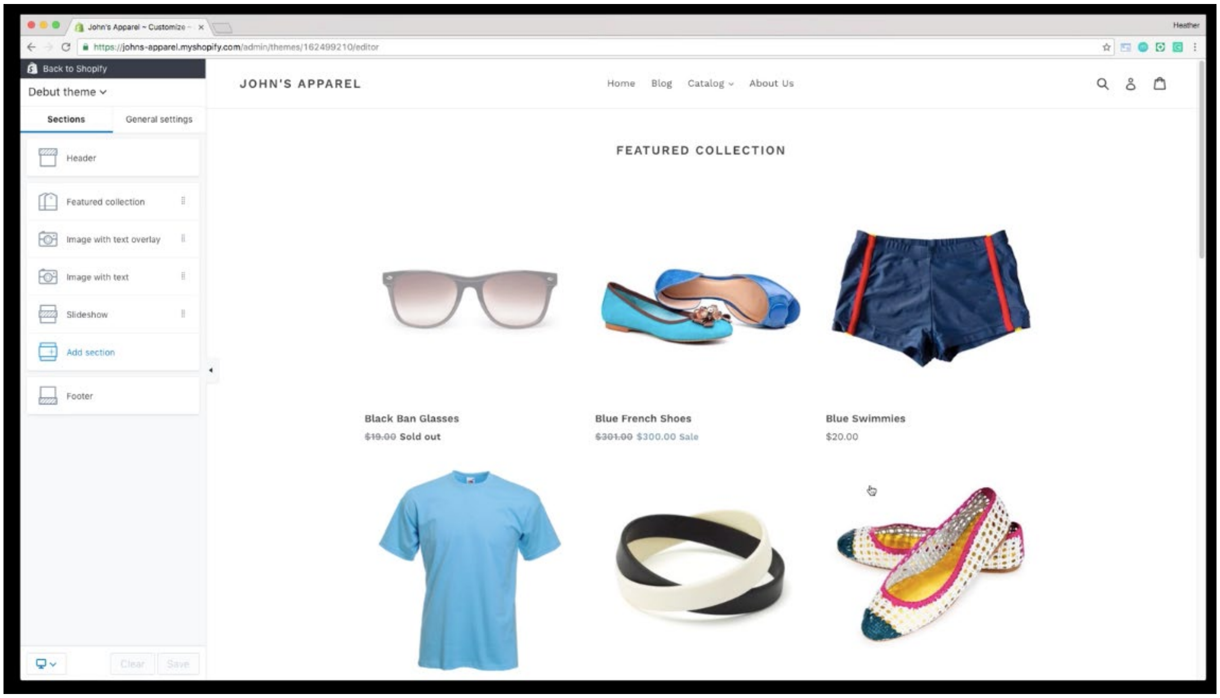
165. The presence of the above referenced features is demonstrated, by way of example, by reference to publicly available information, including https://www.shopify.com/; https://www.shopify.com/website/hosting; https://help.shopify.com/manual/using-themes/change-the-layout; https://help.shopify.com/manual/using-themes/change-the-layout/using-theme-presets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

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

166. 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.

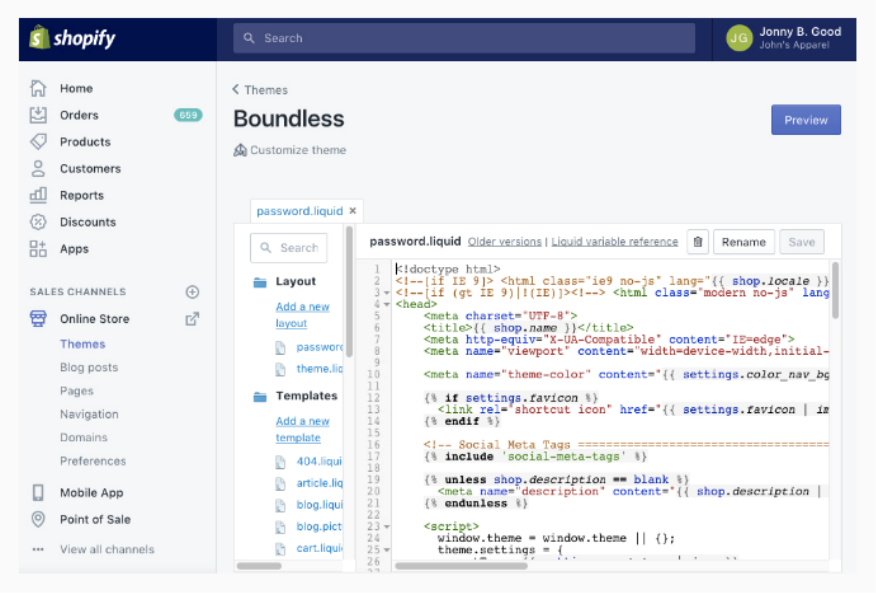
167. The Accused Instrumentalities infringe claim 15 of the ’287 patent through a combination of features which collectively practice each limitation of claim 15. By way of example, the method is for displaying content on a display of a device as shown in the screen shot below.



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

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

Edit HTML/CSS page



```
<!doctype html>
<!--[if IE 9]> <html class="ie9 no-js" lang="{{ shop.locale }}
<!--[if (gt IE 9)|!(IE)]><!--> <html class="modern no-js" lang
<head>
    <meta charset="UTF-8">
    <title>{{ shop.name }}</title>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-

    <meta name="theme-color" content="{{ settings.color_nav_bg

    {% if settings.favicon %}
      <link rel="shortcut icon" href="{{ settings.favicon | im
    {% endif %}

    <!-- Social Meta Tags ==================================
    {% include 'social-meta-tags' %}

    {% unless shop.description == blank %}
      <meta name="description" content="{{ shop.description |
    {% endunless %}

    <script>
      window.theme = window.theme || {};
      theme.settings = {
```

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

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

169. The Accused Instrumentalities include defining a user interface (UI) object for presentation on the display, where the defined UI object corresponds to a web component included in the registry of one or more web components, and where the web component is selected from a group consisting of an input of a web service and an output of the web service. TheShopify Theme Editor defines a UI object for presentation on display, where the UI object corresponds to a web component included in the non-volatile computer memory selected from a group consisting of an input of a web service and an output of the web service by JSON data formatting. JSON names are strings that only represent the symbolic names that are bound both to a web service input and/or output and to a UI object. All JSON names in the name/value pairs are character strings. When the Ruby on Rails interface makes a UI element request a JSON request is sent to the Server and a JSON data response is provided to the Interface.

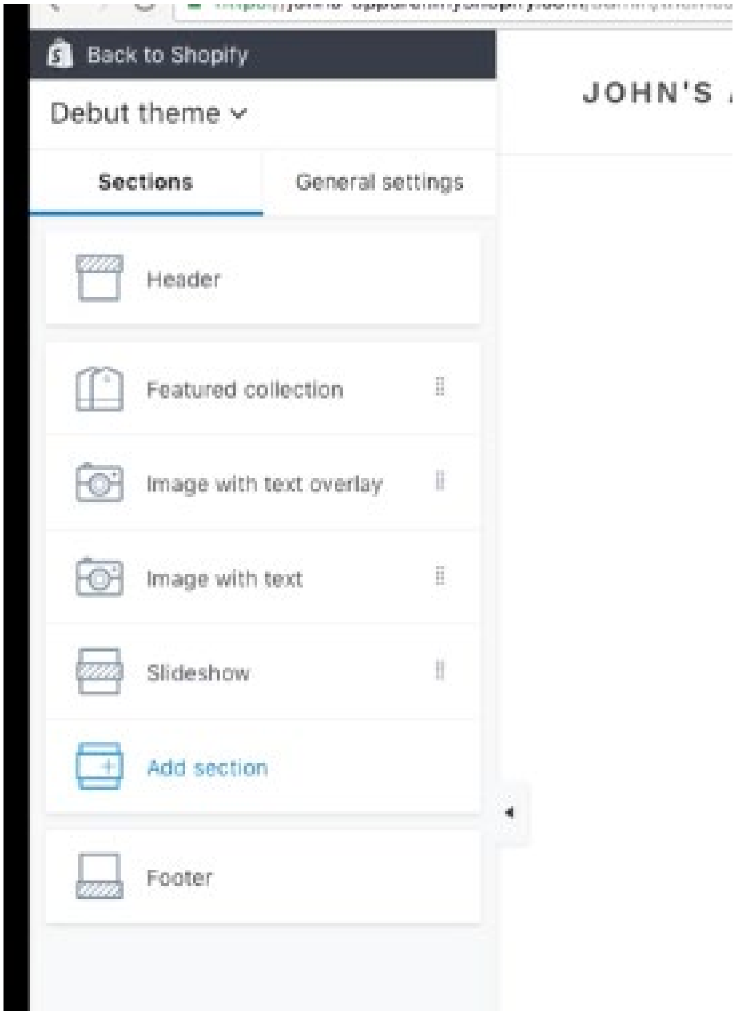
170. Each web component includes a plurality of symbolic names of inputs and outputs associated with each web service. The plurality of symbolic names of inputs and outputs associated with each web service is a feature of their JSON formatting characteristics. Each symbolic name has an associated data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

171. The registry includes: a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service. The registry and Ruby on Rails MySQL database contain symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network as demonstrated by the formatting in conjunction with the Shopify Theme Editor, widget, and plugin authoring tools. JSON names are strings that only represent the symbolic names that are bound both to a web service input and/or output and to a UI object. All JSON names in the name/value pairs are character strings. When the Ruby on Rails interface makes a UI element request a JSON request is sent to the Server and a JSON data response is provided to the Interface.

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

173. 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.

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



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

175. The Shopify Theme Editor's UI objects are automatically selected by the system as the preferred UI object corresponding to the symbolic name of the web component selected by the user of the authoring tool, i.e., a UI object selected by a user is automatically selected.

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

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

178. The Accused Instrumentalities include producing an Application including the selected symbolic name of the defined UI object, where the Application is a device-independent code. Ruby on Rails produces an Application including the symbolic name of the defined UI object.

179. The Accused Instrumentalities feature a system where the Application and Player are provided to the device and executed on the device and when the user of the device provides one or more input values associated with an input symbolic name to an input of the defined UI

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

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

181. 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.

182. 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 receives the output name, output value, and provides instructions for a display as shown by the fact that the defined UI object are ultimately rendered.

The presence of the above referenced features is demonstrated, by way of example, by reference to publicly available information including https://www.shopify.com/;

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

FIFTH COUNTERCLAIM – INFRINGEMENT OF U.S. PATENT NO. 9,928,044

183. The allegations set forth in the foregoing paragraphs 1 through 183 are incorporated into this Fifth Claim for Relief.

184. 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.

185. The inventions of the '044 patent resolve technical problems related to generating and distributing programming to mobile devices over a network. For example, the inventions of the '044 patent feature a computer memory and an authoring tool. The computer memory stores symbolic names required for evoking a web component related to a web service and an address of the web service. The authoring tool is 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. The information representative of the UI object and related settings are stored in a database. An application is built consisting web page views. The application and a player are provided to a device and enables the device to provide one or more input values and corresponding input symbolic name to the web service and 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 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.

186. 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.

187. The claims of the ’044 patent recite inventions that are not merely the routine or conventional use of systems for the computerized generation of content on a display of a device. Instead, the inventions feature systems for use with devices with authoring tools to generate and distribute application and player code that generate displays on a device, such as a mobile device, utilizing information stored in databases and retrieved from web services.

188. The technology claimed in the ’044 patent does not preempt all ways for the computerized generation and distribution of programming to a device, such as a mobile device, nor preempt the use of all authoring tools for the computerized generation and distribution of programming to a device, such as a mobile device, nor preempt any other well-known or prior art technology.

189. 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.

190. Express Mobile 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.

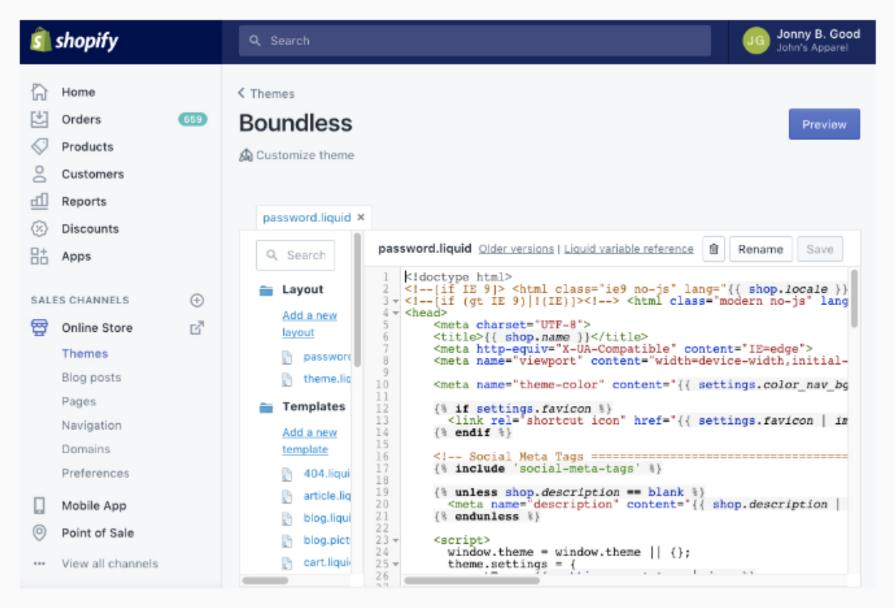
191. Upon information and belief, Plaintiffs has and continues to directly infringe at least claim 1 of the ’044 patent by a system which includes a computer memory and an authoring tool. The computer memory stores symbolic names required for evoking a web component related to a web service and an address of the web service. The authoring tool is 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. The information representative of the UI object and related settings are stored in a database. An application is built consisting web page views. The application and a player are provided to a device and enables the device to provide one or more input values and corresponding input symbolic name to the web service and 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 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. (The “Accused Instrumentalities”). The Accused Instrumentalities include the Shopify Theme Editor, based on the Ruby on Rails platform, that enables the functionality described above. See, for example, https://medium.com/@chris.chimen/build-shopify-app-with-ruby-on-rails-for-beginer-part-1-40471da7d607

192. In particular, claim 1 of the ’044 patent recites a system for generating code to provide content on a display of a device, said 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 said defined UI object corresponds to a web component included in said 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 said computer memory to select the symbolic name corresponding to the web component of the defined UI object, associate the selected symbolic name with the defined UI object, 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 said defined UI object and related settings in a database; retrieve said information representative of said one or more said UI object settings stored in said database; and build an application consisting of one or more web page views from at least a portion of said database utilizing at least one player, where said player utilizes information stored in said database to generate for the display of at least a portion of said 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. 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, as shown in the screen shot below, the Accused Instrumentalities feature a system for generating code, including HTML, CSS, and JSON code, to provide content on a display of a device.

2.

Edit HTML/CSS page



```
<!doctype html>
<!--[if IE 9]> <html class="ie9 no-js" lang="{{ shop.locale }}
<!--[if (gt IE 9)|!(IE)]><!--> <html class="modern no-js" lang
<head>
    <meta charset="UTF-8">
    <title>{{ shop.name }}</title>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-

    <meta name="theme-color" content="{{ settings.color_nav_bg

    {% if settings.favicon %}
      <link rel="shortcut icon" href="{{ settings.favicon | im
    {% endif %}

    <!-- Social Meta Tags ====================================
    {% include 'social-meta-tags' %}

    {% unless shop.description == blank %}
      <meta name="description" content="{{ shop.description |
    {% endunless %}

    <script>
      window.theme = window.theme || {};
      theme.settings = {
```

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

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

193. The Accused Instrumentalities feature a computer memory in servers it owns or controls to store symbolic names required to evoke a web component (e.g., video) and address of a web service (e.g., YouTube video).

194. The Accused Instrumentalities include the ability to select web components for display on a web page. Users can add content to their website accessible by symbolic names. This content includes web components that relate to inputs and outputs of web services over the Internet. Shopify's web design tools integrate with third-party web services such as Google, YouTube, Facebook, Pinterest, among others, integrate with payment providers like PayPal, Amazon Pay, Visa, among others, and integrate with Apps. For example, a user can add a Video

section of type “video” including a video URL and multimedia provided to/from the YouTube web service.

195. Furthermore, each symbolic name has an associated data format class type corresponding to a subclass of defined UI objects, such as buttons, text fields, images and videos, that supports the data format type of the symbolic name, and has a preferred UI object, such as a video display area for videos.

196. The computer memory also stores b) an address of the web service. For example, Shopify maintains the web address of the YouTube web service for retrieving video content from YouTube.

197. The Accused Instrumentalities feature an authoring tool in the form of the Shopify Theme Editor, widget, and plug-in authoring tools. Shopify’s web design tools allow users to design, customize, and configure a website. Shopify’s web design tools, for example, presents a panel to a user that includes numerous web design tools allowing a user to edit the Header, add, remove and edit Sections, edit the Footer, modify the website’s Theme settings, and modify the settings of each Header, Section, Footer, and other elements. Shopify’s website additionally includes tools allowing users to customize the HTML, CSS, Liquid, and other code underlying their website.

198. The authoring tool is configured to allow a Shopify user to define UI objects on their website. Shopify users select content using the web design tools to be placed on their websites that includes input textboxes, buttons, multimedia content, and sliders. The UI objects correspond to the web components included in the computer memory described above. For example, Shopify includes authoring tools that allow the user to define content on their website

for videos. These UI objects correspond to the video web component. These web components include both inputs from and outputs to corresponding web services.

199. Each defined UI object 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.

200. The Accused Instrumentalities' authoring tool is configured to access Shopify's computer memory to select the symbolic names described above. For example, Shopify's web platform is configured to generate a Section of type "video" when a user selects to add a Video Section to their website through Shopify's web design tools.

201. The Accused Instrumentalities' authoring tool is also configured to associate the symbolic names described above with the UI object displayed on the user's website. As explained above, Shopify's web design tools are configured to allow a user to define UI objects that correspond to web components. The UI objects are associated with symbolic names corresponding to the web components. For example, Shopify's web platform is configured to generate a Section of type "video" when a user selects to add a Video Section to their website through Shopify's web design tools.

202. The Accused Instrumentalities authoring tool includes a selected symbolic name, where the selected symbolic name is only available to UI objects that support the defined data format associated with that symbolic name.

203. The Accused Instrumentalities store information representative of said defined UI object and related settings in a database. Shopify's web platform saves information about each user's website, including information representative of the user selected settings in a database. For example, Shopify hosts its user's websites (e.g., at the myshopify.com URL). Shopify uses

memory in the servers it owns or controls to store its users' information, including through the use of mySQL databases. Shopify further stores information in its databases describing the Header, Footer, Sections, themes, and settings of a user's website.

204. The Accused Instrumentalities retrieve said information representative of said one or more said UI object settings stored in said database. Shopify generates a website for its users based on the settings selected by the user during the web design process. The information representative of the user selected settings, which are stored in a database, are then used by Shopify to generate the user's website. For example, Shopify generated the website below including the added image by retrieving the user selections from Shopify's databases.

205. The Accused Instrumentalities build an application consisting of one or more web page views from at least a portion of said database utilizing at least one player. For example, the video section is built from at least a portion of the information in Shopify's databases. The settings selected by the user that are stored in Shopify's databases are used to build the user's web pages reflecting those settings.

206. The Accused Instrumentalities include a player that utilizes information stored in said database to generate for the display of at least a portion of said one or more web pages. For example, player code, such as HTML and JavaScript code, provided by Shopify utilize the information stored in the database for generating at least a portion of a user's web pages on a browser by the browser's engine(s).

207. The Accused Instrumentalities feature a system where the application and player are provided to the device and executed on the device and when the 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. Shopify makes its customers' websites accessible to website visitors' devices. The website visitors connect to the website on their own devices which are supplied the application and player code described above. The website visitors are then able to provide input values (e.g., typed text, mouse clicks, button clicks, touches, swipes, etc.) to an input of the defined UI object associated with an input symbolic name. The website provided by Shopify is designed such that when a website visitor provides input as described above, this input is provided to the web service. For example, when a website visitor clicks buttons, types text, touches a touch screen, or swipes the screen, these inputs are transmitted along with a corresponding input symbolic name to the web service through an HTTP request protocol, such as a POST or GET method call, over the Internet.

208. 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. Based on the received input from a website visitor as described above, the web service generates an output to send to the visitor's browser. For example, when a website visitor clicks to play, start, stop, or Watch later a YouTube video, the YouTube video web service receives an indication of the click (input), and in response generates output values such as video data and result codes, and associated symbolic name.

209. 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 player, described above, receives the output values such as video data and result codes, and associated

symbolic name in an HTTP response from the web service. The player then presents the received output values for display in the UI object. For example, an output of the YouTube web service includes the actual video or associated graphical elements that are then presented within the visitor's browser.

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

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

JURY DEMAND

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

PRAYER FOR RELIEF

WHEREFORE, Express Mobile demands judgment for itself and against Plaintiffs as follows:

A. An adjudication that Plaintiffs has infringed the ’397, ’168, ’755, ’287 and ’044 patents;

B. An award of damages to be paid by Plaintiffs adequate to compensate Express Mobile for Plaintiffs’ past infringement of the ’397, ’168, ’755, ’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 Express Mobile’s reasonable attorneys’ fees; and

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

Dated: September 19, 2019

DEVLIN LAW FIRM LLC

By: <u>*/s/ Timothy Devlin*</u>
Timothy Devlin (No. 4241)
tdevlin@devlinlawfirm.com
Robert Kiddie (*pro hac vice*)
Texas Bar No. 24060092
rkiddie@devlinlawfirm.com
1526 Gilpin Avenue
Wilmington, Delaware 19806
Telephone: (302) 449-9010
Facsimile: (302) 353-4251

James R. Nuttall
Michael Dockterman
Robert F. Kappers
Tron Fu
STEPTOE & JOHNSON LLP
115 South LaSalle Street, Suite 3100
Chicago, IL 60603
(312) 577-1300

Attorneys for Defendant/Counterclaim Plaintiff Express Mobile, Inc.