

CHAPTER 1

INTRODUCTION TO COMPANY

AppDirect is the leading platform for selling, distributing, and managing cloud-based products and services. AppDirect's full service Cloud Monetization Suite allows you to drive revenue growth by powering all of your sales channels, including self-service website checkout, sales teams, re sellers, your marketplace, or third-party marketplaces. After the sale, AppDirect's Cloud Management Suite drives better customer engagement and retention by providing a seamless experience across all your service offerings with unified billing, identity, data, and mobile device management for your customers. Our combined solutions can dramatically reduce your cost of on-boarding, launching, and iterating on new services with a single, highly inter-operable platform. AppDirect provides the technology and expertise to maximize customer lifetime value and help you achieve your cloud business objectives.^[1] Our solutions are open standards based, technology agnostic, and API addressable, which means they easily integrate with existing IT tools and systems. At AppDirect, we help organizations take their cloud initiatives to the next level, not control or limit what already works.

AppDirect powers the cloud marketplaces of trusted companies around the globe—including Vodafone, ATT, Telstra, Rogers, British Telecom, Dustin, Globe Telecom, Deutsche Telekom, Softbank, Samsung, Rackspace, Appcelerator, Telia and more.^[2] It has helped to connect millions of businesses with today's leading web-based applications.

AppDirect is basically a SaaS based company in cloud computing. In the simplest terms, cloud computing means storing and accessing data and programs over the Internet instead of your computer's hard drive. The cloud is just a metaphor for the Internet. What cloud computing is not about is your hard drive. When you store data on or run programs from the hard drive, that's called local storage and computing. Everything you need is physically close to you, which means accessing your data is fast and easy, for that one computer, or others on the local network. Working off your hard drive is how

the computer industry functioned for decades; some would argue it's still superior to cloud computing. For it to be considered "cloud computing," you need to access your data or your programs over the Internet, or at the very least, have that data synced with other information over the Web. The lines between local computing and cloud computing sometimes get very, very blurry. That's because the cloud is part of almost everything on our computers these days. You can easily have a local piece of software (for instance, Microsoft Office 365) that utilizes a form of cloud computing for storage (Microsoft OneDrive).^[3]

Software as a service (SaaS) is a software distribution model in which a third-party provider hosts applications and makes them available to customers over the Internet. SaaS is one of three main categories of cloud computing, alongside infrastructure as a service (IaaS) and platform as a service (PaaS).^[4] SaaS removes the need for organizations to install and run applications on their own computers or in their own data centers. This eliminates the expense of hardware acquisition, provisioning and maintenance, as well as software licensing, installation and support.

AppDirect is providing the best learning experience along with the implementation on live projects. The company is training us from the scratch to the advance part so that the company can use our output. AppDirect is making sure that we are doing our hands on every new technology which they are using. All the latest stack of technologies are trained to us by the respected team leads. After training they give us assignments to make us comfortable and familiar with the final product.

AppDirect's cloud service commerce platform and ecosystem offer providers and developers of cloud services maximum opportunities to make software easily available to their partners and customers from one location. The company acts as a catalyst, helping businesses of any size and vertical go to market effectively in the cloud – enabling this across all cloud models, through multiple sales channels, and across all devices.

AppDirect has also worked in the field of SaaS. It helps customers get the most out of your application in a simple, effective way by offering an add-on store. In doing so, you can make it easier than ever before for users to find solutions that complement your core offering. Look to Increase your customer value proposition by harnessing partner services to form a winning add-on collaboration. You can seamlessly allow customers to install add-ons from an AppDirect powered listing with pre-built connectors to your app. These connectors will let your customers share data between apps they own in your ecosystem. If your clients don't already own an app that ties well to your solution, we give you the option to cross-sell your partners by letting customers buy third-party apps directly from your site. Launching an add-on store will not only delight your customers, but also strengthen your partnerships with services that can then cross-sell your solution on the AppDirect Network.

The company is the cloud service commerce leader making software and products accessible globally. The AppDirect cloud service commerce platform unites providers, developers and consumers of cloud services into a single ecosystem. This makes it easy for businesses to find, buy, manage and monitor cloud services from a central location and delivers new opportunities to distribute, sell, and market cloud services.^[5] AppDirect-powered marketplaces, billing and distribution, and reselling services help providers – including Telstra, ADP, Samsung, Deutsche Telekom and others – connect millions of businesses to solutions from Microsoft, Google, Box and more. AppDirect is headquartered in San Francisco with global offices.

CHAPTER 2

TRAININGS PROVIDED

2.1 GITHUB

GitHub is a web-based Git or version control repository and Internet hosting service. It offers all of the distributed version control and source code management functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project. GitHub offers both plans for private and free repositories on the same account which are commonly used to host open-source software projects. As of April 2017, GitHub reports having almost 20 million users and 57 million repositories, making it the largest host of source code in the world.

Projects on GitHub can be accessed and manipulated using the standard Git command-line interface and all of the standard Git commands work with it. GitHub also allows registered and non-registered users to browse public repositories on the site. A user must create an account in order to contribute content to the site, but public repositories can be browsed and downloaded by anyone. With a registered user account, users are able to discuss, manage, create repositories, submit contributions to others' repositories, and review changes to code.^[6]

My Github user ID is arshdeepsingh22

Learnings :

- Configuring Git
- Adding origin and upstream
- Linux commands
- Adding files
- Committing files
- Pushing files

- Forking
- Cloning
- Branching
- Rebasing
- Squashing
- Merging
- Creating a Pull Request
- Reviewing a Pull Request etc

2.2 CORE JAVA

Java is an object-oriented language similar to C++, but simplified to eliminate language features that cause common programming errors. Java source code files (files with a .java extension) are compiled into a format called bytecode (files with a .class extension), which can then be executed by a Java interpreter. Compiled Java code can run on most computers because Java interpreters and runtime environments, known as Java Virtual Machines (VMs), exist for most operating systems, including UNIX, the Macintosh OS, and Windows. We currently use Java Version 8.^[7]

Integrated Development Environment(IDE) used :

- IntelliJ IDEA

Learnings :

- Variables
- Loops
- Constructors
- Data Types
- Wrapper Classes
- AutoBoxing Unboxing concepts
- Data Structures
- Strings

- Array-lists
- Generics
- Threading
- File Handling
- Exception Handling
- Servlets
- JDBC
- Inheritance
- Polymorphism
- Function Overriding
- Function Overloading
- Abstraction
- Interfaces
- Encapsulation etc

2.3 PARTNER INTEGRATION IN APPDIRECT

- To add a new product to marketplace
- Getting access of superuser
- Creating a new app on marketplace
- Purchasing an application

2.4 HTML

HyperText Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS), and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.^[8]

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. We currently use HTML5 in our designs.

Learnings :

- Elements
- Styles
- Headings
- Paragraphs
- Comments
- Images
- Lists
- Links
- Tables
- Classes
- Responsive
- Inputs
- Forms etc

2.5 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML, the language can be applied to any XML document, including plain XML, SVG, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content. Styles define how to display HTML elements. Styles are normally stored in Style Sheets. Styles were added to HTML 4.0 to solve a problem.^[9] External Style Sheets can save you a lot of work. External Style Sheets are stored in CSS files. Changes to the graphic design of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in the CSS file they use, rather than by changing markup in the documents.

Learnings :

- Colors
- Margins
- Font
- Border
- Background
- Position
- Float
- Display
- Height/Width
- Max/Min-Width
- Max/Min-Height
- Align
- Transform
- Opacity
- Text-transforms
- Overflow etc

2.6 SCSS

The SCSS (Sassy CSS) is an extension of CSS syntax. This means every valid CSS is a valid SCSS as well. SCSS makes much easier to maintain large stylesheets and can recognize vendor specific syntax. Many CSS and SCSS files use the extension .scss. Indented – This is older syntax and sometimes just called as SASS.

Since SCSS is a CSS extension, everything that works in CSS works in SCSS. This means that for a Sass user to understand it, they need only understand how the Sass extensions work. Most of these, such as variables, parent references, and directives work the same; the only difference is that SCSS requires semicolons and brackets instead of newlines and indentation. SCSS supports both comments that are preserved in the CSS output and comments that aren't.^[10] However, SCSS's comments are significantly more flexible. It supports standard multiline CSS comments, which are preserved where possible in the output. These comments can have whatever formatting you like; Sass will do its best to format them nicely.

Learnings :

- Nested Rules
- Referencing Parent Selectors
- Nested Properties
- Placeholder Selectors (@extend directive)
- @import etc

2.7 JAVASCRIPT

JavaScript is an object-based scripting language that is lightweight and cross-platform. JavaScript is a programming language that adds interactivity to your website (for example: games, responses when buttons are pressed or data entered in forms, dynamic styling, animation). JavaScript itself is fairly compact yet very flexible. Developers have written a large variety of tools complementing the core JavaScript language, unlocking a vast amount of extra functionality with minimum effort. JavaScript is not compiled but

translated. The JavaScript Translator (embedded in browser) is responsible to translate the JavaScript code.^[11]

Although there are strong outward similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two are distinct languages and differ greatly in their design. It is mainly used for: Client-side validation, Dynamic drop-down menus, Displaying data and time, Displaying popup windows and dialog boxes (like alert dialog box, confirm dialog box and prompt dialog box), Displaying clocks etc.

Learnings :

- Variables
- Comments
- Arrays
- Assignment
- Data Types
- Functions
- Objects
- Strings
- String Methods
- Date
- Date Functions
- Loops
- Array Methods
- Array Sorting
- Conditions
- Switch
- Debugger
- DOM
- Reg-Exp etc

2.8 **jQuery**

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.^[12]

Learnings :

- Add/Remove CSS classes
- Traversing
- Ancestors
- Descendants

CHAPTER 3

ASSIGNMENTS COMPLETED

3.1 SHOPPING CART

This was an assignment based on Core Java which was based on Model View Control pattern and was to design a shopping cart and finally a bill was to be created and converted to Comma Separated Value(CSV) file. Same as in shopping website there were products in a database, the quantity of them, cost of each item etc.

3.2 JAVA TECH CHALLENGE

The aim of the project is to integrate any Java Based Web Application with AppDirect company marketplace using AppDirect APIs. This application exposes REST APIs that perform CRUD on a resource pertaining to your application. This project aims at integrating “Create” and “Cancel” subscription functionalities for application in the marketplace.^[16]

This project has got several modules to be implemented. The ISVs or the developers registration, creation of product on the marketplace, creating the endpoint URLs, testing the integration report, etc.

Following is the way we went about the tech challenge:

Creation of Product on Marketplace

1. Register as a developer at AppDirect: <https://www.appdirect.com/developers/register>
2. Create a new product as a web product
3. Define the revenue model for the product
4. Define the description and overview of the product.
5. Add Subscription Create Notification URL
6. Add Subscription Cancel Notification URL

Creation of Servlets for each subscription notification

1. Read the request passed through `HttpServletRequest`
2. Initialize the response object with the status code 200 for success and UUID for the subscription
3. Create a `ObjectMapper` class object to transform response object in JSON
4. Respond with the response object on the same url
5. Create a `HttpURLConnection` object with OAuth signature on the eventUrl passed.
6. Parse the `HttpURLConnection` object to the POJO class object

Creation of the Subscription notification services

1. Create & Cancel Subscription Services

Log the notification event triggered into the database.

Create the DAO class object for the account creation & deletion interaction with the database.

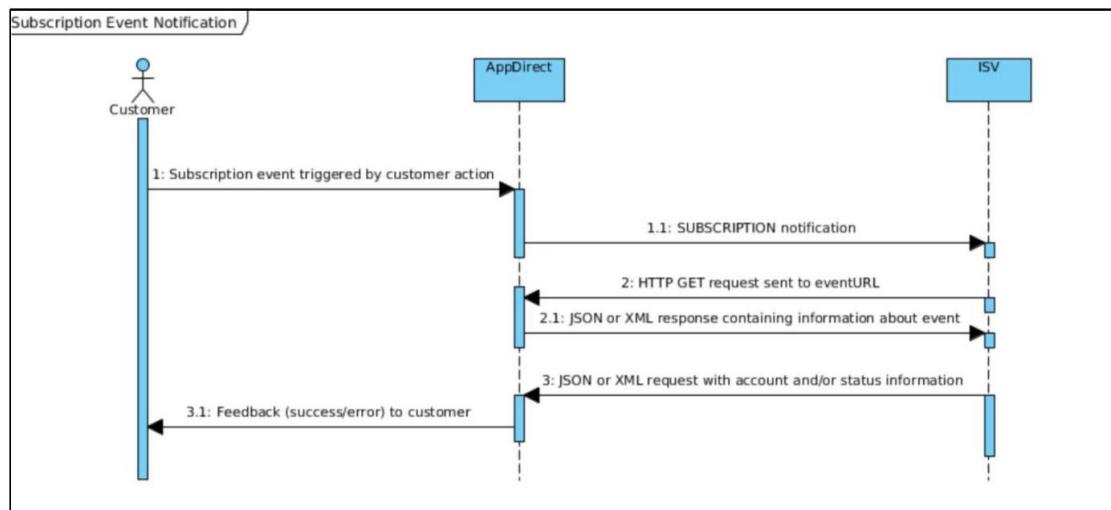


Figure 3.1 Subscription Flow of Tech Challenge

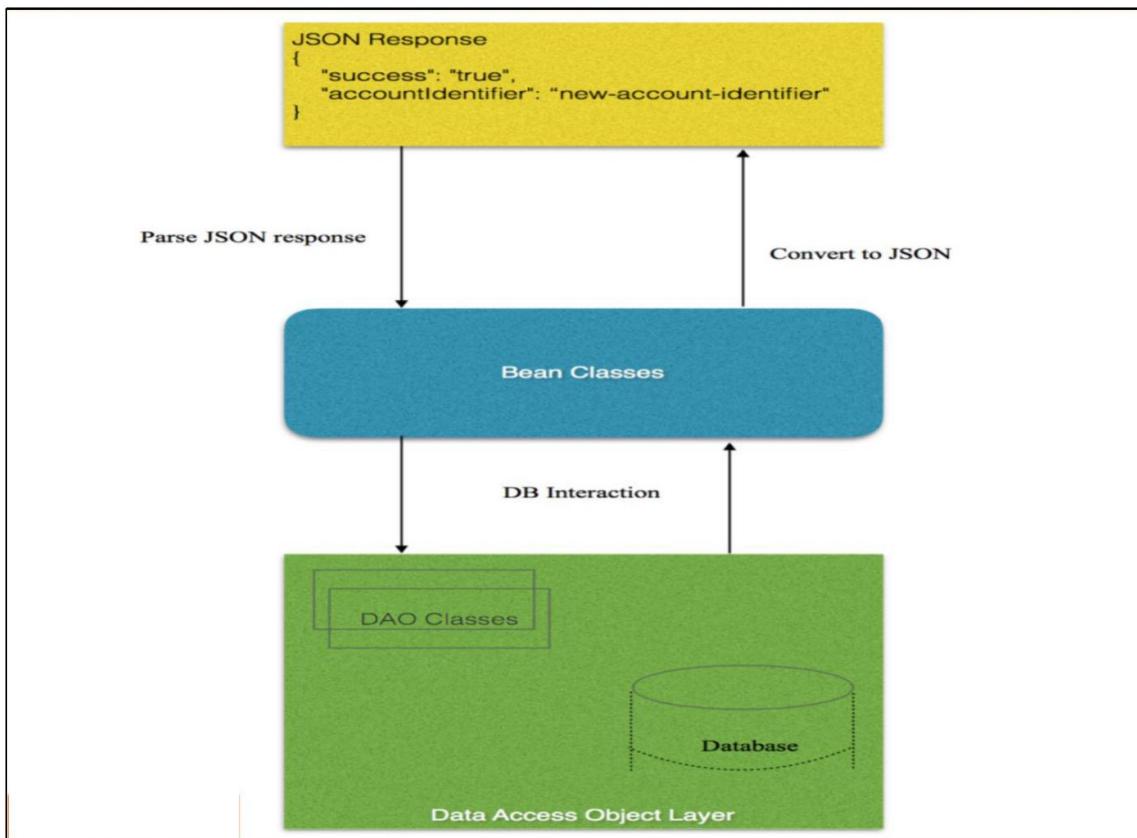


Figure 3.2 Data Flow of Tech Challenge

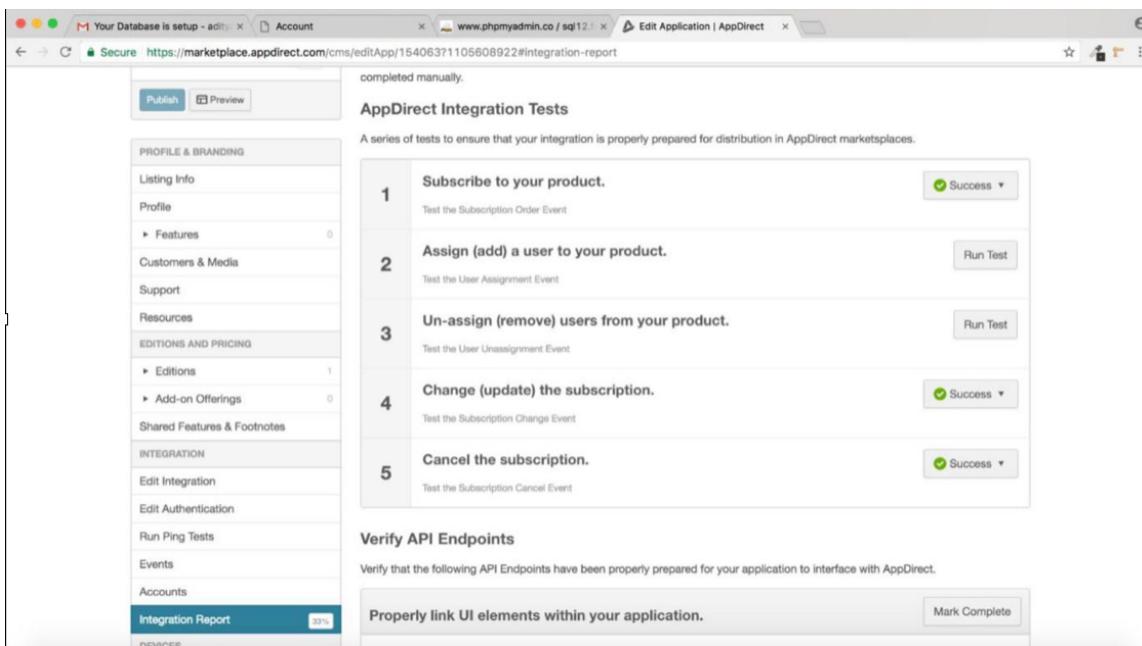


Figure 3.3 Integration Tests of Tech Challenge

The screenshot shows the phpMyAdmin interface for a database named 'sql12169563'. The left sidebar lists tables: 'creatorsib' and 'sublog'. The main area displays the results of a SELECT query:

```

SELECT *
FROM `sublog`
LIMIT 0, 30

```

The results table has columns: SubscriptionType, accountIdentifier, and logtime. There are three rows of data:

SubscriptionType	accountIdentifier	logtime
SUBSCRIPTION_ORDER	1e39fb03-dfea-473f-b2b9-495989cb8828	2017-04-17 22:36:26
SUBSCRIPTION_CANCEL	1e39fb03-dfea-473f-b2b9-495989cb8828	2017-04-17 22:38:56
SUBSCRIPTION_ORDER	b91e916a-c93-477a-a055-ceee546ae089	2017-04-17 22:40:31

Figure 3.4 Run Ping Test Results

3.3 UI ASSIGNMENTS

HTML : To design web pages.

CSS : To use the same styles of designing in different file.

SCSS : Using different styles for different containers/divs.

JavaScript : Some basic codes were given by which the page could become more interactive.

DOM, jQuery : A form was given to design in which jQuery library and DOM functions could be used.

3.4 UI TECH CHALLENGE

Design a website and use the Twitter API to display three columns containing the last 30 tweets from @AppDirect, @laughingsquid and @techcrunch or any 3 users. Each tweet should include following:

The tweet content

A well-formatted created_at date

A link to the tweet

For retweets and mentions, the username should be included. Make an edit layout view that has a form to change the layout settings.

Use LocalStorage to persist and load the layout settings.

Configurable settings like the order or columns, the time range of the tweets shown, number of tweets in each column and overall palette/skin of the page.

The “edit layout” panel can appear either on the same page as the tweets page, on its own page or embedded within tweets layout. There should be a straightforward way to toggle between edit and view modes, and it should be clear to the user which mode they are currently in.

Ensure a good responsive experience on mobile phones and tablets.

Use an interaction (like drag and drop) instead of a form field to order the columns.

The screenshot shows a "Tweet Explorer" interface with three columns for users AppDirect, LaughingSquid, and TechCrunch. Each column displays a list of tweets with their respective user icons, names, and tweet content. The interface includes a header with tabs for "Dashboard" and "Settings".

User	Tweet Content	Date
AppDirect	#SaaS Go-to-Market Success: The Growing Importance of Self-Service Sales. Learn more: https://t.co/2eTfglqnj	May 18th 2017, 2:36:46 am
AppDirect	AppDirect is excited to partner with @bluegrasscell https://t.co/huOahSxMkh #CloudCommerce	May 16th 2017, 9:16:17 pm
AppDirect	@AppDirect_6d Co-CEO @danielsaks talks to @xconomy about Amazon, the growth of cloud service commerce, and partner @bluegrasscell https://t.co/cyPHkWWf1U	May 12th 2017, 9:17:39 am
AppDirect	RT @connectedhub: Monetization in automotive is not different from the telecom market, it's a large ecosystem, explains Mercier @AppDirect...	May 12th 2017, 12:51:32 am
LaughingSquid	Russian Artist Masterfully Sculpts Hauntingly Lifelike, Fully Articulating Dolls From Polymer Clay... https://t.co/kXWBHrOx4f	May 18th 2017, 7:33:05 am
LaughingSquid	A Paper Airplane Thrown Out of a Window Flies Back Into the Same Window After an Improbable Flight... https://t.co/usPDVfArjz	May 18th 2017, 7:03:02 am
LaughingSquid	Dutch Filmmaker Intentionally Lets His Phone Get Stolen In Order to Follow Its Path Through Spyware... https://t.co/Q7i2ZWAniz	May 18th 2017, 6:33:03 am
LaughingSquid	Guzzle Buddy, A Handy Device That Transforms an Entire Bottle of Wine Into a Personal Drinking Glass... https://t.co/f4azuxXTOM	May 18th 2017, 5:33:04 am
TechCrunch	Google's Instant Apps are now generally available to all Android developers https://t.co/FGbQRKoAj0 #IO17	May 18th 2017, 4:05:04 pm
TechCrunch	Google's TensorFlow Lite brings machine learning to Android devices https://t.co/1BDVsEcp2v #IO17	May 18th 2017, 3:48:04 pm
TechCrunch	Google quietly debuts Chatbase, a chatbot analytics platform https://t.co/ZTyUXD030 #IO17	May 18th 2017, 2:29:00 pm
TechCrunch	Google makes Kotlin a first-class language for writing Android apps https://t.co/dBteFNRNis #IO17	May 18th 2017, 1:56:02 pm
TechCrunch	Lucas VonCranach and Ijad Madisch to speak at TechCrunch Disrupt Berlin https://t.co/bRS9g90fZv by @mikebutcher	May 18th 2017, 12:45:00 pm

Figure 3.5 Tweets fetched from 3 Twitter users

CHAPTER 4

THEME CUSTOMIZATION

4.1 INTRODUCTION TO PROJECT

Technology has really brought in the change and made everything much easier and quicker. Working on both back-end and front-end has increased immensely; with people equally inclined to both the technologies, but when the client/customer is able to see the output visually, the satisfaction level is more as compared to when the project simply runs at the backend. The project is titled “Mid-tier Theme Customization” and it is the front end project of AppDirect. The aim of the project is to design AppDirect-powered marketplaces with a vision of designing the marketplaces with same look and feel.

AppDirect, let the customers find, buy and manage cloud services of today, tomorrow. AppDirect have multiples clients and each client have their own ‘look and feel’ requirements. Marketplace designed by AppDirect - code is monolith. For any changes in features, ISVs have to be dependent on AppDirect. AppDirect hence solved their problem with ‘Theme Customization’. It allows developers, ISVs to customize and add controls to the “appearance” that is “Customize the screen”. It allows to add more powerful, interactive customization option under look like color scheme or widgets, button’s color, font choice and under feel, the movement and response of dynamic components like dropdown menus, buttons, forms can be customized. This is the project which I am currently working on to customize the themes. Starting from the header, first of all the logo has to be matched with the respective ISV or any other reseller website in order to give a look and feel of the same website. This is because as this cloud marketplace has to be on the website of ISV or any other reseller. After logo, the full header has to be matched which consists of primary and secondary headers and sometimes even tertiary too. After this, applications from Network Catalog of AppDirect are added into the marketplace. Then comes the designing part, designing banners etc on the home page of marketplace. After all these things comes time for footer. Footer has to be changed as per requirements of the vendor. Sometimes the footer is same as of

AppDirect India to show powered by company and sometimes footer has to be changed that resembles with the original site of vendor.

I started with the my Jira tickets which contain the marketplaces by AppDirect to give them the same UI UX of the the company site so that we can map up our marketplaces with their original sites. Every marketplace we create goes to the production state after going through the code review and then the top mentors give it looks good to go.

4.2 SCOPE

The scope of Mid-tier Theme Customization is designing various kinds of marketplaces such as:

1. Basic Mid-tier Theme/Custom Theme:- When we do this type of customization, only the logged-out version of marketplace covering both the header and the footer are designed.
2. Semi Powerplay:- Whenever the task of semi powerplays is assigned then the existing website of the company is kept in reference and exact features of the website are tried to be replicated with customization to be done for logged-in, logged-out, checkout and body.
3. Powerplay:- In this, full customization of the marketplace is done visiting each and every page, along with matching all the features as depicted in the company's website, covering every minute detail, along with adding banners and sliders.

4.3 BACKGROUND

This project has significant amount of ground work. AppDirect has different clients and each of the client has it's own requirements. Earlier, the clients wanted somewhat modified outputs, but now-a-days clients demand full customization for their marketplaces. This requires a lot of resource and huge amount of power in order to fully customize because full customization requires in depth knowledge. So, in order to

overcome this problem, tool was created so that the clients can themselves modify and customize like changing their own colors in marketplaces etc. But, those changes too require some prior knowledge so we do that for the customers (ISV's , banking, telephonic and Digital Goods).

Even in this tool, later on the coding is simplified and for making any changes only a little code needs to be modified instead of writing complete code.

The major outcome of this project is “Productivity Enhancement”. Not only the designing and the customization part; in case if the customer wants to get it validated, no separate cost is incurred and we deliver the final output only after the marketplace goes through the complete phase of production which is: In progress, In Code Review, QA Validation, Awaiting VTM and then finally when the complete marketplace is ready then Merged.

In order to make just a few changes, lot of files and significant amount of code was required to be changed and whenever deadlines existed along with the marketplace then it became a tedious job to finish the task.

4.4 OTHER PROJECTS WORKED UPON

i. PSFE BACKLOG BUG FIXES

In addition to designing the marketplaces either mid-tier or powerplays we were also fixing the issues in the UI/UX for the current clients. These can be any issues like the improving buttons, fixing text misalignments, coding practice issues, script issues etc. were also assigned to us as a part of our job.

Gulp tasks, Node.JS work to improve the efficiency and performance of designing marketplace was also assigned as a part of work.

Changing the Theme Setup, Improving the UI where marketplaces are created on our local setup to optimize the time.

ii. MICROSERVICES

Under this project, we also worked on ‘Translations’ (the languages which appear on the websites) based on partner, variation and locale. Keys were provided and the respective translations of them was given based on partner, variation and locale.

iii. ADDING ROLE CAPABILITIES FOR USERS

Developer, channel settings etc which were not to be shown unless a user has certain roles/capabilities. So to correct these changes this project was started.

iv. V1-V2 MIGRATION OF THEMES

Old clients whose themes are still in V1, have to be converted in V2. So I was also assigned with the task of V1-V2 migration of theme.

CHAPTER 5

ELABORATION OF PROJECT

5.1 THEME CUSTOMIZATION

Before starting with development of Theme Customization, the figure below describes each section and subsection in a marketplace.

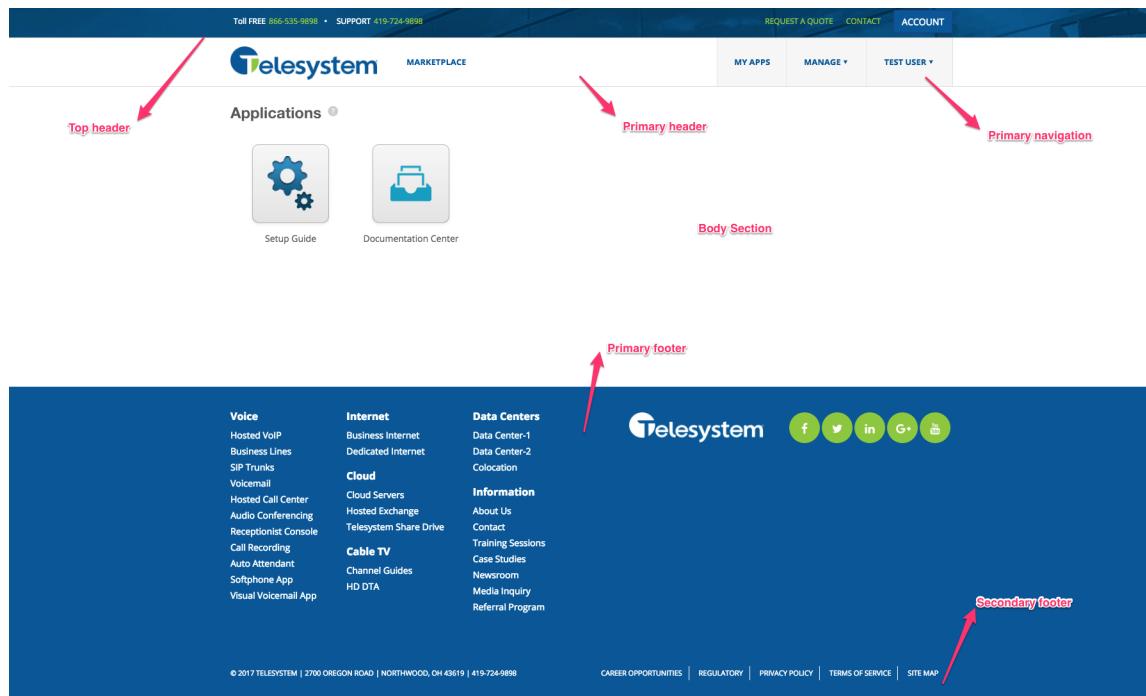


Figure 5.1 Different Sections of Marketplace

The complete project has been implemented using various technologies such as:-

HTML5

CSS3

SASS

GULP :

It is a task runner or build process which helps in:-

- generating HTML from templates and content files
- compressing new and modified images
- compiling Sass to CSS code
- removing console and debugger statements from scripts
- code linting and validation
- concatenating and minifying CSS and JavaScript files
- deploying files to development, staging and production servers

NODE JS :

It is a JavaScript-based framework/platform and is open source and completely free. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in Web applications with many input/output operations.

NPM :

npm is a package manager for the JavaScript programming language. It is the default package manager for the JavaScript run time environment Node.js. It consists of a command line client, also called npm, and an online database of public packages, called the npm registry.

DEVELOPMENT OF THEME CUSTOMIZER

Theme customizer development can be started by creating folder/file structure manually or by copying the structure of starter theme-”FatHeader” from customizer repo. The folder should contain default folder and inside default folder there should be assets folder and html files. See the screenshot below.

In each of these, three things are surely to be done. Assets folder contains scss/components files, any styling related changes has to be done through scss files. css.css file should not be modified manually.

- **Generating Headers :**

There are two types of header files in default folder logged-in.html and logged-out.html which represents the MP headers structure.

These files will have predefined structures for top header (optional, remove it if not needed), primary header and primary header . We need to code header html in appropriate blocks only.

- **Generating Footers :**

Footer.html is used for coding footer content of MP. Like headers this file also have some predefined structures primary footer and secondary footer (optional, remove if not needed). Footer html should be written in appropriate footer blocks only.

- **Styling :**

TC contains bundles all styling in css.css file. This files should never be updated manually. Always use scss files for styling application which gets compiled through gulp task in TC and generates css.css.

I) header.scss : All header related styling (both for logged-in and logged-out) should go in this file.

II) footer.scss : Footer styling go in this file.

III) _dropdown-menus.scss : Any styling related to drop down menu goes here.

IV) _INDEX.scss : This file only used to import all scss files.

V) _vars.scss : This file contains all scss variables or mixings across TC.

VI) _body-customization : If we need styling to override MP body section and any custom styling which is not applicable from TC UI (channel/customizer - themes tab), should go in this file.

Sometimes, we need third-party fonts, icons or styles etc. We can add reference of third-party styling, fonts or icons in footer.html.^[16]

TESTING CHANGES

Once the changes are done then we can test our changes as well on actual MP by importing the TC bundle or go to channel/customizer and paste appropriate file contents on “Templates” tab and click on publish button.

IMPORT/ EXPORT OF A THEME USING THE THEME CUSTOMIZER

- EXPORTING A THEME**

When you click on ”Export Data” button the customizer will

1. Bundle all of your code in a folder named after the default language of the marketplace
 - a. logged-in.html
 - b. logged-out.html
 - c. checkout.html
 - d. footer.html
 - e. css.css
2. Extract any keys and values that you entered into the Internationalization tab into a JSON file.
3. Extract all of the custom values the theme tab into a JSON file. (This file is named design-properties.json.)
4. Extract any custom fonts for the fonts tab.

All of these files will be in theme .zip on your computer. You can use this files to edit the theme, make any changes needed, including updating, or adding keys and values to the JSON files.

• IMPORTING A THEME

1. Select all of the files that are required and compress them into a .zip file.
2. Click on Unpublish in the Templates tab.
3. Click on Publish in the Templates tab.
4. Choose the zip file in the interface.
5. Hit Upload.
6. All of the fields will be populated with all of customization values and code.
7. Click on Unpublish in the Templates tab.
8. Click on Publish in the Templates tab.
9. All changes should now be live and visible in the marketplace.

Earlier, there were many files which were to be changed, but after the implementation of the tool then only 5 files (logged-in.html, logged-out.html, checkout.html, footer.html, css.css) and the scss folder need to be modified. In those files, only a few pre defined variables need to be updated as per the requirement of the marketplace. For a deeper view, let's take an example:-

As the marketplace outlook will remain the same for all the clients therefore certain features such as header height won't change too often. Now, the header comprises of the logo of the company and a few links, along with the dropdowns. Also, the size varies for the logo from company to company. Therefore, every time it becomes quite tedious to measure the height of the logo and adjust the height of the header accordingly. For which a variable is defined which in turn calls the function whose job is to compute the height with respect to the logo height.

Hence, along with doing the customization, the process of doing so is also simplified. Moreover, we also did the changes in the other properties, like in buttons and links on the website.

Firstly, we used to do this task using UI, by just filling out the fields for features such as Button Border Color, Buttons Background, Buttons Hover Background, Buttons Active Background etc.

And then we need to publish it; download the zip file; make changes in the local folder and then again upload the zip to check the outcome. But now even this task is more simplified, as we can directly make changes in the “design-properties” file which are reflected directly on the localhost.[16]

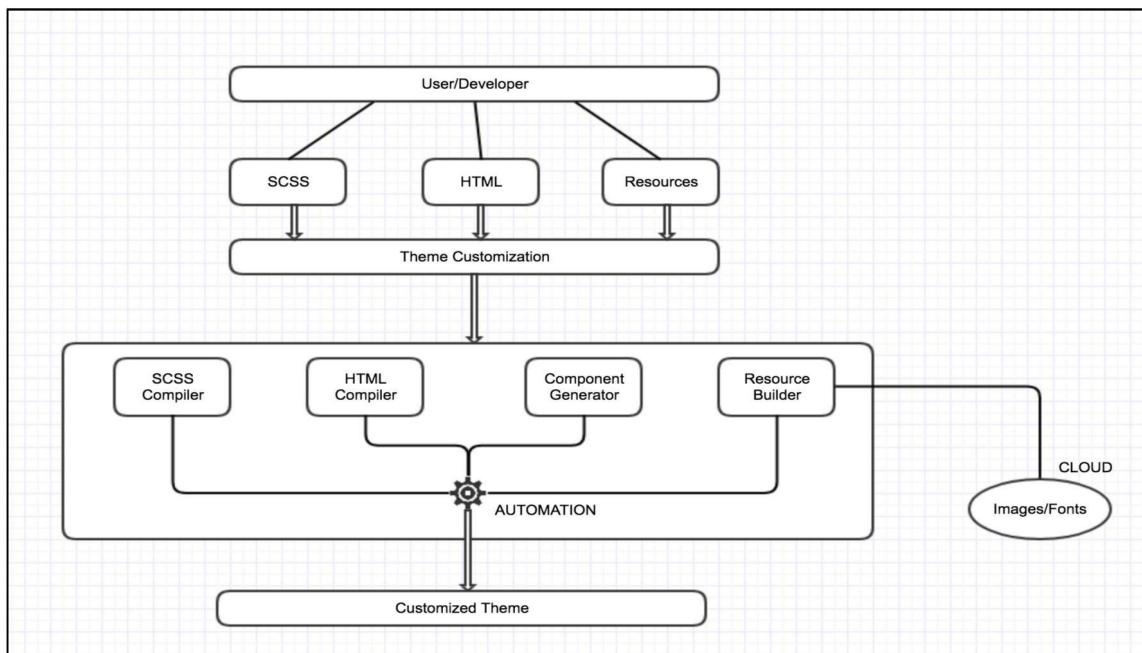


Figure 5.2 Block Diagram of Theme Customizer

The Theme Customizer localhost platform is also changed to a newer version of which the screenshots are attached on the next page.

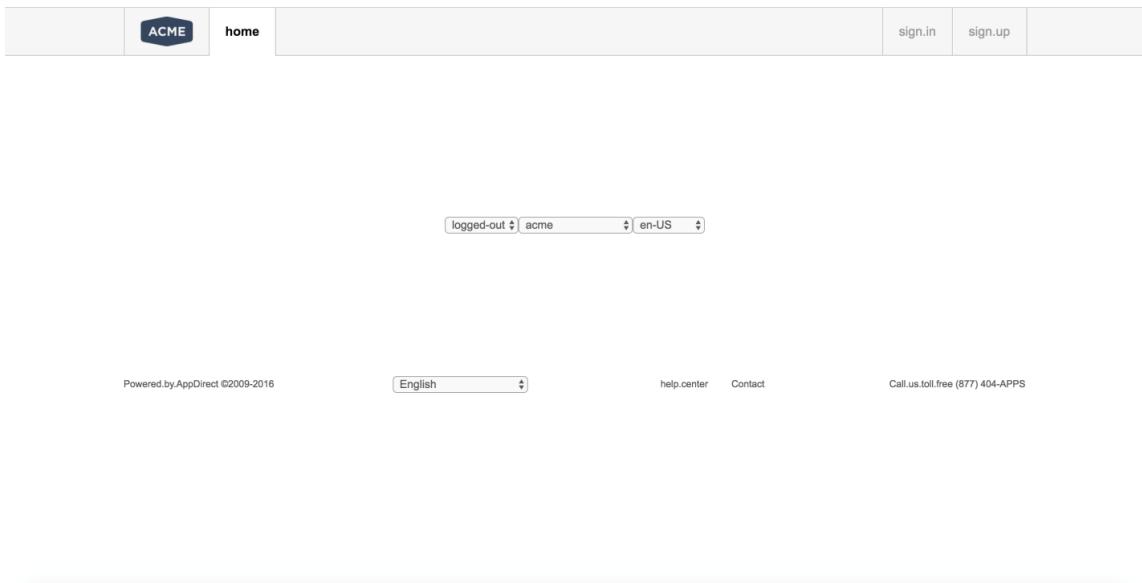


Figure 5.3 Screenshot of old version of Theme Customizer

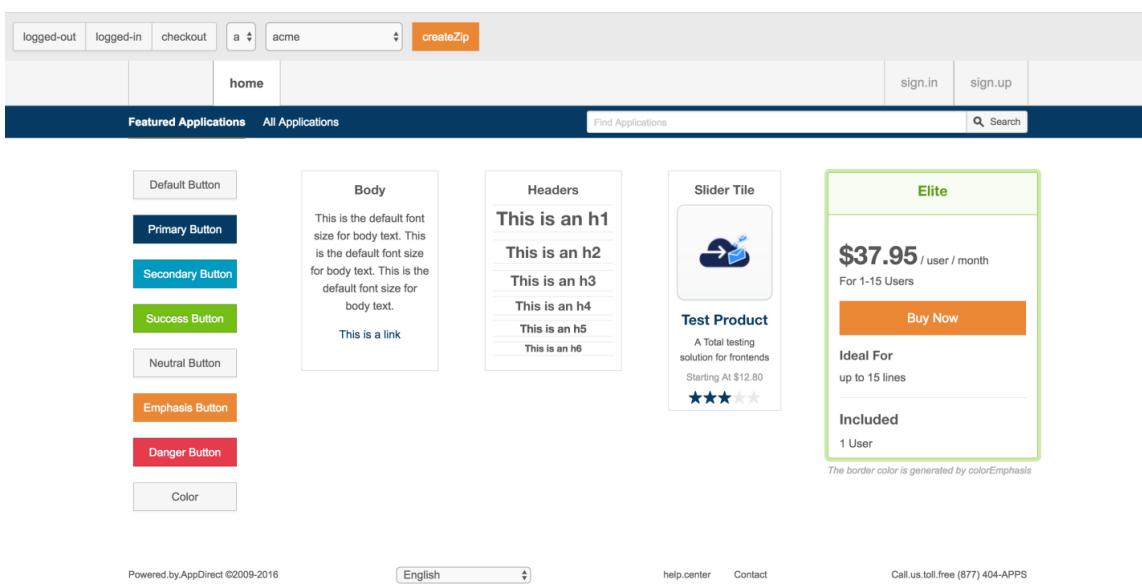


Figure 5.4 Screenshot of current version of Theme Customizer

Full Data Flow Diagram of Theme Customization is shown on the next page.

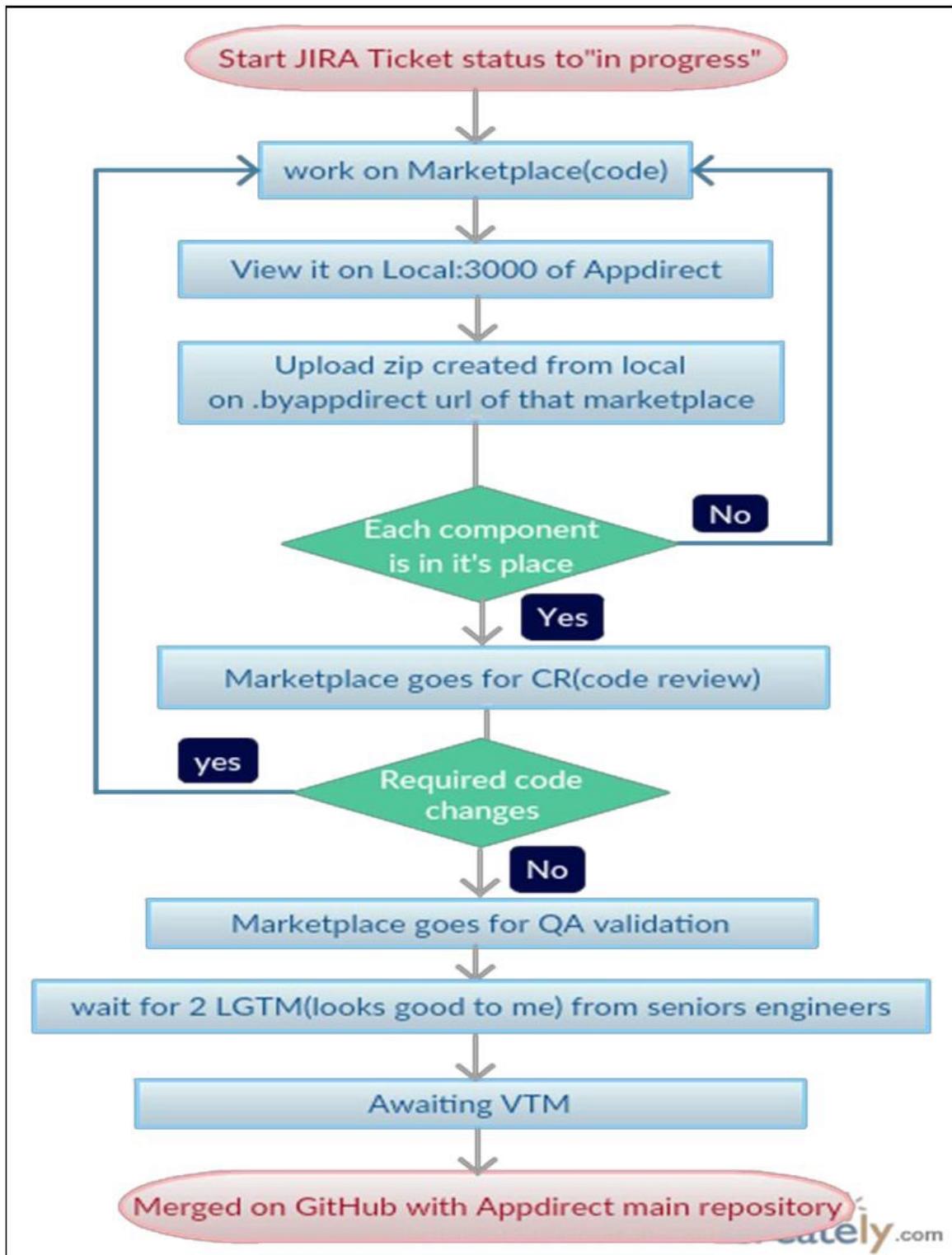


Figure 5.5 Data Flow Diagram of Theme Customizer Project

5.2 PSFE BACKLOG BACKLOG ISSUES

A backlog is a buildup of work that needs to be completed. It is a list of the outstanding user stories, bugs and features for a product (Product Backlog) or sprint (Sprint Backlog). In our project backlog is mainly refer to a bug which is there in marketplace of our clients. The bug may be of any kind some of them are - there is a drop down menu in marketplaces in which different languages are there and sometimes bug is like on selection of particular language, the language of the whole page does not get changed in that particular language, so to do so we need to fix the bug by doing research first and then we fix the bug. Other kind of bug is sometimes on hover or on active color of link is not changed or we can say it's not recognizable so to make if working fine we need to do some changes in code and make it working. One of the major backlog is on click of a particular button pop-up button does not come up. Code base of all the themes is same but if a particular marketplace needs.

We work for PSFE team as well for resolving backlog issues for different repository. We have different Repository for each project some of them are Theme Customization , Customizer Theme and one is Called as monolith or AppDirect Repository.

In AppDirect and Customizer theme repository we came across many issues which are reported by support team or by the Marketplace admin or client itself report it.

Issues or Bugs in UI can be of priority type Major , Minor , Blocker , Trivial ,Critical .

Blocker and Critical are the most high priority issues and must be handle with great care and should be completed on time, blocker means that without this work of particular channel can be affected and must be fixed ASAP.

Critical is time oriented and at the same time is very important with high priority.

Major is also like critical but time is not the limit in this.

Trivial is the low priority task.

Backlog issues can be channel specific or it can be of Base MARKETPLACE FOR APPDIRECT also.

Backlog issue once fixed will has to go through Series of step like

Dev-> Bug is fixed

CR ->(Code Review) on this we need to have at least 2 LGTM to get it approved and approved for QA to test it

QA-> Even QA has three phases to test the bug fixed before getting it to Awaiting VTM.

QA on ORCHARD

QA on TEST environment

QA on Production

Awaiting VTM -> After QA verified the Ticket/PR it went to Awaiting VTM phase , that mean asking for merging the changes to Repository

Merged-> Each backlog issue are merged in specific Build cycle of AppDirect, we need to target the Build in which we need to merge are ticket and work accordingly

We also need to send across the email for the bug fixed and tell the QA to have a Validation in the format for sending Issues to QA for testing is:

To

<your-teams-alias>@appdirect.com

CC

PM/TPM

Subject

Validation: {Branch Name}

Body

Branch Name: (name of the branch)

Release: (number of the release branch, the feature needs to be in)

Blocker for the release: (Y/N)

Jira Ticket: (url of the jira ticket, if there is one)

Pull request: (url of the github pull request)

Marketplace: (ALL- if it is a common feature)

Features: (list of features introduced in the branch)

Sections Updated/Affected: (eg: user management, app management, listing, etc.)

Special Instructions: (if applicable)

Branch was tested locally: (Y/N)

DB updated: If there is a change log associated, mention in the pull request, so that he verifies the sql.

After this it is validated by the QA and reported accordingly.

QA test that whether the Bug fixed is not affecting other channels or base channel and this is QA responsibility. To test it on each environment that AppDirect has.

5.3 MICROSERVICES

To start explaining the microservice style it's useful to compare it to the monolithic style: a monolithic application built as a single unit. Enterprise Applications are often built in three main parts: a client-side user interface (consisting of HTML pages and javascript running in a browser on the user's machine) a database (consisting of many tables inserted into a common, and usually relational, database management system), and a server-side application. The server-side application will handle HTTP requests, execute domain logic, retrieve and update data from the database, and select and populate HTML views to be sent to the browser.^[13] This server-side application is a monolith - a single logical executable. Any changes to the system involve building and deploying a new version of the server-side application.

Such a monolithic server is a natural way to approach building such a system. All your logic for handling a request runs in a single process, allowing you to use the basic features of your language to divide up the application into classes, functions, and namespaces. With some care, you can run and test the application on a developer's laptop, and use a deployment pipeline to ensure that changes are properly tested and deployed into production. You can horizontally scale the monolith by running many instances behind a load-balancer.

Monolithic applications can be successful, but increasingly people are feeling frustrations with them - especially as more applications are being deployed to the cloud . Change cycles are tied together - a change made to a small part of the application,

requires the entire monolith to be rebuilt and deployed. Over time it's often hard to keep a good modular structure, making it harder to keep changes that ought to only affect one module within that module. Scaling requires scaling of the entire application rather than parts of it that require greater resource.^[14]

These frustrations have led to the microservice architectural style: building applications as suites of services. As well as the fact that services are independently deployable and scalable, each service also provides a firm module boundary, even allowing for different services to be written in different programming languages. They can also be managed by different teams. A service-oriented architecture is essentially a collection of services. These services communicate with each other.

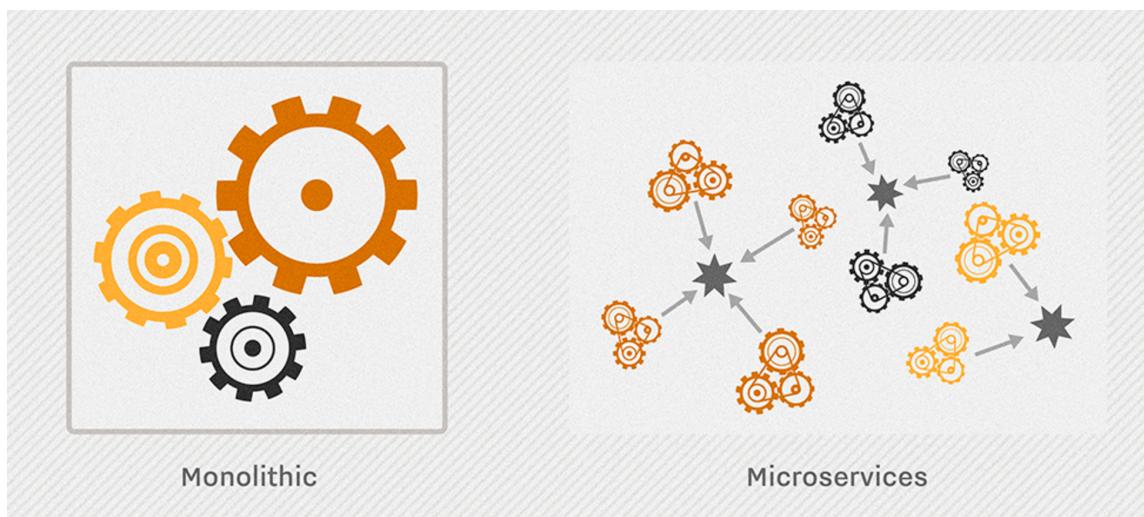


Figure 5.6 Monolith vs Microservices deployment example

Among their advantages for software development, microservices:

- Are easily deployed.
- Require less production time.
- Can scale quickly.
- Can be reused among different projects.
- Work well with containers, such as Docker.
- Complement cloud activities.

However, there are also drawbacks with microservices, such as:

- Potentially too granular;
- Latency during heavy use; and
- Testing can be complex.

In this project, microservice created gave translations of keys given in the JSON object sent. The need for this project was that the localization was not present for the new marketplaces that we were creating the theme customization project. So we created this project using Node.JS. A JSON object was sent using postman(chrome extension) and which contained keys, locale(language), partner, variation.

The flow in the project :

1. The server was created initially which contained on which port the server should run.
2. The server redirected to routes folder where the given action on post request was defined that it will first validate and then further actions.
3. This route is basically /translate and gets activated on post request made by client.
4. The post request first checked the validations on the request object using the validate function which checked that were there keys in the object or not.
5. Then the translation function was made which returned a JSON object which contained filename and the respective translations of keys in that file.
6. The JSON received from the above translation function was then iterated upon to find the translation of keys sent in the request object.
7. Translation of keys JSON object was sent as response in the end.

With the use of AppDirect repository gives the translation of keys based on locale, partner, variation. If only keys are given the keys are translated with en locale. If locale is given it gives translation on the basis of that respective locale. If partner is given the respective keys are translated on the basis of the given parameters.

Microservices leans toward business enhancements in order to improve the overall product, not just the project at hand. In monoliths, work is complete once the project is finished. In microservices, work is complete over the span of the lifetime of a product. Companies such as Amazon and Netflix employ this business model. Micro Services style is usually organized around business capabilities and priorities. Unlike a traditional monolithic development approach, where different teams have a specific focus on, say, UIs, databases, technology layers, or server-side logic, Micro Services architecture utilizes cross-functional teams.^[15] The responsibilities of each team are to make specific products based on one or more individual services communicating via message bus.

This will be helpful as there are many keys while designing a marketplace in Theme Customizer. So these keys are such as sign.in, sign.up which have translations in the translations.json file. These keys can be used in different languages if they are used in the monolith code but in the localhost where MarketPlaces are designed, the translations cannot be made as there is no internationalization/localization i10n/l10n object by which the keys can be translated whereas this object is present in the monolith code.

So to get this translations and the themes can be designed in proper translations and to improve the localizations/internationalizations this project was made. So that if any other new theme was to be made in any other language except EN, it could be made easily. If there is insertion in translation of current clients, also these can be tested by this project.

5.4 ADDING ROLE CAPABILITIES FOR USERS

There were certain terms used in the marketplace such as developer, channel settings etc which were not to be shown unless a user has certain roles/capabilities. So I was assigned the task to change the functions in AppDirect repository and correct these changes so that exactly similar values are shown of these keys if a user has certain roles/capabilities.

5.5 V1-V2 MIGRATION OF THEMES

Basically V1 is Version 1 of AppDirect and V2 is Version 2 of the company.

The themes page we are working on is currently V2 of AppDirect. Whereas there are certain old clients whose themes are still in V1, So these have to be converted in V2. So I was also assigned with the task of V1-V2 migration of theme. Under this project, we also worked on ‘Translations’ (the languages which appear on the websites).

- What's Different in v2 Theme?**

Search bar on top

Standardized categories style

Sliders

- Left-right navigation buttons moved from edges to above the tiles
- View All button added

Internal Theme Editor Tool

I did V1-V2 of Elisa partner of AppDirect.

As stated there were many changes in V2. Figure on the next page shows what things were changed in the newer version of the company.

In the figure on the next page illustration is given what is different in V2 than V1 of AppDirect.

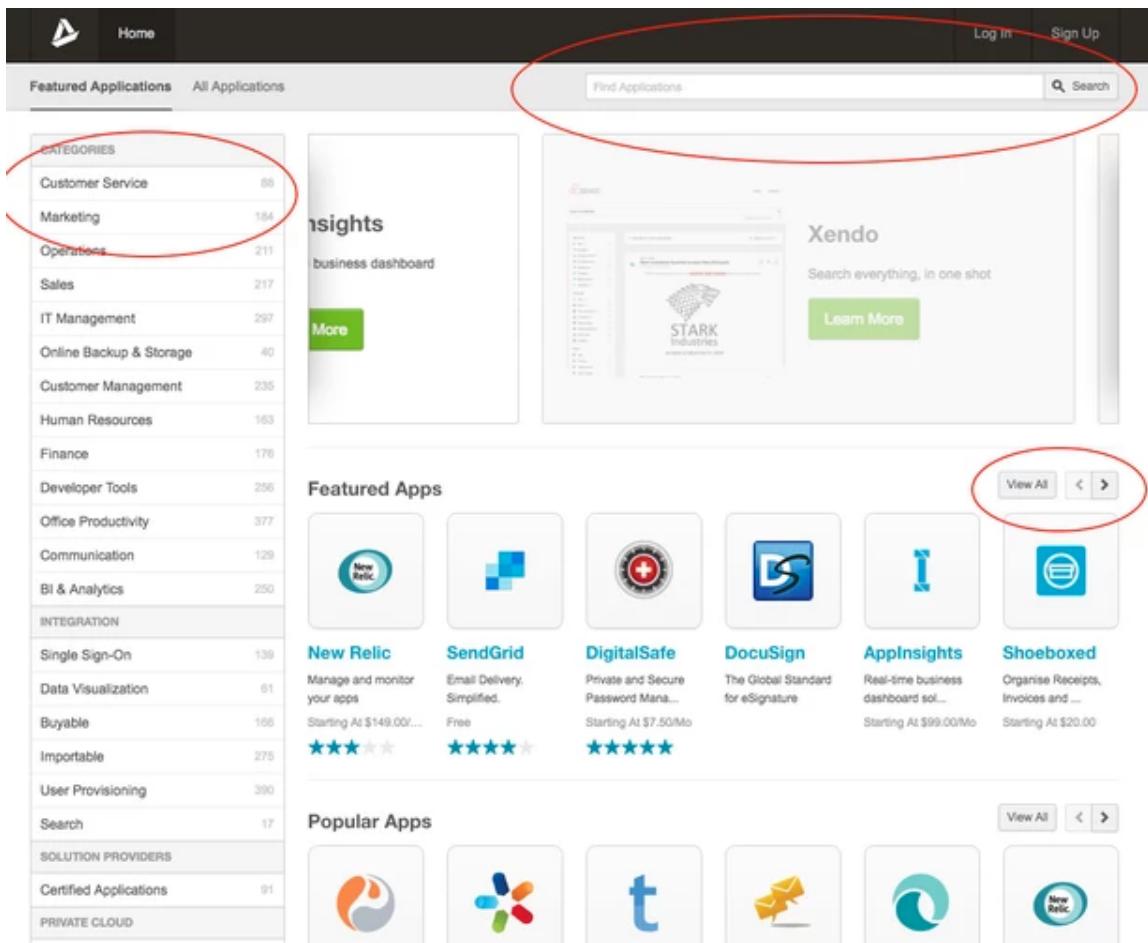


Figure 5.7 Changes in V2 with respect to V1

CHAPTER 6

PROJECT SNAPSHOT

6.1 METRO PCS MARKETPLACE

Mid-Tier Theme.

Business URL : <https://www.metropcs.com/>

AppDirect URL : <https://metropcsmarketplace.byappdirect.com/>

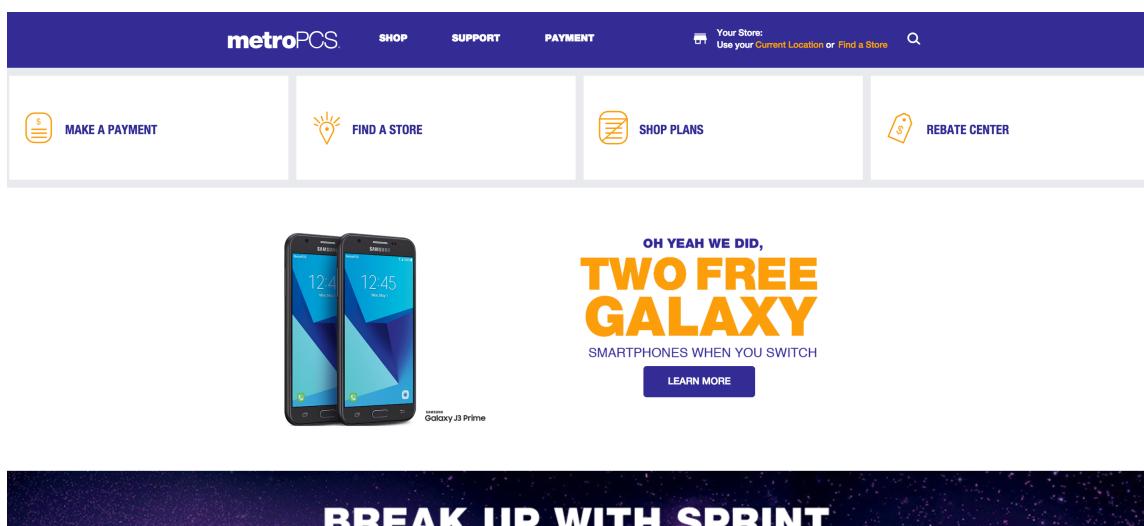


Figure 6.1 Original website of MetroPCS

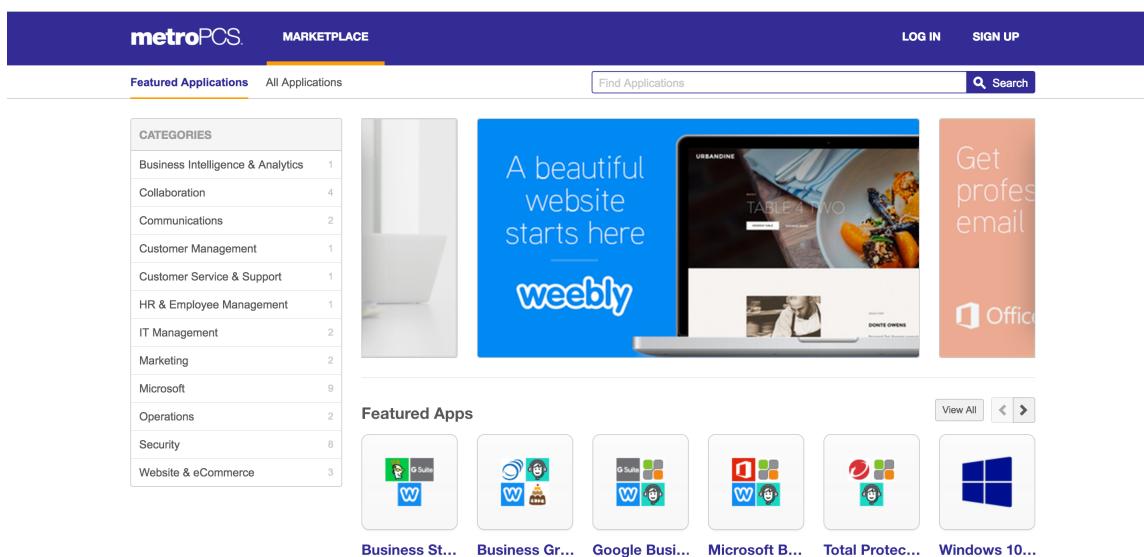


Figure 6.2 Marketplace created of MetroPCS

The screenshot shows the sign-in page for the MetroPCS Marketplace. At the top, there is a dark blue header bar with the "metroPCS" logo on the left and "MARKETPLACE" text next to it. On the right side of the header, there are "LOG IN" and "SIGN UP" buttons. Below the header is a light gray form titled "LOG IN TO YOUR ACCOUNT". The form contains fields for "Email:" (with the placeholder "email@address.com") and "Password:". There are also links for "Forgot password?" and "Keep me logged in." (with a checked checkbox). At the bottom of the form are two buttons: a dark blue "Log In" button and a white "Sign Up" button. Below the form, a small note says "No account? [Sign up for an account](#)". At the very bottom of the page, there is a footer bar with the text "Powered by AppDirect ©2017" on the left and "Call us toll free: (877) 404-APPS" on the right.

Figure 6.3 Marketplace Sign In page of MetroPCS

The screenshot shows the logged-in version of the MetroPCS website. At the top, there is a dark blue header bar with the "metroPCS" logo on the left and "MARKETPLACE" text next to it. On the right side of the header, there are "MY APPS", "MANAGE", and "TEST USER" buttons. Below the header is a section titled "Applications" with a small "0" icon. Under this section, there are two icons: one for "Setup Guide" (two interlocking gears) and one for "Documentation Center" (a computer monitor). At the bottom of the page, there is a footer bar with the text "Powered by AppDirect ©2017" on the left and "Call us toll free: (877) 404-APPS" on the right.

Figure 6.4 Logged In version of MetroPCS

6.2 UBER MARKETPLACE

Powerplay Theme.

Business URL : <https://www.uber.com/drive/vehicle-solutions/>

AppDirect URL : <https://uber.byappdirect.com/>

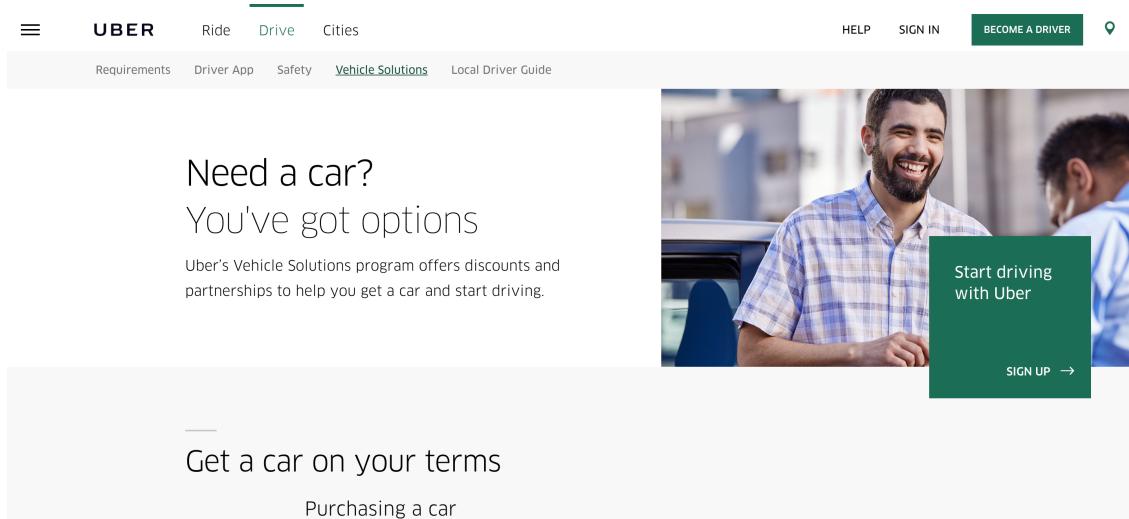


Figure 6.5 Original website of Uber

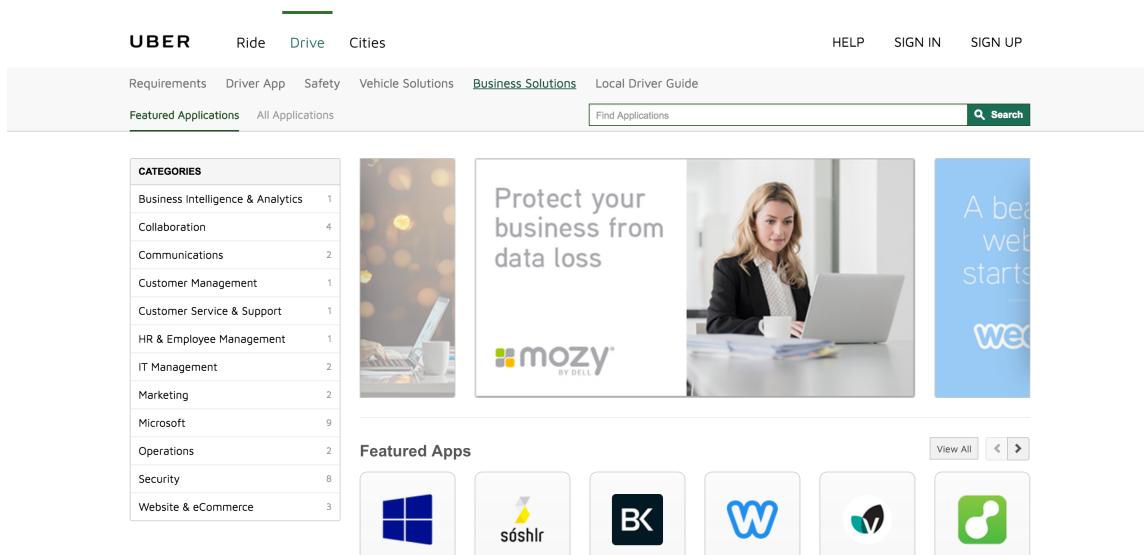


Figure 6.6 Marketplace created of Uber

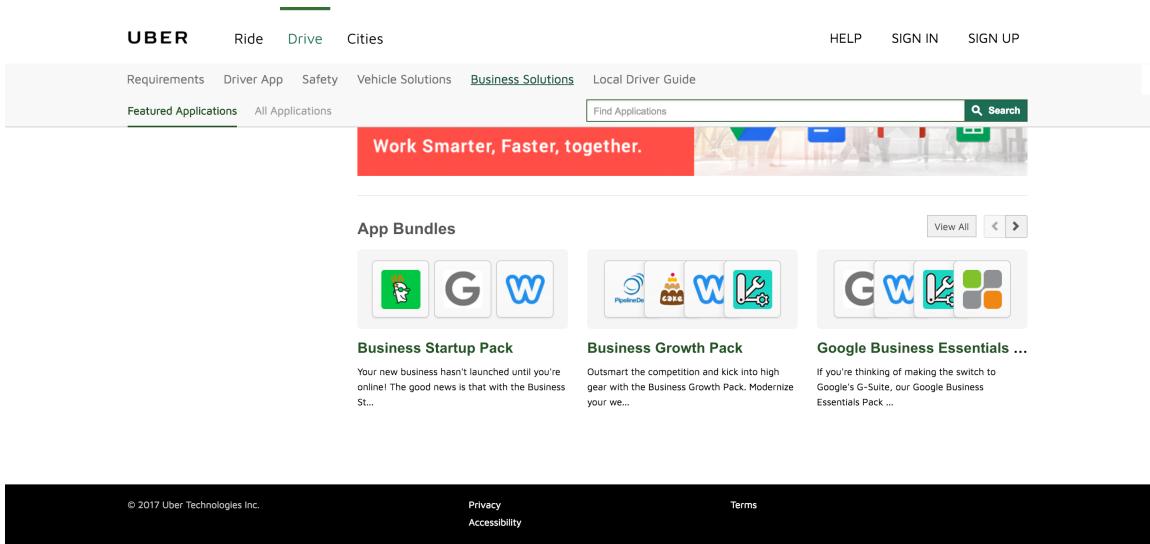


Figure 6.7 Footer created of Uber

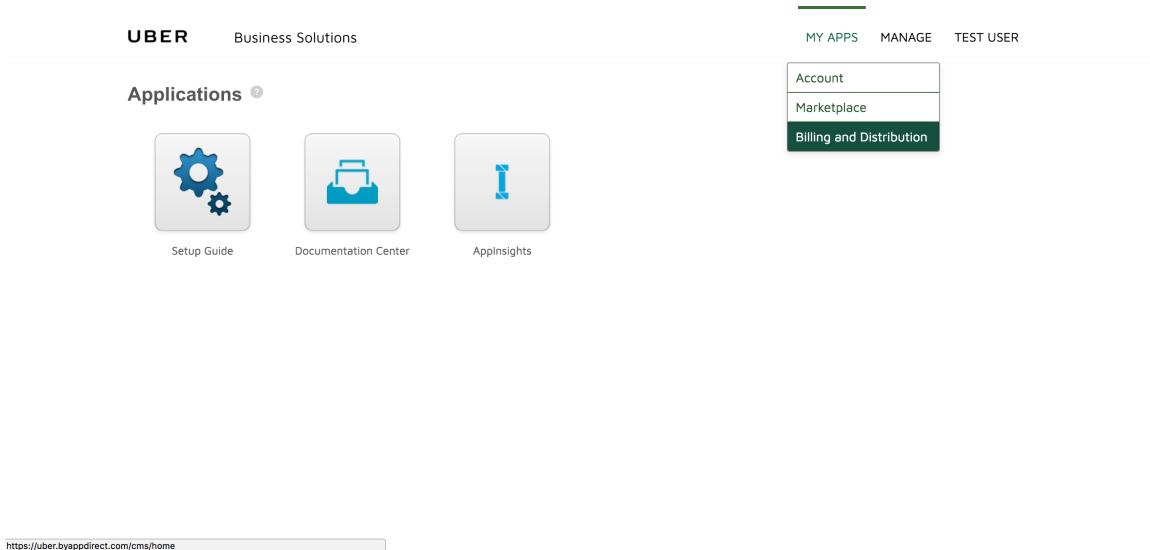


Figure 6.8 Logged In page of Uber

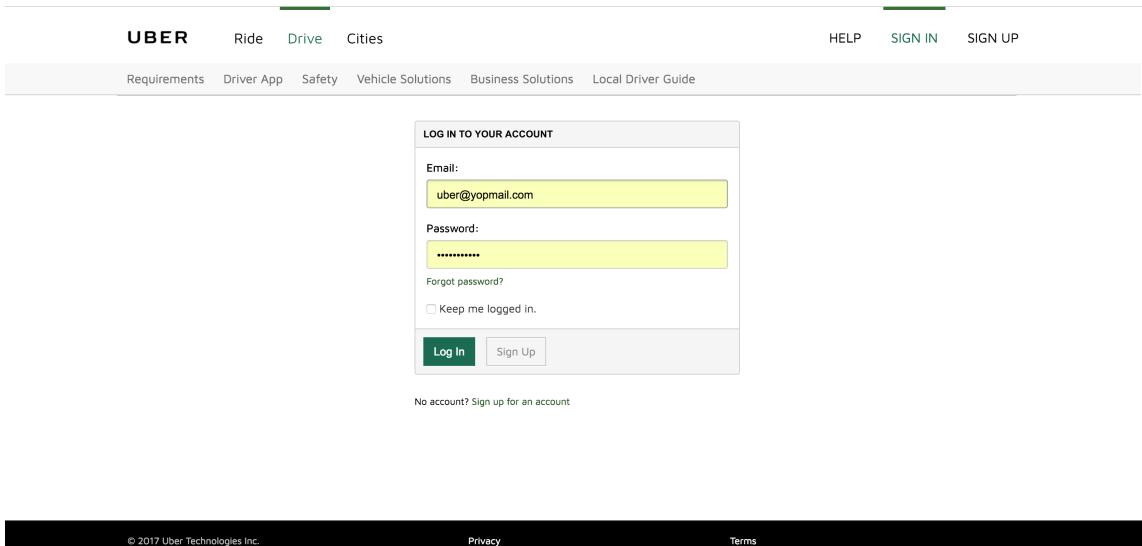


Figure 6.9 Marketplace Sign In page of Uber

6.3 SOCIUS MARKETPLACE

Semi-Powerplay Theme.

Business URL : <http://www.socius1.com/>

AppDirect URL :<https://socius.byappdirect.com/>



Figure 6.10 Original website header of Socius

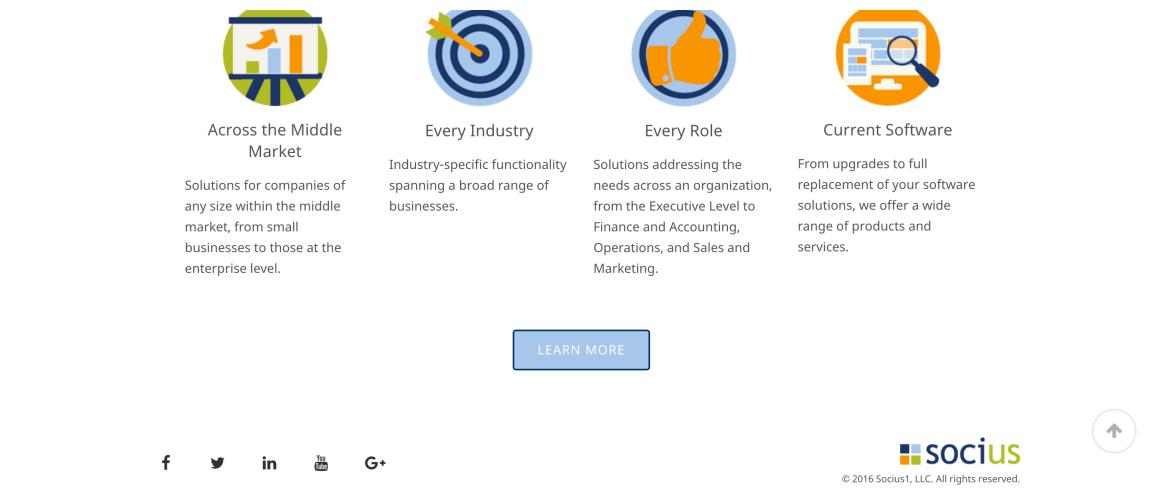


Figure 6.11 Original website footer of Socius

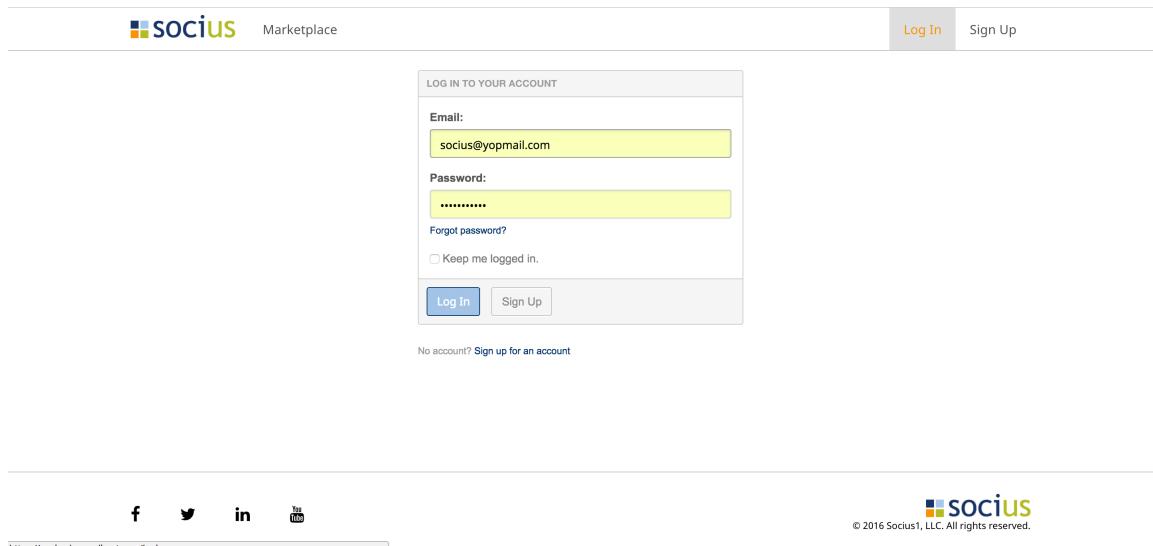


Figure 6.12 Marketplace Sign In page of Socius

SOCIUS Marketplace

Log In Sign Up

Featured Applications All Applications Find Applications Search

CATEGORIES	Count
Business Intelligence & Analytics	1
Collaboration	4
Communications	2
Customer Management	1
Customer Service & Support	1
HR & Employee Management	1
IT Management	2
Marketing	2
Microsoft	9
Operations	2
Security	8
Website & eCommerce	3



Azure Active Directory

Identity and Access - IDaaS

Learn More



Get professional email

Office

Featured Apps

View All < >



Windows 10...

Windows 10 and more



OneDrive for...

Efficient and reliable



Office 365 E...

Office when and where you need it



Dynamics 3...

Run your business



Dynamics C...

Customer



Azure Activ...

Identity and Access

Figure 6.13 Marketplace header created of Socius



GoDaddy D...

GoDaddy Domain Registration

Starting At \$16.58/yr



Visio Pro for...

Create Versatile Diagrams

Starting At \$13.00/M.



AppHelp Bu...

Technical Support & Cloud Adviso...

Starting At \$9.75/Mo



Trend Micro...

Hosted Endpoint Security for Win...

Starting At Free



Trend Micro...

Hosted Email Security and Antisp...

Starting At Free



Trend Micro...

Advanced Protection for Office 365

Starting At Free

App Bundles

View All < >



Total Protect Pack

Keep your transition to the cloud safe and secure with the Total Protect Pack. Mozy will help ke...



Microsoft Business Essential...

If you're thinking of making the switch to Microsoft Office365, our Microsoft Business Essentials...



Google Business Essentials ...

If you're thinking of making the switch to Google's G-Suite, our Google Business Essentials Pack ...



Figure 6.14 Marketplace footer created of Socius

6.4 BANK OF AMERICA MARKETPLACE

Powerplay Theme.

Business URL : <https://www.bankofamerica.com/smallbusiness/>

AppDirect URL : <https://bankofamerica.byappdirect.com/>

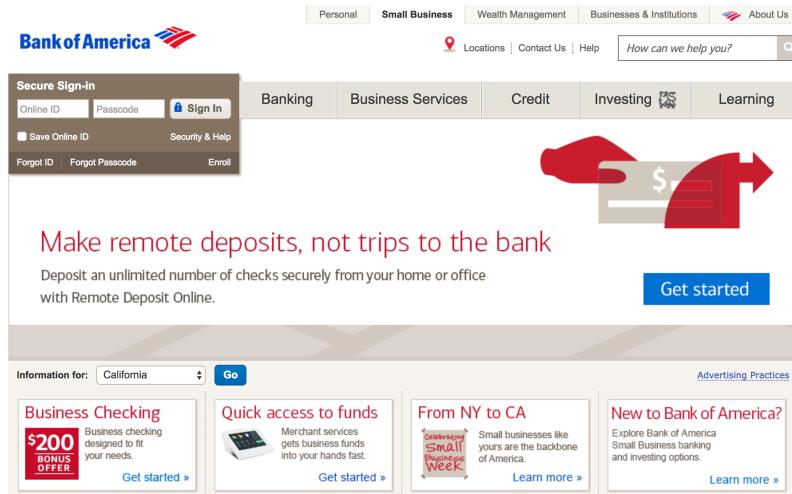


Figure 6.15 Original website header of BankOfAmerica

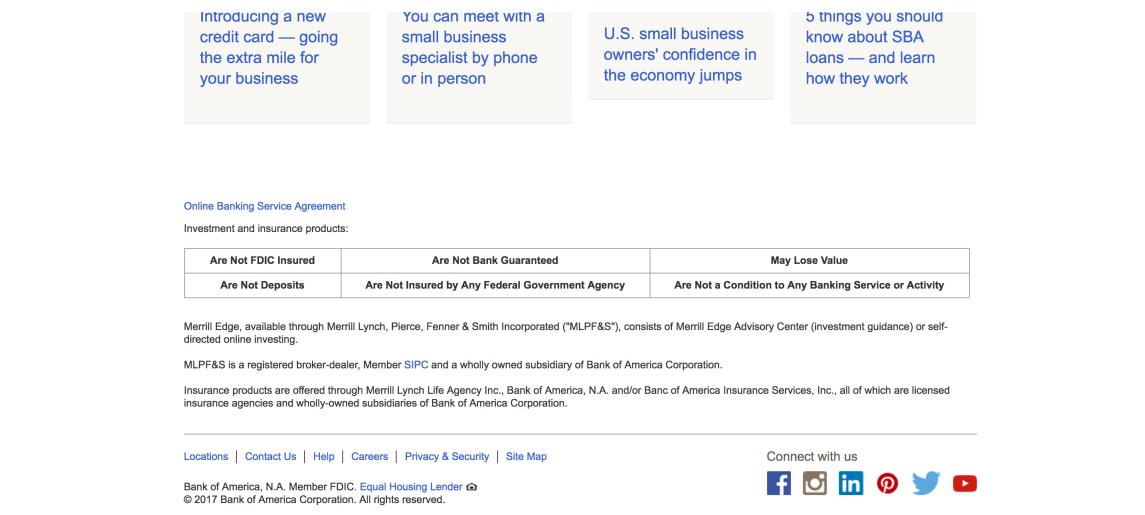


Figure 6.16 Original website footer of BankOfAmerica

Figure 6.17 Marketplace header created of BankOfAmerica

Figure 6.18 Marketplace footer created of BankOfAmerica

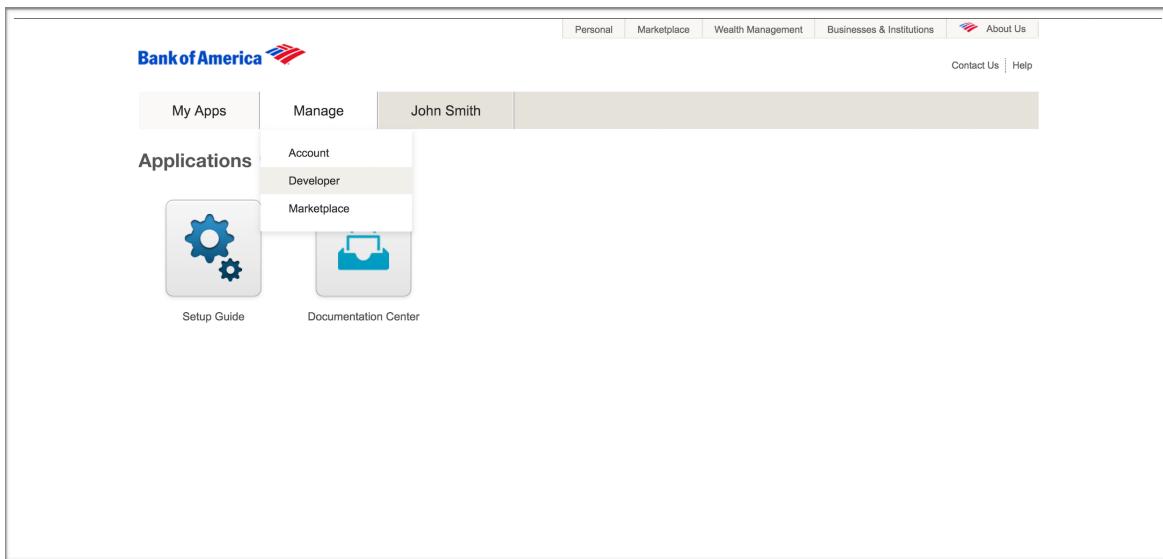


Figure 6.19 Logged In page of BankOfAmerica

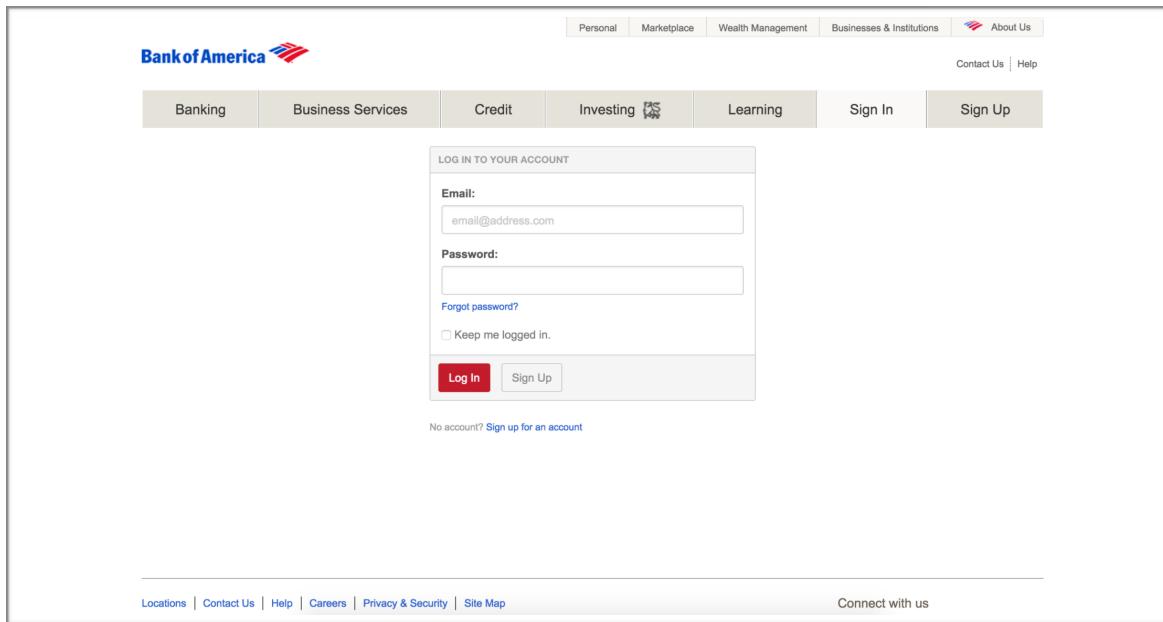


Figure 6.20 Marketplace Sign In page of BankOfAmerica

6.5 AGIO MARKETPLACE

Mid-Tier Theme.

Business URL : www.agio.com

AppDirect URL : <http://agio.byappdirect.com/>



Figure 6.21 Original website of Agio

A screenshot of the Agio Marketplace application. The top navigation bar includes the Agio logo, a "Marketplace" tab, and links for Log In and Sign Up. Below the navigation is a search bar with a "Find Applications" button and a "Search" icon. On the left, there's a sidebar titled "FEATURED APPLICATIONS" with a "CATEGORIES" section showing counts for various software types. To the right, there are three main application cards: one for Mozy (protecting business from data loss), one for a web-based service (with a blue background and partial text "A be... web starts... wed..."), and another partially visible. At the bottom, there's a "Featured Apps" section displaying icons for various Microsoft and Google productivity tools like G Suite, Microsoft Office, and OneDrive.

Figure 6.22 Marketplace created of Agio

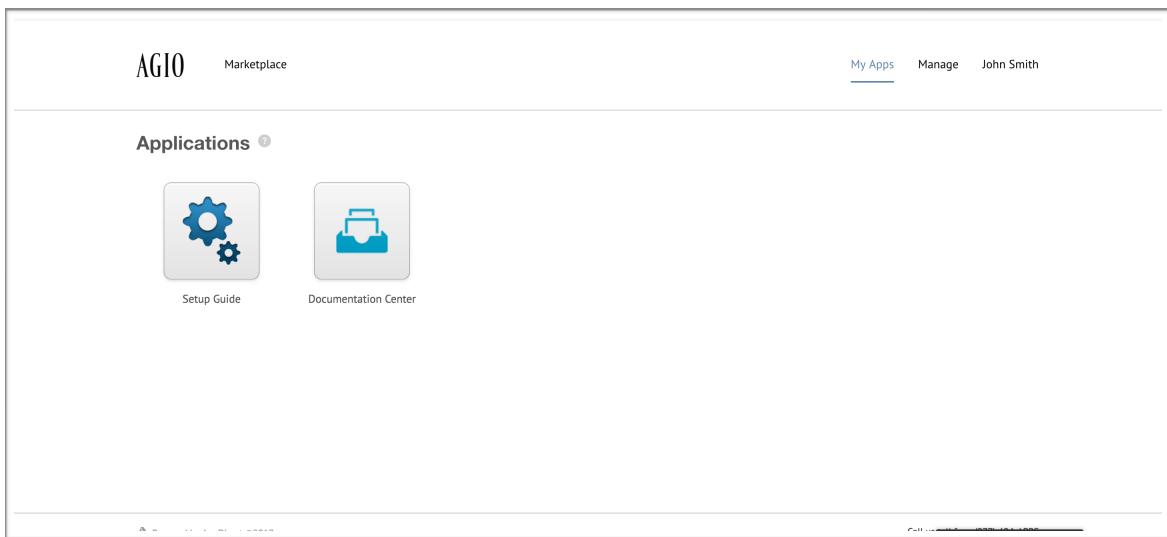


Figure 6.23 Logged In page of Agio

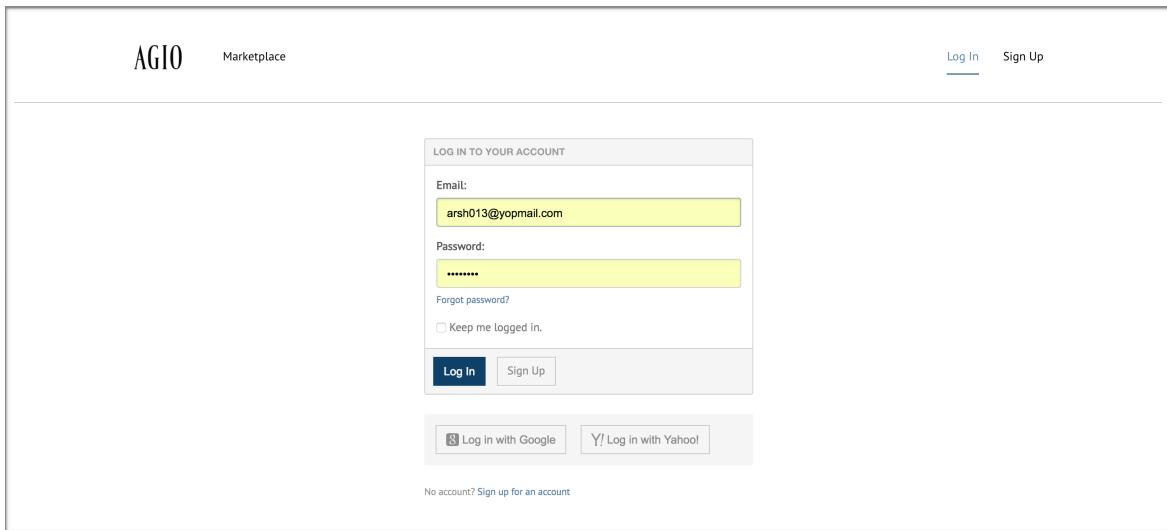


Figure 6.24 Marketplace Sign In page of Agio

6.6 CALTECH MARKETPLACE

Mid-Tier Theme.

Business URL : <http://www.caltech.com>

AppDirect URL : <https://caltech.appdirect.com/home>

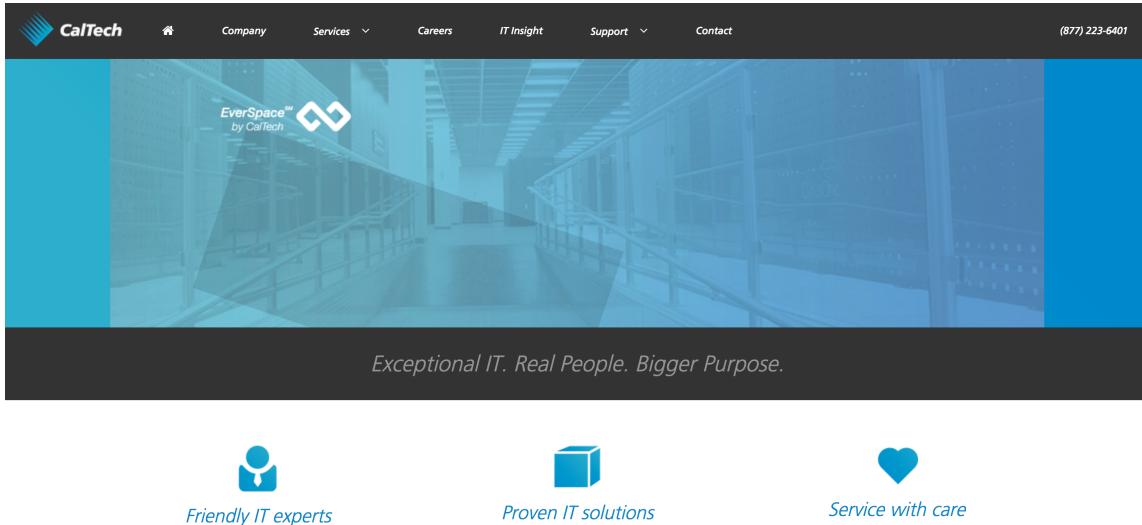


Figure 6.25 Original website of Caltech

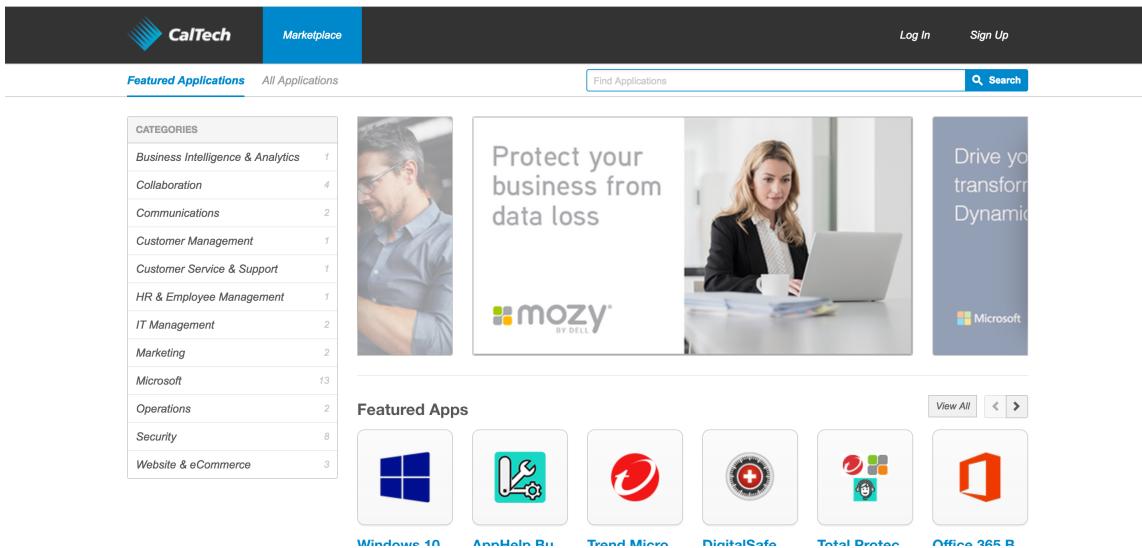


Figure 6.26 Marketplace created of Caltech

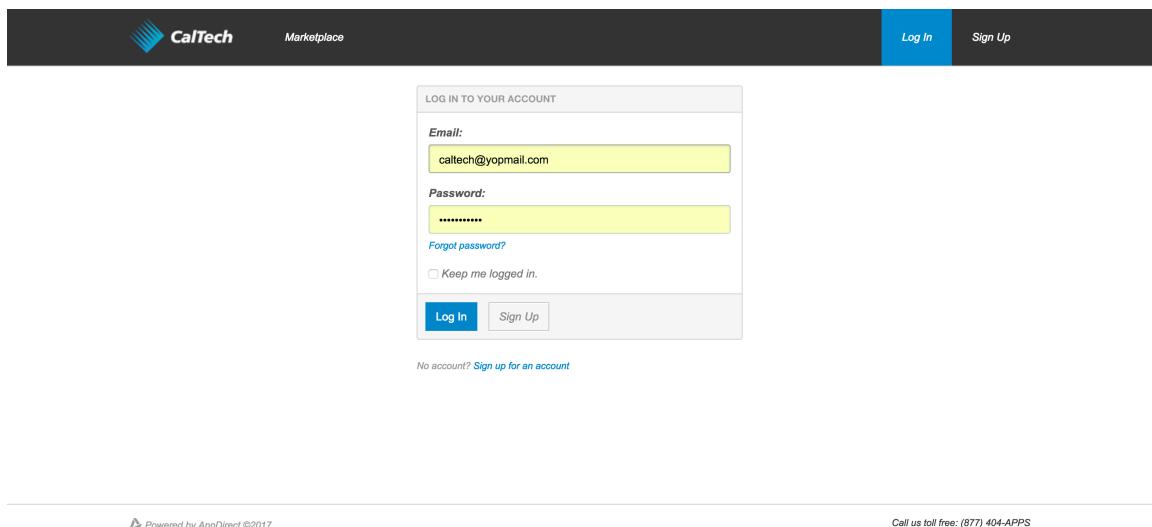


Figure 6.27 Marketplace Sign In page of Caltech

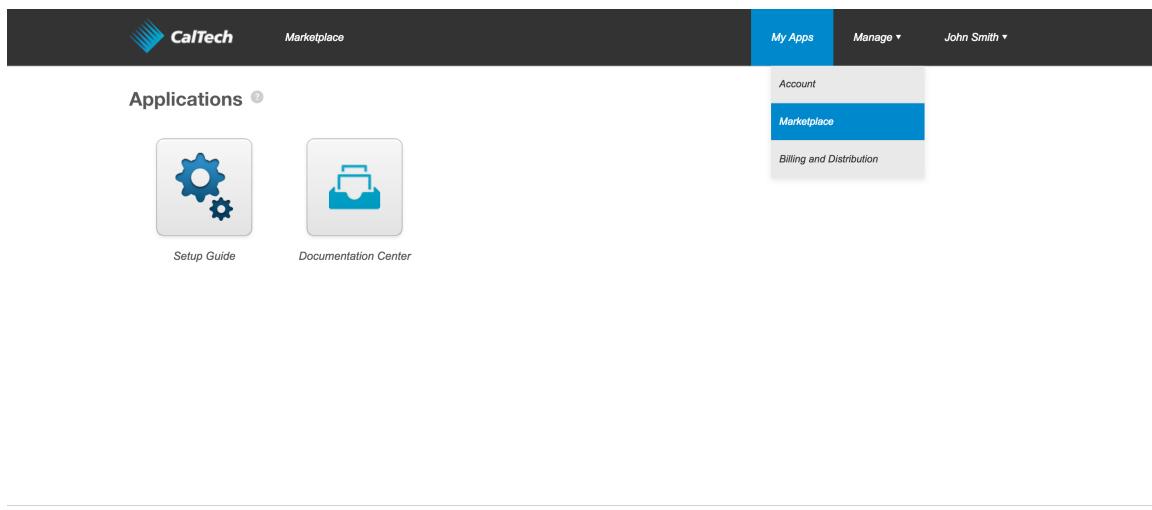


Figure 6.28 Logged In page of Caltech

6.7 TD AMERITRADE MARKETPLACE

Powerplay Theme.

Business URL : <https://www.tdameritrade.com/home.page/>

AppDirect URL : <https://tdameritrade.byappdirect.com/>

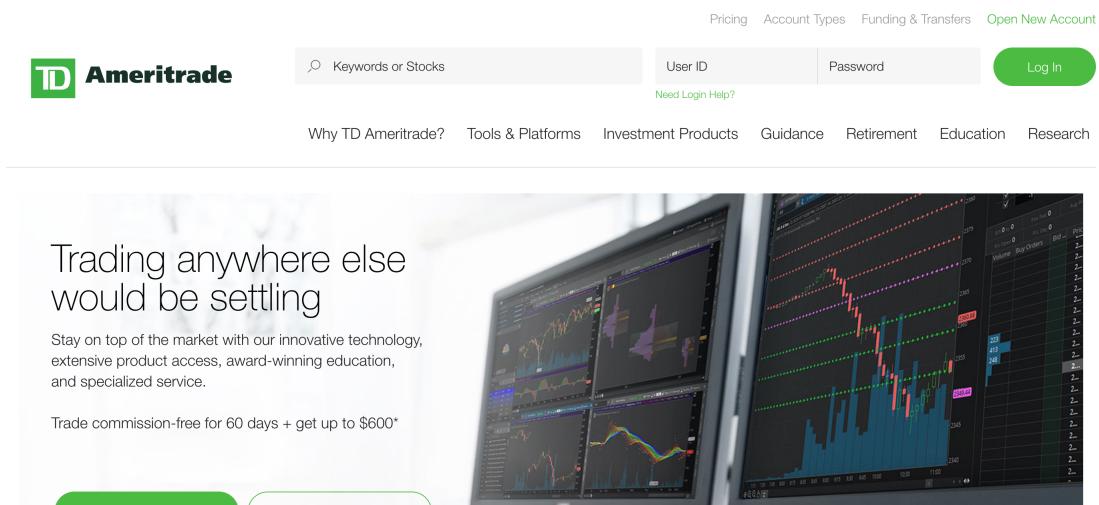


Figure 6.29 Original website of TD Ameritrade

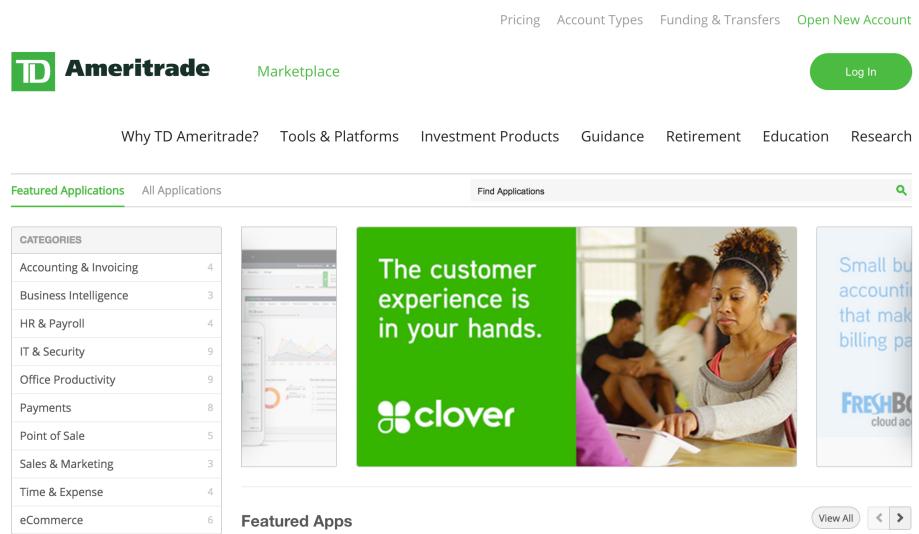


Figure 6.30 Marketplace header created of TD Ameritrade

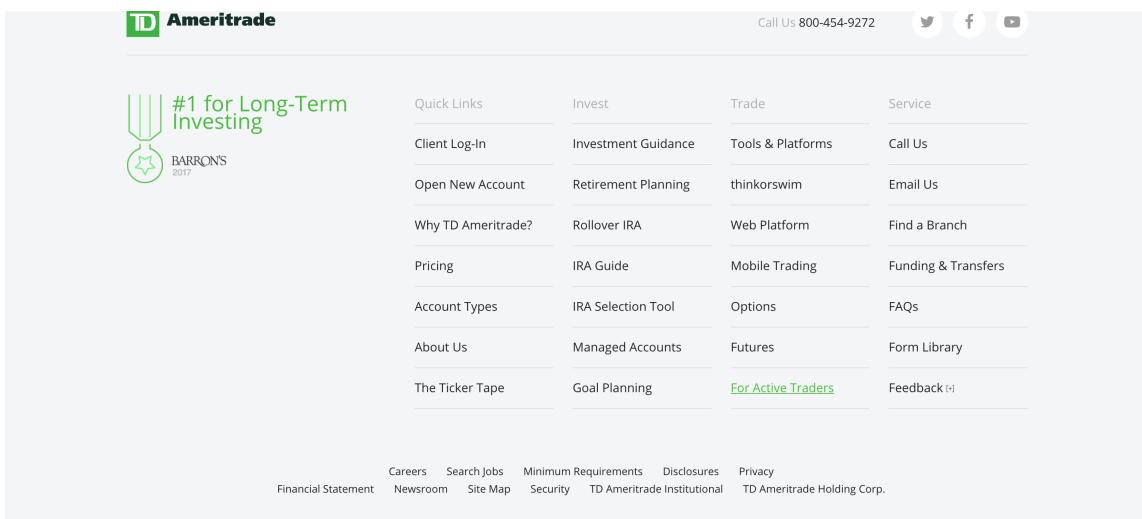


Figure 6.31 Marketplace footer created of TD Ameritrade

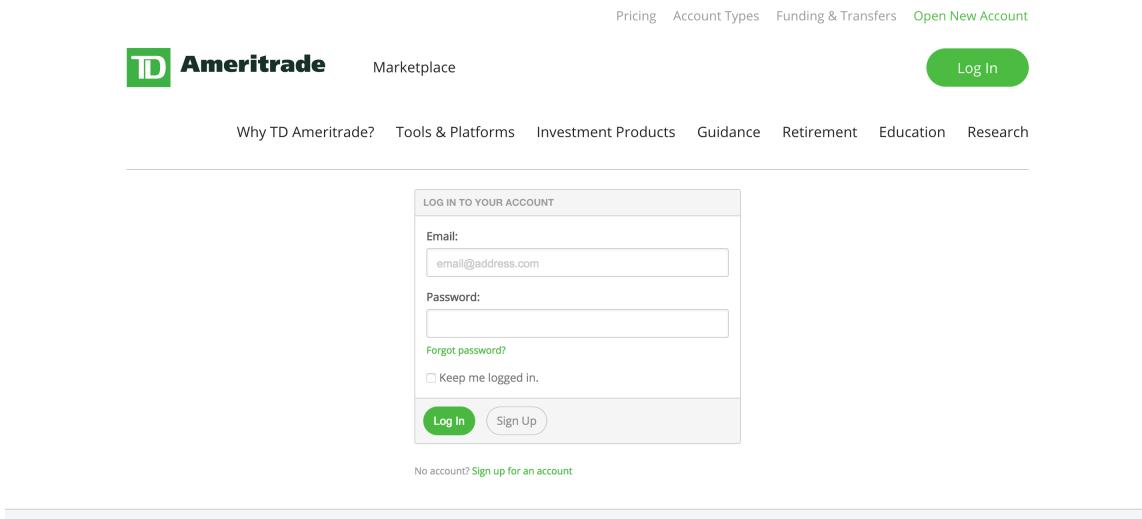


Figure 6.32 Marketplace Sign In page of TD Ameritrade

6.8 ZAYO MARKETPLACE

Mid-Tier Theme.

Business URL : <http://www.zayo.com/ca/>

AppDirect URL : <https://zayo.byappdirect.com/>

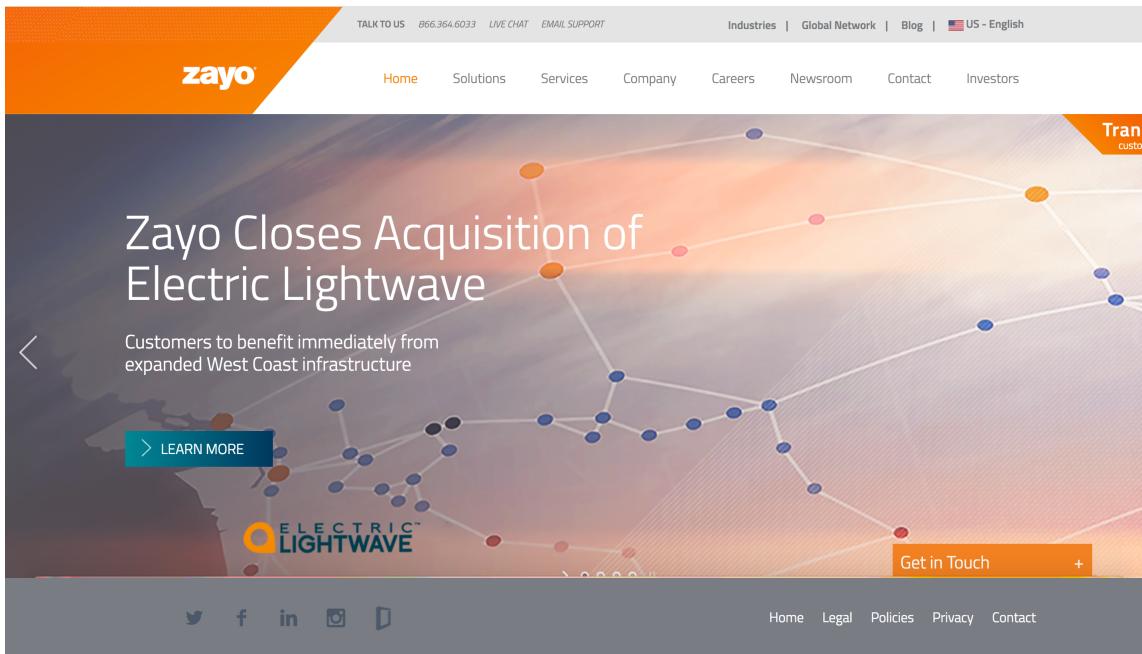


Figure 6.33 Original website of Zayo

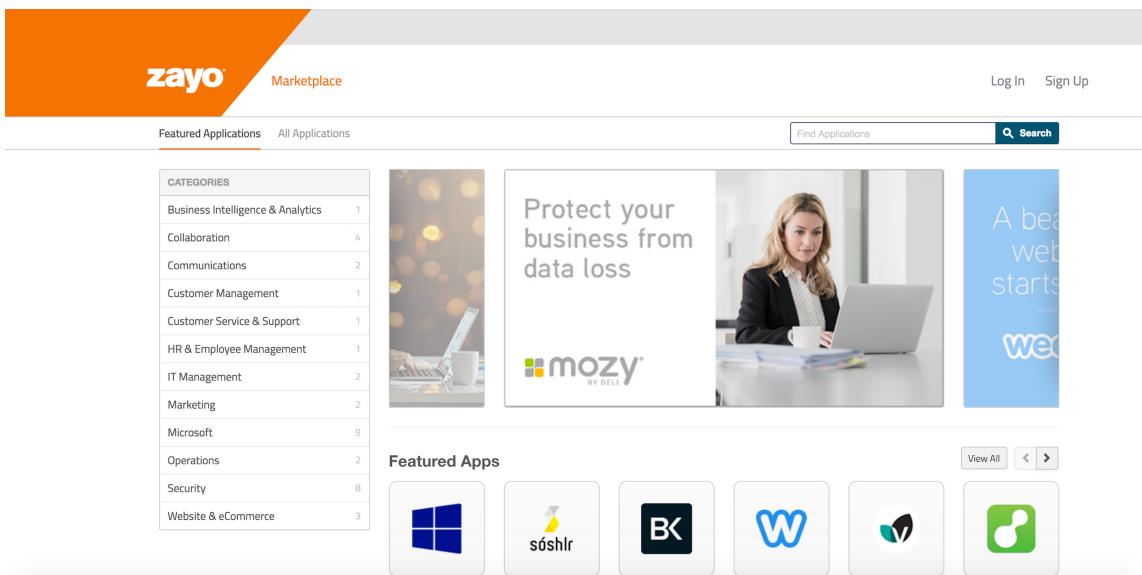


Figure 6.34 Marketplace header created of Zayo

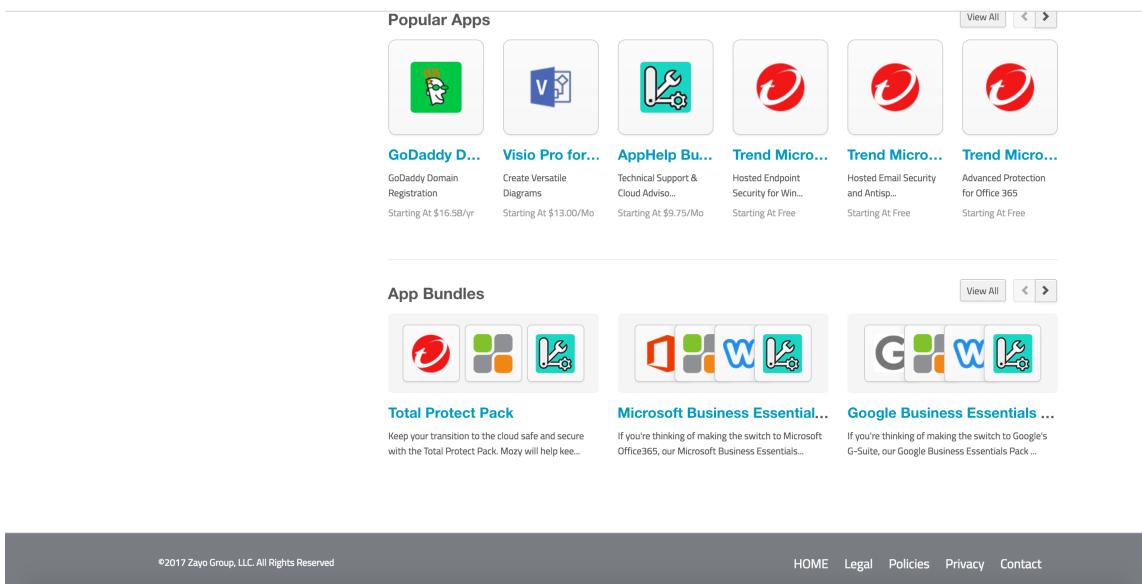


Figure 6.35 Marketplace footer created of Zayo

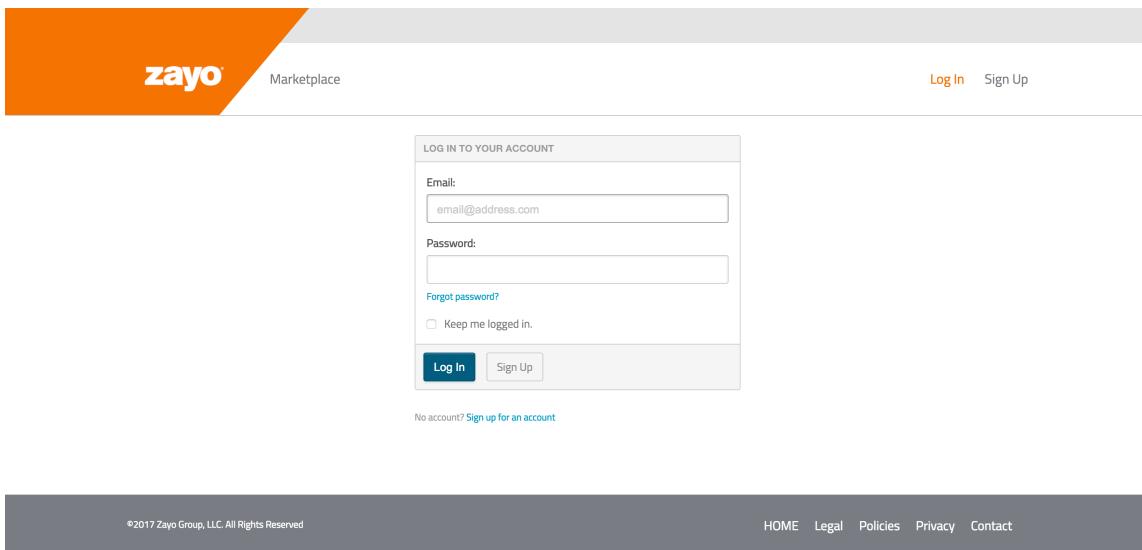


Figure 6.36 Marketplace Sign In page of Zayo

Other market places created :

6.9 REPUBLIC WIRELESS MARKETPLACE

Mid-Tier Theme.

Business URL : <https://republicwireless.com/>

AppDirect URL : <https://republicwireless.byappdirect.com/>

6.10 CAROLINA WEST WIRELESS MARKETPLACE

Mid-Tier Theme.

Business URL : <https://www.carolinawest.com/>

AppDirect URL : <https://carolinawestwireless.byappdirect.com/>

6.11 V1-V2 OF ELISA

Reference url: <https://sovellukset.elisa.fi>

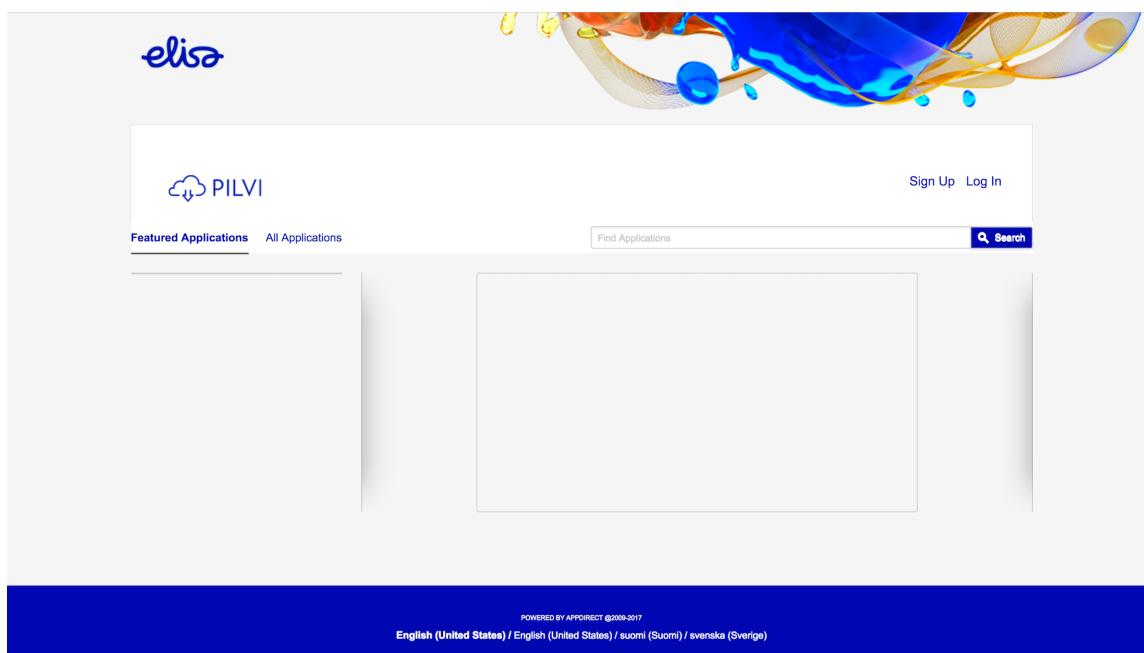


Figure 6.37 V1-V2 Elisa

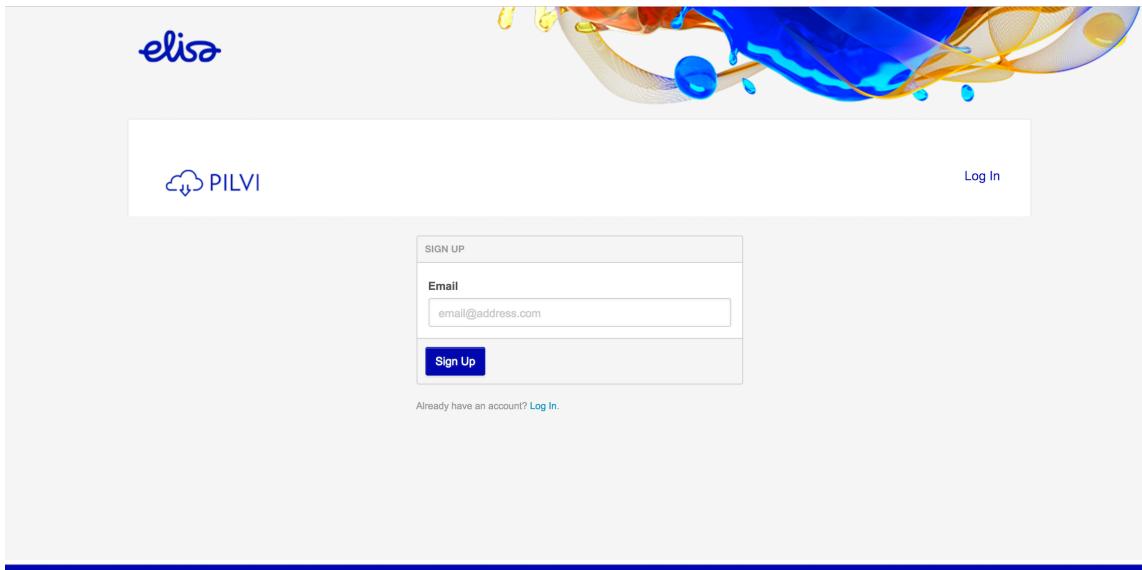


Figure 6.38 Sign Up page of Elisa V1-V2

6.12 MICROSERVICES

Translation of keys based on partner, variation, locale. If file is present then the respective translations will be provided. If key is not there in the file, then the same key will be returned. If the file is not there then the decision will be taken on the basis of locale. If locale is typed wrong by mistake, the default locale will be chosen which is EN. This is made using the express server. If the partner is wrong then again the decision will be taken on the basis of locale. If keys are not there then response will directly “Please provide any key” else the response JSON object will be the translations of keys as given by the user.

IDE used : Sublime 3.0 Text Editor

Platform used : Node.JS

Module used for server : Express

Request used : Post

Route : /translate

The screenshot shows a POST request to `http://localhost:5401/translate`. The request body is a JSON object with the following content:

```

1 {  
2   "keys": ["sign.in", "sign.up", "my.apps", "question.asked.event", "arshdeep.singh"],  
3   "partner": "tsystems",  
4   "locale": "de",  
5   "variation": "deutsche"  
6 }

```

The response status is 200 OK with a time of 13 ms. The response body is also a JSON object:

```

1 {  
2   "sign.in": "Anmelden",  
3   "sign.up": "Registrieren",  
4   "my.apps": "Meine Software",  
5   "question.asked.event": "Ein Benutzer im Business Marketplace for Enterprise hat eine Frage zum Produkt des Partners gestellt.",  
6   "arshdeep.singh": "arshdeep.singh"  
7 }

```

Figure 6.39 Key translations based on partner, locale, variation

The screenshot shows a POST request to `http://localhost:5401/translate`. The request body is a JSON object with the following content:

```

1 {  
2   "keys": ["sign.in", "sign.up", "my.apps", "question.asked.event", "arshdeep.singh"],  
3   "partner": "amex",  
4   "locale": "fr"  
5 }

```

The response status is 200 OK with a time of 13 ms. The response body is a JSON object:

```

1 {  
2   "sign.in": "Connexion",  
3   "sign.up": "Inscription",  
4   "my.apps": "Mes Applications",  
5   "question.asked.event": "Un utilisateur de la boutique a posé une question au sujet du produit du vendeur.",  
6   "arshdeep.singh": "arshdeep.singh"  
7 }

```

Figure 6.40 Key translations based on partner, locale

The screenshot shows a POST request to `http://localhost:5401/translate`. The request body is JSON with the following content:

```

1 {
2   "keys": ["sign.in", "sign.up", "my.apps", "question.asked.event", "arshdeep.singh"],
3   "locale": "en"
4 }

```

The response status is 200 OK with a time of 39 ms. The response body is:

```

1 {
2   "sign.in": "Log In",
3   "sign.up": "Sign Up",
4   "my.apps": "My Apps",
5   "question.asked.event": "A user in the marketplace has asked a question about the vendor's product.",
6   "arshdeep.singh": "arshdeep.singh"
7 }

```

Figure 6.41 Key translations based on locale

The screenshot shows a POST request to `http://localhost:5401/translate`. The request body is JSON with the following content:

```

1 {
2   "keys": ["sign.in", "sign.up", "my.apps", "question.asked.event", "arshdeep.singh"]
3 }

```

The response status is 200 OK with a time of 13 ms. The response body is:

```

1 {
2   "sign.in": "Log In",
3   "sign.up": "Sign Up",
4   "my.apps": "My Apps",
5   "question.asked.event": "A user in the marketplace has asked a question about the vendor's product.",
6   "arshdeep.singh": "arshdeep.singh"
7 }

```

Figure 6.42 Key translations based on only keys

The screenshot shows a POST request to `http://localhost:5401/translate`. The request body is a JSON object with fields `partner` and `locale`. The response status is 200 OK, and the body contains the message `Please provide any key`.

```
1+ [{  
2 "partner": "amex",  
3 "locale": "fr"  
4 }]
```

Body

```
i 1 Please provide any key
```

Figure 6.43 Response if no key is given

The screenshot shows a POST request to `http://localhost:5401/translate`. The request body is a JSON object with fields `keys` and `locale`. The response status is 200 OK, and the body contains a JSON object with translations for various keys.

```
1+ [{  
2 "keys": ["sign.in", "sign.up", "my.apps", "question.asked.event", "arshdeep.singh"],  
3 "locale": "punjabi"  
4 }]
```

Body

```
1+ {  
2 "sign.in": "Log In",  
3 "sign.up": "Sign Up",  
4 "my.apps": "My Apps",  
5 "question.asked.event": "A user in the marketplace has asked a question about the vendor's product.",  
6 "arshdeep.singh": "arshdeep.singh"  
7 }
```

Figure 6.44 Key translations if wrong locale is given

6.13 BUG FIX IN ROGERS

Manage link was not visible as the script of this was not properly available to the code so had to fix this bug and make it visible.

The screenshot shows the Rogers Business App Market dashboard. At the top, there's a navigation bar with the Rogers logo, 'Business App Market', 'Marketplace', 'MyApps', 'Manage' (with a dropdown arrow), and 'ROCH CHESN...'. A user email 'adtestone@mailinator.com' is also at the top right. Below the navigation is a sub-navigation bar with 'Dashboard', 'Applications', 'Users', 'Assign Apps', 'Billing', and 'Company Settings'. The main area is titled 'Dashboard' and contains several sections: 'INVITE USERS' (with 'Add Users from Current User List' and 'Add New Users' buttons), 'LINKS' (with 'Assign Apps', 'Manage Apps', 'Update Billing Information', and 'View Recent Invoices' buttons), and a 'ROCH CHESNAY' section showing '1 Members' and a list of users ('ab c def'). To the right is a 'Recent Events' section divided into 'THIS MONTH' and 'OVER A MONTH AGO' sections, listing events like 'ab c def created a quote for Test WH for ROCH CHESNAY' and 'ROCH CHESNAY has been activated by ab c def'. There are also 'Everyone' and 'Me' filters at the top of the event list.

Figure 6.45 Rogers before bug fix

The screenshot shows the 'Applications' page of the Rogers Business App Market. The top navigation bar is identical to Figure 6.45, with the addition of 'MyApps' and 'CANCEL CTN...'. The main content area displays a grid of application icons. The first row includes 'Setup Guide', 'RECURRING DUMMY APP COMBINATION Make Payment' (with a warning icon), 'Segment 20 App Make Payment' (with a warning icon), and 'Async Application'. The second row includes 'Test WH' (with a 'PENDING PURCHASE' button), 'Bundle 20' (with a 'NOT PURCHASABLE' button), 'RECURRING DUMMY APP COMBINATION' (with a 'PENDING PURCHASE' button), and 'Test WH' (with a 'NOT PURCHASABLE' button). A cursor is visible over the 'PENDING PURCHASE' button for the 'Test WH' app in the second row.

Figure 6.46 Rogers after bug fix

6.14 BUG FIX IN TELIA

Spacing in "Your Free Account Includes" section on sign up page is inconsistent. So SCSS changes were to be made in body customization of Telia customizer theme. One more issue where script had to be written was Drop Downs do not collapse and CTA buttons are small. This was also fixed.

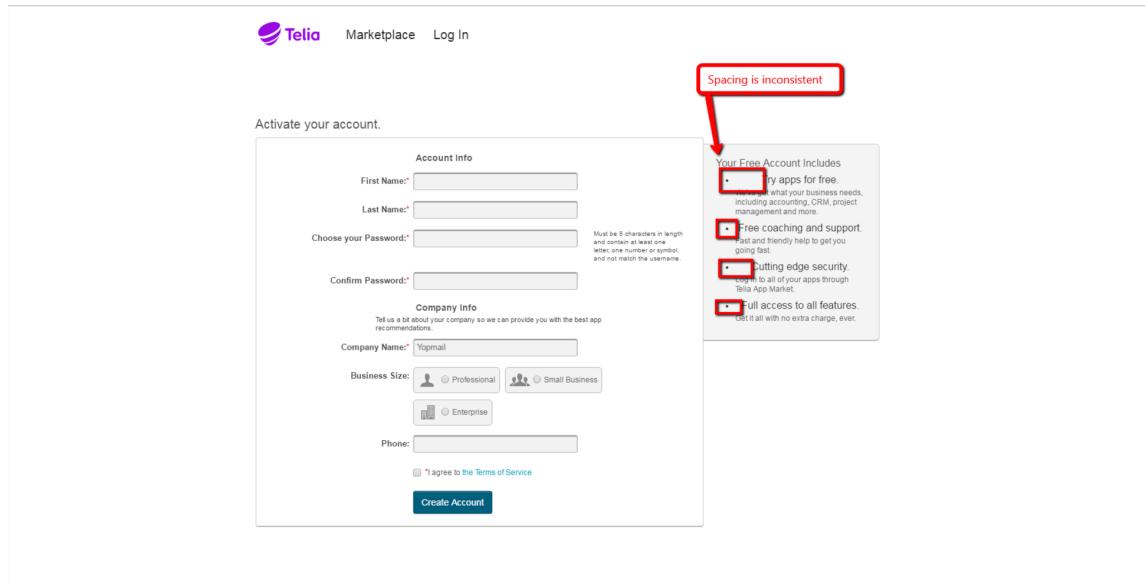


Figure 6.47 Telia before bug fix

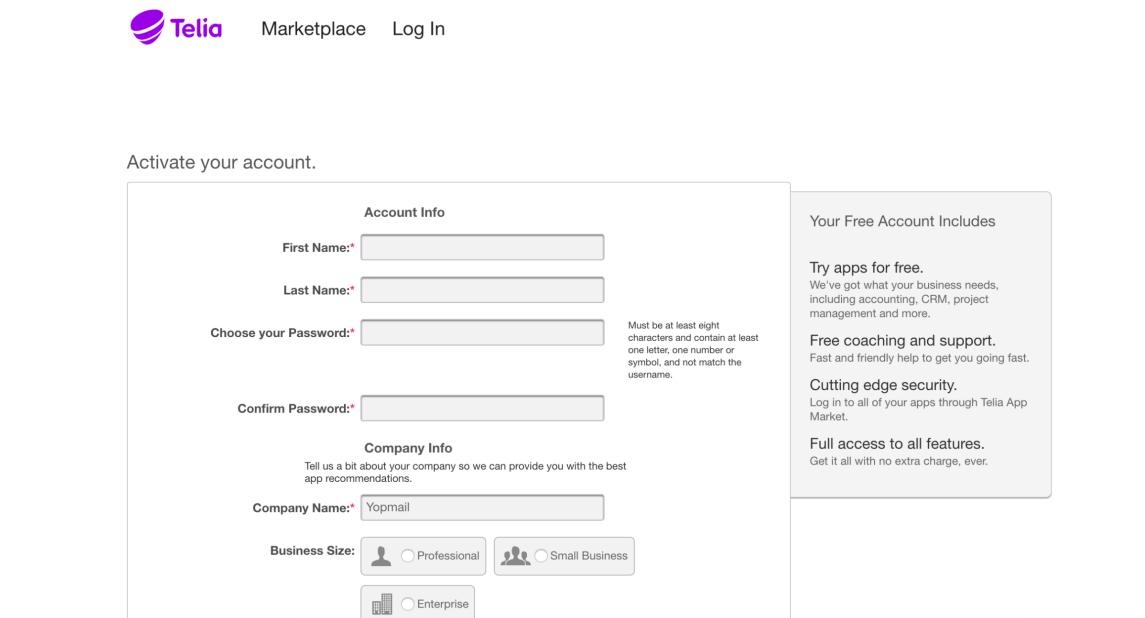


Figure 6.48 Telia after bug fix

6.15 BUG FIX IN APPCELERATOR

Recurring monthly fee price line misaligned for expiring discounts. So SCSS changes were to be made in body customization of Telia customizer theme as well as the monolith AppDirect repository.

The screenshot shows the 'Confirm Order' step of a four-step process. The top navigation bar includes links for Product, Community, Resources, Partners, Customers, Company, and Contact, along with a user icon. Below the navigation is a progress bar with four steps: 1. Create Order, 2. Billing Details, 3. Confirm Order (the current step), and 4. Order Receipt. The main content area is titled 'Confirm Order'. It displays a 'Bill To:' section with the address: Super Appceleratortest, 3282 Scott Road, Somerville MA 01970, United States. A table summarizes the order details:

Name	Edition	Price	Quantity	Total
178-CARL-Templaterecuring	Recurring Edition	\$10.00 / Month	1	\$10.00
this is a discount for Modules		-10%		-\$1.00
		Subtotal:		\$9.00
		Fee due at checkout:		\$9.00
		Total recurring monthly fee as of 03/17/17:		\$9.00
		Total recurring monthly fee as of 05/17/17, after 2 months discount:		\$10.00

Below the table, payment method information is shown: Card ending in 1111. At the bottom right is a red 'Place Order' button.

Figure 6.49 Appcelerator before bug fix

The screenshot shows the 'Confirm Order' step of a four-step process, identical to Figure 6.49 but after the bug fix. The top navigation bar and progress bar are the same. The main content area is titled 'Confirm Order'. It displays a 'Bill To:' section with the address: Abc, Winnipeg MB R2C 0A1, Canada. A table summarizes the order details:

Name	Edition	Price	Quantity	Total
arshtest	Recurring Edition	\$10.00 / Month	1	\$10.00
arshtest	Recurring Edition	\$10.00 / Seat / Month	1	\$10.00
wwwregerhrt		-15%		-\$3.00
		Subtotal:		\$17.00
		Fee due at checkout:		\$17.00
		Total recurring monthly fee as of 03/22/17:		\$17.00
		Total recurring monthly fee as of 05/22/17, after 2 months discount:		\$20.00

At the bottom right is a red 'Place Order' button.

Figure 6.50 Appcelerator after bug fix

Other PSFE bugs fixed :

6.16 BUG FIX IN TELEKOM CLOUD

There were two issues in this marketplace. No message displayed after first clicking sign-up, for trying to register with already registered email address and /signup page takes two clicks of ‘Sign Up’ button before tooltip for already registered account appears. Recurring monthly fee price line misaligned for expiring discounts. So SCSS and in HTML the ref changes were to be made in Telekom Cloud customizer theme as well as the monolith AppDirect repository. A new script had to be written of the 2 click sign up to correct that issue.

6.17 BUG FIX IN BRITISH TELECOM

Clicking Marketplace Link From App Profile Fails to Load Home and Clicking Link Again Causes 404. So HTML changes were to be made in logged in file of British Telecom customizer theme in which the hrefs needed to be changed to the newer ones.

6.18 BUG FIX IN BUCKEYE BROADBAND

Logo changes And footer HTML and SCSS changes were to be made. So the bugs were fixed of Buckeye Broadband mid-tier theme.

6.19 BUG FIX IN SAP

Search button UX can be improved he drawer opens upon hovering (which is good), but if one clicks on the magnifying glass (inside the box) before entering anything, it triggers a refresh of the screen and closing of the drawer, the effect is a bit jarring. So a complete new script was written to improve this issue in the SAP customizer theme.

CHAPTER 7

OBSERVATIONS AND FINDINGS

1. During this project, we have implemented and explored various technologies such as HTML5 CSS, SASS, Gulp, Node.Js, NPM (package manager), jQuery, JavaScript and Ajax.
2. Major findings were that in case we go on writing the CSS it is neither manageable nor maintainable. It has a very hectic Scalability, therefore use of SASS is better. One of the amazing things about SASS that we learned earlier is that with `@imports` you can break your style sheets into meaningful separate sheets. This allows us to individualize all our style sheets for each of the different devices we are targeting.
3. Another benefit of SASS is that it allows us to define variables that are reusable throughout our entire project. This provides a huge benefit when working with things like colors and fixed width sizes.
4. Another feature of SASS that begins to utilize some of its power is the `@extend` attribute. With `@extend` you can include styles into other classes that belong to a base class.
5. In the Microservices task, we got a fair understanding what microservices, web services and monolith architecture are so that it was easy to design. It was learnt that before doing any work it was important that to understand what the work is. We also learnt about localization and internationalization.
6. Writing scripts to make the page more responsive was written, how to find that if a script was getting loaded into a particular page from the sources, network in the inspect of page browser.

CHAPTER 8

LIMITATIONS

It has been tried that the project has minimum limitations but there still exist a few:-

1. Each of the customization task requires prior knowledge of the product and marketplace, this is a drawback as anyone without the prior knowledge of the marketplace can't do the customization.
2. Themes are not auto-deployable which means that once the theme is customized and is ready to be deployed on either the localhost or on the production url, we need to manually do the "Publish" task, which includes transferring .html and .css files of the theme to the internal theme customizer tool.
3. Similarly, whenever we need to have the properties of different elements used on the marketplace then we need to export those from the website itself; every time the changes are made, those many number of times one needs to export the files and a new zip is created after that. Therefore as many times a theme is exported a new zip is created which also utilizes a lot of time.
4. Microservices task to be still to be deployed by the architects of the company and hence a solution has to be found how to change the i18n object and deploy the microservices.
5. Another limitation is during bug fixes if a client's customizer theme is not available in the database of AppDirect, it has to be imported from the test database of AppDirect and a proper procedure has to be followed for that.

CHAPTER 9

CONCLUSION AND FUTURE WORK

Even with a few limitations, the project is able to achieve the main goal of customizing the themes for the marketplaces in an optimized and efficient way. Along with this project, we got to learn and implement many different technologies and how to proceed on a project with a strategic approach.

It is always good to expand the project as it makes its base stronger. In future, we would like to overcome the limitations: the core tool is build using the older version of Backbone and Wicket; which will be built using the React Js.

Backbone Js was having a few cons :

1. For pages with many interactive elements, Backbone becomes slow.
2. Concerns with performance because of many DOM elements.
3. Scopes are easy to use, but hard to debug.
4. Router is limited.

React Js is preferable as :

1. It has significant performance gains.
2. It is easy to import components although having very little dependencies.
3. Great for JavaScript debugging.
4. Fully component based architecture.

Therefore, we want to use React Js and improve the project further.

APPENDIX

REFLECTIVE DIARY

16/1/17

The first day of internship. The machines were provided and the documentation were collected. The documents included 10th, 12th, CG cards, PAN card, Aadhar card etc. A look up around the company and what the company does. There were several meetings with various engineers, leads and head of company today. This happened on day 1 of corporate world.

17/1/17

Today's day consisted of two sessions. First session was of Github. In this session, we created an account on Github and linked it with authy app on our android device. We used Github (git commands) on terminal on mac. We learnt commands of Git such as :

git --version, git config user.name, git config user.email, git init, git remote add origin url, git push origin master -u, creating and changing (modifying) a file in terminal & pushing it to repository on the Github account etc. Second session consisted of machine setup. Several softwares were installed such as Java JDK.

18/1/17

Today's day consisted of two training sessions. One was of Github tutorial. Several commands which were learnt today are - git push, git clone, git merge, git rebase, etc. Also learnt how to fork a repository from another Github account to our account. Second session consisted of machine setup. In this we installed several softwares like Cisco Anyconnect VPN and Maven and Node.js using the Dev Machine Setup documentation and mentors were also present during this.

19/1/17

Today only Dev machine was conducted and there was a training about security. What security measures have to be followed so as to protect the MAC OS X. In machine setup, we were asked to install softwares such as IntelliJ IDE for Java, Javascript, etc. and MySQL SequelPro & configurations were made of IntelliJ such as lombok support etc.

20/1/17

Today's day consisted of only Dev Machine Setup. Certain applications installed were docker and jBilling and the docker was made up by importing four images. The main repository of AppDirect was cloned into the local. jBilling local database was set up in the local machine.

23/1/17

Today there were two sessions. First session was about product overview. Workflow of company's product was taught, capabilities, benefits from the end user's side as well as independent seller's side. Sources of revenue about the company.

Second session: We were taught how to create orchard and about the development services on - testmarketplace.appdirect.com . Development, publication and buying of application were covered up. One assignment was given on the development, publication and purchasing of app.

24/1/17

There was a partner integration session taken. In this session, we were first taught how to sign up on test marketplace with a new company, give reseller rights, get channel admin, set up applications page for users. Purchase flow was completed on this day. An assignment was given on this partner integration training and we are supposed to complete it by tomorrow.

25/1/17

Today's day consisted of no sessions. We were told to complete the pending assignments and ask for any doubts if there in the whole training process till now. Also there were some practices in second half regarding the inauguration of new office of Pune on 30th of January.

27/1/17

There was only one session which consisted of Java Tutorial. First we were taught what were strings, arrays. How to decide name when two classes are there in one file. We learnt about what are interfaces and enums, loops (for, while, do while, for each) were covered up. After this stacks, queues and linked lists were made and studied. After this two assignments were given to complete.

30/1/17

As there was inauguration of new office in Pune. Therefore the session was only of around 45 mins in which we were given an assignment to complete till weekend (Shopping Cart).

31/1/17

So, the Java session continues and exception handling was taught. String buffers and string builders were also taught and after this, file handling operations session was taken. After this assignments were given on the topics taught today.

1/2/17

Another sessions on topics of Java were taken. The topics covered were generics in Java, threading where methods learnt are - wait(), notify(), join(), etc. After this session Java training was called off and daily assignments were given.

2/2/17

There was no session as of today, but tech challenge of Java was given. This tech challenge is to be completed by those who want to be permanent employees of this company. In around a weeks time was given to complete this assignment.

3/2/17

There was again no session today, but we were asked to complete any pending daily assignments of Java and if time permits, we were asked to get a hands on with the Tech Challenge Documentation. Also, there was a sync up with the UI training mentors.

6/2/17

There was a single session of HTML which was the first traing part of UI training. The topics taught were styles, attributes, comments in HTML, image float etc. Overflow : hidden, borders, etc. were also taught. After this time was given to us to complete the Java Tech Challenge.

8/2/17

Today's UI training consisted of CSS (cascading style sheets) which were to be sourced in HTML to give style to HTML. Syntax was learnt of CSS, ways to insert CSS - external style sheets, internal style sheet, incline style and orders (precedence) of this sheets. Background properties, border-styles, padding-styles, etc. were learnt. After this

session, daily assignment was given on CSS and we are supposed to complete it overnight.

9/2/17

Basic SCSS operations were taught in today's sessions. How to import SASS files in CSS files. The assignments we got two days back (HTML and CSS) we were asked to do these both with SCSS.

10/2/17

Today's session consisted of Javascript trainings. Variable date functions, search, replace, match, etc. other things of Javascript. An assignment of eight problems was given and we were asked to complete it today itself. An additional whole company session was also there regarding Tech-talk hackathon.

13/2/17

Today's UI session consisted of DOM session (Document Object Model). In this we learnt- getelementbyID, create element, set Attribute, append child, etc. Insert before, create test node, etc. were also taught. After this an assignment was given of DOM and were given two days to complete this assignment.

14/2/17

There was no session today and we were given time to complete any pending assignment and complete the DOM assignment.

15/2/17

UI training last part session was conducted today which consisted of jQuery training. Syntax was taught and basic functions adding attributes, removing attributes, etc. were taught. The DOM assignment was given again to complete with jQuery.

16/2/17

There was no session again today, and were asked to complete the pending assignments of UI and the jQuery assignment. Also, the Java tech challenge was extended the time to next Monday due to the daily UI assignments.

17/2/17

A certain time was devoted to Java Tech challenge and an introductory session was taken for Theme Customization project. Theme Customization is basically branding of marketplaces. This project, we will be working on for next 1.5-2 months. Teams were also allotted to the interns, I was allotted the UI theme customization project as of now.

20/2/17

Session was taken in which flow of Theme Customization, how theme customization works, how to create a theme, creating a branch and proceeding further and creating a whole new marketplace which will go directly in production.

21/2/17

First ticket was assigned to me in marketplace Production url - dsci.byappdirect.com . I started this ticket, and designed logged in header. Ticket no.- ISMM 70

22/2/17

Same Dsci production site was continued and completed this website today. Pull request was also created for this.

23/2/17

Some changes were told by Senior Front End engineers and were corrected the things to be changed. Finally the production site was completed by my end.

24/2/17

Green signal was given to Dsci by senior engineers. A new ticket was assigned to me. MetroPCS ISMM-110. So today's day went off creating this marketplace url metropcs.byappdirect.com .

27/2/17

MetroPCS was given final look. A pull request was created for this ticket. Also today tech challenge of Java was submitted. And Tech Challenge of Front end was assigned to us to complete and date to be submitted was 10/3/17 .

28/2/17

A few changes were told in metropcs about the logo and front problems as said by senior engineers. Finally at the end of day PR was created about the changes made today.

1/3/17

A new website was assigned to me- Uber Marketplace ISMM-121. Production was started on this ticket.

2/3/17

Uber production continued on local and production url- uber.byappdirect.com . At the end of the day PR was created for this ticket.

3/3/17

Some changes were asked to do on the Uber production site and were made and at the end of the day PR was updated.

6/3/17

Headers were made sticky of Uber and other changes were made and a final look was given to Uber by my side. One more ticket was assigned to me which was a buy in pcm website.

7/3/17

Bug ticket was de-assigned due to some AppDirectBasic layout problem and a new ticket was assigned to me of globe-telecom. Font matching has to be done of globe telecom. Ticket status is now in progress.

8/3/17

Two new tickets were assigned PSFE-233 and TPMMB-1389. These were again bug issues. In one i.e. first ticket, I had to learn about front faces and then apply on the Globe-environment site so that it matches with the globe website.

9/3/17 - 10/3/17

The PR was created for one the tickets and the research work was started for the second ticket. The ticket was fixed on Friday as certain changes had to be made in the customizer themes repository and PR was created for this respective ticket.

13/3/17 - 15/3/17

Globe theme bug ticket was merged and 3 new tickets were assigned. MetroPCS branding ticket was merged. DT Cloud ticket was assigned and research part was done on Monday and buy was fixed on Tuesday and ticket was fixed as an error was not reproducible. On Wednesday, one more ticket was assigned PSFE-68 and research part was not much and ticket was fixed on the same day as error was not reproducible.

16/3/17 - 17/3/17

Two new tickets were assigned. One ticket was of channel partner Appcelerator and was that during checkout the text was misaligned of payment and changes have to be made in the main AppDirect repository. Changes were made and PR was generated. One more ticket was there which had a monolith issue. The HTML had to be changed for this ticket and this was fixed on Friday itself.

20/3/17 - 22/3/17

PR was reviewed and CR was done for the tickets in the last week. New tickets were assigned. In one of the tickets the error was not reproducible and had to resolve it myself. In second ticket issue was about dropdowns and CTA buttons, so had to change the SCSS and complete the ticket on Wednesday itself.

23/3/17 - 24/3/17

Uber again was reverted to make the media query changes which was so it can be used in mobile also. So, first I had to learn about media queries and after that I had to work on this. One more ticket was assigned which was given a bug issue. So, again font faces had to be used in this, and I already had knowledge about this, so it took less time.

27/3/17 - 29/3/17

One of the tickets of the Appcelerator was reverted as some more changes had to be made. PR was created for Uber and all changes were made as per the higher authorities of the company. Two more branding tickets were assigned on Wednesday and I had to work on them. These were of Socius and TD Ameritrade.

30/3/17 - 31/3/17

Finally Uber was said good to go and was finally merged. The ticket which I got on Wednesday was completed and PR was generated for that ticket. This was of Socius branding. On Friday, the code review was done and this ticket of Socius was finally merged.

3/4/17 - 5/4/17

TD Ameritrade was sent (PR) and it was merged after CR. Changing roles task was assigned to me by one of leads in San Francisco. The whole 2 days went in the research part of this task. A completely new repository had to be studied to complete this task. On Wednesday, I completed this task for one particular customer and had to move my work to mentor's branch. No new PR was generated as this task was merge to the mentor's branch.

6/4/17 - 7/4/17

A new ticket was created for the changing roles components and new subtasks were given in that ticket. Last week I had to change a single role but in this I had to change multiple roles. One more similar ticket was assigned to change roles for ADP. So it was completed on Friday.

10/4/17 - 12/4/17

One new ticket was assigned to me on Monday. This was a bug fix issue for a channel partner Cancom. Javascript error was there in this issue. Monday was basically to do the research and reproduce the error. On Wednesday the ‘Sign Up’ error message not reproducing second time was connected.

13/4/17 - 14/4/17

On Thursday, I was assigned a powerplay branding. In a powerplay, we have to design the marketplace exactly similar to the company's website. This was third powerplay of mine. This time it was bank of America. Business URL was- <https://www.bankofamerica.com/smallbusiness/>. I had to build this similar, in Marketplace URL- <https://bankofamerica.byappdirect.com>.

17/4/17-20/4/17

The site was completed on Tuesday and final changes were made on Wednesday. In this period only, there were changes suggested in 2 of previous tickets, so I had also connected those. But most of the time was devoted to design the powerplay. PR was also generated.

21/4/17

The PR was sent for the CR of the powerplay. A totally new projected was assigned to the 2 of interns, one of them was me. It was about the microservices and a documentation was given to read about the task. We have to design microservice.

24/4/17 - 26/4/17

So we have to learn microservices, how to design and what to design. The research part of what have to be designed took about 3 days. This was about translations of keys (language translations). A microservice had to be created for transaction service. This can be created using any platform Java/Python/NodeJS etc. But we chose NodeJS as that was preferred at the point of time.

27/4/17 - 28/4/17

Microservices task had to be stopped as some urgent marketplaces had to be designed before completion of this week. So, I designed 3 marketplaces during these two days as three tickets were assigned to me- Agio, Caltech, EO Johnson. For Agio: Business URL- www.agio.com Marketplace URL- <https://agio.byappdirect.com> . For Caltech: Business URL- <https://www.caltech.com> Marketplace URL- <https://caltech.byappdirect.com> . For EO Johnson: Business URL- <https://www.eojohnson.com> Marketplace URL- <https://eojohnson.byappdirect.com>. PR's were created for all these marketplaces.

2/5/17 - 3/5/17

These two days went in adding products for SAP marketplace. This was a whole new task. First we had to learn how to add a product after that the main work of adding was assigned. SAP marketplace will be a huge customer if the marketing team wins it. The whole AppDirect force was inclined towards SAP marketplace these days. Zayo MarketPlace was also given to design.

4/5/17 - 5/5/17

The 3 marketplace's PRs got merged and one previous issue of Appcelerator had to be rebuilt due to change in the structure. So first I had to understand the new structure to first reproduce the issue and after that had to fix that issue and rebasing issues were also there. So a lot of time was invested in fixing this issue again.

8/5/17 - 11/5/17

On Monday, we continued with our microservices task, first we created a server and routes in a single file with URL extension. Then we were told to modularize the whole service. Instead of URL extension, we had to send a whole object which contained partner, variation, locale (language), and keys which needs to be translated. First we used get request and for object we used post requested in which we could transfer more data in a secure way. Completed Zayo and created PR.

12/5/17

On Friday, we completed the microservice task and pushed to the branch of the AppDirect translation repository. The next task was to deploy and we had to talk to the lead who gave us this task on Monday, what can be done next on this service. Zayo was also merged today.

15/5/17

Microservices task was just that we had to build this service, the deployment was an architectural part and a planning has to be done how to deploy that.

16/5/17 - 18/5/17

Two new marketplaces had to be designed by me as two tickets were assigned to my name. These were- Carolina West, Republic wireless. Carolina West : Business URL- <https://www.carolinawest.com> MP - <https://www.carolinawestwireless.byappdirect.com>. Republic Wireless : Business URL- <https://republicwireless.com> MP - <https://republicwireless.byappdirect.com>

19/5/17

Today PR's were generated for both the marketplaces and they were sent for CRs. A change was suggested in Carolina West and I corrected that and amended the changes in the PR.

22/5/17 - 24/5/17

A new issue of SAP was assigned to me in which how the SAP marketplace header search bar behaved. Monday was basically to do the research and see what is the flow and what and where SCSS and script had to be written. On Tuesday, I created a PR for this ticket and it got merged on Wednesday as it was a critical issue. One more issue was assigned today evening of Buckeye Broadband, so also had to do the research part on that issue of mid-tier theme.

25/5/17 - 30/5/17

The HTML and SCSS had to be changed in the Buckeye Broadband marketplace issue. So, on Friday I fixed the issue and created a PR for that. Rest days till today, I had to work on any pending issues which were not merged so that clearance on any project is not there before going for Project Semester Examinations/Viva. Also, the issue of Buckeye Broadband got merged on Monday.

REFERENCES

- [1] "Curate the buying experience". Retrieved from <https://www.appdirect.com/products/appmarket/features/marketplace-manager>.
- [2] "AppDirect Customers" Retrieved from <https://www.appdirect.com/customers>.
- [3] "AppDirect for SaaS" Retrieved from <https://www.appdirect.com/solutions/saas>.
- [4] "Meet the cloud storefront builder" Retrieved from <https://www.appdirect.com/products/appmarket/features/developer-center>.
- [5] "Full service Cloud Monetization Suite" Retrieved from <https://www.appdirect.com/about>.
- [6] Williams, Alex (9 July 2012). "GitHub Pours Energies into Enterprise – Raises \$100 Million From Power VC Andreessen Horowitz". Tech Crunch. Andreessen Horowitz is investing an eye-popping \$100 million into GitHub
- [7] Binstock, Andrew (20 May 2015). "Java's 20 Years Of Innovation". Forbes. Retrieved 18 March 2016.
- [8] Berners-Lee, Tim. "Information Management: A Proposal." CERN (March 1989, May 1990).
- [9] Meyer, Eric A. (2006). Cascading Style Sheets: The Definitive Guide (3rd ed.). O'Reilly Media, Inc. ISBN 0-596-52733-0.
- [10] Kataria, Saransh. "Getting started with sass development". wisdomgeek. saranshkataria.
- [11] Severance, Charles (February 2012). "JavaScript: Designing a Language in 10 Days". Computer. IEEE Computer Society. 45 (2): 7–8. doi:10.1109/MC.2012.57. Retrieved 23 March 2013.

- [12] Guthrie, Scott (2008-09-28). "jQuery and Microsoft". ScottGu's Blog. Retrieved 2009-01-29.
- [13] Richardson, Chris. "Microservice architecture pattern". microservices.io. Retrieved 2017-03-19.
- [14] Chen, Lianping; Ali Babar, Muhammad (2014). Towards an Evidence-Based Understanding of Emergence of Architecture through Continuous Refactoring in Agile Software Development. The 11th Working IEEE/IFIP Conference on Software Architecture(WICSA 2014). IEEE.
- [15] Hitchens, Ron (Dec 2014). Swaine, Michael, ed. "Your Object Model Sucks". PragPub Magazine. Pragmatic Programmers: 15.
- [16] Other documentations of AppDirect the links of which cannot be disclosed.