CSCE 431 Final Report: Mporium

Hallie Scasta, Derik Wang, Keegan Choudhury, Anu Khatri

1 Project Summary

Aggie Marketplace is an existing product that has a set of limitations and problems that lead to consumers wanting a redesigned user interface or a whole new product. The marketplace is home to a plethora of products from various organizations in which students and faculty wish to purchase. This is why it is so important to create a marketplace that is both aesthetically pleasing as well as provides the functionality for organizations to set up product pages and sell to their customers, as well as the ability for members of organizations to be able to sell their quota for merchandise. Thus, our team, Mporium, set out to create a new product marketplace that allows for easier comprehension of products and interaction between organizations and members.

The stakeholders within this project would be both the organizations and community of Texas AM and its entities, this means the team are trying to allow organizations to sell merchandise to whoever. In order to meet this need the team made both a client side account system as well as a member and organization side account system. This allows organizations to add members that belong to them and allow them to create products to sell while being able to keep track of both member and organization-wide statistics on sales. Furthermore, this provides a cleaner organization system when it comes to selling products to friends by referring them and tying to purchases to a specific member. The client side allows users to browse by various parameters such as which organization, which type of product, and more allowing for a dynamic search feature.

2 User Stories

Through the use of Pivotal Tracker, the team were able to implement an agile development life cycle in which each sprint consisted of a certain amount of user stories that were assigned to each individual. The first set of user stories were to implement the basic functionality of the web page, which consisted of creating a functional web page coupled with a header that would allow the user to navigate between the separate pages of the website. Each page serves a unique purpose with different information. Both of these user stories were 1 point due to the simplicity of it; however, in later sprints the team added more points to refining the UI and functionality of each of the items. Next, the team had to set up the creation of user accounts - both those of organizations as well as those of members and the linking of a member to an organization. Each of these user stories were 2 points each due to the importance of correctly relating all these entities to one another. The account creation process was a building block of our whole application thus had to be implemented carefully. The team initially used inputable information that the user would create an account with login credentials; however, through later sprints, the team implemented Google authentication that was worth 8 points in order to make it more secure, as well as allow for a faster account creation process. Next, the team had to create the user stories for product creation, being able to add an item to the marketplace as a seller, which is the whole basis of our application. Due to the importance of this action, the team gave it 3 points so the assigned individual would spend careful time understanding how the user is able to add a product, as well as making the user interface accessible.

After the team incorporated the addition of products to the marketplace feature, the team created a user story for customers to checkout the products through PayPal that was worth 5 points. The team changed this process throughout the development process to include more mutable features, such as looking at inventory and creating a shopping cart to store various items. Next, the team had to create an admin dashboard to display individual organization statistics so they could track information. Due to the importance and complexity of the information/graphs that needed to be displayed, the team made this a 7 point story to implement.

Hand in hand with this story, the team added a manage an organization's members option, so that they were able to manipulate and change member information for 2 points. In addition, a sales report feature that would allow organizations to see average sale statistics including price, quantity, and more was also completed for 4 points.

Next, the team created the actual product creation page that would allow administrators of organizations to attach an image and tag the image with some information to sell it on the marketplace for 5 points. Along with that, product tags were included to help sort and sift through the marketplace, such as various categories of merchandise, for 4 points, and a product cost option was added for 2 points that would allow the customer to add items to their cart creating a final total to checkout. Next, the team implemented the search feature to look for products by tag or cost for 4 points; this allows users to find the type of item they are looking for efficiently. In addition, the team added cart manipulation in case users either decide they no longer want a product, or they accidentally add a product; this allows users to add and remove products for 4 points and 5 points respectively. Google authentication was one of our important features since users feel more comfortable using Google for security reasons, as well as the efficiency of being able to sign up with the click of a button. Thus, the team implemented sign up and log in functionality through a Google account for 8 points.

Next, the team added the functionality of being able to add multiple of one item or manipulating the quantity of an item in case the user is making a multi item purchase for 7 points. This goes hand in hand with our inventory management system since the more you add, the more it takes away from the remaining items. Next, the team implemented the actual shopping cart function with the PayPal API. the team are able to total the items selected in the cart and send that value to the PayPal API, and then the customer is able to login and pay for the goods to ultimately finalize the purchase for 4 points. The next feature was a facilitator that allows members to give their referral code to friends or family in order to purchase items with their “name” on the marketplace. so they get credit. Allowing for referral codes allows incentive for selling more goods and keeps track of individual users statistics for 7 points. This also allows organizations to be able to see which members are participating in their fundraising goals, as well as they are able to keep track of requirements and minimum sales. This goes hand in hand with our next user story, which tracked individual sales for 4 points so that if organizations have a requirement of how many items each user has to sell, they could track that as well. Lastly, the team had the most important user story of all, which was implementing RSpec and cucumber tests as well as making sure that the existing tests all passed to ensure the integrity of the application and overall functionality of the marketplace. This user story was 8 points as it was the most important to test step by step the different actions the team expect the users to follow to ensure that the marketplace is functioning as intended and users are able to sign up / create an account, post different products, purchase different products, track organization and individual statistics as well as interact with the organizations of am as a whole to ensure a closer knit community.

Feature: Create an organization to Mporium marketplace
Points: 2
Implementation Status: Completed

Feature: Create a member to Mporium marketplace
Points: 3
Implementation Status: Completed

Feature: Create an welcome page to Mporium upon login
Points: 1
Implementation Status: Completed

Feature: Create a functional header
Points: 1
Implementation Status: Completed

Feature: Add a product to Mporium marketplace
Points: 5
Implementation Status: Completed

Feature: Add a search feature to Mporium marketplace
Points: 3
Implementation Status: Not Completed

Feature: Track individual sales within an organization
Points: 5
Implementation Status: Completed

Feature: Add additional members to the organization

Points: 2
Implementation Status: Completed

Feature: Create a Functional Landing Page
Points: 2
Implementation Status: Completed

Feature: Create a Functional Login Page
Points: 1
Implementation Status: Completed

Feature: Create the Sign Up Page
Points: 2
Implementation Status: Completed

Feature: Organization Sign Up
Points: 2
Implementation Status: Completed

Feature: Member Sign Up
Points: 2
Implementation Status: Completed

Feature: Dashboard Pages
Points: 3
Implementation Status: Completed

Feature: Organization Member Data
Points: 3
Implementation Status: Completed

Feature: Manage Organization Members
Points: 3
Implementation Status: Completed

Feature: Admin Dashboard
Points: 8
Implementation Status: Completed

Feature: Product cost
Points: 3
Implementation Status: Completed

Feature: Product tags
Points: 5
Implementation Status: Completed

Feature: Shopping Cart (using API)
Points: 5
Implementation Status: Completed

Feature: Google Authentication
Points: 8
Implementation Status: Completed

Feature: Referral Code for Members to Customers
Points: 8
Implementation Status: Completed

Feature: Sorting by organization and product tag and by price
Points: 5
Implementation Status: Completed

Feature: Add product quantity that tracks with the checkout cart
Points: 8
Implementation Status: Completed

Feature: Checkout products on the Marketplace
Points: 5
Implementation Status: Completed

Feature: Add products products to cart
Points: 5
Implementation Status: Completed

Feature: Remove products from cart
Points: 8
Implementation Status: Completed

Feature: Product In Stock / Inventory
Points: 8
Implementation Status: Completed

Feature: Final Demo
Points: 8
Implementation Status: Completed

Feature: Final Report
Points: 8
Implementation Status: Completed

Feature: Final Poster
Points: 8
Implementation Status: Completed

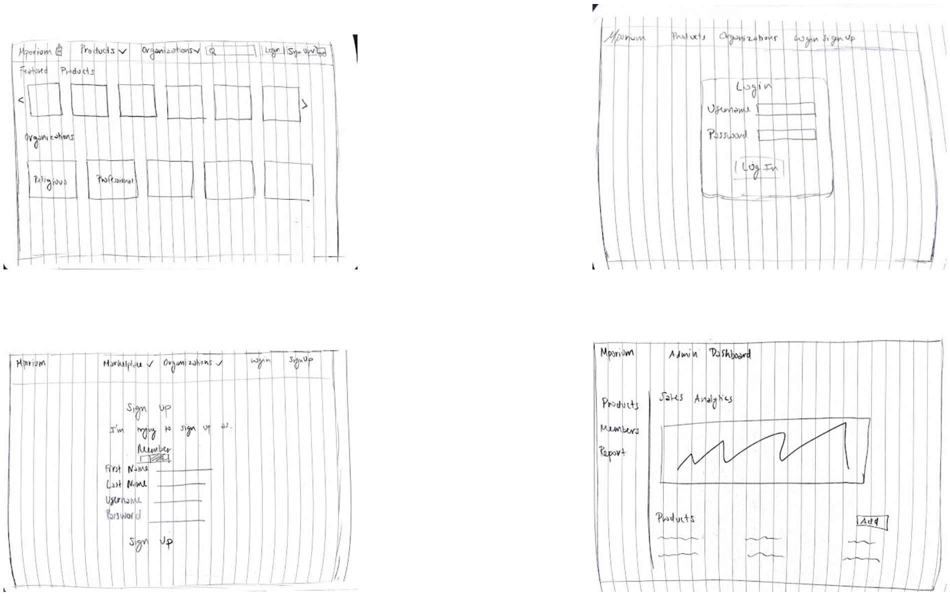

Figure 1: Mock-Ups of the Marketplace, Log-In, Sign-Up, and Organization Dashboard

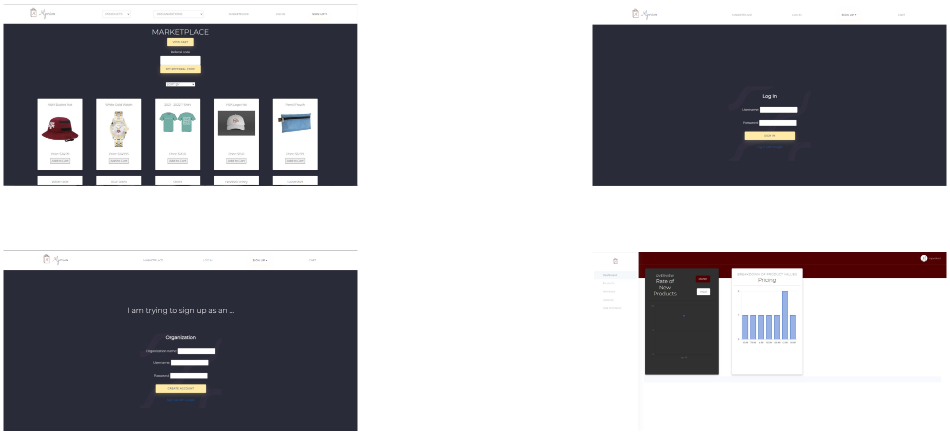

Figure 1: User interface of the Marketplace, Log-In, Sign-Up, and Organization Dashboard

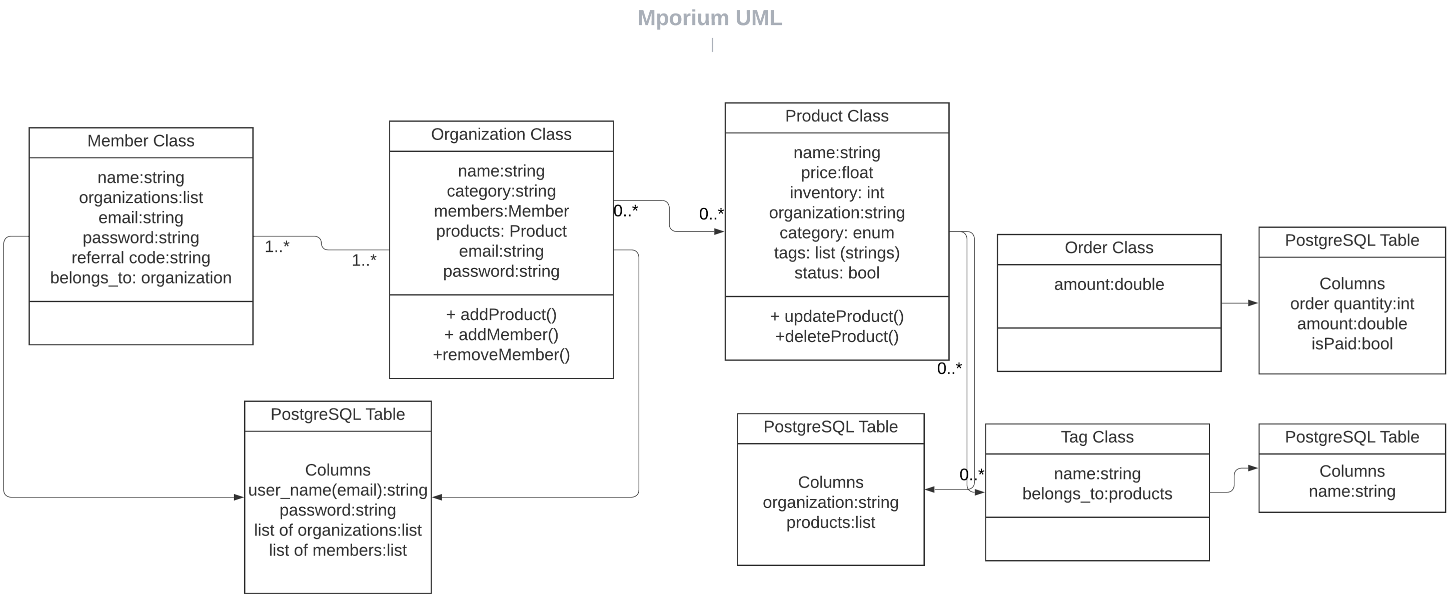

Figure 2: Design UML Diagram

3 Team Roles

Throughout the project, Keegan served as the Scrum Master, Anu served as the Product Owner, Hallie served as a Software Developer, and Derik served as a Software Developer. Keegan focused on the front-end design for the landing page as well as the various content pages while Hallie implemented the admin dashboard as well as the various statistics pages to relay important information to the organization admins. Derik mainly focused on the back end routing and development process while Anu focused on the incorporation of authentication techniques for various users. All members of the group communicated with one another to complete features for the application.

4 Iteration Summaries

Iteration 0

During Iteration 0, the team finalized the project concept, developed general user stories, and created lo-fi mock ups of the product marketplace and account creation. No code was actively being written in during this iteration, as Rails was recently introduced; however the team was able to grasp the scope of the application being created. Furthermore, this iteration also consisted of account creation for PivotalTracker, Code Climate, and AWS to be used throughout the following iterations.
Points Completed: 0 points

Iteration 1

During Iteration 1, the team created the application file system with Rails. The team was able to deploy a working application that included the sign-up and log-in routes for organizations. Furthermore, the application also included the navigation bar that

allowed the users to navigate throughout the application. The logo for Mporium was also established and a design diagram of the early iteration of the product was created. This included the Organization, Member, and Product classes as well as their respective tables in the PostgreSQL server that will store the information.
Points Completed: 18 points

Iteration 2

During Iteration 2, there was a larger emphasis in RSpec and Cucumber testing to ensure that users of the application were able to use the product as expected. Furthermore, the team also introduced the aspect of a sales dashboard for organizations to view how much product was being sold in addition to seeing how many products were being added to Mporium. Furthermore, the team also completed the product upload functionality that allowed organizations to attach products to themselves and display for customers to buy.
Points Completed: 23 points

Iteration 3

During Iteration 3, the team introduced the member dashboard that simply notified the members the they were logged into Mporium. Furthermore, the team also edited the products data table to allows organizations to also add in the cost of the product and the tags for the product such as clothes or shoes. Lastly, the team also updated and redesigned the marketplace so that it would look more professional for customers when they are coming to the website.
Points Completed: 24 points

Iteration 4

During Iteration 4, the team incorporated the PayPal API to allow customers to check out the products that they would like to purchase. Furthermore, organization accounts are able to add members to their organization based on the emails of the member accounts. After this iteration was completed, the Mporium website was finally gaining functionality similar to an actual Marketplace where organizations are able to upload their products for customers to purchase.
Points Completed: 19 points

Iteration 5

During Iteration 5, the team incorporated several features that improved the user experience for Mporium. Google Authentication was added for organization and member accounts to sign up and log in with. Furthermore, a referral code was generated for members upon account creation. Members can give their referral code out to people, and purchases are able to be tracked whenever a referral code is used by customers. Lastly, sorting features were also added into the application to improve user experience. Customers can find products by organizations, tags, increasing price, or decreasing price.
Points Completed: 55 points

Post Iteration

After Iteration 5, the team created the following report, the final poster for the presentation, and videos for the demo and poster.
Points Completed: 24 points

5 Customer Meetings

Iteration 0 Meeting - December 30th, 2021 at 12:30pm
At the Iteration 0 meeting, the team gave a summary on the problem that Mporium is aiming to solve. User stories the team hoped to implement throughout the project were highlighted to the stakeholder. Additionally, a few mock ups for the features were presented. The stakeholder mentioned that the product could be configured to be applicable for different universities depending on the styles and logo used on the site.

Iteration 1 Meeting - January 5th, 2022 at 10am
At this customer meeting, the team had started the implementation of the product, so the team demoed the landing page, as well as the login and sign up page for a user (there was no distinction between an organization and member at this point). the team also demonstrated the welcome page after a user logs in. The RSpec and Cucumber tests were run and shown to the stakeholder. The stakeholder mentioned the possibility of favoriting organizations a customer would like to buy from, but the team had decided that creating accounts for customers in order to do that would be a reach goal. The stakeholder also mentioned creating an organization leader board so that all organizations could see the statistics of how other organizations are doing, as well as incite other organizations to join because of that. Google Authentication also needed to be implemented at a future date.

Iteration 2 Meeting - January 24th, 2022 at 1pm
At this customer meeting, the team demonstrated having separate organization and member sign up pages. Additionally, the team demonstrated drop downs for products and various organizations on the navigation bar. Lastly, the team demonstrated the ability for organizations to view the list of members part of their organization, as well as the ability to add/delete members. the team presented an updated design diagram for the application. The stakeholder mentioned adding tags to products and a cost versus selling price for organizations to calculate to reach their fundraising goals. Additionally, it was advised to contact the inventory group in the class to see if the team would be able to use their products for our inventory purposes.

Iteration 3 Meeting - February 4th, 2022 at 1:30pm
At this customer meeting, the team demoed the organization dashboard after logging in as an organization, as well as the sign up for a member leading to their own dashboard. the team also showed that the products now have an associated cost and tags. Lastly, the team presented further improvements in the user interface of the marketplace. the team also ran the RSpec and Cucumber tests and displayed the results.

Iteration 4 Meeting - February 18th, 2022 at 4pm
At this customer meeting, the team demonstrated the implementation of a shopping cart through an API. Additionally, the team presented additions to the organization dashboard, including sales analytics and the ability to view and add new members. the team also showed the RSpec and Cucumber tests. The stakeholder mentioned that the marketplace should now be populated with 50 or more items with the ability to filter and sort the products. Additionally, there should be enough historical data so that during the demo, there is enough data.

Iteration 5 Meeting - April 25th, 2022 at 4pm
At this meeting, the team demonstrated the Google Authentication for the organizations and members, as well as the addition of a unique referral code for each member. the team also presented inventory tracking and a shopping cart. Lastly, the team demonstrated the ability to sort by price and product tags. the team also showed the RSpec and Cucumber tests The stakeholder mentioned the application needs to have more data, including more organizations and more products. Additionally, it was mentioned that there could be leader board for member referrals on the organization dashboard.

6 Testing

TDD

Creating this large scale of an application leads to constant testing needing to be done to ensure the product is working properly and as intended. If new code is continually being pushed out without testing the product as a whole at iteration, there can be loss in code quality, functionality, as well as completely breaking various features of the product. the team found this out the hard way realizing that every iteration the team would need to create new set of RSpec tests in order to test each separate function and when the team tried to add new things it would sometimes break the old features from previous iterations. This led to constant need for testing each step of the product using strict test cases in RSpec in order to ensure each user story is working as it was intended. For TDD, the team broke down each user story into subsections in which the team would test each individual portion of the user story. Although this caused us to spend more time coding something that could already be working it saved us valuable time in the long run, allowing us to confirm that each part of the product was working. Thus, each sprint the team utilized TDD in order to ensure the application was up to standards.
Once the application is set up, run ‘bundle exec rspec‘ to run the RSpec Tests.

BDD

In addition to TDD, the team also utilized BDD, that tested the various features of the application to ensure that customers and users would not encounter any errors. By putting ourselves in the shoes of the customer the team were able to ensure that it was both intuitive and quick to be able to sign up for the application and post a

product for an organization. For the organization account the team ensured that they understood the creation process as well as adding various users to their organization so they could track their total sales and metrics as a whole allowing for a dynamic report of activity. For example, the team would make a cucumber test that would follow the user pressing buttons, ensuring they are being redirected to the correct pages or seeing the right text to ensure that the program is correctly following the path. In conjunction with RSpec tests for our BDD, the team were able to ensure that the flow of logic was correct allowing for simplicity when creating and using the Mporium marketplace.
Once the application is set up, run ‘bundle exec cucumber‘ to run the Cucumber Tests.

7 Configuration Management Approach

For our approach toward configuration management for Mporium, the team decided to give each member of the team their own branch on GitHub to create features for the application. Then, the branches were accordingly merged to the main branch after extensive testing through RSpec and Cucumber. Furthermore, there were additional branches made to further test the functionality of the features such as Cucumber, Sandbox, and more. The Cucumber branch was used to learn about and conduct testing on the application, and Sandbox was used to create experimental features. In the end, the team had a total of 14 branches, 6 tags, and 1 final release for the application Mporium. The main branch on the GitHub contains all files for the program, including testing. See Section ‘Deployment‘ to learn how to deploy.

8 Issues with Heroku

Deployment to Heroku caused pain for us in that the operations that were working on localhost and the gems the team used in order to facilitate payments, authentication, and more would not work right away on Heroku. Thus, the team had to make additional changes and configurations with the code to ensure that both the localhost and Heroku were able to use Google Authentication, PayPal Payment portal, and more. For example, sometimes the keys that were using were not production-approved, so the team had to manually request for production approval to be used. Furthermore, the team also utilized a JavaScript bundle library called webpacker, which was deprecated on Heroku. As a result, the team had to rollback our codebase slightly to accommodate for the outdated webpacker hosted by Heroku. Occasionally, the the team would also encounter some issues with the database migration files, forcing us to either truncate or drop a few tables in the Heroku PostgreSQL server for ‘rake db:migrate‘ and ’rake db:seed’ to work properly.

9 Issues with Tools

The team encountered a few issues when deploying EC2 instances from Amazon. Particularly setting up Ruby and rbenv on Ubuntu was tedious. The team did not have many unexpected issues when using GitHub as everyone on the team has had experience with GitHub in the past. However, the team did encounter some occasional issues when merging branches into the main. Some reasons for this is that the team did not always pull from the main branch when starting the new feature. The team encountered no issues using AWS Cloud9.

10 Gems / Tools

Some specialized gems that were utilized during this project were SimpleCov, simple_form, chartkick, omniauth, paypal-checkout-sdk, aws-sdk-s3, and many others. SimpleCov was extremely helpful in testing the application and seeing what lines of code were being hit or not through the coverage/index.html file. The interface was simple to get accustomed to and was user-friendly when trying to comprehend code coverage. Another gem that was useful in creating elements was simple_form. Simple forms creates flexible and powerful rails forms that made is easy to issue post requests while making the forms look visually appealing. The chartkick gem was used to create the simple yet informative graphs for the organization dashboard and member chart. The omniauth gem was used to connect to the Google Authentication system that allowed members and organizations to log-in and sign-up to Mporium with Google. In this way, the user experience was drastically improved due to the ease of clicking a button to utilize Mporium. Next, the team used the paypal-checkout-sdk gem to allow users of Mporium to checkout their cart with PayPal, one of the leading payment systems around the world. Without PayPal, Mporium loses one of its primary functions of selling products from organizations to users of the website. Lastly, the aws-sdk-s3 gem was used to connect to the AWS S3 storage server. This S3 storage server was used to manage all of the product images that were uploaded by organizations to Mporium. When an organization uploaded a picture, a compressed version of the image is stored on the S3 storage server that is called when a user accesses the marketplace.

11 Getting Started

Repository Content

The contents of the GitHub repository include all necessary files required to run the application on localhost and deploy on to Heroku. Furthermore, the RSpec tests can be found in the spec/ folder, and the Cucumber tests can be found in the features/ folder. The repository also includes a setup.sh script and Procfile that is used to deploy the applcation to Heroku.

Set-Up

To run the application on localhost, first verify that Ruby 2.6.6 is installed. Then, install PostgreSQL on to your machine with sudo apt install postgresql postgresql-contrib and ensure that a user exists called postgres with the password ‘12345’. Once Ruby is installed, clone the repository via the git clone command. If one wants to use an alternate username and password for PostgreSQL, the user can change their credentials in the config/database.yml file.

To skip over the reset of the following paragraph, the user can consider running the setup.sh script by issuing the command 'bash setup.sh'. If not, continue following the steps in the rest of the paragraph. Now, run bundle install and rails webpacker:install afterwards. Then, run rails assets:clobber and rails webpacker:compile to generate the necessary JavaScript files. Now, run git reset —hard origin/main to reset the webpacker configuration to the files in the GitHub. Then, run rake db:migrate to run the migration files and create the data tables requires to hold the data for the application and rake db:seed to populate the database. Then refer to the Key Generation section below to generate the 3 sets of keys. Lastly, run rails server to prop the application on to port 3000. To access, the local version of the application go to localhost:3000.

To deploy the application to Heroku, the proper Procfile is already provided in the GitHub repository. To clone the repository, run the respective git clone command. Then, set up a Heroku Postgres Server and set the given credentials to the Heroku server in the config/database.yml file. Then refer to the Key Generation section below to generate the 3 sets of keys. Lastly, the user may push the application to a Heroku repository and access it at the URL: myappname.herokuapp.com. To access the data on Mporium, download the Data. To restore the Heroku Database use 'pg_restore –verbose –clean –no-acl –no-owner -h localhost -U myuser -d mydb latest.dump'. To restore the Amazon S3 Storage data, upload the files back to your respective server.

Key Generation

There are three sets of keys required to run the application. The first set of keys required are the AWS keys generated for the S3 server. To generate these keys, please create an account on AWS. Once created, go to the IAM console and create a user and a set of keys. A access key id and secret access key are created and should be copied into the config/storage.yml file where it states 'access_key_id' and 'secret_access_key'. Ensure that the respective bucket and region are filled in as well based on your configuration of Amazon S3. Then, a set of Google Cloud Platform keys will need to be created as well via https://console.cloud.google.com/getting-started. Once again, go to the IAM console to generate a GOOGLE_CLIENT_ID and GOOGLE_CLIENT_SECRET and copy paste the values into the .env file. The last set of keys are for the PayPal checkout function. To generate these set of keys go to https://developer.paypal.com/, and create a developer account. Then, go to Accounts and create a Sandbox Business account. Then, click on My Apps Credentials and the client_id and client_secret will be provided. Copy and paste these values into the order_controller.rb file where it says 'client_id' and 'client_secret'.

12 Links

Pivotal Tracker: https://www.pivotaltracker.com/n/projects/2547058
GitHub Repository: https://github.com/CSCE431-Greece-Team-3/StuOrg
Heroku Deployment: https://stuorg.herokuapp.com/
Code Climate: https://codeclimate.com/github/CSCE431-Greece-Team-3/StuOrg
Demo Link: https://youtu.be/N5iWmsbxOuE