SW Engineering CSC648/848

Development of a Web application: Milestones

Section 01, Team 01

Lead: Jocelyn Guzman

Backend Lead: Miroslav Stavrev

Frontend Lead: Edel Jhon Cenario

GitHub Master: Hira Afzal

Scrum Master: Ansel Ngai

Milestone 4

05/04/2022

Product Summary:

- Name of the Product is Milestone.

- Our Final P1 features are:
  - User should be able to register
  - User should be able to login
  - User should be able to Post a listing
  - User should be able to search a listing
  - User should be able to have filter options on the homepage, which direct the user to view listings with those filters

- What is unique about our product is that it connects providers and parents directly without having to deal with a middle man. The parents are able to find a provider suitable for their child and have direct contact with them.

Unit Testing Test Plan:

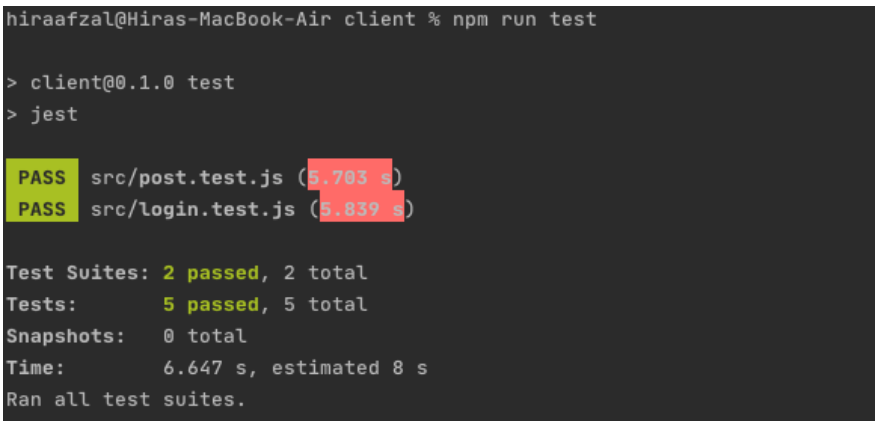
- Test Objectives:
  - 1: Creating accounts: Making sure accounts are properly created for both parents and service providers, no duplicate emails, login pages work for both parents and providers.

  - 2: Creating posts: Posts are created only with the proper information filled out. Information is verified before being added to the database.

  - 3: View large number of posts: Fill the database with generated posts, make sure the code can handle large amounts of data.

  - 4: Filter posts: See only relevant posts based on service type.

- Test Developed: testing P1 features



- For the login page, we developed a test to check the correctness of the form.
- For the post listing page, we also developed a test to check the correctness of the form.

Statement Coverage is a white box testing that tests the execution of all the statements at least once. This way we know what our application can do and cannot do. It is an effective way to test the internal code of the software.

Integration Testing:

Test scenarios and Steps:

1. **User can Register with a name, email, password, whether they are a parent or provider**
    a. Should be in /register page - PASS
    b. Enter information about themselves such as email, password and check whether they are parent or provider - PASS
    c. Click the Sign up button for user to register - PASS
2. **Users can Login with a valid email and password.**
    a. Should be in /login page - PASS
    b. Enter valid info of their email and password - PASS
    c. Click the Login button to log in - PASS
3. **If User is a provider, They can post a listing with their name, service they want to provide, experience, pay rate, and a description about themselves.**
    a. Click the button Create Posts on the Nav Bar - PASS
    b. Should be in the /createPost page - PASS
    c. Enter information that is asked - PASS
    d. Click the submit button to post your listing - PASS
4. **User can search for a service, and get the service listings back to them**
    a. Click on the View Posts button on the Nav Bar - PASS
    b. Should be in the /viewPost page - PASS
    c. On the search bar, type a service to get its listings - PASS
    d. After, listings should be provided to the user. - PASS
5. **Users can click on the card about a service in the homepage and be taken to a list full of providers that are providing that specific service.**

a. Should be in the homepage, / - PASS
b. Click on a service card in the homepage - PASS
c. Be taken to a list of providers providing that specific listing only -PASS

Test id	**Test Case**	**Prerequisite**	**Description**	**Expected Result**	**Test result (pass or fail)**
1	MS_01	Internet access and a browser	A provider Registering an account	User is successfully able to input information and register	PASS
2	MS_02	Browser, valid email and login	A user logging in with their valid email and password	User is able to input valid information and log in	PASS
3	MS_03	Browser, an account already registered and logged in	A provider creating a listing for a service they want to provide and being able to input their information.	A provider user is logged in and can successfully input information about their service and create a listing.	PASS
4	MS_04	Browser, an account already registered and logged in, listing posted	A parent searching a service they want to look for their child using the search bar.	User will be able to get listings back to them after searching on search bar	PASS

5	MS_05	Browser, an account already registered and logged in, listing posted	A user can clicking on a card in homepage and gives a filter of that service to user	User will get a listing of that specific provider	PASS

Analysis of Integration testing:

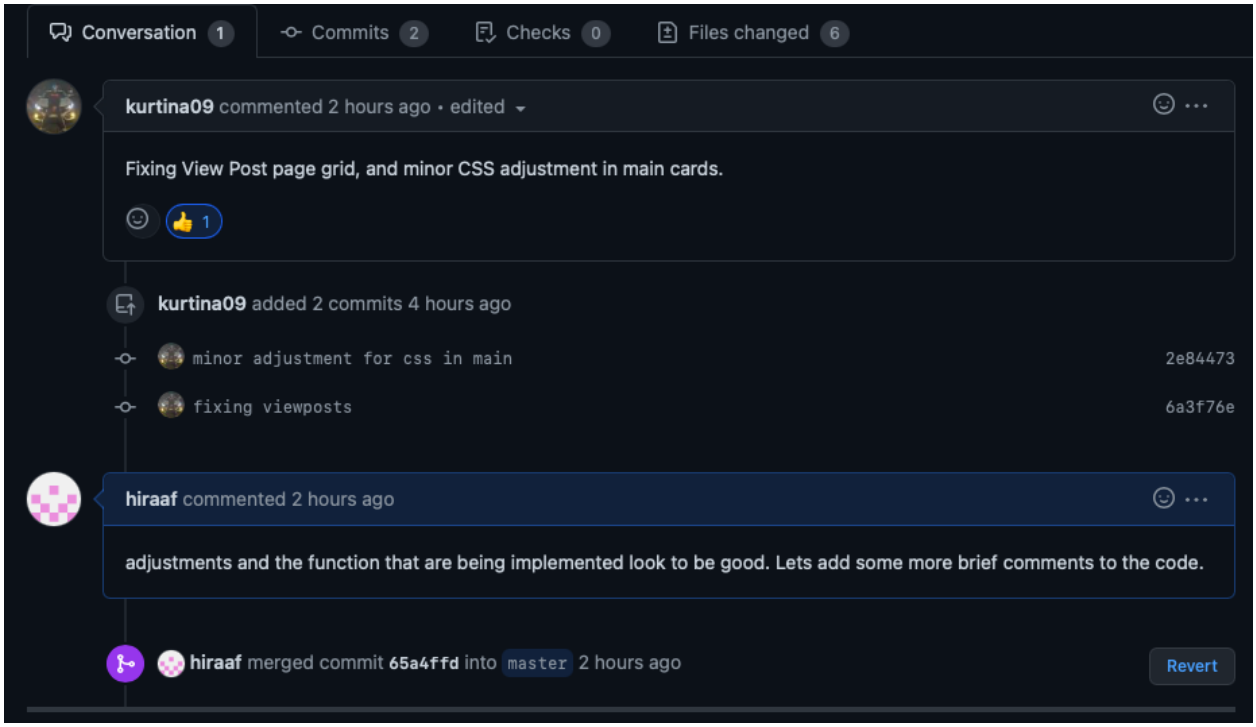
- Users were able to successfully put information such as their email and password and whether they are a parent or provider in testing for the functionality of our registration.
- Users were able to successfully input valid information of their log in such as email and password. This helped us in testing for the functionality of our login page.
- Provider users were able to successfully create a post where they can input information about their service, experience, pay rate and description about themselves.
- Parent users were able to use the search bar for a specific service and have a filtering of that specific service available to them
- Overall, we tested all of our P1 features and were able to pass them in integration testing.

Code Review:

a) Coding Style that we have agreed on is having proper naming conventions such as camelCase with functions and classes. Code Formatting is focusing on proper indentation and spacing between codes. Also making it look as neat as possible. Another is commentating, we have commenting on our code in which they are clear and relevant to the code. This helps other team members be able to look at the code and clearly understand the code's purpose and implementation.

b) Here is a code done by one member about viewing posts, which is a p1 feature. The member was in the feature branch and pushed his changes to

the feature branch. He later made a pull request in which he added the description of what changes were done. It was then reviewed by another member and then merged with our integration branch which is the master branch.



Self-Check All non functional requirements:

- Application should be developed with tools such as Express, React, MondoDb, and Javascript. - **DONE**

- Application should be able to deploy on Google Cloud with a customized url. - **ON TRACK**

- Application should be able to be used in browsers such as Google Chrome or Safari. - **DONE**

- User's privacy is protected with sensitive information such as the password is hashed by Bcrypt - **DONE**

- User’s information is stored in the chosen database, MongoDb. -**DONE**

- The Application is easy and readable to use for the user - **DONE**

- Clear Instructions that guide the user to other pages. -**DONE**

- Application should be able to manage the system and respond to user action efficiently as the app gets more users. - **DONE**

- Rendering pages for the user in a timely manner -**DONE**

- Github branches are used to organize and efficiently merge changes -**DONE**

- Main branch must always be running and working everytime -**DONE**

- Effective collaborative process and practice used. -**DONE**