

Menti

technical Report

Version 1.0

**Start Date: ?? /?? /????**

**Authors : Saransh Sandeepbhai Seth, s3515746**

**Rittidate Kittisarathamma, s3502607**

**Amandeep Nagi, s3503567**

**Qiuwen Yao, s3492704**

**Ahmed Fahad M Alajmi, s3446288**

Document Control

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented** | **Implementation** | **Reviewed** | **Approval** | **Reason** |
| **By** | **Date** | **By** | **Date** |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |

**STAFF or ENTITIES CONSULTED**

|  |  |
| --- | --- |
| **Name** | **Position / Organization** |
| Maria Spichkova | Lecturer-IF, RMIT University |

TABLE OF CONTENTS

1 Executuve Summary 4

2 introduction 4

3 Requirements 5

3.1 FUNCTIONAL REQUIREMENTS SPECIFICATION 5

3.2 NON FUNCTIONAL REQUIREMENTS SPECIFICATION 10

4 architecture 10

3.1 DEVELOPMENT AND DEPLOYMENT ARCHITECTURE 10

3.2 FRAMEWORK ARCHITECTURE 11

3.2 WEBSITE APPLICATION ARCHITECTURE 12

5 TEchnical framework 15

6 IMPLEMENTATION 17

7 Deployment Instructions 24

8 Test specifications 5

9 testing results 5

10 OTHER CONSIDERATIONS 5

11 references 5

# 

# 

# Executuve Summary

This report provides an analysis, evaluation and implementation details of the website, which is implemented to cater the needs of people who are looking for mentors and mentees. This project include the design and construction of the Menti, an exclusive website that allows mentoring become more efficient by using non face-to-face way. The primary aim of this was to design and construct such a website, whose members can successfully contact between mentor and mentee and do mentoring.

Menti is a just in time mentoring resource for time poor individuals who are looking for a better way to grow and move forward. It provides choices in mentoring that suit a variety of people. It captures relevant information of both the Mentee and Mentor and successfully match mentee and Mentor based on given details.

Menti is a chance to grow faster and more powerfully using the wealth of experience of others. It gives you the opportunity to fast track your experience using the experience of wiser heads. You can select a mentor from a list that is provided, of the most compatible mentors for your interest area or you can choose from any of the mentors available. One can also become a mentor in a field that you are experienced and passionate in. All of the users at different times will be a mentor or a mentee.

# introduction

In future, it is anticipated that many of the baby-boomer generation will have retired and 75% of the workforce will be millennials. This new workforce will consist of digitally active and highly educated, but relatively low-experienced persons, aged between 20 and 35.

The solution is mentoring, which provides a way in which one can gain experience and learn quickly, without having to go through the hard mistakes of others. Menti offers a solution to an old problem that is currently addressed manually.

The task is to access the experience of earlier generations in a timely, cost effective and relevant way. The ‘baby boomers’ have seen the most change of any prior generation, and also kept themselves more current in their knowledge and personal growth. Their generation has a wealth of experience to share.

Therefore, this project can successfully bring the best of both generations. It acts as a mean of providing fast and supportive sharing of experience between generations. A just-in-time mentoring resource that offers a better way to deliver experiential learning for millennials. Moreover, the access to generic information and resources is free then the mentor charges their mentees according to the level of services provided to them.

The project is about developing a website that:

* Capture relevant information on both the Mentee and Mentor
* Successfully match Mentee and Mentor based on given values
* Draw out the Mentee’s aspirations
* Offer a pathway for the Mentee to define their desired growth areas

In addition, provides:

* Explicit guidelines for the Mentee-Mentor relationship
* Clear and measurable objectives to be achieved
* Content for further learning
* Opportunities to recognise and reward progress
* A range of mentoring choices that suit a wide variety of people and outcomes.

The stakeholders are:

* Core management Solutions
* Anyone who is looking for a mentoring quick fix or a longer-term solution through an interactive online contact to mentor.
* The development team

Moreover, the target audience for the starting product is technically literate and time poor individuals that are smart and know that they need support.

# Requirements

## 3.1 FUNCTIONAL REQUIREMENTS SPECIFICATION

## 3.1.1 in order to use the website user need to register and login, the system needs to store the user's’ information correctly in data file, which means the system need to validate the information entered by user while registration.

|  |  |
| --- | --- |
| Requirement ID | Description |
| REQ-01 | The system shall let the user to register and confirms that the user has been registered by using their email account. |
|  | If the user haven’t registered for menti website, the system should not let them log in. |
| REQ-03 | The system shall confirm registers’ email cannot be same with previous user. |
| REQ-04 | The system shall confirm every blank in register page is filled. |
| REQ-04.1 | The system shall verify if the email address is valid. |
| REQ-04.2 | The system shall confirm if the username is unique. |
| REQ-04.3 | The system shall send error message if the user is filling wrong date of birth while registration. |
| REQ-04.4 | The system shall confirm if the user has not selected security questions while registration. |
| REQ-05 | The system shall allow the user login with their registered email. |
| REQ-06 | The system shall give error message for filling in wrong password. |
| REQ-07 | The system shall let the user to register through their Facebook account. |

## 3.1.2 The user should be directed to the home page after successful login into the website Home page

|  |  |
| --- | --- |
| Requirement ID | Description |
| REQ-08 | The system shall show the news feed on the bottom of the home page in right order, with every post having its shared time. |
| REQ-09 | The system shall let the user to view various quotes with images on the homepage. |
| REQ-10 | The system shall let the user to access all pages from homepage by clicking on the icons available on the header. |
| REQ-11 | The system shall let the user to go into their homepage by clicking on the home icon available on the header. |

## 3.1.3 The user should be directed to the Dashboard page by clicking on the dashboard icon available on the homepage header.

|  |  |
| --- | --- |
| Requirement ID | Description |
| REQ-12 | The system shall show all the information of Menti including their username, image, categories and mentors should be displayed on the left side of the page. |
| REQ-13 | The system shall let the user type the text in the text box available on the top right hand side of the page. |
| REQ-14 | The system shall let the user to share the post by clicking on the share button, which will appear on the dashboard post area and in news feed on home page. |
| REQ-15 | The system shall let the user to upload and share any image on the dashboard by using the attachment icon available on the dashboard. |
| REQ-16 | The system shall allow menti to rate mentor services by clicking on the stars. Menti can rate mentor only after taking their services. |
| REQ-17 | The menti can follow a mentor by clicking on the follow button on mentor’s profile. |
| REQ-18 | The system shall not allow menti to contact mentor if the mentor is booked out by disabling the contact button on mentor’s profile. |
| REQ-19 | The system shall allow users to view the various areas of mentors on their profile. |
| REQ-20 | The system shall show the information for About, program, terms and price on mentor’s dashboard. |

## 3.1.4 Mentors can be able to limit the number of their mentees, so that when they are full no other menti can contact them.

|  |  |
| --- | --- |
| Requirement ID | Description |
| REQ-21 | The System shall let Mentor change their ‘current availability’ by selecting status from the drop down menu. |
| REQ-22 | The system shall let the users to follow each other by clicking on the follow button. |
| REQ-23 | The system shall let the menti follow mentors by clicking on the follow button even if mentor is booked up or full. |
| REQ-24 | The system shall let Menti send contact request to mentor only if the mentor is not booked out. |

## 3.1.5 Users can visit other’s profile by searching them using the search option.

|  |  |
| --- | --- |
| Requirement ID | Description |
| REQ-25 | The system let the user to use the search option by clicking on the search icon available on the header. |
| REQ-26 | The system shall let the users to search mentors by Firstname. |
| REQ-27 | The system shall let the menti to search a mentor by their category. |
| REQ-28 | The system shall show the Mentor’s Name, Number of mentees, compatibility percentage, areas, location and price information as a result. |

## 3.1.6 Receive Notification.

|  |  |
| --- | --- |
| Requirement ID | Description |
| REQ-29 | The system shall let the user to view the notifications by clicking on the notification icon available on the header. |
| REQ-30 | The system shall show the red coloured bubble on the notification icon in case of any notification. |
| REQ-31 | The system shall remove the red bubble from the notification icon when user read all the notifications. |
| REQ-32 | The system shall send notification to the user after messaging and contacting. |
| REQ-33 | The system shall let the user to delete the notifications. |

## 3.1.7 Both mentor and menti can send message and chat with each other.

|  |  |
| --- | --- |
| Requirement ID | Description |
| REQ-34 | The system shall let the Menti to send contact request to the mentor by selecting the areas menti is interested in. |
| REQ-35 | The system shall let the menti to review the contact request before sending it to the mentor. |
| REQ-36 | The system shall allow the mentor to accept or decline the contact request received from the menti. |
| REQ-37 | The system shall let Mentees send messages to their Mentors. |
| REQ-38 | Menti can only send contact request to the mentor after giving the payment details. |
| REQ-39 | The system shall show the status of the declined contact request as Declined in the notifications when mentor rejects it. |

## 3.1.8 The user needs to add the payment details in order to make the payment for the services taken by menti.

|  |  |
| --- | --- |
| Requirement ID | Description |
| REQ-40 | The system shall allow the user to pay by credit/debit card. |
| REQ-41 | The payment from Menti card will only be start deducting after mentor has accepted their request and start providing Menti the services. |
| REQ-42 | The system shall allow the user to select their default credit card for payments. |
| REQ-43 | After adding the payment method, the system shall validate that user has filled the payment details appropriately. |
| REQ-44 | The system shall let the user to change their default payment card |

## 3.1.9 View and upload the resources

|  |  |
| --- | --- |
| Requirement ID | Description |
| REQ-45 | The system shall let the user to go onto resources page by clicking on the resource icon in the header. |
| REQ-46 | The system shall let the user search for resources in the resource page. |
| REQ-47 | The system shall let the User to upload resources in resource page in .doc, .docx, .pdf, .xls and all images formats. |
| REQ-48 | The system shall list official resources at the left side of the resource page. |
| REQ-49 | System shall list user resources at right side of the resource page. |

## 3.1.10 Editing of details entered by the user in the settings page.

|  |  |
| --- | --- |
| Requirement ID | Description |
| REQ-50 | The system shall let the user upload their profile image. |
| REQ-51 | The system shall let the user change their profile information in personal data page. |
| REQ-52 | The system shall let the user to change their account password. |
| REQ-53 | The system shall confirm the new password is not same as previous one. |
| REQ-54 | The system shall let the user to add and update their payment details. |
| REQ-55 | The system shall let the mentor to edit their mentor details and fee amount. |
| REQ-56 | The system shall let the Menti to set their categories progress. |

## 3.1.11 Logout

|  |  |
| --- | --- |
| Requirement ID | Description |
| REQ-57 | The system shall let the user to logout from the website by clicking on the logout icon available on the header. |
| REQ-58 | The system shall notify that the user has been logged out successfully. |

## 3.2 NON FUNCTIONAL REQUIREMENTS SPECIFICATION

## 3.2.1 Usability

|  |  |
| --- | --- |
| Requirement ID | Description |
| NFR-01 | The system shall let the users to access all of its functionality without any training and referring to the user manual. |

# architecture

**4.1 DEVELOPMENT AND DEPLOYMENT ARCHITECTURE**

There are 3 applications used to control development and to deploy to server:

**4.1.1 Trello**

Trello is used for separating tasks to each member, keep track of the on going, and completed tasks according to each Sprint.

**4.1.2 Github** Github is a source control application and also has powerful functions to connect other applications such as Heroku, which is used for deploying in this project. There are 3 important tasks that Github help the team to collaborate.

**4.1.2.1** Source control: After separating tasks to each team member, each member develop their task on their own branch, then, merge to a sprint branch which is checked before delivering to the client every week.

**4.1.2.2** Each sprint brach is sent to Heroku application to deploy on reviewing server to review the web application’s workflow.

**4.1.2.3** After each sprint approved, they are merged to master brach and deploy to staging server on Heroku.

**4.1.3 Heroku**

Heroku is a cloud platform that lets developers build, deliver, monitor and scale apps. There are 2 main roles for develop and deployment.

**4.1.3.1** Review and test web application.

**4.1.3.2** Deploy on real environment.

# *Figure 4.1. Development and deployment architecture*

**4.2 FRAMEWORK ARCHITECTURE**

Architecture of Ruby on Rails framework has 3 main parts:

**4.2.1 Model**

The model relates to data in database which this project use PostgreSQL as database engine.

**4.2.2 Controller**

The controller can send parameters to the model to update the model's state. It can also send parameters to its associated view to change the view's presentation of the model. It also connect to outside application’s api There 3 applications used in this project.

**4.2.2.1** Facebook Login

**4.2.2.2** Braintree

**4.2.2.3** Amazon S3

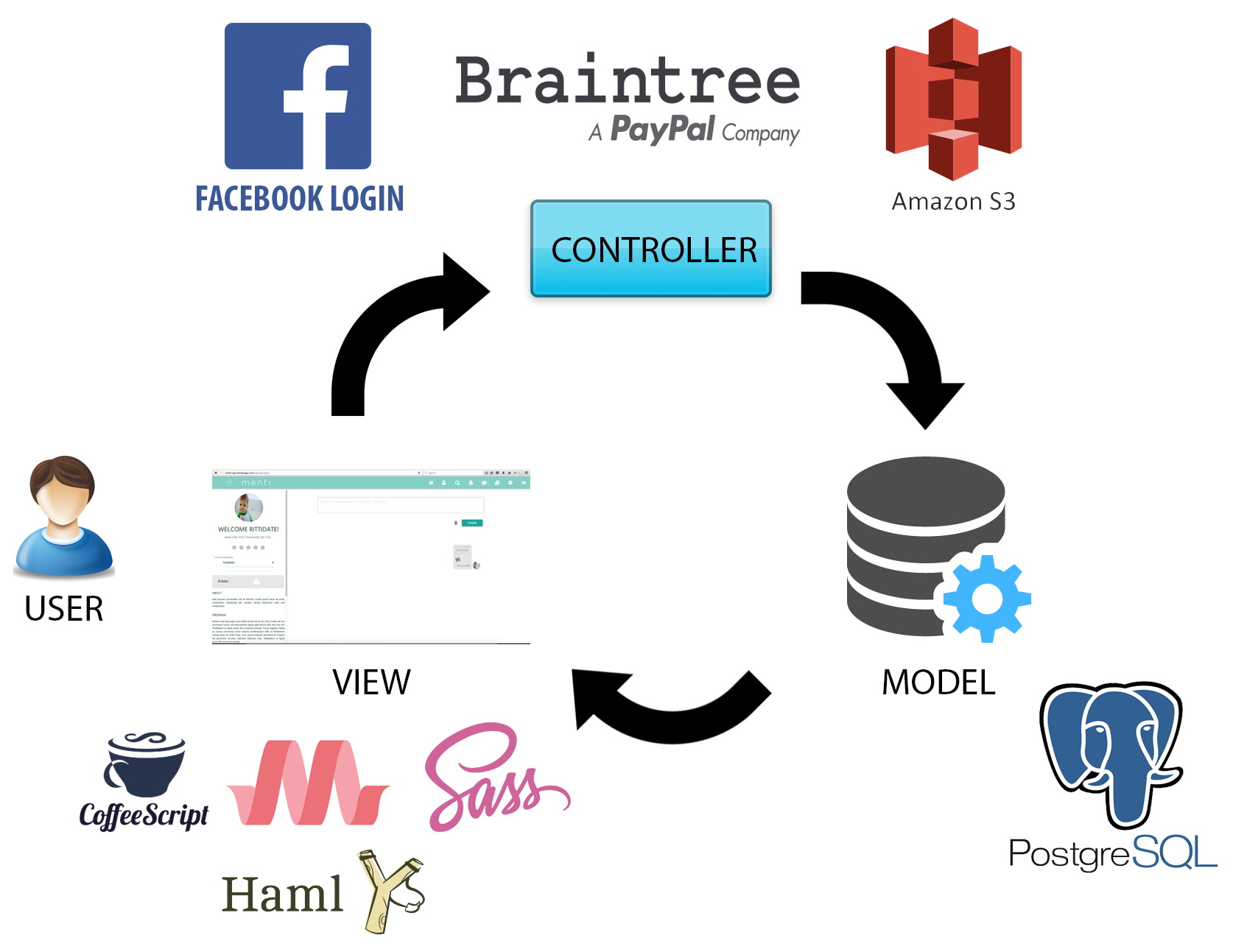
**4.2.3 View**

The view generates new output to the user based on changes in the model. It also interact with a user. There are many extensions that facilitate the developers to display on screen.

**4.2.3.1** Sass is one of CSS component to compile programmatic CSS code.

**4.2.3.2** Materialize CSS is a library to help the view to display in many platforms such as PCs and mobiles.

**4.2.3.3** CoffeeScript is a little language that compiles into JavaScript. **4.2.3.4** Haml is a language to complies into html. It can compile both html and other programmatic language such as ruby and PHP.



*Figure 4.2. Framework architecture*

**4.3 WEBSITE APPLICATION ARCHITECTURE**

In figure 4.3, there are two types of block, dark blocks and white blocks. The dark blocks are display pages, which users can see. The white blocks are pages that get parameters to process with database and return data as JSON.

**4.3.1. Display pages**

**Homepage**: the main page of the website

**Registration**: the users’ registration page

**Login**: sign in page

**Dashboard**: a page that comes after sign in

**My profile**: a page that shows user’s information, desired categories and feed message.

**Search**: a page used to search mentors and mentees matching a keyword

**Notification**: a page to show notifications that notify interacting between mentors and mentees.

**Message**: send message between mentors and mentees.

**Resource**: a page gather all documentation and upload.

**Settings**: User info: update a user information.

Change password: update password.

Default payment: update credit card information.

Category: update desired categories of mentees,

Mentor’s setting: update course and price of mentors.

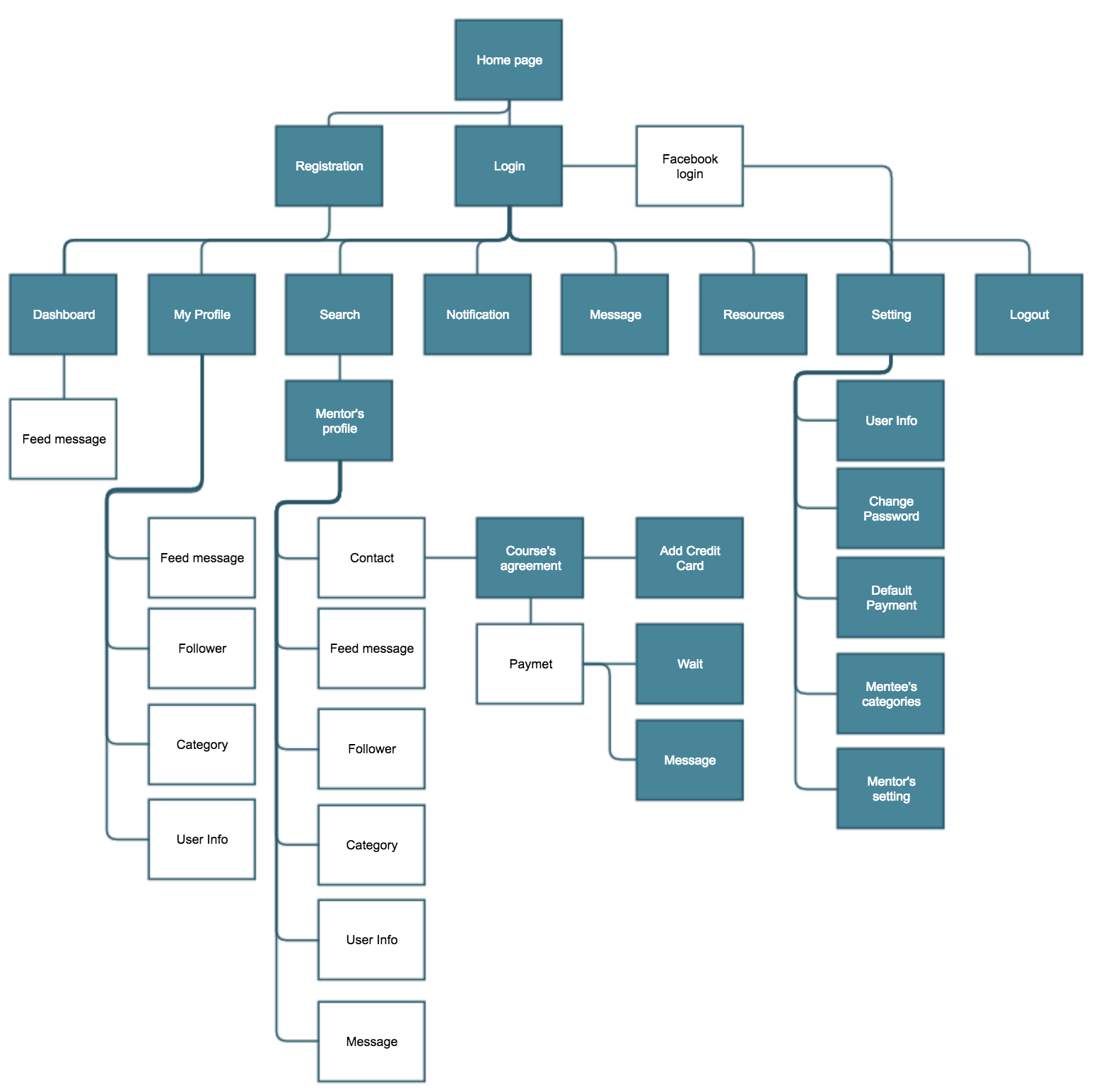
**Logout**: Sign out from user’s authorization.

**Mentor’s** **profile**: mentor’s information, following, feed message and contact.

**Course’s agreement**: agree mentee's request in notification page.

**Add credit card**: add a credit card similar to default payment page.

**Wait**: a page that a user wait for a mentor’s response.1



*Figure 4.3. Site map*

* 1. **DATABASE ARCHITECTURE**

The database has the following tables:

**4.4.1** **Users table**

- username

- password

- email

- user’s info such as address education and preferred language

- user’s type such as mentor, mentee or both

- user profile’s image

- mentor status

### 4.4.2 Categories table

### 4.4.3 Categories users relations table: This table shows a user relates to categories

### 4.4.4 Conversations table

This table shows a link between two users that can send message to each other

**4.4.5 Conversations replies table**

This table shows that a user sends message to a conversation’s room.

### 4.4.6 Conversations histories table

This table shows that a user deleted latest message from a conversation’s room.

**4.4.7 Courses table**

This table shows that a mentor create a course links to a category and define a price.

**4.4.8 Courses user relations table**  
 This table shows that a mentee applies a course of a mentor.

**4.4.9 Feed messages table**

This table is used to create and share a message in a user’s feed

**4.4.10 Follows table**

This table shows that links between followers and a user.

**4.4.11 Identities table**

This table is used to store data from Facebook login api.

### Notifications table

This table shows that notifications are sent from conversations, apply courses and resource functions.

**4.4.13 Payment table**

This table is used to store client’s number which return from Braintree api.

**4.4.14** **Ratings table**

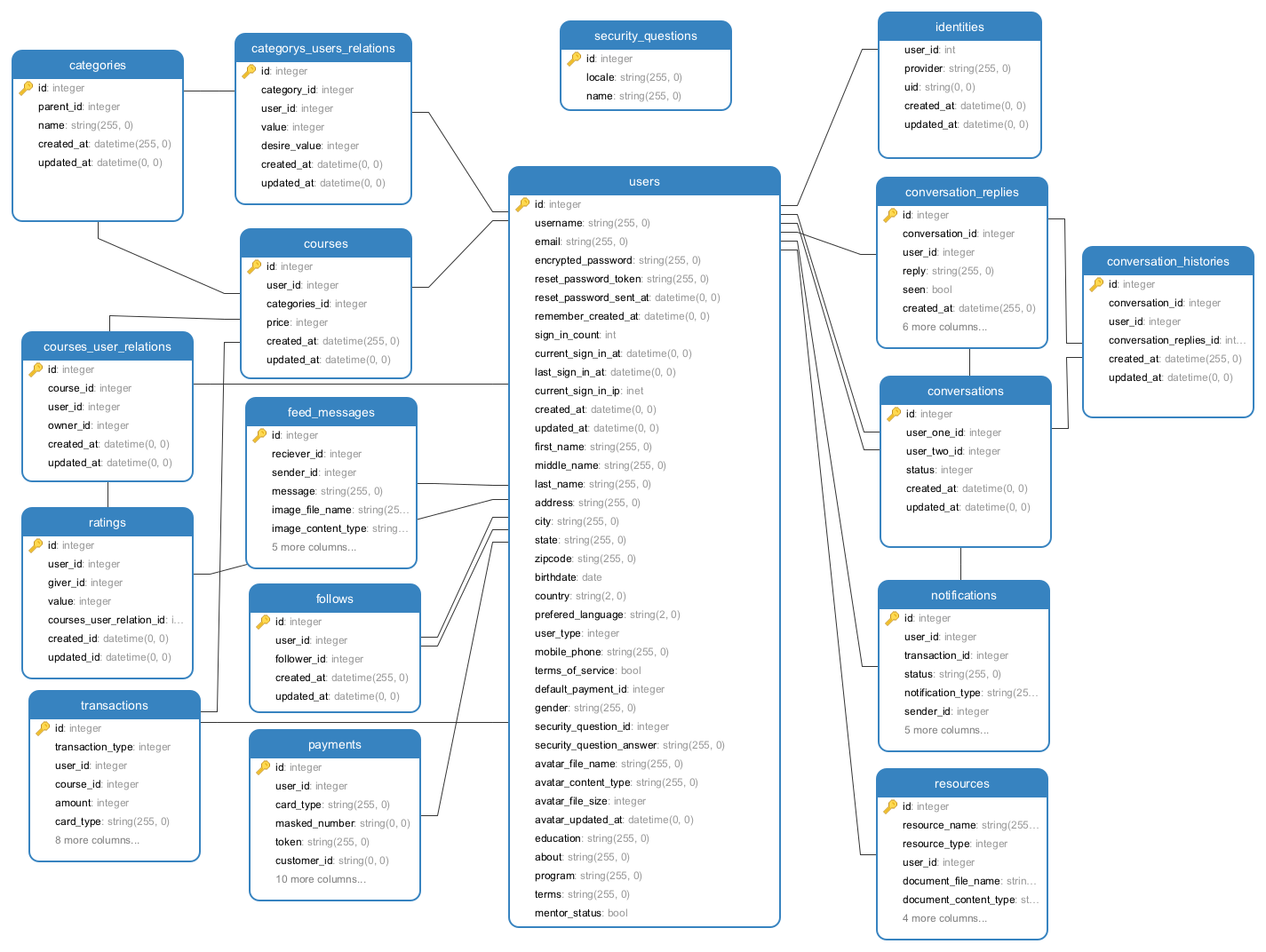
This table shows other users give rating to a mentor.

**4.4.15** **Resources table**

This table is used to store file name and link to show on view.

**4.4.16** **Transactions table**

This table shows referred number getting Braintree transactions.



*Figure 4.4. Database architecture diagram*

# TEchnical framework

This project is developed by Ruby on Rails framework. Ruby on Rails, or Rails is server-side web application framework written in Ruby programming language under the MIT license. Rails is model-view-controller (MVC) framework providing default structures for a database, which this project use PostgreSQL, a web service and web pages. There are many framework in many programming languages but Rails allows developers to write less code while accomplishing more than many other languages and frameworks.

| **Program and extension** | **Version** |
| --- | --- |
| ruby | 2.3.1 |
| Rails | 4.2.6 |
| Postgresql | 0.18 |
| sass-rails | 5.0 |
| coffee-rails | 4.1.0 |
| haml | 4.0.7 |
| materialize-sass | 0.97.7 |
| font-awesome-sass | 4.6.2 |
| devise | 4.2.0 |
| aws-sdk | 1.6 |
| paperclip | 4.3 |
| braintree | 2.49.0 |

*Table 5.1:version details of technologies used*

**5.1 Model**

**5.1.1** PostgreSQL

PostgreSQL is a powerful, open source object-relational database system. It has been actively developed more than a decade and proven an architecture that has earned it a strong reputation for reliability, data integrity and correctness. It also support all major operating systems including Linux, UNIX and Windows. PostgreSQL boasts Multi-Version Concurrency Control(MVCC), point in time recovery, tablespaces, asynchronous replication, nested transactions (savepoints), online/hot backups, a sophisticated query planner/optimizer, and write ahead logging for fault tolerance.

**5.2 View**

**5.2.1** Haml

Haml (HTML abstraction markup language) is based on one primary principle: markup should be beautiful. It’s not just beauty for beauty’s sake either; Haml accelerates and simplifies template creation down to veritable haiku. Haml is used for cleanliness, readability, and production speed.

**5.2.2** Sass

Sass is an extension of CSS that adds power and elegance to the basic language. It allows you to use variables, nested rules, mixins, inline imports, and more, all with a fully CSS-compatible syntax. Sass helps keep large stylesheets well-organized, and get small stylesheets up and running quickly, particularly with the help of the Compass style library.

**5.2.3** CoffeeScript

CoffeeScript is a little language that compiles into JavaScript. Underneath that awkward Javaesque patina, JavaScript has always had a gorgeous heart. CoffeeScript is an attempt to expose the good parts of JavaScript in a simple way. The code compiles one-to-one into the equivalent JS, and there is no interpretation at runtime.

**5.2.4** Materialize

Materialize is a modern responsive front-end framework based on Material Design, created and designed by Google. Material Design is a design language that combines the classic principles of successful design along with innovation and technology. Google's goal is to develop a system of design that allows for a unified user experience across all their products on any platform.

**5.2.5** Font Awesome

Font Awesome gives you scalable vector icons that can instantly be customised - size, colour, drop shadow, and anything that can be done with the power of CSS.

**5.3 Controller**

**5.3.1** Amazon S3

Amazon Simple Storage Service (Amazon S3), provides developers and IT teams with secure, durable, highly-scalable cloud storage. Amazon S3 is easy to use object storage, with a simple web service interface to store and retrieve any amount of data from anywhere on the web. With Amazon S3, you pay only for the storage you actually use. There is no minimum fee and no setup cost.

**5.3.2** Braintree

Braintree is a simple, robust way to accept payments or enable buying from almost anywhere, in your mobile app or online. Braintree’s easy integrations give customers access to multiple payment methods, simple pricing, top-notch protection, and world-class support. Braintree provides its customers with a merchant account and a payment gateway, along with various features.

**5.3.3** Facebook login

Facebook Login gives Menti users a seamless experience across multiple channels. People can register using their facebook account without going into the registration process, to access the Menti website. After registering with facebook, they can fill their other details later on. This is all made possible by Facebook Login.

# IMPLEMENTATION

# 6.1 Sprint 1

**6.1.1** The project was initiated. All the team members met and decided the frameworks and APIs to be used in the project.

**6.1.2** The team installed all the required frameworks and software required for the project. The working environment was ready and set to work.

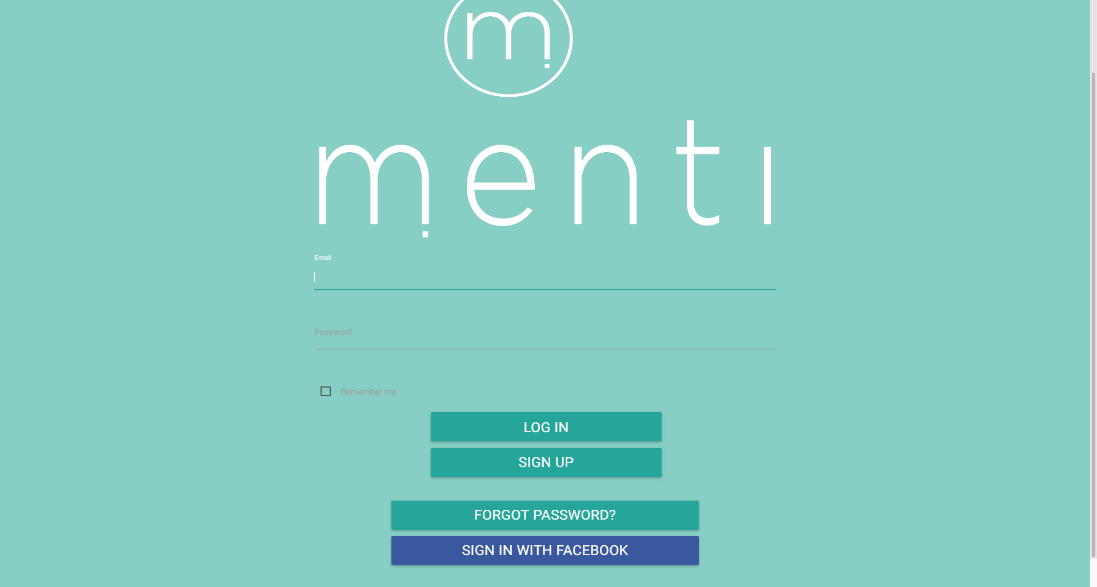
**6.1.3** Landing page was made in this sprint. (*Figure 6.1)*

**6.1.4** Login functionality was implemented where user can login into the website using correct credentials. *(Figure 6.2)*

**6.1.5** Basic registration page was also created to test the login functionality.



*Figure 6.1: Main landing page*

**

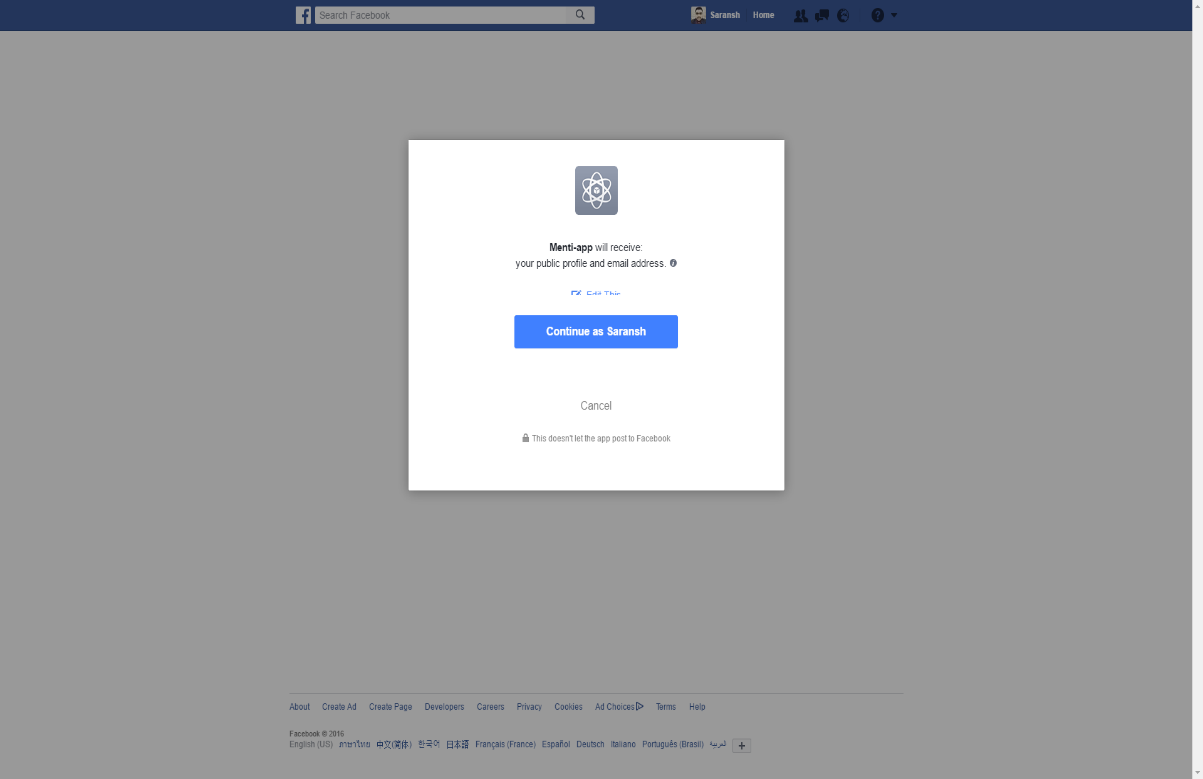
*Figure 6.2: Login page*

# 6.2 Sprint 2

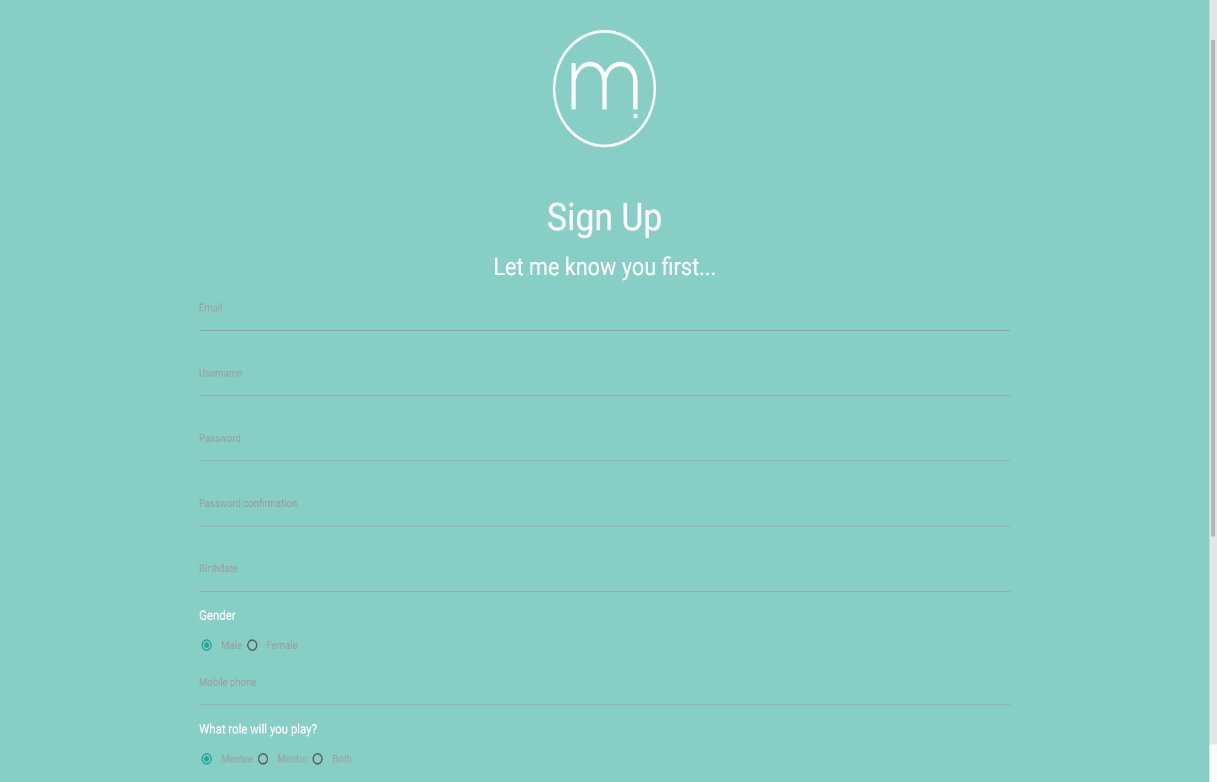
**6.2.1** The login page was made responsive and mobile friendly.

**6.2.2** A better version of registration page was implemented according to the requirements *(Figure 6.4.1 & Figure 6.4.2)*

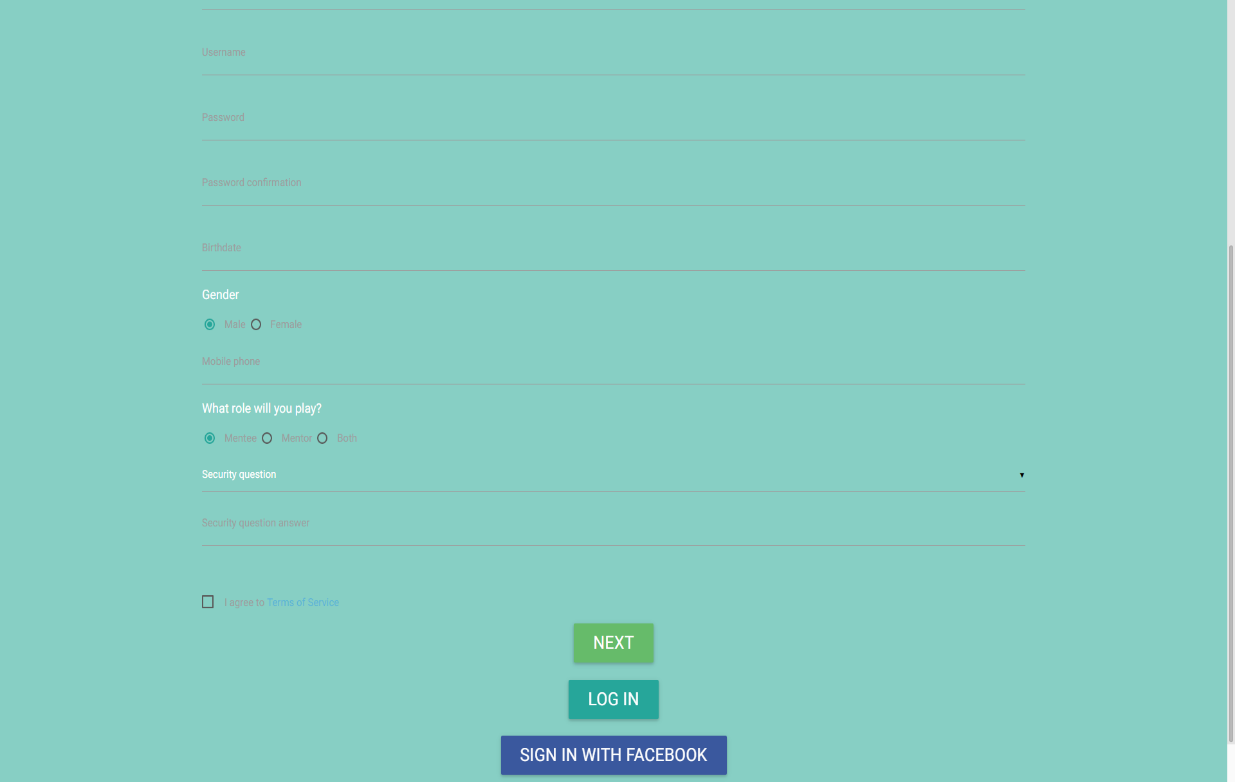
**6.2.3** Sign up with Social media accounts (Facebook) was added in this sprint. *(Figure 6.3)*



*Figure 6.3: Facebook Authentication*

**

*Figure 6.4.1: Registration page*

**

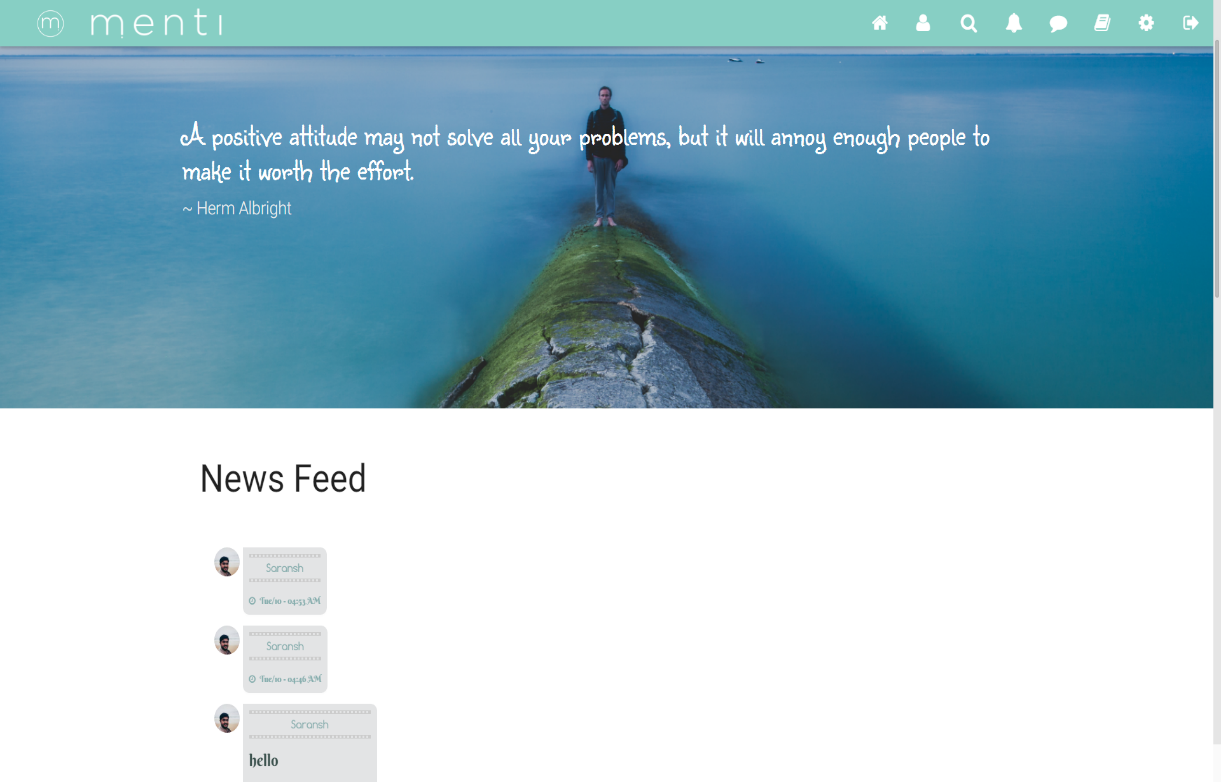
*Figure 6.4.2: Registration page*

# 6.3 Sprint 3

**6.3.1** Initial dashboard for the website was created.

**6.3.2** Registration page was enhanced as requirements changed.

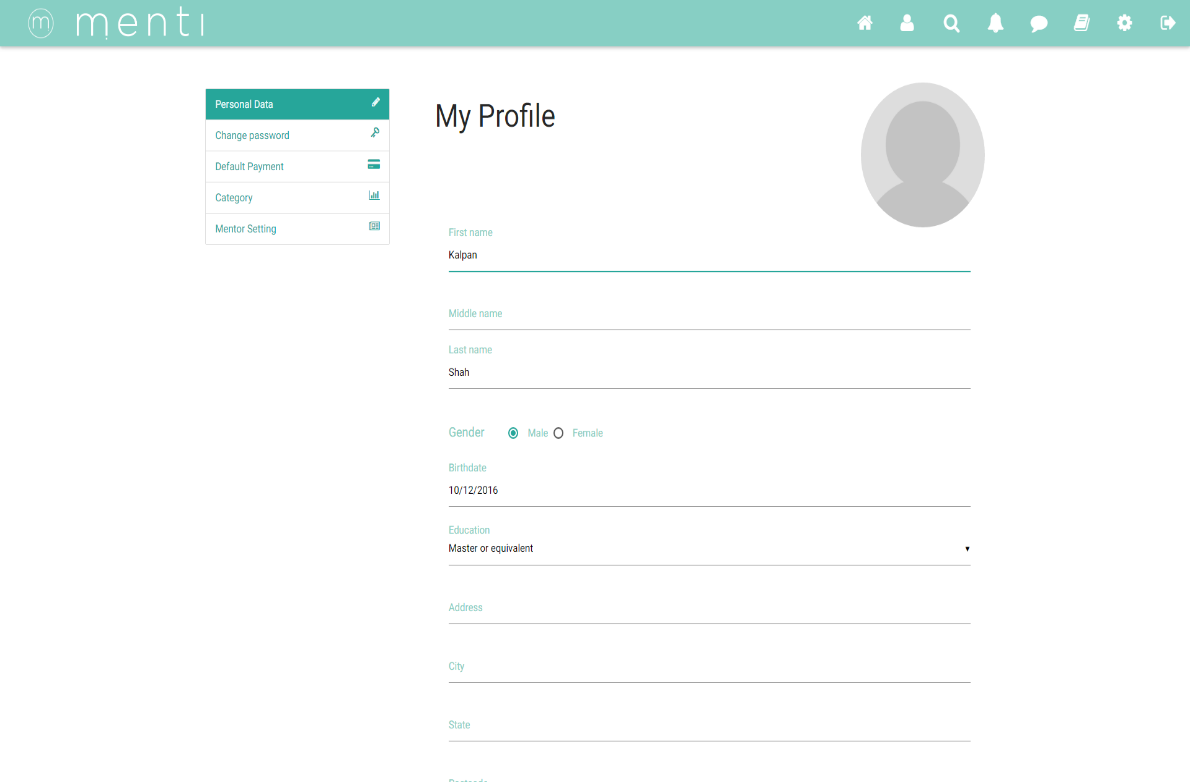
**6.3.3** Back-end for registration page was worked upon.



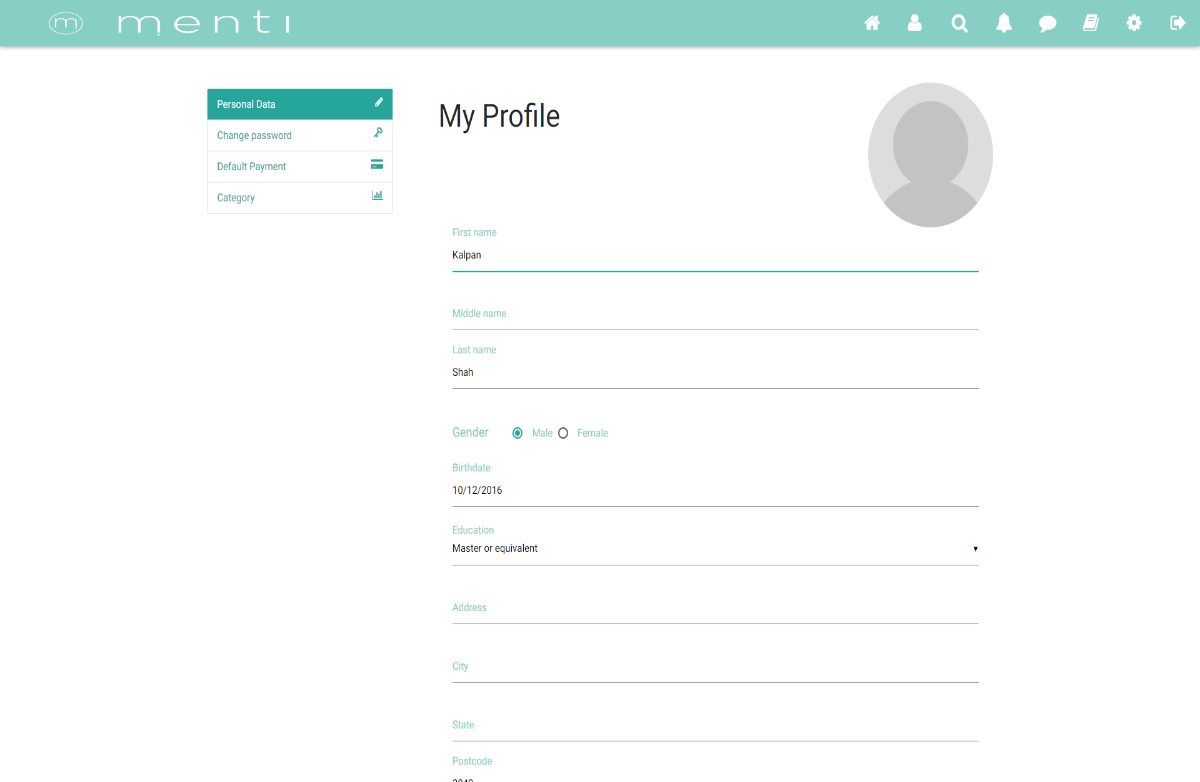
*Figure 6.5: Dashboard*

# 6.4 Sprint 4

**6.4.1** Settings page was created. *(Figure 6.6.1 & Figure 6.6.2)*

**

*Figure 6.6.1: Settings page for mentor*

**

*Figure 6.6.2: Settings page for mentee*

# 6.5 Sprint 5

**6.5.1** Designed ‘My Profile page’. (Figure 6.6.1 & 6.6.2)

**6.5.2** Implemented Payment BrainTree API.

**6.5.3** Enhanced the dashboard. (Figure 6.5)

**6.5.4** Database schema for Feeds and Payment.

# 6.6 Sprint 6

**6.6.1** Initiated Mentor profile design.

**6.6.2** Initiated Mentor profile backend.

**6.6.3** Added Notification icon on header.

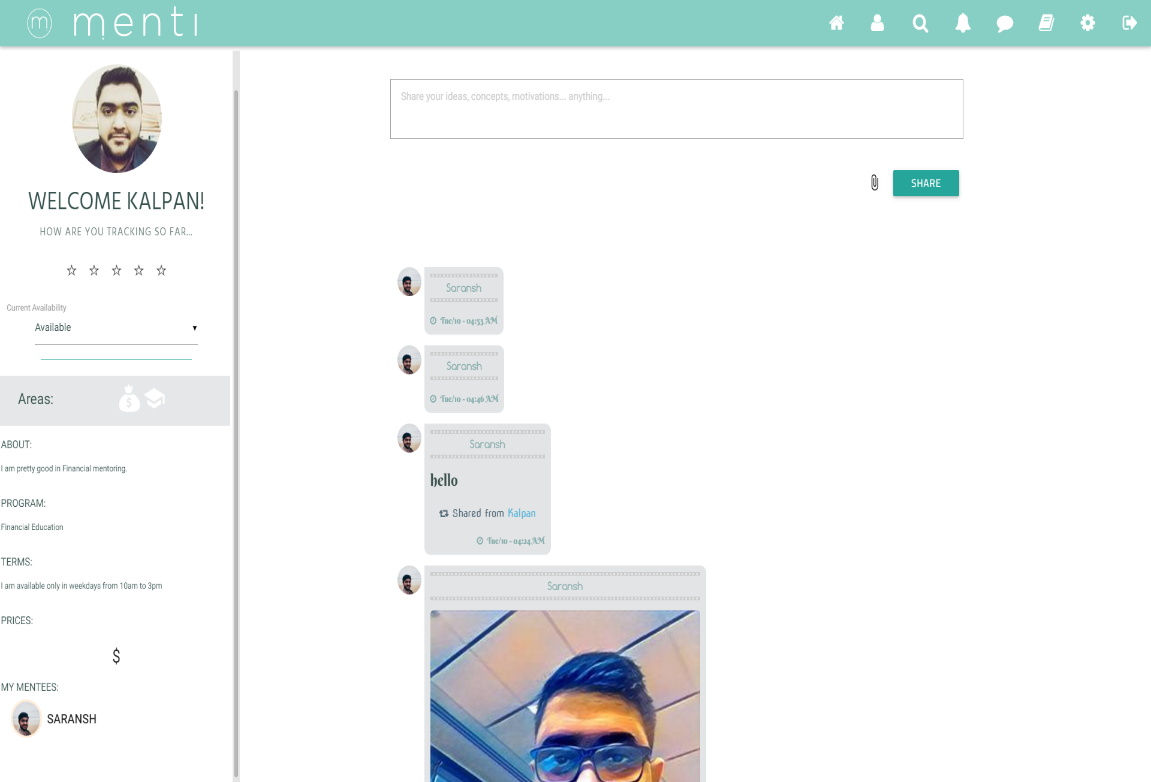
**6.6.4** Created Follow database schema.

**6.6.5** Enhanced Payment List design.

# 6.7 Sprint 7

**6.7.1** Enhanced mentor profile design. *(Figure 6.7)*

**6.7.2** Enhanced mentor profile back-end

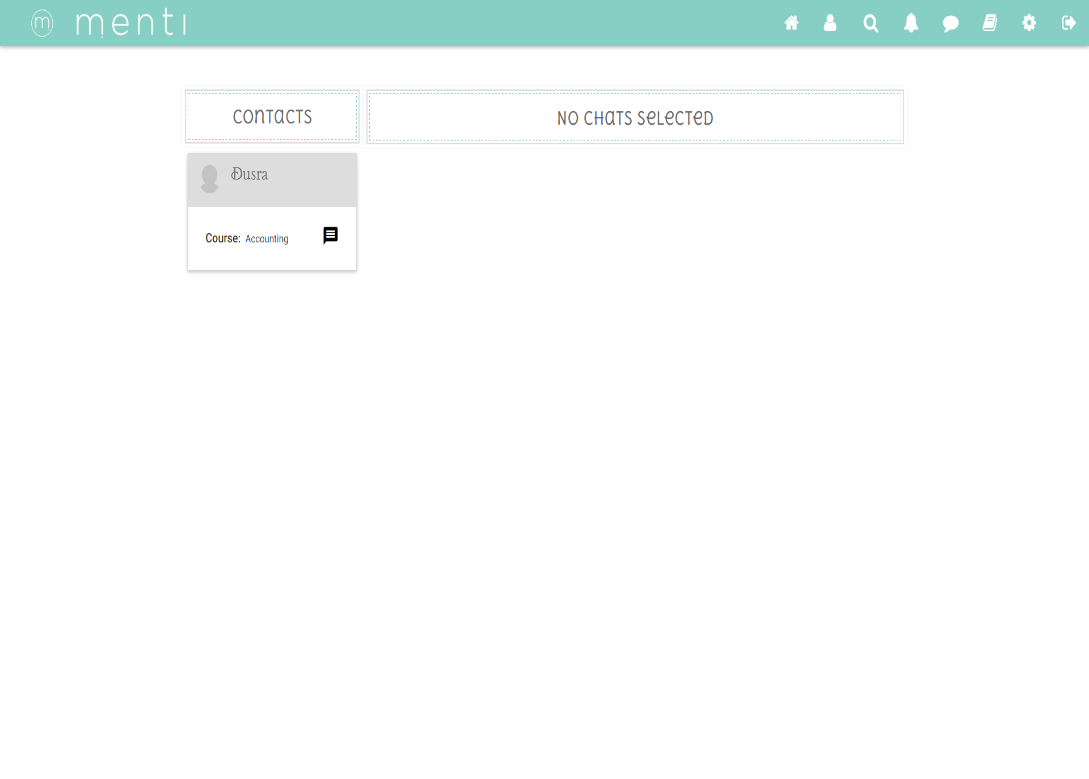


*Figure 6.7: Mentor Profile*

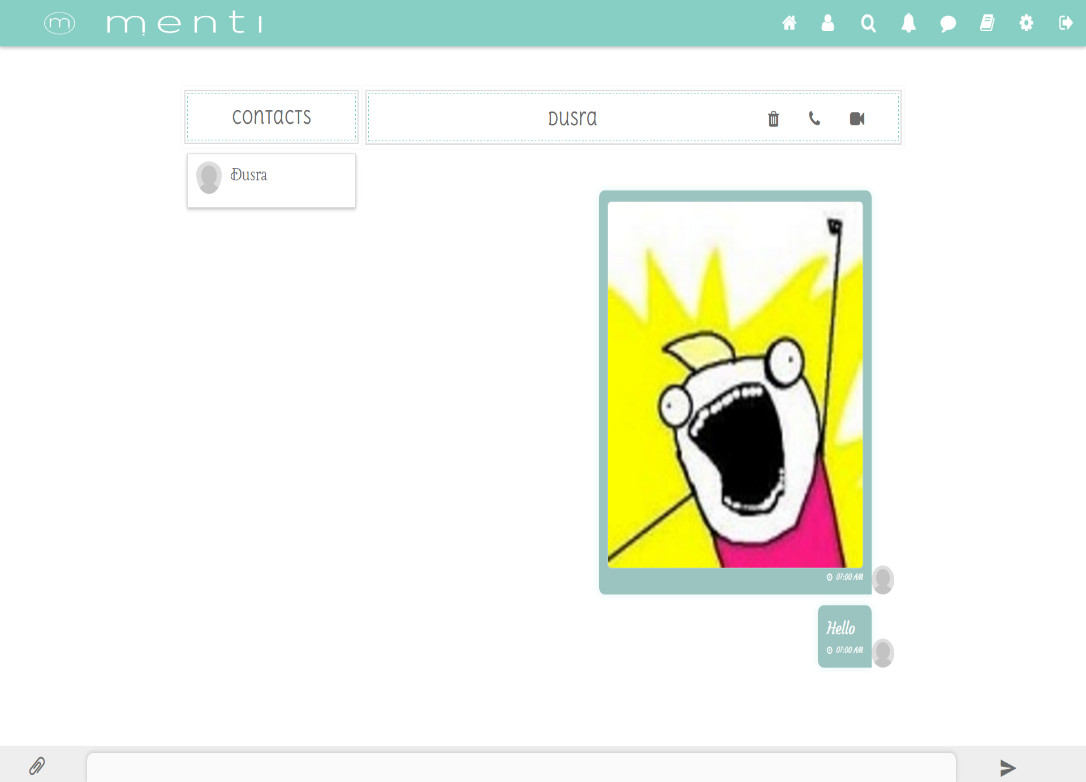
# 6.8 Sprint 8

**6.8.1** Implement design for Messaging page. *(Figure 6.8.1 & 6.8.2)*

**6.8.2** Implement back-end for Messaging page.



*Figure 6.8.1: Messaging landing page*

**

*Figure 6.8.2: Messaging main page*

# 6.9 Sprint 9

**6.9.1** Implemented search page design. *(Figure after update)*

**6.9.2** Implemented search page back-end.

# 6.10 Sprint 10

**6.10.1** Implemented resource page design. *(Figure after update)*

**6.10.2** Implemented resource page back-end.

**6.10.3** Fix major website bugs.

# Deployment Instructions

**7.1** How to clone project to server

**7.1.1** Clone the project repository

|  |
| --- |
| git clone <https://github.com/rittidate/menti-app.git> |

**7.2.** Setting up environment (OSX)

**7.2.1** Install Homebrew

|  |
| --- |
| ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)" |

**7.2.2** Install PostgreSQL

|  |
| --- |
| brew install postgres |

### 7.2.3 To change called directory

|  |
| --- |
| cp /usr/local/Cellar/postgresql/9.4.5\_2/homebrew.mxcl.postgresql.plist ~/Library/LaunchAgents/ |

**7.2.4** To reload postgresql

|  |
| --- |
| launchctl unload ~/Library/LaunchAgents/homebrew.mxcl.postgresql.plist  launchctl load ~/Library/LaunchAgents/homebrew.mxcl.postgresql.plist |

\*\*\*9.4.5\_2 is version of postgresql on your machine.

**7.2.5** Install Ruby on rails

install RVM

|  |
| --- |
| curl -sSL https://get.rvm.io | bash -s stable |

**7.2.6** Install Ruby version 2.3.1

|  |
| --- |
| rvm install 2.3.1 |

**7.2.7** Install rails version 4.2.6

|  |
| --- |
| gem install rails -v 4.2.6 --no-ri --no-rdoc |

**7.3** Install bundle gem install bundle

Bundler provides a consistent environment for Ruby projects by tracking and installing the exact gems and versions that are needed.

**7.3.1** install bundler

|  |
| --- |
| gem install bundler |

**7.3.2** go to project directory and install gems needed

|  |
| --- |
| bundle install |

**7.4** Install heroku tool

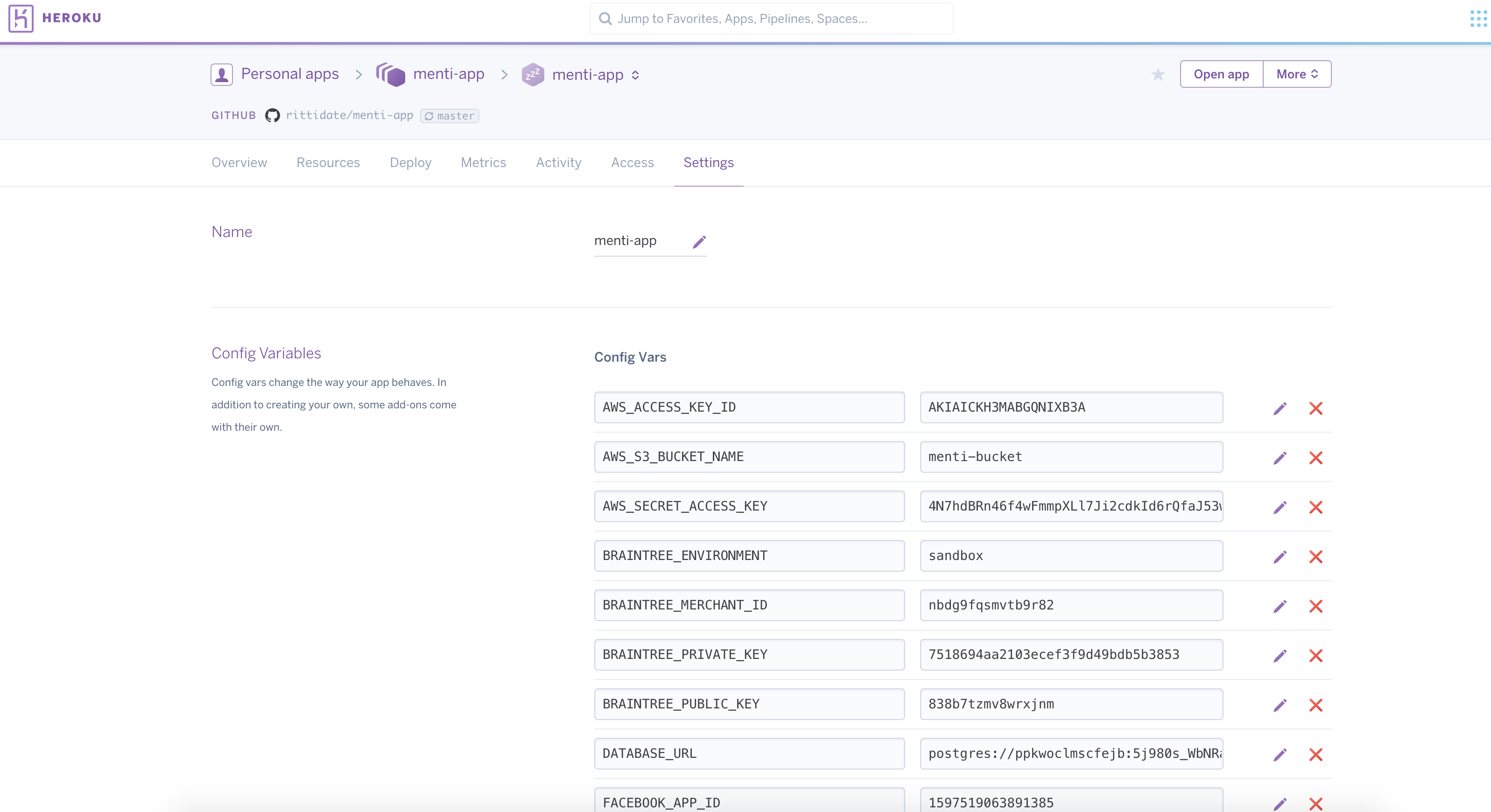
<https://devcenter.heroku.com/articles/heroku-command-line>

Once installed, you will have access to the $ heroku command from your command shell. Log in using the email address and password you used when creating your Heroku account

|  |
| --- |
| heroku login |

**7.5** Change environment in ‘/.env’ or heroku environment **7.5.1** All needed parameter put in this file.

|  |
| --- |
| RACK\_ENV=  PORT=  SECRET\_KEY\_BASE=  FACEBOOK\_APP\_ID=  FACEBOOK\_APP\_SECRET\_KEY=  BRAINTREE\_ENVIRONMENT=  BRAINTREE\_MERCHANT\_ID=  BRAINTREE\_PUBLIC\_KEY=  BRAINTREE\_PRIVATE\_KEY=  AWS\_S3\_BUCKET\_NAME=  AWS\_ACCESS\_KEY\_ID=  AWS\_SECRET\_ACCESS\_KEY= |

**7.5.2** Set up environment on heroku server as shown in the figure

*Figure 7.1 Heroku environment setup*

**7.6** Setup the project database by using

|  |
| --- |
| #initial setup database  bundle exec rake db:setup    # migrate all database schema  bundle exec rake db:migrate  # insert initial data needed  bundle exec rake db:seed |

**7.7** Run rails on machine

Rails server

**7.8** Other Deployment

For hosting, you have choices to choose the Heroku hosting or alternative hosting.

**7.8.1** Using the Heroku hosting

**7.8.1.1** Moving staging environment to production environment.

<https://devcenter.heroku.com/articles/multiple-environments>

**7.8.1.2** Changing from heroku domain to your domain

<https://devcenter.heroku.com/articles/custom-domains>

**7.8.1.3** Heroku pricing

<https://www.heroku.com/pricing>

**7.8.2** Alternative hosts which are compatible with this project.

-OpenShift

|  |
| --- |
| rails server |

-Cloud Foundry

-AWS

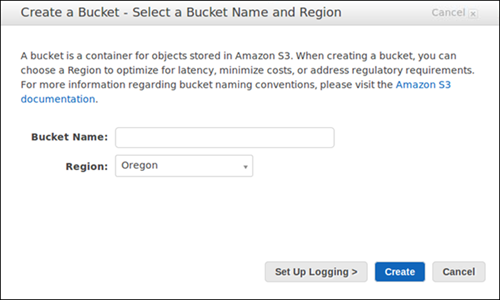
For more information, http://blog.sudobits.com/2013/02/15/heroku-alternatives-for-deploying-rails-applications/

**7.8.3** The other services are developed in this project. **7.8.3.1** We have used AWS S3 service to store files such as, images, documents and other files.

<https://aws.amazon.com/s3>

**7.8.3.3.1 Steps for Amazon s3 bucket creation and setup:**

1. Sign into the AWS Management Console and open the Amazon S3 console at <https://console.aws.amazon.com/s3>.
2. Click Create Bucket.



**3.** In the Create a Bucket dialog box, in the Bucket Name box, enter a bucket name.

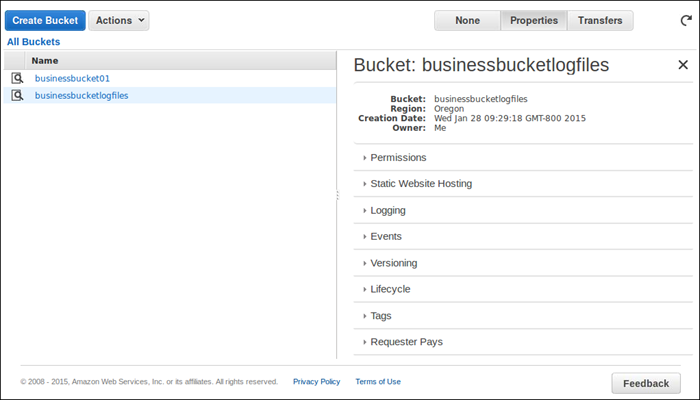
The bucket name you choose must be unique across all existing bucket names in Amazon S3. One way to help ensure uniqueness is to prefix your bucket names with the name of your organization. Bucket names must comply with certain rules. For more information, see [Bucket Restrictions and Limitations](http://docs.aws.amazon.com/AmazonS3/latest/dev/BucketRestrictions.html) in the *Amazon Simple Storage Service Developer Guide*.

Note:After you create a bucket, you cannot change its name. In addition, the bucket name is visible in the URL that points to the objects stored in the bucket. Ensure that the bucket name you choose is appropriate.

**4**. In the Region box, select a region. For this exercise, select Oregon from the drop-down list.

You can choose a region to optimize latency, minimize costs, or address regulatory requirements. Objects stored in a region never leave that region unless you explicitly transfer them to another region. For more information about regions, see [Accessing a Bucket](http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingBucket.html#access-bucket-intro) in the *Amazon Simple Storage Service Developer Guide*.

**5.** Click Create. When Amazon S3 successfully creates your bucket, the console displays your empty bucket in the Buckets panel.



**7.8.3.2** Facebook login app

**1.** Login to Facebook

Before creating a developer account, login in to your Facebook account.

**2.** Developer Account

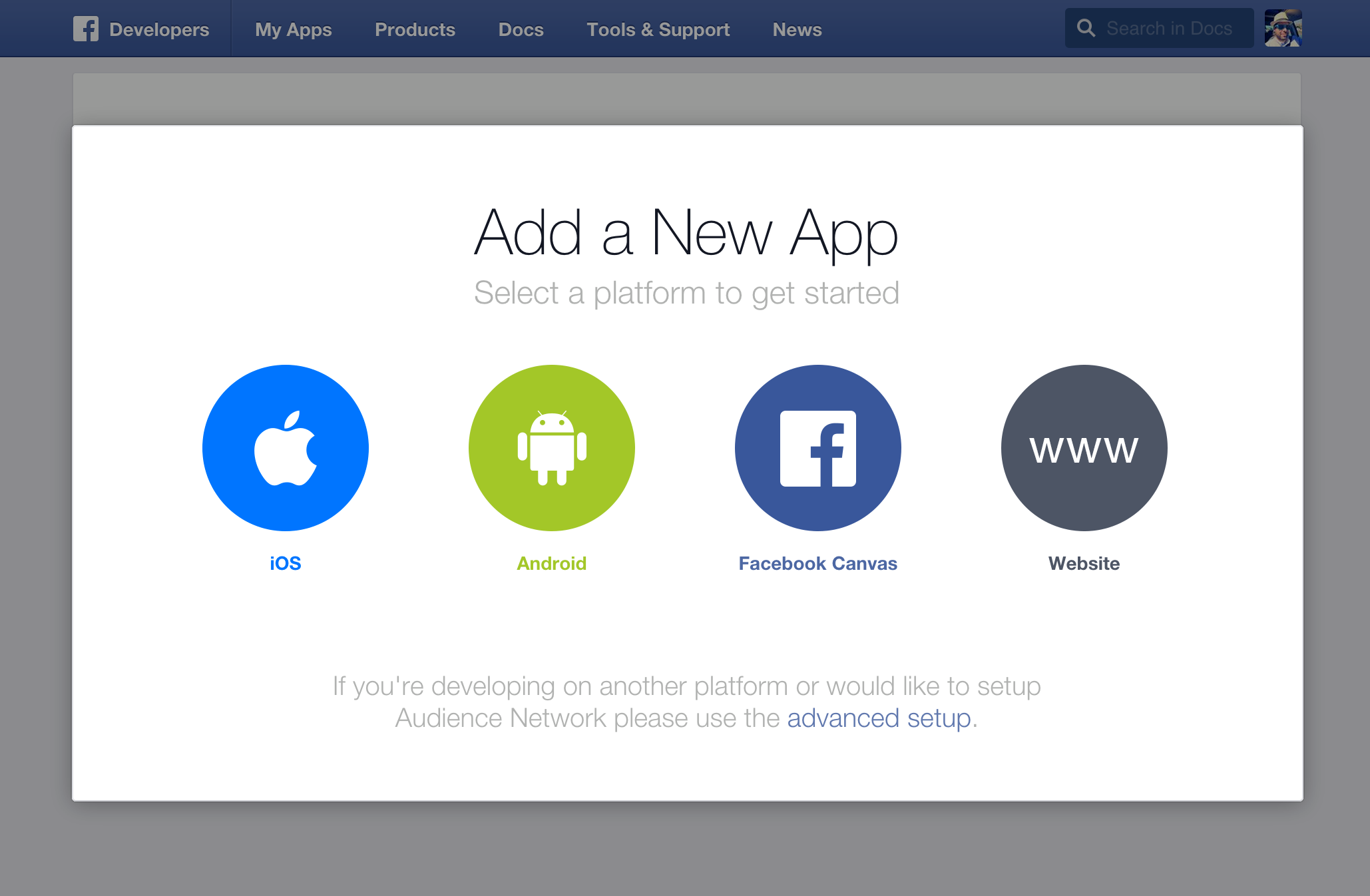
You'll need Facebook developer account to get started. If you don't have one upgrade your personal Facebook account to a Facebook Developer account now. Skip this step, if you already have a developer account.

**3.** Create new Facebook app

Choose [Apps](https://developers.facebook.com/apps/) in the header navigation and select Add a New App

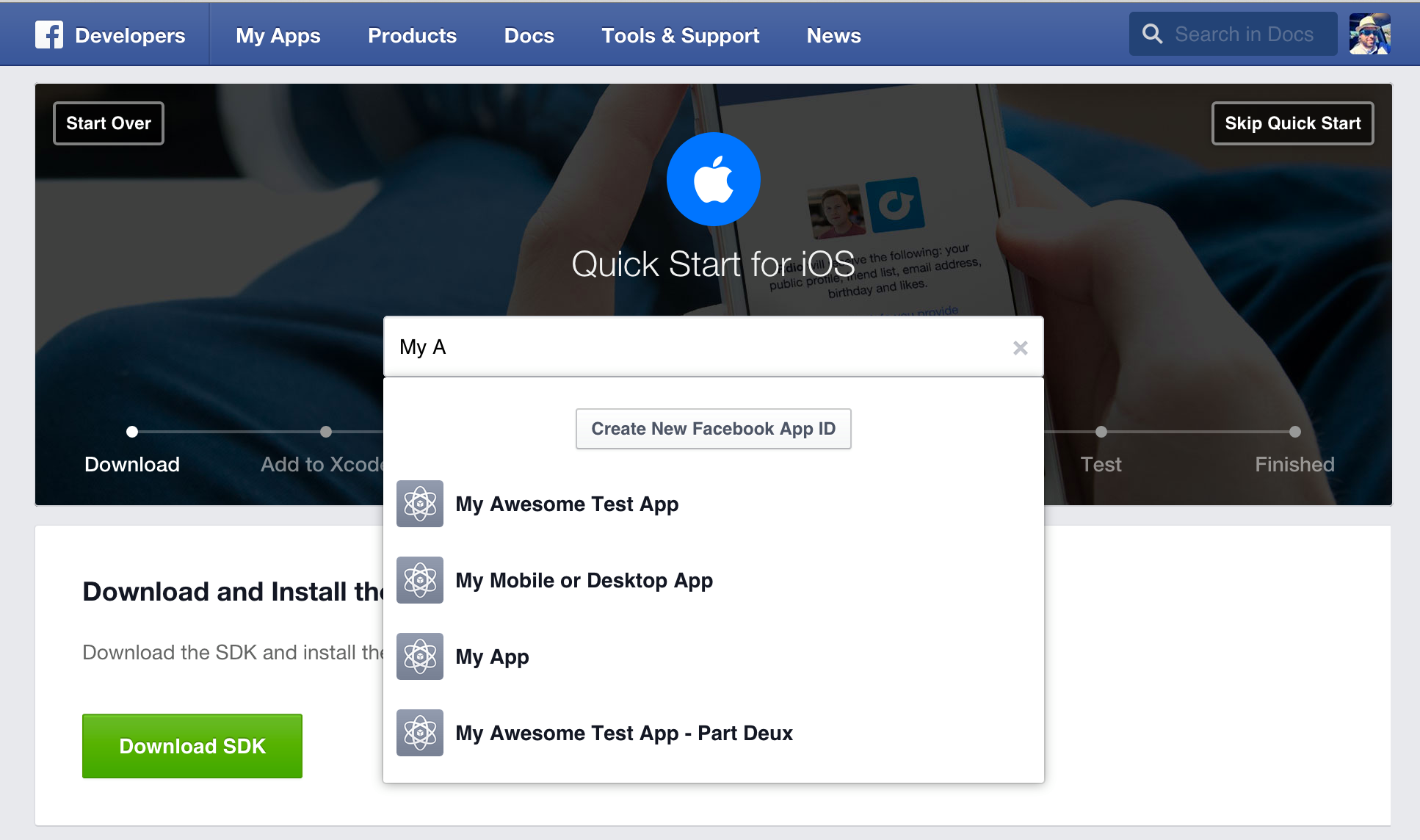
**4.** Choose Platform

Choose the platform you want to add. If your app is on multiple platforms, you'll be able to add them to your Facebook app later.



**5**. Choose a Name

Choose a name for your app and select Create New Facebook App ID. Choose a category for your app. If you're creating a test version of an existing Facebook app, select the app you're testing.



**6.** Follow "Quick Start" Steps

The quick start wizard will help you to set up your Facebook app for the selected platform. You can come back to the quick start wizard at a later time to either continue your setup or add a new platform to your app.

**7.** App ID

Your app will have a unique app ID. You will use this ID whenever you use one of our [SDKs](https://developers.facebook.com/docs/apis-and-sdks) or [Open Graph tags for sharing](https://developers.facebook.com/docs/sharing/webmasters/). You can find your app ID in your [app's dashboard](https://developers.facebook.com/apps/).

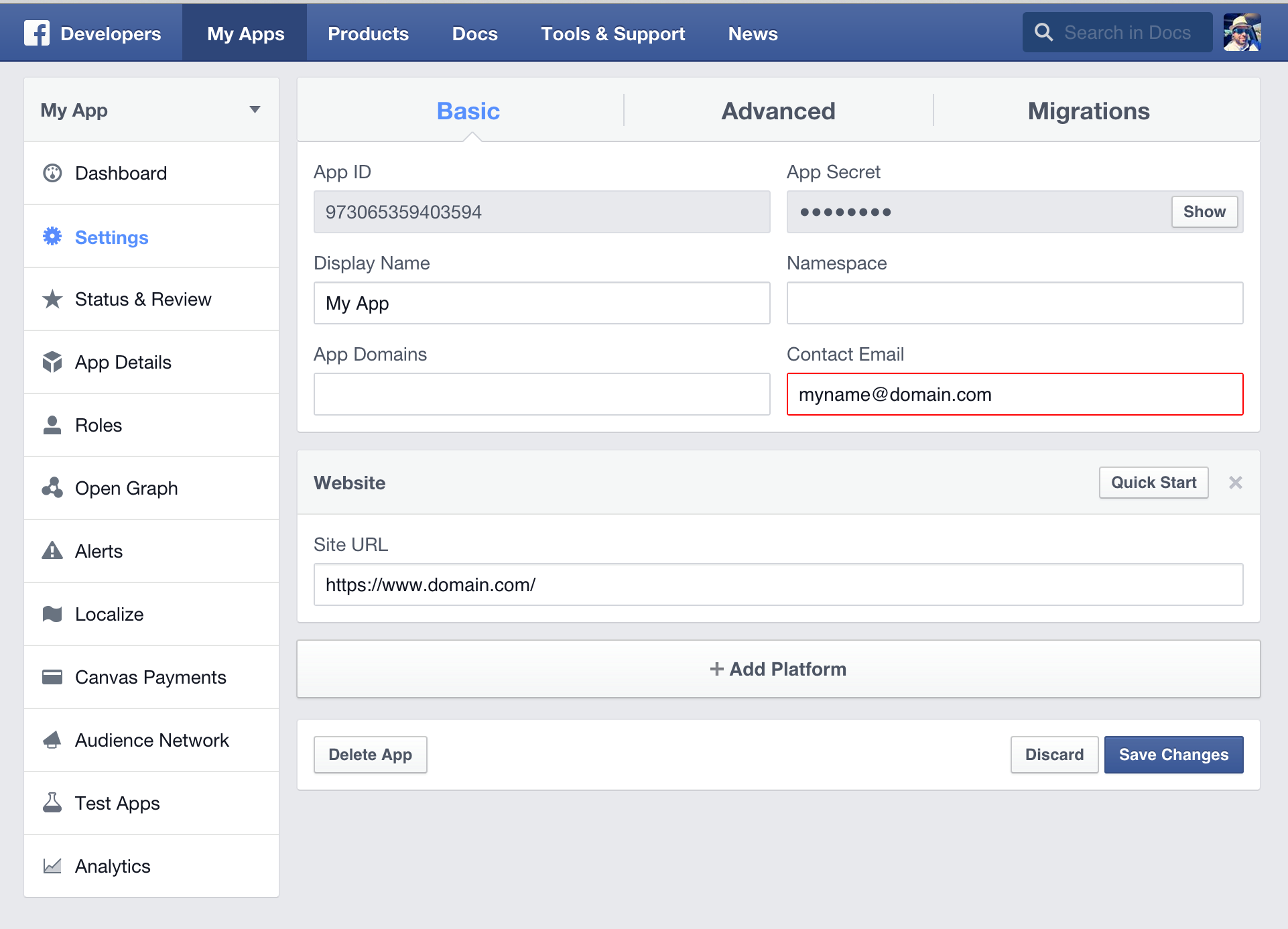
**8.** Protect your App Secret

Facebook auto-generates an app secret associated with your application. The application secret authenticates requests made by your application to Facebook servers and should be treated like a password.

Do not output your Facebook application secret in your web page markup; be careful when committing files containing this value into source control software such as a public Git or Subversion repository.

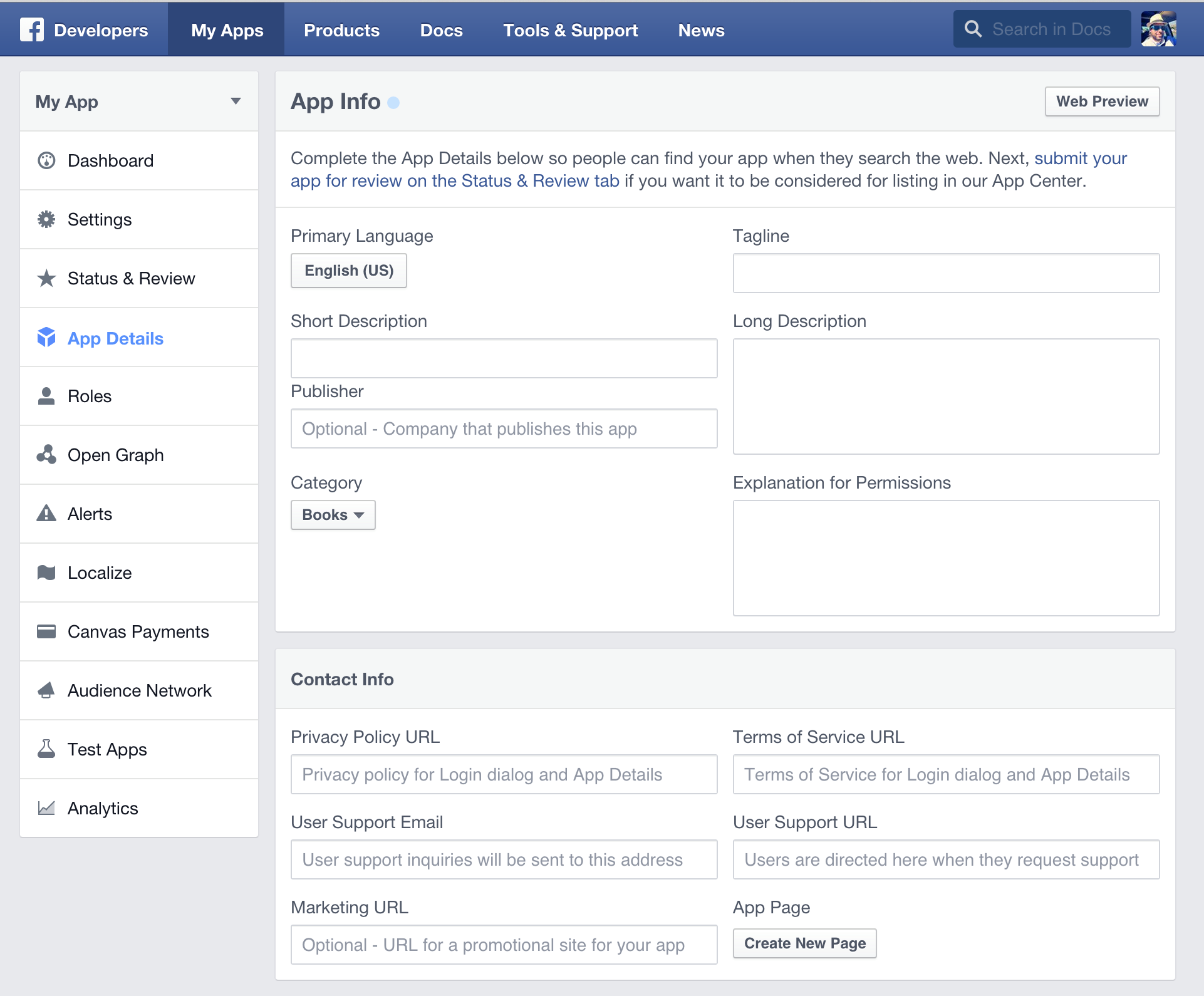
**9.** Contact Email

You need to add a contact email for each app. Go to your [app's dashboard](https://developers.facebook.com/apps/) and select the navigation item Settings in the left sidebar and provide a valid contact email.



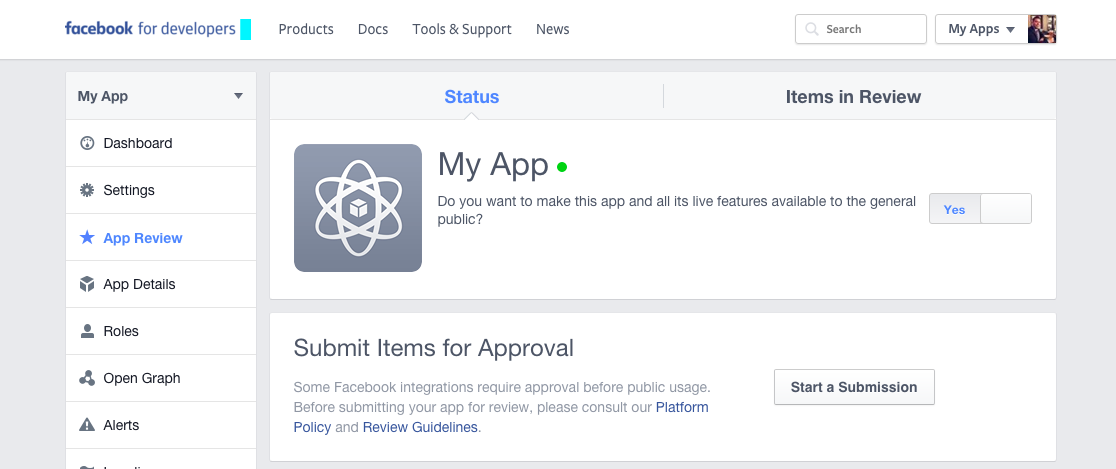
**10.** App Details

Select the App Details on the left sidebar to view and add additional information about your app. If you want your app to be listed in app center, turn on the App Center Listed Platform setting for your website or mobile site. Add icons, promotional images, and videos. Make sure to follow the [App Center Guidelines](https://developers.facebook.com/docs/games/appcenter/guidelines) to be approved for listing in App Center.



**11.** Disable Development Mode

New Facebook apps exist in development mode by default. This limits published actions and related objects such as friend tagging to Facebook Timelines owned by a developer of the application. Once your application is ready for public use you can switch from development mode to public mode to allow your app to interact with Facebook accounts not associated with your app.



**12.** Submit Your App for Review

Once your app is ready you need to submit it for review. Learn more about the review process and what's required to pass review.

**7.8.3.3** Braintree payment

<https://developers.braintreepayments.com/start/hello-> server/ruby?\_ga=1.215030231.197112222.1476859006

# Test specifications

## 

Describe the testing frameworks you used within the project (if any), present the test cases related to the requirement specification.

# testing results

Summarise the testing results.

### 

# OTHER CONSIDERATIONS

(if applicable)

# references

Please follow the referencing standards.