User Registration and Authentication

Implementer: Jyhdel Mae L. Pamonag

API name: /login, /logout, /register, /unregister

# Problems and Solutions

Problems

* The current application doesn’t support user registration and authentication.
* Users can access the application freely with how the application is currently set up.
* Unregistered users should not be given access to any features in the application.

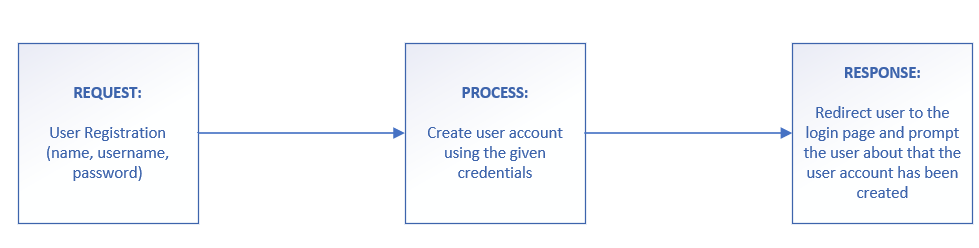
Solutions

* Implement a log in feature to the application
* Authenticate user before giving access to the application by checking if the current user is registered or not.
* Unregistered users will not be able to access any of the features of the application. If user tries to access a route, they will automatically be routed to the log-in page as default.

# Requirements (User Stories)

* As a new user, I want to register by creating a username and password so that I have my information and password so that I have my information stored in the system.
* As a registered user, I want to log-in using my credentials so that I can gain access to the application and its features.
* As a registered user, I want to be able to unregister my account so that I can delete the information stored about me when I no longer need access to.
* As a registered user, I want to be able to logout from the application so that will not have access to the application and its features.

# Design



Timeline

Description automatically generated with low confidenceGraphical user interface, application

Description automatically generatedTimeline

Description automatically generated with medium confidence

# Usage Example

* <http://webapp/login>
  + *Description:* This feature allows user to login their credentials which will be verified and authenticated in the server. Once authenticated, the user will be redirected to the home page where the user basic profile is located. The user will now be able to have access to all the other features of the application excluding the login feature and the register feature.
  + *Direction:*

Graphical user interface, application, Teams

Description automatically generatedGraphical user interface, application

Description automatically generated

* [http://webapp/register](http://webapp/login)
  + Graphical user interface, application

    Description automatically generated*Description:* This feature allows user to register and create a new user account. If successful, user will be redirected to the log-in page where the user can his/her new credentials to login to the application.
  + *After Successful Registration:*

Graphical user interface, application

Description automatically generated

* <http://webapp/unregister>
  + *Description:* This feature allows user to unregister his or her account. If successful, user will be redirected to the log-in page.
  + *Direction:*

Graphical user interface, application

Description automatically generated

* <http://webapp/logout>
  + *Description:* This feature allows user to logout from the application. The user account will lose access from application and all its features.
  + Graphical user interface, application, chat or text message

    Description automatically generated*Direction:*

# User Manuals

*Direction: make a user manual for your clients – focus on easy usage of your feature*

* /login
  + users enter their username and password in the input fields. If user is registered, the user will be redirected the profile page and will have access to all the features of this application
* /logout
  + Logout button is located on the upper right of navigation. Once clicked, user will not be able to have access on the application and its features.
* /register
  + Users enter their first name, last name, username and password and account is created. If successful, the user will be redirected to the login page where the user needs to put his new credentials to gain access to the application.
* /unregister
  + Unregister button is located on the home page under the user information section. Once clicked, the user account will be deleted. If user tried to log in with the old credentials, the system will say that it does not exist.

# Implementation Details

*Direction:*

The modules I used to implement my feature are bcrypt, passport, express-flash, and express-session. I used bcrypt to hash the password of the user before saving it in the database. I used passport as my main tool for authenticating the user credentials for logging in and accessing different features of the application which I used together with express-flash and express-session.

My code is mainly located in two areas, passport-config.json and index.js which should be located on the root directory of the team repository. I also have written home.ejs, log-in.ejs, and register.ejs, which should be located in /views directory.

As for the home.ejs, log-in.ejs, log-in.ejs, and passport-config.json, I have written all the code from it. For index.js , I have written code from lines 35.38, 45-49, 68-69, 84-166, and 378 - 393

# Test

*Direction:*

Because of the nature of how the feature was implemented, I created a separate application for testing call testregister.js and its test file named testregister.test.js. I created tests for the resgister and unregister api specifically. Unfortunately, I have been having server errors and bugs which could not let it to pass. These files are located under the /tests directory:

**Path**: ./tests/testregister.js and ./tests/testregister.test.js

As for the final application, I’ve tested every route that I have specified on my features and are working as expected shown on the usage examples.

# Deployment

GitHub URL: https://github.com/jmlpamonag/ASE-285-final-project