

**Final Project: Ashesi Social Network**

Jochebed Afua Basil

40602024

Ashesi University

CS341: Web Technologies

Cohort A

Dennis Owusu

April 25, 2023

**Links**  
GitHub: <https://github.com/Ashesi-Org/AU_socialNetwork>

Firebase deployment: <https://au-socialnetwork-api.web.app/>

**Description of the entire system**

The system is a platform designed exclusively for students of Ashesi University. The system’s development involves using REST API for the backend, Serverless for storage, Cloud Deployments for the various functionalities, and Flutter technologies for the front end. The platform will have six main functionalities: creating a profile page, editing the profile page, viewing the profile page, creating a post page, a feed page, and email notification. The API that allows requests to the server makes use of the Flask Framework and therefore Python. To store data, Firestore is used. A user can log in and out of the system. The six functionalities of the system include creating a profile page, allowing users to submit their student ID number, name, email, date of birth, year group, major, campus residence status, favorite food, and movie. Users will also be able to edit their profile information except for their name, email, and student ID. The view profile page will display the profile information of a specific user. Additionally, the system will allow users to create posts by entering their email and the text they want to share. The feed page will show all posts made by users in descending order of time, and the page will update in real-time, achieved by using Flutter Dart’s stream property. Finally, the system sends an email notification to all users when a new post is made, indicating the name of the user who created the post in the database's email trigger function.

**Description of how to test the system**

The testing of the system can be divided into two parts. Testing the API and storage using Firestore. This can be done using Postman by sending requests to the API and the responses received will validate the test. Another segment of the testing is using Flutter Dart as the front end that runs and navigates the various requests to the API which then stores in the Firestore. The first step is to either sign up by creating a profile or log in using a student id and password. Confirm that the information entered is displayed correctly on the profile page and submit. The user is then led to the feed page where they can navigate to create a post. To make a post, navigate to the create post page, enter a text, and submit. The user will be directed to the feed to see a display of posts in descending order of time on the feed page. Additionally, create new posts and ensure that they appear at the top of the feed page in real time, and an email notification is sent to all users indicating the name of the user who made the post. From the feed, the user can also navigate to view their profile and from there navigate to edit their profile where they can insert details to be changed except for the name, email, and id. Finally, to log out of the system, the user can either do that on the feed page or from their profile page. By following these steps, the system's functionality can be tested comprehensively, and any issues identified can be addressed to ensure the system is functioning optimally.

**Screenshots of Functionalities**

1. Create Profile Page

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

1. View Profile Page

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

1. Edit Profile Page

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

1. Create Post Page

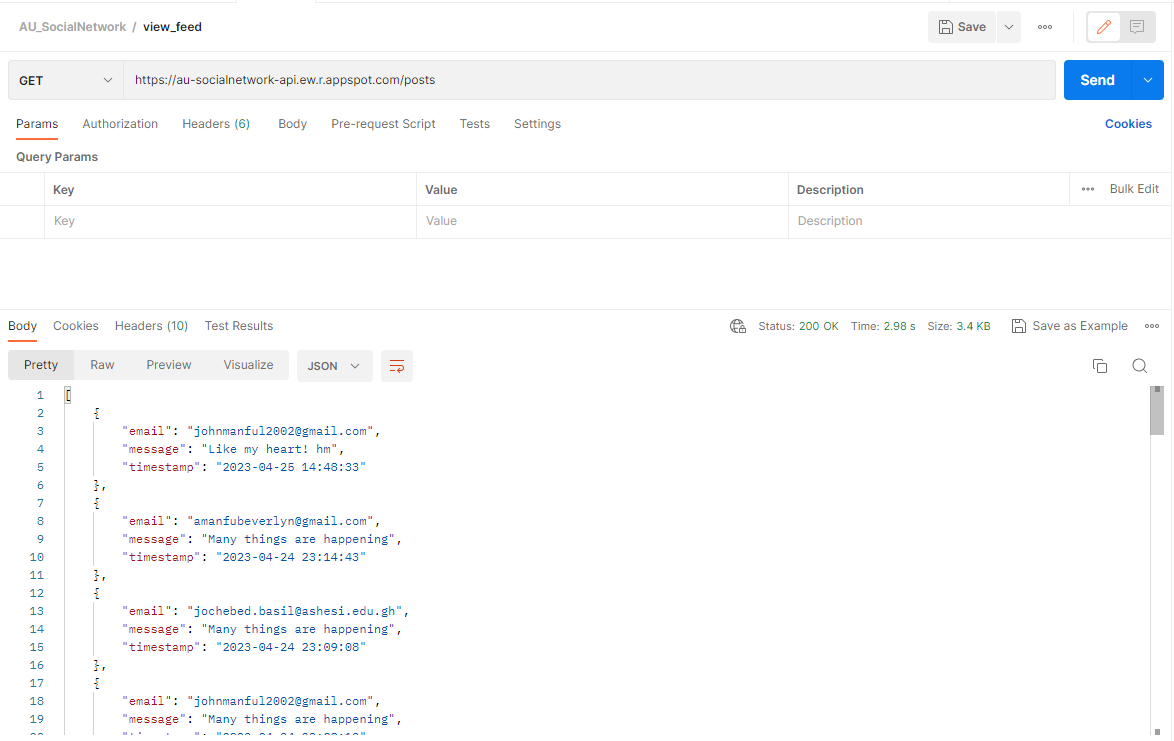
Graphical user interface, text, application, chat or text message, website

Description automatically generated

Graphical user interface, text, application

Description automatically generated

1. Feed Page



Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

1. Email Notification

Graphical user interface, text, application

Description automatically generated