**TRINITY INTERNATIONAL SS & COLLEGE**



**Project Work - Confessout**

**(COMPUTER SCIENCE-427)**

**SUBMITTED BY: SUBMITTED TO:**

**NAME: Sargam Poudel Kapil Dhungel**

**GRADE: XI 'DH2' Department of Computer Science**

**REGISTRATION NUMBER: 783270071451**

**STUDENT ID NUMBER: 22404**

**KATHMANDU, NEPAL**

**2021**

# 

**Project Work**

**Confessout**

**Submitted By: Submitted to: Sargam Poudel Parbati Bhatta Section: DH2**

# 

**ACKNOWLEDGEMENT**

I extend my deep sense of gratitude and sincere thanks to our Chairman of **Trinity International SS & College** for giving me an opportunity to be a student of this reputed institution.

I extend my respect to **Mr. Lal Krishna Malik** for his valuable support in carrying out my case study. It is a privilege to thank my computer teacher **Mrs. Parbati Bhatta** for her support and guidance for doing case study.

I also express my gratitude to all the faculty members, parents and my fellow mates who have helped me to carry out this work. Last but not the least, I thank my almighty God for his blessing shown to me during this period.

# 

# Table of Contents

[**Table of Contents**](#_atltlgf91psj) **4**

[**Objectives**](#_ewtrlvykx4jp) **5**

[**Planning**](#_t3kajgat7pgl) **6**

[**System Design**](#_kgl8zno062yr) **7**

[**System Development**](#_ccyz91shpilb) **8**

[**Screenshots**](#_ybk2tmex2x70) **15**

[**Conclusion**](#_67qu1ug3z0zn) **19**

# 

# Objectives

* To apply the knowledge from the classroom learning to a real world project
* To improve debugging skill in student by building large projects
* To improve variable naming thinking by using snake\_case, camelCase and PascalCase in necessary areas as well as using long descriptive naming to improve the meaning of variable in large project
* To emphasis problem solving as a developer
* To understand the proper importance and necessity of backend and frontend in web development and how database offers more flexibility
* To understand the CRUD(create, read, update, delete) operations
* To understand the different design principles for large scale applications
* To understand how various different components of web are interrelated with each other

# Planning

For this project I have used the following tech stack for the development of a proper CRUD based web application

* **Frontend**
  + Reactjs (state based frontend framework used to build user interfaces)
  + Tailwind Css (utility based css framework)
  + Headless Ui (tailwind css based component library for react)
* **Backend**
  + Flask (Python based backend micro framework for building rest apis)
  + Sqlite (database for mini projects)

# System Design

# System Development

How does it all work?

A rest api endpoint that supports CRUD is implemented in Flask. React app tries to fetch various data from the api and from the database. Here the frontend and backend communicate with the help of rest apis. Fetch function is used in javascript that is responsible for API requests to the server. The server after further processing of the request sends the required response for it. Here the flask application communicates with the sqlite database with the help of flask sqlalchemy module which is an ORM(object relational mapper) for python and flask.

Here all the styling is done with tailwind css and with headless ui components which allows developers to build good looking web pages without even writing a single line of css.

Brief information about react:

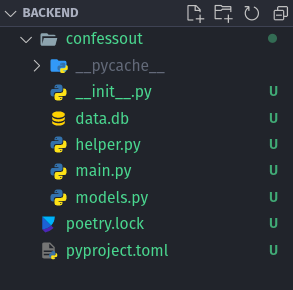
React is a javascript based library which dynamically created HTML elements are stored in it’s virtual DOM which it then appends to the real DOM with the help of different efficient functions.

It helps make HTML more powerful since it supports Javascript within the file and also supports code reusability.

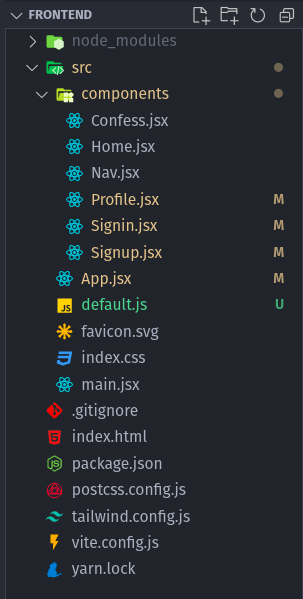
This project contains a lot of code so only the important portion has been displayed here

Folder Structure

**Backend Folder Structure**

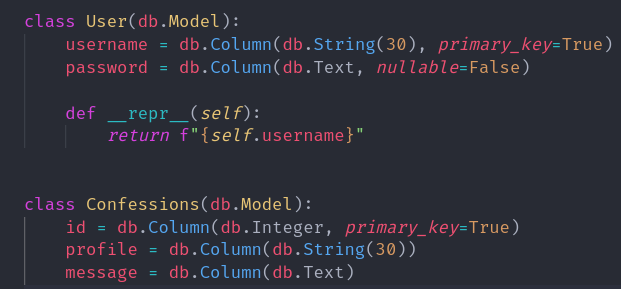


**Frontend Folder Structure**



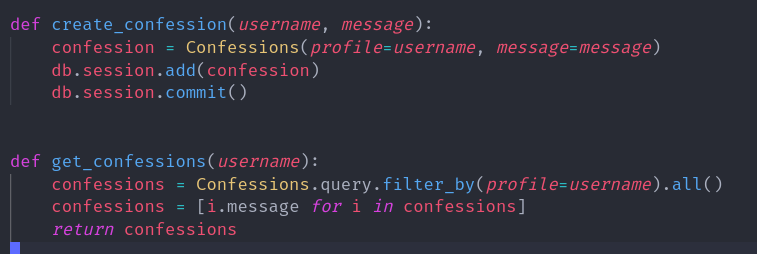
**Important files for backend**

Models.py



This file is responsible for handling the different database schemas and their creations

Helper.py



This file is responsible for different helpful functions that are required throughout the app

Main.py

This is the file responsible for all API request handling and error handling and communicating with db.

**Important files for frontend**

Index.html



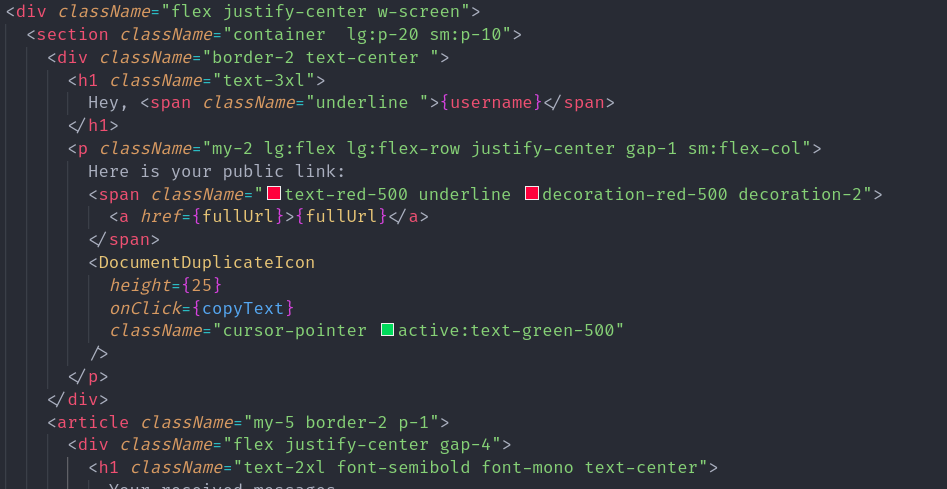
It is the only barebone html file with just an element and the rest other elements are dynamically created with javascript and rendered with a technique called CSR(client side rendering).

App.jsx



This is the first and major file in the project here all the components (html elements) are rendered.

Profile.jsx



Here all the code inside the className is for the styling with tailwind css.

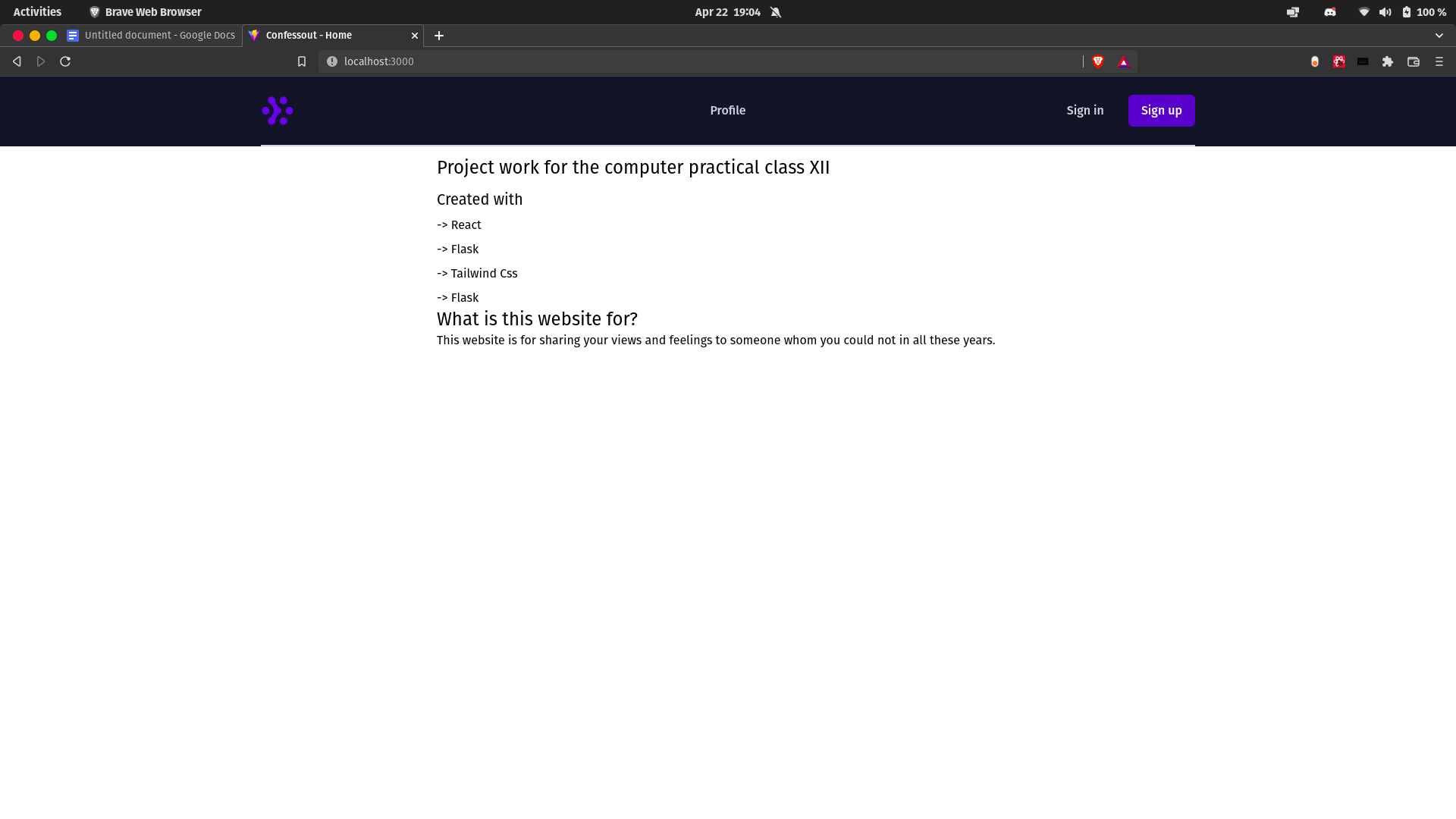
Signup.jsx

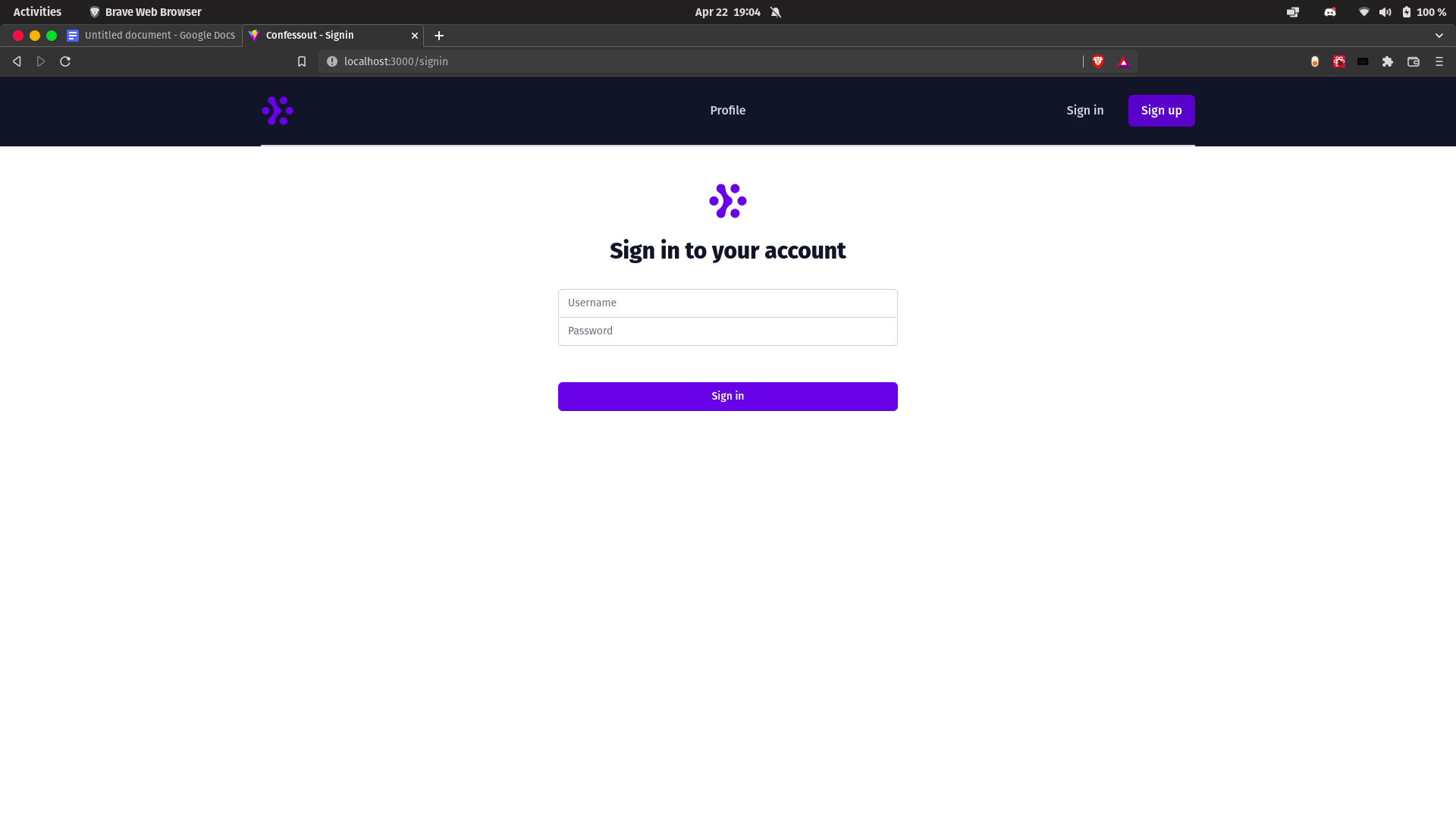


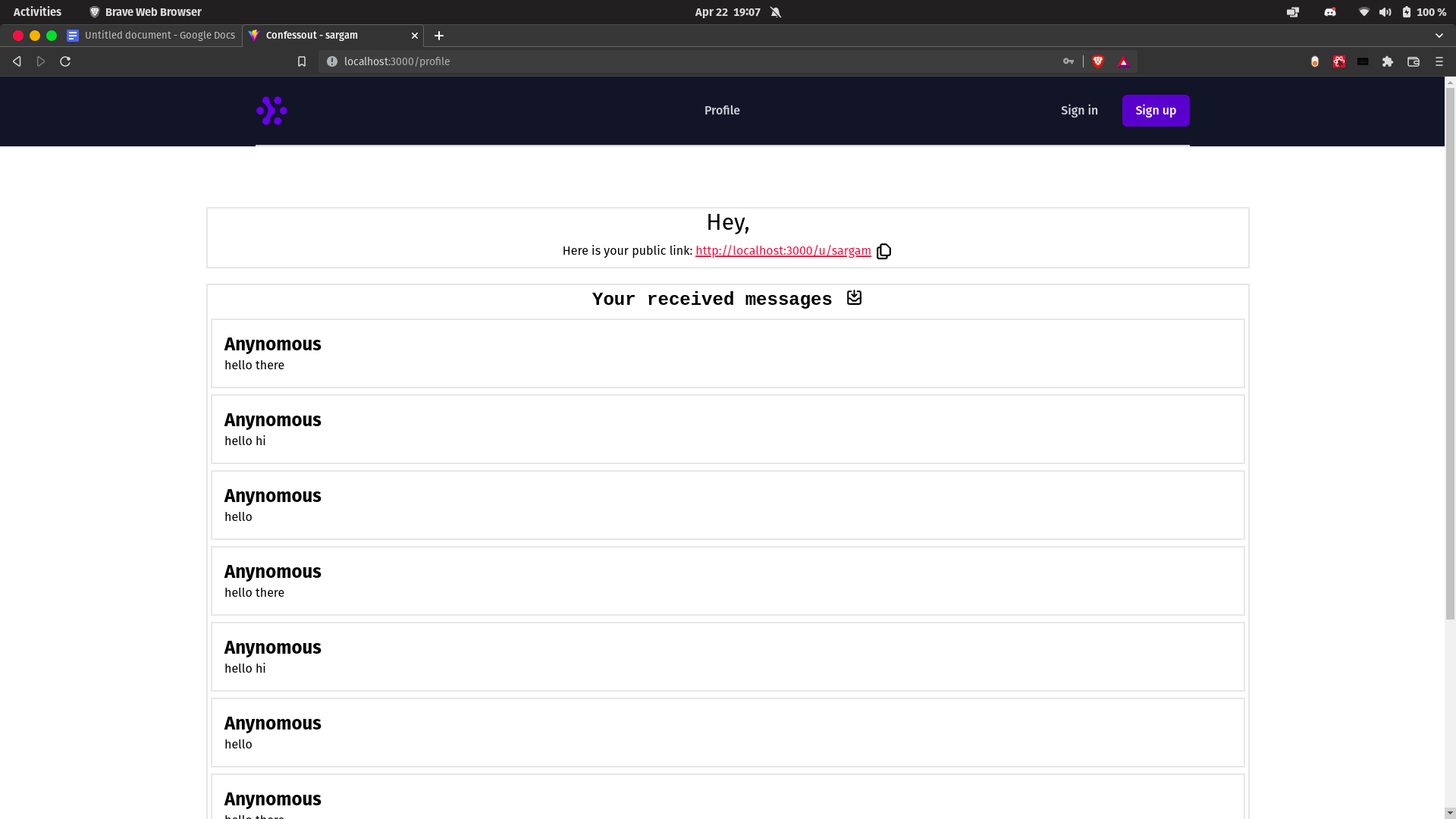
This is how all API requests happen in javascript viz with the help of fetch function. It returns a promise(can be considered as a value that will come after some time) which can be processed to get the required data.

# Screenshots

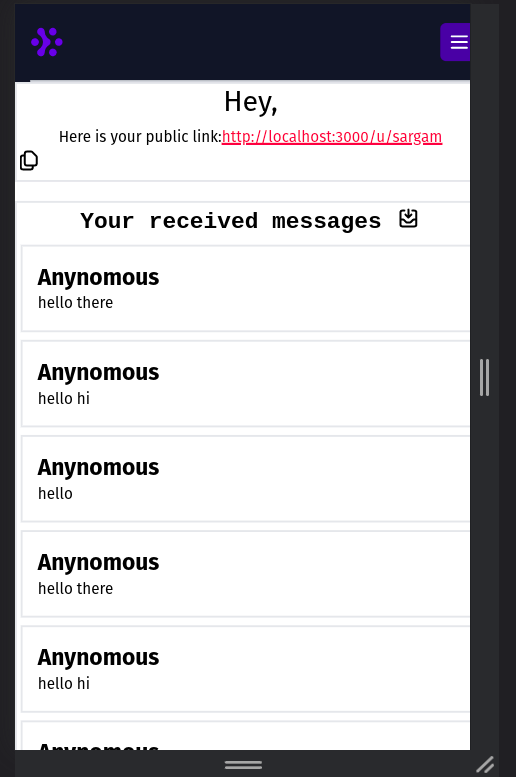
Desktop View

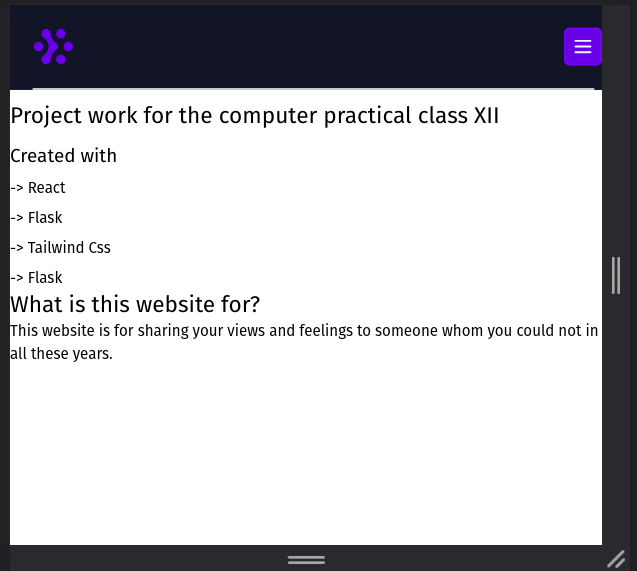


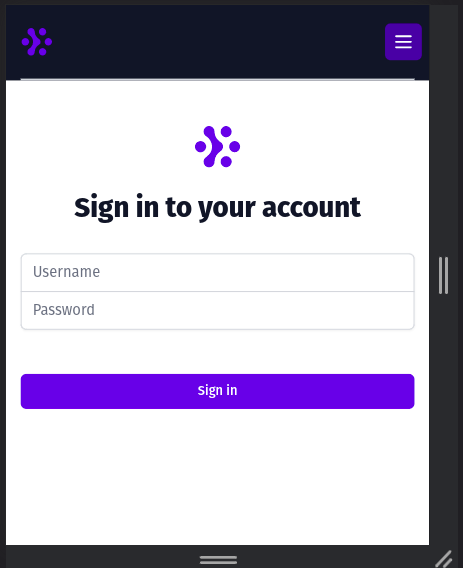




Mobile View







# Conclusion

In conclusion, this project has explained different concepts to the readers. I hope this project was helpful to whoever decides to read this project carefully and see the outputs.

After doing this project, I have gained a lot of information regarding different concepts in web development and this has also given me an opportunity to create and display beautiful and functional webpages. It certainly has increased my ability to ponder over the new problems and think logically about their solution with the help of code. This project has provided me with the basic knowledge that I can apply to create real world websites. Also, it has explained to me how to use different debugging tools like visual studio code and testing tools like postman.