UNIVERSITY OF GHANA

Department of Computer Engineering

SCHOOL OF ENGINEERING SCIENCES

COLLEGE OF BASIC AND APPLIED SCIENCES

SECOND SEMESTER 2022/2023 ACADEMIC YEAR

COURSE CODE: CPEN 208

COURSE INSTRUCTOR: MR. JOHN KORANKYE ASSIAMAH

NAME: DOE AGUDEY DANIEL

ID: 10956661

DATE: 10/06/2023

FRONT-END, BACK-END AND DATABASE CONNECTIVITY

ABSTRACT

This project focuses on the design and development of a database-driven web application for the School of Engineering using React. The goal is to create a seamless user experience by implementing essential functionalities such as student registration, login, dashboard, and student information entry. The application ensures secure access to registered students, allowing them to log in and interact with the system effectively.

INTRODUCTION

The School of Engineering requires an effective web application for managing student information. This project utilizes React for front-end development and establishes connectivity with a database. The application aims to streamline processes by providing functionalities such as student registration, secure login, intuitive dashboard access, and student information entry.

SOFTWARES

- Reactjs
- Vite
- Bootstrap
- Npm
- Visual studio code
- Figma

METHODOLOGY

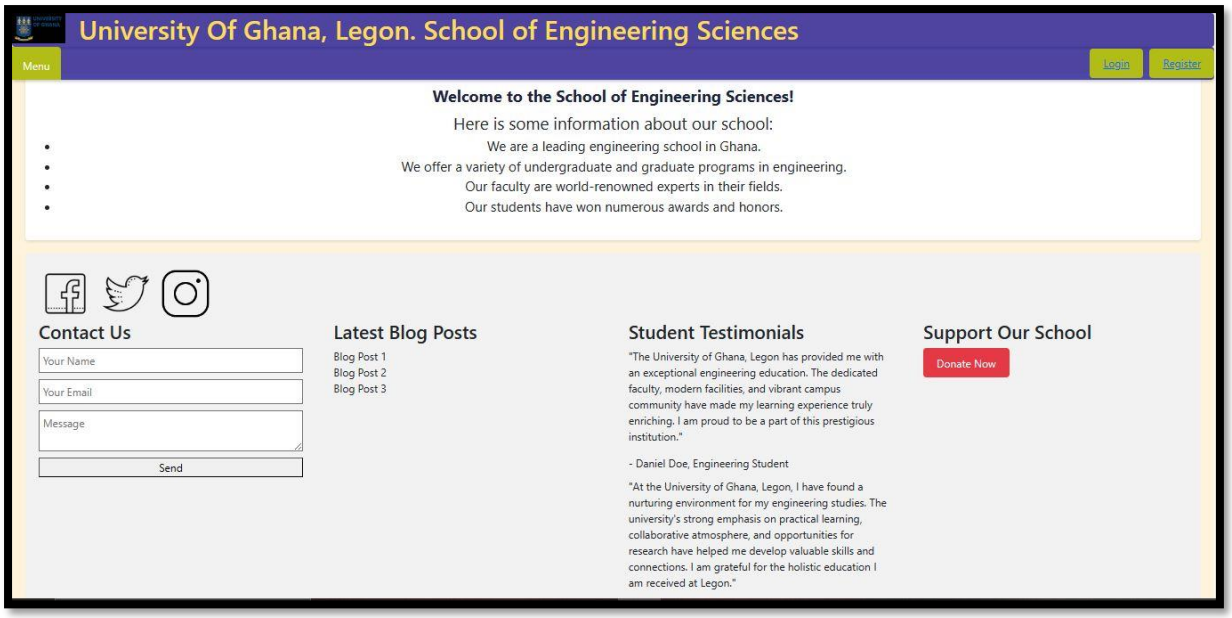
The project follows a two-step approach:

1. Sketching Web Pages: Detailed sketches are created to visualize the layout, structure, and user interface of each web page. These include student registration, login, dashboard, and student info entry.
2. Creating Web Pages: The sketched web pages are implemented using React, ensuring interactivity and responsiveness. React components are utilized to manage the application's state, handle user input, and render dynamic content.

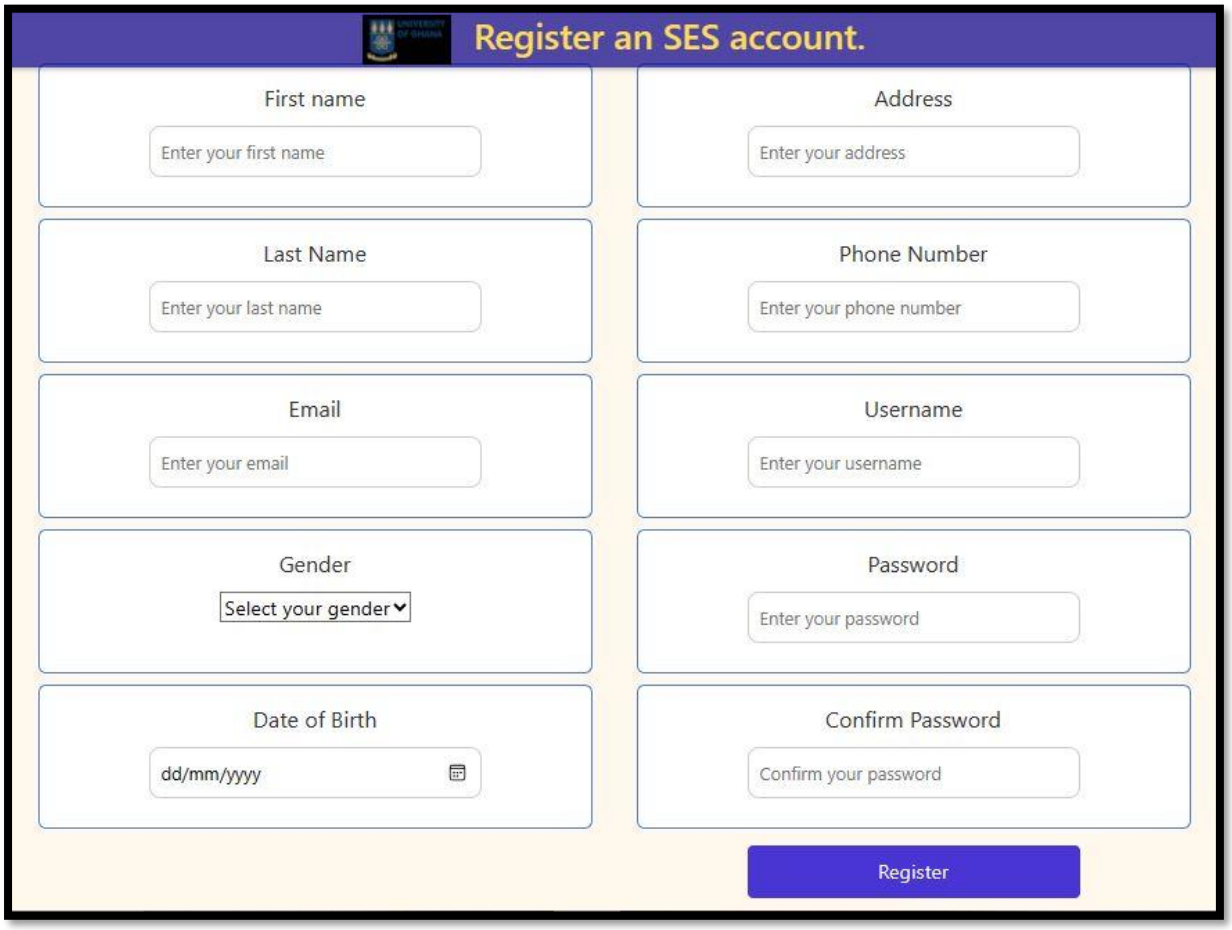
SKETCHES OF THE PAGES

A sketch of each page was first done in Figma, and later converted into a pictorial format will all the necessary designs and features.

1.Homepage



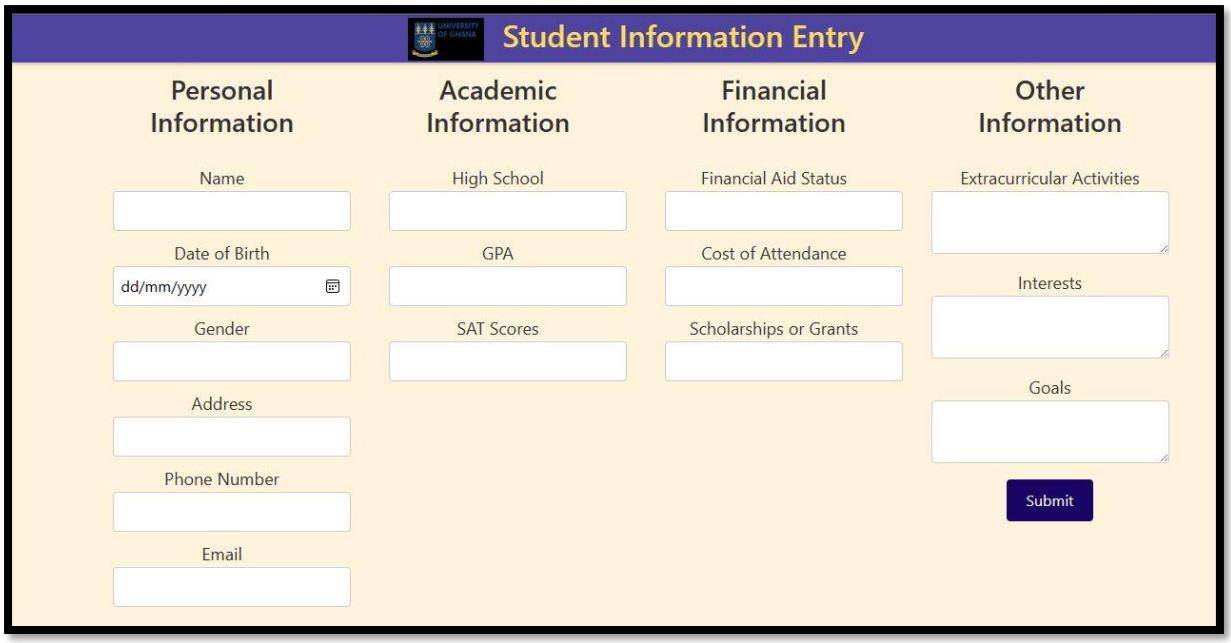
2.Registration Page



3.Log In

4.Dashboard

5.Student Information Entry



PAGE DESIGN AND IMPLEMENTATION

The implementation section provides detailed descriptions of each web page:

1. **Homepage:**
   The homepage is the first page that appears when the app is started using the command “npm run dev”.

   Functionality:
   - The component renders the homepage of the School of Engineering Sciences.
   - It includes a header with the university logo and the school's name.
   - The menu section provides links to different pages or sections.
   - Login and Register buttons allow users to navigate to the respective authentication pages.
   - The content section displays multiple widgets containing information about the school, programs offered, reasons to choose the school, and ways to get involved.
   - The footer section includes social media links, a contact form, a blog section, student testimonials, and a donate button.

2. **Student Registration Page:** This page features a user-friendly form to capture essential information from new students. Data validation is performed on both the client and server sides, ensuring data integrity. Valid information is securely stored in the database.

   Functionality:

- The component renders a registration form for users to enter their personal information, including name, email, gender, date of birth, address, phone number, username, password, and confirm password.
- It uses the **useState** hook to manage the state of the form inputs and the submitted state to track whether the form has been submitted.
- The **handleSubmit** function is called when the form is submitted, but the registration logic is currently not implemented.

3. **Login Page:** The login page provides a secure gateway for registered students. Students enter their credentials, which are then validated against the database. Upon successful validation, students gain access to the dashboard.

   Functionality:

   - The component renders a login form for users to enter their username and password.
   - It uses the useState hook to manage the state of the username and password inputs.
   - The handleSubmit function is called when the form is submitted, but the login logic is currently not implemented.

4. **Dashboard Page:** The dashboard serves as a centralized hub for students after login. It displays relevant information such as enrolled courses, upcoming events, and announcements. The responsive design ensures an optimal user experience, and students can manage their profiles, update personal information, and view academic progress.

   Functionality:

   - The component displays a header with the student's name and a welcome message.
   - It renders a set of widgets and a list of enrolled courses.
   - The widgets and enrolled courses are populated with sample data for demonstration purposes.
   - There is an authentication check to determine whether the user is authenticated or not.

5. **Student Info Entry Page**: Students can update their personal information through this page. A pre-filled form allows modifications, and the updated data is securely stored in the database.

   Functionality:

- The component renders a form to collect various details about a student, including personal information, academic information, financial information, and other information.
- It uses the **useState** hook to manage the state of each form field.
- The **handleSubmit** function is called when the form is submitted, but the code currently lacks the implementation to handle the form submission and perform any required actions.

TESTING AND DISCUSSION

The implemented web application successfully provides functionalities for student registration, login, dashboard access, and student information entry. The front-end developed using React offers a responsive and intuitive user interface, enhancing user experience. The integration with the database enables efficient data management and retrieval.

Future Enhancements

In the future, the project can be expanded to include additional features such as course enrollment, grading system integration, and communication tools for students and faculty.

To enable database-driven functionality, the web application will also connected to a database system. This involves establishing a connection to the database, designing appropriate database schemas, and implementing necessary queries to fetch and update student information.

Enhancements will also be made to improve the user interface, optimize database performance, and strengthen security measures.

CONCLUSION

The design and development of the database-driven web application for the School of Engineering using React have been accomplished successfully. The application provides a user-friendly interface for student registration, login, dashboard access, and student information entry. It streamlines the management process and ensures secure access to authorized users.