Cpen208 project1 report

Samia Soleimani

ID: 11010910

- # Introduction:

The objective of this project is to develop a relational database for the Computer Engineering Department's software that manages student information, fees payments, course enrollment, and lecture assignments. Additionally, a Next.js application will be created to provide functionalities such as login, registration, and a dashboard for users. The relational database will efficiently store and organize data related to students, fees, courses, and lectures, while the Next.js application will offer a user-friendly interface for accessing and interacting with this information.

- # Objectives:

1. Develop a relational database to manage student and academic information.
2. Create a Next.js application with user authentication and dashboard functionalities.
3. Implement a database function to calculate and display outstanding student fees.

- # Task1:

Database design and implementation:

Database Creation: A comprehensive database was developed to manage the functionalities required by the Computer Engineering Department.

Schema Development:

1) Student Schema: Contains information on a student's name, ID, and email address.
2) Lecturers Schema: Contains the names of the lecturers.

3) Teaching Assistants (TA) Schema: Preserves the teaching assistants' names.

4) Courses Schema: Includes course codes (primary keys) and titles.

5) Fee Schema: Keeps track of payments, including the total amount paid and the amount owed, and links it to the student table using student ID numbers.

6) Enrollments Schema: Tracks course enrollments by students, detailing the date, course code, and student ID. This schema utilizes foreign keys to link to the Student and Courses tables.

7) Course Assignment Schema: Contains assignment information and links lecturers to their corresponding courses.

8) TA to Courses Schema: Connects teaching assistants with the courses they assist in.

Data Population: After implementing the schemas, sample data was populated into the tables using information from our class.

Function for Outstanding Fees: A database function was created to calculate each student's outstanding fees. This function returns the results in a JSON array. The implementation code for this function is included in the database script.

Below are images showcasing the results of the implemented database.

	course_code [PK] character varying (20)	course_name character varying (100)
1	CBAS210	Academic Writing II
2	CPEN 202	Computer system design
3	CPEN 204	Data structures and Algorithms
4	CPEN 206	Linear circuits
5	CPEN 208	Software Engineering
6	CPEN 212	Data Communications
7	SENG202	Differential equations

FIG1. COURSES TABLE

	student_id [PK] integer	name character varying (100)	student_number character varying (10)	email character varying (255)
1	1	Daniel Akwetey Akunyumu-Tetteh	10975105	edward.opokuagyemang@ug.st.edu.gh
2	2	Ishaan Bhardwaj	11004272	ishaan.bhardwaj@ug.st.edu.gh
3	3	Samia Soleimani	11010910	samia.soleiman@ug.st.edu.gh
4	4	Arthur Ebenezer	11012330	ebenezer.haydenarthur@ug.st.edu.gh
5	5	Kumi Kelvin Gyabaah	11012343	kumi.kelvingyabaah@ug.st.edu.gh
6	6	Annan Chioma Praise	11014727	annan.chiomapraise@ug.st.edu.gh
7	7	Mohammed Salihu Hamisu	11014977	mohammed.salihuhamisu@ug.st.edu.gh
8	8	Daniel Agyin Manford	11015506	daniel.agyinmanford@ug.st.edu.gh
9	9	Pius Oblie	11018690	pius.oblie@ug.st.edu.gh
10	10	Iddrisu Tahiru	11021544	iddrisu.tahiru@ug.st.edu.gh
11	11	Nyavor Cyril Etornam	11023595	nyavor.cyrilmorkporkpor@ug.st.edu.gh
12	12	David Kwaku Ntow Anno	11025159	david.ntowanno@ug.st.edu.gh
13	13	Agyepong Kwesi	11038081	agyepong.kwesi_asante@ug.st.edu.gh
14	14	Asare Marvin	11049492	asare.marvin@ug.st.edu.gh
15	15	Peggy Esinam Somuah	11049523	peggy.esinamsomuah@ug.st.edu.gh
16	16	Ampomah Samuel	11053386	ampomah.samuel@ug.st.edu.gh
17	17	Andrews Kwarteng Twum	11105235	kwarteng.andrewstwum@ug.st.edu.gh
18	18	Fiavor Isaac Sedem	11112276	fiavor.isaacsedem@ug.st.edu.gh
19	19	Yakubu Tanko Mohammed-Awal	11116537	mohammed.awaltanko@ug.st.edu.gh
20	20	Eririe Jeffery	11116737	jeffrey.eririe@ug.st.edu.gh
21	21	Kafu Kwame Kemeh	11116804	kafu.kemeh@ug.st.edu.gh

FIG 2. STUDENT TABLE

	ta_name [PK] character varying (100)
1	Bamzy
2	Ben
3	Foster
4	Hakeem
5	Kevin
6	Samed

FIG 3. TA'S TABLE

	lecturer_name [PK] character varying (100)
1	Agyare Debrah
2	Dr. Godfrey Augustus Mills
3	Dr. Isaac Adjaye Aboagye
4	Dr. John Kutor
5	Dr. Margaret Ansah Richardson
6	Dr. Percy Okae
7	Mr. John Asiammah

FIG 4. LECTURER'S TABLE

	id [PK] integer	student_number character varying (20)	amount numeric (10,2)	date_paid date	total_fee numeric (10,2)	status character varying (20)
1	1	10975105	500.00	2024-01-10	1000.00	paid
2	2	11004272	300.00	2024-02-15	1000.00	paid
3	3	11010910	200.00	2024-03-20	1000.00	paid
4	4	11012330	400.00	2024-04-10	1000.00	paid
5	5	11012343	600.00	2024-05-15	1000.00	paid
6	6	11014727	450.00	2024-01-20	1000.00	paid
7	7	11014977	350.00	2024-02-25	1000.00	paid
8	8	11015506	500.00	2024-03-30	1000.00	paid
9	9	11018690	250.00	2024-04-05	1000.00	paid
10	10	11021544	300.00	2024-05-10	1000.00	paid
11	11	11023595	700.00	2024-01-15	1000.00	paid
12	12	11025159	200.00	2024-02-20	1000.00	paid
13	13	11038081	400.00	2024-03-25	1000.00	paid
14	14	11049492	600.00	2024-04-30	1000.00	paid
15	15	11049523	500.00	2024-05-05	1000.00	paid
16	16	11053386	300.00	2024-01-18	1000.00	paid
17	17	11105235	200.00	2024-02-22	1000.00	paid
18	18	11112276	400.00	2024-03-28	1000.00	paid
19	19	11116537	600.00	2024-04-02	1000.00	paid
20	20	11116737	500.00	2024-05-07	1000.00	paid
21	21	11116804	300.00	2024-01-20	1000.00	paid

FIG 5. FEE'S TABLE

	assignment_id [PK] integer	course_code character varying (20)	lecturer_name character varying (100)
1	1	CPEN 202	Agyare Debrah
2	2	CPEN 204	Dr. Margaret Ansah Richardson
3	3	CPEN 206	Dr. Godfrey Augustus Mills
4	4	CPEN 208	Mr. John Asiammah
5	5	CPEN 212	Dr. Isaac Adjaye Aboagye
6	6	SENG202	Dr. John Kutor
7	7	CBAS210	Dr. Percy Okae

FIG 6. COURSE ASSIGNMENT

	assignment_id [PK] integer	course_code character varying (20)	ta_name character varying (100)
1	1	CPEN 202	Bamzy
2	2	CPEN 204	Foster
3	3	CPEN 206	Hakeem
4	4	CPEN 208	Foster
5	5	CPEN 212	Samed
6	6	SENG202	Ben
7	7	CBAS210	Kevin

FIG 7. TA TO COURSE ASSIGNMENT

	enrollment_id [PK] integer	student_number character varying (20)	course_code character varying (20)	enrollment_date date
1	1	10975105	CPEN 202	2024-01-10
2	2	11004272	CPEN 202	2024-01-10
3	3	11010910	CPEN 202	2024-01-10
4	4	11053386	CPEN 202	2024-01-10
5	5	11105235	CPEN 202	2024-01-10
6	6	11208328	CPEN 202	2024-01-10
7	7	11209640	CPEN 202	2024-01-10
8	8	11275876	CPEN 202	2024-01-10
9	9	11285635	CPEN 202	2024-01-10
10	10	11348310	CPEN 202	2024-01-10
11	11	11353826	CPEN 202	2024-01-10
12	12	11012330	CPEN 204	2024-02-15
13	13	11012343	CPEN 204	2024-02-15
14	14	11112276	CPEN 204	2024-02-15
15	15	11116537	CPEN 204	2024-02-15
16	16	11213307	CPEN 204	2024-02-15
17	17	11218951	CPEN 204	2024-02-15
18	18	11292620	CPEN 204	2024-02-15
19	19	11293871	CPEN 204	2024-02-15
20	20	11356825	CPEN 204	2024-02-15
21	21	11014727	CPEN 206	2024-03-20

FIG 8. ENROLLMENT TABLE

- # TASK 2:

Next.js Application:

After completing Task 1, Next.js 14 was used to implement Task 2 as follows:

1) **Setup and Installation:** In VS Code, the "Project" folder was opened.
   The Next.js project was created in the VS Code terminal by using the command npx create-next-app@latest my-project --typescript --eslint, where my-project is the project name. This was done after installing the npm library.

2) **Project Initialization:** After creating the project folder, cd my-project was used in the terminal to enter the project directory.
   npm run dev was used to start the Next.js project.

3) **UI Setup:** The project was set up by running the shadcn-ui init command in the terminal.

4) **Directory and Component Creation:** Two directories, components and auth, were created under the src directory.
   The command to import the button framework from shadcn-ui was used to import an ui folder into components.

5) **Component Development:** Two components, Navbar and SideNavbar, were made with the button feature.
   Additionally, a CSS file was made in order to add buttons to SideNavbar.

6) **Form Implementation:** Files for the sign-in and sign-up features were placed in a different folder called form that was made under components.
   The libraries required for constructing the sign-in and sign-up frameworks were imported and installed from shadcn-ui. After being uploaded to the ui folder, the form, input, and label files were used as needed.

7) **Page Structure:** Two new folders, auth and dashboard, were created under the app directory in the src folder. Sign-in and Sign-up folders with their own page files were established inside the auth folder. All of the project's containers and frameworks were put into use here.
   The signup and sign-in boxes were centered by editing the layout file located in the auth folder.

8) **Dashboard Development:** An admin folder was made under the dashboard folder once the sign-up and sign-in pages were linked.
   The dashboard functionality was created in the admin folder using parts like the SideNavbar.
   In order to modify the SideNavbar on the dashboard page, a CSS file was also made.

Below are images showcasing the results of the implemented Next.js project.

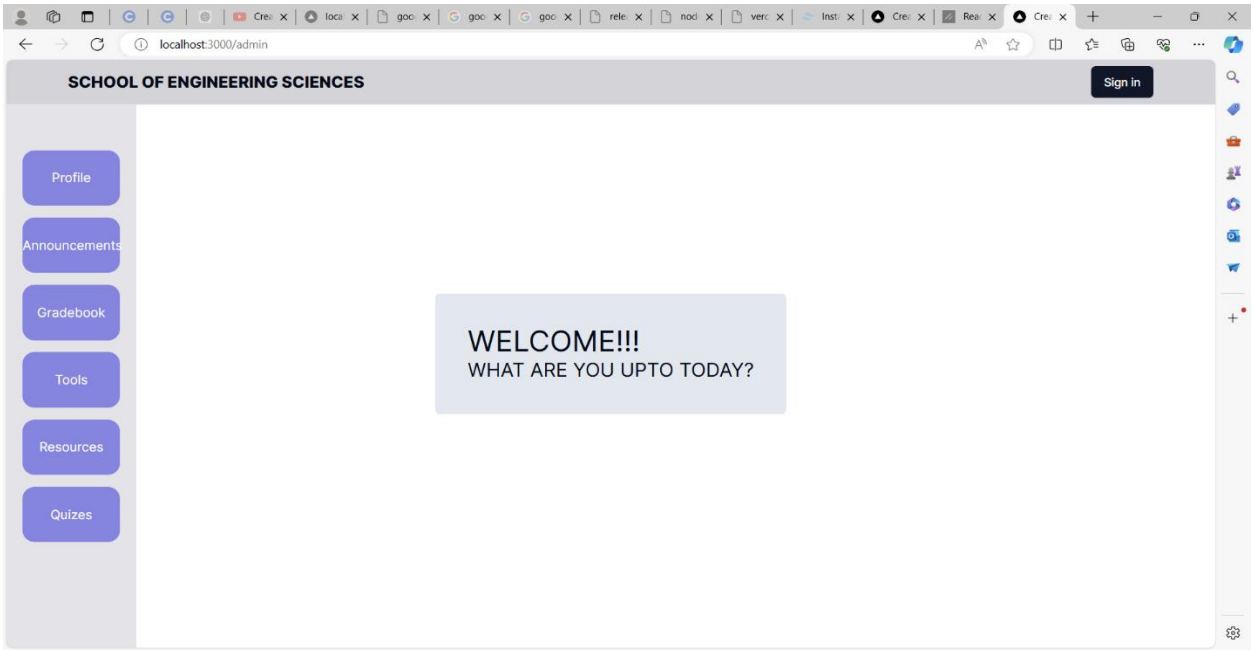

FIG 9. DASHBOARD PAGE

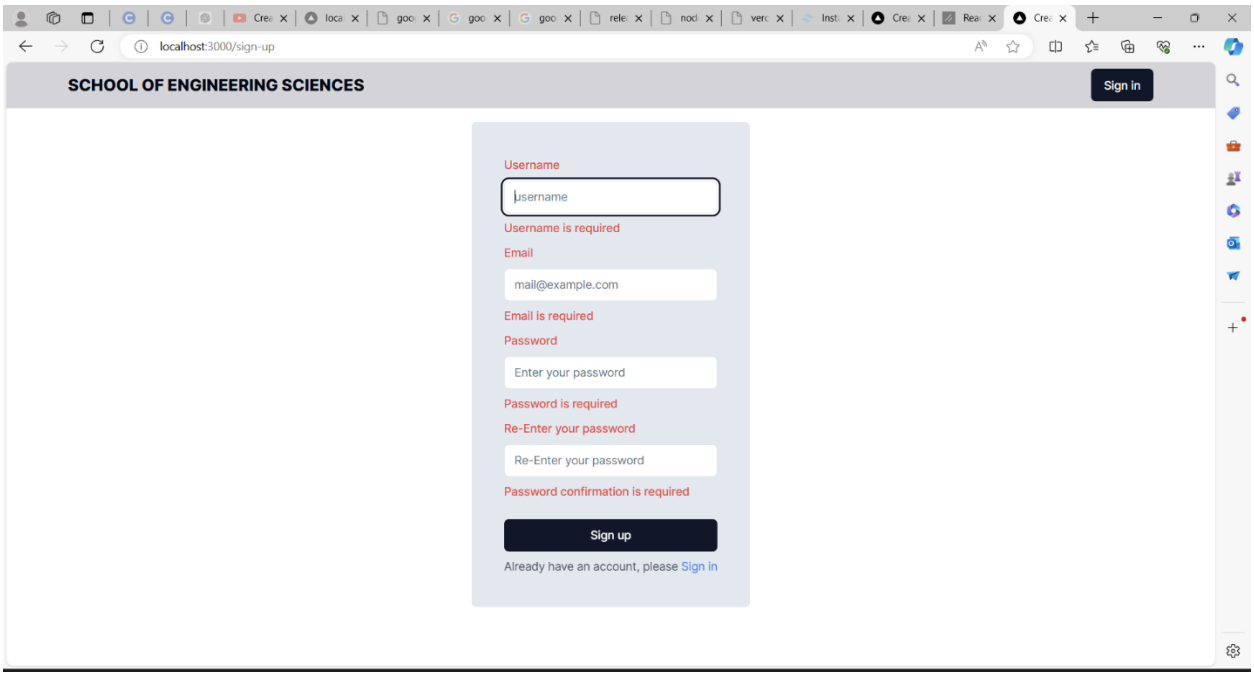

FIG 10. SIGN UP PAGE

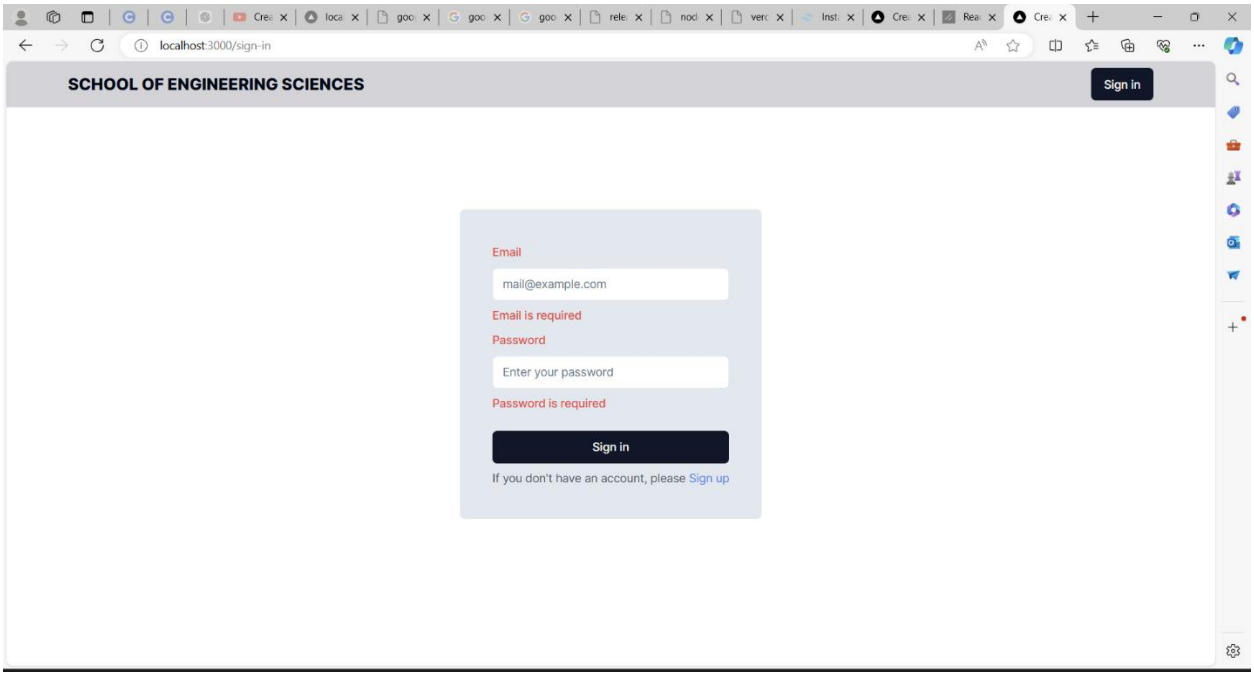

FIG 11. SIGN IN PAGE

- **Conclusion:**

In this project, a comprehensive relational database was developed to manage various academic functionalities. To provide an easy-to-use user interface for working with the database, a matching Next.js application was developed. Despite some challenges in setting up the environment and ensuring seamless integration between the database and the application, the project successfully met its objectives.