

# **CS 432 - DATABASES**

## **ASSIGNMENT - 3**



## **TEAM: OCTACORE**

**Assignment 3:** Implementing a Web App using MySQL

G1	G2
Sujith - 21110100	Jethru - 21110089
Uday - 21110084	Himasagar - 21110158
Keerthi - 21110176	Sudharshan - 21110218
Ajith - 21110045	Rudreshwar - 21110172
	Vinay - 21110125

### **3.1 Responsibility of G1:**

We have successfully designed the application's front end using HTML, CSS, and JavaScript to create a visually appealing user interface. The design follows modern standards and user experience principles to ensure users' ease of use and clarity.

Additionally, We have integrated Flask into the application to provide dynamic content generation, handle form submissions, and manage user sessions. Flask enables efficient communication between the front and back end, allowing for seamless interaction and data processing.

Overall, the combination of front-end design with HTML/CSS/JSL and Flask and MySQL database integration for the backend creates a robust and user-friendly web application that meets the specified requirements.

### **3.2 Responsibility of G2:**

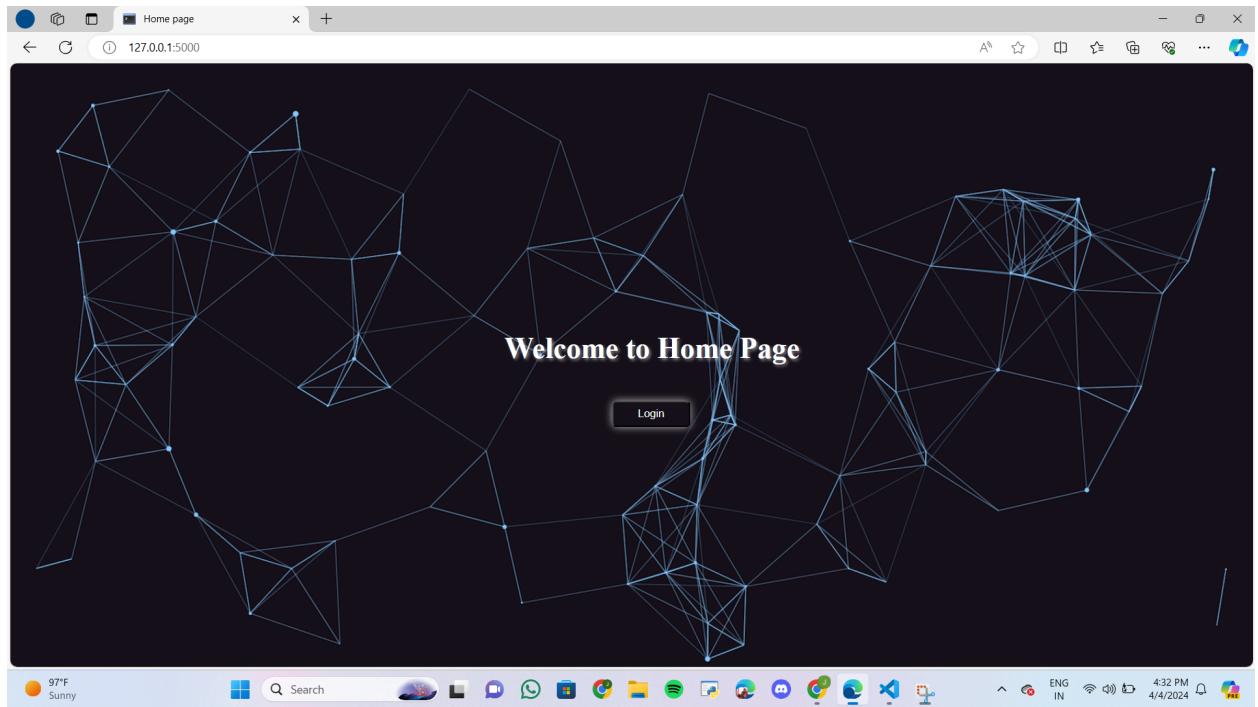
After integrating MySQL into WebApp, we ensured that the backend functions correctly by managing the database interactions, ensuring data integrity, and handling queries efficiently. This involves:

- Establishing a connection to the MySQL database from the Flask application.
- Implementing CRUD (Create, Read, Update, Delete) operations to interact with the database.
- Creating and maintaining database schemas, tables, and relationships.
- Executing SQL queries to retrieve, update, and delete data based on user input.
- Optimizing database queries and performance to ensure scalability and responsiveness of the WebApp.
- Testing database functionality to ensure the reliability and correctness of data operations.

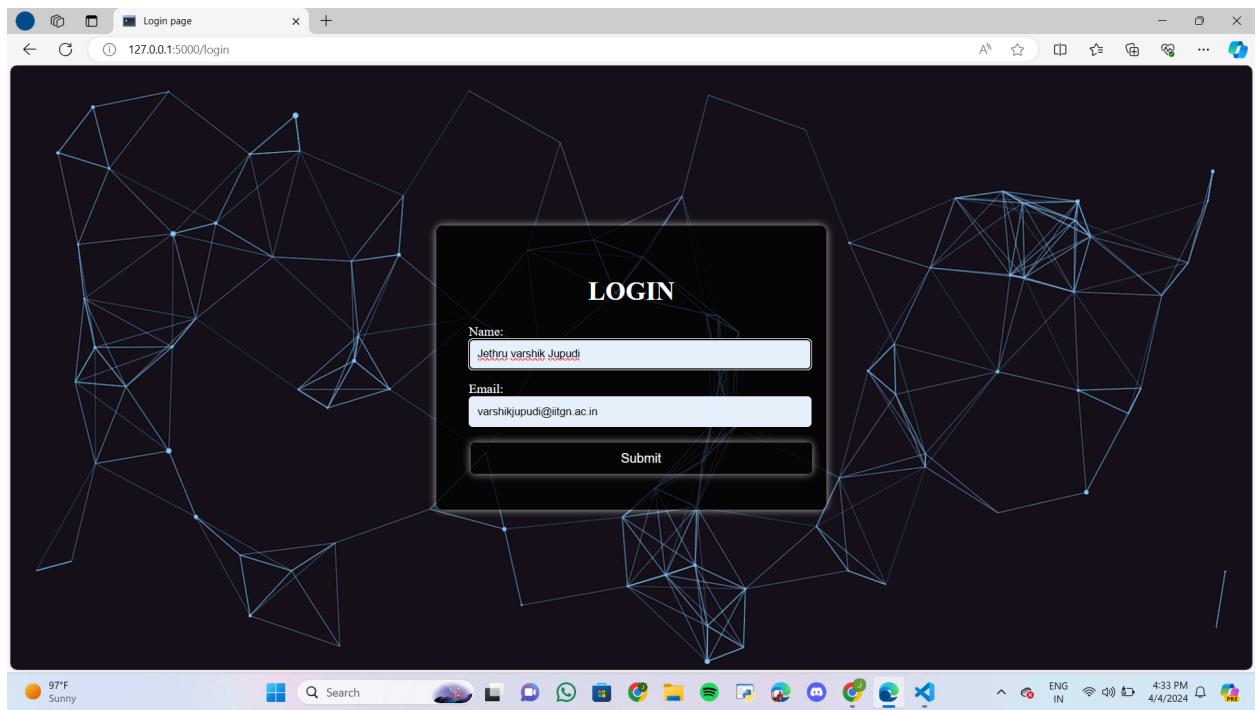
Overall, G2 played a crucial role in integrating MySQL into WebApp and ensuring the proper functioning of the backend to manage data effectively and support the application's overall functionality.

### **3.3 Responsibility of G1 & G2:**

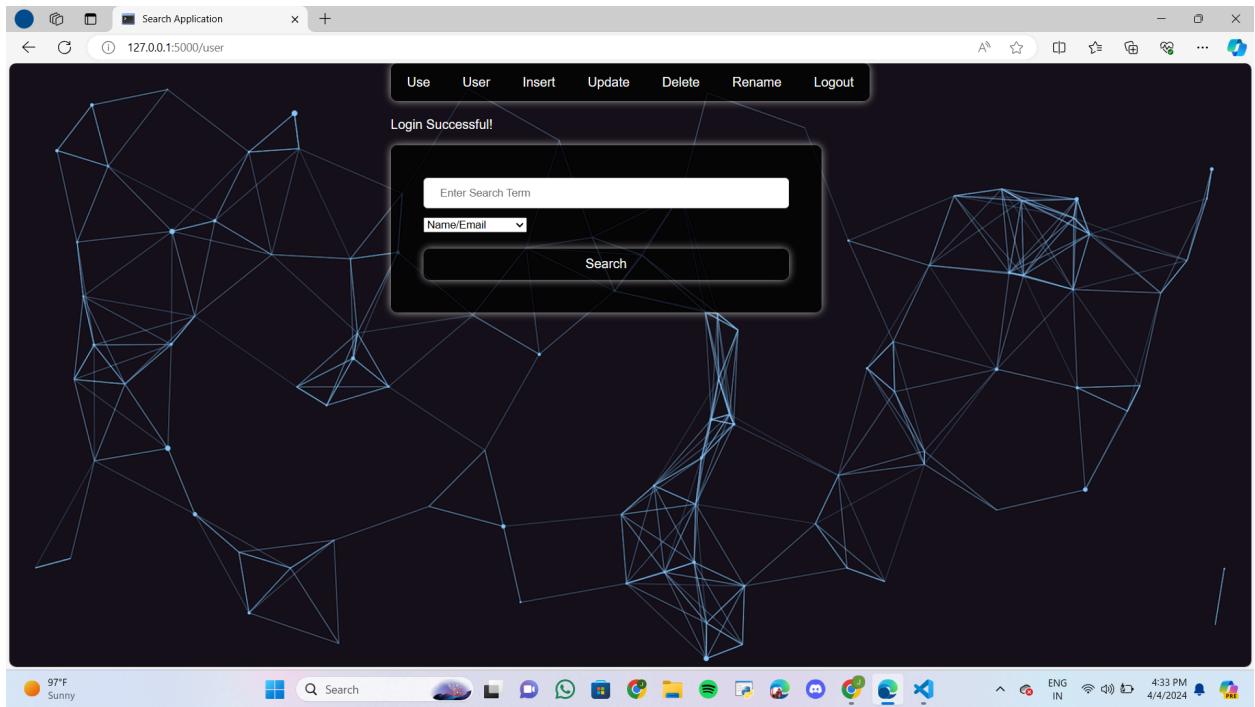
First, we will land on the homepage and configure the MySQL database correctly. Here is what it looks like



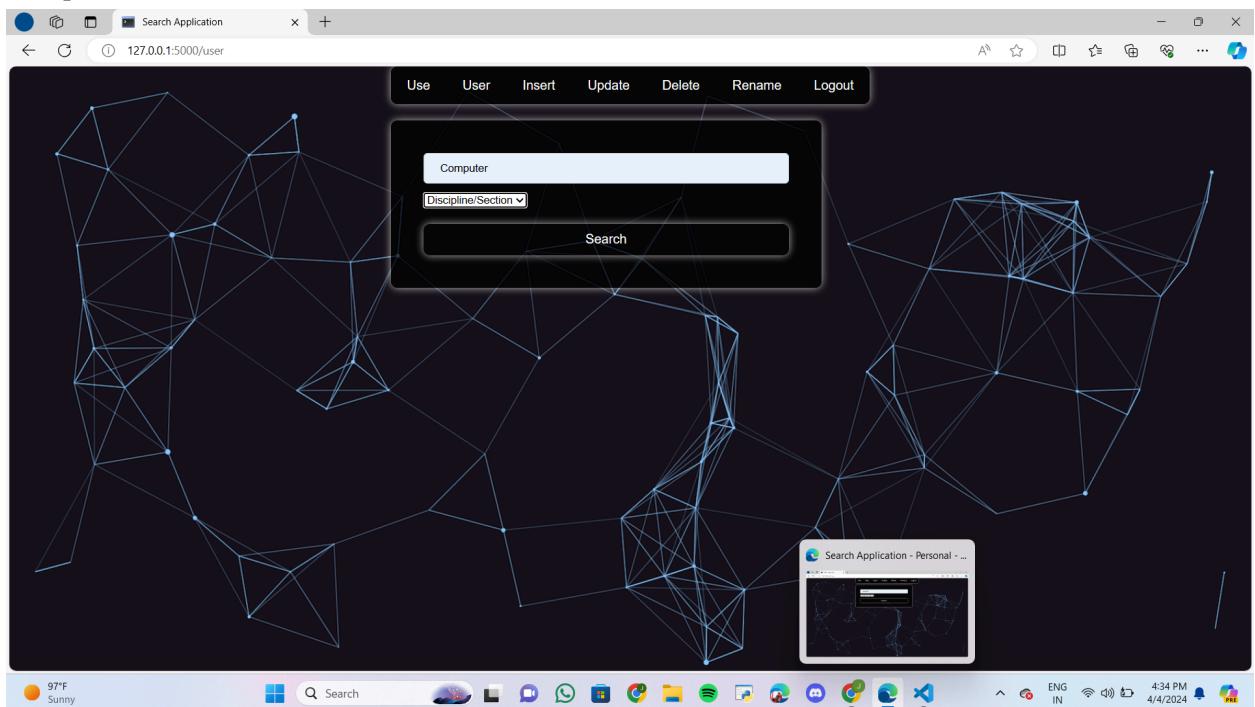
Next, we will log in as a regular user who is not an Administrator.



Our phone directory/app is only accessible to the IITGN community. So, we should always log in using the @iitgn.ac.in email address. The below image shows how it will be after login. It will be redirected to the User page or index1.html. The earlier home page is index.html



In this user page, we can search for a person's complete information using search via Name/Email or Discipline/Section column:



As we search for "Computer" in the Discipline/Section column, we will get all records that have Computer in the Discipline/Section column regardless of whether the person is a Teaching staff or non-teaching staff member, Student Facility or Block. Below is the output:

The screenshot shows a web browser window titled "Search Application" at the URL "127.0.0.1:5000/user". The navigation bar includes options: Use, User, Insert, Update, Delete, Rename, and Logout. A search interface with a search term input field and dropdown menu is visible. Below it is a table with columns: Name, Designation, Email, Discipline/Section, Work, Home/Emergency, and Office. The table contains data for six users:

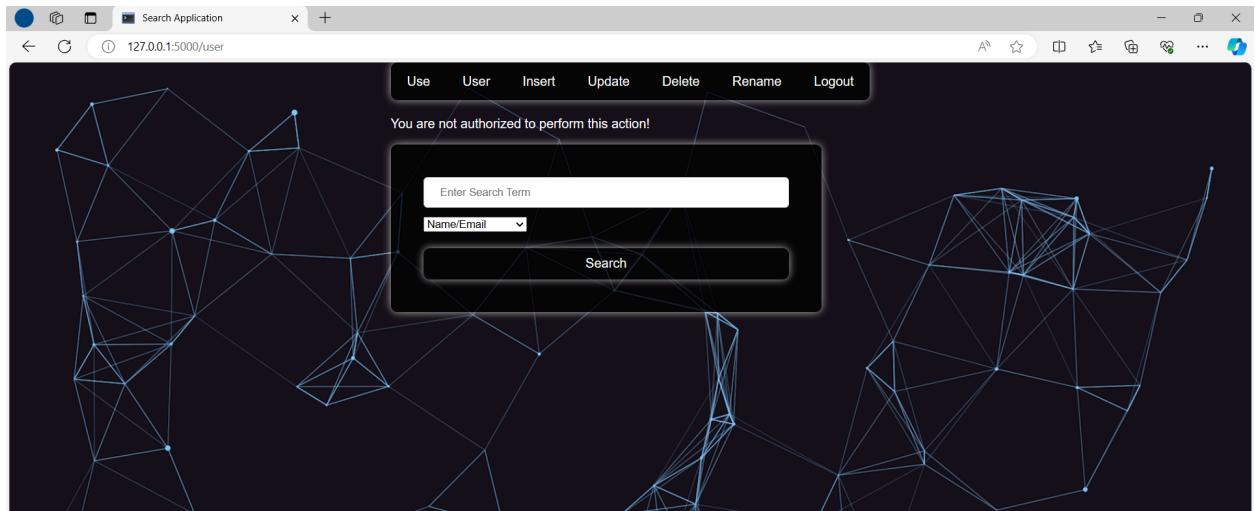
Name	Designation	Email	Discipline/Section	Work	Home/Emergency	Office
Abhishek Bichhawat	Assistant Professor	abhishek.b@iitgn.ac.in	Computer Science and Engineering	2573	1573/	AB13/405A
Anirban Dasgupta	Professor	anirbandg@iitgn.ac.in	Computer Science and Engineering	2463	1463/	AB13/405G
Bhavesh B Jain	Btech	jainbhavesh@iitgn.ac.in	Computer Science Engineering	9000000006	/	
Dhairya Balaji Shah	Btech	shahdhairy@iitgn.ac.in	Computer Science Engineering	9000000004	/	
Mayank Singh	Assistant Professor	singh.mayank@iitgn.ac.in	Computer Science and Engineering	2538	1538/	AB13/403B
Progyan Balaji Das	Btech	progyandas@iitgn.ac.in	Computer Science Engineering	9000000003	/	

Similarly, we can search for information about anyone according to our needs.

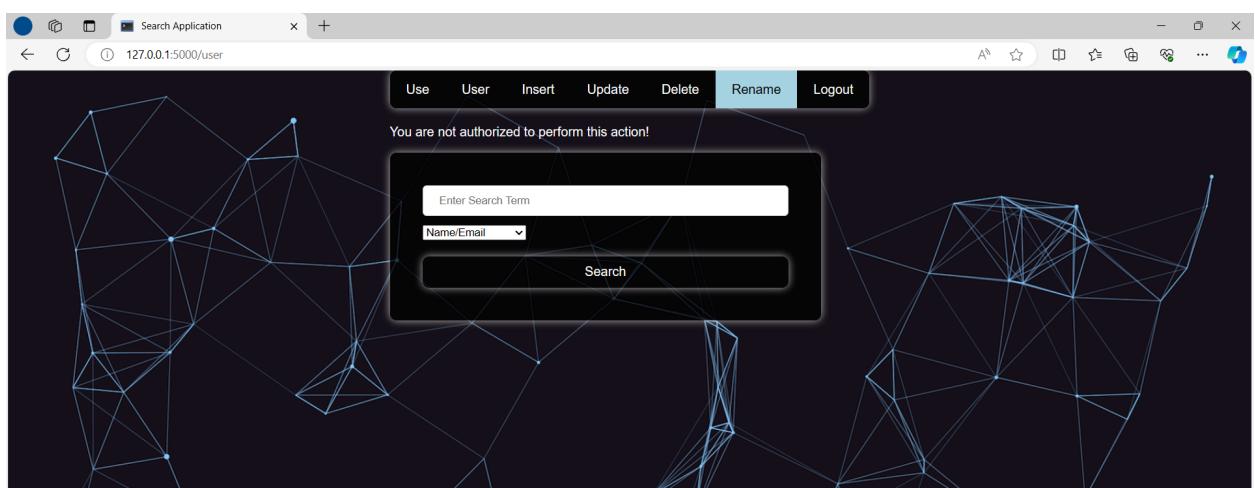
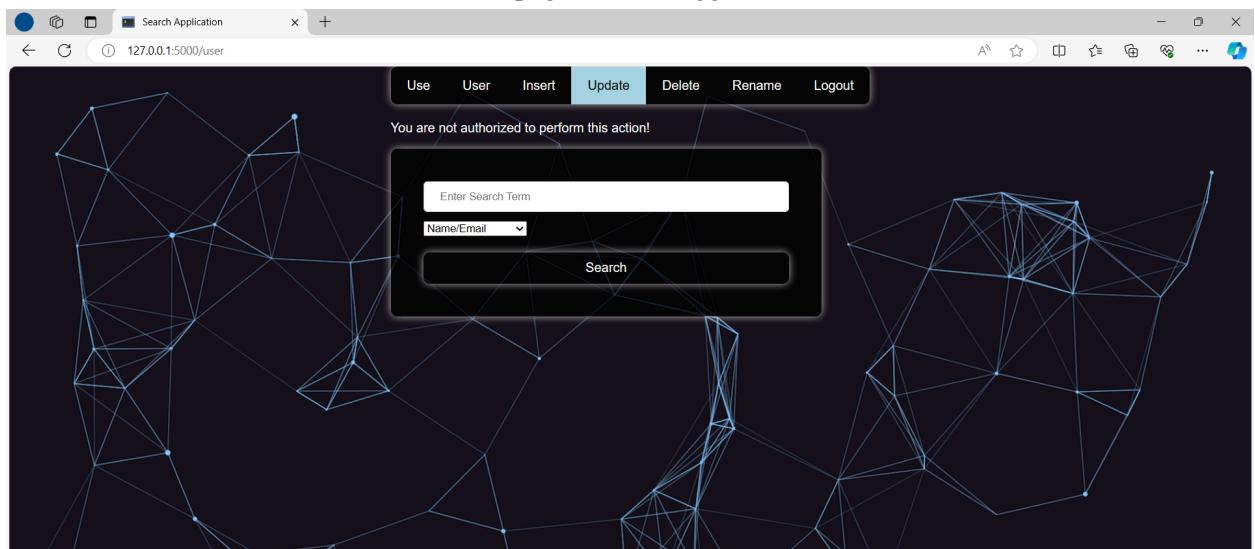
Now, let us look at the different options on the navigation bar. We are currently on the user page, so if we click User, We are redirected to the same page without any table showing. But if we opt for anything other than the User page, say Use page. Here is what it looks like select the Use option.

The screenshot shows the same web browser window, but the "Use" button in the navigation bar is now highlighted in blue. The rest of the interface remains the same, including the search bar and the table of user data.

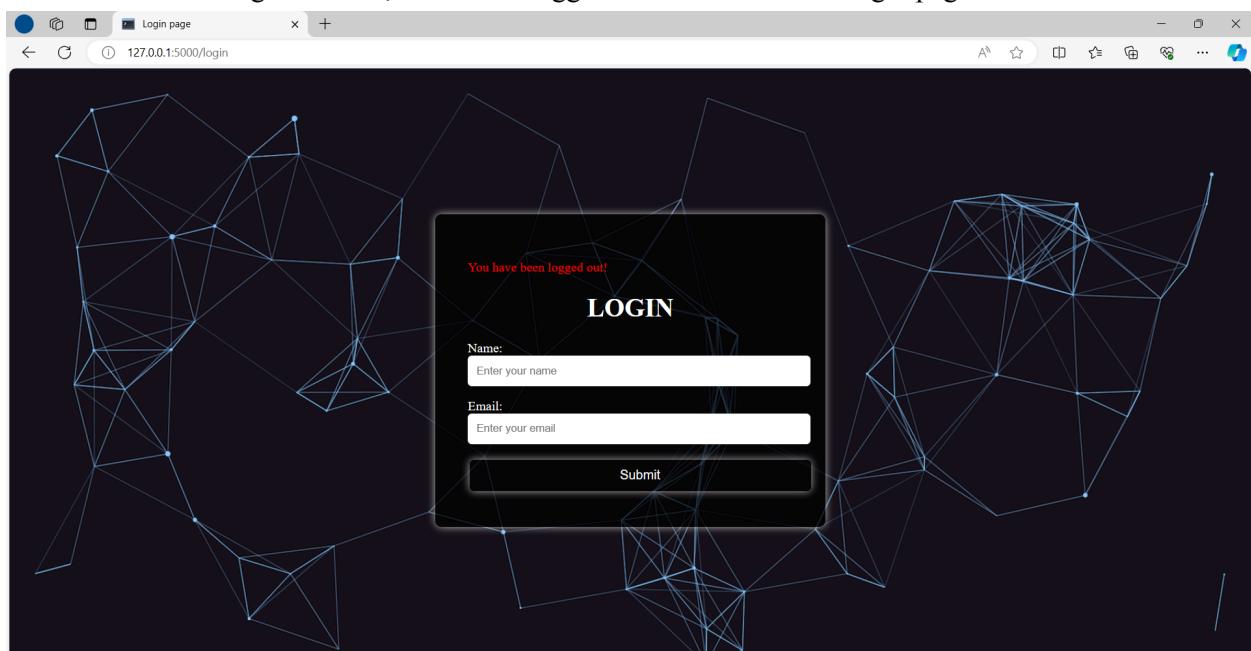
As shown below, we will be redirected to the User page, again saying, "You are not authorized to perform this action!" This is expected because we are not administrators. Only administrators can access the Use, Insert, Update, Delete, and Rename options. So, it will send a message and redirect you to the User page.



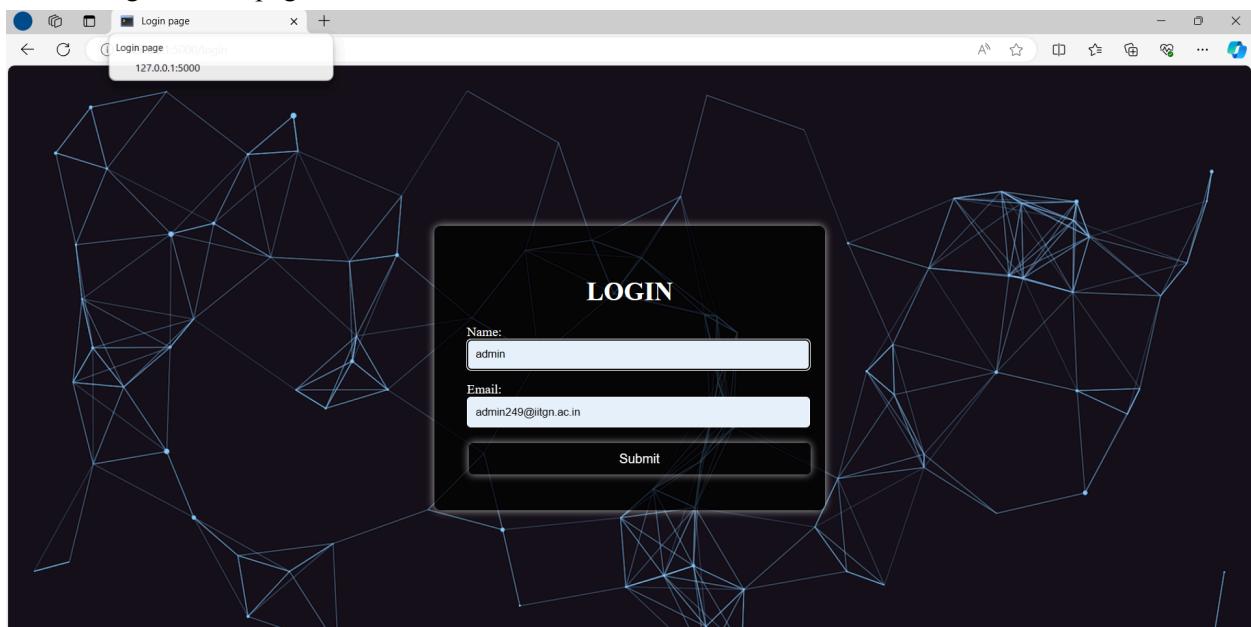
This is shown below; we can't access these pages unless logged in as administrators.

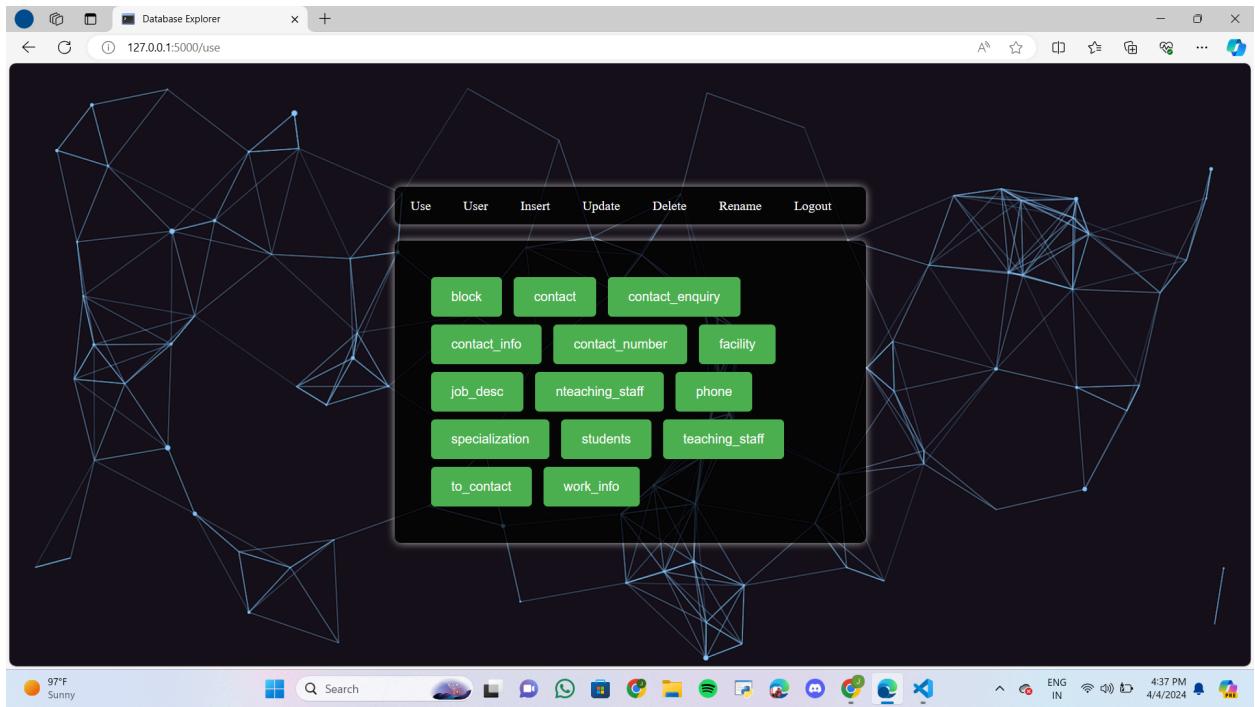


When we click the logout button, we will be logged out and land on the login page.

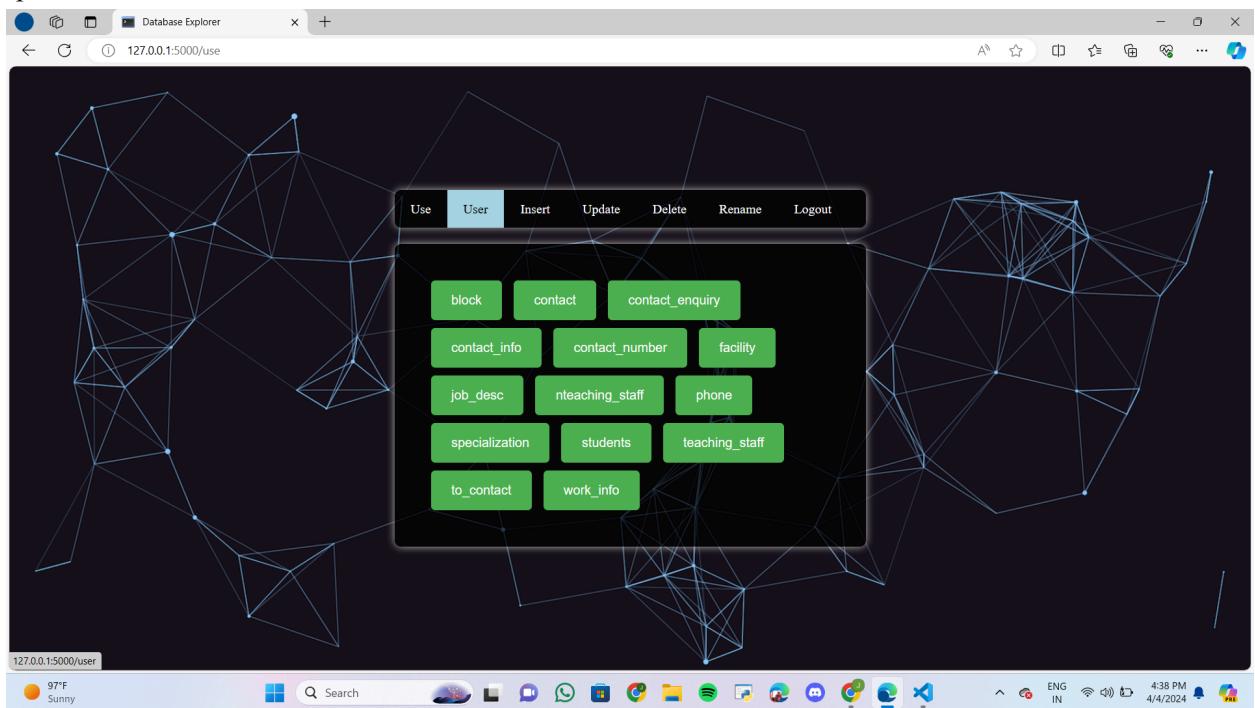


Now, we will try to log in as an administrator. The administrator's Name is "admin", and the email is "[admin249@iitgn.ac.in](mailto:admin249@iitgn.ac.in)". With this, we will log in as an administrator and land on the Use page rather than the regular User page.

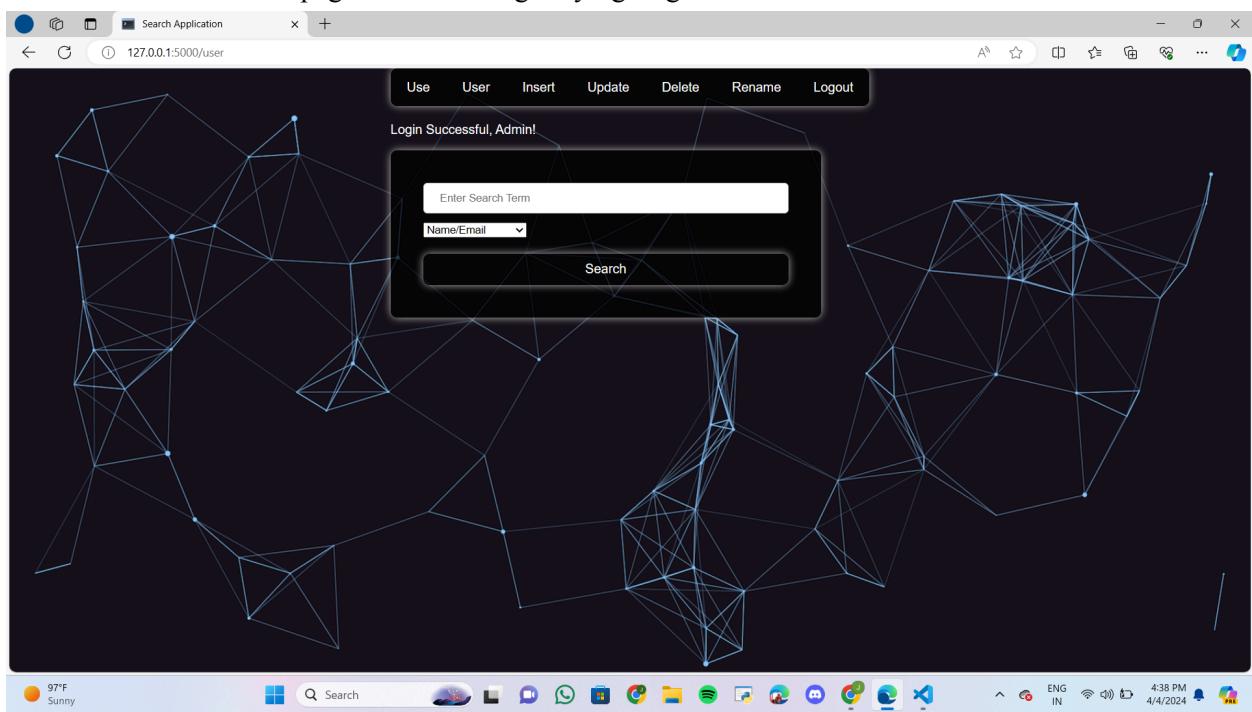




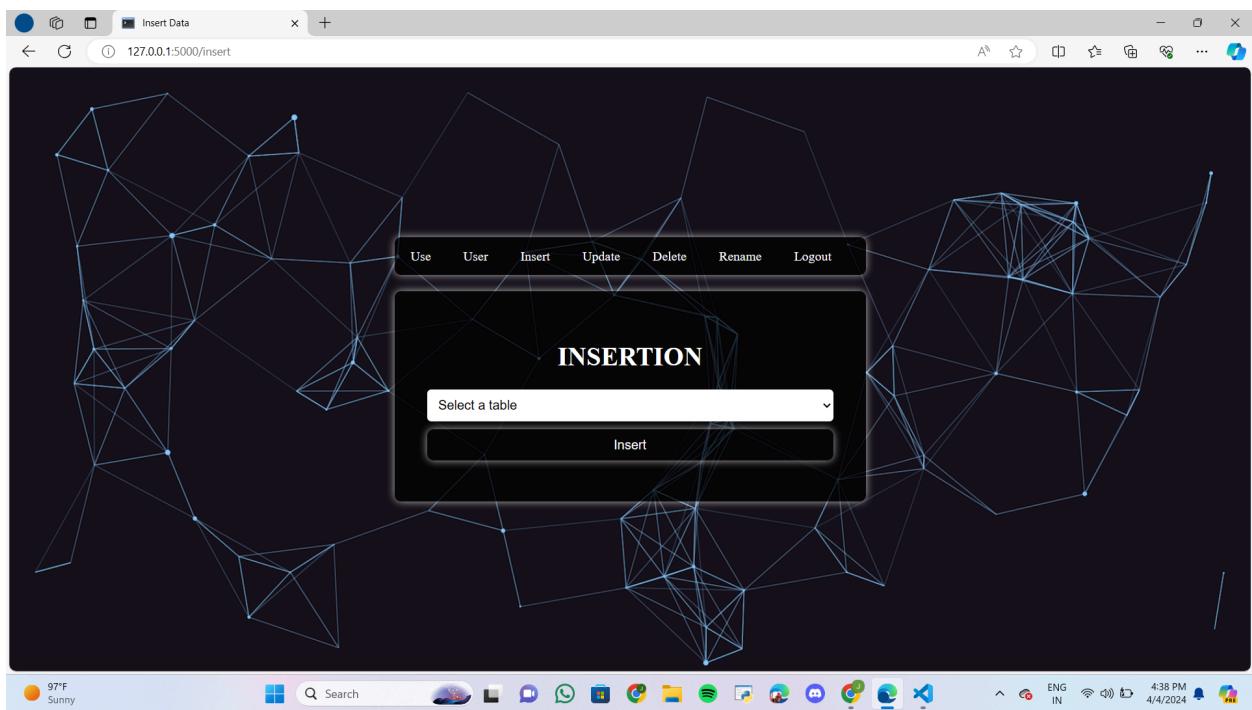
The above page is the Use page; here we can see all the tables in our database. Now, let's check the User option in Navbar.



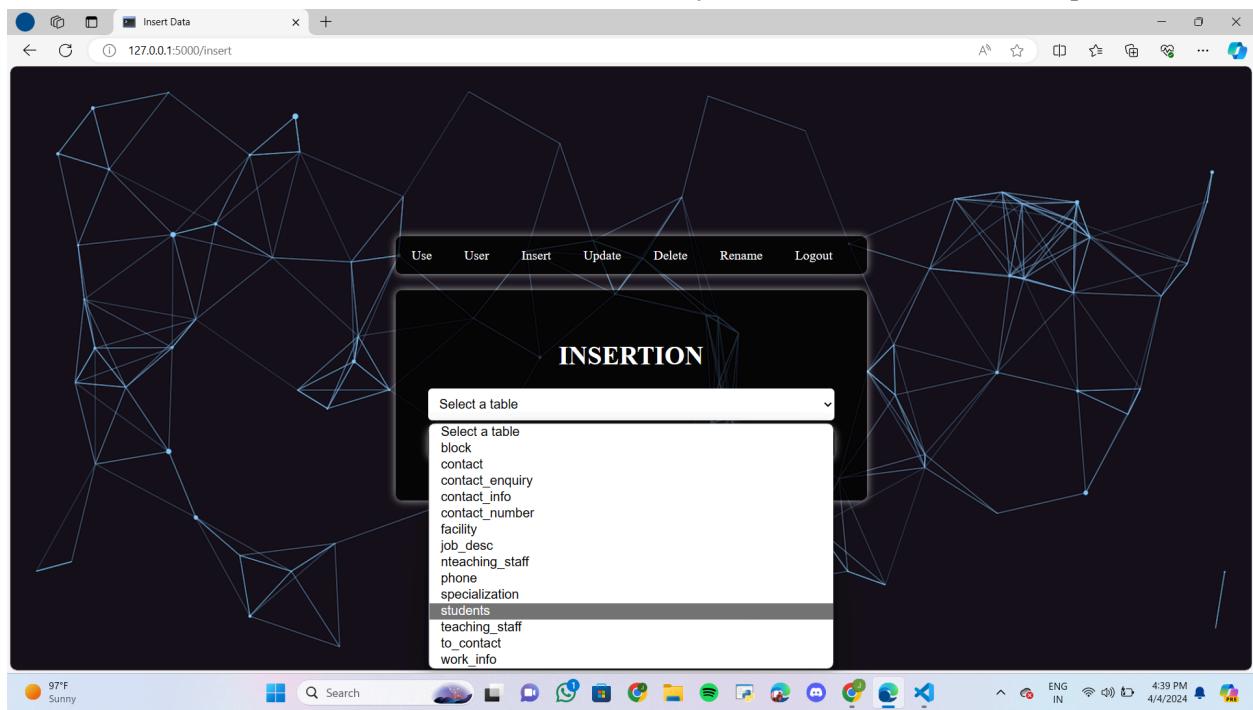
We will land on the User page with a message saying Login successful Admin!



As we log in as the admin, we can access the Use, Insert, Update, Delete, and Rename options. Shown Below:



Let us see how this insertion is done; select the table name you want to insert into from dropdown:

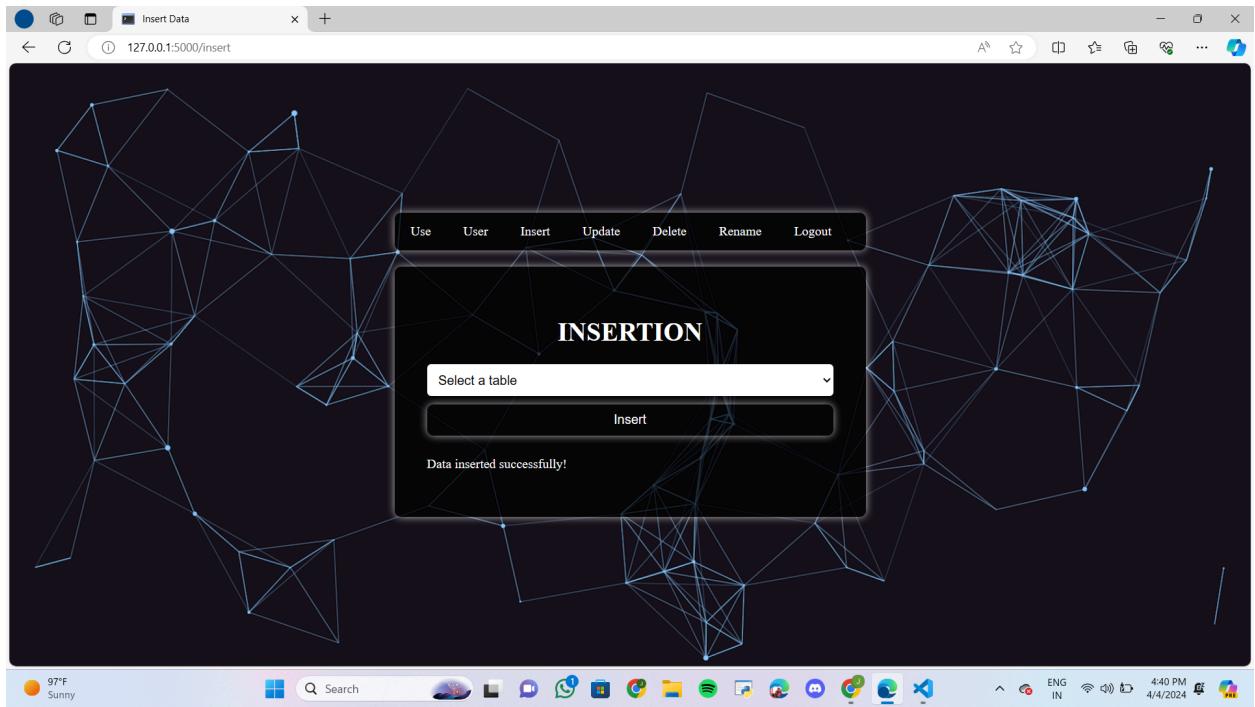


Within the same page, we will get attributes of the selected table, asking to insert the values:

The screenshot shows a web browser window titled "Insert Data" with the URL "127.0.0.1:5000/insert". The page has a dark background with a light blue network graph overlay. At the top is a navigation bar with links: Use, User, Insert, Update, Delete, Rename, and Logout. Below the navigation bar is a central "INSERTION" form. The form has a dropdown menu set to "students" and seven input fields. The input fields are labeled: Enter Roll\_number, Enter First\_name, Enter Middle\_name, Enter Last\_name, Enter Email\_id, Enter Discipline, and Enter Program. A large "Insert" button is at the bottom of the form.

This screenshot shows the same web application after values have been inserted. The "Insertion" form now contains the following data:  
- Roll\_number: 2111007  
- First\_name: K  
- Middle\_name: K  
- Last\_name: K  
- Email\_id: K  
- Discipline: K  
- Program: K

As shown above, give some values to insert. Note that we did not handle exceptions here. So when we try to provide Null values to an attribute with NOT NULL constraint, it will give a MySQL error. This can be done and could have a popup saying not good insertion in future. Once inserted it will say data inserted.



We may check this on the Use page. Go to the Use page and select the button we want to check, which is the Students table here: We can see the 2111007 entry. This can also be seen on the MySQL workbench with the right code. When checked, it works wonderfully.

Roll_number	First_name	Middle_name	Last_name	Email_id	Discipline	Program
20110002	Abhishek	Balaji	Mungekar	abhishek.mungekar@iitgn.ac.in	Mechanical Engineering	Btech
20110003	Progyan	Balaji	Das	progyandas@iitgn.ac.in	Computer Science Engineering	Btech
20110004	Dhairya	Balaji	Shah	shahdhairya@iitgn.ac.in	Computer Science Engineering	Btech
20110005	Varad	Desh	Seshpande	varadseshpande@iitgn.ac.in	Chemical Engineering	Btech
20110006	Bhavesh	B	Jain	jainbhavesh@iitgn.ac.in	Computer Science Engineering	Btech
2111007	K	None	K	K	K	K
21110089	JAY	None	JAY	ja@	B	C
252	Vinay	Amit	Mappa	mappavinay@iitgn.ac.in	Mechanical Engineering	Btech

Now let's check the update option. Let's update the Name of roll number 2111007 student's First name from K to Jay:

Update Data

127.0.0.1:5000/update

students

Enter value for Roll\_number  
2111007

Enter value for First\_name  
Enter value for First\_name

Enter value for Middle\_name  
Enter value for Middle\_name

Enter value for Last\_name  
Enter value for Last\_name

Enter value for Email\_id  
Enter value for Email\_id

Enter value for Discipline  
Enter value for Discipline

Enter value for Program  
Enter value for Program

Enter attribute to update:  
First\_name

Enter new value:  
Jay

Tomorrow's high  
To break record

Search

ENG IN 4:41 PM 4/4/2024

Database Explorer

127.0.0.1:5000/usage/students

Roll_number	First_name	Middle_name	Last_name	Email_id	Discipline	Program
20110002	Abhishek	Balaji	Mungekar	abhishek.mungekar@iitgn.ac.in	Mechanical Engineering	Btech
20110003	Progyan	Balaji	Das	progyandas@iitgn.ac.in	Computer Science Engineering	Btech
20110004	Dhairya	Balaji	Shah	shahdhairya@iitgn.ac.in	Computer Science Engineering	Btech
20110005	Varad	Desh	Seshpande	varadseshpande@iitgn.ac.in	Chemical Engineering	Btech
20110006	Bhavesh	B	Jain	jainbhavesh@iitgn.ac.in	Computer Science Engineering	Btech
2111007	Jay	None	K	K	K	K
21110089	JAY	None	JAY	ja@	B	C
252	Viney	Amit	Mappa	mappavinay@iitgn.ac.in	Mechanical Engineering	Btech

### Search

Roll\_number

First\_name

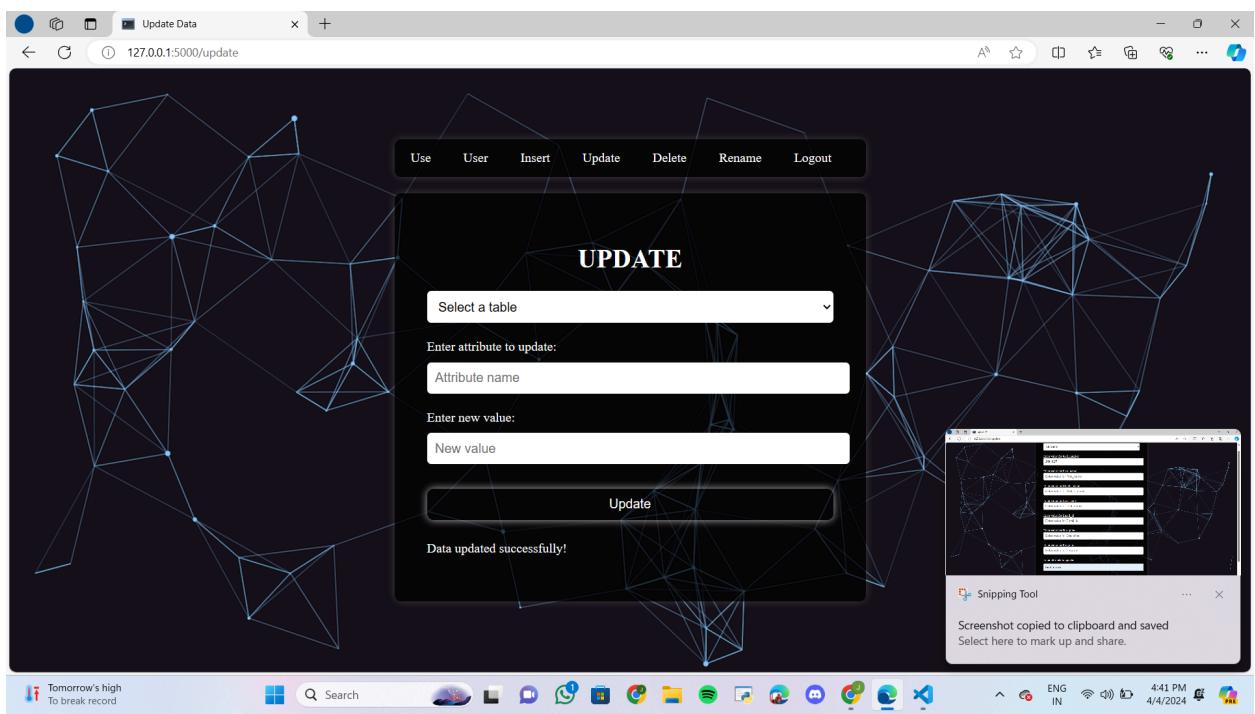
Middle\_name

Last\_name

Email\_id

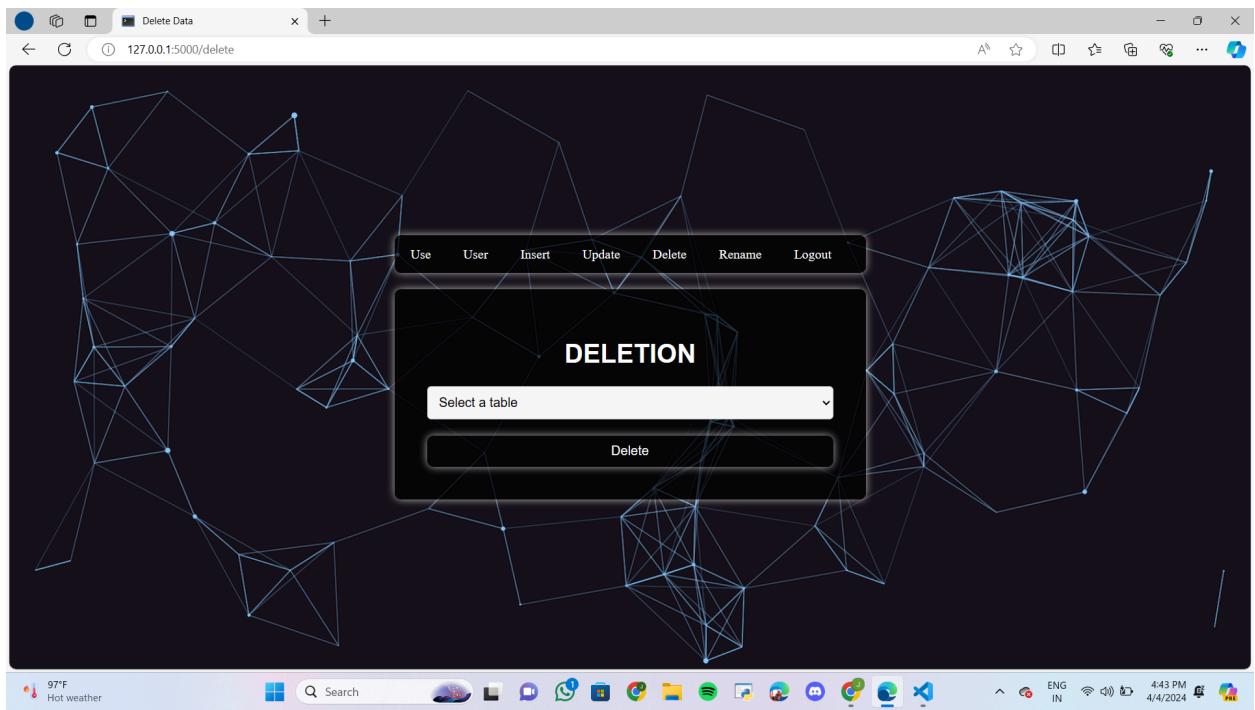
Discipline

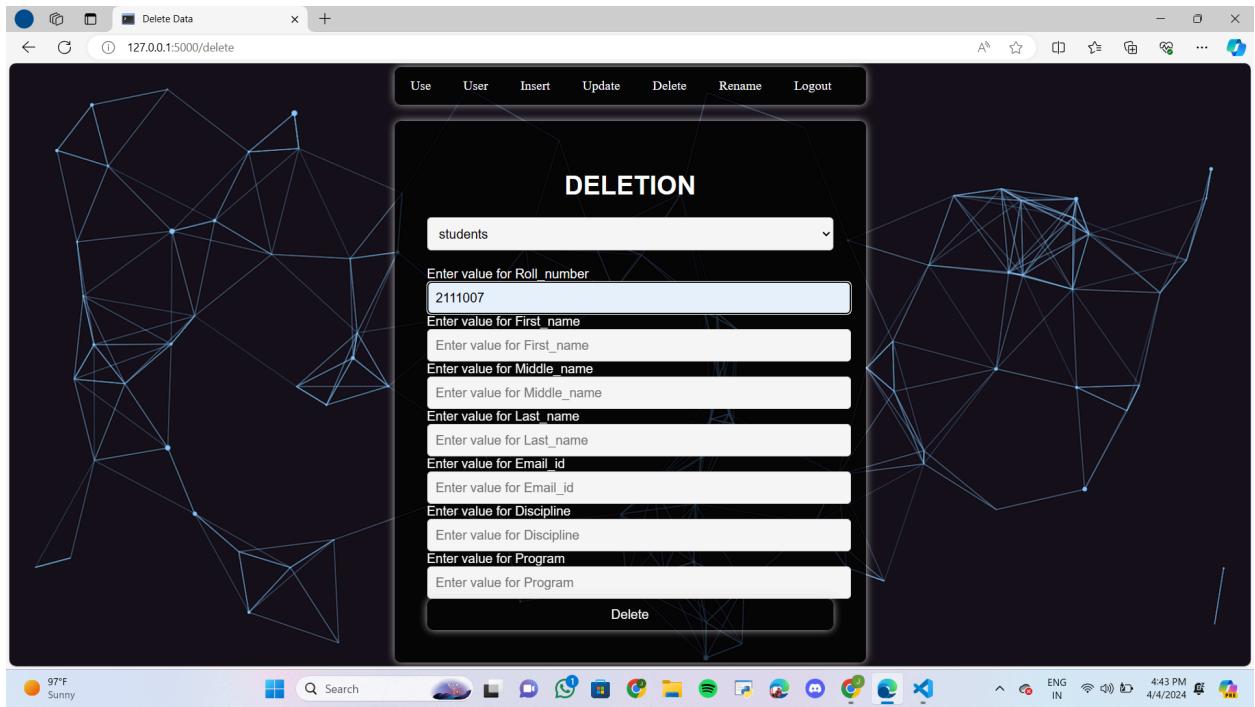
Program



We can see that it's updated in the Students table also 2111007 Jay None K K entry.

Now let us explore the delete page:





We can see that in the table the entry with roll number 2111007 is deleted:

Roll_number	First_name	Middle_name	Last_name	Email_id	Discipline	Program
20110002	Abhishek	Balaji	Munekar	abhishek.munekar@iitgn.ac.in	Mechanical Engineering	Btech
20110003	Progyan	Balaji	Das	progyandas@iitgn.ac.in	Computer Science Engineering	Btech
20110004	Dhairya	Balaji	Shah	shahdhairya@iitgn.ac.in	Computer Science Engineering	Btech
20110005	Varad	Desh	Seshpande	varadseshpande@iitgn.ac.in	Chemical Engineering	Btech
20110006	Bhavesh	B	Jain	jainbhavesh@iitgn.ac.in	Computer Science Engineering	Btech
21110089	JAY	None	JAY	ja@	B	C
252	Vinay	Amit	Mappa	mappavinay@iitgn.ac.in	Mechanical Engineering	Btech

**Search**

Roll\_number

First\_name

Middle\_name

Last\_name

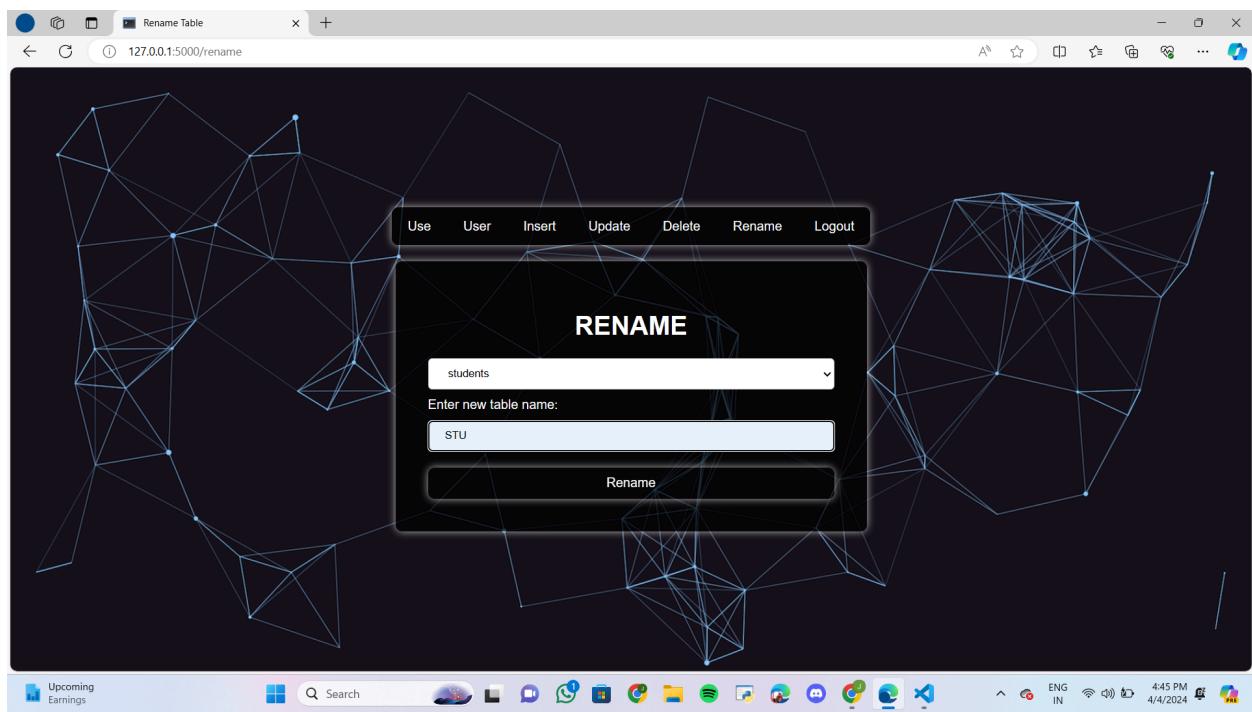
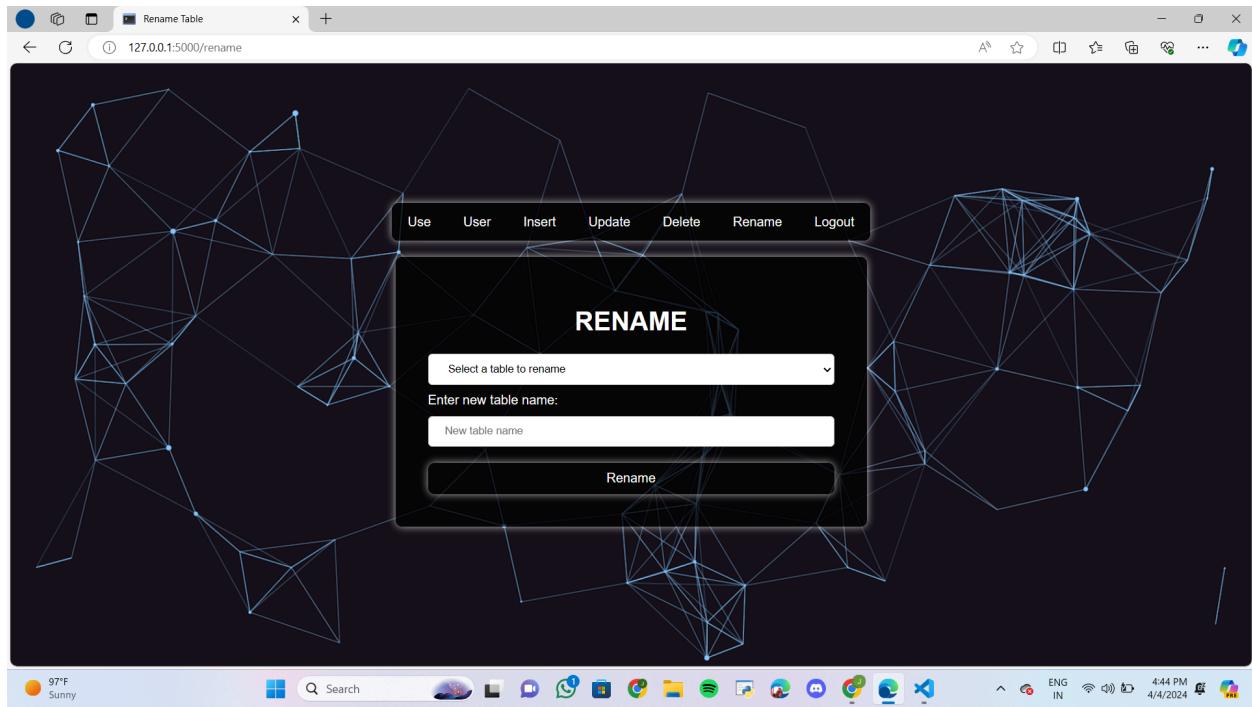
Email\_id

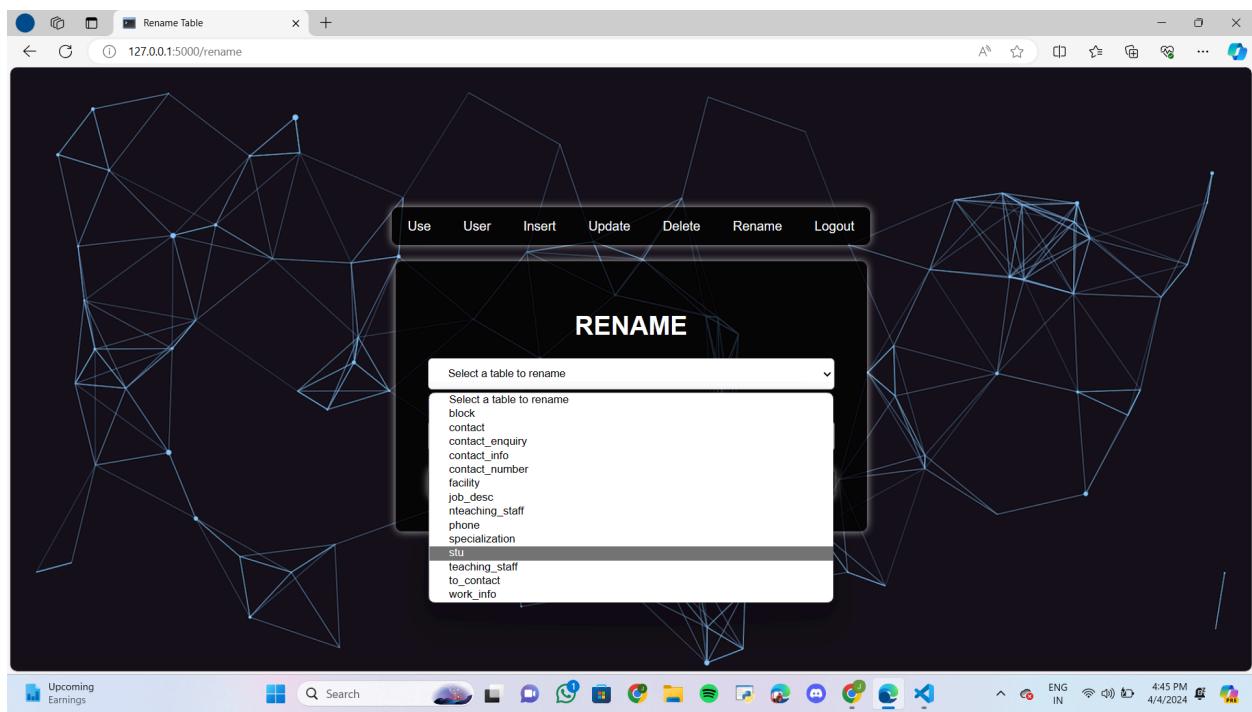
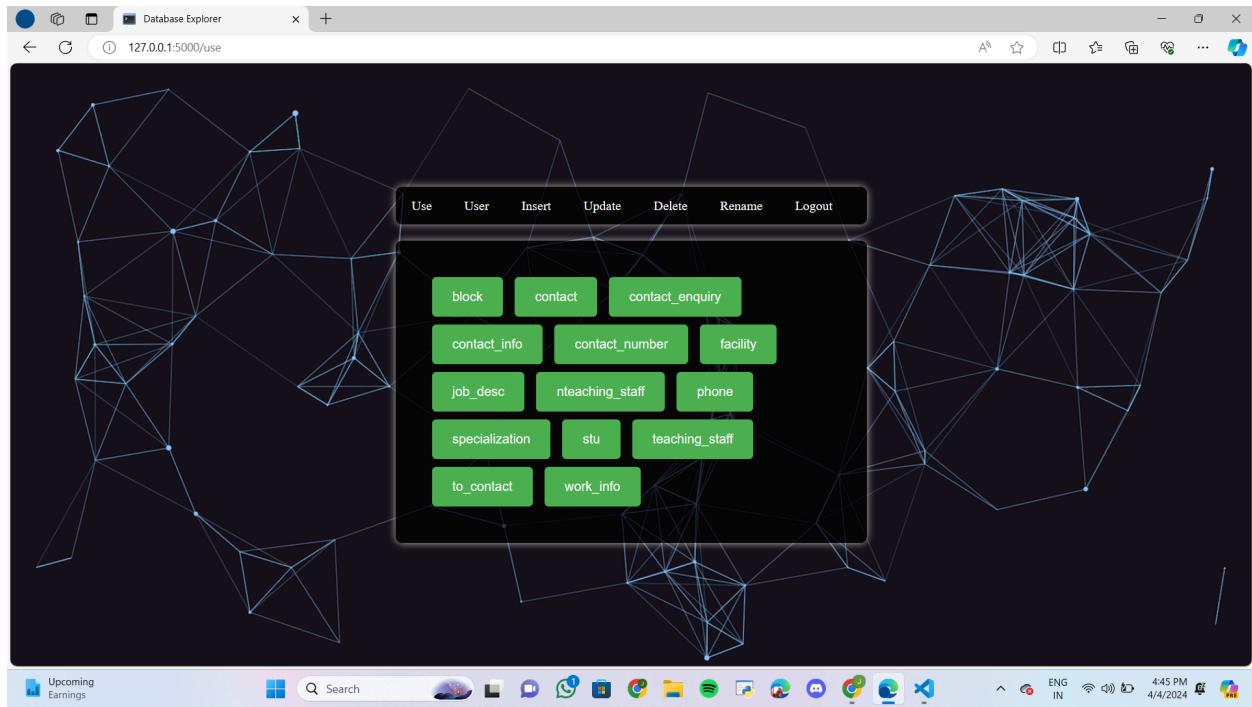
Discipline

Program

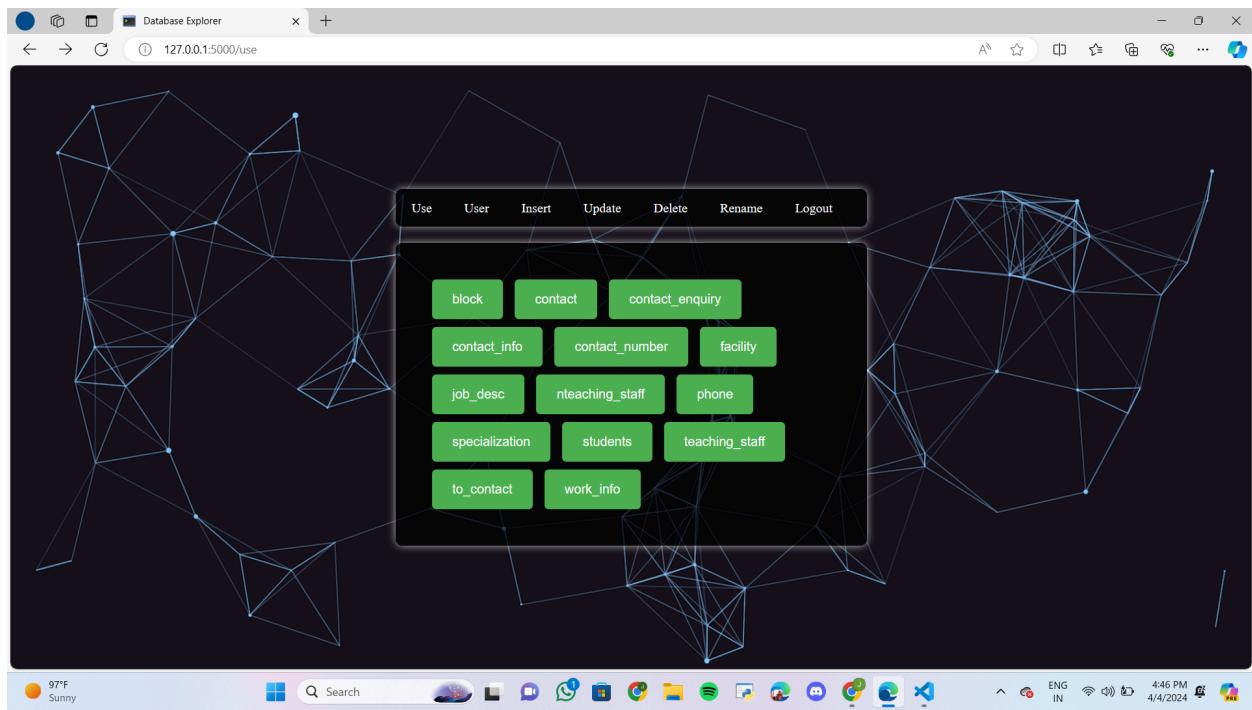
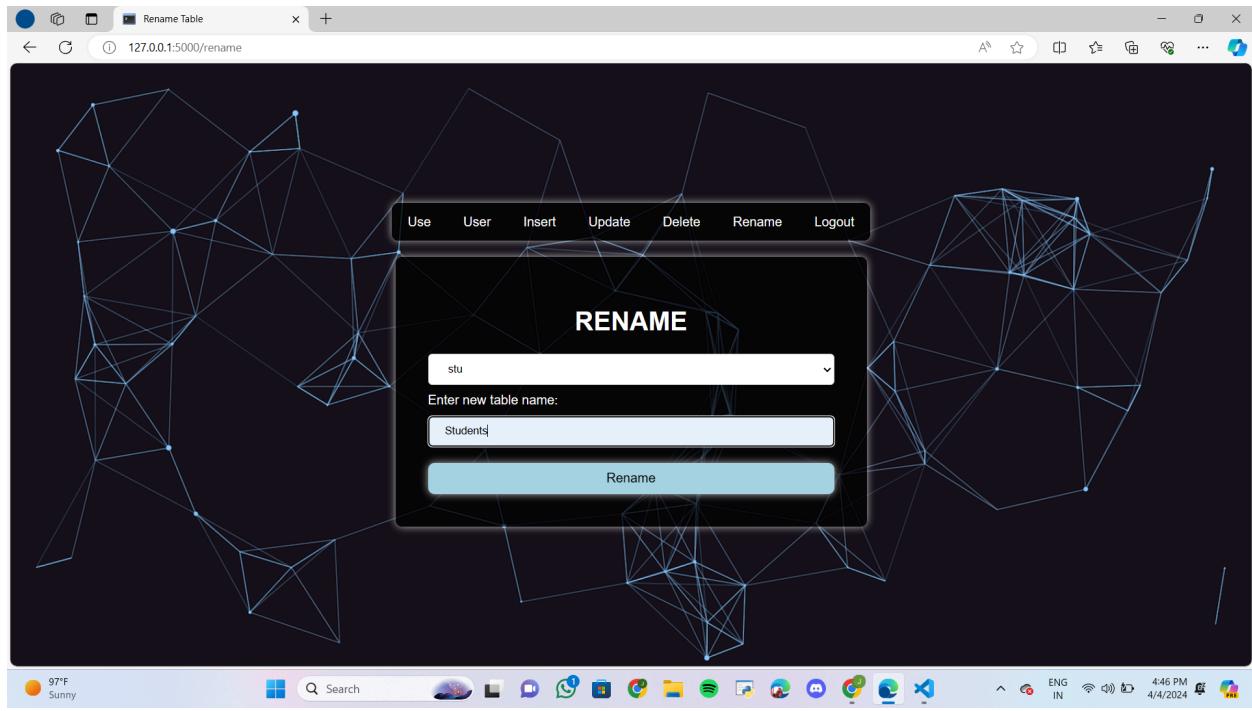
**Search**

Similarly, for Rename, we are renaming the table name as of now and keeping the foreign key constraints in mind, we will again change it back to its previous name, but in between, we can see it updated on our app at the Use page showing all the tables in the database. Rename table Students to STU.





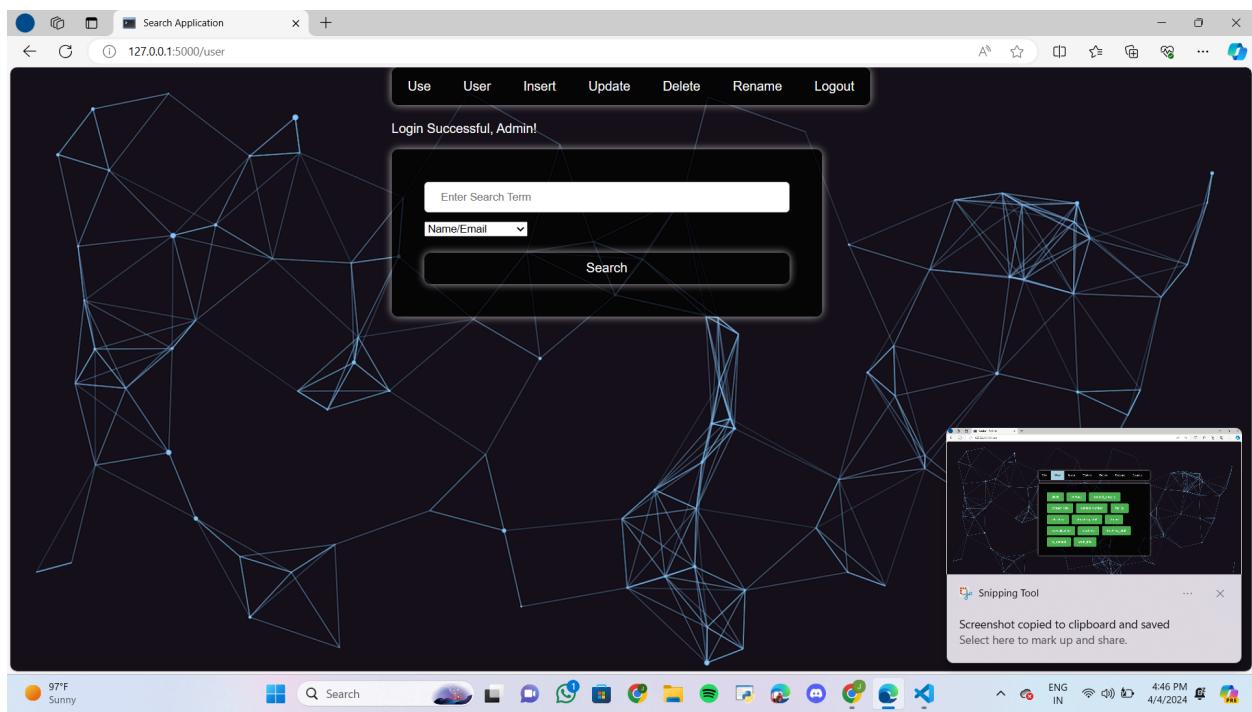
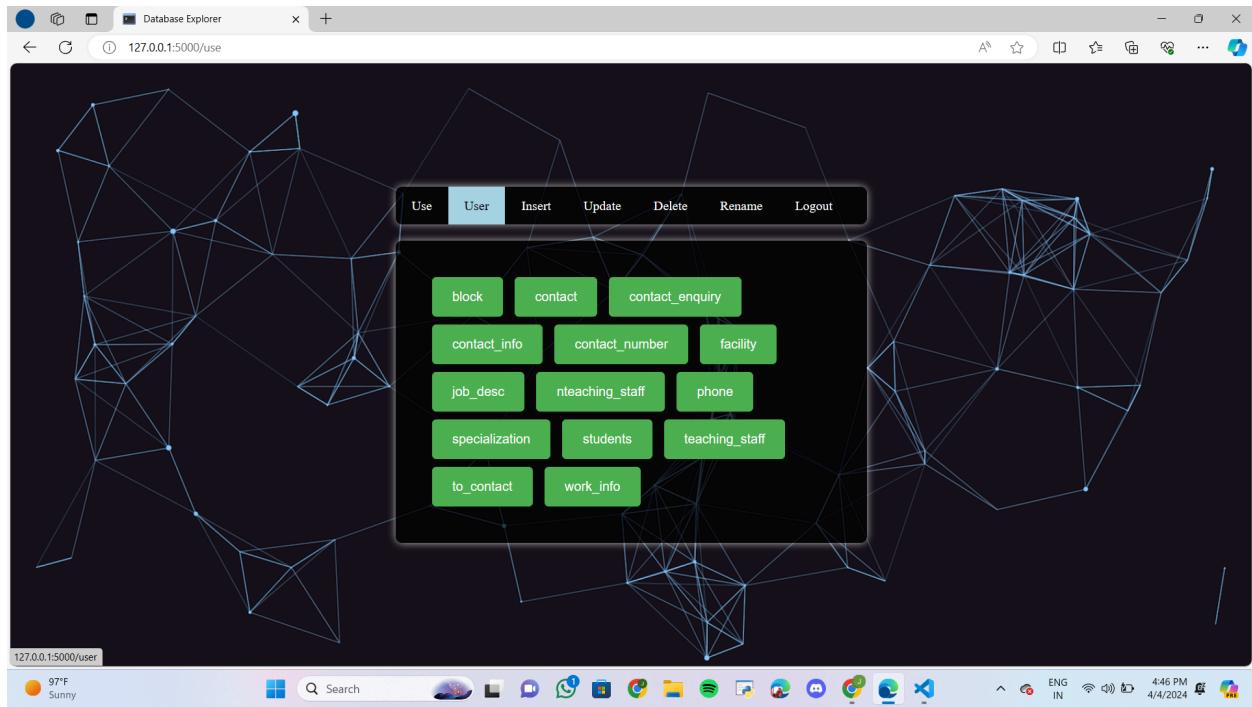
In the above snap, we can see that it is also dynamically updated in the dropdowns.



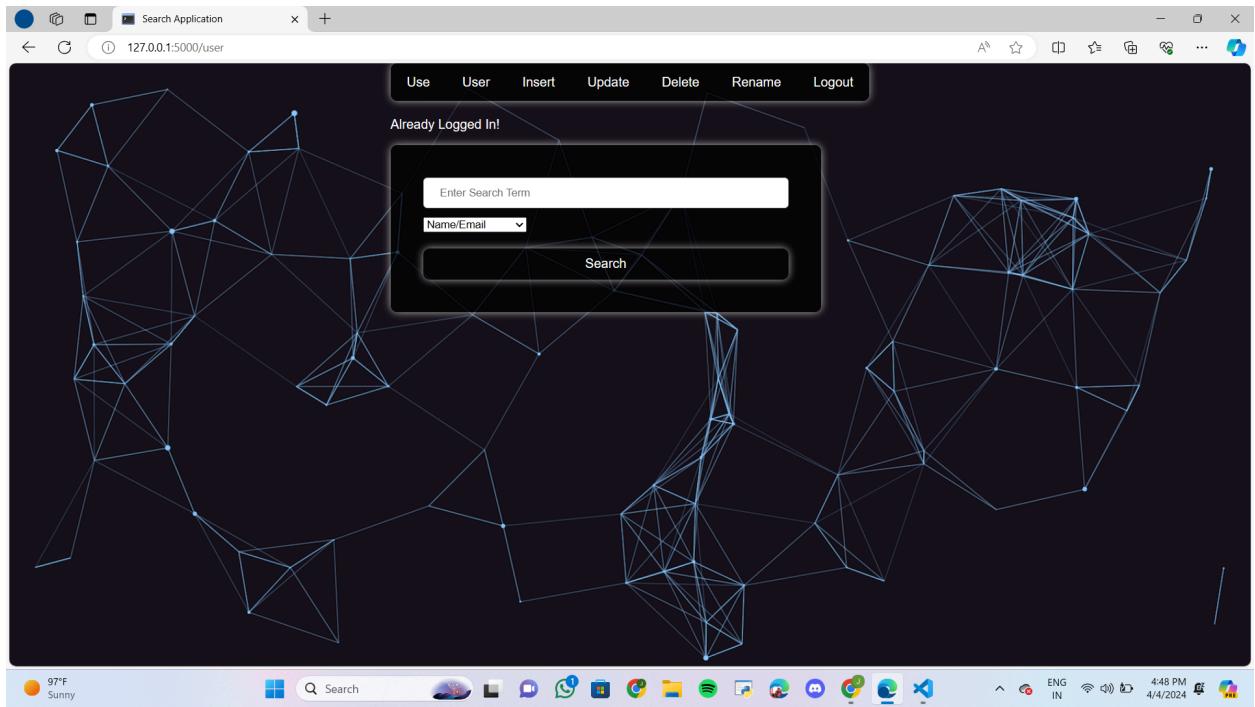
The above is after changing the name back to its original name.

One more point we are using direct original names in the code/app and using joins to fetch data for the table in the User page so when we are doing this we will miss records from the table renamed.

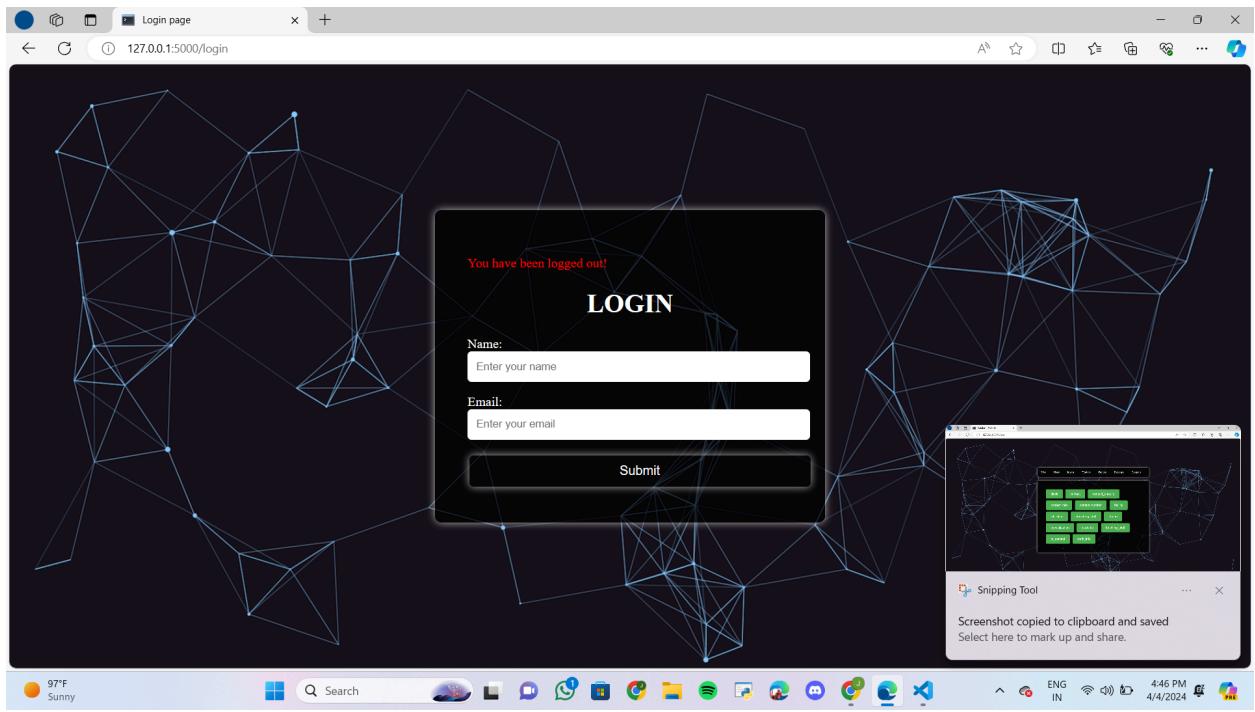
When we try to choose a User page as an admin it will be not different from that of the non administrator one's.



Refresh it to get already logged in as previous this is because we are using sessions for a 5 minute period so that is why this is happening.



Logout page also no difference it will be as below:



As for the Where clause asked we are using it in the User page only while searching by Name/Email or Discipline/Section.

```

if search_by == 'Name/Email':
    cur.execute(query + " WHERE Name LIKE %s OR Email LIKE %s ORDER BY Name", (f"%{search_term}%", f"%{search_term}%"))
else: # search_by == 'Discipline/Section'
    cur.execute(query + " WHERE Discipline_Section LIKE %s ORDER BY Name", (f"%{search_term}%",))

data.extend(cur.fetchall())

cur.close()

return render_template('index1.html', data=data)

```



## CONTRIBUTIONS:

### **Group 1 Members:**

#### **Uday Kumar:**

- Involved in designing the front-end user interface.
- Contributed innovative ideas and actively participated in group discussions.
- Brainstormed the required information regarding front-end development.
- Assisted in documenting the whole file.

#### **Sujith:**

- Contributed to front-end development, focusing on user experience and interface design.
- I also contributed to answering the questions in section 3.3.
- Brainstormed the required information regarding front-end development.
- Assisted the other team members of G2 to complete the responsibility of G1 and G2 (3.3 section).

#### **Ajith:**

- Participated in designing front-end elements and user interactions.
- Assisted the teammates of G2 with the final draft of the responsibility of the G2 section.
- Assisted in drafting the whole document.

#### **Keerthi:**

- Contributed to front-end development, focusing on user experience and interface design
- Assisted the other team members of G1 to complete the responsibility of G1 and G2 (3.3 section).
- Actively engaged in the responsibility of G1 and provided creative and original ideas.
- Assisted in completing tasks related to G1 and G2 responsibilities.

### **Group 2 Members:**

#### **Jethru:**

- Guided and motivated the whole team from the beginning to the end of the assignment.

- Worked on the back-end development, integrating MySQL into the WebApp.
- Actively engaged in the responsibility of G2 and provided creative and original ideas.
- Made sure that every team member gets enough share to work on.

**Sudharshan:**

- I worked on the back-end development and performed the update, delete, and rename operations.
- Actively engaged in the responsibility of G2 and provided creative and original ideas.
- Assisted the other team members in completing the responsibility of G1 and G2 (3.3 section).
- Assisted in documenting the whole file.

**Vinay Marka:**

- Contributed to the back-end development, integrating MySQL into WebApp.
- Engaged in brainstorming sessions to determine the essential information needed.
- I also contributed to answering the last section 3.3.
- Assisted in documenting the whole file.

**Hima Sagar:**

- I worked on the back-end development, primarily performing the update, delete, and rename operations.
- Assisted the other team members in completing the responsibility of G1 and G2 (3.3 section).
- Also assisted in documenting the responsibility of the G1 and G2 sections.
- Assisted the teammates in documenting the whole file.

**Rudreshwar:**

- Worked on the back-end development, integrating MySQL into the WebApp.
- Assisted in completing tasks related to G1 and G2 responsibilities.
- Participated in discussions to ensure all necessary information was identified and included.
- Responsible for answering the last section.

In our team, we ensured each person had a fair share in the project, showing our commitment to fairness and collaboration. Whether in Group 1 or Group 2, everyone played their part, ensuring that our collective effort led to a fruitful outcome.