

CS 432: Databases

Assignment 4: DEPLOYING THE DBMS

3.1 Responsibility of G1:

1. The G1 takes two feedbacks from the stakeholders, one initial feedback (on or before 8th April 11:59 PM), and then makes relevant changes as suggested per the first feedback, then final feedback (on or before 11th April 11:59 PM) post changes. The write-up/documentation should have screenshots before the first feedback, after the first feedback, and after the second feedback. If a team discusses with multiple stakeholders, please fill out the forms again (Initial and final feedback forms).

Feedback:

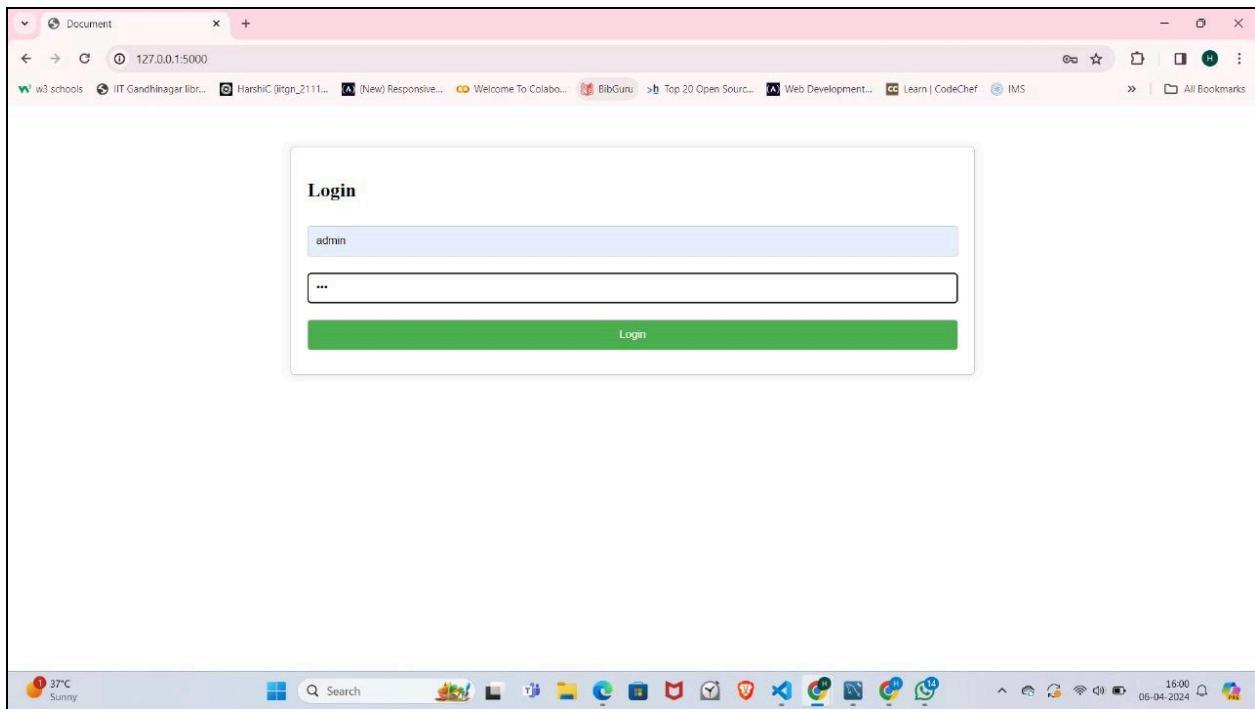
Shweta Bichawat and Barkha Govind from the hostel office provided valuable feedback on our website. They appreciate its utility but suggested incorporating additional features like hostel norms. They raised concerns about the website's performance with increased data volumes, emphasising its need to remain user-friendly while handling large datasets. They also recommended consolidating similar data columns, such as those for stay rooms and store rooms, into one column and proposed including information about secretarial roles and the hostel in charge.

- It should be able to accommodate large datasets.

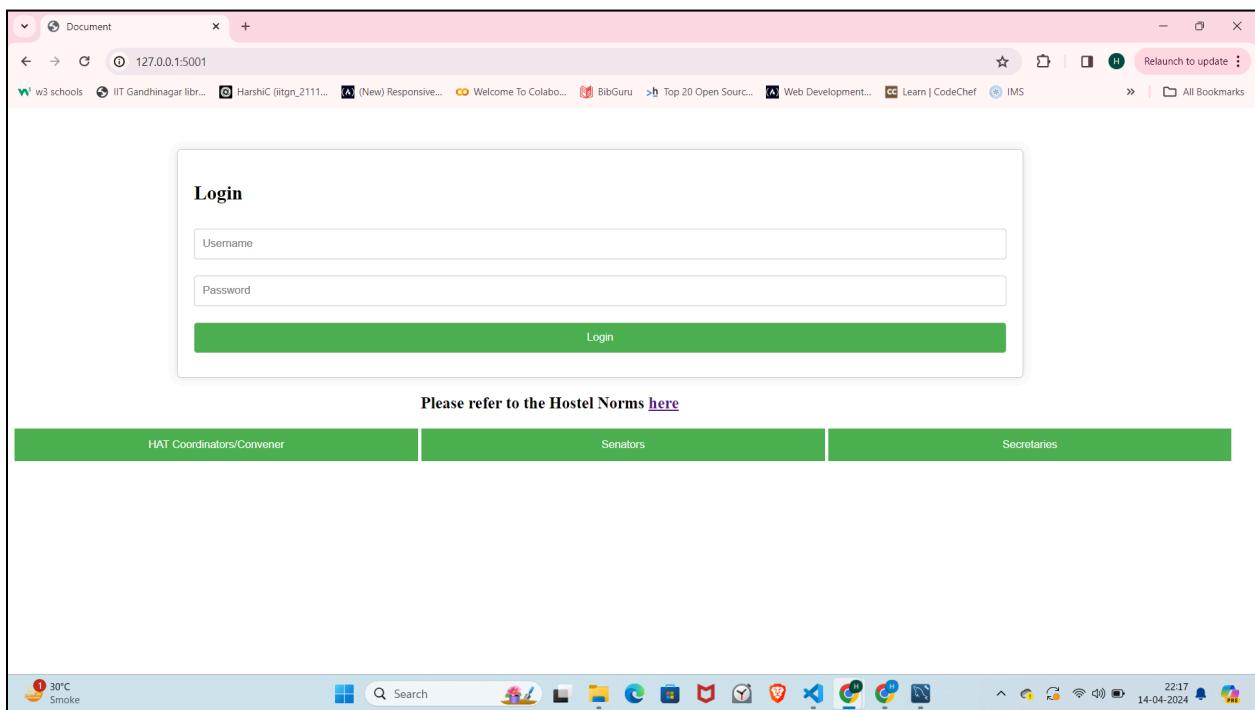
One of the stakeholders' concerns was whether this website would work for larger datasets. For this, the student's table in our database has around 10,000 entries and works properly. If we reduce the entries to 2000 (the approximate number of students in IITGN), the website will work more smoothly compared to before.

- Should contain information about Hostel norms

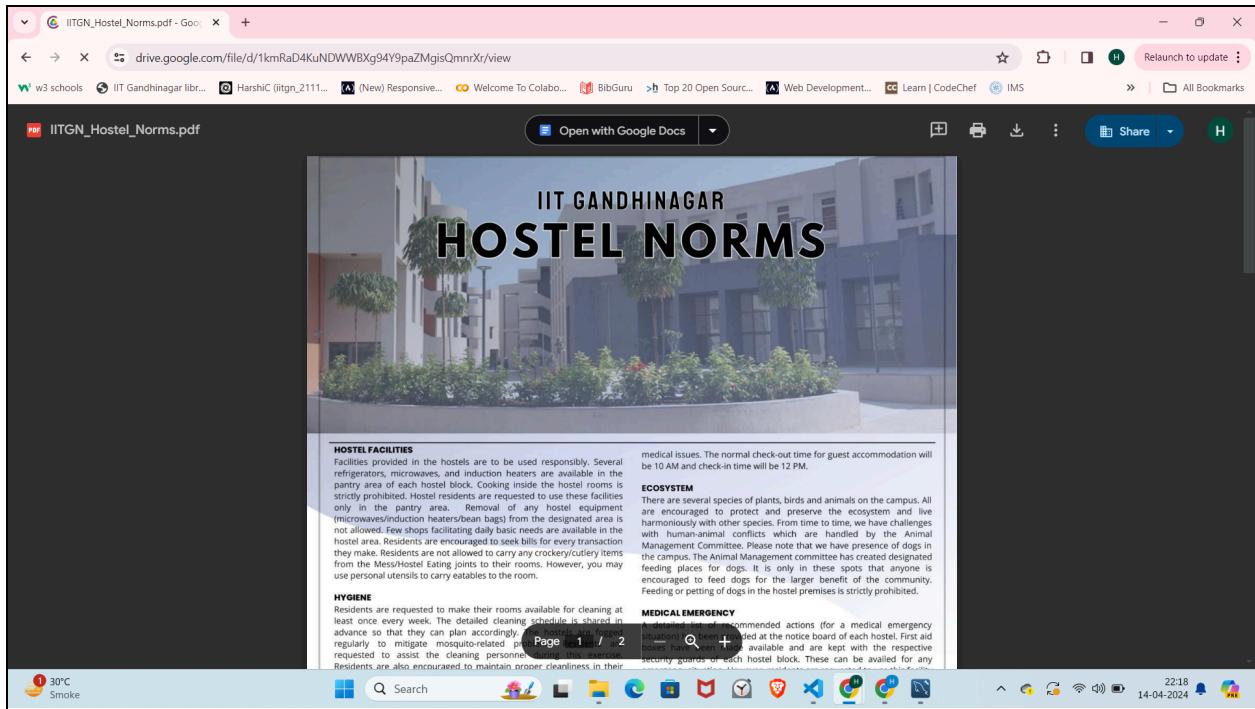
- Before Feedback:



- After feedback:

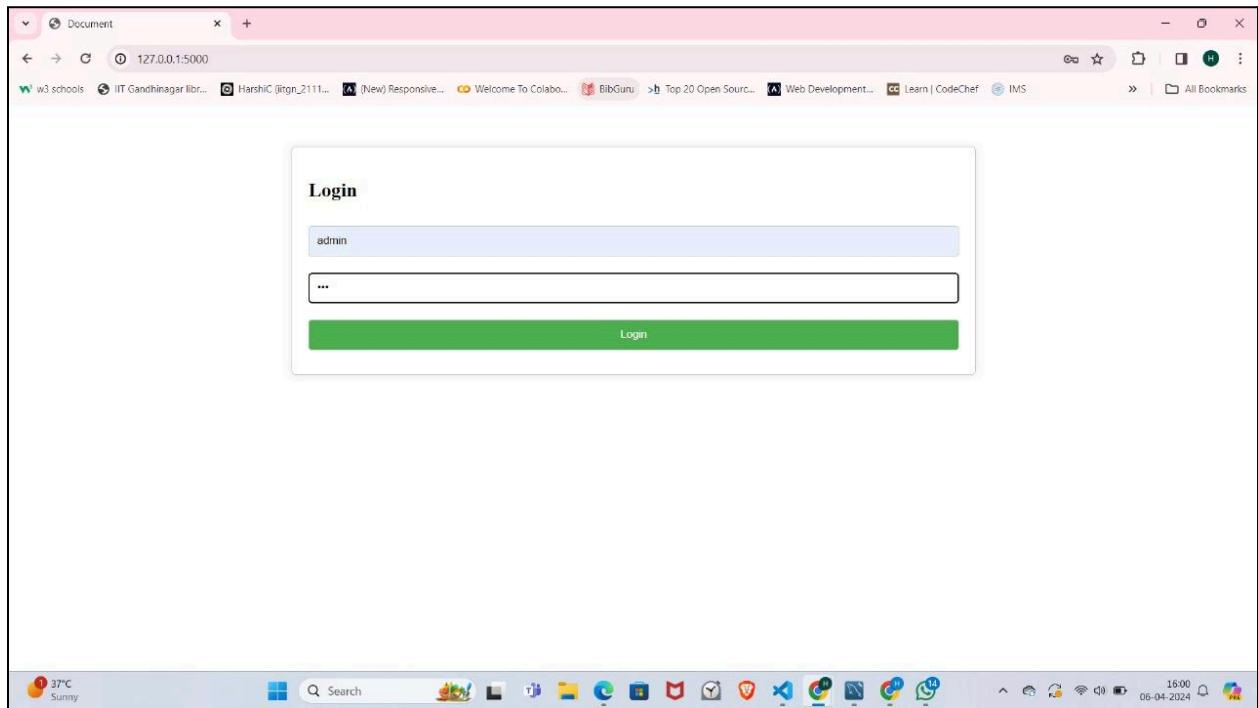


In the above image, you can see that we have added a link to show hostel norms - "Please refer to the Hostel Norms here."



When you click on that link, this page opens up.

- Should contain information about HAT Coordinators, Senators and Secretaries
- Before Feedback:



- After feedback:

The screenshot shows a web browser window with a pink header bar. The address bar displays "127.0.0.1:5001". The page content includes a "Login" form with fields for "Username" and "Password", and a green "Login" button. Below the form is a link: "Please refer to the Hostel Norms [here](#)". Underneath is a table titled "Secretaries" with columns: "Position", "Name", "ID", "Email", and "Program". The table lists seven positions: Welfare Secy, General Secy, Sports Secy, Cultural Secy, IRP Secy, Academic Secy, and Technical Secy, each with their respective names, IDs, emails, and programs (Btech'21 or Btech '21). The browser's taskbar at the bottom shows various pinned icons.

HAT Coordinators/Convener		Senators		Secretaries
Position	Name	ID	Email	Program
Welfare Secy	Mahendra	21110070	mahendra@iitgn.ac.in	Btech'21
General Secy	Yash Ahire	22110052	ahireyash@iitgn.ac.in	Btech '21
Sports Secy	Adit Rambia	22310065	rambiaudit@iitgn.ac.in	Btech'21
Cultural Secy	Abhishek Meena	22310065	meenaabhishek@iitgn.ac.in	Btech'21
IRP Secy	Parth Deshpande	22310065	deshpandeparth@iitgn.ac.in	Btech'21
Academic Secy	Aditya Gupta	22310065	rambiaudit@iitgn.ac.in	Btech'21
Technical Secy	Naman Dharmani	22310065	dharmaninaman@iitgn.ac.in	Btech'21

When you click on the secretaries button, this information opens up.

The screenshot shows a web browser window with a pink header bar. The address bar displays "127.0.0.1:5001". The page content includes a "Login" form with fields for "Username" and "Password", and a green "Login" button. Below the form is a link: "Please refer to the Hostel Norms [here](#)". Underneath is a table titled "Senators" with columns: "Position", "Name", "ID", "Email", and "Program". The table lists three positions: Senator, Senator, and Senator, each with their respective names, IDs, emails, and programs (Btech'21 or Btech '21). The browser's taskbar at the bottom shows various pinned icons.

HAT Coordinators/Convener		Senators		Secretaries
Position	Name	ID	Email	Program
Senator	Dhruv Gupta	21110070	guptadhruv@iitgn.ac.in	Btech'21
Senator	Shrijan Sahu	22110052	sahushrijan@iitgn.ac.in	Btech '21
Senator	Nishant Tatar	22310065	tatarnishant@iitgn.ac.in	Btech'21

When you click on the senator's button, this table opens up.

- **Students should not be given the privilege to see room allocation tables.**
- Before feedback:

The screenshot shows a web browser window with multiple tabs open at the top. The active tab displays a student dashboard titled "STUDENT". At the top right is a green "Logout" button. Below it is a navigation bar with several links: Hostels, Students, Person Details, Stay Rooms (which is highlighted in green), Housekeeper, Room Allocation, Furniture and Utilities, Eateries/Shops, Rooms, and Caretaker. A sub-menu for "Rooms" is visible, showing "Store Common Room". The main content area contains a table with columns: Room Number, Availability Status, and Capacity. The data in the table is as follows:

Room Number	Availability Status	Capacity
A.304	0	2
A.305	2	3
A.403	1	2
A.405	2	3
B.203	1	2
B.205	2	3
B.304	0	2
B.305	2	3
B.403	1	2

At the bottom right of the table is a green "Filter" button. The browser's status bar at the bottom right shows the date as 06-04-2024 and the time as 17:46.

- After Feedback:

The screenshot shows a web browser window with multiple tabs open at the top. The active tab displays a student dashboard titled "STUDENT". At the top right is a green "Logout" button. Below it is a navigation bar with links: Hostels, Students, Furniture and Utilities, Eateries/Shops, Caretaker, and Store Common Room. A sub-menu for "Students" is visible. The main content area contains a table with columns: Name, Total Rooms, Gender, and Occupancy Status. The data in the table is as follows:

Name	Total Rooms	Gender	Occupancy Status
Aibaan	23	Male	1
Beauki	20	Female	5
Chimair	20	Female	5
Duven	20	Male	5
Emiet	20	Male	5
Firpeal	20	Male	5
Griwiksh	20	Male	5
Hiqom	20	Male	5
Ijokha	20	Female	5
Jurqia	20	Male	5

At the bottom right of the table is a green "Filter" button. The browser's status bar at the bottom right shows the date as 14-04-2024 and the time as 22:29.

After the feedback tables related to the rooms had been removed from the students' page,

2. Attach screenshots of different views [along with a write-up on their privileges] of the database as seen by different classes of users.

Different views for different users:

1. Admin

- The admin can edit, delete, and filter all the tables and all their entries.

The screenshot shows a web browser window titled "Database Assignment 3" at the URL "127.0.0.1:5001". The page has a pink header bar with various links. Below it, a dark navigation bar contains the word "ADMIN" and a "LogOut" button. The main content area displays a table of data with columns: Name, Total Rooms, Gender, Occupancy Status, and Action. The table lists nine entries. At the bottom right of the table are three buttons: "Filter", "Add Data", and "Delete". The table data is as follows:

Name	Total Rooms	Gender	Occupancy Status	Action
Aibaan	23	Male	1	Edit Delete
Beauki	20	Female	5	Edit Delete
Chimair	20	Female	5	Edit Delete
Duven	20	Male	5	Edit Delete
Emiet	20	Male	5	Edit Delete
Firpeal	20	Male	5	Edit Delete
Griwiksh	20	Male	5	Edit Delete
Hiqom	20	Male	5	Edit Delete
Ijokha	20	Female	5	Edit Delete

The browser's taskbar at the bottom shows various open tabs and system icons.

This page opens up when the admin logs in.

2. Caretaker

- The caretaker can edit and delete entries in a few tables not all. However, they can view all the tables that the admin can view.

The screenshot shows a web browser window titled "Database Assignment 3" at the URL "127.0.0.1:5001". The page is titled "CARETAKER" and features a green "LogOut" button. A navigation bar below the title includes links for Hostels, Students, Person Details, Stay Rooms, Housekeeper, Room Allocation, Furniture and Utilities, Eateries/Shops, Rooms, and Caretaker. A sub-menu for "Rooms" is open, showing the option "Store Common Room". The main content area displays a table with columns: Name, Total Rooms, Gender, and Occupancy Status. The table lists eight entries:

Name	Total Rooms	Gender	Occupancy Status
Aibaan	23	Male	1
Beauki	20	Female	5
Chimair	20	Female	5
Duven	20	Male	5
Emiet	20	Male	5
Firpeal	20	Male	5
Griwiksh	20	Male	5
Hiqom	20	Male	5
Ijokha	20	Female	5

A green "Filter" button is located in the top right corner of the table area. The browser's status bar at the bottom shows the date and time as "14-04-2024 22:29".

This page opens up when the caretaker logs in. The caretaker can edit/delete only a few tables. It cannot edit the above table of hostels.

The screenshot shows a web browser window titled "Database Assignment 3" at the URL "127.0.0.1:5001". The page is titled "CARETAKER" and features a green "LogOut" button. A navigation bar below the title includes links for Hostels, Students, Person Details, Stay Rooms, Housekeeper, Room Allocation, Furniture and Utilities, Eateries/Shops, Rooms, and Caretaker. A sub-menu for "Rooms" is open, showing the option "Store Common Room". The main content area displays a table with columns: Student Roll Number, Room Number, and Action. The table lists ten entries:

Student Roll Number	Room Number	Action
23110018	A 303	<button>Edit</button> <button>Delete</button>
23110057	A 304	<button>Edit</button> <button>Delete</button>
23110030	A 403	<button>Edit</button> <button>Delete</button>
23110007	B 203	<button>Edit</button> <button>Delete</button>
23110049	B 204	<button>Edit</button> <button>Delete</button>
23110019	B 303	<button>Edit</button> <button>Delete</button>
23110059	B 304	<button>Edit</button> <button>Delete</button>
23110036	B 403	<button>Edit</button> <button>Delete</button>
23110012	C 203	<button>Edit</button> <button>Delete</button>

A green "Filter" button is located in the top right corner of the table area. The browser's status bar at the bottom shows the date and time as "14-04-2024 22:29".

The caretaker can edit the above table of students.

3. Students

- Students only have the privilege to view and filter data. Also, as suggested by the hostel office, students should not be able to view data regarding room allocations; therefore, those tables have been removed.

Name	Total Rooms	Gender	Occupancy Status
Aibaan	23	Male	1
Beauki	20	Female	5
Chimair	20	Female	5
Duven	20	Male	5
Emiet	20	Male	5
Firpeal	20	Male	5
Griwiksh	20	Male	5
Hiqom	20	Male	5
Ijokha	20	Female	5
Jurqia	20	Male	5

This page opens up when a student logs in.

3.2 Responsibility of G2:

40 Pts.

1. **Concurrent multi-user access:** Multiple users with different roles can access and update the database concurrently. In such a scenario, two different users cannot update the same item. For example, locks can be applied to tables in MySQL. (10 Points)

The screenshot shows a web application titled "CARETAKER" running on a local server at 127.0.0.1:5000. The interface includes a navigation bar with links for Hostels, Students, Person Details, Stay Rooms, Housekeeper, Room Allocation, Furniture and Utilities, Eateries/Shops, Rooms, and Caretaker. A sub-menu for "Rooms" is open, showing a link to "Store Common Room". The main content area displays a table of student room assignments:

Student Roll Number	Room Number	Action
23110018	A 303	Edit Delete
23110057	A 304	Edit Delete
23110030	A 403	Edit Delete
23110007	B 203	Edit Delete
23110049	B 204	Edit Delete
23110019	B 303	Edit Delete
23110059	B 304	Edit Delete
23110036	B 403	Edit Delete
23110012	C 203	Edit Delete
23110058	C 204	Edit Delete

A green "Filter" button is located in the top right corner of the table area. The system interface includes a header bar with tabs for ChatGPT, MySQL error code:1175 during, Document, Database Assignment 3, and Database Assignment 3. The status bar at the bottom shows system information like temperature (30°C), date (14-04-2024), and time (22:14).

We have implemented locks on our system. We logged in to our website using two portals. One is from the caretaker ID, and the other is from the admin ID. If the admin tries to edit/ delete any of the data, he will be unable to do so because the first user locks the table. It will pop up a message that another user is currently using the table.

Since the caretaker has logged in and is editing the same page, another user(admin in this case) cannot make changes.

2 Implement the changes in the database as per the feedback received from stakeholders.

-According to the feedback from the stakeholders, not many changes were required in the backend. The front end only required changes, like adding hostel norms, HAT Coordinators, and Secretaries. We merged similar columns and removed columns that were not required. Their primary concern was the effectiveness of the website and its speed, but since our website already has large datasets with 10,000 entries, it will easily be able to accommodate data from IIT Gn.

3 Add Google authentication for login and registration. (only IITGN users can log in and register)

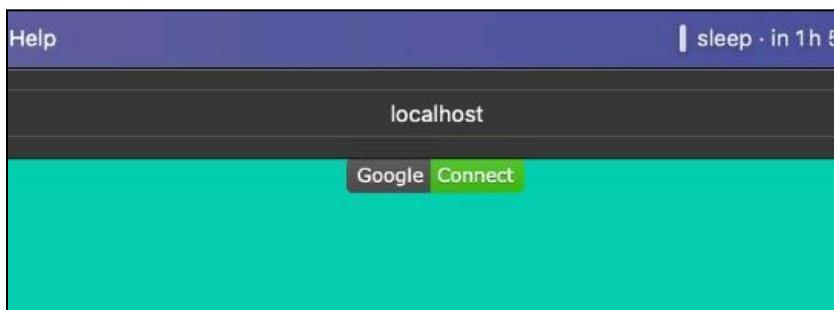
We have implemented Google authentication to allow only IITGN users to log in and added an option to sign in if you are a new user. For new users, the student page is rendered. For this, we used the reference provided in Assignment 4.

At line 58 in `__init__.py`, add this piece of code:

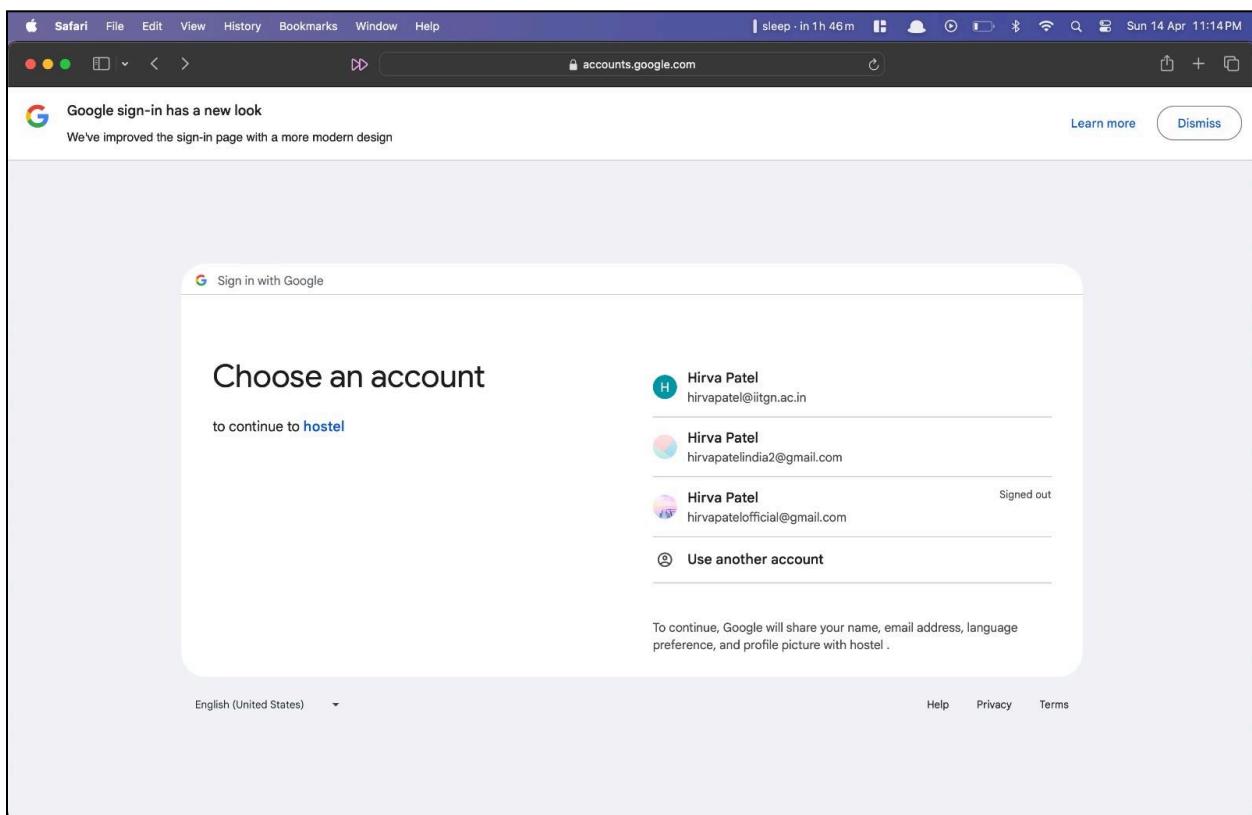
```
GOOGLE_CLIENT_ID = "640414763215-saog82ngnv8ljkhimv9midh1u517kpaj.apps.googleusercontent.com"
GOOGLE_CLIENT_SECRET = "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"
```

```
CONF_URL= 'https://accounts.google.com/.well-known/openid-configuration'
oauth.register(
    name='google',
    client_id=GOOGLE_CLIENT_ID,
    client_secret=GOOGLE_CLIENT_SECRET,
```

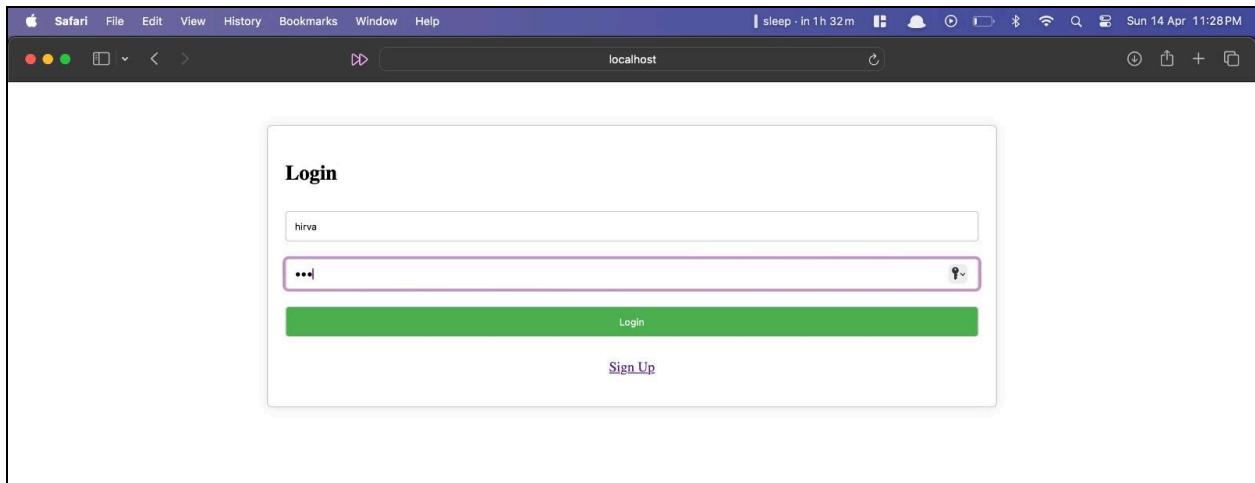
```
server_metadata_url=CONF_URL,  
client_kwargs={  
    'scope': 'openid email profile'  
}  
)
```



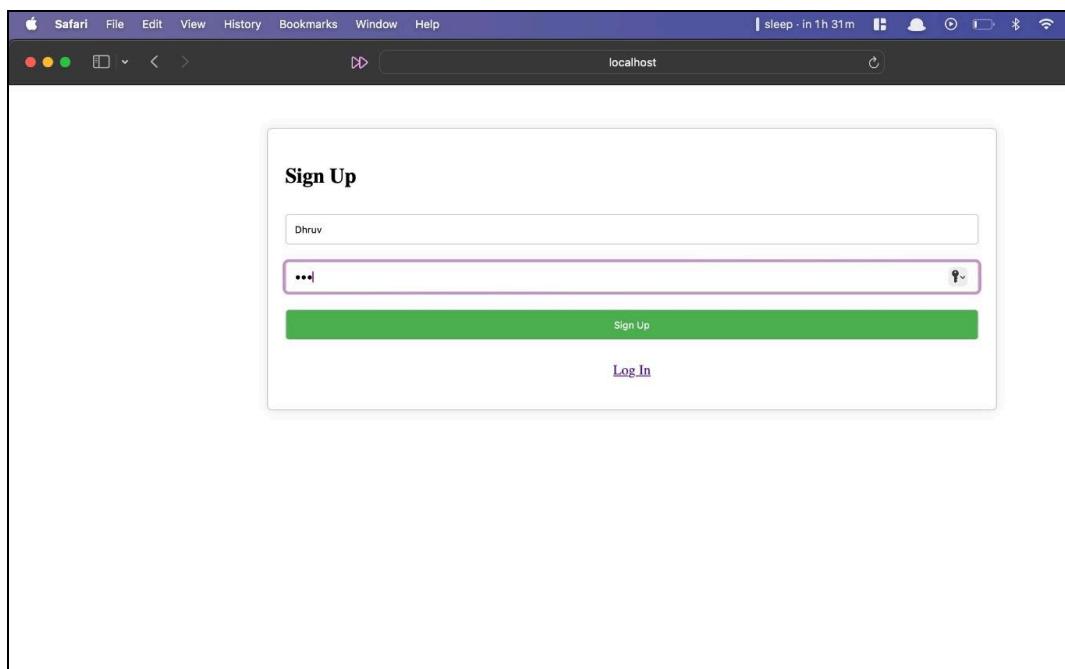
Button was added for Google authentication.



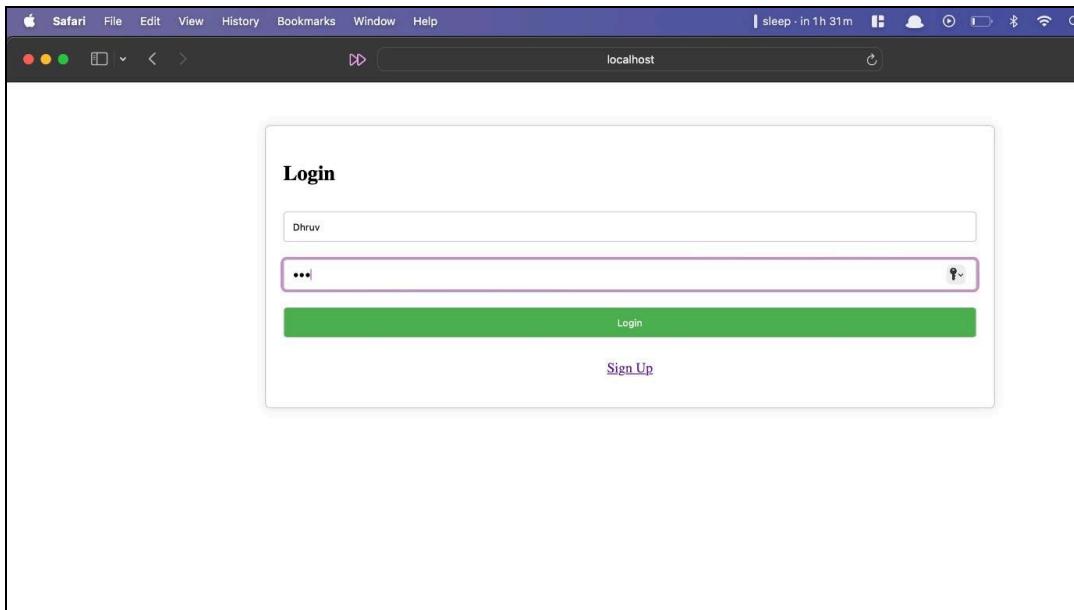
Pop-up After you click on button



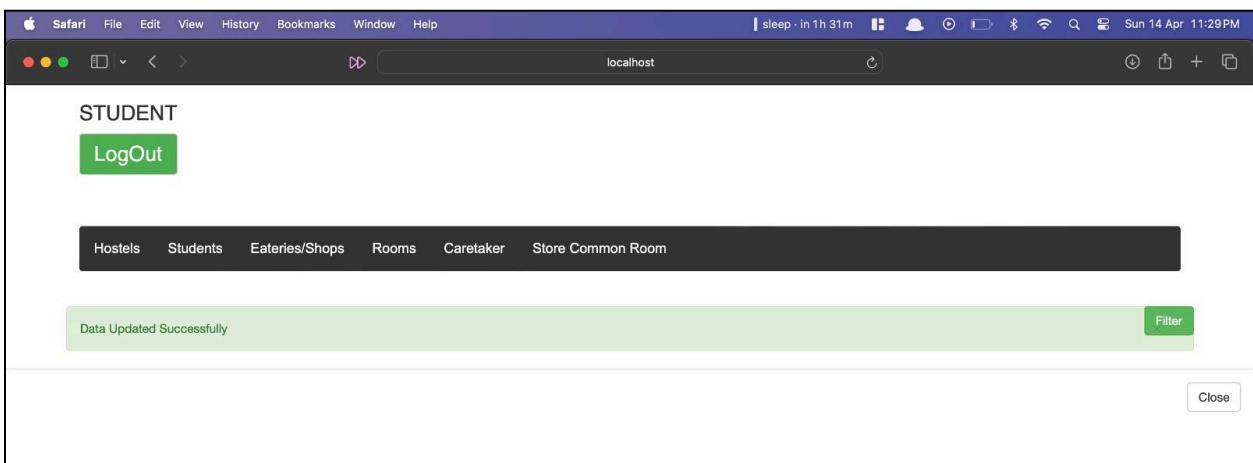
Already existing users log in using the IITGN email.



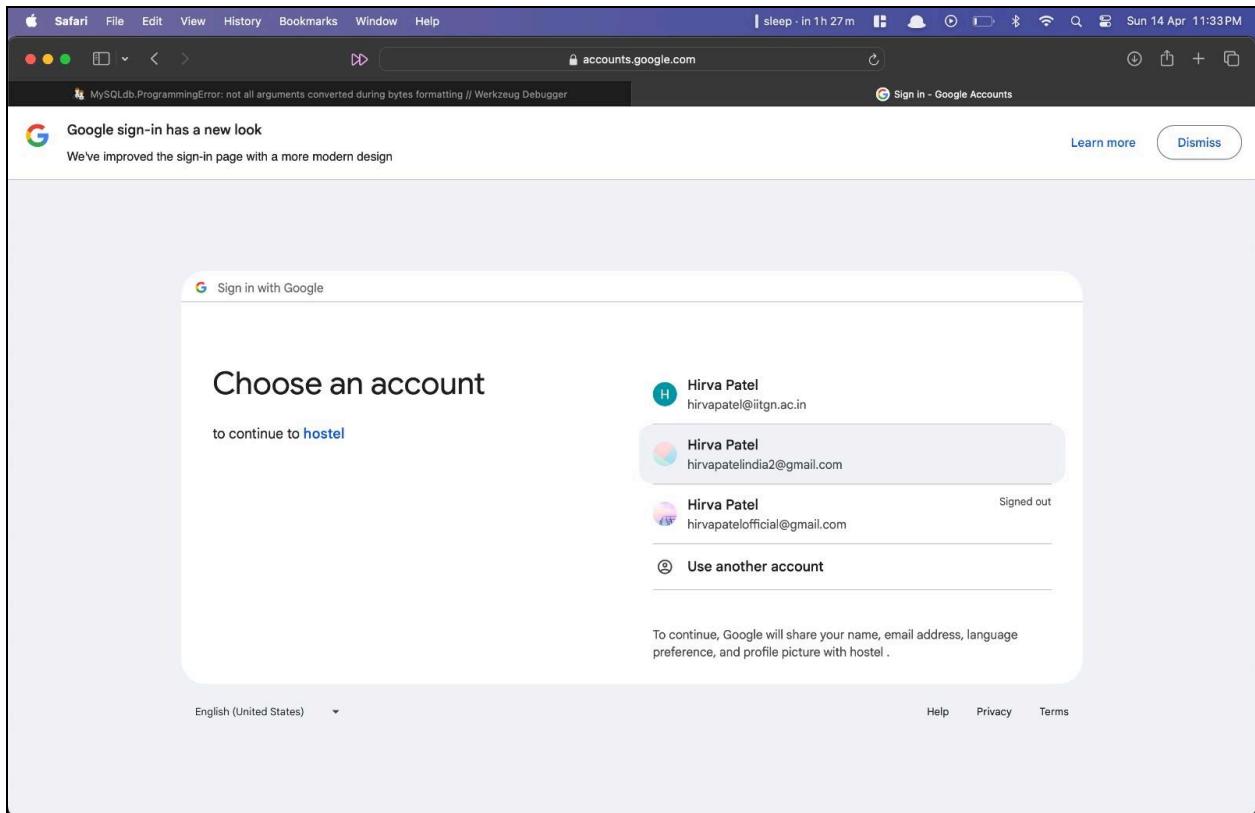
A new user signed in.



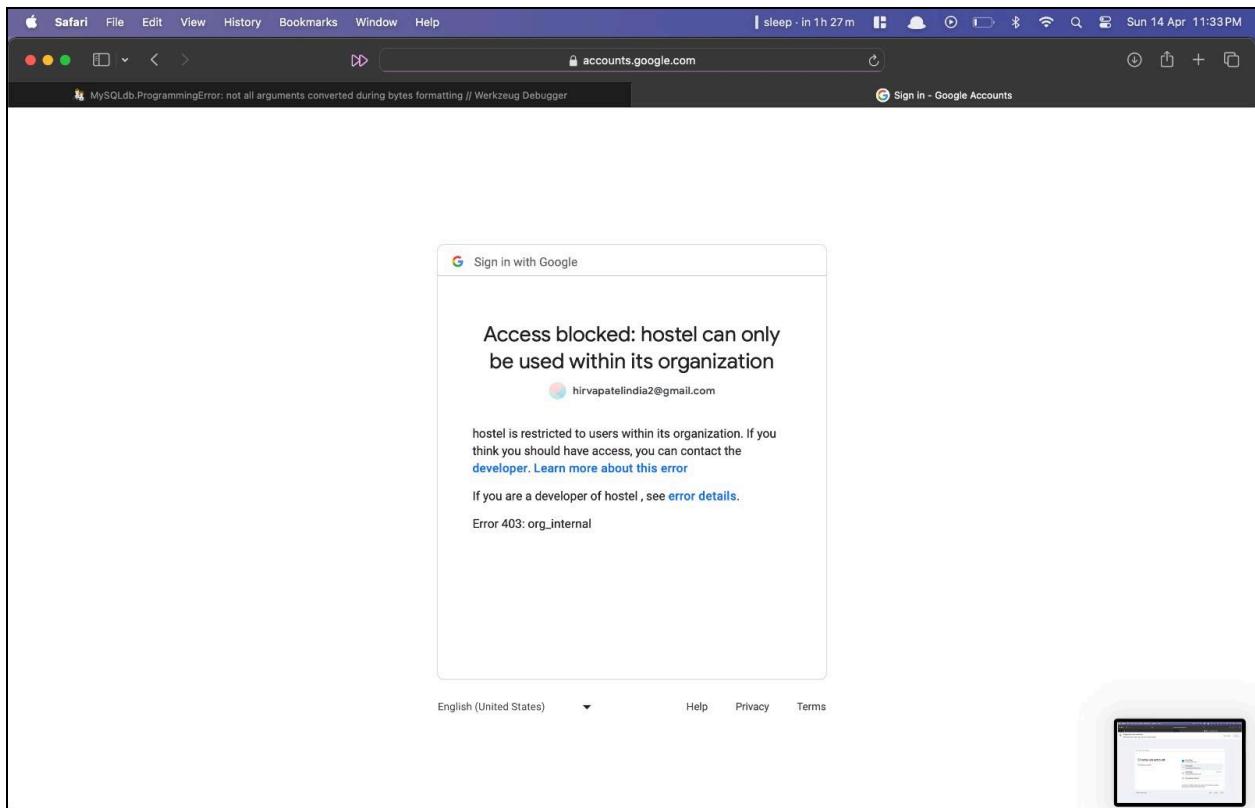
Logging in after signing up.



Data of the new user was inserted successfully.



Trying to log in use personal email ID.



Cannot log-in using personal email ID. (IIT GN's ID required)

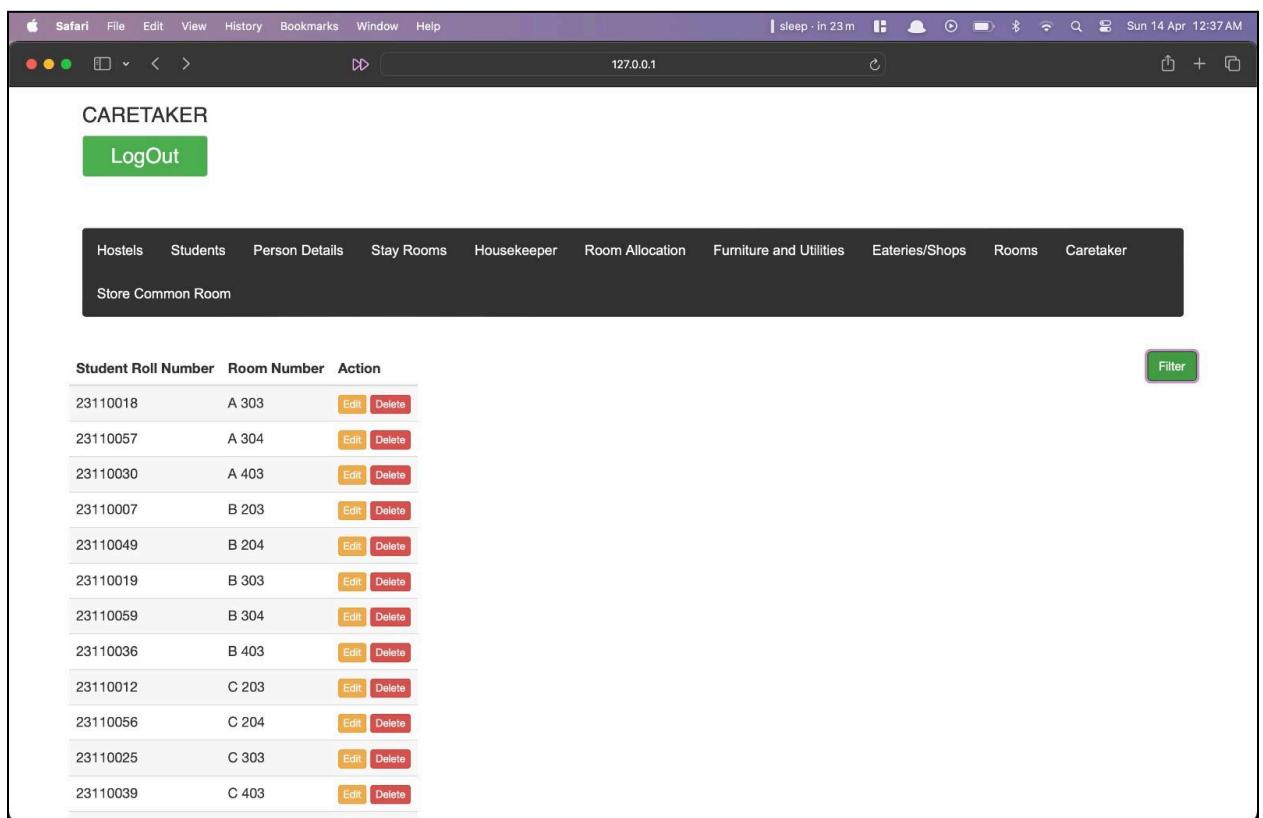
3.3 Responsibility of G1 & G2:

20 Pts

1. Documentation and screenshots of a total of 2 attacks [SQL Injection and XSS] performed and the defences against those attacks. (15 points)

We have made Four attacks in total.

SQL injection:



The screenshot shows a web application interface for 'CARETAKER' on a Mac OS X desktop. The browser title bar indicates the URL is 127.0.0.1. The main content area displays a table titled 'Room Allocation' under the 'Rooms' tab. The table has columns for 'Student Roll Number', 'Room Number', and 'Action'. Each row contains two entries: '23110018 A 303' and '23110057 A 304'. Each entry has 'Edit' and 'Delete' buttons. A 'Filter' button is located at the top right of the table. The table is set against a dark background with white text.

Student Roll Number	Room Number	Action
23110018	A 303	<button>Edit</button> <button>Delete</button>
23110057	A 304	<button>Edit</button> <button>Delete</button>
23110030	A 403	<button>Edit</button> <button>Delete</button>
23110007	B 203	<button>Edit</button> <button>Delete</button>
23110049	B 204	<button>Edit</button> <button>Delete</button>
23110019	B 303	<button>Edit</button> <button>Delete</button>
23110059	B 304	<button>Edit</button> <button>Delete</button>
23110036	B 403	<button>Edit</button> <button>Delete</button>
23110012	C 203	<button>Edit</button> <button>Delete</button>
23110056	C 204	<button>Edit</button> <button>Delete</button>
23110025	C 303	<button>Edit</button> <button>Delete</button>
23110039	C 403	<button>Edit</button> <button>Delete</button>

In the above image, we have opened the Room allocation table.

The screenshot shows a web application interface for 'CARETAKER'. At the top, there's a navigation bar with links for 'Hostels', 'Students', 'Person Details', 'Store Common Room', 'Series/Shops', 'Rooms', and 'Caretaker'. A 'Logout' button is also visible. Below the navigation, a table displays room allocations with columns for 'Student Roll Number', 'Room Number', and 'Action'. The table contains 12 rows of data. A modal dialog box is open, prompting the user to 'Enter the condition to filter the data' with a 'Condition:' input field containing the query: 'Student Roll Number = 23110018 UNION SELECT * FROM USERS'. There are 'Filter Data' and 'Close' buttons on the dialog. A 'Filter' button is also located at the bottom right of the main table area.

Student Roll Number	Room Number	Action
23110018	A 303	Edit Delete
23110057	A 304	Edit Delete
23110030	A 403	Edit Delete
23110007	B 203	Edit Delete
23110049	B 204	Edit Delete
23110019	B 303	Edit Delete
23110059	B 304	Edit Delete
23110036	B 403	Edit Delete
23110012	C 203	Edit Delete
23110056	C 204	Edit Delete
23110025	C 303	Edit Delete
23110039	C 403	Edit Delete

We open the filter section for the room allocation table and write a query to display the user table.

The screenshot shows a web application running in a Safari browser on a Mac. The title bar indicates the URL is 127.0.0.1. The main header is 'CARETAKER' with a 'LogOut' button. Below the header is a navigation menu with links: Hostels, Students, Person Details, Stay Rooms, Housekeeper, Room Allocation, Furniture and Utilities, Eateries/Shops, Rooms, and Caretaker. A sub-menu for 'Stay Rooms' is open, showing the option 'Store Common Room'. The main content area displays a table with the following data:

Student Roll Number	Room Number	Action
23110018	A 303	Edit Delete
admin	123	Edit Delete
caretaker	123	Edit Delete
guard	123	Edit Delete
guest	123	Edit Delete
house_keeping	123	Edit Delete
student	123	Edit Delete

At the top right of the table, there is a green 'Filter' button. In the bottom right corner of the main window, there is a small inset window showing a different part of the application's interface.

Before developing the defence mechanism, after filtering, we could also see usernames and passwords.

The screenshot shows a web application titled "CARETAKER" running in a Safari browser. The URL is 127.0.0.1. The main menu includes Hostels, Students, Person Details, Stay Rooms, Housekeeper, Room Allocation, Furniture and Utilities, Eateries/Shops, Rooms, and Caretaker. A sub-menu under "Rooms" shows "Store Common Room". A green banner at the top says "Can't process such queries". Below it is a table with columns "Student Roll Number", "Room Number", and "Action". The table contains the following data:

Student Roll Number	Room Number	Action
23110018	A 303	Edit Delete
23110057	A 304	Edit Delete
23110030	A 403	Edit Delete
23110007	B 203	Edit Delete
23110049	B 204	Edit Delete
23110019	B 303	Edit Delete
23110059	B 304	Edit Delete
23110036	B 403	Edit Delete
23110012	C 203	Edit Delete
23110056	C 204	Edit Delete
23110025	C 303	Edit Delete

Implementing the defence mechanism shows that such queries cannot be processed.

XSS:

The screenshot shows a web application interface titled "STUDENT". A modal dialog box is open, prompting the user to "Enter the condition to filter the data". Inside the dialog, there is a text input field containing the following malicious code:

```
Gender = "Female" <script> alert("hacked") </script>
```

Below the input field is a blue "Filter Data" button. In the background, the main page displays a table of student data with columns including Roll Number, First Name, Second Name, Last Name, Contact Number, Email ID, Date of Birth, Date of joining, Gender, Country, State, and City. The table contains six rows of sample data.

XSS attack is not valid in our case. All the conditions entered in filter boxes are directly used in the SQL Query statement and are not run or used in the HTML files.

Brute Force:

Someone might be able to log in by trying different passwords many times. To prevent this attack, we have implemented the website so that after three unsuccessful login trials, the time taken to log in will increase exponentially (Two power number of times).

```

if user:
    session['logged_in'] = True
    session['user_type'] = username
else:
    if 'attempts' not in session:
        session['attempts'] = 1
    else:
        session['attempts']+=1
    if session['attempts']>=3:
        time.sleep(2**session['attempts'])
        flash("Incorrect Credentials")
        session['logged_in'] = False
cur.close()
return redirect(url_for('Index'))

```

Path Traversal:

Here as we can see, when we try to input any path that the student does not have access to, it does not get followed up. Thus, our website is already defended.

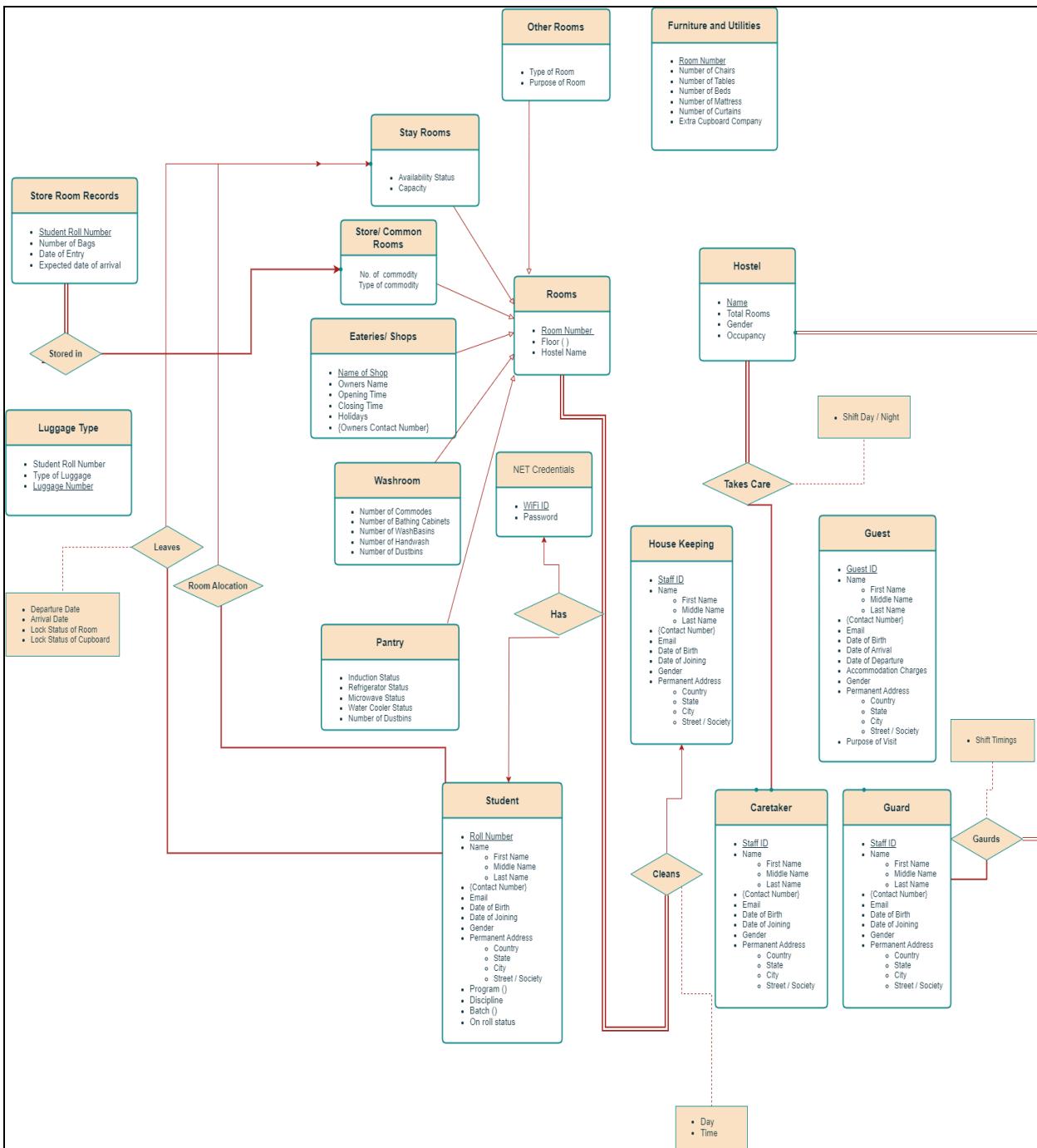
The screenshot shows a web browser window with the URL `127.0.0.1:5000/select/hostels`. The page has a sidebar on the left labeled "STUDENT" with a "Logout" button. Below the sidebar is a navigation bar with "Hostels", "Students", and "Person Details". A table lists student names, total rooms, gender, and occupancy. The table rows are as follows:

Name	Total Rooms	Gender	Occupancy
Aibaan	23	Female	2
Beauki	20	Female	5
Chimair	20	Female	5
Duven	20	Male	5
Emiet	20	Male	5
Firpeal	20	Male	5
Griwiksh	20	Male	5
Hiqom	20	Male	5
Ijokha	20	Female	5
Jurqia	20	Male	5
Kyzeel	20	Male	5
Lekhaag	20	Male	5

The main content area contains several sections: "Favourites" with icons for Apple, iCloud, Google, Yahoo, Bing, Wikipedia, Facebook, Twitter, LinkedIn, The Weather..., Zomato, and TripAdvisor; "Frequently Visited" with icons for ES335 - Schedule, CS328_Homework..., kdtree_simulation_s..., and L; and a "Privacy Report" section stating "In the last seven days, Safari has prevented 66 trackers". There is also a "Filter" button and a "Close" button at the bottom right.

The screenshot shows a web browser window with the URL `127.0.0.1`. The page displays an error message: "Method Not Allowed" and "The method is not allowed for the requested URL."

2 Show that all the relations and their constraints, finalized after the second feedback, are present and valid as per the ER diagram constructed in Assignment 1.



After the two feedbacks, all the relations of the tables on our website are valid as per the updated ER diagram in assignment 2.

D. Contributions

Name	Contribution
Divyanshu Pandey	-Took the hostel office's feedback and suggested website changes accordingly.
Hirva Patel	Worked on attack and defence system and created 4 such attacks with its defence mechanism. Created the signup page and helped implement google authentication.
Harshi Chandrafari	Worked on the website's frontend and made changes after the first and second feedback. Contributed in the write-up of the document.
Disha Chopra	-Worked on implementing the Google authentication system and rectified errors and bugs throughout.
Siddhi Rajpurohit	-Helped make changes in the front end by adding tables and norms. Contributed in the final documentation. ‘
Shubham More	-Contributed to the final documentation and helped with the feedback.
Dhruv Gupta	-Worked on the locking mechanism and helped implement changes after receiving feedback.