UB Study Hall Reservation System
Final Report

The "UB Study Hall Reservation System" is a dynamic digital platform developed to revolutionize the traditional classroom reservation process within the University at Bridgeport(UB). This project aims to develop an automated and user-friendly interface. The system caters to both administrators and students, providing an efficient and streamlined approach to managing study hall reservations. Moreover, the system introduces additional features such as an online library, PC rentals, a community-driven chat forum, study hall mapping, job opportunity submissions, a store for purchasing various items, a donation platform, and a rewarding points system. Leveraging MySQL as the database management system and Python as the primary programming language, the system is hosted on the Google Cloud Platform (GCP) to ensure scalability, reliability, and accessibility to a broad user base.

by

Yagnam Venisri - 1174310
Lalith S Lankalapalli - 1215621
Abhishek Dyawarishetti - 1211926
Brandon Galvan - 1107863

Step 1: Project Team

1. Yagnam Venisri
2. Lalith S Lankalapalli
3. Abhishek Dyawarishetti
4. Brandon Galvan

Step 2:

1. **Problem Statement:** Ub Study Hall Reservation System - The "UB Study Hall Reservation System" aims to provide an efficient and user-friendly platform for managing study room reservations within the University at Bridgeport (UB). This is an integrated and automated "UB Study Hall Reservation System" that not only streamlines the reservation process but also offers additional services such as an online library, PC rental, community forums, a store, and a donation platform. These functionalities aim to enrich the academic and social aspects of the study hall, providing students with a comprehensive and user-friendly platform for their various needs.

    a. User Types:
        i. The system will cater to two main types of users: Admin and Normal users.
        ii. Admin users will have full system management capabilities and access to reports.
        iii. Normal users can reserve, modify, and cancel classroom reservations.
    b. Authentication and Registration:
        i. Normal users will be able to register in the system, and subsequently, sign in to access the reservation features.
    c. Reservation Functionality:
        i. Normal users can book, modify, and cancel classroom reservations based on availability.

    d. Feedback and Rating:
        i. Normal users can provide feedback and ratings regarding the service and quality of their reservation experience.
    e. Information Visibility:
        i. All users can view the current, past, and future status of classroom reservations to make informed decisions.
    f. Communication:
        i. Normal users can communicate with Admin users by sending messages regarding any issues related to their reservations.
    g. Additional Functionalities:

i. Online Library:
    1. Normal users can browse and access books online through the website.
ii. Rent a PC:
    1. Normal users can rent a PC for a specific duration.
iii. Chat Forum:
    1. Normal users can post questions in the forum and receive replies from the community.
iv. Map:
    1. Normal users can see a picture where they can see the map/blueprint of the Study Hall.
v. Careers:
    1. Normal users can submit their resumes for positions within the study hall.
vi. Store:
    1. Normal users can access a small store within the study hall to purchase stationery items, coffee mugs, t-shirts, etc.
vii. Points:
    1. Normal users can earn points every time they book a hall and use them to purchase items from the store.
viii. Donate:
    1. Normal users can contribute by donating items or funds that help fellow students.

ix. Forgot Password:
    1. Users can reset passwords using this feature.
x. FAQ:
    1. Users can find answers to commonly asked questions.
xi. Contact Us:
    1. Normal users can contact via email or office phone number during office hours.
xii. About Us:
    1. Normal users can explore the inspiration behind the features of the study hall and gather more information about it.

Step 3: Queries

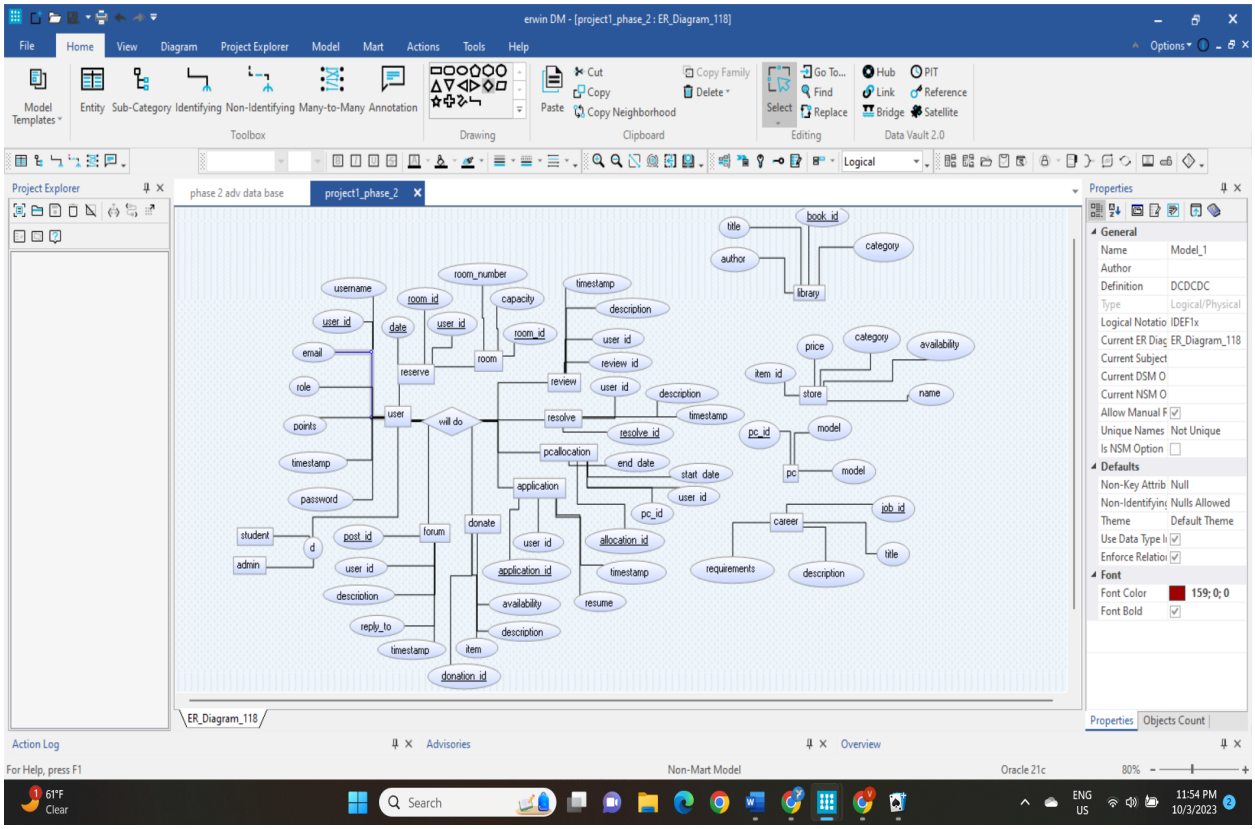
1. Retrieve a list of available rooms for a specific date.
2. Get all the available books in the online library.
3. Check the availability of PCs for rent during a specified duration.
4. Retrieve forum posts in a chronological order and their respective replies.

5. List out all available job positions within the study hall.
6. Retrieve a list of items available for purchase in the study hall store.
7. Retrieve the current points balance for a specific user.
8. Retrieve a list of donation items available for contribution.
9. Retrieve a list of past reservations for a specific user.
10. Retrieve a list of frequently asked questions and their answers.
11. Retrieve contact details (email, phone) for reaching out to the administration.
12. Retrieve detailed information about the study hall and its features.
13. Retrieve all the user and reservation information for the admin
14. Retrieve all the complaints for the admin to resolve.
15. Retrieve the room information such as how many people can accommodate, ac, fan etc.,
16. Retrieve all the allocated and unallocated pcs for admin.
17. Retrieve all the previous reservations for a user.

Step 4: Database Draft

1. User: user_id(primary key), username, password, email, role, points, timestamp.
2. Room: room_id(primary key), room_number, capacity, location.
3. Reserve: room_id(partial primary key), user_id(partial primary key, foreign key), date(partial primary key), timestamp.
4. Reviews: review_id(primary key), username, description, rating, timestamp.
5. Resolve: resolve_id(primary key), username, description, timestamp.
6. Library: book_id(primary key), title, author, category.
7. PC: pc_id(primary key), model, availability.
8. Forum: post_id (primary key), user_id (foreign key), description, replies, timestamp.
9. Store: item_id (primary key), name, price, category, availability.
10. Career: job_id (primary key), title, description, requirements.
11. Donate: donation_id (primary key), item_name, description, availability.
12. Application: application_id(primary key), user_id, resume, timestamp.
13. PCAllocation: allocation_id(primary key), pc_id(foreign key), user_id(foreign key), duration, start_timestamp, end_timestamp.

Step 5: ER Model



Step 6: Relational Database Schema

User

user_id	username	email	password	role	points	timestamp

Room

room_id	room_number	capacity	location

Reserve

room_id	user_id	date	timestamp

Review

review_id	user_id	description	rating	timestamp

Resolve

resolve_id	user_id	description	timestamp

Library

book_id	title	author	category

PC

pc_id	model	availability

Forum

post_id	user_id	description	reply_to	timestamp

Store

item_id	name	price	category	availability

Career

job_id	title	description	requirements

Donate

donation_id	item	description	availability

Application

application_id	user_id	resume	timestamp

PCAllocation

allocation_id	pc_id	user_id	start_date	end_date	timestamp

Step 7: Relational Algebra Expressions

1. Retrieve a list of available rooms for a specific date.
    a. AvailableRooms(date) = π room_id (σ date = ‘specific_date’ (Reserve))
2. Get all the available books in the online library.
    a. AvailableBooks = π title (Library)

3. Check the availability of PCs for rent during a specified duration.
    a. AvailablePCs(startDate, endDate) = π model, quantity (σ start_date >= 'specified_start_date' AND end_date <= 'specified_end_date' (PC))
4. Retrieve forum posts in a chronological order and their respective replies.
    a. AllForumPostsAndReplies = π username, description, timestamp, replies (ForumPost) ⋈ (ForumPost.post_id = ForumPost.reply_to) ρ (post_id, description, timestamp, replies)(ForumPost)

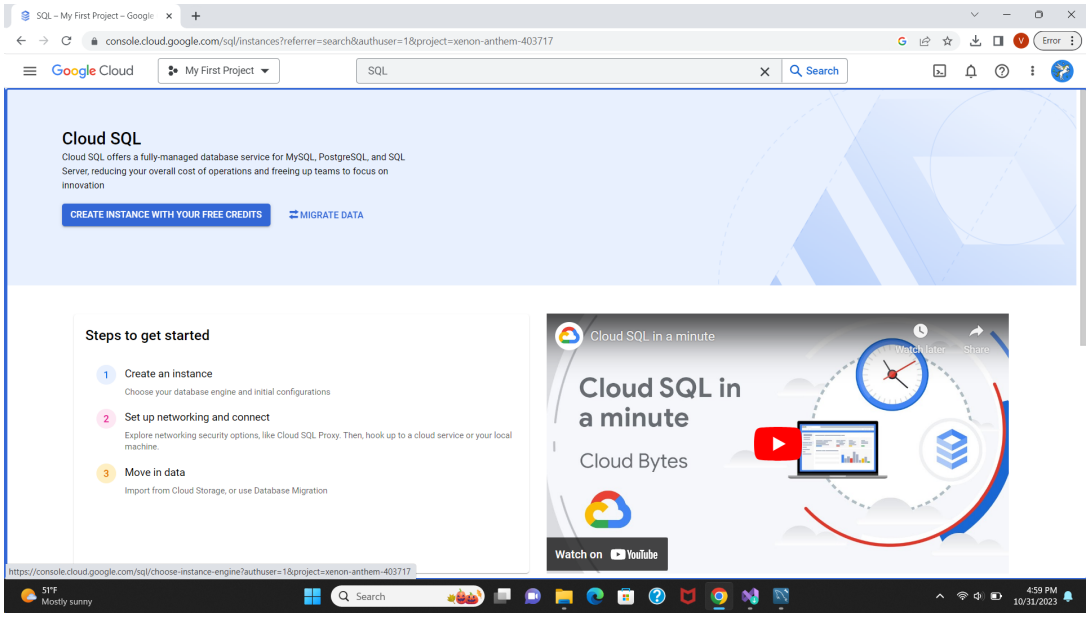
5. List out all available job positions within the study hall.
    a. AvailableJobPositions = π position, description, requirements (Career)
6. Retrieve a list of items available for purchase in the study hall store.
    a. AvailableStoreItems = π name, price, category (Store)
7. Retrieve the current points balance for a specific user.
    a. CurrentPoints(user) = π points (σ user_id = user) (User)
8. Retrieve a list of donation items available for contribution.
    a. DonatedItems = π item_name, description (Donate)
9. Retrieve a list of past reservations for a specific user.
    a. PastReservations(user) = π room_number, date (σ user_id = user) (Reserve)
10. Retrieve contact details (email, phone) for reaching out to the administration.
    a. AdminContactDetails = π email, phone (Admin)
11. Retrieve detailed information about the study hall and its features.
    a. StudyHallInformation = π * (StudyHall)
12. Retrieve all the user and reservation information for the admin
    a. Reservations = π * (Reserve)
13. Retrieve all the complaints for the admin to resolve.
    a. Complaints = π * (Resolve)
14. Retrieve the room information such as how many people can accommodate, ac, fan etc.,
    a. RoomInfo = π * (Room)
15. Retrieve all the allocated and unallocated pcs for admin.
    a. PCInfo = π * (PC)
16. Retrieve all the upcoming reservations for a user.
    a. Upcoming(user, date) = π room_number (σ user_id = user AND timestamp >= date) (Reserve)

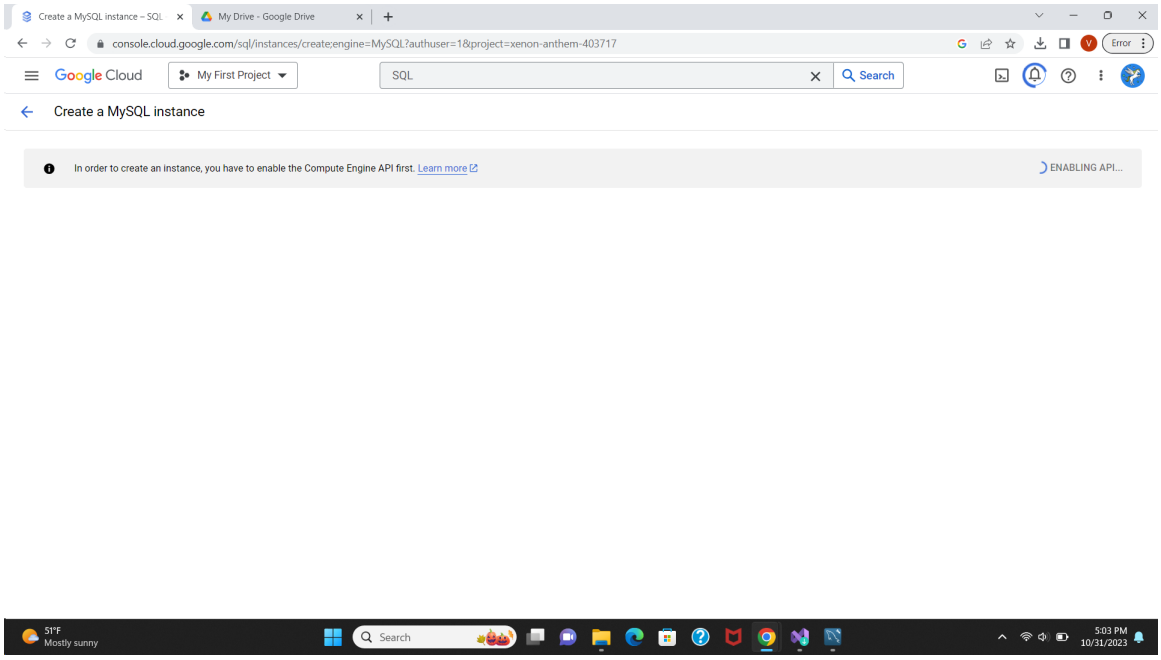
Step 8: Creating Database Server

To establish the Database Server, we followed the below steps

1. **Project Creation:** We initiated a GCP project. This project served as the central hub for our database and web application components.

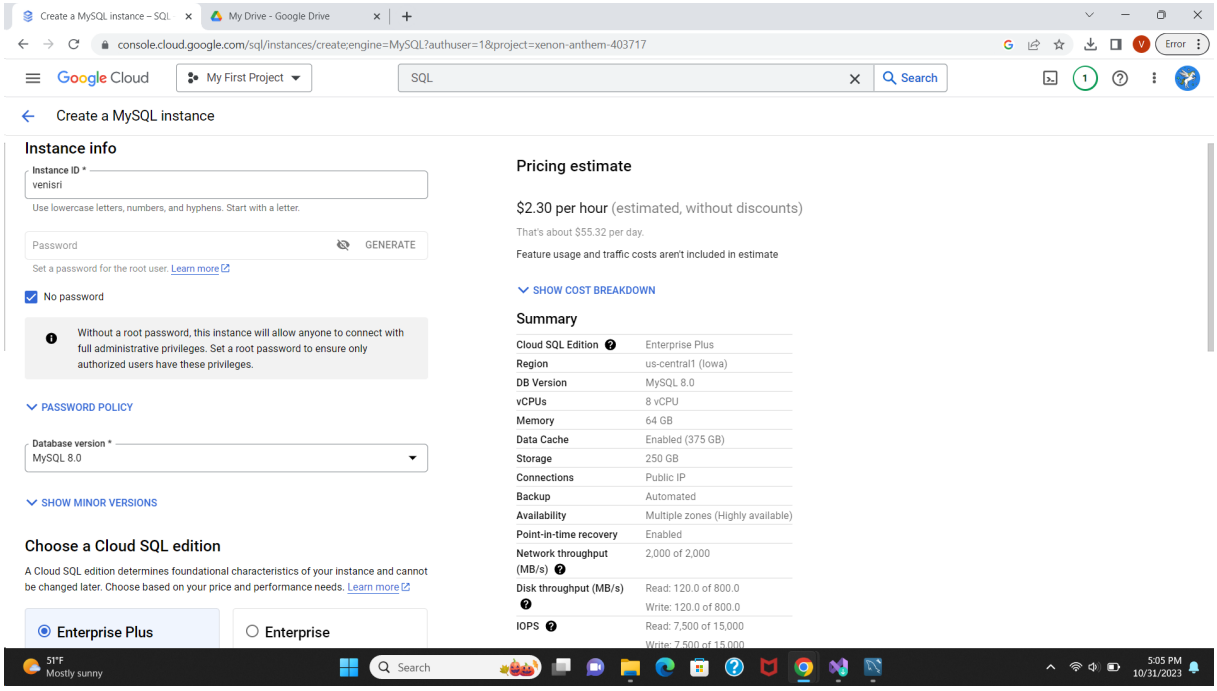
2. **Billing Account:** We assigned a billing account to the project based on a coupon provided in the exercise lab. This is essential to manage the costs associated with running resources on GCP.

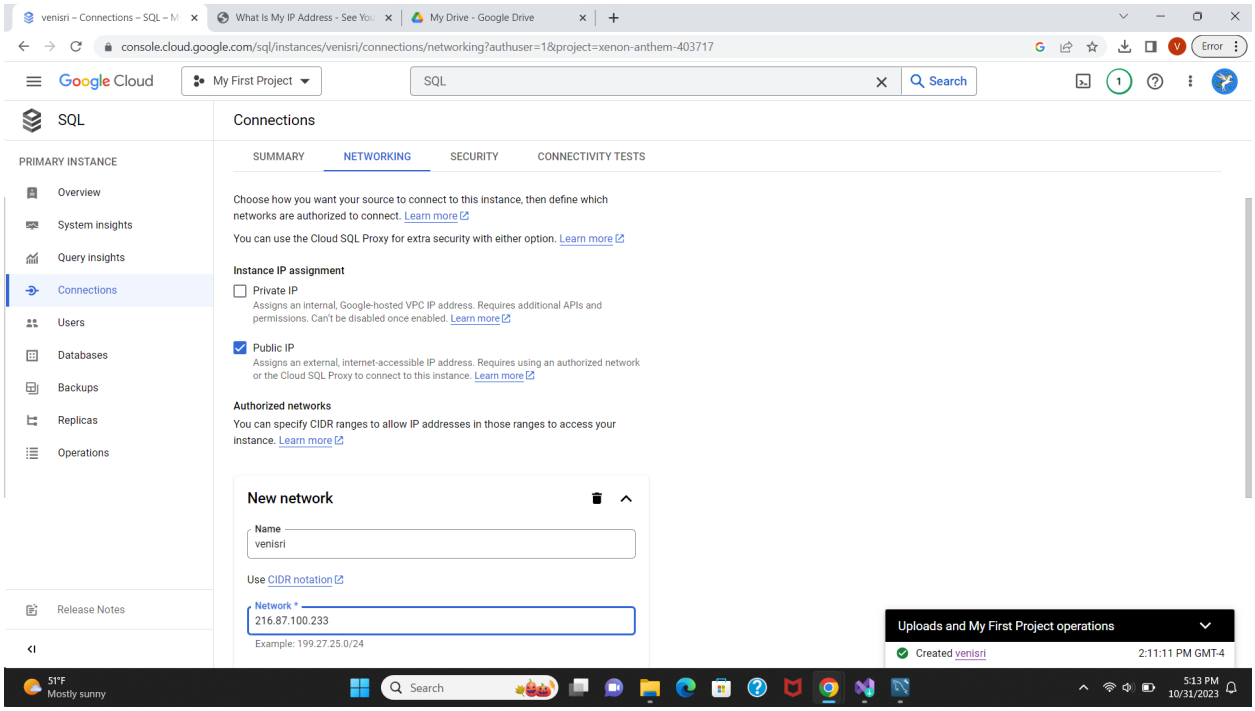


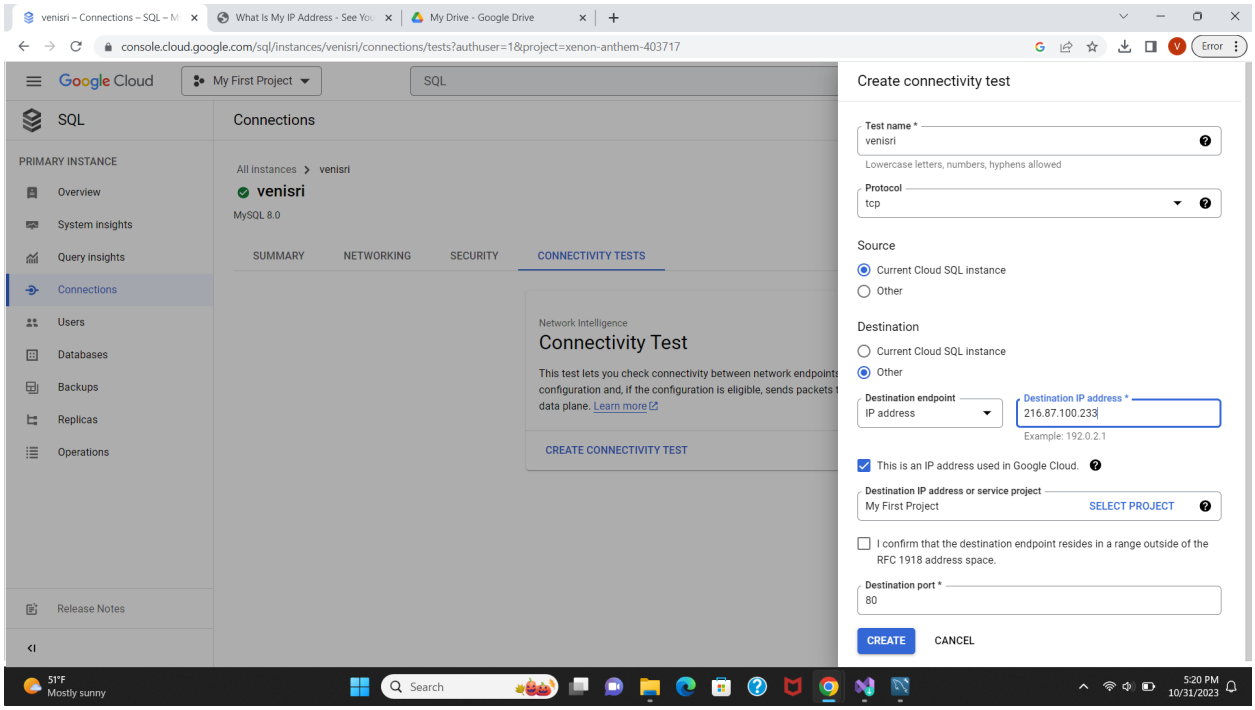


3. **RDBMS Selection and Installation:** We selected MySQL as our relational database management system (RDBMS) and installed it on a Google Compute Engine

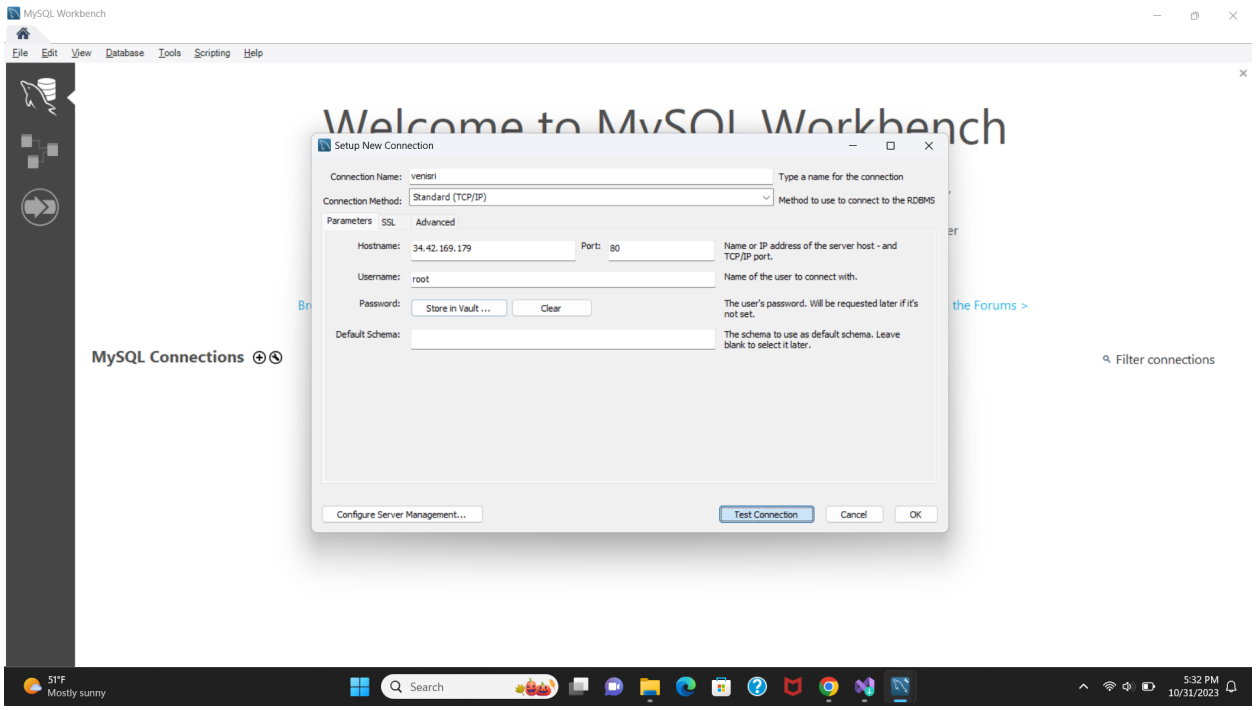
instance. This choice was made due to MySQL's reliability, scalability, and compatibility with various web frameworks.

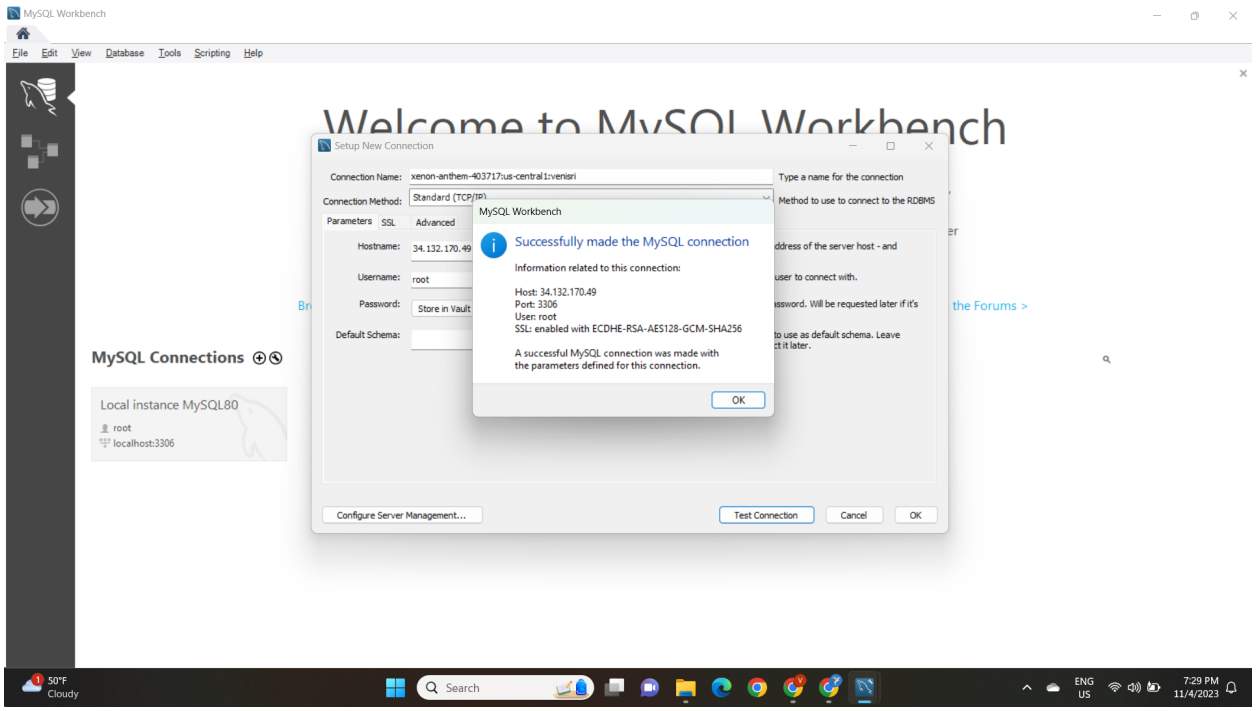






4. **Database Connectivity using MySQL Workbench:** In addition to setting up the database server on Google Compute Engine, we leveraged MySQL Workbench, a powerful database management tool, to facilitate the development and management of our MySQL database instance. This step enhanced our ability to interact with the database for various purposes, including schema design, SQL query development, and database administration.
5. **Connection to the GCP-hosted MySQL Database Instance:** We established a connection between MySQL Workbench and our MySQL database instance on GCP. To achieve this, we used the following steps:
    a. **Hostname Configuration:** We obtained the hostname or IP address associated with our MySQL database instance on Google Compute Engine within the GCP project.
    b. **MySQL Workbench Configuration:** In MySQL Workbench, we configured a new database connection, providing the GCP-hosted MySQL database instance's hostname.
    c. **Testing and Usage:** With the successful connection, we were able to utilize MySQL Workbench to perform tasks such as database schema design, executing SQL queries, and managing the database server directly from our local development environment.

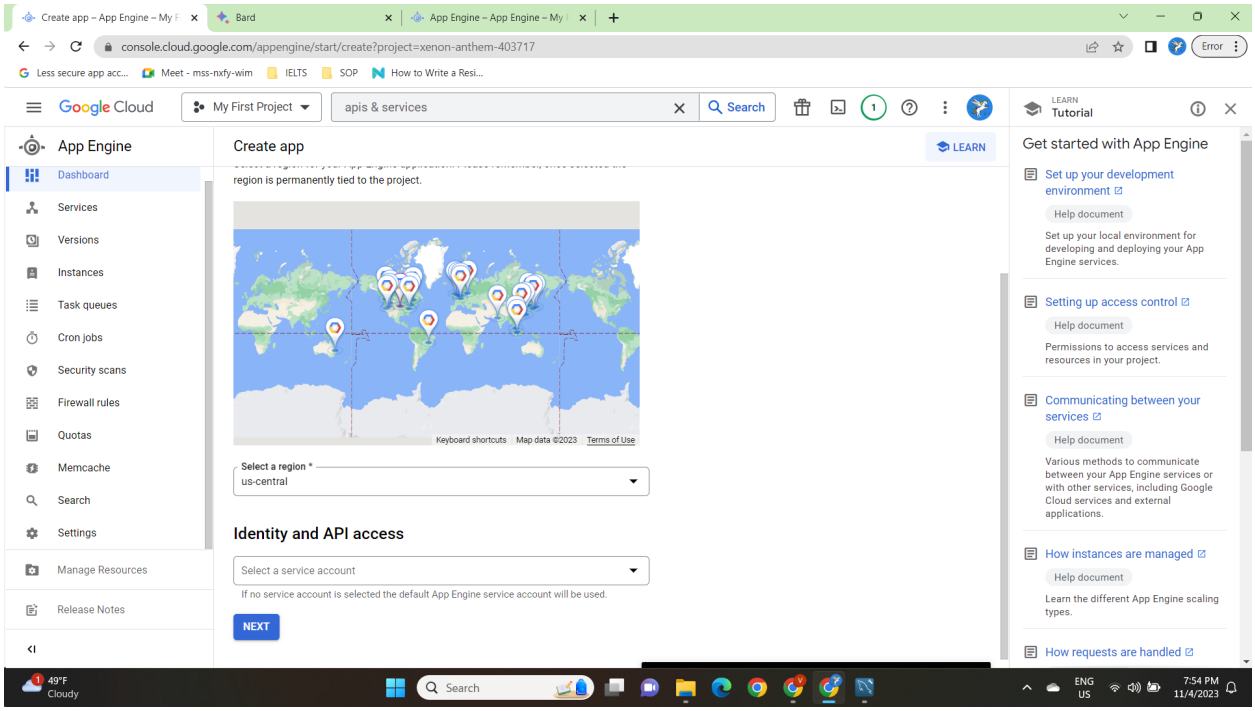




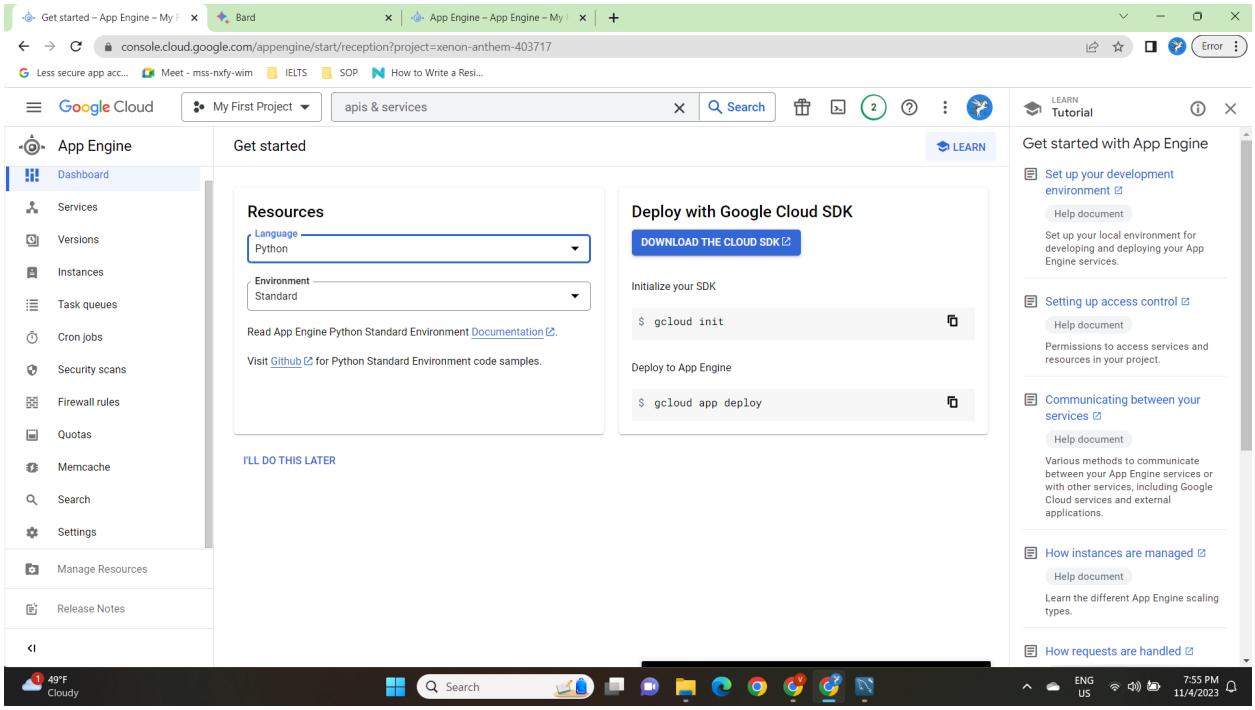
This integration between MySQL Workbench and our GCP-hosted MySQL database instance allowed us to streamline our development process, making it more efficient and enabling real-time database interaction for the development of the UB Study Hall Reservation System.

Step 9: Creating a Web Application Server

1. **Google App Engine Setup:** We created a Google App Engine instance to host our web application. Google App Engine provides a scalable and serverless platform for running web applications.



2. **Framework Selection:** We selected the Django framework for our web application. Django is a popular and robust web framework that simplifies the development process and provides a wide range of tools and libraries for creating web applications.



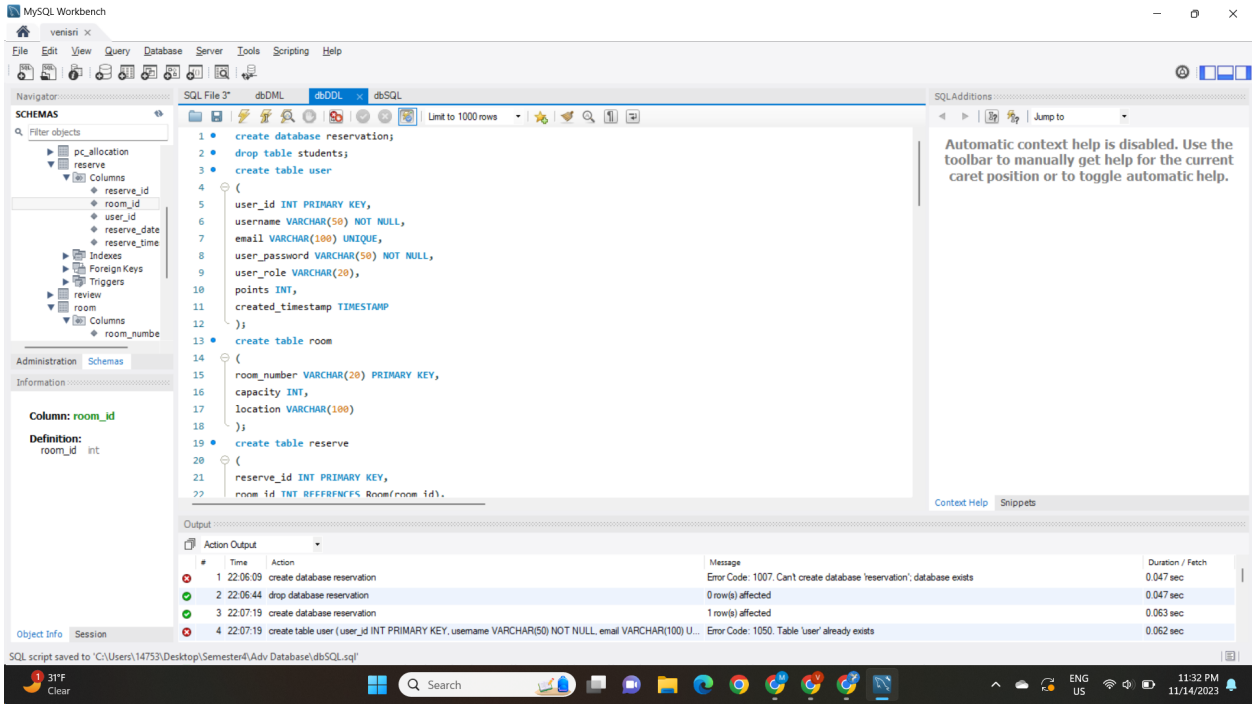
3. **Application Deployment:** We are going to deploy our room reservation system using Django on Google App Engine. This allows us to create a dynamic and user-friendly web application for users to interact with the database.

Step 9: Database Creation

- Created a database named reservation.
- Defined tables to model various entities: user, room, reserve, review, complaints, library, library_history, pc, pc_allocation, forum, store, career, donate, application.

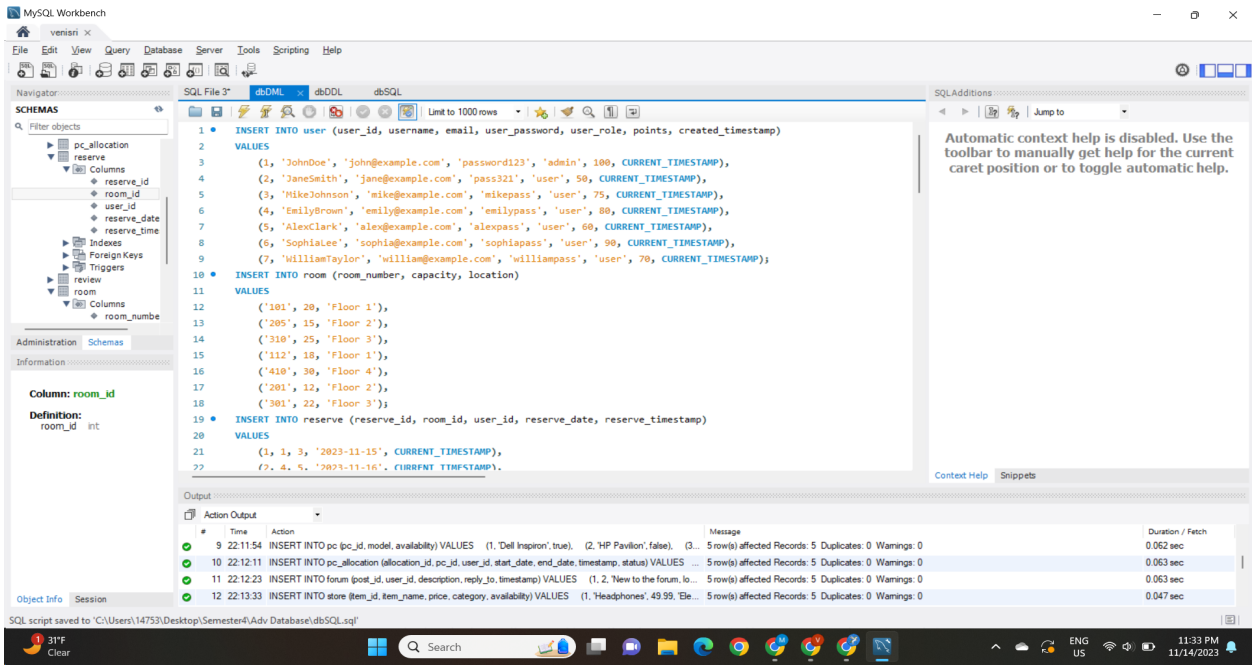
Step 10: DDL Script

- Produced a DDL script named dbDDL.sql that encompasses the entire database schema including tables with constraints, indexes, etc.



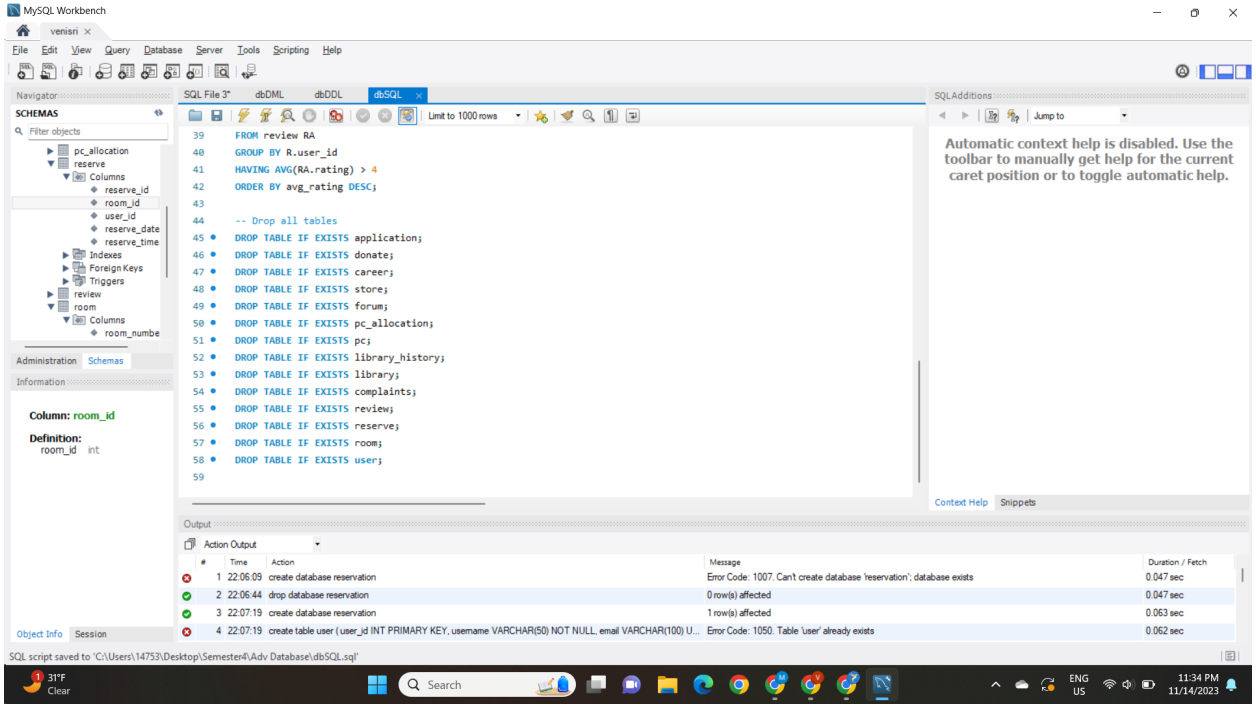
Step 11: DML Script

- Created a DML script named dbDML.sql to insert sample data into tables.



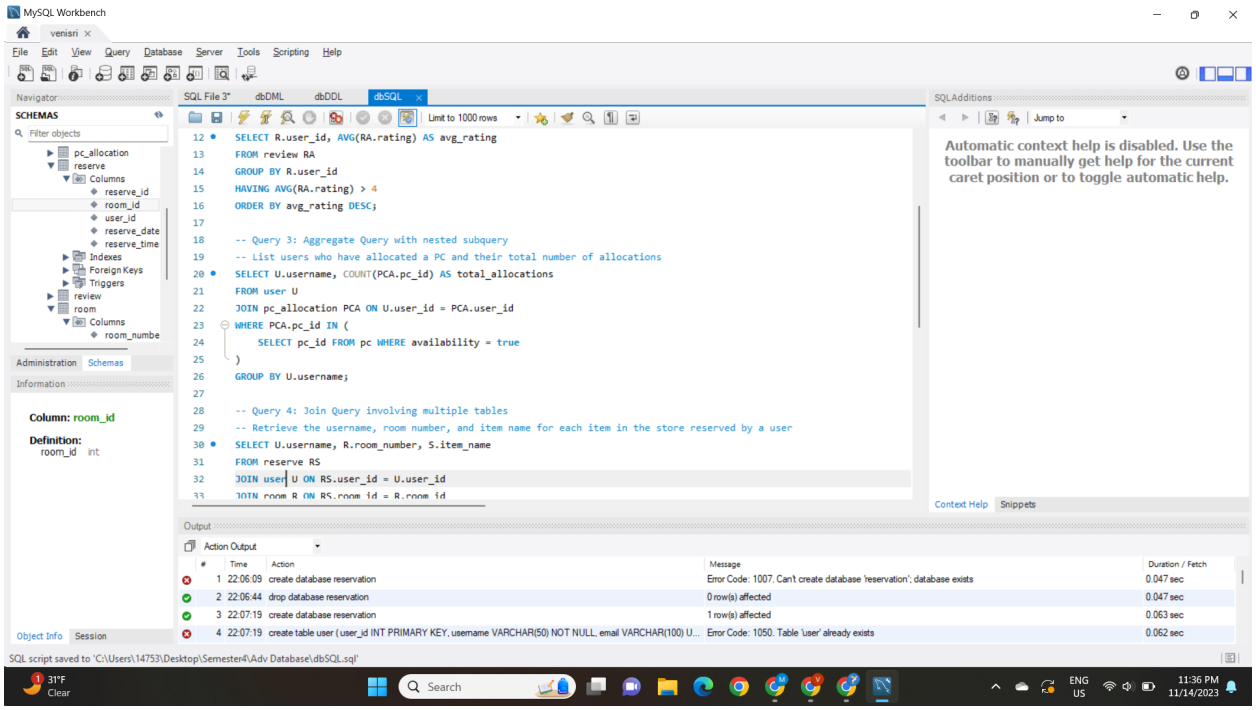
Step 12: Drop Script

- Formulated a drop script named dbDROP.sql to remove all database objects, facilitating a clean database state.

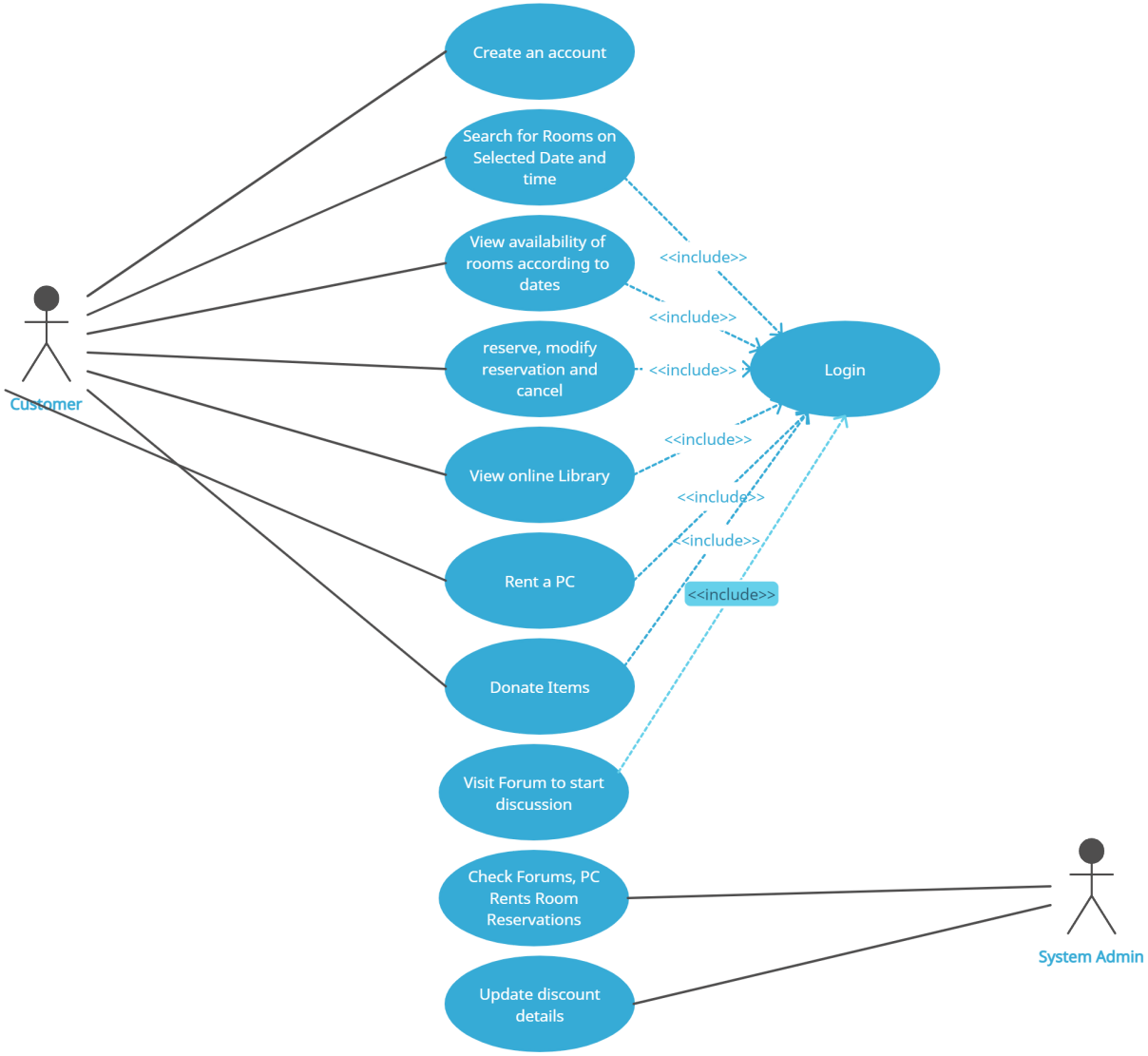


Step 13: SQL Script

- Generated an SQL script named dbSQL.sql with complex queries satisfying specific criteria:
- Joins involving multiple relations.
- Aggregate queries using GROUP BY, HAVING, and ORDER BY clauses.
- Utilized nested subqueries.



System configuration of the project:



Step 14: Application Development

In step 14, we concentrated on building a user-friendly front-end application that interfaces seamlessly with the backend database management system (DBMS) developed using MySQL. The application aimed to simplify the process of reserving study hall rooms for students.

Technology and Tools Utilized:

1. Backend Language: Python with Django Framework
2. Database Management System: MySQL Workbench
3. Frontend Interface: Developed using Django templates for HTML rendering
4. Integration Tools: Google Cloud Platform (GCP) for hosting the database

Functionalities Implemented:

1. User Authentication: Users can register, log in, and authenticate credentials securely.
2. User Registration: New user registration involves capturing essential information such as username, email, and password.
3. Reservation Management: Students can reserve study hall rooms for a specified time slot.
4. Points System: Upon successful reservation, users are awarded points, encouraging consistent use and engagement.
5. Feedback Submission: Allows users to submit messages along with their usernames and email addresses for improved communication and issue resolution.
6. Administrative Review: Enables administrators to access submitted feedback, review messages, and address user concerns or suggestions promptly.
7. Administrative Reservation Monitoring: Provides a comprehensive list of reservations, detailing room bookings, timestamps, and user information accessible to authorized administrators.
8. Reservation Modification: Enabled users to modify their reservations before the scheduled time through an intuitive user interface. Users have the flexibility to adjust reservation details like time, date, or room number based on availability and their needs.
9. Cancellation Feature: Implemented a cancellation option, allowing users to revoke their reservations within a specified window before the scheduled reservation time. Ensures users have the autonomy to manage their bookings, promoting a user-friendly experience.
10. User-Friendly Navigation: Integrated a navigation bar across all pages to enhance user experience and enable easy traversal between different sections of the application. Consistently accessible navigation across the interface for effortless movement between functionalities and pages.

Database Connectivity:

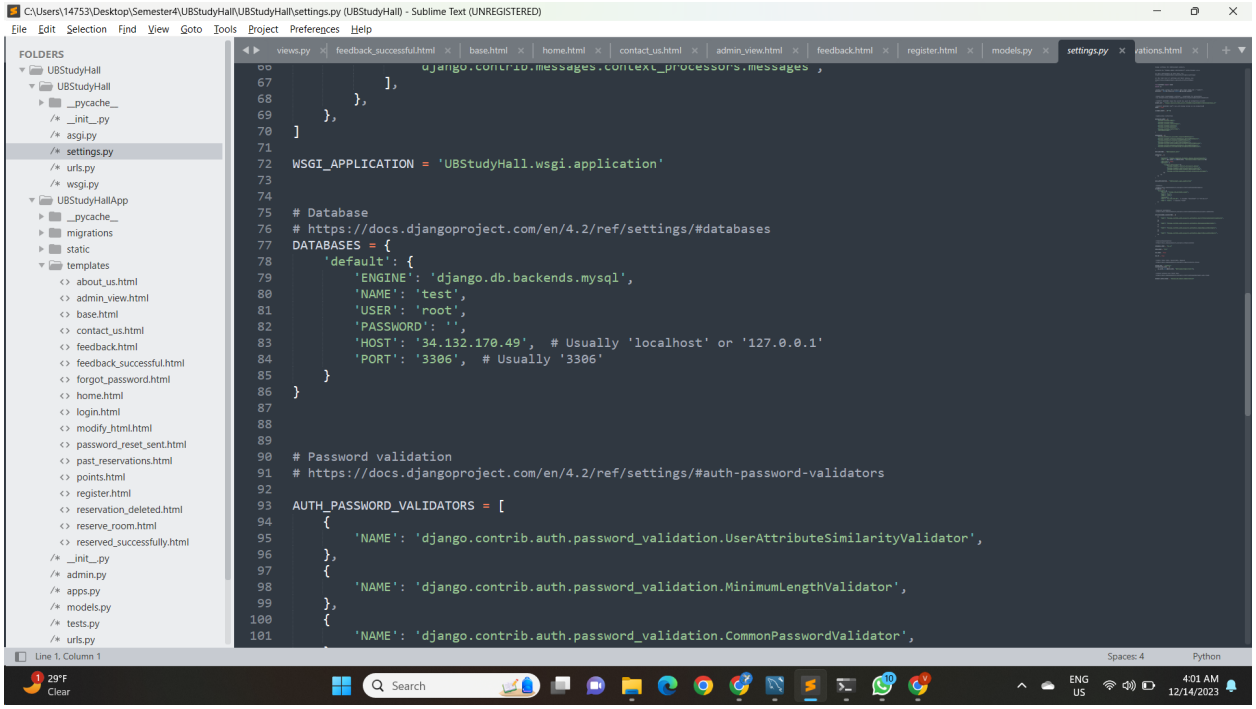
1. MySQL Workbench: Utilized for designing, creating, and managing the database schema.
2. Google Cloud Platform (GCP): The backend Django application was connected to the MySQL database hosted on GCP, ensuring secure and scalable access.

User Interface:

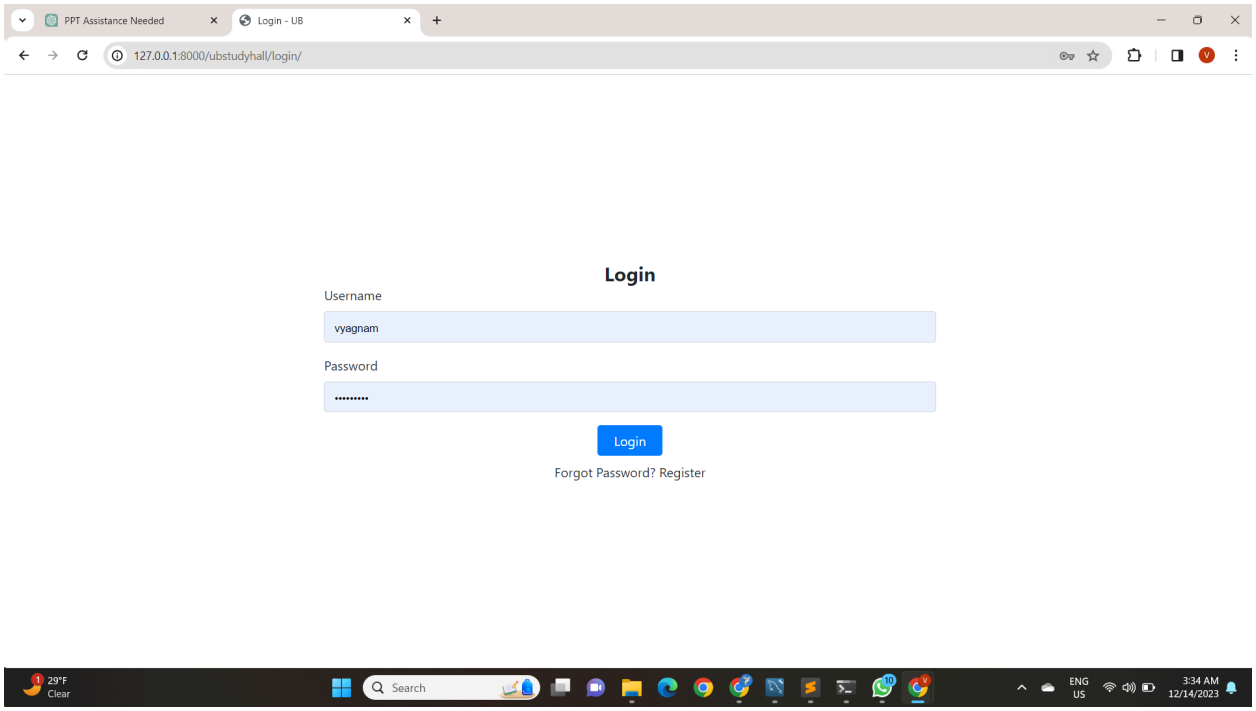
1. Graphical User Interface (GUI): Incorporated a simple, intuitive interface for users to navigate through various functionalities.
2. Menu-Driven Input: Designed a menu-driven approach for users to interact with the database without needing SQL queries, enhancing usability.

Results:

Database Connectivity:



Page 1: Login Page



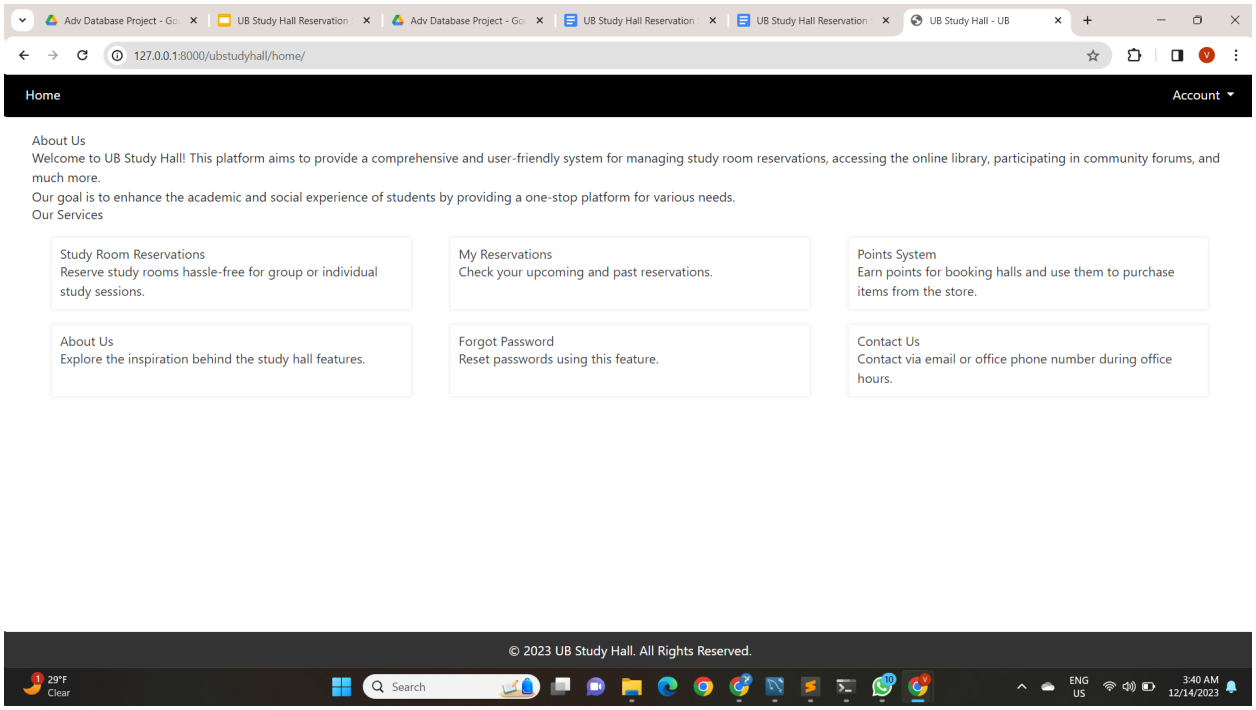
Users can log in using their credentials if they are already registered. The login form prompts for a username/email and password to access their accounts.

New users who don't have accounts can easily register by clicking on the registration link/button. This process involves providing essential details such as username, email, and password for creating a new account.

For users encountering login issues due to forgotten passwords, this feature provides a streamlined process to reset their password.

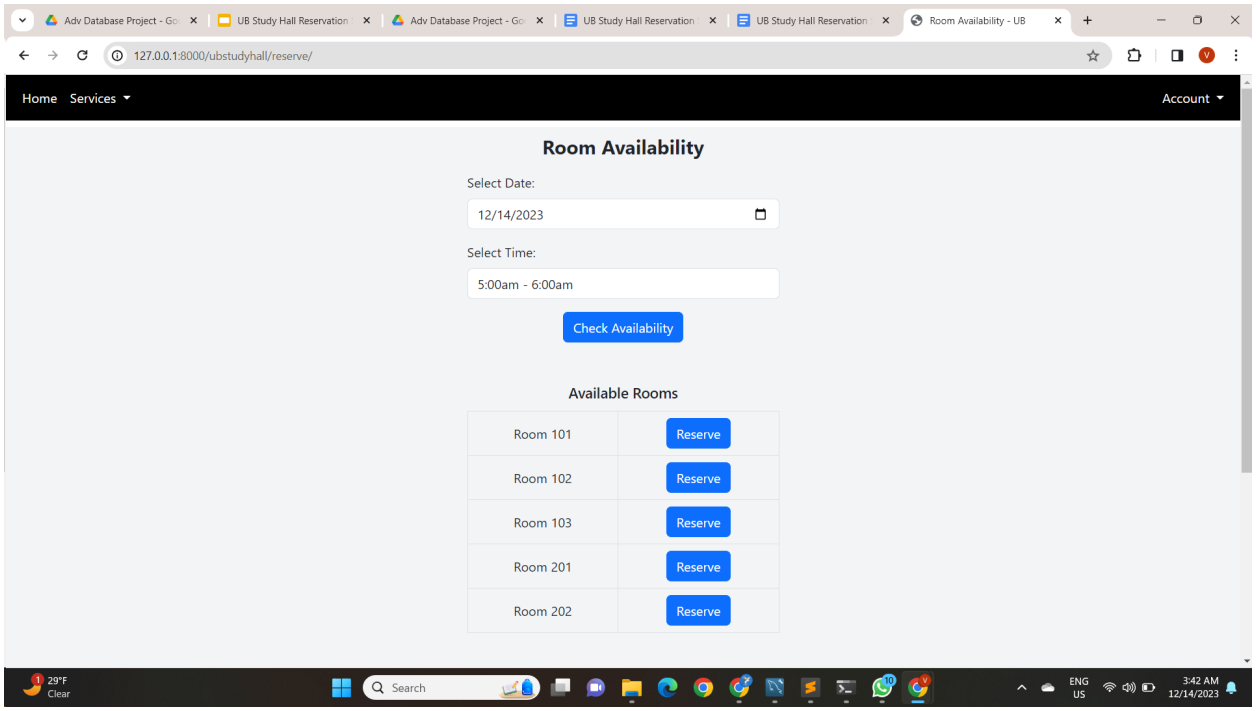
Page 2: Home Page - Resource Navigation

The Home Page serves as a central hub, offering users an intuitive and comprehensive navigation system to access various resources and functionalities within the application. It prominently showcases essential options that users can explore for managing their study hall experience.



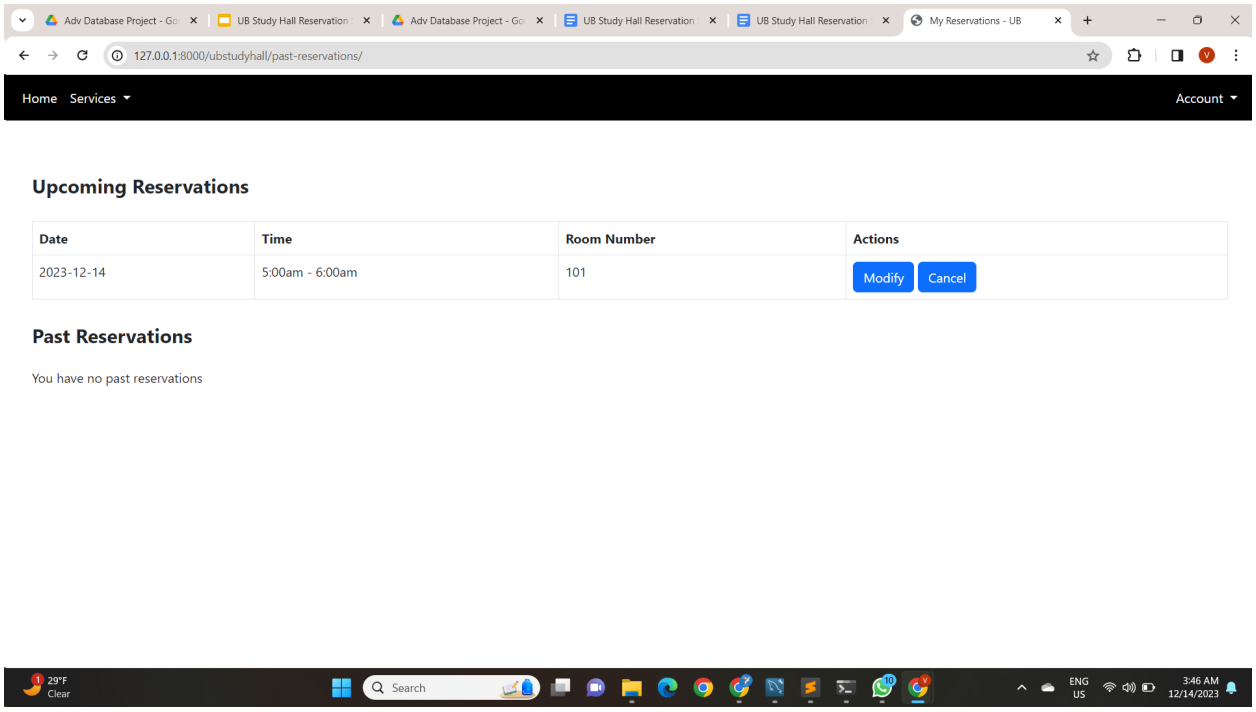
Page 3: Room Reservation - Time and Date Selection

The Room Reservation page empowers users to book study hall rooms based on their preferred date and time slots. It serves as a platform to explore room availability and make reservations according to individual study schedules.



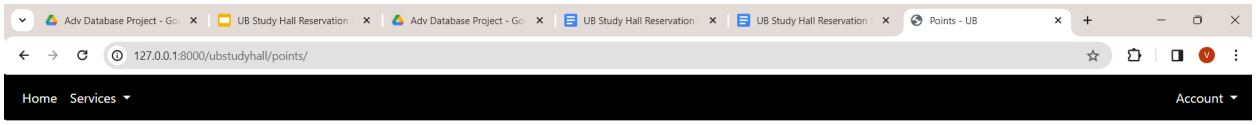
Page 4: Upcoming and Past Reservations - Modification and Cancellation

Page 4 serves as a centralized hub for users to manage their study hall reservations, both upcoming and past ones. It empowers users to view, modify, or cancel their upcoming reservations, providing a convenient platform for reservation management.



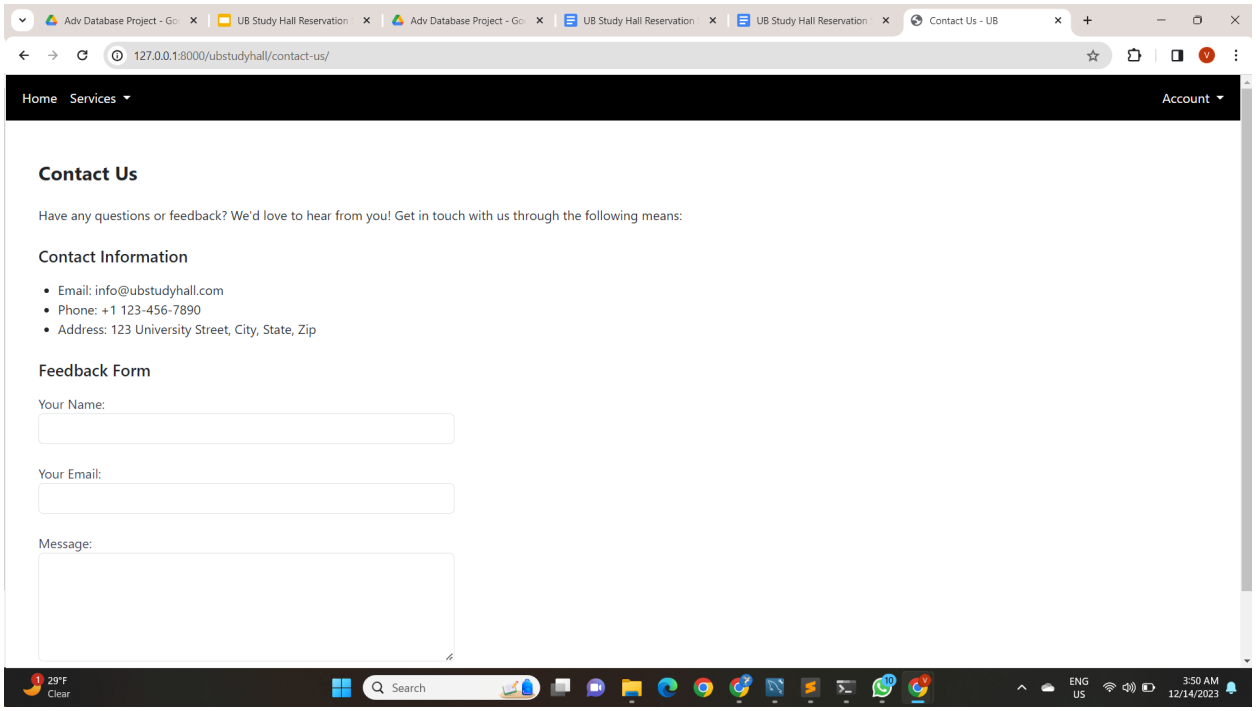
Page 6: Earning Points for Reservations

Page 6 is dedicated to the points system integrated into the study hall reservation platform. It allows users to earn points upon successful completion of room reservations, encouraging consistent engagement and rewarding users for their frequent usage.



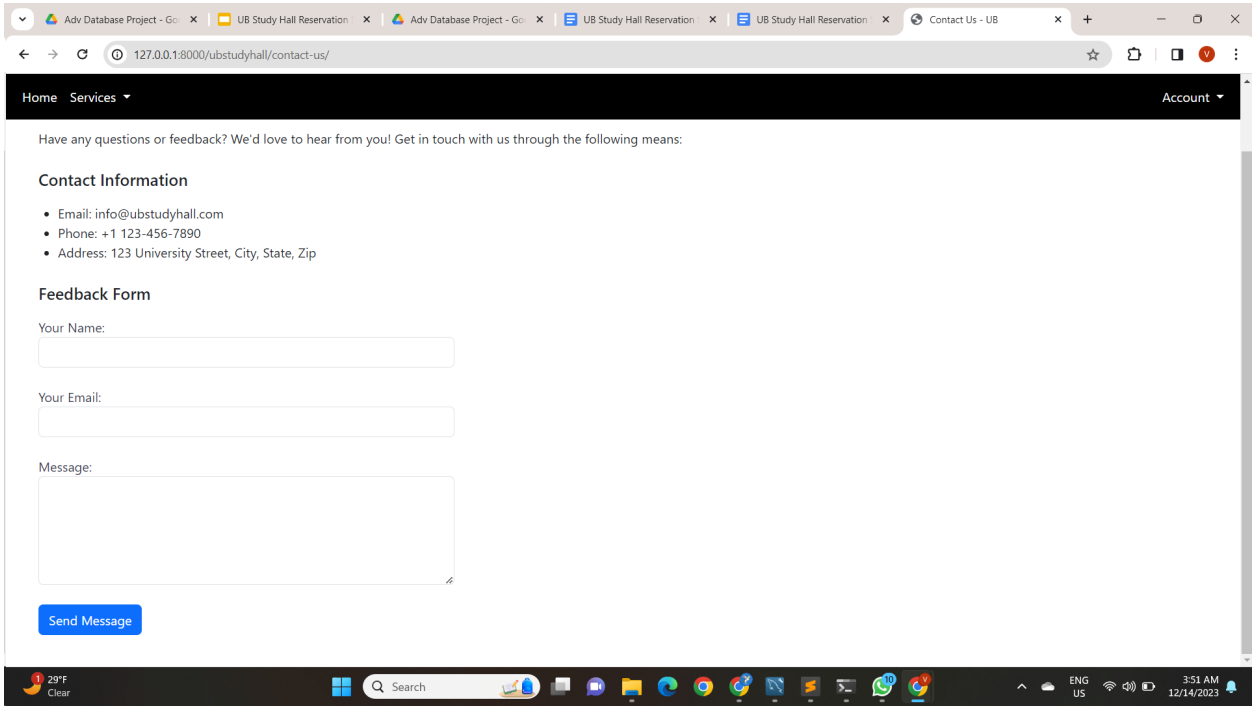
Page 7: Contact Us

The "Contact Us" page serves as a channel for users to reach out to the administration or support team for any inquiries, concerns, or assistance related to the study hall reservation system.



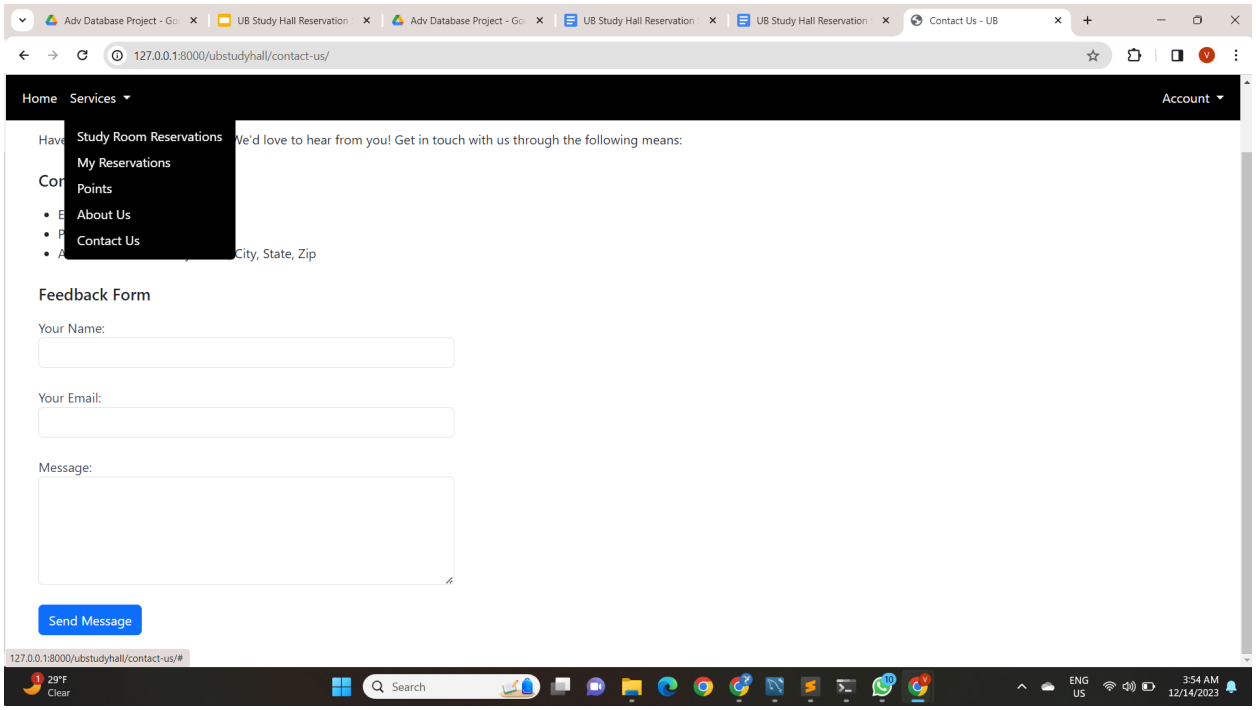
Page 8: Feedback Form

The "Feedback Form" is a dedicated section allowing users to provide feedback, suggestions, or report issues encountered while using the study hall reservation system.



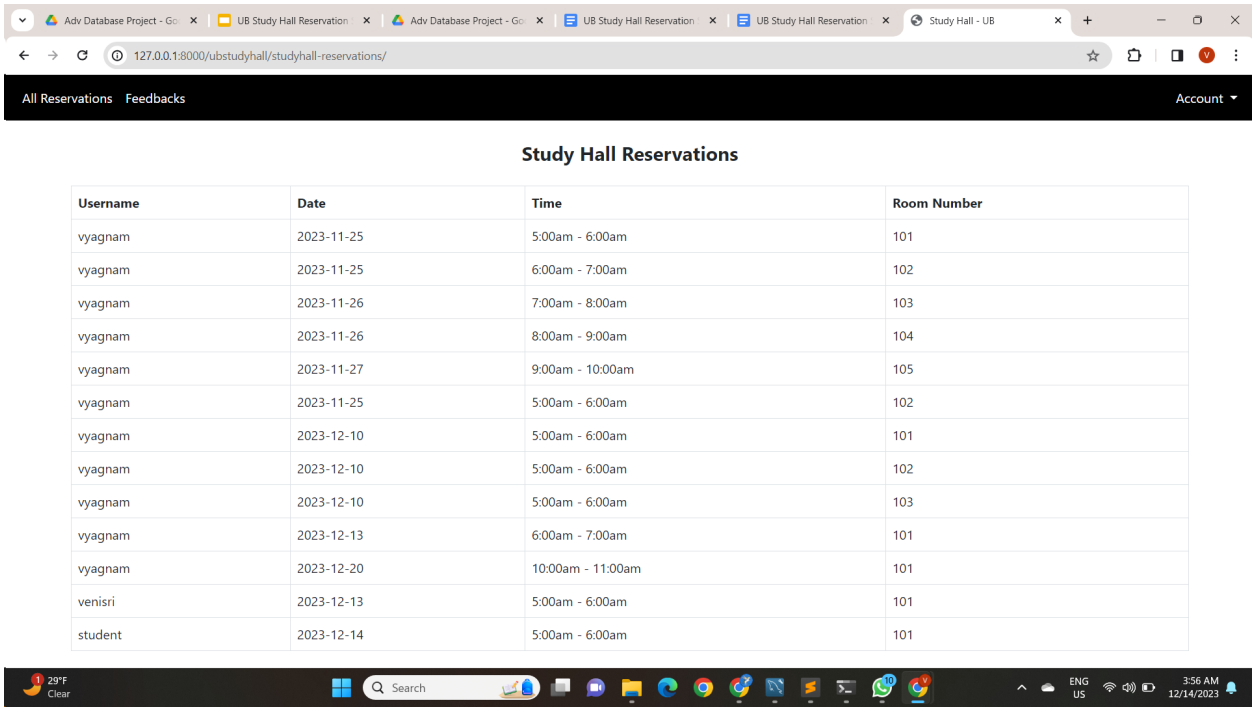
Navigation Bar:

The "Navigation Bar" is a persistent component placed across all pages of the study hall reservation system. It acts as a quick-access menu, allowing users to effortlessly transition between different sections and functionalities of the application.



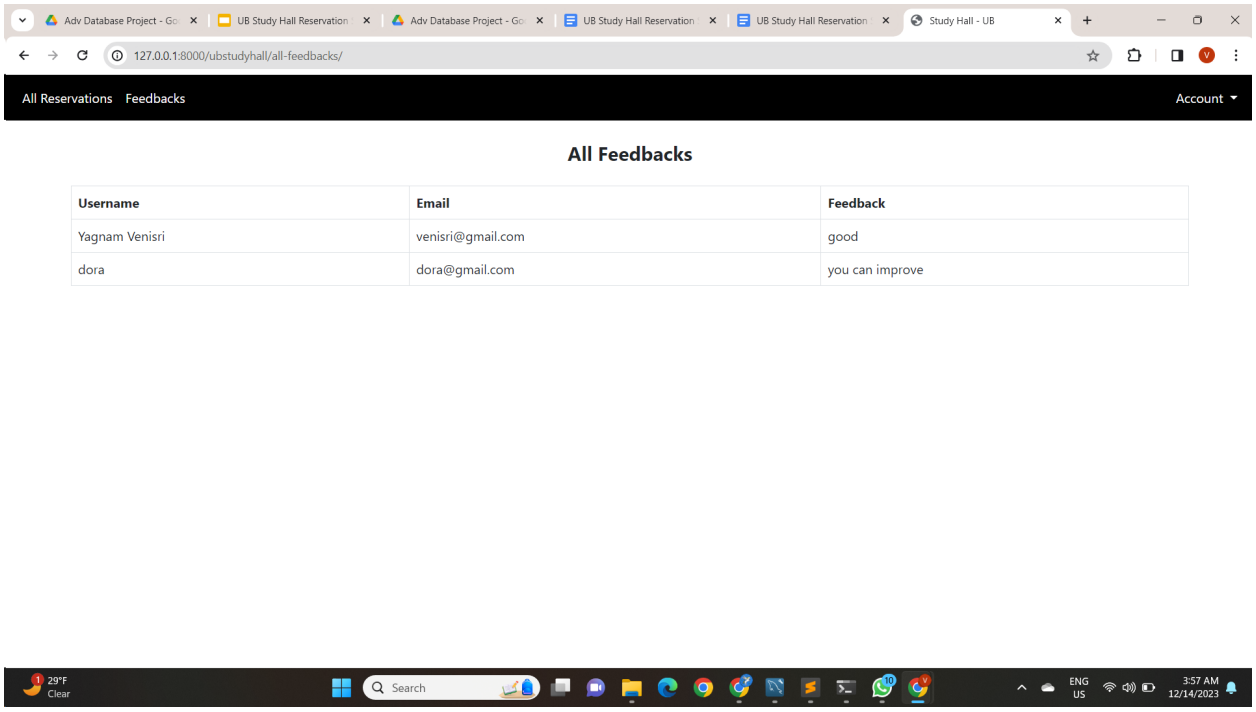
Admin View: All Reservations

The "Admin View" provides an overview of all reservations made within the study hall reservation system. It's an exclusive feature accessible only to authorized administrators, allowing them to monitor, manage, and analyze reservation activities comprehensively.



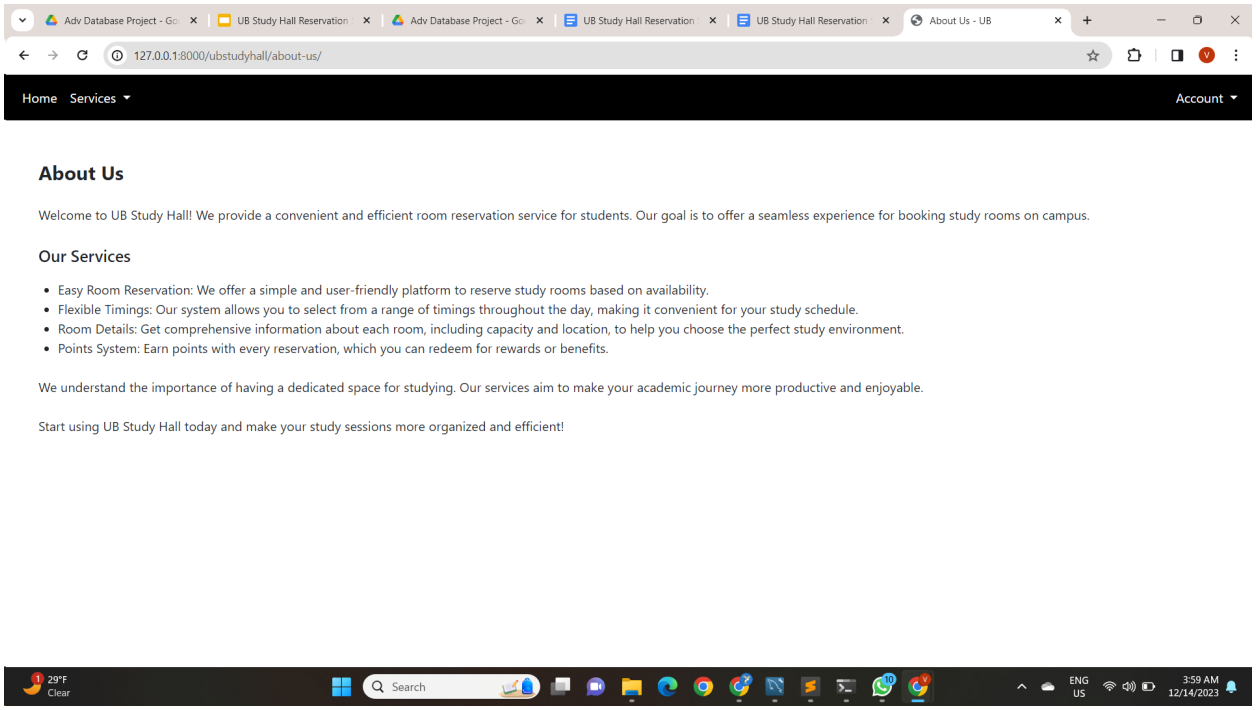
Username	Date	Time	Room Number
vyagnam	2023-11-25	5:00am - 6:00am	101
vyagnam	2023-11-25	6:00am - 7:00am	102
vyagnam	2023-11-26	7:00am - 8:00am	103
vyagnam	2023-11-26	8:00am - 9:00am	104
vyagnam	2023-11-27	9:00am - 10:00am	105
vyagnam	2023-11-25	5:00am - 6:00am	102
vyagnam	2023-12-10	5:00am - 6:00am	101
vyagnam	2023-12-10	5:00am - 6:00am	102
vyagnam	2023-12-10	5:00am - 6:00am	103
vyagnam	2023-12-13	6:00am - 7:00am	101
vyagnam	2023-12-20	10:00am - 11:00am	101
venisri	2023-12-13	5:00am - 6:00am	101
student	2023-12-14	5:00am - 6:00am	101

Admin View: All Feedbacks



The "Admin View: All Feedbacks" section offers administrators a comprehensive overview of all feedback submitted by users within the study hall reservation system. This exclusive feature provides insights into user opinions, suggestions, or concerns, facilitating efficient management and prompt responses.

About Us: Project Motto and Mission



The "About Us" section embodies the ethos and aspirations of the UB Study Hall Reservation System. It serves as a platform to articulate the core values, objectives, and mission driving the project and services provided.

Conclusion:

The "UB Study Hall Reservation System" project has not only met but exceeded the initial objectives set for the system. By leveraging modern web technologies, a robust database management system, and cloud hosting capabilities, we've created a comprehensive solution that enhances the student experience in reserving study hall rooms.

This project allowed us to gain insights into the complexities of developing an integrated system, managing databases, and designing user-friendly interfaces. Additionally, it emphasized the importance of user feedback and adaptability in refining the system for optimal user satisfaction.

The successful completion of this project underscores the importance of collaborative teamwork, technical proficiency, and adaptability in delivering efficient solutions to real-world problems. Our learnings from this project will undoubtedly influence our future endeavors in developing innovative and user-centric applications.
Teamwork played a pivotal role, demonstrating the importance of effective collaboration, communication, and division of tasks, ensuring the project progressed smoothly.

Future iterations could focus on enhancing functionalities like the points system, feedback mechanisms, and reservation management for increased user engagement and efficiency. Consideration of additional features, such as real-time chat support or advanced analytics, to further enrich the user experience.