Library Seat Reservation System

Chatla Sowmya Sri 200293 srisowmya20@iitk.ac.in

CS315 Final Project: Group 4

Abstract

This project presents a Command-Line Interface (CLI)-based Library Seat Reservation System aimed at improving the efficiency and fairness of study space allocation for students. The system allows users to view real-time seat availability, make session-based reservations that expire if not claimed within 15 minutes, and validate bookings via QR code scans at entry.

For administrators, it offers tools to manage and claim reservations, monitor seat usage, and oversee the overall booking process. The backend is built using Node.js with PostgreSQL managing the data, and the CLI features dedicated dashboards for students and admins to ensure a smooth and role-specific user experience.

1 Motivation and Problem Statement

Exam season at IITK is stressful enough with late-night study sessions, endless revisions, and the pressure to perform. Yet, for students at IITK, one of the biggest frustrations isn't the exams themselves, it's showing up to the library only to find every seat taken.

This isn't just inconvenient, it's unfair. Some students take up seats for hours, while others get turned away, forced to study in less productive spaces. The proposed solution is simple and transparent: a seat-booking system that lets students check real-time availability before leaving their rooms, reserve seats in advance with guaranteed QR codes, and walk in to scan and start studying immediately.

Also, this: admins scan the QR code at the entrance to allow entry. Since the website only lets IITK students log in, only IITK-ians can enter the library. With no more manual ID checks, the process becomes smoother.

No more guesswork, no more wasted time. Just one less distraction during the most critical weeks of the semester. Because when it comes to exams, the biggest challenge should be the material, not fighting for a place to study.

2 Methodology

Technology Stack

The system is built using the following core technologies:

- **Backend:** Node.js with Express.js, providing an efficient and scalable framework for handling user interactions and managing business logic.
- **Database:** PostgreSQL, a relational database, is used to manage structured data, including user credentials, reservations, and group associations.
- **Interface:** The user interface is built as a Command-Line Interface (CLI) using Inquirer.js, offering a straightforward and intuitive experience.
- **QR Code Generation:** The qrcode package is utilized for generating unique QR codes to validate reservations.

System Architecture

The Library Seat Reservation System follows a modular structure using the Model-View-Controller (MVC) pattern, with some modifications due to the project's specific design.

- **Database Design** The database schema was defined and imported into PostgreSQL via Supabase, automatically generating the necessary tables to store data such as Users, Reservations, Seats, and Groups.
- **Controllers** Controllers encapsulate business logic, handling interactions like seat reservations, availability checks, user registrations, and QR code generation.
- **Views (CLI Interface)** The system uses a Command-Line Interface (CLI) for user interaction, where students and admins interact through prompts generated by the inquirer library.

Database Design

The system employs a well-structured database design with several key entities, each serving a specific purpose:

- **Users**: Stores student and admin credentials and role information (e.g., student, admin).
- **Seats**: Maintains a theater-style row and column layout, representing the physical seating arrangement in the library.
- **Reservations**: Tracks individual and group reservations, including booking timestamps and expiration times.
- **Admins**: Admin users with privileges to claim reservations and manage library data.
- **Library Hours**: A table for managing and scheduling library opening and closing times. This table is seeded with data for the next 10 days during initial server setup and is updated daily at midnight to delete outdated entries and add the upcoming 10th day's data.

Additional design considerations include:

- **Indexes**: Indexes are employed on critical columns to optimize query performance, ensuring fast retrieval of reservation data and seat availability.
- **Superadmin Role**: A superadmin can manage the creation of other admin accounts, ensuring appropriate access control within the system.

Booking Logic and Session Management

Each booking is timestamped and tracked in the system. The current time is compared with the booking timestamp to determine the validity of the reservation. If a session is not claimed within the defined window

of 15 minutes from the booked time, a scheduled task that runs every minute to check for unclaimed reservations automatically frees up those seats for other students. Additionally, there is another scheduled task that ends all sessions immediately when the library's closing time is reached, removing the need for manual session endings from the user's side.

3 Implementation and Results

The system was implemented with a Node.js backend connected to a PostgreSQL database hosted on Supabase. All core features were exposed via secure API endpoints and integrated with a user-friendly command-line interface using **inquirer.js**.

Every backend route was rigorously tested to ensure correct handling of authentication, seat reservation logic, session expiration, and QR code generation. To validate the QR code functionality, a custom test script was written to simulate both generation and decoding. Additionally, database update behaviors were carefully monitored to ensure consistency.

Once deployed, the CLI presents users with an intuitive interactive menu. Users can choose to log in as a student or admin, sign up, or exit the application. In addition, a special option for creating new admins is accessible exclusively to superadmins.

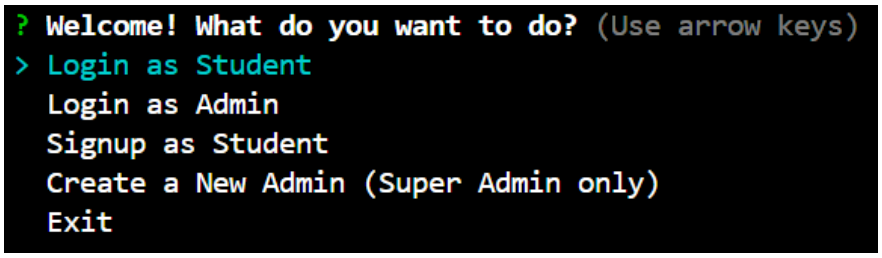

Figure 1: Menu Demonstration

For instance, when a new student signs up, the system prompts them to enter their name, email, password, and roll number. These inputs are validated and then sent to the backend via secure API endpoints. The backend handles the database operations using PostgreSQL hosted on Supabase.

A successful signup is acknowledged with a confirmation message.

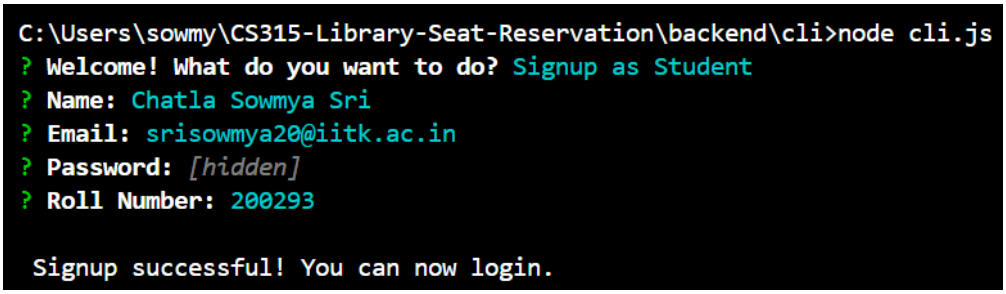

Figure 2: Student Signup Demonstration

Upon successful signup, users are granted access to the system, where they can proceed with the secure login process. Once logged in, users are guided into their respective dashboards, where they can interact with the library seat reservation features and manage their activities based on their roles within the system.

Student Interaction Flow

Students access the system through a secure login restricted to IITK email domains. Once authenticated, they can:

- View the live library seat layout with real-time availability.
- Reserve seats either individually or for a group they've created.
- Receive a unique QR code after a successful reservation, required for claiming the seat on arrival.
- If not claimed within 15 minutes of the booked time, the reservation automatically expires, freeing up the seat for others.
- View their personal booking history, cancel active reservations, and end ongoing study sessions when leaving.

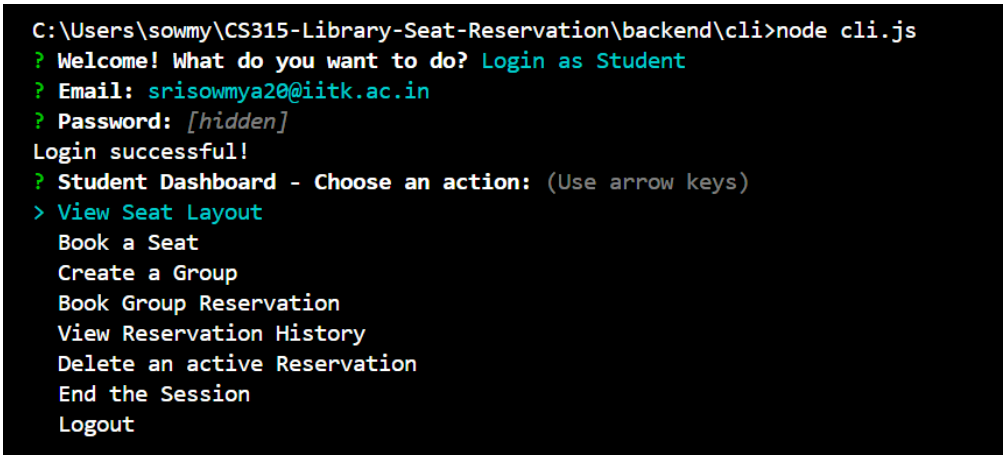

Figure 3: Student Dashboard Demonstration

The seat layout is displayed as a table with row labels (A, B, C...) on the left and column numbers at the top. Each cell shows the current seat status: available, booked, or occupied for easy selection.

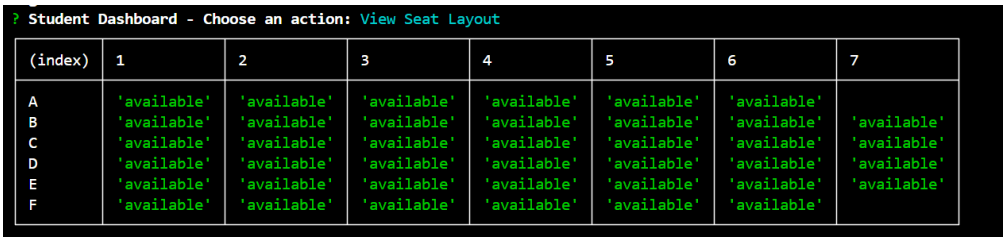

Figure 4: Seats Layout Demonstration

Admin Interaction Flow

Admins access the system through a secure login. Once authenticated, they can:

- View the live seat layout, showing the current availability and occupancy of library seats.
- Update library hours for specific dates, including the option to make the library open 24/7 during busy periods like exams.
- Claim reservations by scanning QR codes for individual bookings.
- Claim group reservations by scanning a group QR code, validating all members at once.
- View a student reservation history by entering his/her roll number.

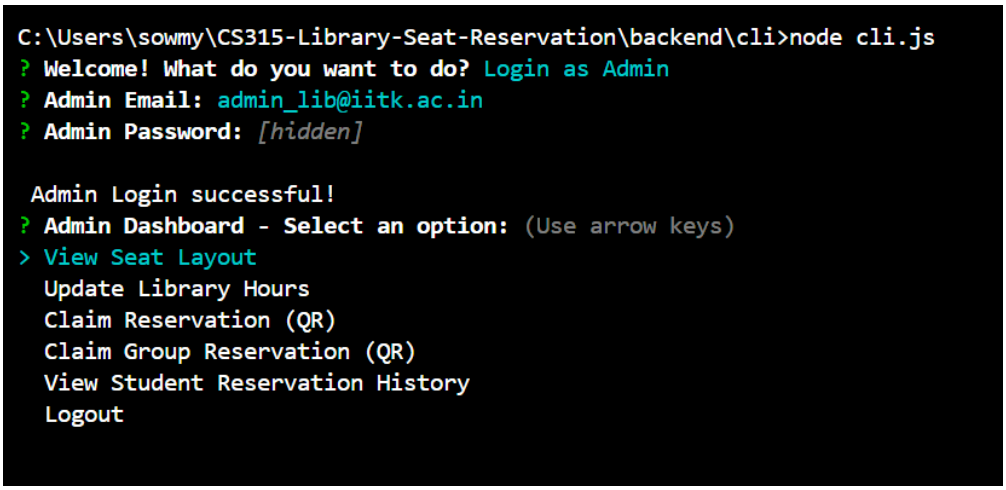

Figure 5: Admin Dashboard Demonstration

Code Repository

All source code is available at: `https://github.com/KreyulTWSA/CS315-Library-Seat-Reservation`

4 Discussion and Limitations

While the Library Seat Reservation System offers a robust backend and user-friendly CLI interface, there are some limitations to consider:

- **Lack of Frontend**: The system does not currently include a full frontend, which could improve accessibility and user experience. A graphical interface would indeed make seat reservations, group management, and reservation details more intuitive.
- **Manual QR Code Handling**: Currently, QR code data is manually stored and provided as a base64 string in the CLI.
- **Notifications**: The system lacks notifications for various actions such as reservation confirmations, seat availability, and session expiry. Push notifications could enhance user engagement.
- **Limited Session Tracking**: The current system does not support automatic logout after a session expires, which could be a useful feature to prevent inactive sessions.
- **Seat Layout Representation**: The current seat layout used in the system does not reflect the actual library layout.

Future Work

The system has the potential for several future enhancements:

- **Frontend Development:** A full front-end interface using a modern web framework (like React) would provide a more user-friendly experience, allowing users to manage reservations visually.
- **Integration with Institutional Database:** Linking the student login system to the institution database for real-time validation of student IDs would improve security and streamline the registration process.

The system could be enhanced by implementing a waitlist feature to notify users when a seat becomes available, adding session management features such as auto-logout, updating the seat layout to reflect the actual library setup, and introducing 'forgot password' and email verification using OTP functionalities for improved security.

Despite these limitations, the system has already demonstrated its potential to significantly improve study space accessibility and fairness during high-demand periods.

5 Contributions

This project was fully developed by Chatla Sowmya Sri. Work done includes:

- Designing the complete database schema, normalization structure, and implementing PostgreSQL queries.
- Building all backend API endpoints using Node.js and Express.
- Implementing logic for time-based reservation expiry, session management, and QR code validation.
- Designing and structuring student and admin dashboard workflows within the CLI.
- Setting up automated tasks (schedulers) to manage expired reservations and enforce library closing rules.
- Preparing thorough documentation and will be conducting the final demonstration of the system.