

Hostel Room Allocation System



2025

Presented by:

TEAM-“Hello World”

TEAM MEMBERS:-

NIKHIL PRASAD 24SCSE1420144

SATYENDRA TRIPATHI 24SCSE1420072

MANDEEP 24SCSE1420030

SIDDHARTH 24SCSE1420043



ABOUT

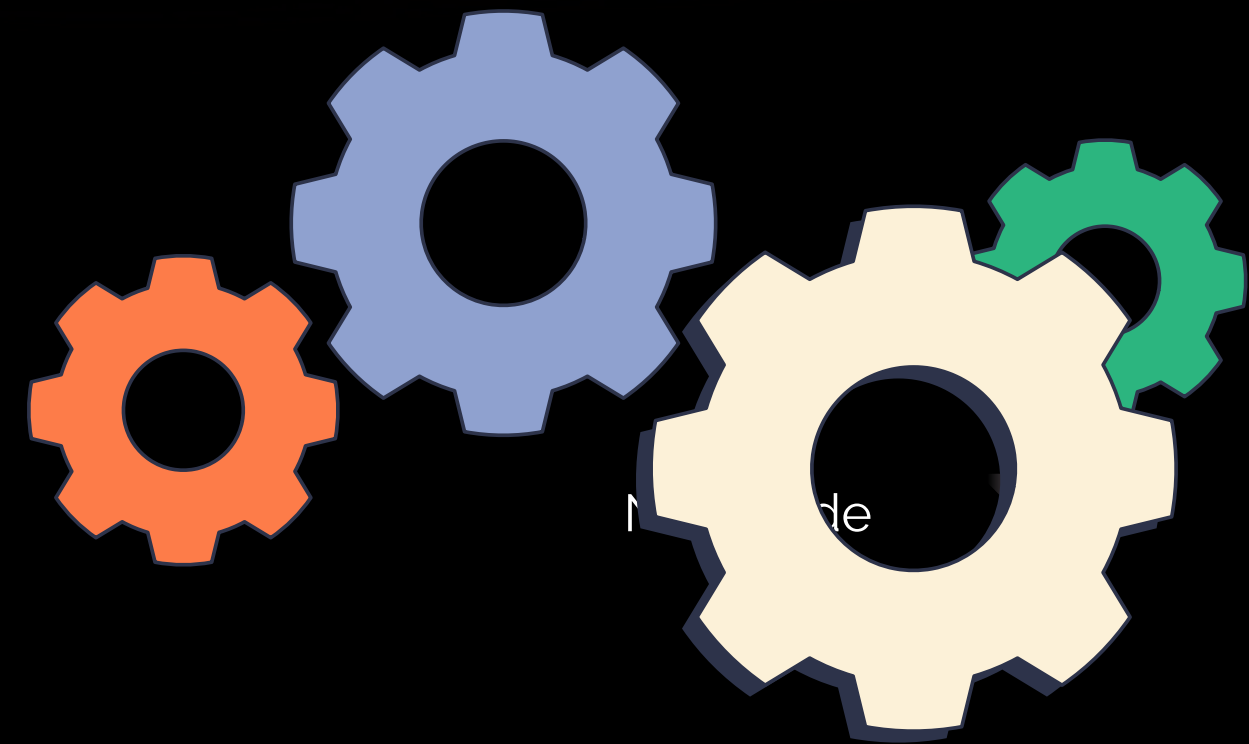


2025

This project enables admin users to manage hostel room allocation for students in a user-friendly Java GUI. Students' information can be added, updated, deleted, and viewed with proper database management using MySQL.

Key Functionalities:

- Add, update, and delete student records.
- View allocated rooms.
- Room allocation based on availability.
- Real-time database sync.



PROBLEM STATEMENT



2025

- Manual room allocation is error-prone and time-consuming.
- Requires a system that automates allocation and makes data easily accessible..
- Data must be stored persistently and be modifiable by the admin..
- Goal: Build a system that simplifies hostel room management with GUI and database backend.

Next Slide





2025

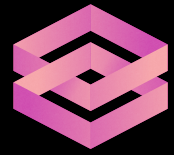
OBJECTIVES



- Simplify room allocation using Java GUI.
- Enable easy management of student records and room availability.
- Provide real-time database sync using MySQL.
- Offer intuitive admin interface with all CRUD operations.

Next Slide





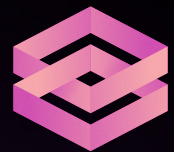
SYSTEM ARCHITECTURE



2025

- Frontend: Java Swing GUI for interacting with system.
- Backend: Java logic using JDBC.
- Database: MySQL for data storage.
- Flow :

Admin \rightleftarrows GUI \rightleftarrows JDBC \rightleftarrows MySQL



TECHNOLOGY STACK

- Frontend: Java Swing (GUI Components).
- Backend: Java (JDK 8+), JDBC.
- Database: MySQL.
- IDE: Visual Studio Code / IntelliJ.
- OS: Windows 11.

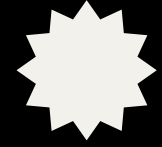


Next Slide



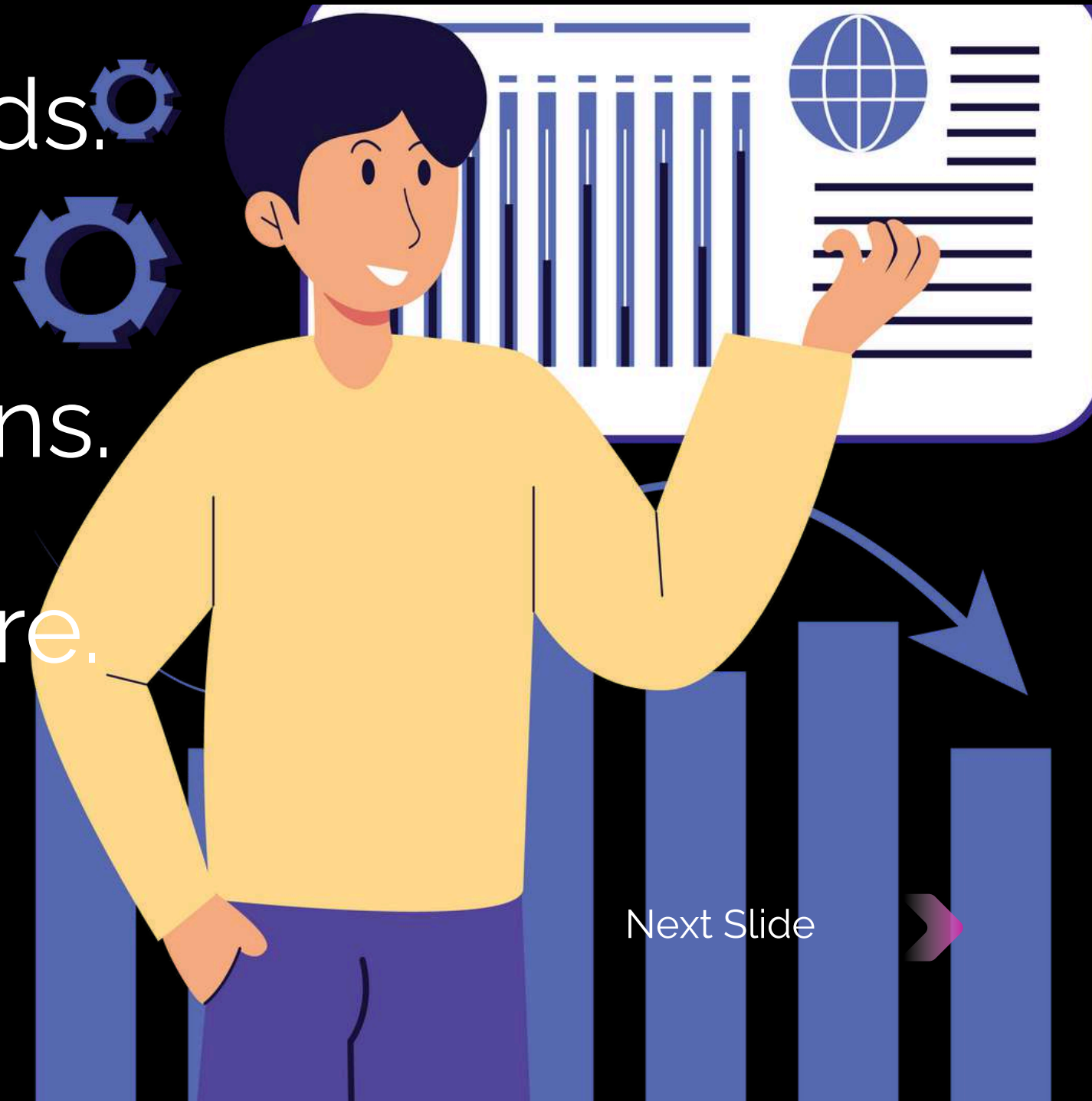


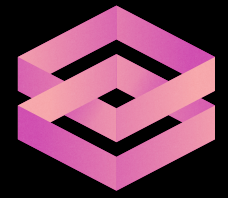
KEY FEATURES



2025

- Add, update, and delete student records.
- Add rooms with availability check.
- Allocate rooms to students.
- View all students, rooms, and allocations.
- Input validation and error messages.
- Transaction rollback on allocation failure.
- Neat and organized GUI interface.



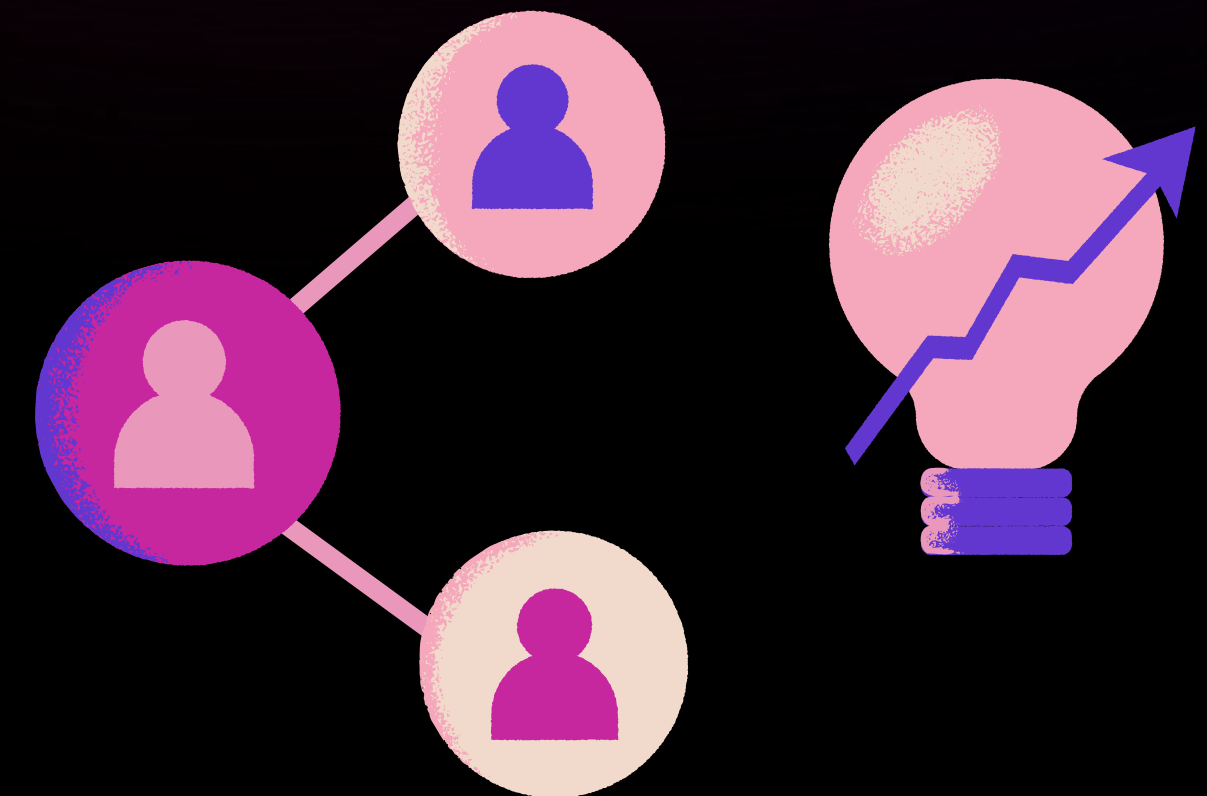


Challenges & Solutions



2025

- **Data Sync:** Solved using JDBC for persistent connection.
- **GUI Management:** Handled using modularized Swing components.
- **Error Handling:** Added alerts, try-catch, and validations.
- **Room Logic:** Ensured no overallocation by checking DB constraints.

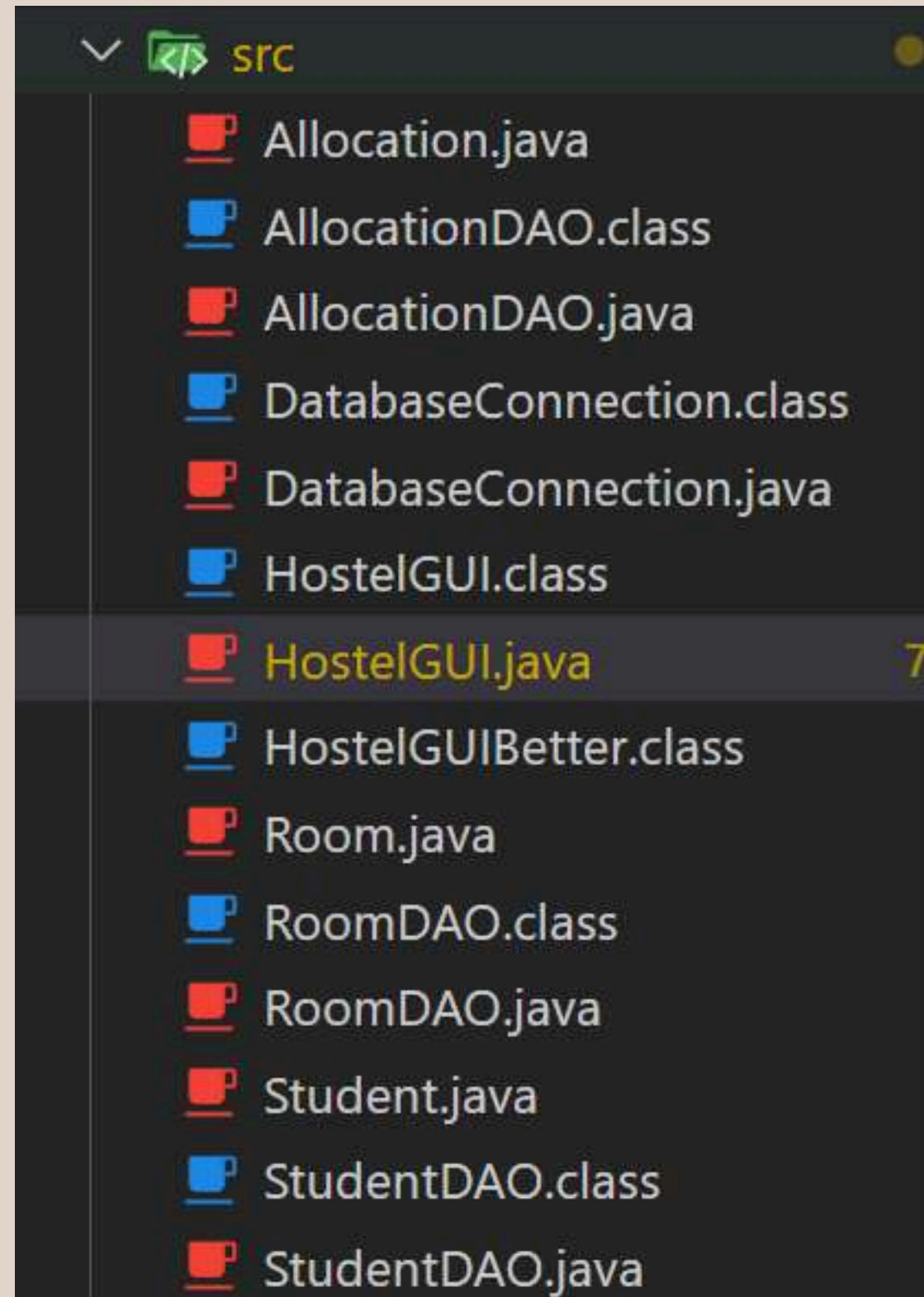


Next Slide



Some Screenshots For The Projects

Project Folder in VS Code



MySQL Database and Table Creation

```
mysql> USE hostel_db;
Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_hostel_db |
+-----+
| allocations          |
| rooms                |
| students             |
+-----+
3 rows in set (0.169 sec)
```

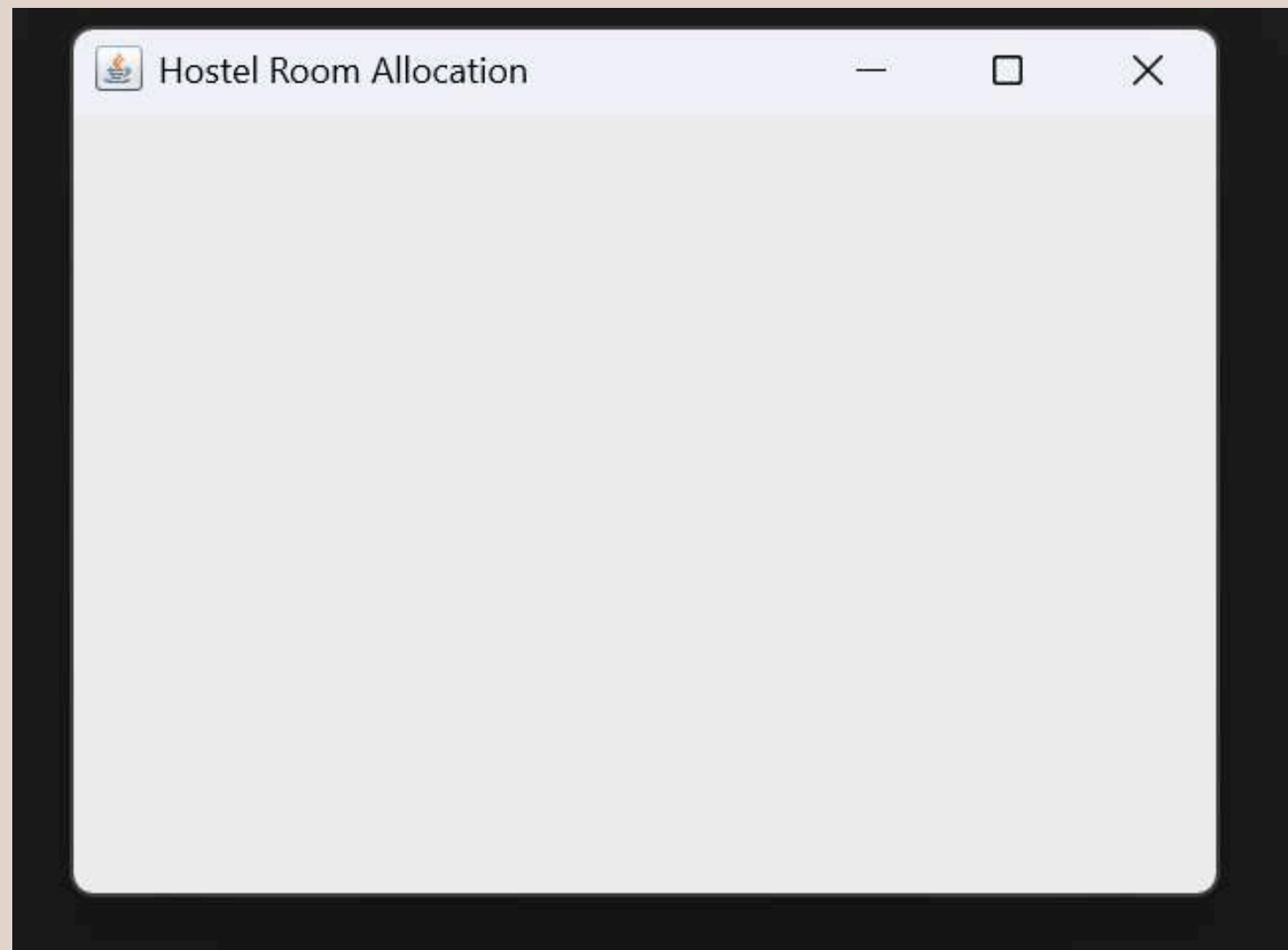

Some Screenshots For The Projects

Filled Add Student Form

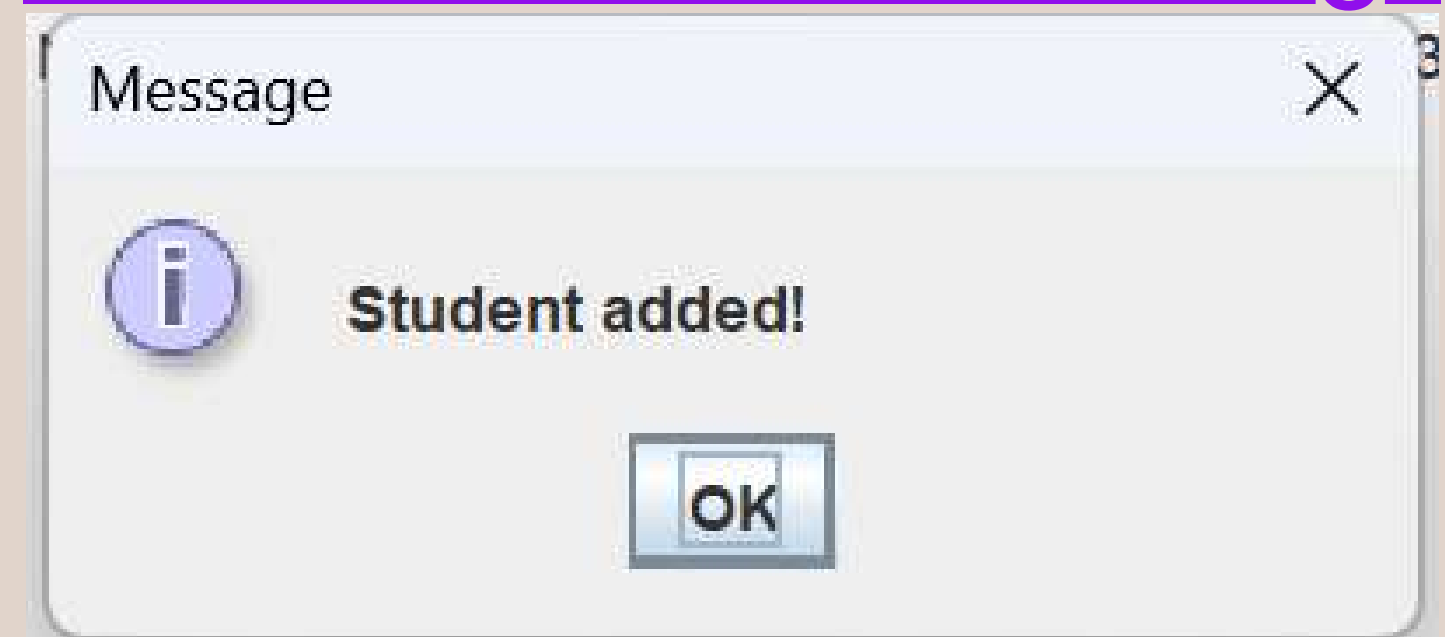
Welcome to Hostel Room Allocation Student ID: Name: Email: Contact: Room ID:

ID: 1, Name: , Email: , Contact:
ID: 4, Name: John Doe, Email: john@example.com, Contact: 1234567890
ID: 6, Name: Nikhil, Email: aklijdf@gmail.com, Contact: 9876543210

Initial GUI (Blank Form)



Add Student Success Message



Some Screenshots For The Projects

Students Table After InsertionI (Blank Form)

```
mysql> SELECT * FROM students;
```

student_id	name	email	contact
1			
4	John Doe	john@example.com	1234567890
6	Nikhil	akljdf@gmail.com	9876543210
9	Mandeep	mandeep765@gmail.com	6200875880

Filled Update Student Form

Welcome to Hostel Room Allocation Student ID: Name: Email:

Contact: Room ID:

ID: 1, Name: , Email: , Contact:

ID: 4, Name: John Doe, Email: john@example.com, Contact: 1234567890

ID: 6, Name: Nikhil, Email: akljdf@gmail.com, Contact: 9876543210

ID: 9, Name: Mandeep, Email: mandeep765@gmail.com, Contact: 620087

Update Success Message

Welcome to Hostel Room Allocation Student ID: Name: Email:

Contact: Room ID:

ID: 1, Name: , Email: , Contact:

ID: 4, Name: John Doe, Email: john@example.com, Contact: 1234567890

ID: 6, Name: , Email: , Contact: 9876543210

ID: 9, Name: , Email: , Contact: 620087

Message

Student updated!

OK

Some Screenshots For The Projects

Filled Delete Student Form

Welcome to Hostel Room Allocation Student ID: Name: Email:

Contact: Room ID:

ID: 1, Name: , Email: , Contact:

ID: 4, Name: John Doe, Email: john@example.com, Contact: 1234567890

ID: 6, Name: Nikhil, Email: akjldfl@gmail.com, Contact: 9876543210

ID: 9, Name: Mandeep, Email: mandeep765@gmail.com, Contact: 620087

Delete Success Message

Welcome to Hostel Room Allocation Student ID: Name: Email:

Contact: Room ID:

ID: 1, Name: , Email: , Contact:

ID: 4, Name: John Doe, Email: john@example.com, Contact: 1234567890

ID: 6, Name: , Email: , Contact: 3210

ID: 9, Name: , Email: , Contact: 620087

Message

Student deleted!

MySQL: Students Table After Deletion

```
mysql> SELECT * FROM students;
+-----+-----+-----+-----+
| student_id | name       | email           | contact |
+-----+-----+-----+-----+
| 1          |           |                 |         |
| 4          | John Doe  | john@example.com | 1234567890 |
| 9          | Mandeep Singh |                 |         |
+-----+-----+-----+-----+
3 rows in set (0.009 sec)
```


Some Screenshots For The Projects

Filled Room Allocation Fields & Message

Welcome to Hostel Room Allocation Student ID: Name: Email:

Contact: Room ID:

ID: 1, Name: , Email: , Contact:
ID: 4, Name: John Doe, Email: john@example.com, Contact: 1234567890
ID: 9, Name: f

Message

Room allocated!

MySQL: Allocations Table After Allocation

```
mysql> SELECT * FROM allocations;
```

allocation_id	student_id	room_id	allocated_date
1	1	1	2025-05-01
2	1	1	2025-05-20
4	1	1	2025-05-20
7	9	2	2025-05-24

```
4 rows in set (0.009 sec)
```


Some Screenshots For The Projects

MySQL: Rooms Table Showing Availability Update

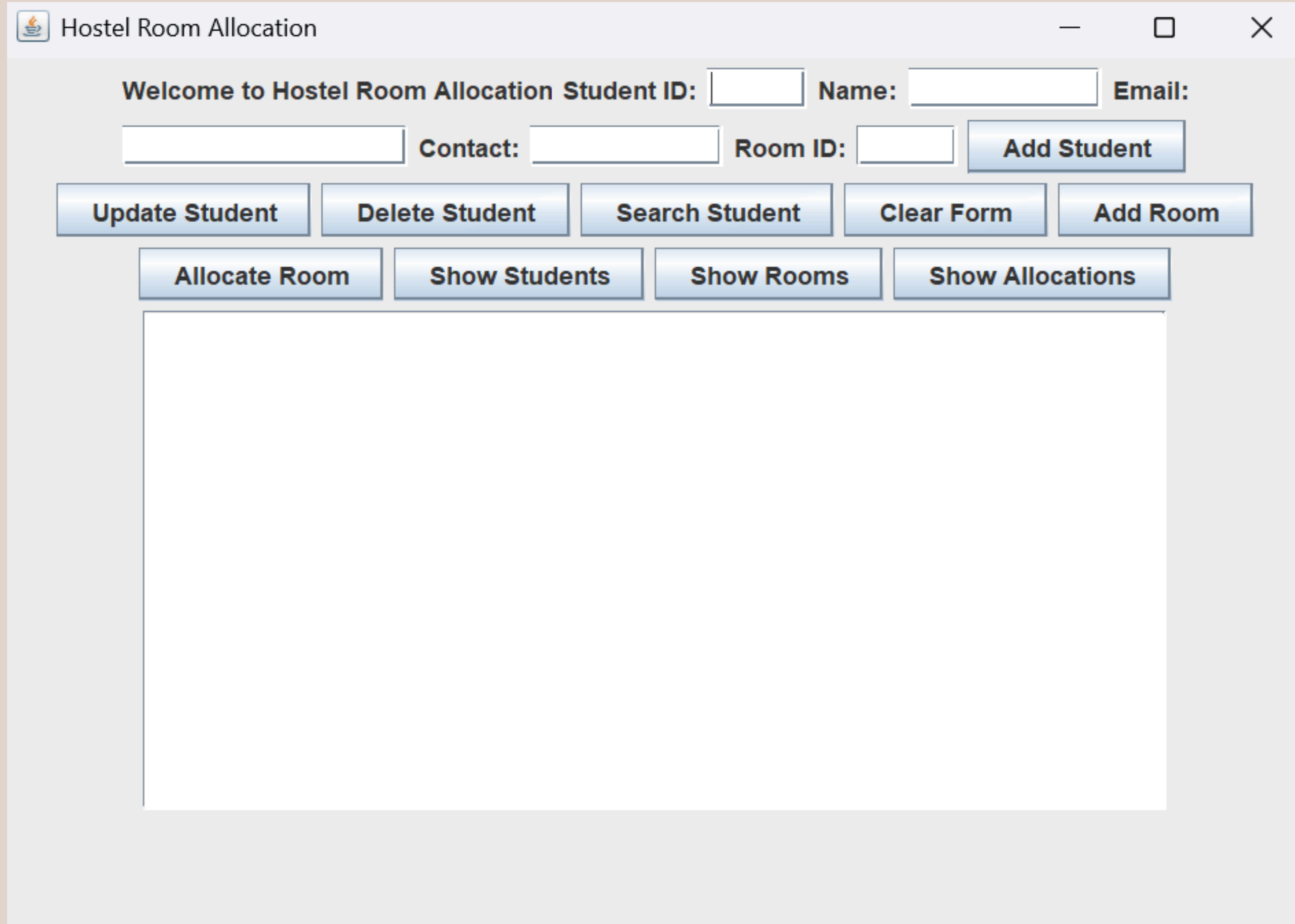
```
mysql> SELECT * FROM rooms;
```

room_id	room_number	capacity	available
1	R101	2	0
2	R102	1	0

```
2 rows in set (0.012 sec)
```

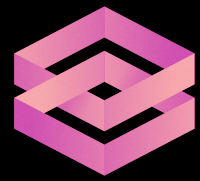
Some Screenshots For The Projects

GUI Display of All Students (JTable)



The screenshot displays a Java Swing window titled "Hostel Room Allocation". The window contains a form for adding students with the following fields and buttons:

- Form fields: "Student ID:", "Name:", "Email:", "Contact:", and "Room ID:".
- Buttons: "Add Student", "Update Student", "Delete Student", "Search Student", "Clear Form", "Add Room", "Allocate Room", "Show Students", "Show Rooms", and "Show Allocations".
- A large, empty rectangular area at the bottom, which is a JTable, intended for displaying the list of students.



Conclusion & Future Scope

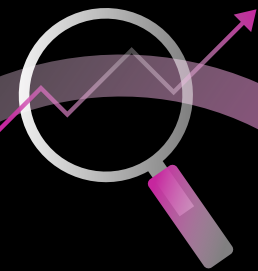


2025

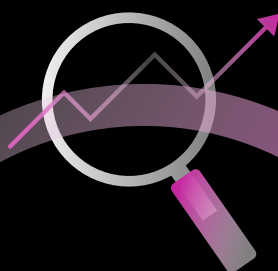
✓ The Hostel Room Allocation System is successfully implemented with essential features.



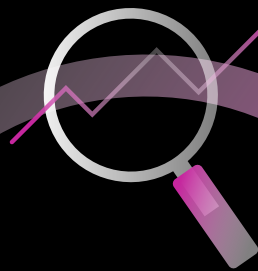
Future Scope:



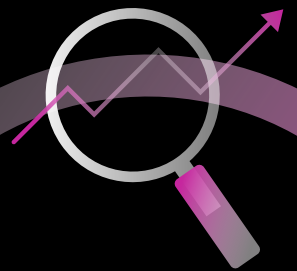
Add student
login
feature.



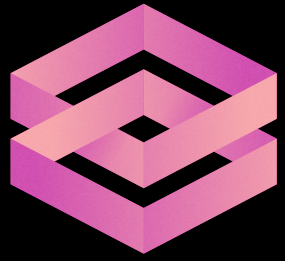
Improve
UI/UX.



Add automated
room
deallocation.



• Add
notifications
email features



2025

Thank You!

Thank you for your attention!

