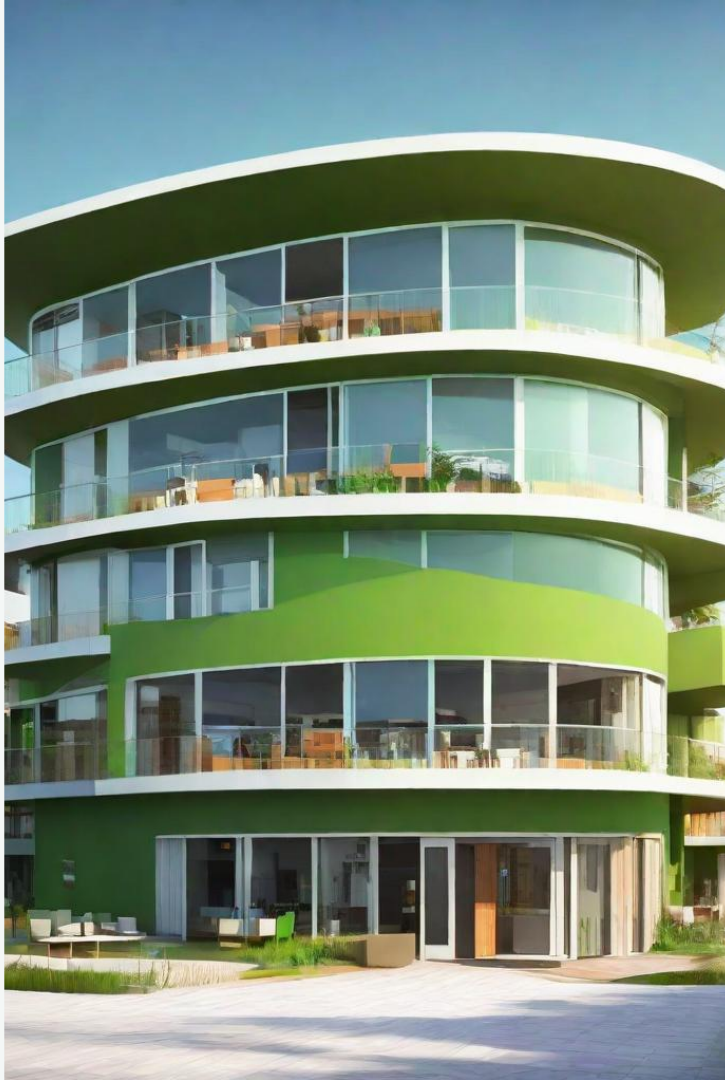
Hostel Management System

Simplifying Student Accommodation

Team Names:

- Sundar Teja Akula
- Pranav Jeksani
- Akhil Goud Burra



Hostel Management System

This presentation discusses the need for a modern and efficient hostel management system to address the challenges faced by the students in india.

Project Description

The rising number of educational institutions has led to a corresponding increase in private hostel facilities for student accommodation.

Primary issue: Relying on offline System.

Challenges Faced:

- Long Waiting for assigning hostel rooms.
- Absence of an online system for booking private study rooms.
- Reporting lost items in the hostel was restricted to offline channels.
- Lack of online access to view whether the books are available in the library or not.

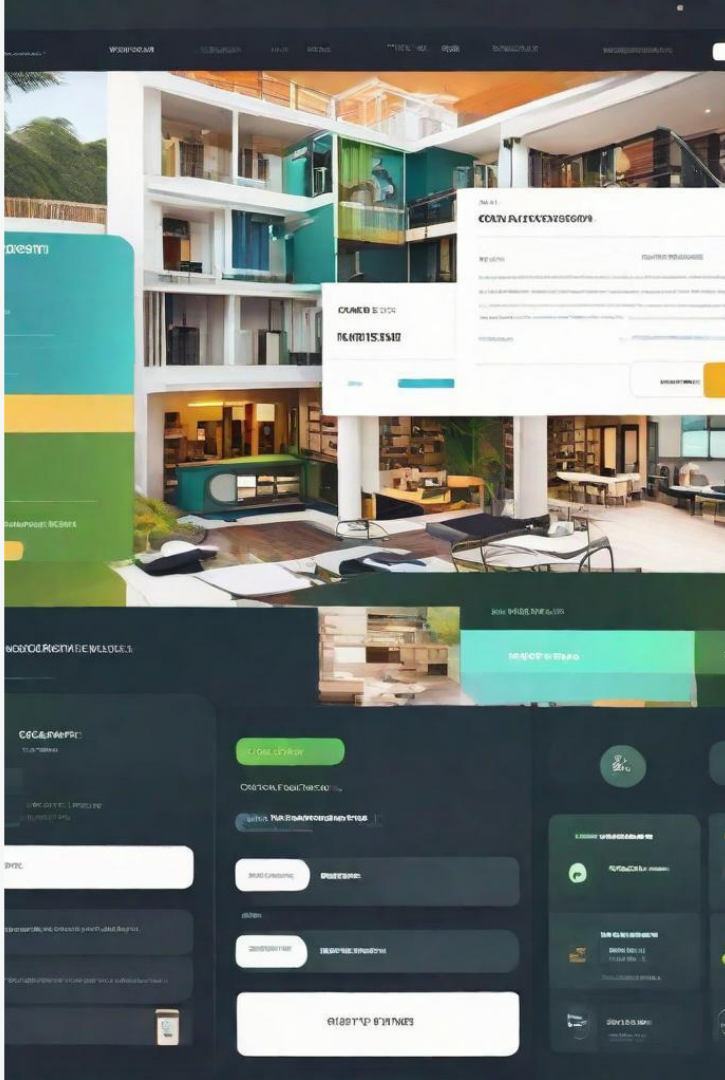
Goal:

Addressing Student Challenges:

Our primary goal is to mitigate these challenges faced by the students in the private hostels by developing a modern and efficient Hostel Management System.

Simplicity and Clean Design.

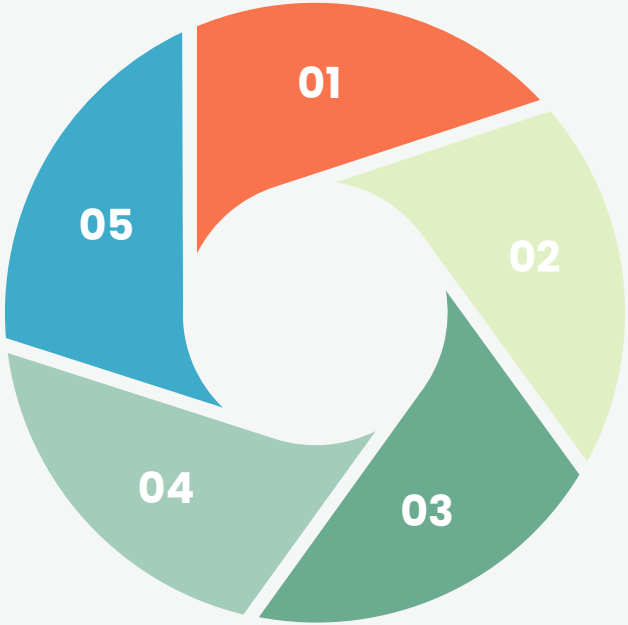
Developing a Simple and Clean Web Application.



Motivations/Objectives

Provide an online booking system for hostel and study rooms.

Eliminate waiting time for room assignments.



Enable online access to library books availability.

Integrating the complaint registration system.

Software Features

End Users:

Feature 1: Login – For the login, end users should login first before you can access the system.

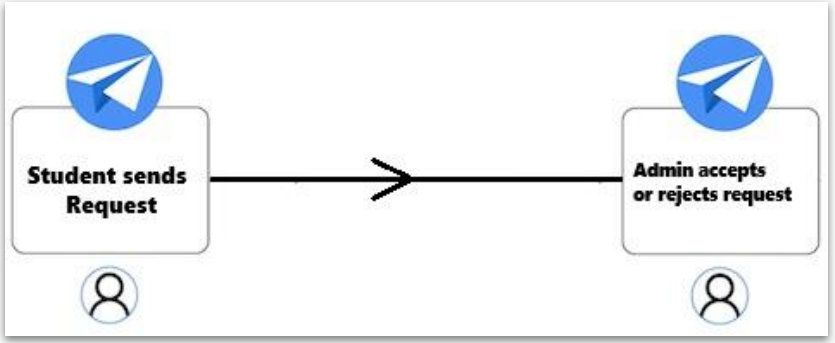
Feature 2: Sign Up – For the sign-up, end users should fill in the forms, such as your username, password, and email address.

Admin Users:

Feature 1: Login – For the login, admin users must log in first before you can access the system.

Feature 2: Sign Up – For the sign-up, admin users should fill in the forms, such as your username, password, and email address.

End Users:



Hostel Room Booking Module

Feature 1: End-users can book a hostel room by sending a “Request” to the admin user.

Feature 2: End-users can book a hostel room monthly or daily.

Study Room Booking Module

Feature 1: End users can book the hostel study room monthly or daily.

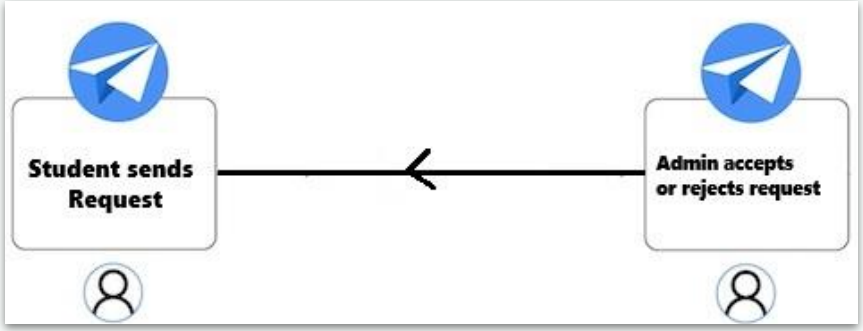
Admin Users:

Hostel Room Management Module

Feature 1: Admin user can add rooms by specifying Room No and No of beds.

Feature 2: Admin user will also have an option to view all his added rooms.

Viewing New Requests Module



Feature 1: Admin user can accept or reject the request made by the end user by checking the availability of the room.

End Users:

Posting Missing Items Module

Feature 1: End user can post the missed items by identifying name of the item, item description, contact information, uploading lost item Image.

Viewing Paid Transactions Module

Feature 1: End user can view all the transactions that they made for booking hostel room and study room.

Admin Users:

Viewing Paid Transactions Module

Feature 1: Admin user can view all the transactions that are made by the end users.

End Users:

Payment Module

Feature 1:

After end user sends a request for hostel room booking, the administrator can see the request and upon the availability of rooms, admin will accept or reject the request.

If admin accepts the request, end user will get an option to pay.

If admin rejects the request, end user will not get the option to pay.

Fetching Library Books Module

End User

Feature 1: End users will have the capability to fetch all the books for their specific stream and their availability status.

Feature 2: End users will be authenticated by authentication layer.

Feature 3: End users are protected by authorization layer.

Feature 4: End users are restricted to fetch 10 times per minute.

Managing Library Books Module

Admin User

Feature 1: Admin user will have capability to perform CRUD operations on the books, streams, and books availability status.

Feature 2: Admin users will be authenticated by authentication layer.

Feature 3: Admin users are protected by authorization layer.

Feature 4: Admin users are restricted to fetch 20 times per minute.

Choosing the Right Approach

A Trade-off Between Simplicity and Flexibility

Model 1: Direct Data Access

- Simplicity and Performance

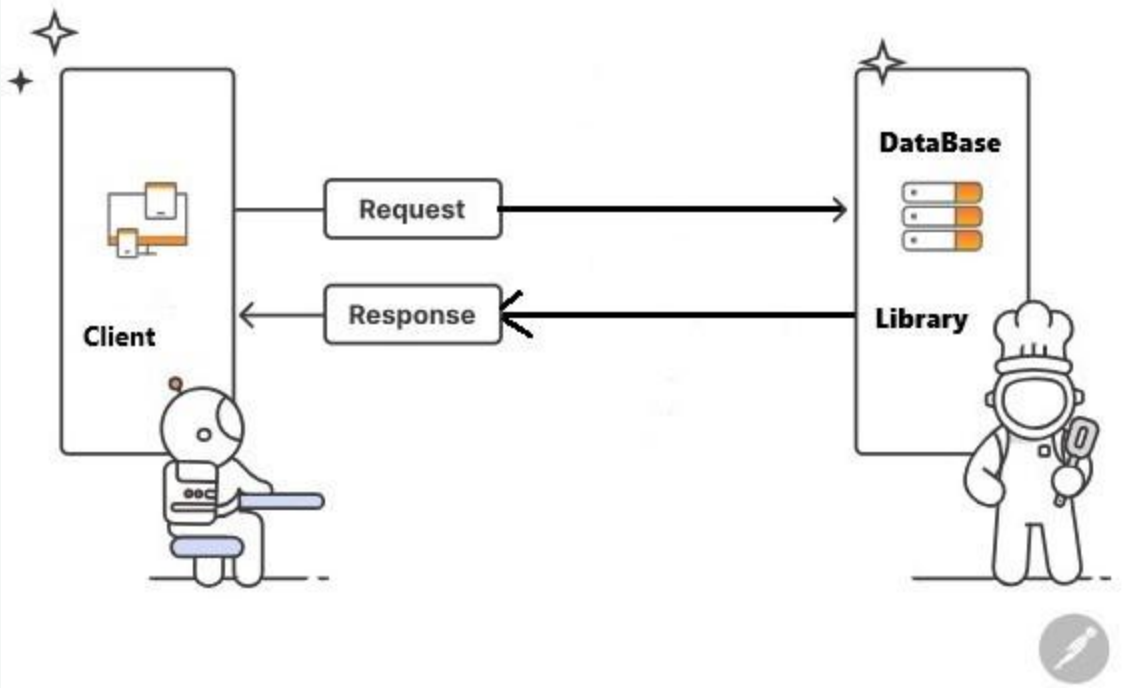
Model 2: Indirect Data Access

- Enhancing Flexibility and Security by adding Interface Layer

Direct Data Access

Pros:

- Simplicity
- Better Performance
- Real-Time Access



Cons:

- Coupling: tightly coupled
- Security: exposing the database directly

Indirect Data Access

Pros:

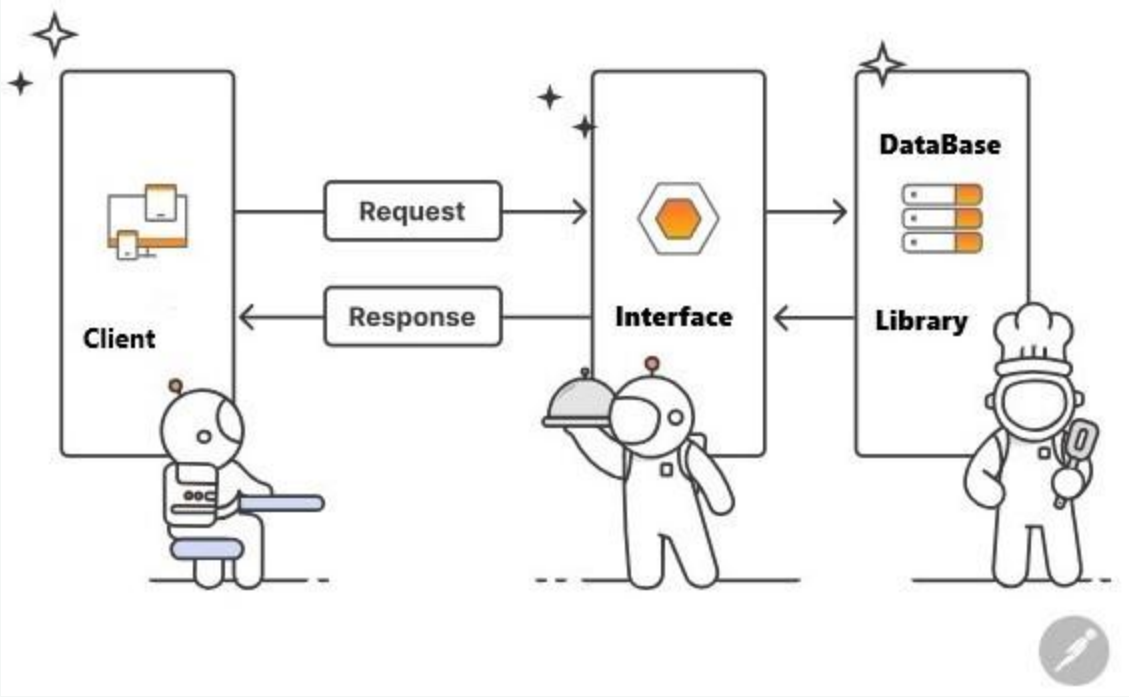
- Decoupling: flexible
- Security
- Abstraction

Cons:

- Complexity
- Latency

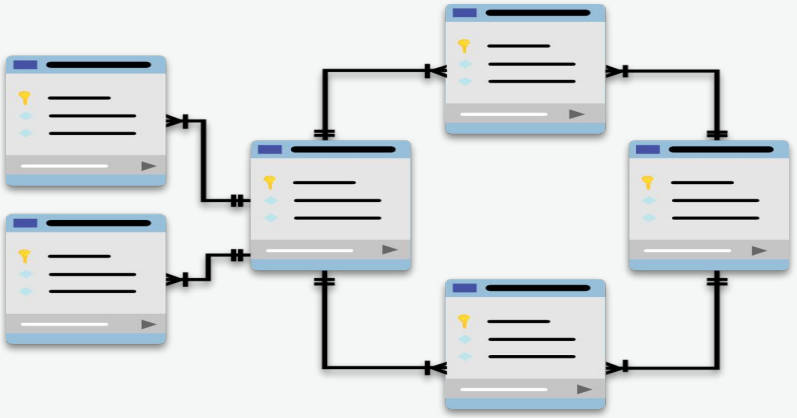
Considerations:

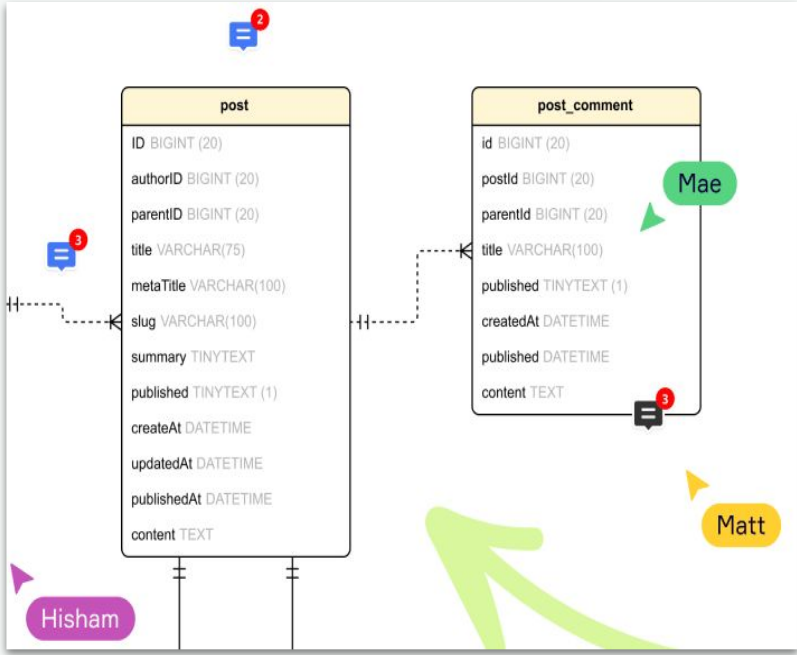
- Scalability and flexibility
- Security
- Future Changes



Database Design

- Define Purpose and Scope.
- Identify Entities and Relationships.
- Define Attributes.
- Define Data Types and Constraints.
- Creating Tables.
- Identifying Functional Dependencies.
- Normalize Data:
  - Eliminating Redundancy:
    - Anomalies.
  - Decompose Tables:
    - Lossless Decomposition.
    - Preserving Dependencies.
  - Reduce to 3NF
- Establish Relationships.
- Implement Database.





Testing

- Unit testing

- integration testing

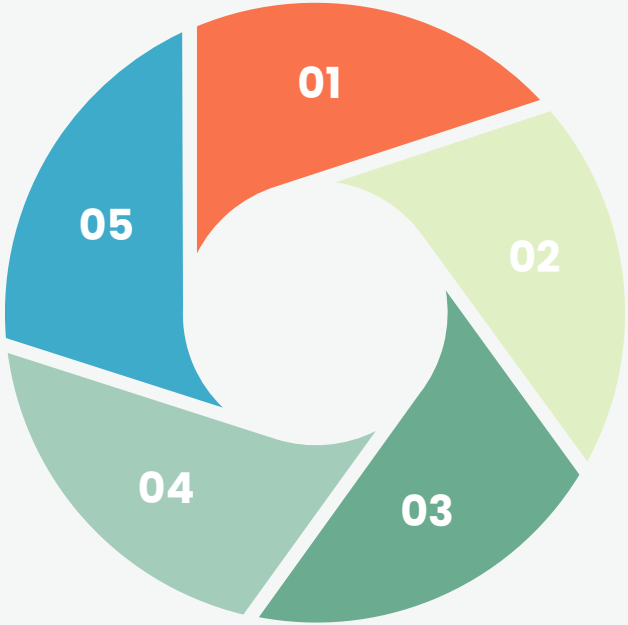
Technology Stack

Frontend technologies:

HTML,CSS, JS

Database:

Postgresql/MYSQL



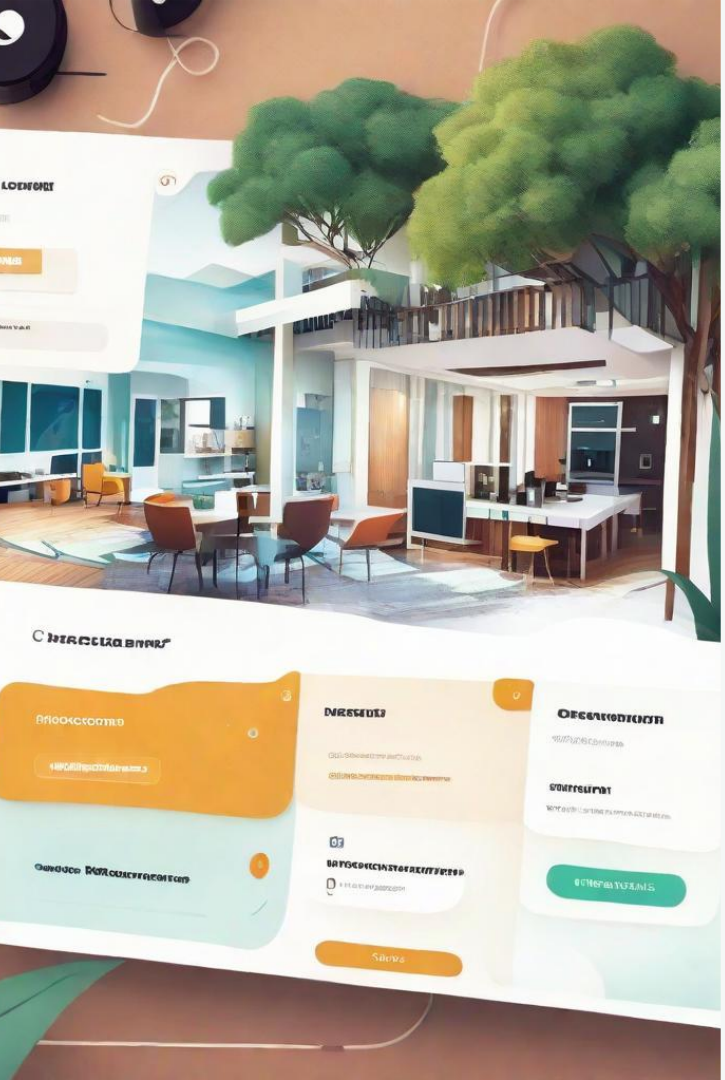
The technology stack used includes:

Backend technologies:

Python

Frameworks and tools:

Django, Django Rest Framework, Git and Insomnia.

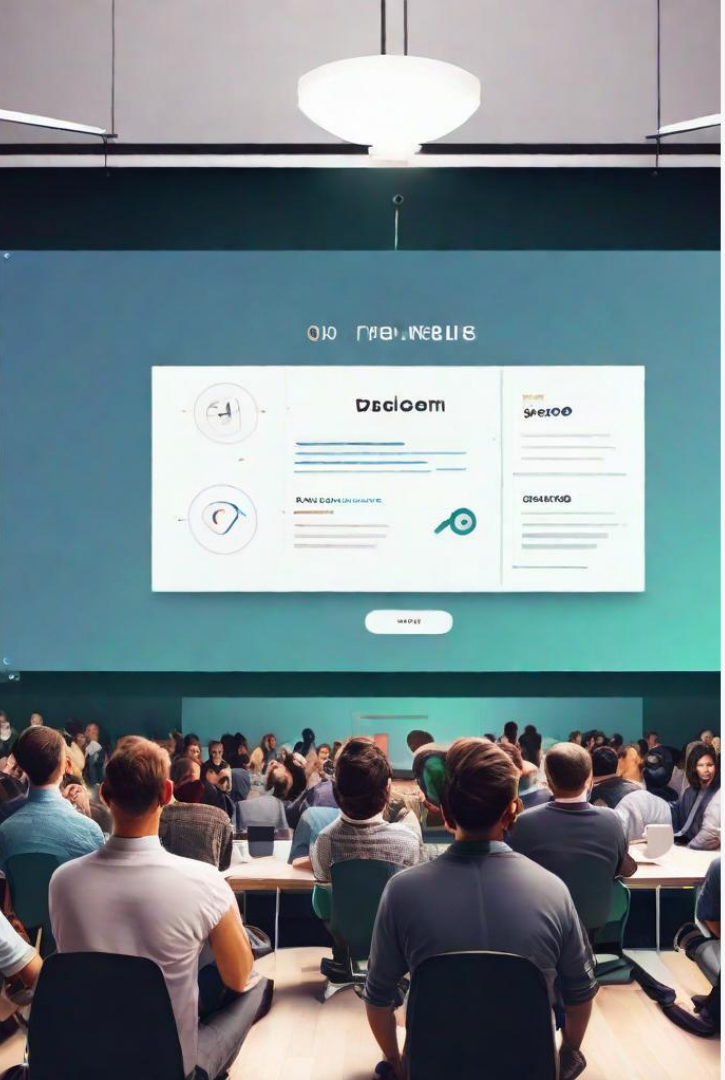


Conclusion

By addressing the challenges like lack of online access, we want to develop a simple and clean web application that resolves the issues.

This web app will offer an online booking system for hostel rooms, study rooms, a complaint registration process, and an online system to check if the books are available for the specific stream.

This ensures a more convenient and accessible hostel management system.



Q&A

01

02

Thank you for your time and attention 🙂