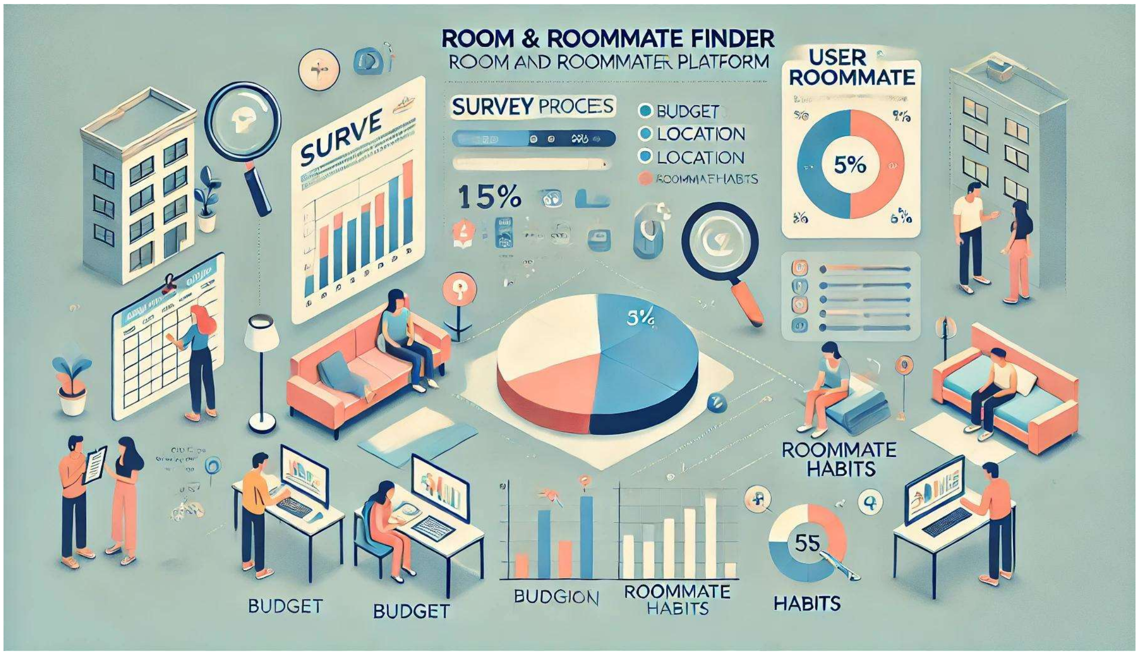
PROBLEM STATEMENT

The **Room and Roommate Finder** platform aims to address the challenges and inefficiencies faced by individuals in finding suitable accommodations and compatible roommates. Traditional methods, such as word-of-mouth or outdated listing websites, often lack personalization, real-time updates, and ease of use. These limitations make it difficult for users to find rooms or roommates that align with their specific preferences, such as location, budget, or lifestyle compatibility. Additionally, issues like poor data security, complex interfaces, and limited search options can further discourage users from effectively utilizing existing platforms.

The **Room and Roommate Finder** platform seeks to resolve these issues by creating a modern, efficient, and user-friendly solution. It will offer advanced filtering options, secure user authentication, and real-time updates to ensure users can find their ideal accommodations or roommates quickly and securely. Below are the broad aims and objectives of the project.



PROPOSED METHODOLOGY

1. Investigation and Requirement Gathering

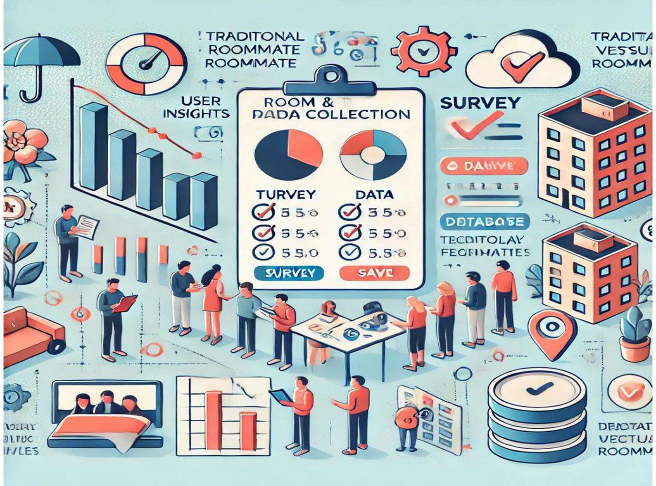
Objective: To understand the needs of individuals seeking rooms and roommates and gather insights to design a user-friendly platform tailored to their preferences.

- User Research: Conduct surveys and interviews with students, working professionals, and landlords to understand their requirements, preferences, and common challenges when searching for rooms or roommates. Key areas of interest include budget, location, cleanliness habits, and compatibility criteria.
- Market Analysis: Examine existing platforms for finding rooms and roommates to identify their common features, strengths, and weaknesses. Applications like Roomster, SpareRoom, and Craigslist will be analyzed to determine what makes them effective and where they could improve.

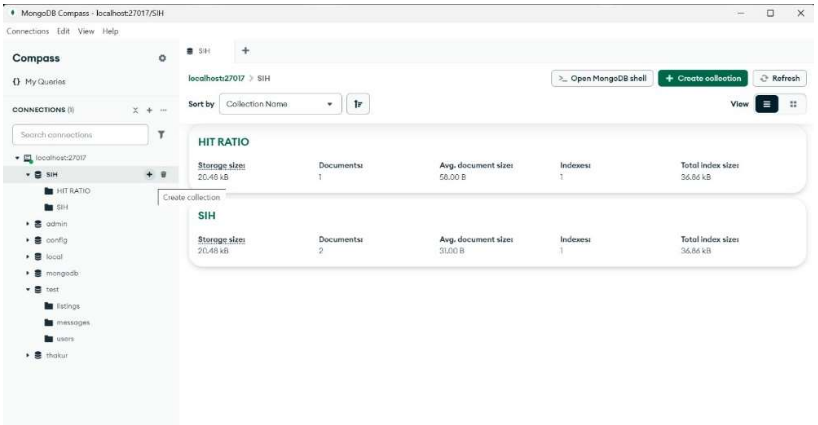
Technical Feasibility

- Review the selected technologies (Node.js, Express.js, MongoDB, HTML, CSS, and Tailwind CSS) to ensure their suitability for building a scalable and secure platform.
- Evaluate how these technologies will be utilized to implement features like advanced search filters, user authentication, and secure data storage while ensuring a seamless user experience**.**

Goal: To provide a data-driven understanding of user needs, allowing the development of a robust, intuitive, and efficient Room and Roommate Finder platform.



By using MongoDB, the application can handle dynamic and flexible data storage, allowing for future scalability and adaptability.



HIT RATIO

Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20.48 kB	1	58.00 B	1	36.86 kB

SIH

Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20.48 kB	2	31.00 B	1	36.86 kB

Authentication and Security:

For authentication, the platform will use JWT (JSON Web Tokens) and bcrypt for secure password hashing, integrated with Node.js. The following components will be included in the authentication process:

1. User Registration:
   - Users (room seekers and room owners) can sign up using an email address and password.
   - Passwords will be securely stored using bcrypt hashing.
   - Upon successful registration, a JWT token will be generated to authenticate users for future requests.
2. Login Process:
   - Users can log in using their email and password.
   - The system will verify the credentials by comparing the entered password with the hashed password stored in the database.
   - If successful, a JWT token is returned for subsequent requests, providing authenticated access to the platform.
3. Password Recovery:
   - Users who forget their passwords can request a password reset link via email.
   - A secure link will be sent, allowing them to reset their password.
4. Session Management:

- JWT will be used to handle session management. Once authenticated, users will have access to protected routes, such as the ability to view, modify, and apply for room listings, and manage their profiles.

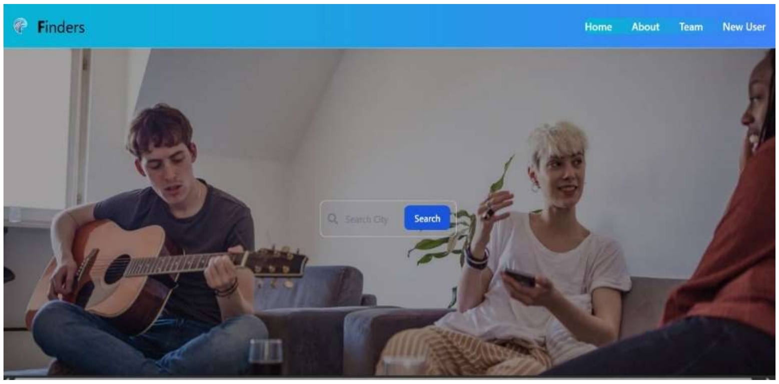
5. Authorization:
    - Roles will be implemented to differentiate between room seekers and room owners. This ensures that only authorized users can access their respective dashboards and perform actions such as creating, modifying, or deleting listings.

3. Development and Implementation

Objective: To develop the core functionality of the Room and Roommate Finder platform, implementing both frontend and backend components, as well as integrating real-time data processing.

Frontend Development:

1. Room Search Interface:
   Develop the room listings page with advanced search filters (location, price range, room type, amenities, etc.) using HTML for structure and Tailwind CSS for styling, ensuring the design is responsive and user-friendly. Room seekers will be able to easily browse through available rooms and filter results according to their preferences.



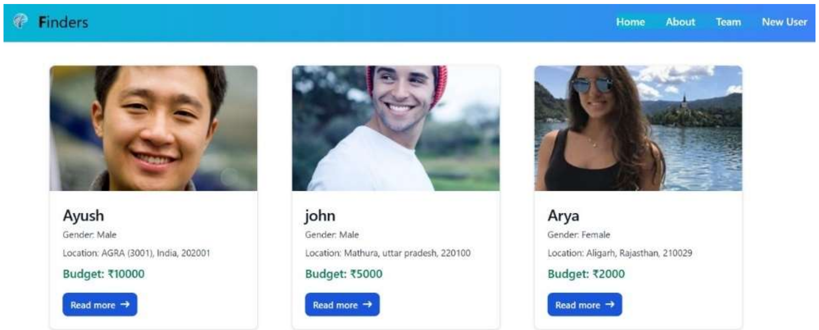
2. Room Application Form:
   Implement a streamlined form for room seekers to apply for rooms directly from the listing page. The form will collect relevant data such as:
    a. Personal details (name, contact, etc.)
    b. Room preferences (location, budget, etc.)

   c. Any additional information or requests

3. The form will also allow users to upload documents like identification or other required files, making the application process efficient and user-friendly.
4. User Dashboard:

   Create a user dashboard where room seekers can:

   a. View and manage their applications (pending, accepted, rejected)
   b. Update their profile information (bio, preferences, etc.)
   c. Track the status of their applications and communicate with room owners
   d. Room owners will also have their dashboard to manage room listings, view applications, and respond to potential tenants.



Ayush
Gender: Male
Location: AGRA (3001), India, 202001
Budget: ₹10000

john
Gender: Male
Location: Mathura, uttar pradesh, 220100
Budget: ₹5000

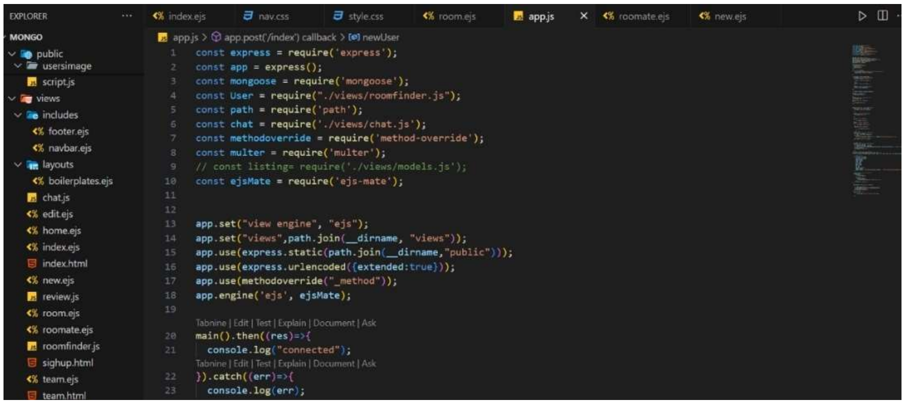
Arya
Gender: Female
Location: Aligarh, Rajasthan, 210029
Budget: ₹2000

5. Frontend Tech: The user interface will be created using HTML and styled with Tailwind CSS to ensure responsiveness and a modern look.

Backend Development:

1. Real-Time Room Listings:

   The backend will handle storing and managing room listings. MongoDB will be used to store room details, including descriptions, prices, amenities, and available dates. The backend will handle requests for adding new rooms, updating existing listings, and removing rooms.
2. Using Node.js with Express.js, RESTful APIs will be implemented to fetch room listings, add new rooms, and update listings. MongoDB will be connected for storing data and handling real-time interactions.



```javascript
const express = require('express');
const app = express();
const mongoose = require('mongoose');
const User = require("./views/roomfinder.js");
const path = require('path');
const chat = require('./views/chat.js');
const methodoverride = require('method-override');
const multer = require('multer');
// const listing= require('./views/models.js');
const ejsMate = require('ejs-mate');


app.set("view engine", "ejs");
app.set("views",path.join(__dirname, "views"));
app.use(express.static(path.join(__dirname,"public")));
app.use(express.urlencoded({extended:true}));
app.use(methodoverride("_method"));
app.engine('ejs', ejsMate);

main().then((res)=>{
  console.log("connected");
}).catch((err)=>{
  console.log(err);
```

3. Application Management:

   The backend will manage room applications. Once a room seeker applies for a room, their application will be saved in MongoDB. Room owners can review the applications, approve or reject them, and manage the status of each application.

4. The system will also allow room seekers to cancel or modify their applications. The backend will ensure that applications are securely stored and updated in real-time.

5. Authentication and Authorization:

   JWT (JSON Web Tokens) and bcrypt will be used for securely handling user authentication. The following features will be implemented:

   a. User Registration: Users (room seekers and room owners) can sign up using their email and password. Passwords will be securely hashed using bcrypt before being stored in the database.

   b. Login: Users can log in with their email and password, and a JWT will be generated to authenticate further requests.

   c. Password Recovery: A password reset process will be implemented to allow users to securely reset their passwords.

   d. Role-Based Access: Room seekers and room owners will have different roles with different permissions. Room seekers can apply for rooms, while room owners can manage listings and view applications.

6. Session Management will use JWT to manage user sessions. Tokens will be checked for validity, and users will be logged out when their token expires or is manually invalidated.