PROJECT REPORT

On

AirBnb Booking

Submitted in partial fulfilment of the requirement for the Course
BEE (22CS026) of

COMPUTER SCIENCE AND ENGINEERING B.E.
Batch-2022

in

June-2025

Submitted to:-
Mr. Rahul Sir

Submitted by:-
Ishika Goel (2210991690)

Ishneet Kaur(2210991691)

Group – G22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CHITKARA UNIVERSITY
PUNJAB

CERTIFICATE

This is to be certified that the project entitled "**Airbnb Booking**" has been submitted for the Bachelor of Computer Science Engineering at Chitkara University, Punjab during the academic semester January 2025 – June 2025 is a bonafide piece of project work carried out by "Ishika Goel(2210991690) and Ishneet Kaur(2201991691)" towards the partial fulfillment under the guidance of "Mr. Rahul Sir" and supervision.

Sign. of Project Guide :

Mr. Rahul Sir

CANDIDATE'S DECLARATION

We, **"Ishika Goel(2210991690) and Ishneet Kaur(2201991691)",** B.E.-2022 of the Chitkara University, Punjab hereby declare that the Integrated Project Report entitled **"Airbnb Booking"** is an original work and data provided in the study is authentic to the best of our knowledge. This report has not been submitted to any other Institute for the award of any other course.

Sign. of Student

Name of the Student: Ishika

Goel

ID No. 2210991690

Name of the Student: Ishneet kaur
ID No. 2210991691

Place: Chitkara University

Date: 20-05-2025

ACKNOWLEDGEMENT

It is my pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behaviour and acts during the course of study.

We express our sincere gratitude to all for providing me an opportunity to undergo Integrated Project as the part of the curriculum.

I am thankful to "Mr. Rahul Sir for his support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

I extend my sincere appreciation to *"Mr. Rahul Sir"* who provided his valuable suggestions and precious time in accomplishing our Integrated project report.

Lastly, I would like to thank the almighty and our parents for their moral support and friends with whom I shared my day-to day experience and received lots of suggestions that improve our quality of work.

**Name of the Student: Ishika
Goel
ID No. 2210991690**

**Name of the Student: Ishneet Kaur
ID No. 2210991691**

INDEX

Abstract

Airbnb is a comprehensive, user-centric vacation rental platform designed to simplify property management and enhance the booking experience for hosts and travelers. In today's travel industry, where streamlined operations and personalized experiences are crucial, property owners and guests often face challenges in managing bookings, communication, and payments. Airbnb addresses these challenges by offering a digital platform with intuitive features tailored to the needs of both hosts and travelers.

The platform allows users to manage property listings, schedule bookings, and track payment histories effortlessly. Hosts can coordinate with guests seamlessly, ensuring smooth workflows and minimizing operational overhead. Its user-friendly design ensures accessibility for users of all technical skill levels, allowing them to focus on creating memorable travel experiences instead of navigating complex systems. Features such as real-time booking updates, property availability tracking, and secure payment integration enhance organization and simplify decision-making for all users.

Airbnb is built using the MERN stack (MongoDB, Express.js, React.js, Node.js), ensuring scalability and reliable performance. The backend is powered by RESTful APIs for secure and efficient data handling, while the frontend provides dynamic and interactive dashboards for an engaging user experience. Its lightweight architecture supports deployment on both local and cloud environments, making it adaptable to a wide range of operational needs.

The mission of **Airbnb** is to empower property owners and travelers by reducing the complexity of managing rentals and bookings. Through its user-first approach, Airbnb transforms traditional vacation rental processes into a seamless, digital experience. By providing a reliable, goal-oriented platform, Airbnb supports hosts and guests in achieving hassle-free, enjoyable travel experiences.

Introduction to the project

- **Background:**

In today's dynamic travel industry, hosts and travelers often face challenges in managing vacation rentals efficiently. This project, Airbnb, addresses the critical need for an integrated rental management solution. It provides a streamlined interface to manage property listings, schedule bookings, and coordinate payments and communication, ensuring seamless workflows and enhanced travel experiences without unnecessary complexity.

- **Problem Statement:**

Traditional vacation rental platforms often suffer from being overly complex, lacking userfriendliness, or too basic, failing to meet the comprehensive needs of property owners and travelers. Many solutions require extensive customization or include unnecessary features that complicate workflows instead of enhancing convenience. This creates a significant gap between the operational needs of hosts and travelers and the functionality provided by existing tools.
For property owners and travelers seeking a reliable solution, the challenge lies in finding a platform that simplifies property management and booking processes without compromising functionality—an application that allows users to focus on what truly matters: providing exceptional stays and memorable experiences. From managing property listings to scheduling bookings, users need a streamlined system that boosts efficiency and eliminates unnecessary complexities.

Software and Hardware Requirement Specification

1.) Methods:

- Agile development methodology for iterative development.

- RESTful API design for server-client communication.

2.) Programming/Working Environment:

- Backend: Node.js, Express.js

- Frontend: React.js

- Database: MongoDB Atlas

- Version Control: Git/GitHub

- IDE: Visual Studio Code

3.) Requirements to Run the Application:

- **Hardware:** o Minimum 4GB RAM, dual-core processor.

- **Software:**

  - o Node.js   (latest LTS   version). o   Web browser (Chrome/Edge).

  - o MongoDB database server or cloud setup.

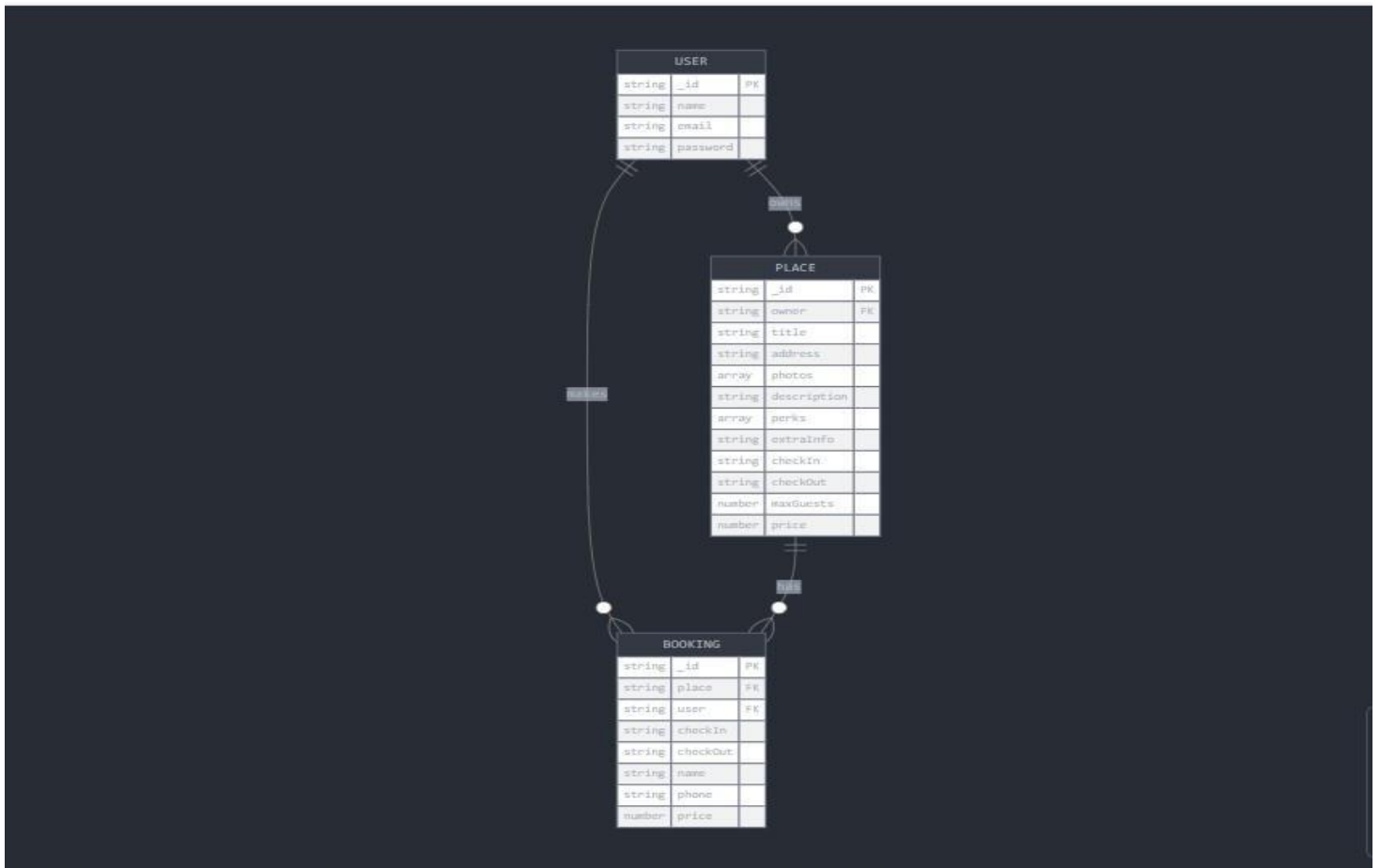# Database Analyzing, design and implementation

- **Entities:** User, Property.

- **Attributes:**

  User: ID (Primary Key), username, email, password.

  Property: ID (Primary Key), title, description, location, price, availability, userId (Foreign Key).Relationships: One-to-many relationship between User and Task.
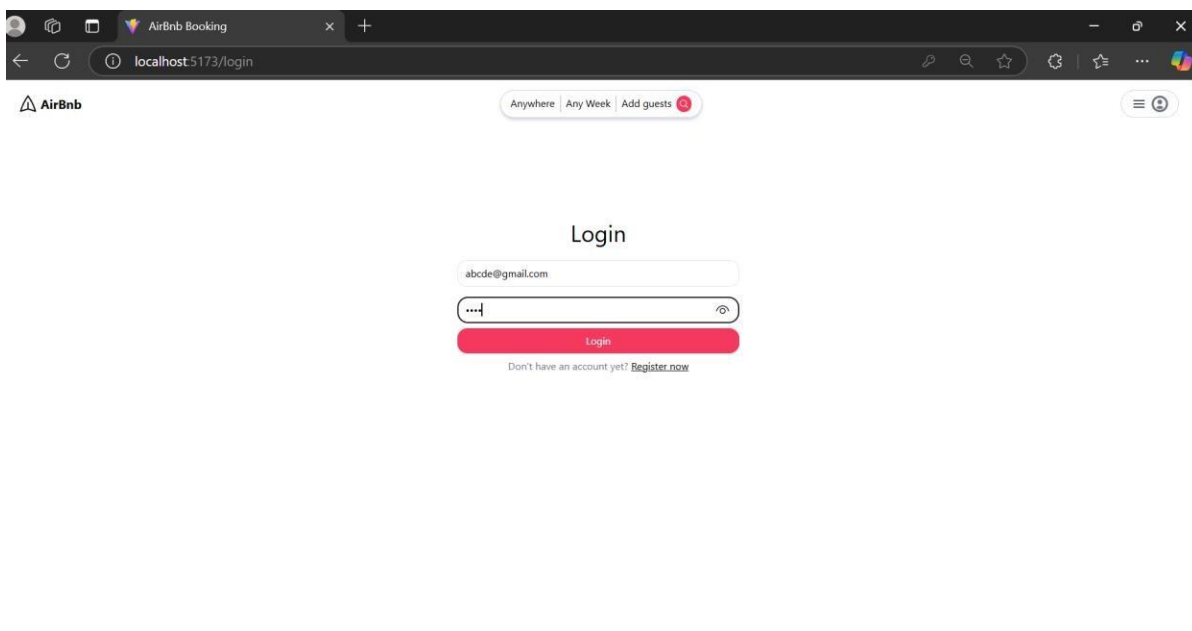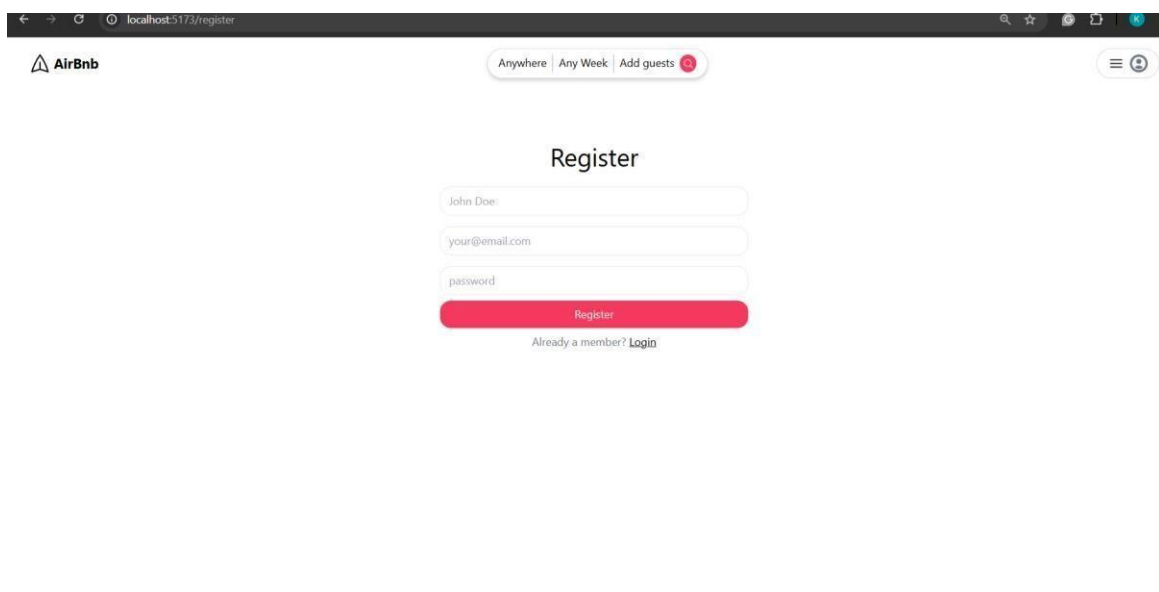
- ER Diagram: Visualize the relationships:

Program's Structure Analyzing and GUI Constructing

- **Program Structure:**

    o Client(frontend)   folder: Components,   Pages

    Server(backend) folder: Models,api •

    Snapshots:

Title
Title for your place, should be short and catchy as in advertisement

Heaven Valley

Address
Address to this place

Kulu

Photos
more = better

Add using a link... jpg   Add
                                                                                 photo

            ☁ Upload

Description
Description of the place

Very Pretty and peacefull place

Perks

△ **AirBnb**   Anywhere | Any Week | Add guests 🔍   ≡ 👤 abcde

Mal Road,Manali
My Lovely Apt
$100 per night

Goa,India
Sandy Toes, Salty Air
$100 per night

Nainital, Uttarakhand
Crystal Crest
$100 per night

Nainital, Uttarakhand
BeautifulPalce
$80 per night

△ **AirBnb**   Anywhere | Any Week | Add guests 🔍   ≡ 👤 abcde

＋ Add new place

   Heaven Valley
Very Pretty and peacefull place

My Profile · ≡ My bookings · 🏠 My accommodations

Heaven Valley
☾ 4 Nights : 🗓 2024-12-20 → 🗓 2024-12-24
▭ Total Price : $480

Sandy Toes, Salty Air
☾ 2 Nights : 🗓 2024-12-27 → 🗓 2024-12-29
▭ Total Price : $200

Heaven Valley
📍 Kulu

Your booking information:
☾ 4 Nights : 🗓 2024-12-20 🗓 2024-12-24

Total Price
$480

My Profile · ≡ My bookings · 🏠 My accommodations

Logged in as abcde (abcde@gmail.com)

Logout

Description
place is very pretty

Check-in : 8:00am
Check-out : 12:00pm
Max number of guests : 4

Price: $80 / per night

| Check In dd-mm-yyyy 📅 | Check Out dd-mm-yyyy 📅 |

Number of guests

1

Book this place

Extra Info
do not smoke inside the property

Anywhere | Any Week | Add guests 🔍

☰ 👤 qwertyuiop

Complete Your Payment

Booking Details

Jibhi

Payment Summary

Total Amount: **$100**

Proceed to Payment

← New business sandbox 📦 Sandbox

Booking Payment
$100.00

Pay with ▶ link

Or

Email
email@example.com

Card information
1234 1234 1234 1234   VISA 💳 💳 💳
MM / YY   CVC

Cardholder name
Full name on card

Country or region
India ⌄

☐ Securely save my information for 1-click checkout
Pay faster on New business sandbox and everywhere Link is accepted.

Pay

Notwithstanding the logo displayed above, when paying with a co-branded eftpos debit card, your payment may be processed through either card network.

Code-Implementation and Database Connections

○ **Database Schema Setup in MongoDB**

Airbnb uses a MongoDB database to manage users and properties. The database schema is designed to handle relationships between users (hosts and guests) and their respective properties and bookings.

Schema Explanation:

Users Collection: Stores user credentials (name, email, password) and roles (host or guest). It also references the properties listed by hosts and bookings made by guests.

• **Properties Collection:** Manages property-specific information such as title, description, location, price, availability, and the host associated with each property.

• **Bookings Collection:** Tracks booking details, including the guest, the property booked, and booking dates.

○ **RESTful API Endpoints**

The backend provides RESTful API endpoints to interact with the database.

APIs:

- o **POST /register:** Registers a new user (host or guest). o **POST /login:** Logs in the user (host or guest) using their credentials. o **POST /createproperty:** Allows a host to list a new property for booking. o **GET /getallproperties:** Retrieves a list of all properties available for booking. o **GET /getbookings:** Retrieves all bookings made by a guest (requires JWT verification).

- o **POST /generate-report:** Generates an activity report for the properties.

System Testing

○ **API Testing:** To To verify the functionality of the backend, I used Thunder Client, a lightweight tool for API testing. I tested multiple endpoints, starting with the user authentication APIs, where I validated the sign-up and login processes by inputting valid

and invalid credentials for both hosts and guests. Successful responses were generated with the correct authentication tokens, confirming the accuracy of the authentication process. I tested the full range of functionality, including property listing and booking. Each operation was validated by sending appropriate requests such as: POST /createproperty for adding a new property, GET /getallproperties for retrieving all available properties for booking, GET /getbookings for retrieving all bookings made by a guest.

Additionally, I tested the report generation feature through the POST /generate-report endpoint, ensuring that the report data was returned accurately. All tests returned the expected responses, and edge cases (such as invalid property IDs) were handled appropriately.

○ **Frontend Testing:** For frontend testing, the focus was on ensuring the application's responsiveness across different screen sizes and devices. Using Chrome's DevTools, I tested the application's performance and appearance on various devices, including mobile phones, tablets, and desktops. The goal was to ensure that the vacation rental features, such as property listings, booking, and report generation, were fully functional, and that the user interface adapted seamlessly to different screen resolutions.

This testing verified that all critical functionalities, including property management, booking process, and report generation, were easily accessible and user-friendly across all devices, providing a consistent experience for hosts and guests regardless of the device they were using.

Through these tests, I was able to confirm that both the backend and frontend of the Airbnb platform were fully functional and responsive, ensuring that users could efficiently manage bookings and properties in a seamless and user-friendly environment.

Limitations

1. Dependency on Internet:

As a cloud-based application, **StayConnect** relies on a stable internet connection for its operation. In areas with unreliable or limited internet connectivity, users may face difficulties accessing property listings, making bookings, or managing other tasks. This limitation can disrupt the experience for both hosts and guests, particularly in remote areas where internet access is inconsistent. Users in such environments may not be able to interact with the platform when offline, leading to delays in bookings, cancellations, or difficulty in managing reservations. For **StayConnect** to be truly effective in such regions, offline capabilities or local data storage solutions would be necessary to ensure uninterrupted access to the platform's features.

2. Data Privacy Concerns:

Storing sensitive user data, such as personal information, payment details, and booking histories, on a third-party cloud platform (e.g., MongoDB Atlas) raises significant concerns about data privacy and security. While Airbnb employs strong security measures, such as encryption, access control, and data backups, there is always the potential risk of data breaches or unauthorized access. User data is highly sensitive, and any breach can lead to serious consequences, including financial loss, legal ramifications, and loss of trust from users. It is crucial for Airbnb to implement stringent security protocols and ensure compliance with data protection regulations to maintain user confidence and trust .

3. Scalability and Performance Issues:

As the platform grows, with an increasing number of hosts, properties, and bookings, Airbnb may face scalability and performance challenges. With the growing volume of data and interactions, the platform may experience slower response times, especially for larger property owners or those managing multiple listings. To maintain smooth performance, Airbnb must ensure its infrastructure can handle increased load and continue providing a seamless booking experience even during peak times. This may involve optimizing backend processes, utilizing scalable cloud solutions, and continuously monitoring performance to address potential bottlenecks.

Future Scope

1.) **Adding Team Collaboration Features :** A key enhancement for Airbnb would be enabling collaboration between multiple users within the same property management team. Features such as shared task management, task assignments, internal messaging, real-time updates, and teambased scheduling could greatly improve coordination for hosts managing multiple properties or working with property managers and support staff. This feature would make Airbnb more valuable

for large property owners and rental management companies, improving operational efficiency.

2.) **Integration with Third-Party Tools :** Integrating Airbnb with third-party tools such as payment gateways, calendar systems, CRM software, and listing platforms (e.g., Booking.com, Vrbo) would enhance its functionality. This integration would allow hosts to manage all aspects of their property business from a single platform, streamlining workflows and improving efficiency. Such integration could also benefit guests by offering a seamless experience across various platforms.

3.) **Mobile Application Development:** While **Airbnb** is accessible via web browsers, developing additional features for the existing mobile application would offer greater convenience to both hosts and guests. A mobile app would allow users to manage their bookings on the go, enabling hosts to accept reservations, update listings, and communicate with guests, while travelers can easily search for accommodations, book stays, and manage reservations in real-time, providing flexibility and convenience for all users.

4.) **Advanced Reporting and Task Prioritization Features :** Airbnb could incorporate advanced analytics and AI-powered features to provide hosts with data-driven insights, such as occupancy rates, revenue predictions, and guest satisfaction trends. Additionally, task prioritization features could help hosts stay on top of critical maintenance tasks, guest inquiries, and booking deadlines.

Customizable reports, tracking guest reviews, and real-time performance metrics could empower hosts to make informed decisions and optimize their property management strategies, ensuring higher efficiency and better guest experiences.

Conclusion

Airbnb simplifies the vacation rental process by offering a streamlined, user-friendly platform for listing properties, booking stays, and managing reservations. It enhances the user experience by focusing on simplicity and usability, providing essential features that enable hosts to manage their properties effectively and guests to find and book their ideal accommodations. The platform's design emphasizes reliability and ease of use, making it simple for users to browse properties, make bookings, and track their stay details. While there is room for future enhancements and new features, Airbnb already provides a solid solution for property owners and travelers, making it an ideal tool for modernizing the vacation rental industry. Future updates, such as advanced reporting features and better integration with third-party tools, could further increase its usefulness for both hosts and guests.