

# **PROJECT REPORT ON AIRBNB PROTOTYPE WEB APPLICATION**



**Submitted by:**

**Fnu Akhilesh**

**Moksh Aggarwal**

**Shivam Mahendru**

**Department of Applied Data Science, San Jose State University**

**DATA-236 Sec 21 & 71 - Distributed Systems**

**Simon Shim**

**October 27, 2025**

## Technology Stack

Layer	Technology	Purpose
Frontend	React 18 + Vite	Single-page application with component-based UI
Build Tooling	Tailwind CSS + PostCSS	Utility-first styling and CSS processing
Routing	React Router	Client-side navigation
HTTP Client	Axios	API communication with interceptors
Backend	Node.js 18 + Express.js	REST API server
ORM	Prisma	Type-safe database access and migrations
Database	MySQL 8.0	Relational data persistence
Sessions	express-mysql-session	Server-side session storage
AI Service	Python 3.11 + FastAPI	AI concierge API
AI Framework	LangChain	LLM orchestration and prompt management
LLM Providers	Ollama, Groq, OpenAI, Google Gemini	Language model inference
External APIs	Tavily, Open-Meteo	Search and weather data enrichment
Containerization	Docker + Docker Compose	Service orchestration and deployment
Web Server	Nginx (alpine)	Static file serving for production frontend

**Data Persistence Strategy:****Database Schema (Prisma Models):**

<b>Model</b>	<b>Key Fields</b>	<b>Relationships</b>
<b>User</b>	<b>id, email, password, role (TRAVELER/OWNER)</b>	<b>Has many: properties, bookings, favorites</b>
<b>Property</b>	<b>id, title, location, pricePerNight, photos</b>	<b>Belongs to: owner (User)</b> <b>Has many: bookings, favorites</b>
<b>Booking</b>	<b>id, checkIn, checkOut, status, totalPrice</b>	<b>Belongs to: traveler (User), property</b>
<b>Favorite</b>	<b>Composite key: userId, propertyId</b>	<b>Junction table for many-to-many</b>
<b>Session</b>	<b>session_id, expires, data</b>	<b>Managed by express-mysql-session</b>

## **Abstract**

This project presents the design and development of an Airbnb Prototype Web Application that demonstrates the integration of modern full-stack web technologies with intelligent automation. The system was developed using React JS, Node.js + Express.js, and MySQL, with an AI-powered travel concierge built on Python FastAPI, LangChain, and Ollama. It supports two user roles—Traveler and Owner (Host)—to facilitate property listing, booking, and personalized trip planning.

The backend exposes secure RESTful APIs with session-based authentication, structured validation, and file handling via Multer, while the frontend offers a responsive, accessible interface created with React 18 + Vite + Tailwind CSS. The AI concierge generates custom itineraries, restaurant suggestions, and packing lists based on traveler inputs. All components are containerized with Docker Compose, ensuring seamless orchestration and portability across environments.

Overall, this project showcases the application of current web-engineering practices, cloud-ready deployment, and agentic AI integration to deliver a scalable, production-style prototype replicating Airbnb's core user experience.

## Introduction

This project is an Airbnb Prototype Web Application that was created as a component of the Node.js, ReactJS, and Agentic AI integration Lab 1 assignment. The goal was to use ReactJS for the front end, Node.js + Express.js for the back end, MySQL for persistent storage, and a Python FastAPI service to integrate an AI-powered travel concierge function into a full-stack application that showcases contemporary web development techniques.

The two main identities supported by the system are Owner (Host) and Traveler. Registering, managing profiles, searching and booking hotels, keeping track of favorites and booking history, and using an AI concierge for tailored recommendations and itineraries are all available to travelers. Through a dedicated dashboard, owners may process booking requests, list, amend, and manage their properties, as well as monitor statistics.

Reliable connection between the database and the frontend is ensured by a RESTful API layer constructed with Express.js. Secure session-based login is used for all authentication processes, and structured error handling and validation are features of essential endpoints that ensure data integrity. Multer is used to handle file uploads, and Prisma ORM is used for database transactions on the backend.

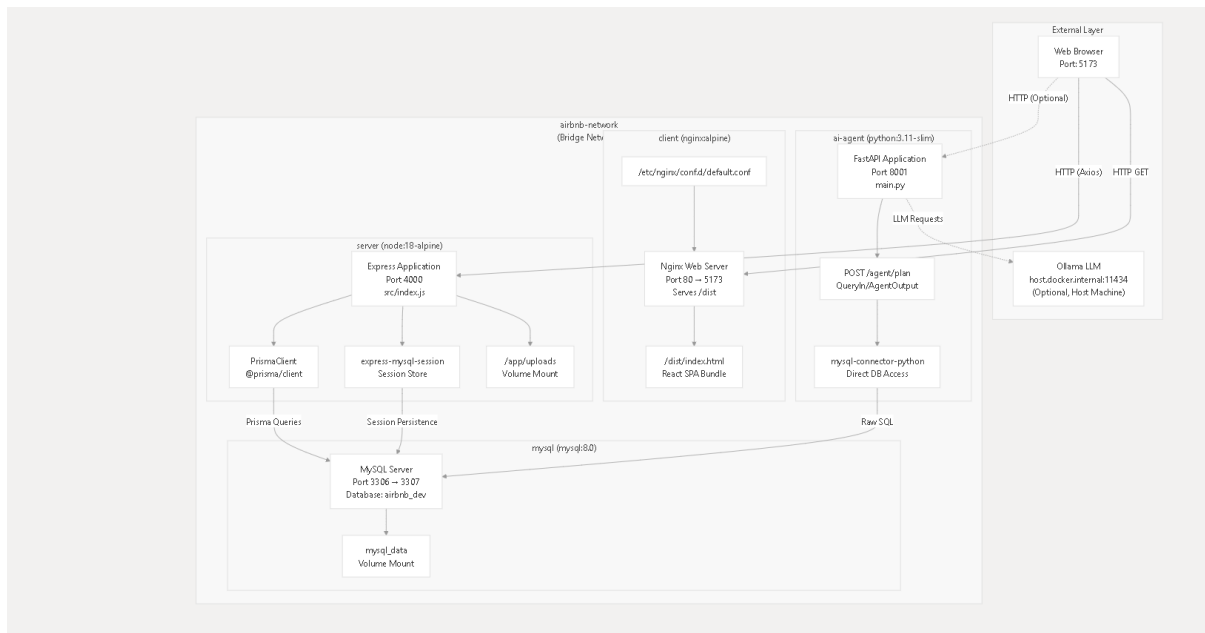
A responsive, accessible, and scalable interface is provided on the client side using the React 18 + Vite architecture, which is stylized with Tailwind CSS and controlled by Axios. By creating itineraries, restaurant recommendations, and packing lists depending on user input, the optional AI Agent—implemented in Python FastAPI + LangChain + Ollama—expands the functionality.

Docker Compose can orchestrate all services, including the frontend, backend, database, and AI agent, to ensure platform-to-platform consistency in the environment and speed up development and deployment. All things considered, this solution combines clever automation, contemporary UI design, and a strong API architecture to simulate an Airbnb-style experience that is ready for production.

## Core Services Architecture:

**Figure 1:Service Topology and Code Entities:**

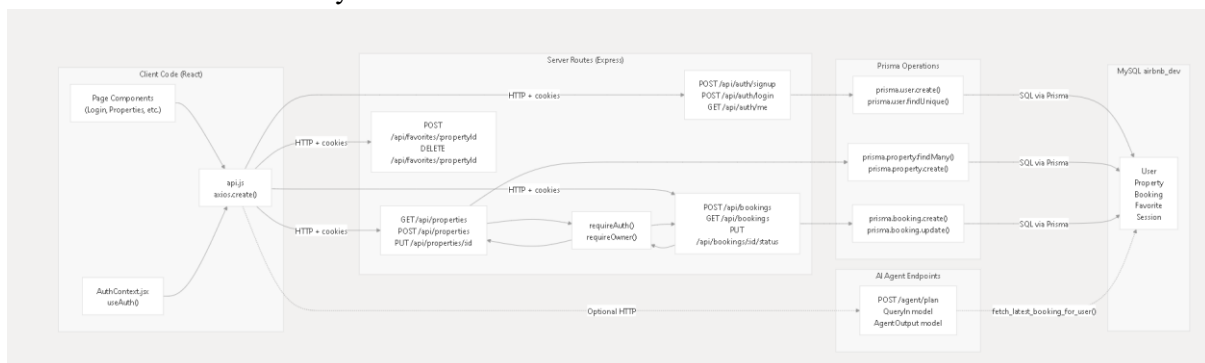
This diagram illustrates the overall system architecture of the Airbnb Prototype Web Application. It highlights the interaction between the frontend (React + Vite), backend (Node.js + Express.js + Prisma), database (MySQL 8.0), and the AI Agent (FastAPI + LangChain + Ollama). Each component functions as an independent service, connected through RESTful APIs and containerized using Docker Compose for scalable orchestration and cross-platform deployment consistency



## API Communication Patterns:

**Figure 2:Request Flow with Code Entities:**

This diagram represents the communication flow between the client, server, and database layers. User requests initiated from the React frontend are transmitted via Axios to the backend API routes in Express.js, which process, validate, and forward them to MySQL through Prisma ORM. Responses are then sent back as structured JSON data, ensuring secure, efficient, and asynchronous communication across all layers of the stack.



### Traveler Features:

- Signup – Traveler signs up with name, email ID, and password (store securely using bcrypt.js).

bnb.prototype Sign up

Sign up Log in Create a Traveler or Owner account

Name  
Your name

I am ☒ Traveler ☐ Owner (Host)

Email  
you@example.com

Password  
Minimum 6 characters

Create account

Already have an account? Log in

- Login/Logout – Implement basic session-based authentication using Express-session.

bnb.prototype Sign up

Sign up Log in Role is auto-detected on login

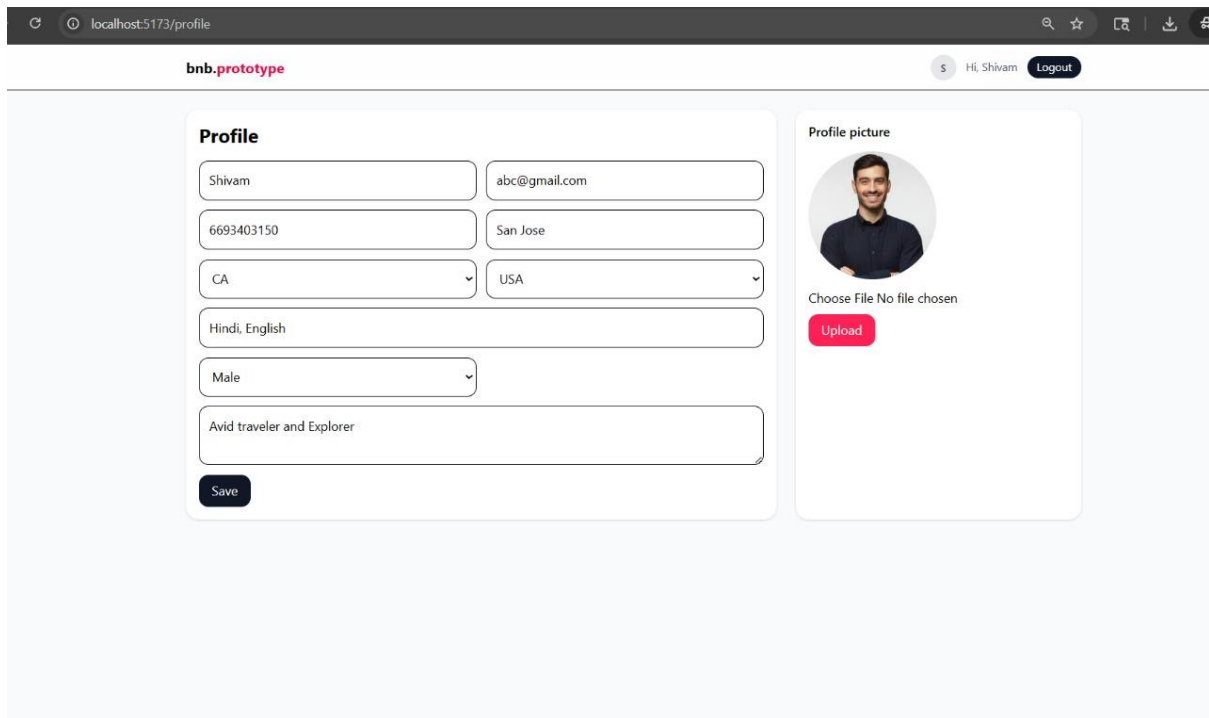
Email  
you@example.com

Password  
Minimum 6 characters

Sign in

New here? Create an account

- Profile Page - Display customer details, Upload profile picture, Update profile information (name, email, phone number, about me, city, country, languages, gender).

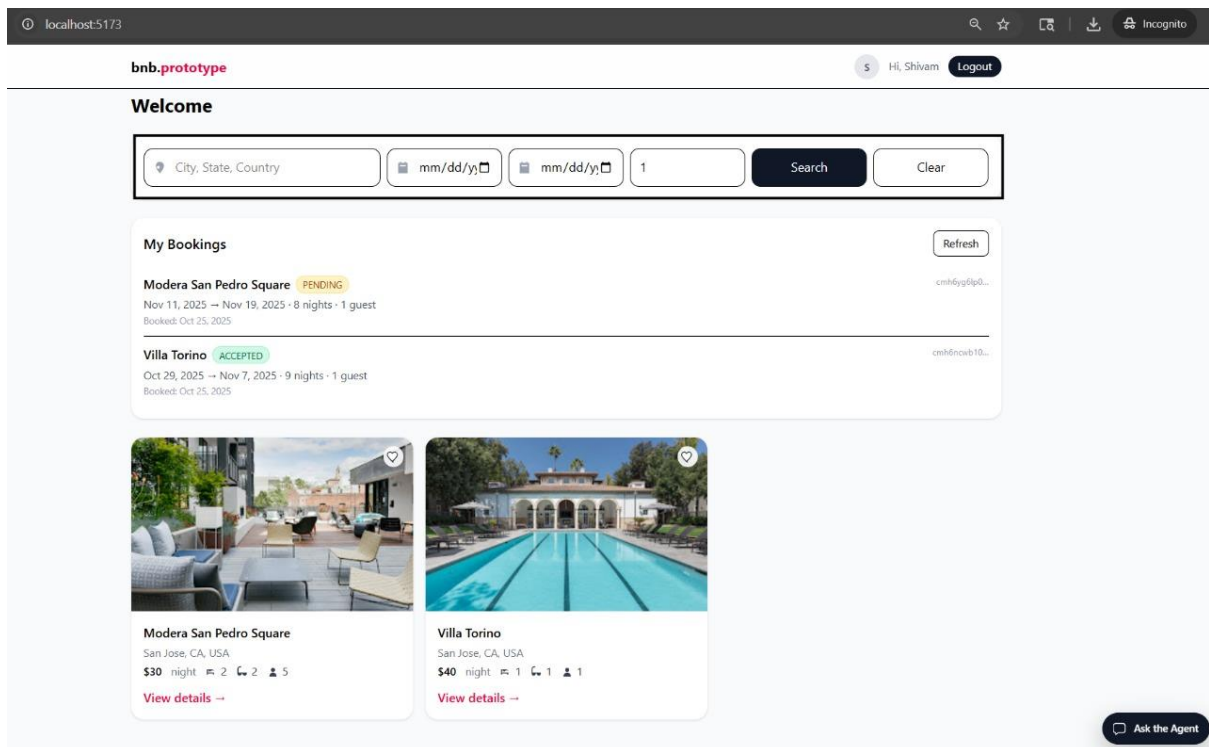


The screenshot shows a web browser at localhost:5173/profile. The page has a header with 'bnb.prototype', a user greeting 'Hi, Shivam', and a 'Logout' button. The main content area is divided into two sections:

- Profile:** A form with fields for:
  - Name: Shivam
  - Email: abc@gmail.com
  - Phone: 6693403150
  - City: San Jose
  - State: CA (dropdown)
  - Country: USA (dropdown)
  - Languages: Hindi, English
  - Gender: Male (dropdown)
  - About: Avid traveler and Explorer
  - A 'Save' button at the bottom.
- Profile picture:** A section showing a profile picture of a man, the text 'Choose File No file chosen', and an 'Upload' button.

Country field should be a dropdown list, state should be abbreviated

- Property Search Page/Dashboard – Search available properties by:
  - Location
  - Start Date, End Date
  - Number of Guests



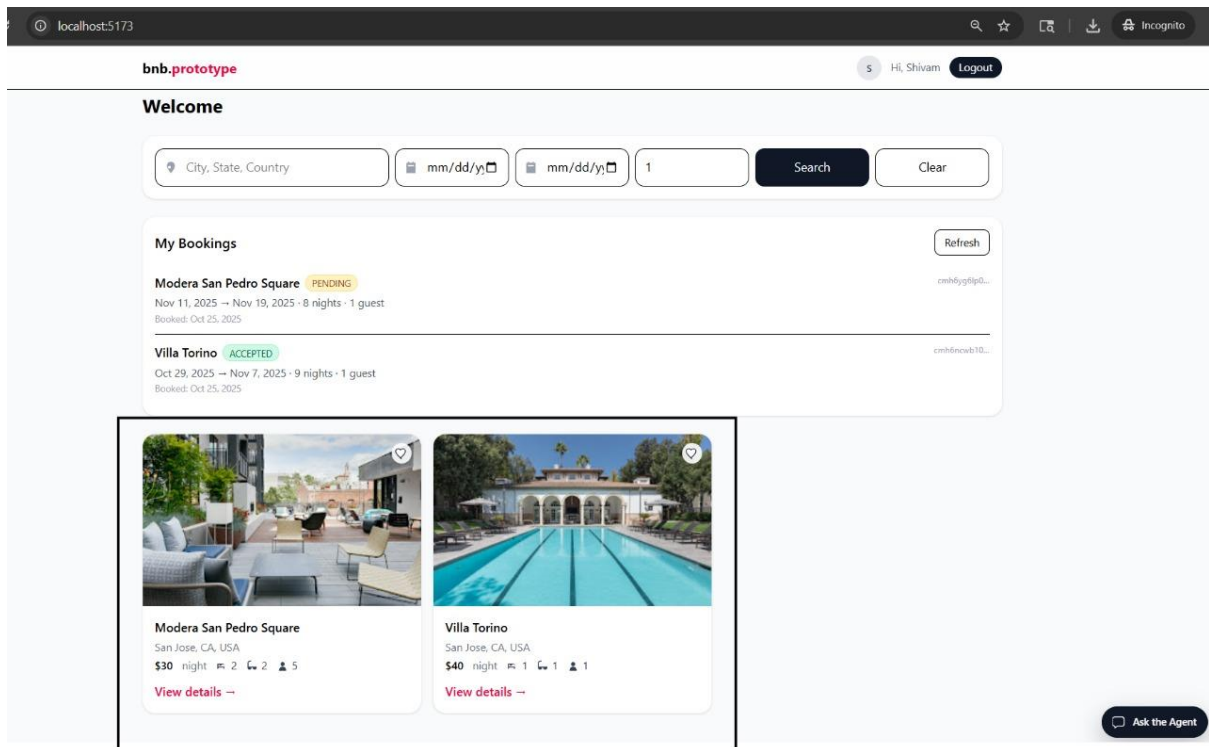
The screenshot shows a web browser at localhost:5173. The page has a header with 'bnb.prototype', a user greeting 'Hi, Shivam', and a 'Logout' button. The main content area is divided into three sections:

- Welcome:** A search bar with fields for:
  - City, State, Country
  - Start Date: mm/dd/yy
  - End Date: mm/dd/yy
  - Number of Guests: 1
  - A 'Search' button and a 'Clear' button.
- My Bookings:** A section with a 'Refresh' button and a list of bookings:
  - Modera San Pedro Square** (PENDING): Nov 11, 2025 → Nov 19, 2025 - 8 nights - 1 guest. Booked: Oct 25, 2025.
  - Villa Torino** (ACCEPTED): Oct 29, 2025 → Nov 7, 2025 - 9 nights - 1 guest. Booked: Oct 25, 2025.
- Property Cards:** Two cards showing property details:
  - Modera San Pedro Square:** San Jose, CA, USA. \$30 night. 2 bedrooms, 2 bathrooms, 5 guests. View details →
  - Villa Torino:** San Jose, CA, USA. \$40 night. 1 bedroom, 1 bathroom, 1 guest. View details →

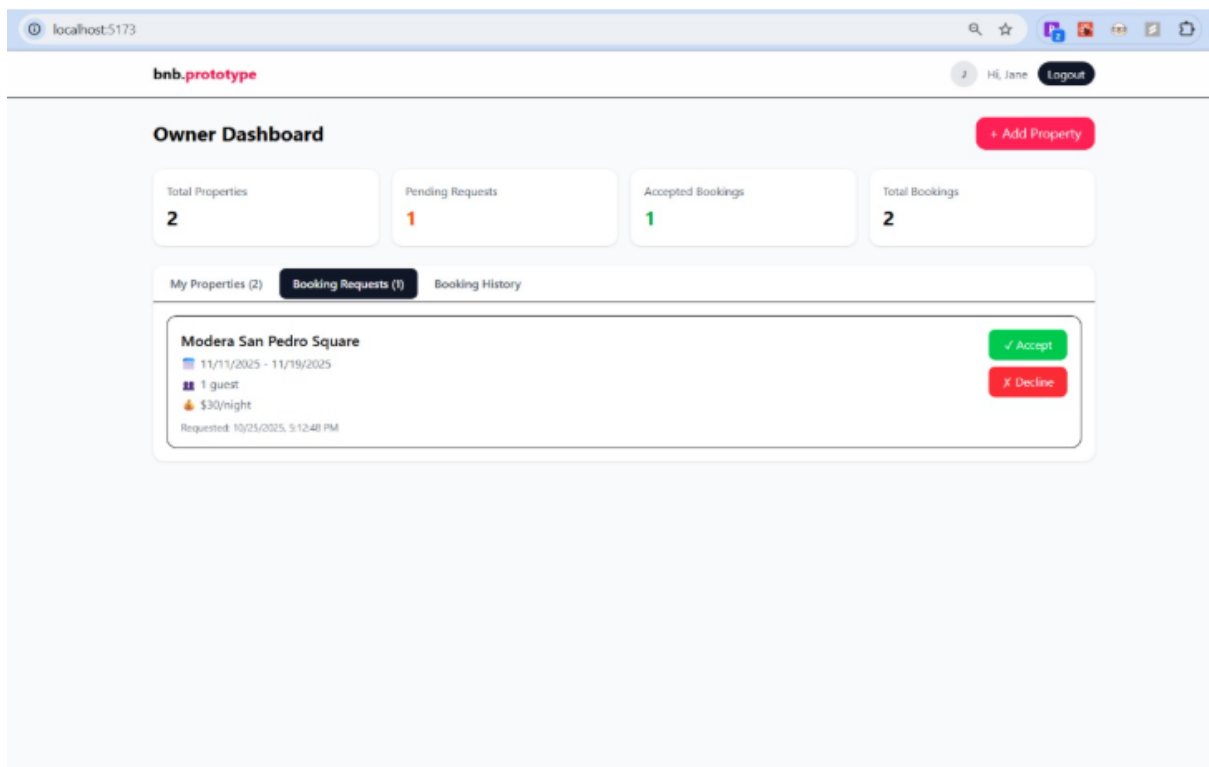
An 'Ask the Agent' button is located at the bottom right.



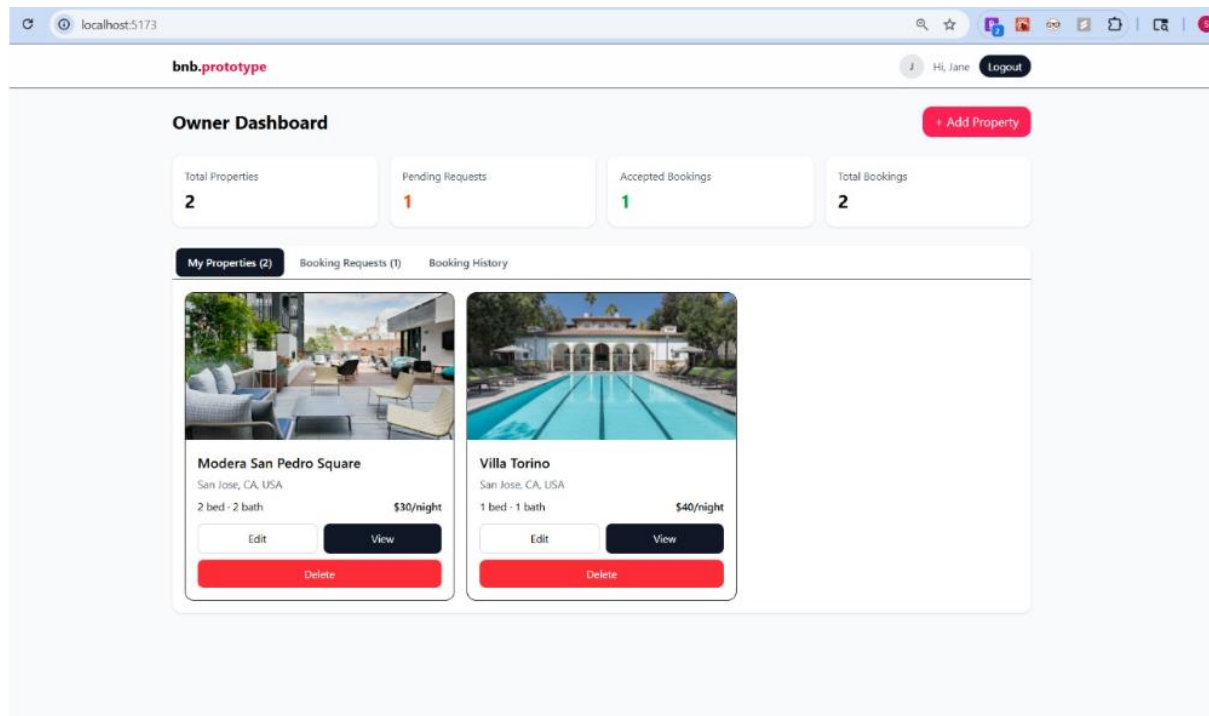
- Property Details View – View details such as property name, type, amenities, pricing, bedrooms, bathrooms, and availability. Allow booking.



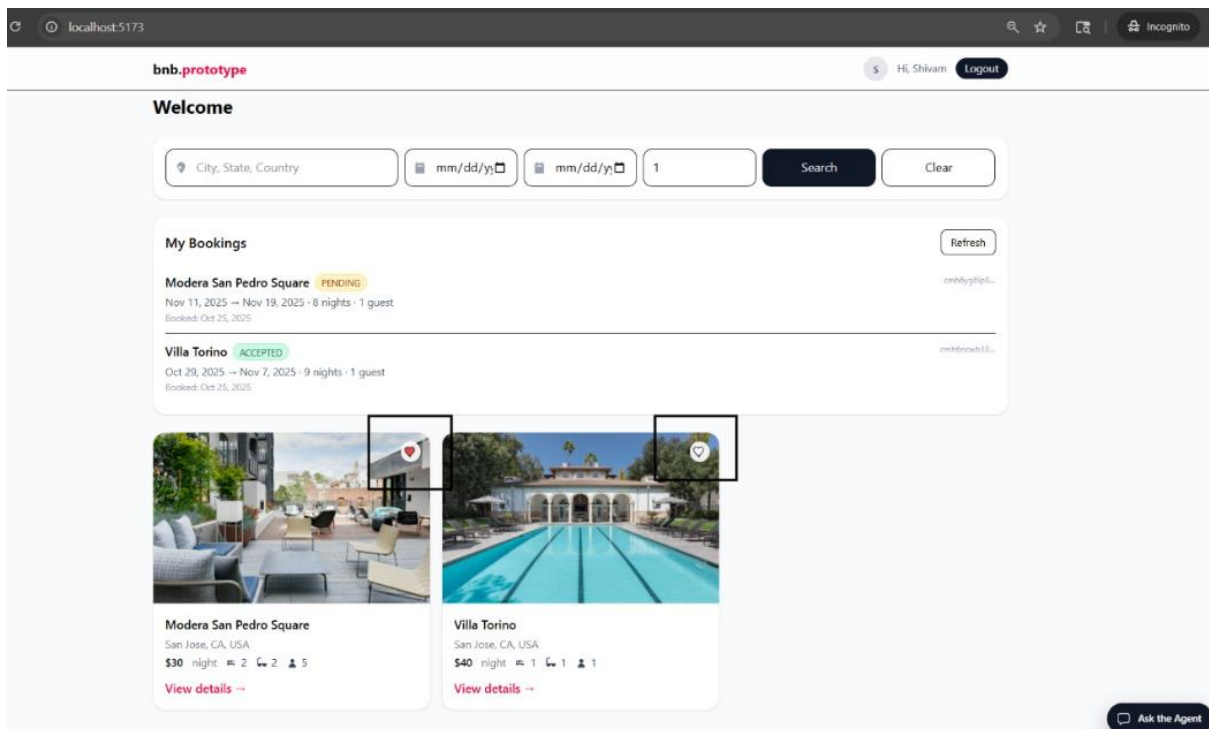
- Booking Flow – Create a booking request for selected dates/guests. A booking starts in Pending state until the Owner responds. Travelers can view Pending, Accepted, and Cancelled bookings.



## Property Owner Dashboard:



- Favourites - Mark properties as favorites, Display a favourites tab



userId	propertyId
cmh6gf5840000a2dkis75xf3c →	cmh6i8bn40002grc7bgj01a5z →

- Traveler History – Display a history tab for any previous bookings/trips taken.

localhost:5173

bnb.prototype
Hi, Shivam Logout

Welcome

My Bookings
Refresh

Modera San Pedro Square
PENDING
Nov 11, 2025 – Nov 19, 2025 • 8 nights • 1 guest
Booked: Oct 25, 2025
cmh6y6p0...

Villa Torino
ACCEPTED
Oct 29, 2025 – Nov 7, 2025 • 9 nights • 1 guest
Booked: Oct 25, 2025
cmh6ncwb10...

Modera San Pedro Square

San Jose, CA, USA

\$30 night 2 2 5

[View details →](#)

Villa Torino

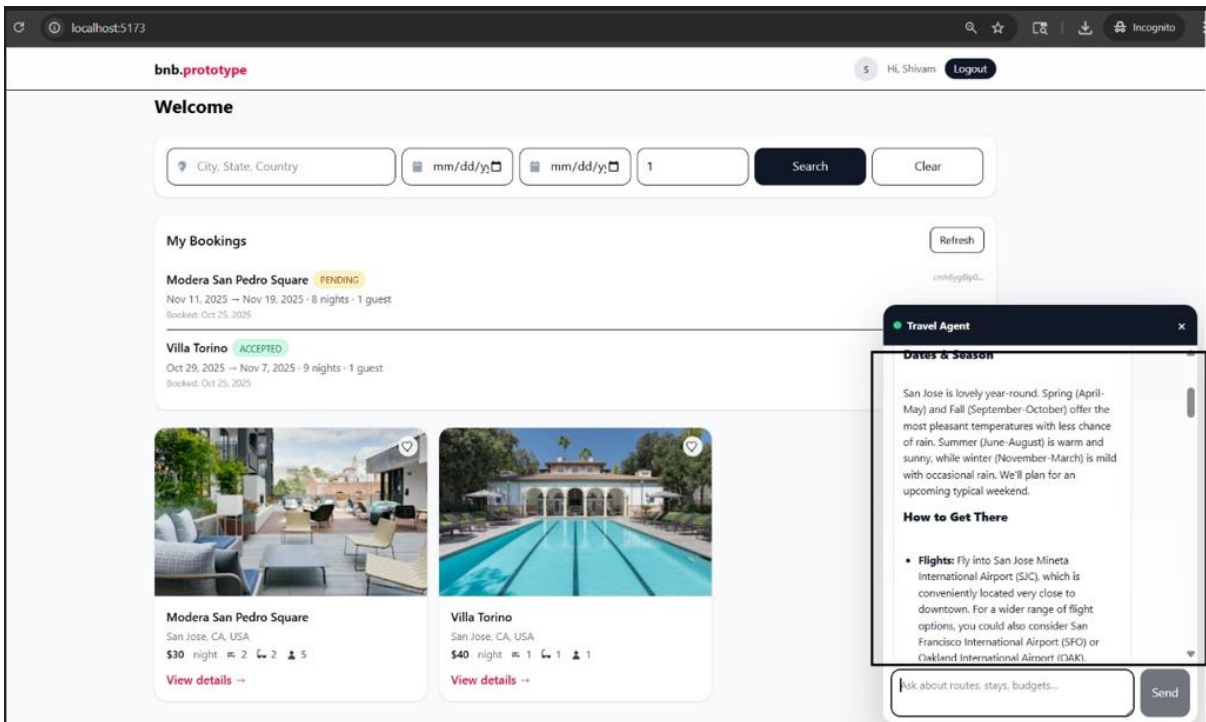
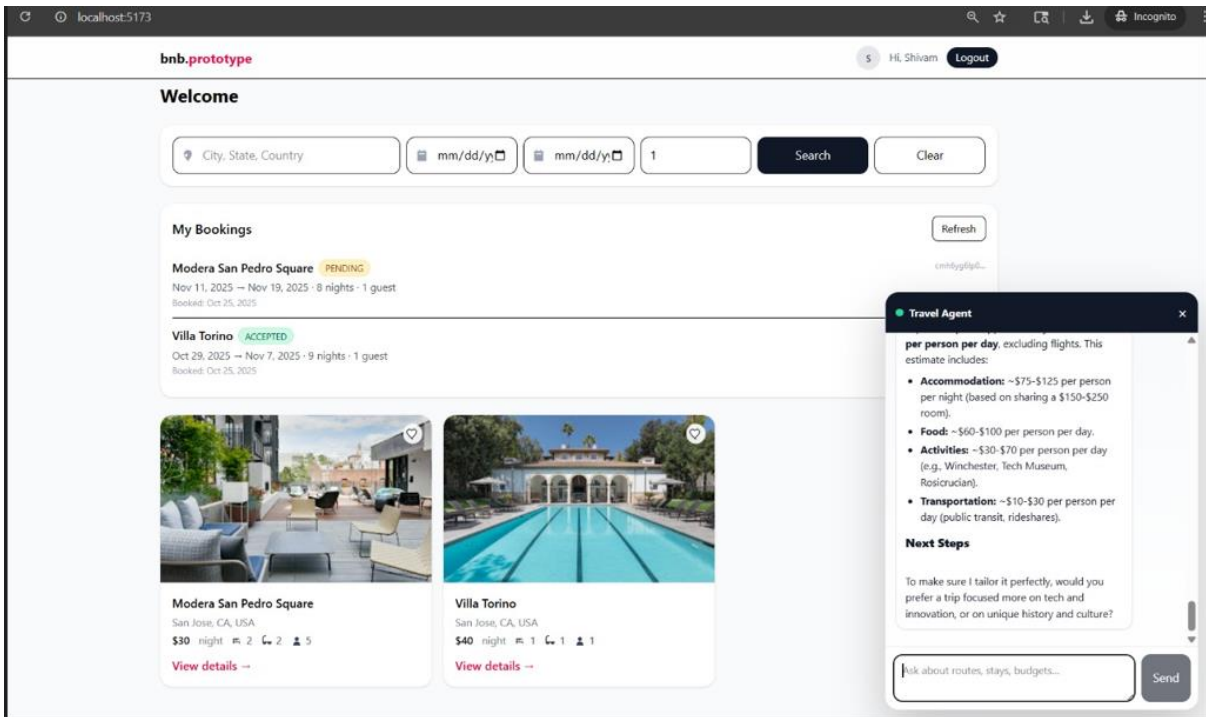
San Jose, CA, USA

\$40 night 1 1 1

[View details →](#)

Ask the Agent

- Agent AI: the agent’s button should be accessible to the user, from the dashboard



## Property Owner Profile Page:

The screenshot shows a web browser window with the address bar displaying 'localhost:5173/profile'. The page title is 'bnb.prototype'. The user is logged in as 'Hi, Jane' with a 'Logout' button. The main content area is titled 'Profile' and contains a form with the following fields:

- First Name: Jane
- Last Name: xyz@gmail.com
- Phone Number: 6693403150
- City: San Jose
- State: CA (dropdown)
- Country: USA (dropdown)
- Language: Spanish
- Gender: Female (dropdown)
- About me: (text area)
- Save button

To the right of the form is a 'Profile picture' section. It features a circular profile picture of a woman. Below the picture, it says 'Choose File No file chosen' and has an 'Upload' button.

## New Property Form:

The screenshot shows a web browser window with the address bar displaying 'localhost:5173/property/add'. The page title is 'bnb.prototype'. The user is logged in as 'Hi, Jane' with a 'Logout' button. The main content area is titled 'Basic Information' and contains a form with the following fields:

- Property Name \*: e.g., Cozy Downtown Apartment
- Property Type \*: Apartment (dropdown)
- Price per Night (\$) \*: 100
- Description: Describe your property... (text area)
- Location section:
  - Address: Street address
  - City: City
  - State: Select State (dropdown)
  - Country: USA (dropdown)
- Property Details section:
  - Bedrooms \*: 1
  - Bathrooms \*: 1
  - Max Guests \*: 1

bnb.prototype

Hi, Jane

Logout

City

Select State

Country

USA

Property Details

Bedrooms \*

1

Bathrooms \*

1

Max Guests \*

1

Amenities

☐ WiFi

☐ Kitchen

☐ Washer

☐ Dryer

☐ Air conditioning

☐ Heating

☐ TV

☐ Pool

☐ Gym

☐ Parking

☐ Pets allowed

Availability

Available From

mm/dd/yyyy

Available To

mm/dd/yyyy

Photos

Upload Photos (max 10)

Choose Files No file chosen

Cancel

Create Property

localhost:5173/property/add

Backend:

User table:

id	role	name	email	passwordHash	phone	about	city	state	cou...	languages	gender	avatarUrl	createdAt
cmh6gf5840000...	TRAVELER	Shivam	abc@gmail...	\$2a\$10\$jT5EcHyqBhr/s9u2jlmxebyQ...	6693403150	NULL	San J...	CA	NULL	Hindi, En...	Male	/uploads/_...	2025-10-25 15:48:07.349
cmh615j1q0000...	OWNER	Jane	xyz@gmail...	\$2a\$10\$y4L9xroINznSXZ1CnsI7u0a9K...	6693403150	NULL	San J...	CA	NULL	Spanish	Female	/uploads/_...	2025-10-25 16:36:37.934
cmh6x2woe0000...	TRAVELER	User 2	user@gma...	\$2a\$10\$.72MYqI/wp4SMyxMbzsEeFMF...	6693403159	NULL	San J...	CA	NULL	Spanish	Female	/uploads/_...	2025-10-25 23:34:29.869

Booking table:

id	travelerId	propertyId	startDate	endDate	guests	status	createdAt
cmh6ncwb10001hw2vj9mea0jr	cmh6gf584... →	cmh618b... →	2025-10-29 07:0...	2025-11-07 0...	1	ACCEPTED	2025-10-25 19:02:19.788
cmh6y6g61p00048fjokzv48d8s	cmh6gf584... →	cmh6yfc... →	2025-11-11 08:0...	2025-11-19 0...	1	PENDING	2025-10-26 00:12:48.877

Property table:

id	ownerId	name	type	description	address	city	state	country	pricePerNight	bedrooms	bathrooms	maxGuests
cmh6i8bn40002grc7bgj0ia5z	cmh615j1q0000grc7qu01sd2v → Villa Torino	Apartment	Cozy Apartment	in San Jose Downtown.	39 W Julian Street	San Jose	CA	USA	40	1	1	1
cmh6yfcby00028fjok1kz33yu	cmh615j1q0000grc7qu01sd2v → Modera San Pedro Square	Apartment	Luxury Apartments	in the heart of San Jose	28 N Almaden Avenue	San Jose	CA	USA	30	2	2	5

bedrooms	bathrooms	maxGuests	amenities	photos	availableFrom	availableTo	createdAt
1	1	1	["WiFi", "Dryer", "TV", "Par...	["/uploads/_...	2025-10-30 00:00:00.000	2025-11-08 00:00:00.000	2025-10-...
2	2	5	["WiFi", "Dryer", "TV", "Par...	["/uploads/_...	2025-10-27 00:00:00.000	2025-12-09 00:00:00.000	2025-10-...

Session table:

session_id	expires	data
DZDo4BE1gUBXG953RCp82C_UPcafcrIU	1762033215	{"cookie":{"originalMaxAge":604800000,"expires":"2025-11-01T21:13:29.738Z", "...
Elg7rmHdz9cYft7AjiS1CD4VLhdBy9u7	1762030779	{"cookie":{"originalMaxAge":604800000,"expires":"2025-11-01T16:36:37.973Z", "...
SikeR76_E0ig43RMe2WVwc78G1e-NhUe	1762046497	{"cookie":{"originalMaxAge":604800000,"expires":"2025-11-02T00:21:59.855Z", "...
WDiPzZ2E7Fi4rJ7IvgBTaocC5ONdJhTS	1762016350	{"cookie":{"originalMaxAge":604800000,"expires":"2025-11-01T16:59:10.140Z", "...
_J9Z_BRmyv-CLj7qkpgYg1bnYEsEOvBS	1762013629	{"cookie":{"originalMaxAge":604800000,"expires":"2025-11-01T16:04:48.794Z", "...
bzZ4anSDrGCD6znyg1HPCS8kvEsX02GV	1762066664	{"cookie":{"originalMaxAge":604800000,"expires":"2025-11-02T01:26:57.777Z", "...

## API Documentation:

# Airbnb Prototype API

1.0.0 OAS 3.0

REST API for Traveler/Owner flows — session-auth, profiles, properties, bookings, favorites.

Servers

/

Authorize

## Auth

POST /api/auth/signup Traveler/Owner signup

Parameters

Cancel

Reset

No parameters

Request body required

application/json

Edit Value Schema

```
{
  "name": "Shivam",
  "email": "shivam@gmail.com",
  "password": "shivam123",
  "role": "OWNER"
}
```



Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:4800/api/auth/signup' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "name": "Shivan",
    "email": "shivam@gmail.com",
    "password": "shivam123",
    "role": "OWNER"
  }'
```

Request URL

http://localhost:4800/api/auth/signup

Server response

Code	Details
201	<div><div>Response body</div><pre>{   "id": "cmh7imwn800i7bwSuae8qlxh",   "name": "Shivan",   "email": "shivam@gmail.com",   "role": "OWNER",   "avatarUrl": null }</pre><div>Download</div><div>Response headers</div><pre>access-control-allow-credentials: true access-control-allow-origin: http://localhost:5173 connection: keep-alive content-length: 189 content-type: application/json; charset=utf-8 date: Sun, 26 Oct 2025 09:37:54 GMT etag: W/"6d-EDr3zY3biQXsAuWYznhXVgvEq4o" keep-alive: timeout=5 vary: Origin x-powered-by: Express</pre></div>

Responses

Code	Description	Links
201	<div>Created</div> <div><div>Media type</div><div>application/json</div><div>Controls Accept header.</div><div>Example Value   Schema</div></div>	No links

Code	Description	Links
	<pre>{   "id": "string",   "role": "TRAVELER",   "name": "string",   "email": "string",   "avatarUrl": "string" }</pre>	
409	Email already exists	No links

POST

/api/auth/login Login (session cookie)

🔒 ^

Parameters

Cancel

Reset

No parameters

Request body required

application/json

Edit Value

Schema

```
{
  "email": "moksh@gmail.com",
  "password": "moksh123"
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:4000/api/auth/login' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "email": "moksh@gmail.com",
```

```

    "password": "moksh123"
  },
}

```

**Request URL**

http://localhost:4000/api/auth/login



**Server response**

Code	Details
200	<p><b>Response body</b></p> <pre> {   "id": "cmh7idzhc00007bw5g9gh8u81",   "role": "TRAVELER",   "name": "Moksh",   "email": "moksh@gmail.com" } </pre> <p><a href="#">Download</a></p> <p><b>Response headers</b></p> <pre> access-control-allow-credentials: true access-control-allow-origin: http://localhost:5173 connection: keep-alive content-length: 93 content-type: application/json; charset=utf-8 date: Sun, 26 Oct 2025 09:59:15 GMT etag: W/"5d-zBvC3B/w86/1sd5etUzlgVCQTuc" keep-alive: timeout=5 vary: Origin x-powered-by: Express </pre>

**Responses**

Code	Description	Links						
200	<p>Logged in (session cookie set)</p> <p><b>Headers:</b></p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>Set-Cookie</td> <td></td> <td>string</td> </tr> </tbody> </table>	Name	Description	Type	Set-Cookie		string	No links
Name	Description	Type						
Set-Cookie		string						
401	Invalid credentials	No links						

---

POST /api/auth/logout Logout (destroy session)  

**Parameters** Cancel

No parameters

Execute
Clear

**Responses**

#### Curl

```
curl -X 'POST' \
  'http://localhost:4000/api/auth/logout' \
  -H 'accept: */*' \
  -d ''
```

#### Request URL

http://localhost:4000/api/auth/logout

#### Server response

##### Code Details

204

##### Response headers

```
access-control-allow-credentials: true
access-control-allow-origin: http://localhost:5173
connection: keep-alive
date: Sun, 26 Oct 2025 09:58:50 GMT
keep-alive: timeout=5
vary: Origin
x-powered-by: Express
```

#### Responses

Code	Description	Links
204	Logged out	No links



## Bookings

POST	/api/bookings	Create booking (Traveler) -> PENDING	🔒	✓
GET	/api/bookings/mine	List my bookings (Traveler)	🔒	✓
GET	/api/bookings/incoming	Owner — incoming PENDING requests for my properties	🔒	✓
PUT	/api/bookings/{id}/status	Owner — Accept or Cancel a request	🔒	✓

## Favorites

GET	/api/favorites	List my favorite properties	🔒	✓
POST	/api/favorites/{propertyId}	Add property to favorites	🔒	✓
DELETE	/api/favorites/{propertyId}	Remove property from favorites	🔒	✓

## Properties

POST /api/properties Create property (OWNER)  

Parameters Cancel Reset

No parameters

Request body required application/json

Edit Value Schema

```
{
  "name": "Villa Torino",
  "type": "Apartment",
  "description": "Near Downtown",
  "address": "29 W Julian St",
  "city": "San Jose",
  "state": "CA",
  "country": "USA",
  "pricePerNight": 100,
  "bedrooms": 2,
  "bathrooms": 2,
  "maxGuests": 4,
  "amenities": [
    "string"
  ],
  "availableFrom": "2025-10-26",
  "availableTo": "2025-10-29"
}
```

Execute Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:4000/api/properties' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "name": "Villa Torino",
    "type": "Apartment",
    "description": "Near Downtown",
    "address": "29 W Julian St",
    "city": "San Jose",
    "state": "CA",
    "country": "USA",
    "pricePerNight": 100,
    "bedrooms": 2,
    "bathrooms": 2,
    "maxGuests": 4,
    "amenities": [
```

```

    "string"
  },
  "availableFrom": "2025-10-26",
  "availableTo": "2025-10-29"
}

```

#### Request URL

http://localhost:4000/api/properties

#### Server response

Code	Details
200	<p><b>Response body</b></p> <pre> {   "id": "cmh7iry0z00037bw57r7u71d2",   "ownerId": "cmh7imwn800017bw5uae8q1xh",   "name": "Villa Torino",   "type": "Apartment",   "description": "Near Downtown",   "address": "29 W Julian St",   "city": "San Jose",   "state": "CA",   "country": "USA",   "pricePerNight": 100,   "bedrooms": 2,   "bathrooms": 2,   "maxGuests": 4,   "amenities": [     "string"   ],   "photos": [],   "availableFrom": "2025-10-26T00:00:00.000Z",   "availableTo": "2025-10-29T00:00:00.000Z",   "createdAt": "2025-10-26T09:41:49.954Z" } </pre> <p><a href="#">Download</a></p> <p><b>Response headers</b></p> <pre> access-control-allow-credentials: true access-control-allow-origin: http://localhost:5173 connection: keep-alive content-length: 436 content-type: application/json; charset=utf-8 date: Sun, 26 Oct 2025 09:41:49 GMT etag: W/"1b4-hE/yVQmoaKuphDrLrIOtellEvu0" keep-alive: timeout=5 vary: Origin x-powered-by: Express </pre>

#### Responses

Code	Description	Links
200	<p>Created property</p> <p>Media type</p> <p><b>application/json</b></p> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre> {   "id": "string",   "ownerId": "string",   "name": "string", </pre>	No links

Code	Description	Links
	<pre>"type": "string", "description": "string", "address": "string", "city": "string", "state": "string", "country": "string", "pricePerNight": 0, "bedrooms": 0, "bathrooms": 0, "maxGuests": 0, "amenities": [   "string" ], "photos": [   "string" ], "availableFrom": "2025-10-27", "availableTo": "2025-10-27" }</pre>	
401	Not authenticated	No links
403	Requires OWNER role	No links

GET

/api/properties Search properties

POST

/api/properties/{id}/photos Upload property photos (OWNER)

Parameters

Cancel

Reset

Name	Description
<b>id</b> <small>required</small>	
string	cmh7iry0z00037bw57r7u71d2
(path)	

Request body required

multipart/form-data

photos

array<string>

Choose File

2024-03-26.webp

-

Add string item

☐ Send empty value

Execute

Clear

## Responses

## Curl

```
curl -X 'POST' \
  'http://localhost:4800/api/properties/cmh7iry8z80037bw57r7u71d2/photos' \
  -H 'accept: */*' \
  -H 'Content-Type: multipart/form-data' \
  -F 'photos=@2024-03-26.webp;type=image/webp'
```

## Request URL

http://localhost:4800/api/properties/cmh7iry8z80037bw57r7u71d2/photos

## Server response

## Code      Details

200

## Response body

```
{
  "id": "cmh7iry8z80037bw57r7u71d2",
  "ownerId": "cmh7imwn800017bw5uae8qlxh",
  "name": "Villa Torino",
  "type": "Apartment",
  "description": "Near Downtown",
  "address": "29 W Julian St",
  "city": "San Jose",
  "state": "CA",
  "country": "USA",
  "pricePerNight": 100,
  "bedrooms": 2,
  "bathrooms": 2,
  "maxGuests": 4,
  "amenities": [
    "string"
  ],
  "photos": [
    "/uploads/properties/1761471863599-102636828.webp"
  ],
  "availableFrom": "2025-10-26T00:00:00.000Z",
  "availableTo": "2025-10-29T00:00:00.000Z",
  "createdAt": "2025-10-26T09:41:49.954Z"
}
```

[Download](#)

## Response headers

```
access-control-allow-credentials: true
access-control-allow-origin: http://localhost:5173
connection: keep-alive
content-length: 486
content-type: application/json; charset=utf-8
date: Sun, 26 Oct 2025 09:44:23 GMT
etag: W/"1e6-vuIaptsU2o0i0CJNMiWJIshvwa0"
keep-alive: timeout=5
vary: Origin
x-powered-by: Express
```

## Responses

Code	Description	Links
200	Updated property with photo URLs	<i>No links</i>
401	Not authenticated	<i>No links</i>



Code	Description	Links
403	Must be the property owner	No links
404	Not found	No links

PUT

/api/properties/{id} Update property (OWNER)

🔒 ⬆

Parameters

Cancel Reset

Name	Description
id <span>★ required</span>	
string (path)	cmh7iry0z00037bw57r7u71d2

Request body required

application/json

Edit Value Schema

```

{
  "name": "Villa Torino",
  "type": "Apartment",
  "description": "Near Downtown",
  "address": "29 W Julian St",
  "city": "San Jose",
  "state": "CA",
  "country": "USA",
  "pricePerNight": 200,
  "bedrooms": 2,
  "bathrooms": 2,
  "maxGuests": 4,
  "amenities": [
    "string"
  ],
  "photos": [],
  "availableFrom": "2025-10-26T00:00:00.000Z",
  "availableTo": "2025-10-29T00:00:00.000Z",
  "createdAt": "2025-10-26T09:41:49.954Z"
}

```

Execute Clear

Responses

Curl

```

curl -X 'PUT' \
  'http://localhost:4000/api/properties/cmh7iry0z00037bw57r7u71d2' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \

```

```

-d '{
  "id": "cmh7iry0z00037bw57r7u71d2",
  "ownerId": "cmh7imwn800017bw5uae8qlxh",
  "name": "Villa Torino",
  "type": "Apartment",
  "description": "Near Downtown",
  "address": "29 W Julian St",
  "city": "San Jose",
  "state": "CA",
  "country": "USA",
  "pricePerNight": 200,
  "bedrooms": 2,
  "bathrooms": 2,
  "maxGuests": 4,
  "amenities": [
    "string"
  ],
  "photos": [],
  "availableFrom": "2025-10-26T00:00:00.000Z",
  "availableTo": "2025-10-29T00:00:00.000Z",
  "createdAt": "2025-10-26T09:41:49.954Z"
}'

```

#### Request URL

http://localhost:4000/api/properties/cmh7iry0z00037bw57r7u71d2

#### Server response

Code	Details
200	<p><b>Response body</b></p> <pre> {   "id": "cmh7iry0z00037bw57r7u71d2",   "ownerId": "cmh7imwn800017bw5uae8qlxh",   "name": "Villa Torino",   "type": "Apartment",   "description": "Near Downtown",   "address": "29 W Julian St",   "city": "San Jose",   "state": "CA",   "country": "USA",   "pricePerNight": 200,   "bedrooms": 2,   "bathrooms": 2,   "maxGuests": 4,   "amenities": [     "string"   ],   "photos": [],   "availableFrom": "2025-10-26T00:00:00.000Z",   "availableTo": "2025-10-29T00:00:00.000Z",   "createdAt": "2025-10-26T09:41:49.954Z" } </pre> <p><a href="#">Download</a></p> <p><b>Response headers</b></p> <pre> access-control-allow-credentials: true access-control-allow-origin: http://localhost:5173 connection: keep-alive content-length: 436 content-type: application/json; charset=utf-8 date: Sun, 26 Oct 2025 09:50:10 GMT etag: W/"1b4-37LgyUSA35xzt+thT1LDWy3mk08" keep-alive: timeout=5 vary: Origin x-powered-by: Express </pre>

#### Responses

Code	Description	Links
200	Updated property	No links
401	Not authenticated	No links
403	Must be the owner	No links
404	Not found	No links

GET /api/properties/{id} Get property by id
🔒
⌵

Parameters
Cancel

Name	Description
id <span>• required</span>	
string (path)	cmh7iry0z00037bw57r7u71d2

Execute
Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:4800/api/properties/cmh7iry0z00037bw57r7u71d2' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:4800/api/properties/cmh7iry0z00037bw57r7u71d2
```

Server response

Code	Details
200	<p>Response body</p> <pre>{   "id": "cmh7iry0z00037bw57r7u71d2",   "ownerId": "cmh7imwn800017bw5uae8qixh",   "name": "Villa Torino",   "type": "Apartment",   "description": "Near Downtown",   "address": "29 W Julian St",   "city": "San Jose",   "state": "CA",   "country": "USA",   "pricePerNight": 200, }</pre>

Code	Details	
	<pre>       "bedrooms": 2,       "bathrooms": 2,       "maxGuests": 4,       "amenities": [         "string"       ],       "photos": [],       "availableFrom": "2025-10-26T00:00:00.000Z",       "availableTo": "2025-10-29T00:00:00.000Z",       "createdAt": "2025-10-26T09:41:49.954Z"     }   </pre>	Download
	<b>Response headers</b> <pre> access-control-allow-credentials: true access-control-allow-origin: http://localhost:5173 connection: keep-alive content-length: 436 content-type: application/json; charset=utf-8 date: Sun, 26 Oct 2025 09:50:30 GMT etag: W/"1b4-37LgyUSA35xzt+thT1l0Ny3ek08" keep-alive: timeout=5 vary: Origin x-powered-by: Express   </pre>	
<b>Responses</b>		
Code	Description	Links
200	Property	No links
	<p>Media type</p> <div>application/json</div> <p>Controls Accept header.</p> <p><b>Example Value</b>   <b>Schema</b></p> <pre> {   "id": "string",   "ownerId": "string",   "name": "string",   "type": "string",   "description": "string",   "address": "string",   "city": "string",   "state": "string",   "country": "string",   "pricePerNight": 0,   "bedrooms": 0,   "bathrooms": 0,   "maxGuests": 0,   "amenities": [     "string"   ],   "photos": [     "string"   ],   "availableFrom": "2025-10-27",   "availableTo": "2025-10-27" }   </pre>	

## Users

GET

/api/users/me

Get my profile

🔒

⌵

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

curl -X 'GET' \\\n'http://localhost:4000/api/users/me' \\\n-H 'accept: application/json'

Request URL

http://localhost:4000/api/users/me

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{\n  "id": "cmh7imwn800017bw5uae8q1xh",\n  "role": "OWNER",\n  "name": "Shivan",\n  "email": "shivan@gmail.com",\n  "phone": null,\n  "about": null,\n  "city": null,\n  "state": null,\n  "country": null,\n  "languages": null,\n  "gender": null,\n  "avatarUrl": "/uploads/avatars/1761472295084-758150375.png",\n  "createdAt": "2025-10-26T09:37:54.885Z"\n}</pre></div><div>Download</div></div> <div><div>Response headers</div><div><pre>access-control-allow-credentials: true\naccess-control-allow-origin: http://localhost:5173\nconnection: keep-alive\ncontent-length: 287\ncontent-type: application/json; charset=utf-8\ndate: Sun, 26 Oct 2025 09:52:09 GMT\netag: W/"11f-VHNVv8cDVGfUr56qB125fEHuePM"\nkeep-alive: timeout=5\nvary: Origin\nx-powered-by: Express</pre></div></div>

Responses

Code	Description	Links
200	Profile <div>Media type application/json Controls Accept header.</div> <div>Example Value Schema</div> <pre>{  "id": "string",  "role": "TRAVELER",  "name": "string",  "email": "string",  "avatarUrl": "string"}</pre>	No links
401	Not authenticated	No links

PUT /api/users/me Update my profile

Parameters

Cancel

Reset

No parameters

Request body required

application/json

Edit Value Schema

```
{  "name": "Shivan",  "email": "shivan@gmail.com",  "phone": "string",  "about": "string",  "city": "string",  "state": "string",  "country": "string",  "languages": "string",  "gender": "Male"}
```

Execute

Clear

## Responses

### Curl

```
curl -X 'PUT' \
  'http://localhost:4000/api/users/me' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "name": "Shivam",
    "email": "shivam@gmail.com",
    "phone": "string",
    "about": "string",
    "city": "string",
    "state": "string",
    "country": "string",
    "languages": "string",
    "gender": "Male"
  }'
```

### Request URL

http://localhost:4000/api/users/me

### Server response

Code	Details
200	<div><div><b>Response body</b></div><div><pre>{   "id": "cmh7imwn800817bw5uae8q1xh",   "role": "OWNER",   "name": "Shivam",   "email": "shivam@gmail.com",   "phone": "string",   "about": "string",   "city": "string",   "state": "string",   "country": "string",   "languages": "string",   "gender": "Male",   "avatarUrl": "/uploads/avatars/1761472295084-758150375.png",   "createdAt": "2025-10-26T09:37:54.885Z" }</pre></div><div><a href="#">Download</a></div></div> <div><div><b>Response headers</b></div><div><pre>access-control-allow-credentials: true access-control-allow-origin: http://localhost:5173 connection: keep-alive content-length: 313 content-type: application/json; charset=utf-8 date: Sun, 26 Oct 2025 09:54:04 GMT etag: W/"139-J5RgKjnESXYvHrWMUnHJ1sqR6gQ" keep-alive: timeout=5 vary: Origin x-powered-by: Express</pre></div></div>

### Responses

Code	Description	Links
200	Updated user	<a href="#">No links</a>
401	Not authenticated	<a href="#">No links</a>

POST
/api/users/avatar
Upload my avatar

Parameters
Cancel
Reset

No parameters

Request body
required
multipart/form-data

avatar
string(\$binary)
Choose File
Jordi.png

☐ Send empty value

Execute
Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:4000/api/users/avatar' \
  -H 'accept: */*' \
  -H 'Content-Type: multipart/form-data' \
  -F 'avatar=@Jordi.png;type=image/png'
```

Request URL

http://localhost:4000/api/users/avatar

Server response

Code	Details
200	<div> Response body <div> { "__id": "cmh71mw880017bw5uae8q1xh", "role": "OWNER", "name": "Shivan", "email": "shivan@gmail.com", "phone": null, "about": null, "city": null, "state": null, "country": null, "languages": null, "gender": null, "avatarUrl": "/uploads/avatars/1761472295004-758150375.png", "createdAt": "2025-10-26T09:37:54.885Z" } </div> Download </div>

Response headers



Code	Details	
	<pre>access-control-allow-credentials: true access-control-allow-origin: http://localhost:5173 connection: keep-alive content-length: 287 content-type: application/json; charset=utf-8 date: Sun, 26 Oct 2025 09:51:35 GMT etag: W/"11f-VHNWv8cDVGfUr56q8L25fEHuePW" keep-alive: timeout=5 vary: Origin x-powered-by: Express</pre>	
Responses		
Code	Description	Links
200	Updated user	No links
400	No file uploaded	No links
401	Not authenticated	No links

Schemas	
User	
Property	
Booking	

## Non-Functional Requirements:

The Airbnb Prototype Web Application also fulfills key non-functional objectives that ensure performance, scalability, and usability. The system's **responsive design** adapts seamlessly to desktops, tablets, and mobile devices using Tailwind CSS breakpoints and component-based layouts. **Accessibility** is enhanced through semantic HTML, proper labeling, and contrast-aware styling, while **scalability** is achieved via Docker Compose, allowing effortless orchestration of the frontend, backend, database, and AI microservices across environments.

The screenshot displays the Airbnb Prototype Web Application interface. At the top, the logo "bnb.prototype" is visible, along with a user profile icon labeled "S" and a "Logout" button. Below the header, there is a search form with a location input field (placeholder: "City, state, country"), two date pickers (placeholder: "mm/dd/yyyy"), and a dropdown menu (placeholder: "1"). A "Search" button and a "Clear" button are positioned below the form. The "My Bookings" section features a "Refresh" button and lists two bookings. The first booking, "Modera San Pedro Square", is marked "PENDING" and shows dates "Nov 11, 2025 → Nov 19, 2025 · 8 nights · 1 guest", booked on "Oct 25, 2025". The second booking, "Villa Torino", is marked "ACCEPTED" and shows dates "Oct 29, 2025 → Nov 7, 2025 · 9 nights · 1 guest", also booked on "Oct 25, 2025". At the bottom, there is a banner image and a button labeled "Ask the Agent".



### 101 Aprtments

San Jose, CA, USA

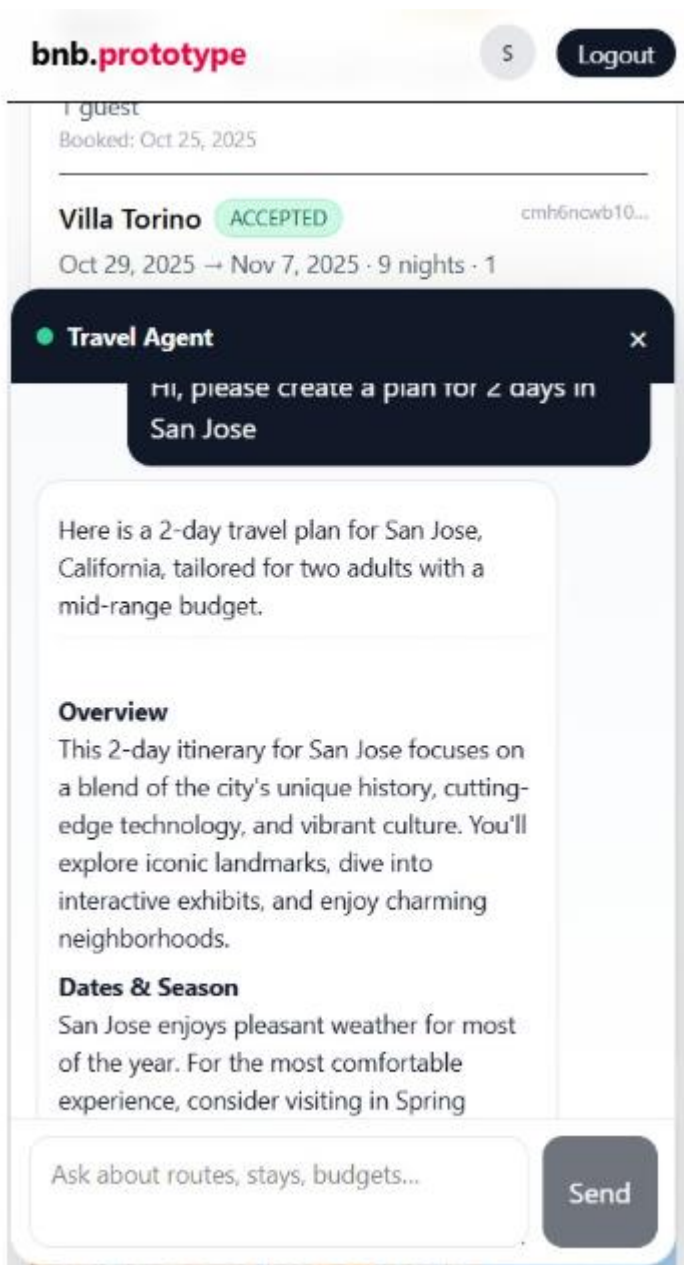
\$25 night 🏠 1 🚗 1 👤 1

[View details →](#)



### 27 North Apartments

💬 Ask the Agent



## Conclusion

The creation of the Airbnb Prototype Web Application is more than just the conclusion of a software project; it is the result of combining modern full-stack engineering with intelligent automation. Instead of just copying Airbnb's features, this system reimagines them using a contemporary technology framework that combines AI-driven personalization, scalability, and reactivity.

The project demonstrates how several frameworks may function well together inside a single architecture to produce a flawless digital experience by combining ReactJS, Node.js, Express.js, MySQL, and FastAPI. From the clever AI agent to the RESTful backend and interactive frontend, every layer was designed to communicate seamlessly, reflecting the complexity of dispersed systems found in the real world.

The AI concierge turns straightforward reservations into contextually informed travel experiences, elevating the customer experience from transactional to intelligent. This ecosystem is kept reproducible, portable, and prepared for deployment at any scale through the use of Docker Compose.

In the end, this project represents how online applications have changed over time, moving from static sites to self-governing, learning systems that anticipate user wants. In the age of digital travel, it is evidence of how careful engineering and inventiveness can turn code into an experience, bridging the gap between technology and human convenience.

## REFERENCES

1. React Documentation – <https://react.dev>
2. Express Framework – <https://expressjs.com>
3. Prisma ORM Docs – <https://www.prisma.io/docs>
4. FastAPI Docs – <https://fastapi.tiangolo.com>
5. MySQL Docs – <https://dev.mysql.com/doc>

Github Link:

[https://github.com/aggarmoksh/airbnb\\_prototype](https://github.com/aggarmoksh/airbnb_prototype)