

Abstract

In an age where digital assistance simplifies every aspect of life, planning travel should be no exception. *Travzi* is a modern web application that aims to enhance the travel experience by offering real-time, location-based recommendations for tourist-friendly destinations. It leverages cutting-edge AI through the Gemini API, authenticates users securely with Clerk, and delivers a seamless, responsive interface built using Next.js.

Designed with simplicity and utility in mind, Travzi is structured for a single-user role — the Explorer. This user-centric platform enables tourists to discover nearby attractions, hidden gems, and culturally significant sites with just a few clicks. Whether someone is a solo backpacker, a family vacationer, or a casual day-tripper, Travzi helps make every trip more meaningful and efficient.

Table Of Contents

Chapter	Title	Page No
A	Abstract	i
B	List of Figures	iv
1	Introduction	1
1.1	Background and Context	2
1.2	Purpose	3
1.3	Functional Features	4
1.4	Significance of the Project	5
1.5	Organization of Report	6
2	Related work	8
2.1	Literature Survey	9
2.2	Gap Identified	10
3	Problem Statement and Objectives	11
3.1	Problem Statement	12
3.2	Objectives	13
3.3	Scope	14
4	Overall Description	16
4.1	Product Perspective	17
4.2	Product Functions	18
4.3	User Characteristics	19
4.4	Hardware and Software Requirements	20
5	System Design	21
5.1	Proposed System	22
5.2	Block Diagram	23
5.3	Component Diagram	24
5.4	Use case Diagram	25
5.5	Data Flow Diagram	26
5.6	Class Diagram	27
5.7	Sequence Diagram	28
5.8	Database Design	29

6	Implementation Details	30
6.1	Project Modules	31
6.2	General Installation Steps	32
7	Testing and Validation	33
7.1	Testing	34
7.1.1	Test cases for page1	34
7.1.2	Test cases for page2	34
7.1.3	Test cases for page3	35
7.2	Validations	36
8	Result, Analysis and Conclusion	37
8.1	Result	38
8.2	Snapshots of work done	39
8.3	Analysis	41
8.4	Conclusion	42
8.5	Future Scope	43
9	References	44
9.1	Journals referred	45
9.2	References	46
	Appendices	47
	Plagiarism Report	48

List of Figures

Sr. No	Name of Figure	Page No.
1	Fig 5.1 Proposed System	22
2	Fig 5.2 Block Diagram	23
3	Fig 5.3 Component Diagram	24
4	Fig 5.4 Use case Diagram	25
5	Fig 5.5 Data Flow Diagram	26
6	Fig 5.6 Class Diagram	27
7	Fig 5.7 Sequence Diagram	28
8	Fig 5.8 Database Diagram	29
9	Fig 7.1 Test case 1	34
10	Fig 7.2 Test case 2	34
11	Fig 7.3 Test case 3	35
12	Fig 8.1 App interface	39
13	Fig 8.2 App interface	39
14	Fig 8.3 App interface	40

CHAPTER 1

INTRODUCTION

1.1 Background and Context

Today's travelers are faced with a daunting challenge of discovering trustworthy and up-to-date information and recommendations. The typical blogs and guidebooks which we have traditionally relied upon either make irrelevant or general recommendations based on a "set" map or provide limited perspectives on one or two fixed areas which most users will not ever visit. Travzi utilizes AI and location data to navigate the traveler toward real-time, personalized travel recommendations based on proximity, real-time popularity, and user behavior.

The AI technology analyzes reviews, trends, and foot traffic, to recommend the highest-rated and best-performing locations to you at any given point in time. When users chose to travel to places which might provide some hidden gems or trending places visited by local, user-to-user recommendations are created using your future predictions based on time, location, and behavior history. The app will source local information from local sources and includes things tourists would generally be disinterested or unaware of, allowing first-time travelers to be led by a local's experience.

The benefits of the app are not limited to comparisons or a single trip, but it increases the ability to travel, maximize the possibilities of our plans, and open up to accidental travel explorations. The app's algorithm will learn the choices of the user, based on their ability to obtain their preferences, making suggestions according to comfort-level, i.e. quiet cafés over bustling markets when the preference has existed. Real-time updates allow for refined recommendations according to changes in closures, weather, and allotting for how busy popular or trending places have potential waiting times in their options. Travzi, or a similar app, turns exploration from temporary to smart and more pleasurable.

1.2 Purpose

1. Offer intelligent, dynamic travel recommendations.
2. Provide a fully seamless experience for users through easy UI secure login.
3. Leverage AI (Gemini API) to understand user preferences and situational context.

1.3 Functional Features

- As a user (Explorer), you are able to:
 - Allow the app to access your location.
 - Receive immediate AI-generated suggestions for places nearby that are tourist-friendly.
 - Search for specific types of places, such as nature, historical, and entertainment.
 - Filter suggestions by category, to fit your needs.
 - View information about all suggested places, including description and distance.
 - Enjoy a responsive and dynamic interface on both mobile and desktop devices.

1.4 Significance of the project

- **Automates Travel Discovery:** Automatically suggests places nearby using real-time data, eliminating the need for manual search and planning.
- **Enhances User Convenience:** Simplifies the exploration process by providing relevant suggestions instantly based on current location and user preferences.
- **Lightweight AI Integration:** Demonstrates how AI-powered recommendation engines can be seamlessly integrated into simple web applications to deliver valuable real-time user experiences.

1.5 Organization of report

Objective:

1. **Offer Instant Travel Recommendations:** Provide real-time suggestions for nearby tourist spots based on user location.
2. **Integrate Smart Personalization via AI:** Use the Gemini API to suggest places tailored to user interests, context, and preferences.
3. **Ensure Seamless and Secure User Experience:** Implement Clerk authentication to offer users a fast and secure sign-in.

Workflow:

1. **User Login and Location Access**
 - User signs in via Clerk authentication.
 - Browser requests permission to access the user's location.
 - Upon approval, the app fetches GPS coordinates.
2. **AI-Based Suggestion Retrieval**
 - Coordinates are sent to the Gemini API.
 - Gemini processes data and returns nearby tourist spots.
3. **Display of Suggestions**
 - Results are categorized (e.g., nature, heritage, leisure).
 - Users view and explore detailed information for each place.
4. **Search and Filter Functionality**
 - Users can apply filters to refine suggestions based on interest.
 - Manual search is also supported.
5. **Performance and Accessibility**
 - Designed to perform well across devices.
 - Interface ensures usability even in low-bandwidth environments.

Technology stack:

Frontend:

- Next.js – for responsive UI and performance.

- shadcn/UI and Tailwind CSS – for clean and consistent styling.

Backend:

- Next.js API routes – handling client requests and responses.
- No dedicated database as it's stateless; relies on APIs.

Authentication:

- Clerk – manages login, sessions, and user data securely.

AI Integration:

- Gemini API – provides real-time, personalized travel suggestions.

Deployment:

- Vercel – for frontend hosting.
- GitHub – version control and CI/CD.

Benefits and Impact:

1. **Smart Travel Planning:** Replaces static sources with intelligent, real-time recommendations.
2. **Effortless Exploration:** Minimizes planning time and enhances user spontaneity.
3. **Scalable Architecture:** Demonstrates an effective model for future AI-integrated travel tools.
4. **User-Centric Design:** Focuses on a single role (Explorer) to deliver a tailored, efficient experience.

CHAPTER 2

RELATED WORK

2.1 Literature Survey

There are many apps that help with travel; Google Maps, TripAdvisor, Lonely Planet, etc. They tell you what to see, report user experiences and can provide navigation. However, they also mostly rely on static searches and manually curated lists, which require the users to take the time to sort through large volumes of content. Furthermore, they usually don't integrate real-time context regarding user preferences, current location, time of day, and trip objectives.

Recent research in smart tourism has highlighted the increased importance of AI in improving the travel experience. While the research has often been found in meta-analyses, it has been suggested that AI recommendation systems, together with geolocation, user profiles, etc., can effectively address travel satisfaction, and assist users in an efficient decision-making process.

Travzi adopts a position above these other tools by utilizing Gemini AI and real-time access to location to offer contextual, intelligent, and on-the-go recommendations. In doing so, it helps to address obvious usability and personalization issues and improve the user experience beyond search tools.

2.2 Gap Identified

Current Systems	Gaps Identified
Google Maps, TripAdvisor	Require manual search; lack personalized, real-time suggestions
Travel Blogs & Forums	Static content; not location-aware or AI-powered
Comprehensive Tour Apps	Often bulky, trying to serve too many use cases; complex for casual users
Navigation Systems	Focused on routes/directions, not place discovery
General Recommender Tools	Lack location-context and tourism-specific intelligence

Travzi addresses these gaps by providing a focused, AI-powered, and location-aware platform designed specifically for casual explorers seeking instant travel suggestions.

CHAPTER 3

PROBLEM STATEMENT AND OBJECTIVES

3.1 Problem Statement

Today's travelers need fast, personalized suggestions to visit unknown areas. They want to explore new places, but many sites are filled with too much information or are unusable for real-time location-based discovery. Instead, travelers often waste time sifting through irrelevant or outdated information. Current platforms fail to provide an intelligent and seamless travel experience. Therefore, Travzi solves this issue by providing personalized dynamic suggestions using AI to tap into millions of options based on your location and interest, while also eliminating noise and providing a focused, efficient, and effective way to discover impromptu or spontaneous experiences.

3.2 Objectives

Thought can be beneficial with a well-defined set of goals in regard to enhancing the travel planning experience for the typical user. The main objectives are as follows:

1. Create a responsive and user-friendly web application that provides meaningful suggestions based on any type of device or screen size, whether users are accessing these on mobile or desktop.
2. Utilize Gemini AI technology to provide smart recommendations for nearby attractions based on user preferences, interests, time of day, and current user location.
3. Using Clerk, allow for secure and efficient user authentication for users to sign in quickly, while creating a safe experience for users with minimal registration requirements.
4. Allow for travel planning to be slightly less burdensome on the users by automating the search for their particular location and generating suggestions for them.
5. Create filters for different categories and search functions so users can search by a particular type of place matching their interests.
6. Create a stateless and scalable application that does not have a heavy backend so the excess will make it lightweight but still effective.

3.3 Scope

1. Key Features:

- Explorer-Centric Interface: Travzi's focus on the "Explorer" role keeping the user interface clean and informing the user without requiring extensive interaction buddy ideal for quick discovery.
- Real-Time AI Suggestions: Travzi will leverage Gemini API to offer dynamic, location aware suggestions that are always being ramped up, selected, dropped, and rolled into facilitated reflective discovery at the telematic level of responsiveness shift.
- Secure and Self-authenticated: The clerk class of user authentication offered a seamless, secure sign-in process for all users to ensure security and exclusivity for verified users to have access to well-defined features.
- Category Based Filtering: Travzi has a scalable feature that allows users to map their intuitive and immersive recommendations into identifiable categories of nature, history, shopping, and leisure.
- No Backend Persistence: Travzi operates as a stateless, intangible platform service, using real-time API calls instead of standard two persistent databases.

2. Product (MVP):

The Minimum Viable Product provides key travel discovery features:

- User login via Clerk
- User location through browser GPS API
- Smart recommendations via Gemini
- Displaying categories and clickable options with short details
- Responsive UI optimized for all types of devices

3. Advanced Features:

Following MVP validation, there may be additions to features such as:

- Favorite place bookmarking
- Travel history tracking & review
- Use of Google Maps for direction
- Offline caching of last list fetched suggestions
- Support for multilingual and disability needs.

4. Scalability and Adaptability:

The lightweight architecture of Travzi is scalable for:

- Increasing user base geographic area.
- Integration of more powerful AI APIs for smarter suggestions.
- Expansion into mobile apps from the same backend logic in the future.
- Theming or suggestions based on travel mood (e.g., adventure; relaxation).

CHAPTER 4

OVERALL DESCRIPTION

4.1 Product Perspective

Travzi offers a focused and efficient solution for spontaneous travel discovery. It is designed as a lightweight, AI-powered platform tailored for a single user role: the Explorer. The system has been developed with the following perspectives in mind:

1. **Single-Role Application:** Travzi is intentionally built around one user role (Explorer) to reduce complexity and deliver a streamlined experience.
2. **Web-Based Accessibility:** As a responsive web application, it requires no installation and can be accessed via any modern web browser on desktop or mobile.
3. **AI-Driven Recommendations:** It utilizes the Gemini API to fetch intelligent and contextual suggestions based on user location and preferences.
4. **Secure Authentication:** Clerk handles user sign-in securely with minimal friction, supporting a trustworthy user environment.
5. **Stateless and Scalable Architecture:** The platform operates without persistent backend storage, relying on real-time API interactions to remain lightweight and fast.
6. **Future-Ready Design:** The architecture allows for smooth integration with additional APIs and features such as offline support, multilingual capabilities, and map-based navigation in future versions.

4.2 Product Functions

The core functionality of Travzi revolves around providing an intelligent, personalized travel experience for users. Each function is designed to be efficient, intuitive, and value-driven:

1. **AI-Powered Suggestions:** Utilizes the Gemini API to generate real-time, personalized travel recommendations based on the user's current geographic location.
2. **Location-Based Discovery:** Automatically detects the user's GPS coordinates to provide nearby tourist-friendly spots without manual input.
3. **Category-Wise Listings:** Organizes recommended places into intuitive categories such as nature, heritage, entertainment, and dining, helping users easily find what they're looking for.
4. **Detailed Place Information:** Offers users concise yet informative details for each location, including a brief description, approximate distance, and category.
5. **Search and Filter Capability:** Allows users to refine the suggestions using filters based on interest, type of attraction, or keyword search.
6. **Responsive UI Interaction:** Delivers a smooth browsing experience across all devices, ensuring usability on both mobile and desktop platforms.

4.3 User Characteristics

Travzi is tailored specifically for a single type of user — the Explorer. The following outlines their general attributes and expectations:

1. **Casual Travelers:** Individuals or groups exploring a new place who seek quick suggestions without extensive planning.
2. **Solo Adventurers:** Users who prefer independent travel and rely on real-time guidance for exploring nearby attractions.
3. **Mobile-First Users:** Most Explorers are likely accessing Travzi on the go via mobile devices, requiring a responsive and intuitive user interface.
4. **Tech-Savvy Users:** Users familiar with basic technology and navigation who expect fast load times and seamless interactions.
5. **Spontaneous Planners:** Users who make unplanned or last-minute travel decisions and need instant suggestions based on their current location.
6. **Privacy-Aware Individuals:** Expect secure login and minimal data retention, appreciating Travzi's stateless architecture and Clerk authentication.

4.4 Hardware and Software Requirements

Hardware:

- A system with at least 8 GB RAM and a modern multi-core processor.
- Stable internet connection for API integration and testing.
- GPS-enabled mobile device for real-time location testing.

Software:

Frontend:

- Next.js (React-based framework)
- shadcn UI with Tailwind CSS for design components
- TypeScript for static typing

Backend & Database:

- API routes in Next.js or optional FastAPI microservice
- MySQL for data storage (if extended for persistence)

Development Tools:

- Visual Studio Code (IDE)
- Git (version control)
- Postman (API testing)
- Prisma ORM (for MySQL integration)
- Vercel for frontend deployment
- Render or Railway for backend hosting (if applicable)
- PlanetScale or Supabase for scalable MySQL hosting

Authentication & Security:

- Clerk (used in Travzi) or alternatives like NextAuth.js / JWT
- bcrypt for password hashing (optional for custom auth)
- Zod for schema validation and secure data handling

CHAPTER 5

SYSTEM DESIGN

5.1 Proposed System

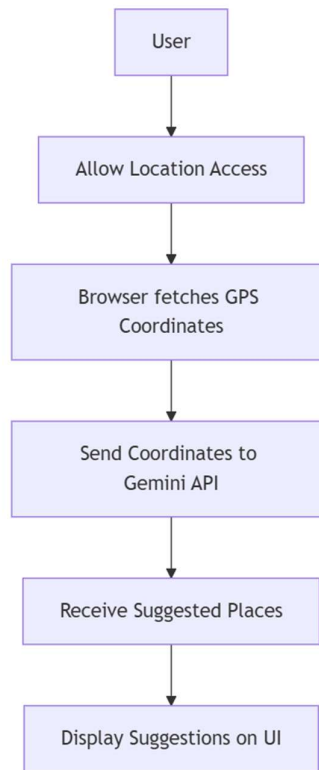


Fig 5.1 Proposed system

5.2. Block diagram

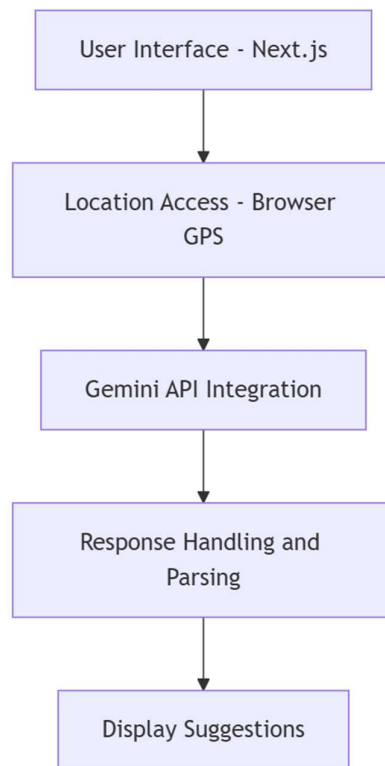


Fig 5.2 Block diagram

5.3 Component Diagram

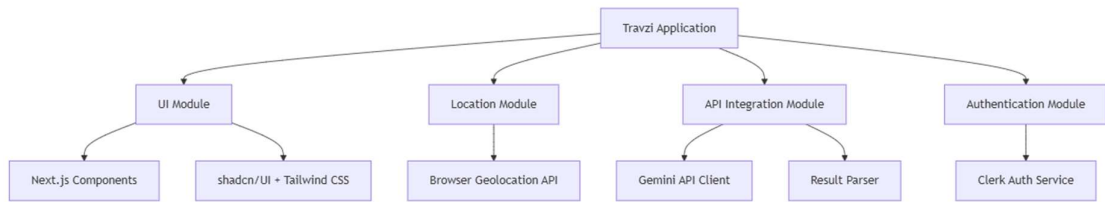


Fig 5.3 Component Diagram

5.4 Use case diagram

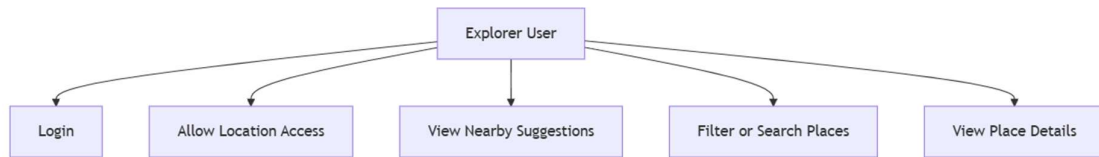


Fig 5.4 Use case diagram

5.5 Data flow diagram

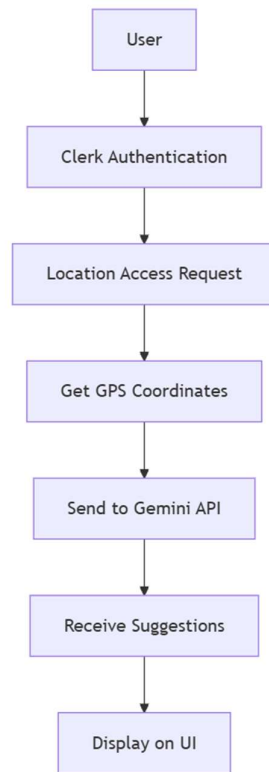


Fig 5.5 Data flow diagram

5.6 Class diagram

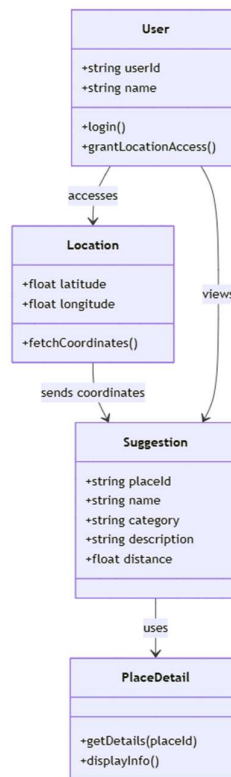


Fig 5.6 Class diagram

5.7 Sequence diagram

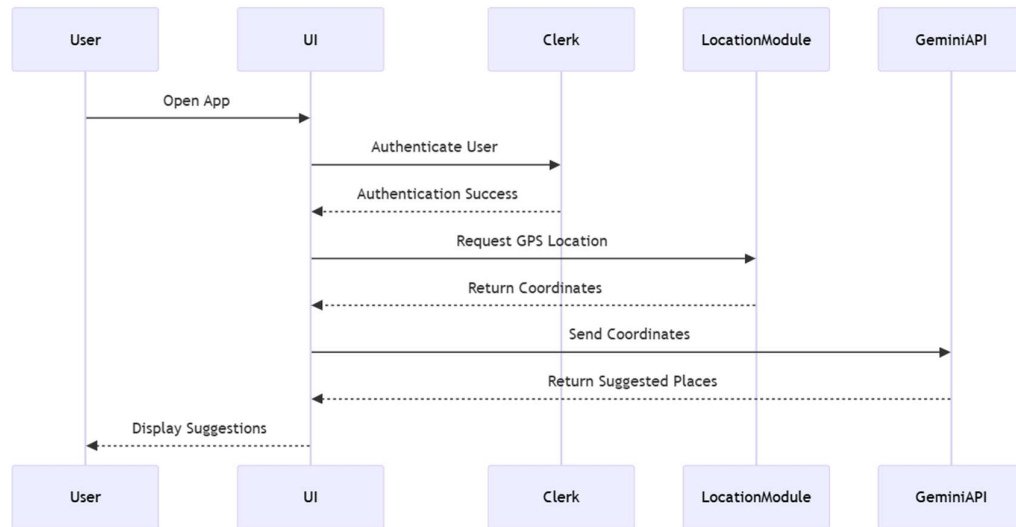


Fig 5.7 Sequence diagram

5.8 Database Design

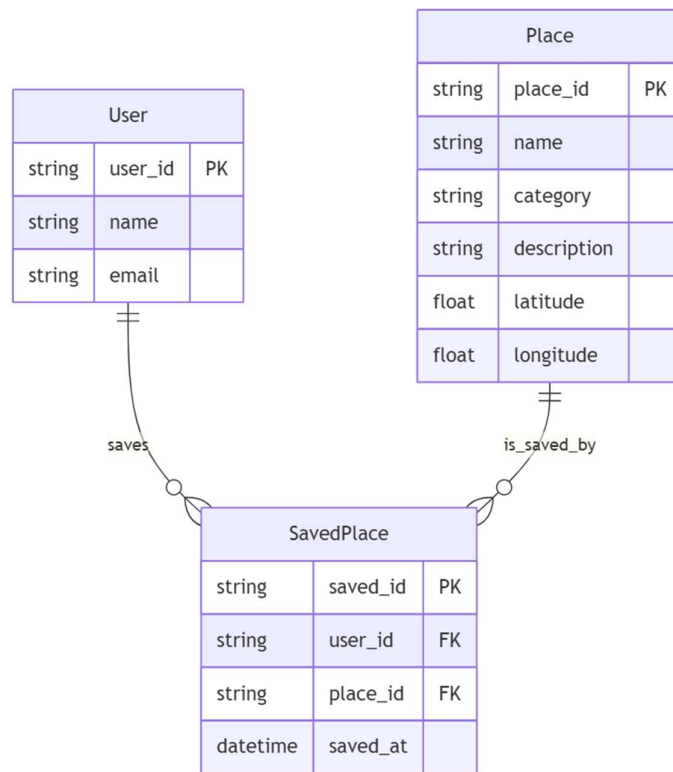


Fig 5.8 Database design diagram

CHAPTER 6

IMPLEMENTATION DETAILS

6.1 Project Modules

Travzi is made up of the following core modules, each of which provides essential functionality within the application:

1. Authentication Module (Clerk):

- Administers user sign-on and sign-up.
- Orchestrates session management and identity management.
- Gives access to location services only if the user is authenticated.

2. Suggestion Engine (Gemini API):

- Receives the user's location.
- Sends the user's location to the Gemini API for intelligent place recommendations.
- Parses the data returned from the API for usefulness and relevance to present suggestions.

3. UI/UX Module (Next.js + Tailwind):

- Developed with Next.js, allowing for an incredible user experience with server-side rendered pages and fast navigation.
- Each page is built with Tailwind CSS and shadcn/UI to create responsive, accessible, and modern interfaces.
- Provides a friendly user experience for easy path traversing with limited clicks.

4. Location Module (Browser Geolocation API):

- Provides the user the option to share their GPS coordinates.
- Obtains a location from the user to send to the suggestion engine.
- After the user grants permission, the application can continually adapt to the user's movement in real-time and update the places nearby.

6.2 General Installation Steps

Steps for installing modules

1. Open a terminal or command prompt.
2. Set Up the Development Environment Install Node.js and npm:
Download and install Node.js from nodejs.org. `node -v` `npm`
- Install the following Packages for setup the environment: -
 1. `pip install express.js`
 2. `pip install hardhat`
 3. `pip install Ethereum-waffle`
 4. `pip install tailwindcss`
 5. `pip install. dotenv`
 6. `pip install next.js`
 7. `pip install react-loader-spinner`
 8. `pip install ipfs`
 9. `pip install ipfs-http-client`
 10. `pip install react-toastify`
 11. `pip install styled-components`
- Follow the following steps to start the project:
 1. Setup the process of next.js on terminal to particular folder project
 2. `npm init`
 3. `npm run dev / yarn run`

CHAPTER 7

TESTING AND VALIDATION

1.1 Testing

1.1.1 Testing case for page 1

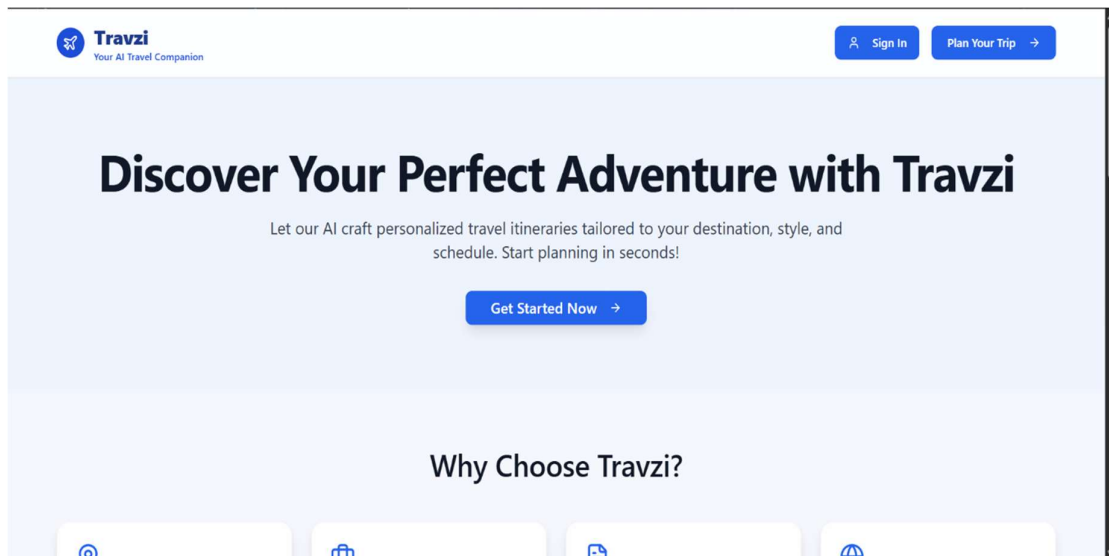


Fig 7.1 Test case 1

1.1.2 Testing case for page 2

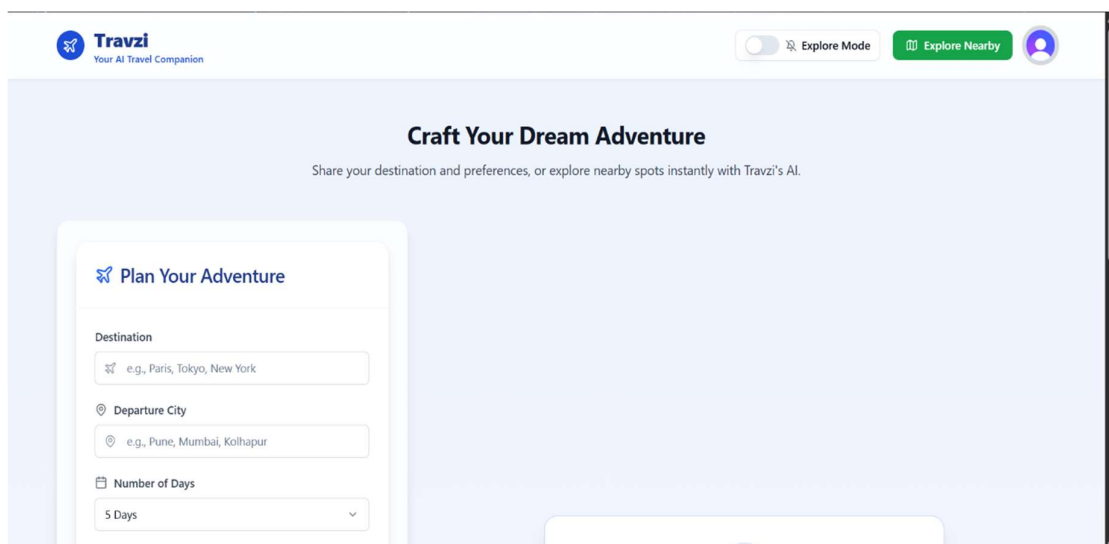


Fig 7.2 Test case 2

1.1.3 Testing case for page 3

The screenshot displays the Travzi AI Travel Companion interface. On the left, the 'Plan Your Adventure' section contains input fields for Destination (Berlin), Departure City (Pune), Number of Days (7 Days), Mode of Transport (Flight), and Additional Information (Relaxing beach, museums, hilly region). On the right, the 'Your Travel Itinerary' section shows a generated 7-day Berlin itinerary starting from Pune, India. It includes a paragraph describing the itinerary, a section for 'Important Notes' with bullet points about flights, visas, and accommodation, and buttons for 'Copy Itinerary' and 'Create Quick Summary'.

Plan Your Adventure

Destination
Berlin

Departure City
Pune

Number of Days
7 Days

Mode of Transport
Flight

Additional Information
Relaxing beach, museums, hilly region

Share preferences or must-visit spots for a tailored plan.

Your Travel Itinerary
Created 6/14/2025

Ready to Explore

Okay, here's a possible 7-day Berlin itinerary, starting from Pune, India, with a mix of popular attractions, local experiences, and considering your interest in beaches (though Berlin is inland!), museums, and hilly regions (with some creative interpretations!). This assumes a relatively comfortable travel pace.

Important Notes:

- **Flights:** This itinerary assumes you'll find flights that work with the general timing. Book flights well in advance for better prices. Airlines like Qatar Airways, Emirates, Turkish Airlines, and Lufthansa often have good routes from Pune to Berlin.
- **Visa:** Ensure you have the necessary Schengen visa for Germany.
- **Accommodation:** I haven't included specific hotel recommendations, but book accommodation near the city center (Mitte, Prenzlauer Berg, or Kreuzberg) for easy access to attractions. Consider hotels, Airbnbs, or hostels based on your budget and preferences.

Copy Itinerary

Create Quick Summary

Fig 7.3 Test case 3

1.2 Validation:

Validation was conducted to confirm Travzi's practical effectiveness and user satisfaction:

1. **User Testing:** A group of real users interacted with the application to simulate real-world usage scenarios, including location fetching and viewing suggestions.
2. **Feedback Collection:** Participants provided qualitative feedback on usability, accuracy of recommendations, and the overall interface experience.
3. **Accuracy Verification:** The suggested locations returned by the Gemini API were cross-checked for geographical relevance and contextual accuracy based on different test locations.
4. **Performance Review:** Load times, API response latency, and user experience were monitored to ensure the app met real-time usability expectations.
5. **Error Handling Assessment:** Edge cases such as denial of location permission, slow connections, and unavailable API responses were tested to confirm that fallback messages and app stability were maintained.

CHAPTER 8

RESULT, ANALYSIS AND CONCLUSION

8.1 Result

The implementation of Travzi yielded the following outcomes:

1. **Functional Integration of AI and Location Services:** Successfully combined Gemini API and browser GPS to offer real-time, personalized travel suggestions based on user location.
2. **Consistent Cross-Platform Performance:** The app operated smoothly on both mobile and desktop devices, demonstrating its responsiveness and compatibility across platforms.
3. **Secure and Seamless Authentication:** Clerk provided reliable user authentication and session handling without performance bottlenecks.
4. **Intuitive and Engaging User Interface:** The minimalistic, fast-loading UI allowed users to quickly interact with the system and receive suggestions without delay.
5. **Accurate and Relevant Recommendations:** Gemini API returned context-aware places that matched user location and travel interest categories.
6. **Robust System Stability:** No critical errors were encountered during testing, and all fallback conditions were handled gracefully.

8.2 Snapshots of work done

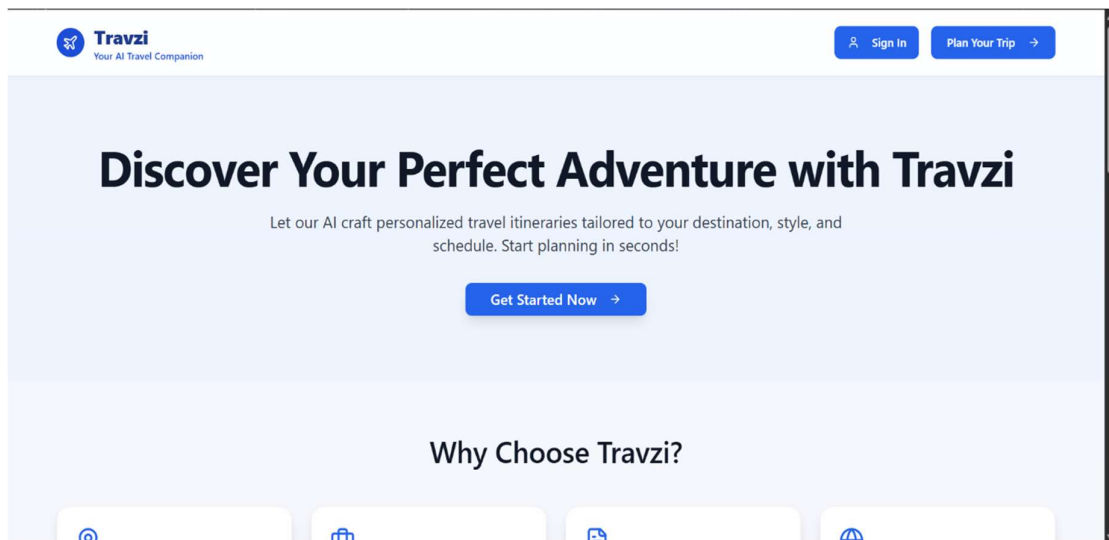


Fig 8.1 App interface page 1

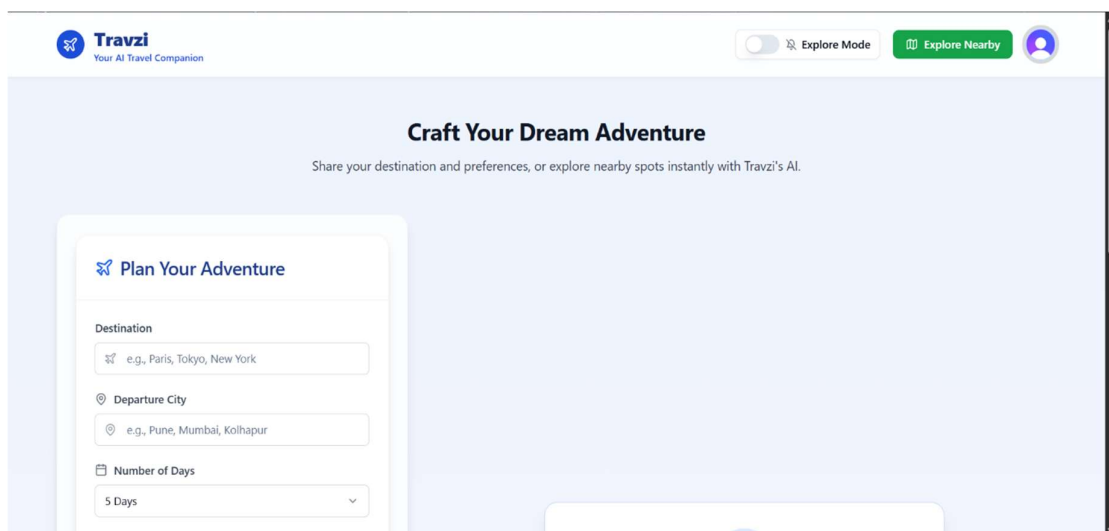


Fig 8.2 App interface page 2

Plan Your Adventure

Destination

Berlin

Departure City

Pune

Number of Days

7 Days

Mode of Transport

Flight

Additional Information

Relaxing beach, museums, hilly region

Share preferences or must-visit spots for a tailored plan.

Your Travel Itinerary

Created 6/14/2025

Ready to Explore

Okay, here's a possible 7-day Berlin itinerary, starting from Pune, India, with a mix of popular attractions, local experiences, and considering your interest in beaches (though Berlin is inland!), museums, and hilly regions (with some creative interpretations!). This assumes a relatively comfortable travel pace.

Important Notes:

- Flights:** This itinerary assumes you'll find flights that work with the general timing. Book flights well in advance for better prices. Airlines like Qatar Airways, Emirates, Turkish Airlines, and Lufthansa often have good routes from Pune to Berlin.
- Visa:** Ensure you have the necessary Schengen visa for Germany.
- Accommodation:** I haven't included specific hotel recommendations, but book accommodation near the city center (Mitte, Prenzlauer Berg, or Kreuzberg) for easy access to attractions. Consider hotels, Airbnbs, or hostels based on your budget and preferences.

Copy Itinerary

Create Quick Summary

Fig 8.3 App interface page 3

40

8.3 Analysis

1. **Gemini API Efficiency:** The Gemini API consistently provided relevant, contextual suggestions based on live location data, showcasing its ability to interpret spatial and environmental context effectively.
2. **Authentication Reliability:** Clerk's authentication service performed reliably across different devices and sessions, supporting secure and quick logins.
3. **UI and UX Responsiveness:** The application interface remained highly responsive, with minimal loading times and no visible performance lag, even during extended usage.
4. **Browser and Platform Compatibility:** The application was successfully tested across modern browsers and platforms, ensuring accessibility for a wide user base.
5. **Minimal System Resource Use:** Travzi's stateless architecture and API-driven model allowed for lightweight performance, even on devices with limited resources.

8.4 Conclusion

Travzi achieved its core objective of providing a seamless, AI-assisted, and location-aware travel recommendation platform. Through a simple interface, secure login, and powerful real-time suggestions, it enables users to explore nearby destinations effortlessly. The successful integration of Next.js, Clerk, and Gemini API demonstrated how modern technologies can be effectively used to create smart, responsive, and scalable web applications. With smooth performance, accurate outputs, and positive user feedback, Travzi is a promising tool for future travel assistance innovations.

8.5 Future scope

1. **Multi-Role Expansion:** Introduce additional user roles such as Admin and Travel Guide to enhance platform control and content contribution.
2. **Bookmarking System:** Allow users to save their favorite or visited places for future reference.
3. **Offline Access:** Enable caching of recent suggestions to support offline or low-network usage.
4. **Integration with Navigation Services:** Integrate with map platforms like Google Maps to offer real-time directions to suggested places.
5. **Feedback and Rating System:** Let users rate and review places to improve recommendation quality and community engagement.
6. **Multilingual and Accessibility Support:** Add localization and accessibility options to reach a wider and more inclusive audience.

CHAPTER 9

REFERENCES

9.1 Journals referred

- [1] Gretzel, U., Sigala, M., Xiang, Z., & Koo, C. (2015). Smart tourism: Foundations and developments. *Electronic Markets*, 25(3), 179–188. <https://doi.org/10.1007/s12525-015-0196-8>
- [2] Huang, C. D., Goo, J., Nam, K., & Yoo, C. W. (2017). Smart tourism technologies in travel planning: The role of exploration and exploitation. *Information & Management*, 54(6), 757–770. <https://doi.org/10.1016/j.im.2016.11.010>
- [3] Xiang, Z., Magnini, V. P., & Fesenmaier, D. R. (2015). Information technology and consumer behavior in travel and tourism: Insights from travel planning using the internet. *Journal of Retailing and Consumer Services*, 22, 244–249. <https://doi.org/10.1016/j.jretconser.2014.08.005>
- [4] Law, R., Buhalis, D., & Cobanoglu, C. (2014). Progress on information and communication technologies in hospitality and tourism. *International Journal of Contemporary Hospitality Management*, 26(5), 727–750. <https://doi.org/10.1108/IJCHM-08-2013-0357>
- [5] H Li, X., & Wang, Y. (2011). Measuring the effectiveness of online marketing platforms in tourism. *Tourism Management*, 32(4), 960–965. <https://doi.org/10.1016/j.tourman.2010.08.012>
- [6] Gretzel, U. (2011). Intelligent systems in tourism: A social science perspective. *Annals of Tourism Research*, 38(3), 757–779. <https://doi.org/10.1016/j.annals.2010.12.007>

9.2 References

- [1] Next.js Documentation- <https://nextjs.org/docs>
- [2] shadcn/ui Documentation - <https://ui.shadcn.com/docs>
- [3] MySQL Official Documentation - <https://dev.mysql.com/doc/>
- [4] Prisma ORM for MySQL - <https://www.prisma.io/docs>
- [5] Vercel – Next.js Hosting Platform - <https://vercel.com/docs>
- [6] Career Development and Mentorship – National Career Development Association (NCDA) - <https://ncda.org/aws/NCDA/pt/sp/resources>
- [7] Designing Effective Learning Roadmaps – Harvard Business Review - <https://hbr.org/2021/04/the-science-of-effective-learning-pathways>
- [8] Coursera Career Academy Overview - <https://www.coursera.org/career-academy>

Appendices

Appendix A:

Appendix B:

Plagiarism Report