Project Report On

Compassly (AI Trip Planner)

Submitted in partial fulfillment for the award of

Post Graduate Diploma in Advanced Computing

from

C-DAC ACTS (Pune)

Guided by

Ms. Tejaswini Apte

Presented By

Ankit Sayane – 240340120036

Nihal Bambale – 240340120115

Lokesh Shirode – 240340120184

Shriram Kumbhar – 240340120191

Vaibhav Bhoyar – 240340120215

Centre of Development of Advanced Computing (C-DAC), Pune

CERTIFICATE

TO WHOMSOEVER IT MAY CONCERN

This is to certify that

Ankit Sayane – 240340120036
Nihal Bambale – 240340120115
Lokesh Shirode – 240340120184
Shriram Kumbhar – 240340120191
Vaibhav Bhoyar – 240340120215

have successfully completed their project titled

“Compassly (AI Trip Planner)”

Under the Guidance of Ms. Tejaswini Apte

Project Guide **HOD ACTS**

ACKNOWLEDGEMENT

This project **"Compassly (AI Trip Planner)"** was a great learning experience for us and we are submitting this work to Advanced Computing Training School (CDAC ACTS).

We all are very glad to mention the name of **Ms. Tejaswini Apte** for her valuable guidance to work on this project. Her guidance and support helped us to overcome various obstacles and intricacies during the course of project work.

Our most heartfelt thank goes to Ms. **Swati mam** (Course Coordinator, PG- DAC) who gave all the required support and kind coordination to provide all the necessities like required hardware, internet facility and extra Lab hours to complete the project and throughout the course up to the last day here in C-DAC ACTS, Pune.

Ankit Sayane –240340120036
Nihal Bambale – 240340120115
Lokesh Shirode – 240340120184
Shriram Kumbhar – 240340120191
Vaibhav Bhoyar – 240340120215

TABLE OF CONTENTS

1. Introduction

Compassly (AI Trip Planner) is an innovative application designed to transform the way users plan their travels. This platform offers a streamlined approach by gathering essential information such as destination, travel duration, budget, and group size to craft a customized trip itinerary. By leveraging the **Gemini API (Generative AI & prompt engineering)**, Compassly delivers personalized recommendations, including tailored hotel options and detailed daily activity plans, ensuring a unique and enjoyable travel experience.

The application's frontend is developed using **React JS**, providing a dynamic and interactive user interface. **ViteJs** is utilized for rapid builds, **React Router** ensures smooth navigation, and **Shadcn** is employed for efficient component styling. **Tailwind CSS** enhances the design with its responsive and visually appealing features, contributing to an engaging user experience across various devices.

On the backend, Compassly is powered by **Spring Boot**, which handles business logic and API interactions with efficiency and reliability. **MongoDB** is selected for its flexible and scalable data management capabilities, allowing the application to handle diverse data needs seamlessly. The integration of **Google APIs** further enhances the functionality of Compassly, providing additional features and improving the overall travel planning experience.

Furthermore, Compassly is designed with a user-centric approach, featuring a front-end interface that is both immersive and intuitive. The platform’s design ensures that users can easily navigate and interact with the application, offering a smooth journey from start to finish. Real-time updates and personalized features make travel planning more convenient and enjoyable, reflecting Compassly’s commitment to modern and user-friendly solutions.

The application's responsiveness extends beyond mere adaptability, employing a fluid design that scales effortlessly across different devices. Touch-friendly buttons, intuitive navigation, and efficient loading times contribute to a seamless user experience. By leveraging **React.js** for dynamic content rendering, Compassly ensures a high level of user satisfaction and fosters a strong connection with its audience, delivering a contemporary and enjoyable travel planning experience.

2. Software/Hardware Requirement

Server:

Processor: Intel Core i5 or equivalent AMD processor.

RAM: Minimum 8GB RAM.

Storage: SSD storage for improved performance.

Network: Ethernet or Wi-Fi connectivity.

Operating System: Linux distribution (Ubuntu, CentOS) preferred for server deployment.

Client Devices:

Processor: Dual-core processor or higher.

RAM: Minimum 4GB RAM.

Storage: Sufficient storage for caching and local data.

Network: Ethernet or Wi-Fi connectivity.

Browser: Compatible with latest versions of popular browsers like Google Chrome, Mozilla Firefox, and Safari.

3. Tools and technologies used

- React JS
- ViteJs
- React Router
- Shadcn
- Tailwind CSS
- Spring Boot
- MongoDB
- Gemini API (Generative AI & prompt engineering)
- Google Places API
- Google Image API
- HTML and CSS
- Axios
- Git

1. **Spring Boot:** Utilized to develop the backend of the application, providing a robust framework for building Java-based web applications with ease.

2. **React JS:** Employed to build the dynamic and interactive user interface of the application, facilitating efficient rendering of components and ensuring a smooth user experience.

3. **ViteJs:** Used for rapid development and faster builds, enhancing the development workflow with quick hot module replacement and optimized build processes.

4. **React Router:** Manages navigation and routing within the React application, enabling smooth transitions between different views and components.

5. **Shadcn:** Utilized for streamlined component styling, offering a set of reusable and customizable components that enhance design consistency across the application.

6. **Tailwind CSS:** Employed to create a responsive and visually appealing design, providing a utility-first approach to styling that allows for rapid design changes and customization.

7. **Spring Boot:** Used to develop the backend of the application, providing a robust framework for building Java-based web applications with ease and integrating various services.

8. **MongoDB:** Chosen for its flexibility and scalability, serving as the database solution to manage and store application data efficiently and support complex data structures.

9. **Gemini API (Generative AI & prompt engineering):** Integrated to provide personalized trip itineraries and recommendations through advanced AI-driven prompts and generative techniques.

10. **Google Places API:** Retrieves and displays location-specific information, such as restaurants, landmarks, and accommodations, thereby enhancing the trip planning process.

11. **Google Image API:** Utilized to fetch and display relevant images, enriching the user experience by providing visual context for destinations and attractions.

12. **HTML and CSS:** Employed for structuring and styling the web pages, ensuring that the application's content is well-organized and visually appealing.

13. **Axios:** Implemented for making HTTP requests to interact with backend services and APIs, simplifying data fetching and management within the application.

14. **Git:** Used for version control and collaborative development, tracking changes to the codebase and facilitating team collaboration through branching and merging.

4. Project Flow Diagram

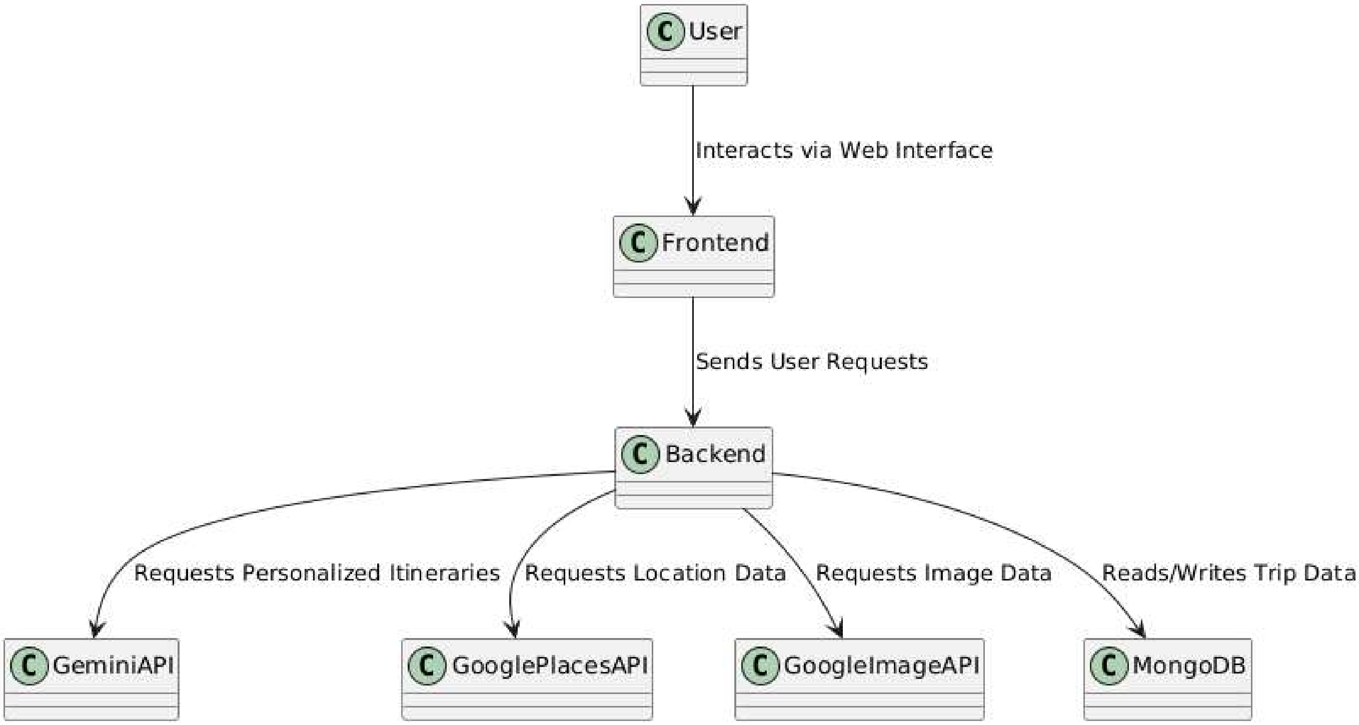


Fig-1: Project Flow Diagram

5. Advantages

Use of React JS for Frontend Development

1. Dynamic User Interface: React JS provides a robust framework for creating a dynamic and interactive user interface. Its component-based architecture allows for the development of reusable UI components, enhancing the efficiency of the development process.

2. Improved Performance: React's Virtual DOM optimizes performance by minimizing direct manipulation of the real DOM. This leads to faster updates and rendering, resulting in a more responsive user experience.

3. State Management: React JS efficiently manages the state of the application with tools like Context API and Redux, making it easier to handle complex data flows and ensure consistent state management across the application.

4. Rich Ecosystem: The extensive ecosystem of React includes numerous libraries and tools that integrate seamlessly, such as React Router for navigation and Axios for API requests, enhancing the functionality and flexibility of the project.

5. Strong Community Support: With a large and active community, React JS benefits from continuous updates, extensive documentation, and a wealth of resources for troubleshooting and best practices.

Use of ViteJs for Fast Builds

1. Rapid Development: ViteJs provides a development server with fast hot module replacement (HMR), allowing developers to see changes instantly without needing to refresh the entire application.

2. Optimized Build Performance: ViteJs leverages modern build tools like ESBuild to offer highly optimized build performance, reducing the time required for compiling and bundling code.

3. Out-of-the-Box Support: ViteJs offers built-in support for features such as TypeScript, JSX, and CSS modules, simplifying the development setup and reducing the need for additional configuration.

4. Improved Efficiency: By serving source files directly in development and utilizing pre-bundling for dependencies, ViteJs streamlines the development workflow and improves overall efficiency.

Use of Spring Boot for Backend Services

1. Simplified Development: Spring Boot simplifies backend development with its convention-over-configuration approach, allowing developers to focus on writing business logic without extensive setup or configuration.

2. Integrated Development Environment: It provides a comprehensive set of tools and libraries for building robust backend services, including support for data access, security, and RESTful APIs.

3. Rapid Prototyping: Spring Boot's auto-configuration and embedded server capabilities enable rapid prototyping and deployment, accelerating the development process.

4. Scalability: The framework is designed to handle high traffic and large-scale applications, making it suitable for enterprise-level solutions.

Use of MongoDB for Data Storage

1. Flexible Schema Design: MongoDB's schema-less design allows for flexible and dynamic data modeling, accommodating evolving data requirements without the constraints of a fixed schema.

2. Horizontal Scalability: MongoDB supports horizontal scaling through sharding, enabling the distribution of data across multiple servers to handle increased loads and ensure high availability.

3. High Performance: Its in-memory storage engine and indexing capabilities provide high performance for read and write operations, optimizing data access and retrieval times.

4. Rich Query Capabilities: MongoDB offers a powerful query language with support for aggregation, indexing, and text search, enhancing data retrieval and manipulation capabilities.

Use of Gemini API for AI and Prompt Engineering

1. Personalized Recommendations: The Gemini API leverages AI to generate personalized trip itineraries and recommendations based on user inputs, enhancing the user experience with tailored suggestions.

2. Advanced Data Processing: It utilizes prompt engineering to process and analyze user data efficiently, providing more accurate and relevant results for travel planning.

3. Seamless Integration: The API integrates smoothly with the backend services, enabling the application to fetch and display customized content dynamically.

4. Enhanced User Engagement: By offering personalized and contextually relevant information, the Gemini API helps increase user engagement and satisfaction with the travel planning application.

Use of Google APIs for Location and Image Services

1. Accurate Location Data: Google Places API provides reliable and up-to-date location information, enhancing the application's ability to suggest relevant travel destinations and points of interest.

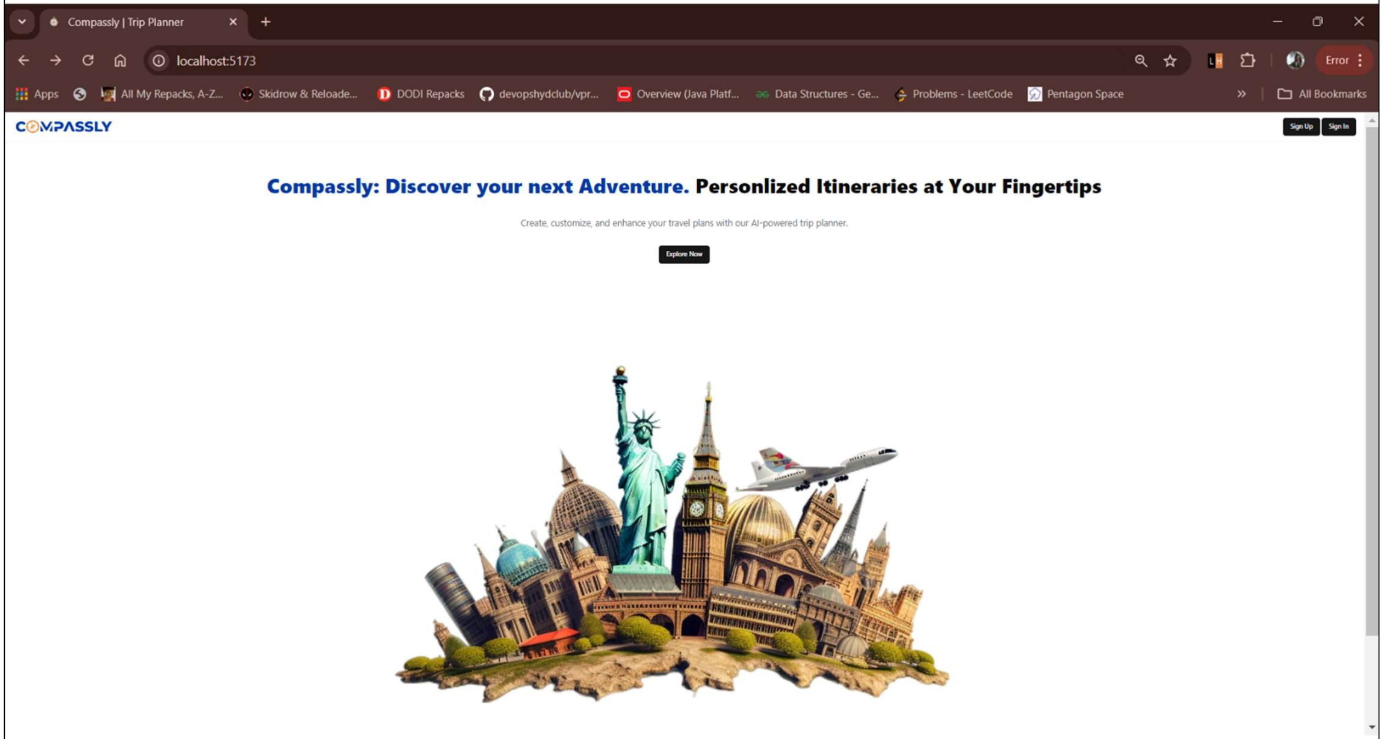
2. High-Quality Images: Google Image API allows for the integration of high-quality images, improving the visual appeal of the application and providing users with rich, engaging content.

3. Easy Integration: Google APIs offer comprehensive documentation and straightforward integration, simplifying the process of incorporating location and image services into the application.

4. Consistent User Experience: Leveraging Google's well-established services ensures a consistent and reliable user experience, with access to a broad range of location and imagery data.

6. Screenshots

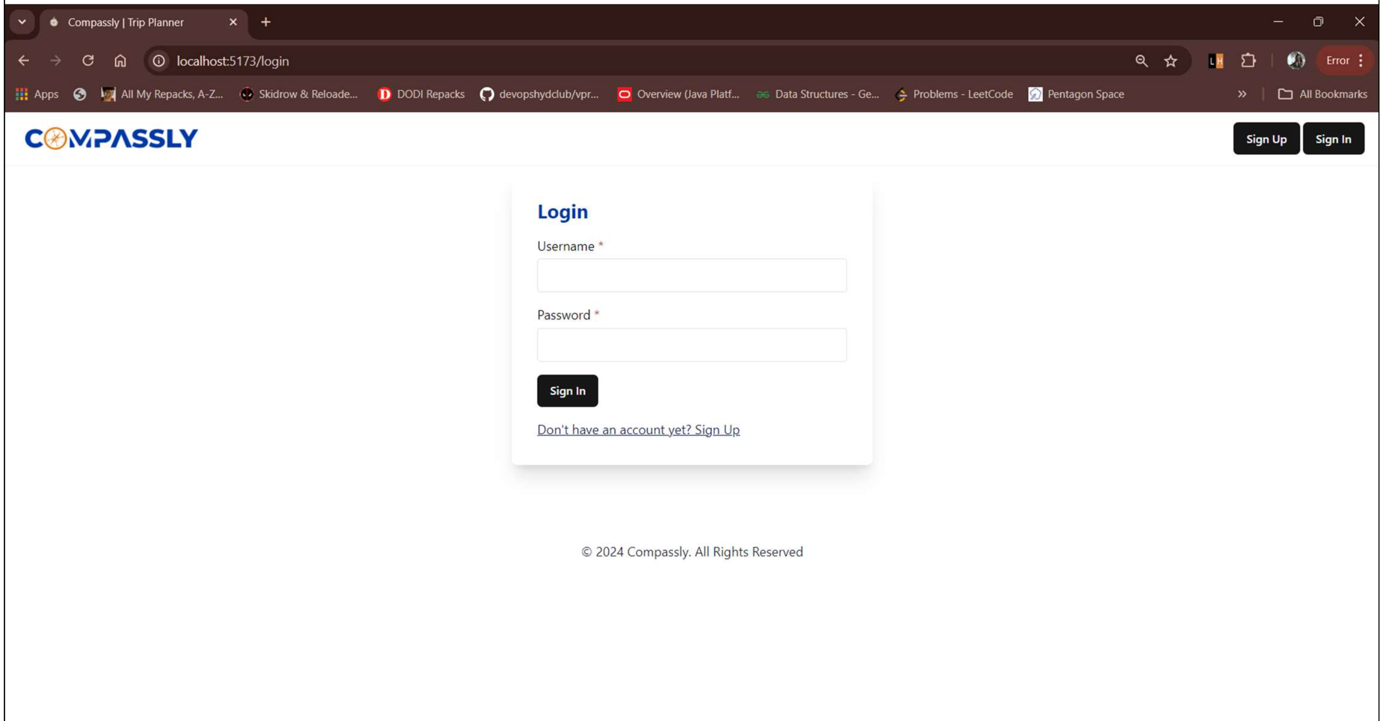
A) Functionalities



Compassly: Discover your next Adventure. Personlized Itineraries at Your Fingertips

Create, customize, and enhance your travel plans with our AI-powered trip planner.

Fig- 2: Home Screen



Login

Username *

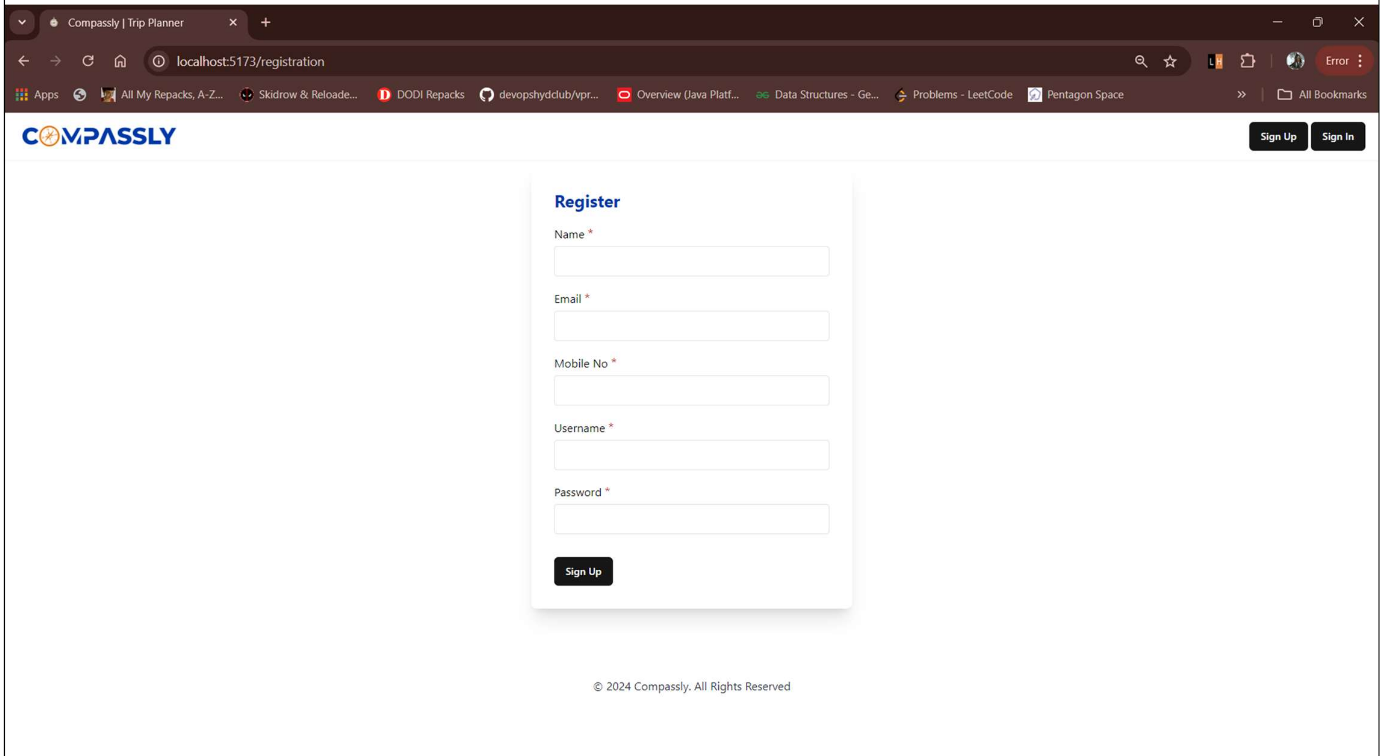
Password *

Sign In

Don't have an account yet? Sign Up

© 2024 Compassly. All Rights Reserved

Fig- 3: Login Page



Register

Name *

Email *

Mobile No *

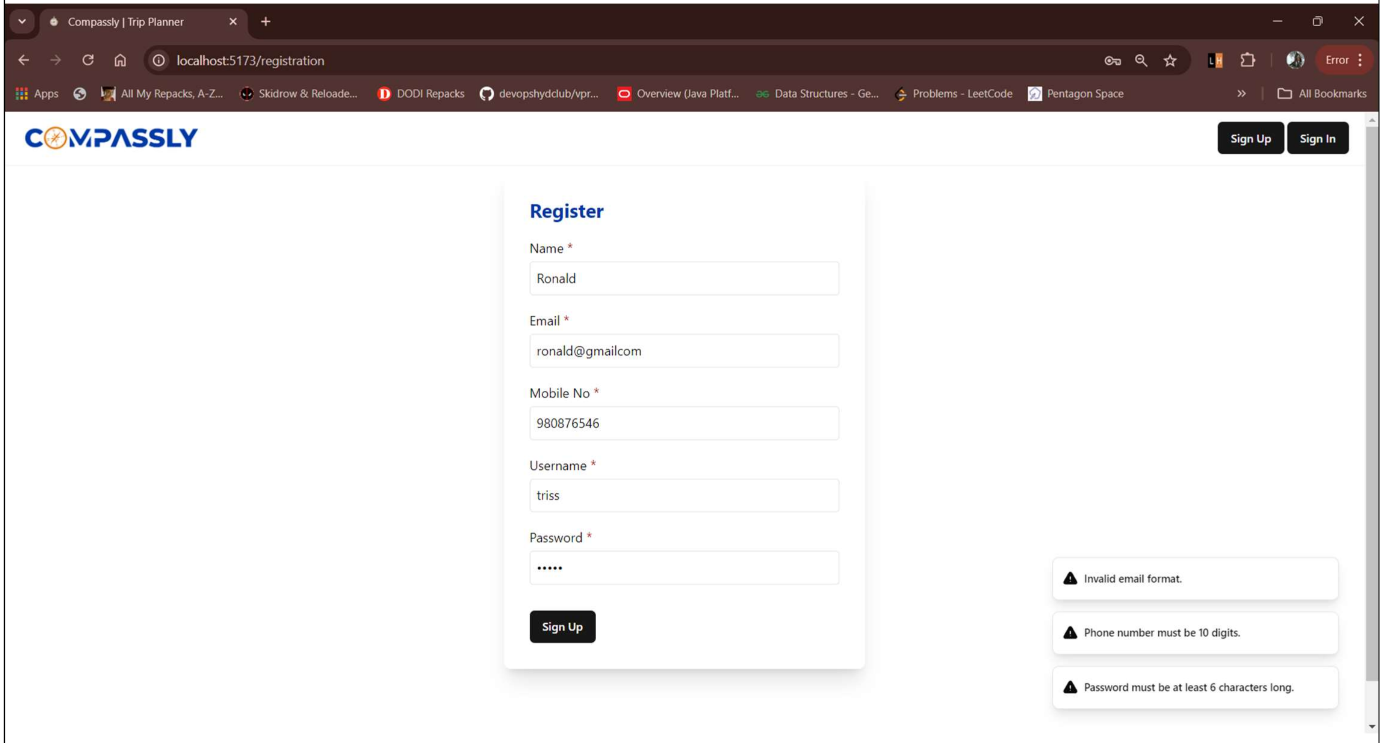
Username *

Password *

Sign Up

© 2024 Compassly. All Rights Reserved

Fig- 4: Sign Up Page



Register

Name *
Ronald

Email *
ronald@gmailcom

Mobile No *
980876546

Username *
triss

Password *
•••••

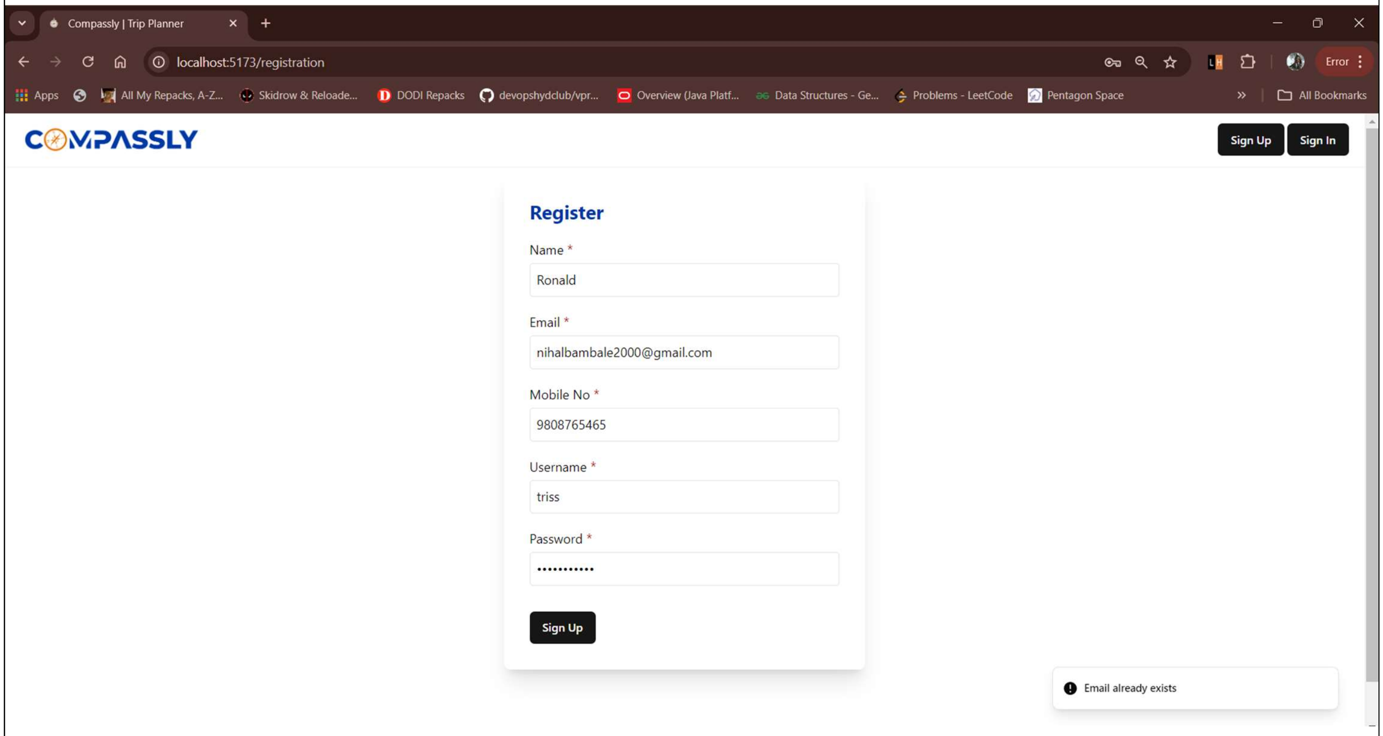
Sign Up

Invalid email format.

Phone number must be 10 digits.

Password must be at least 6 characters long.

Fig- 5: Sign Up Validations



Register

Name *
Ronald

Email *
nihalbambale2000@gmail.com

Mobile No *
9808765465

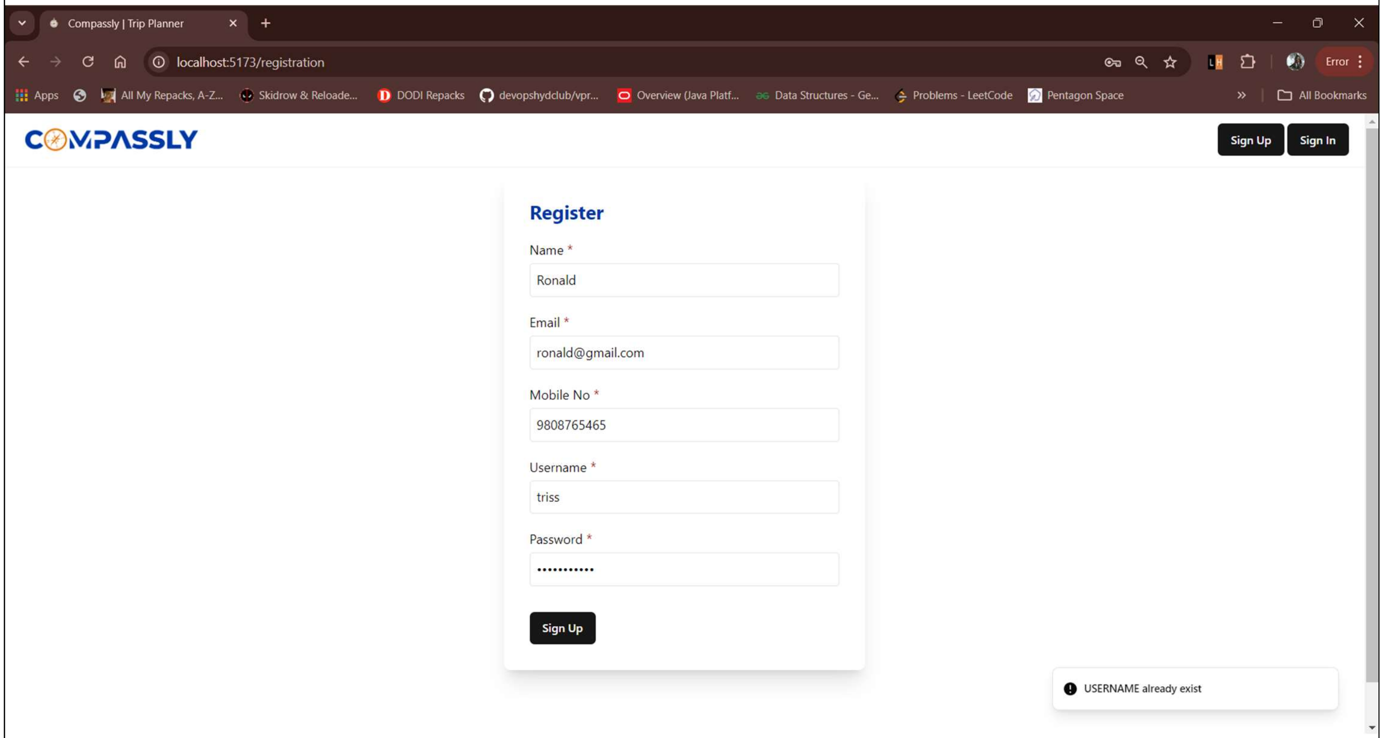
Username *
triss

Password *
•••••••••••

Sign Up

Email already exists

Fig- 6: Server-side Validation



Register

Name *
Ronald

Email *
ronald@gmail.com

Mobile No *
9808765465

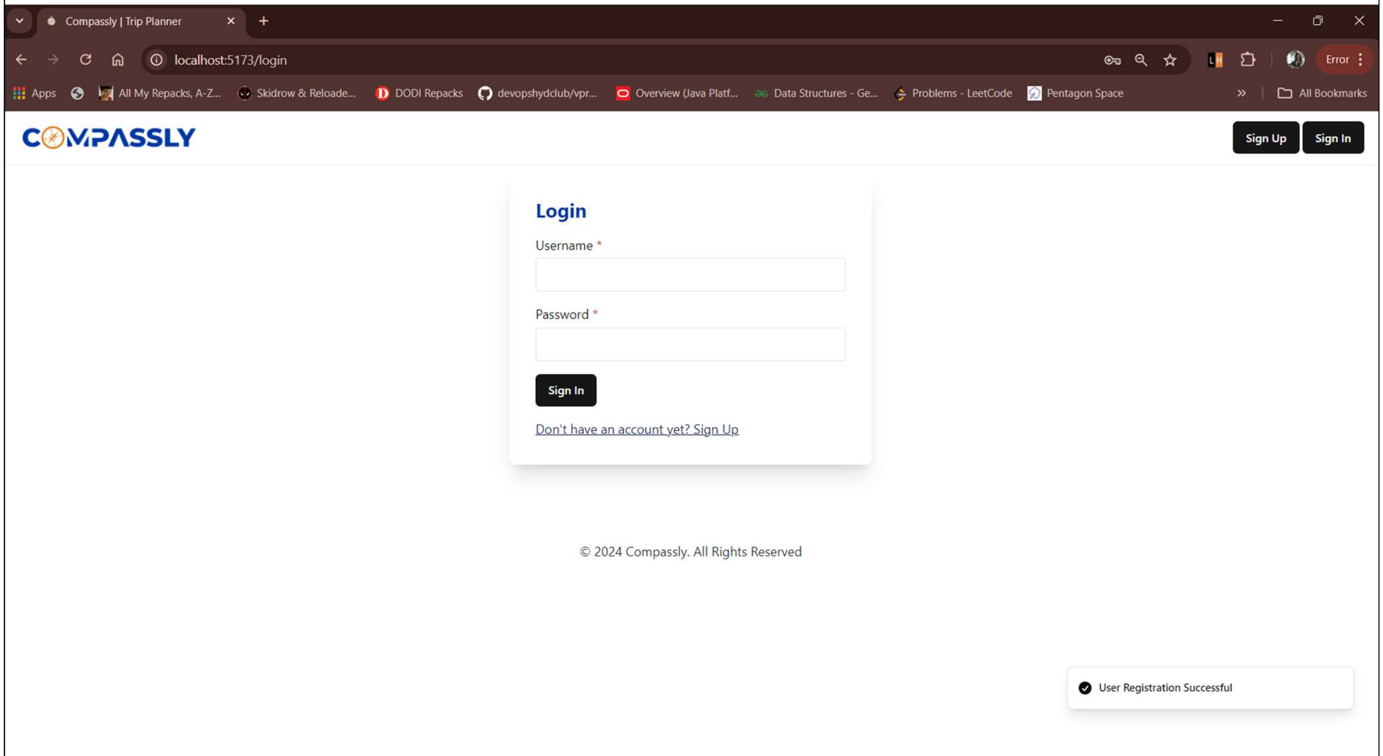
Username *
triss

Password *
•••••••••••

Sign Up

USERNAME already exist

Fig- 7: Second Server-Side Validation



Login

Username *

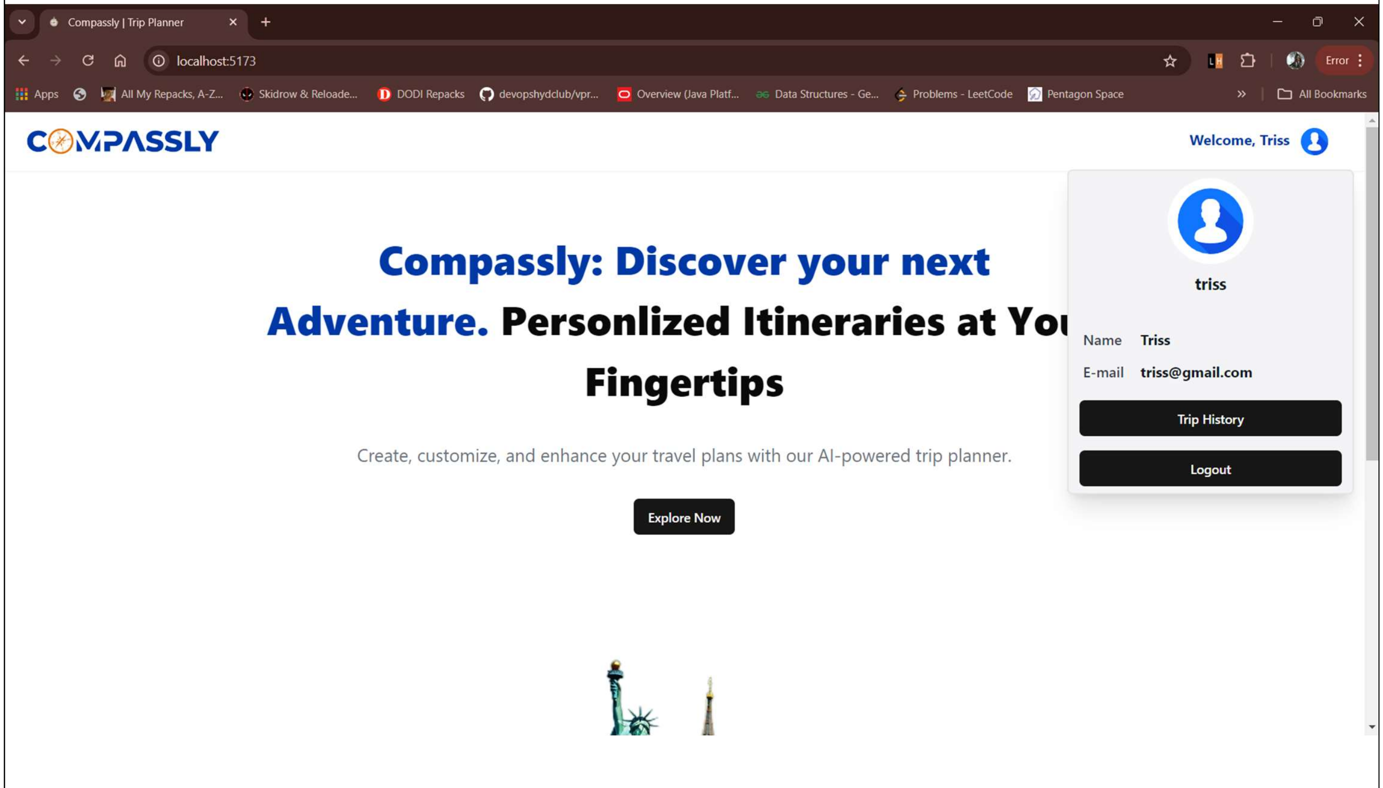
Password *

Don't have an account yet? Sign Up

© 2024 Compassly. All Rights Reserved

User Registration Successful

Fig- 8: User Successfully Registered



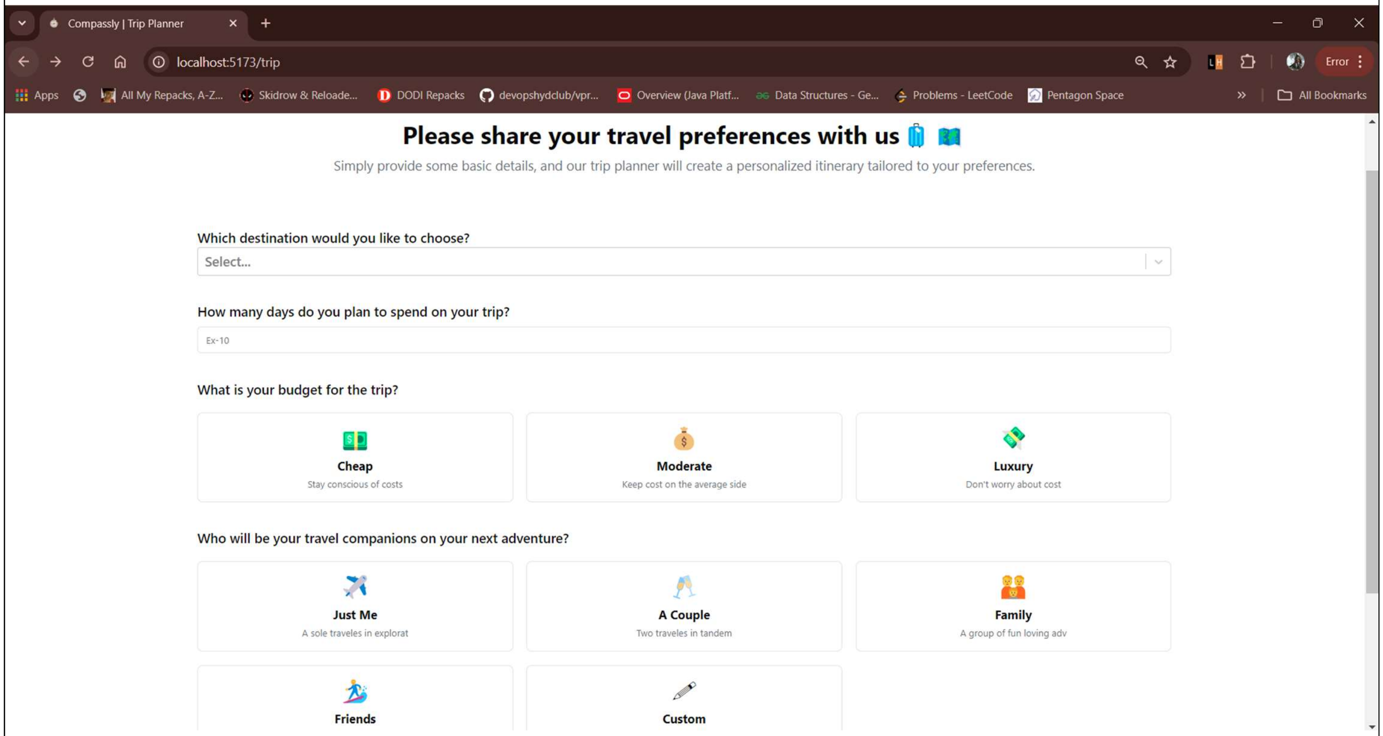
Compassly: Discover your next Adventure. Personlized Itineraries at You Fingertips

Create, customize, and enhance your travel plans with our AI-powered trip planner.

Name Triss

E-mail triss@gmail.com

Fig- 9: User Flyout



Please share your travel preferences with us 🧳 🗺️

Simply provide some basic details, and our trip planner will create a personalized itinerary tailored to your preferences.

Which destination would you like to choose?

Select...

How many days do you plan to spend on your trip?

Ex-10

What is your budget for the trip?

Cheap Stay conscious of costs

Moderate Keep cost on the average side

Luxury Don't worry about cost

Who will be your travel companions on your next adventure?

Just Me A sole traveles in explorat

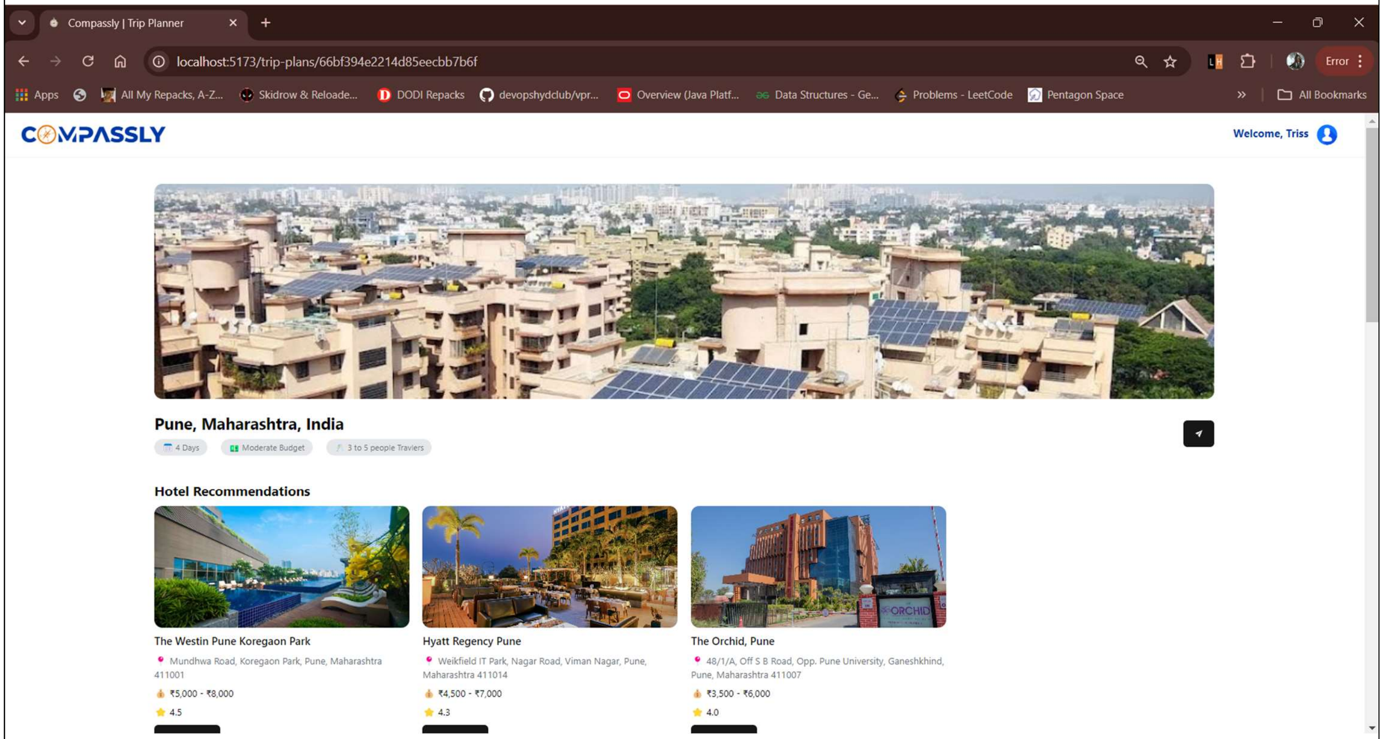
A Couple Two traveles in tandem

Family A group of fun loving adv

Friends

Custom

Fig- 10: Trip Form



Pune, Maharashtra, India

4 Days | Moderate Budget | 3 to 5 people Travlers

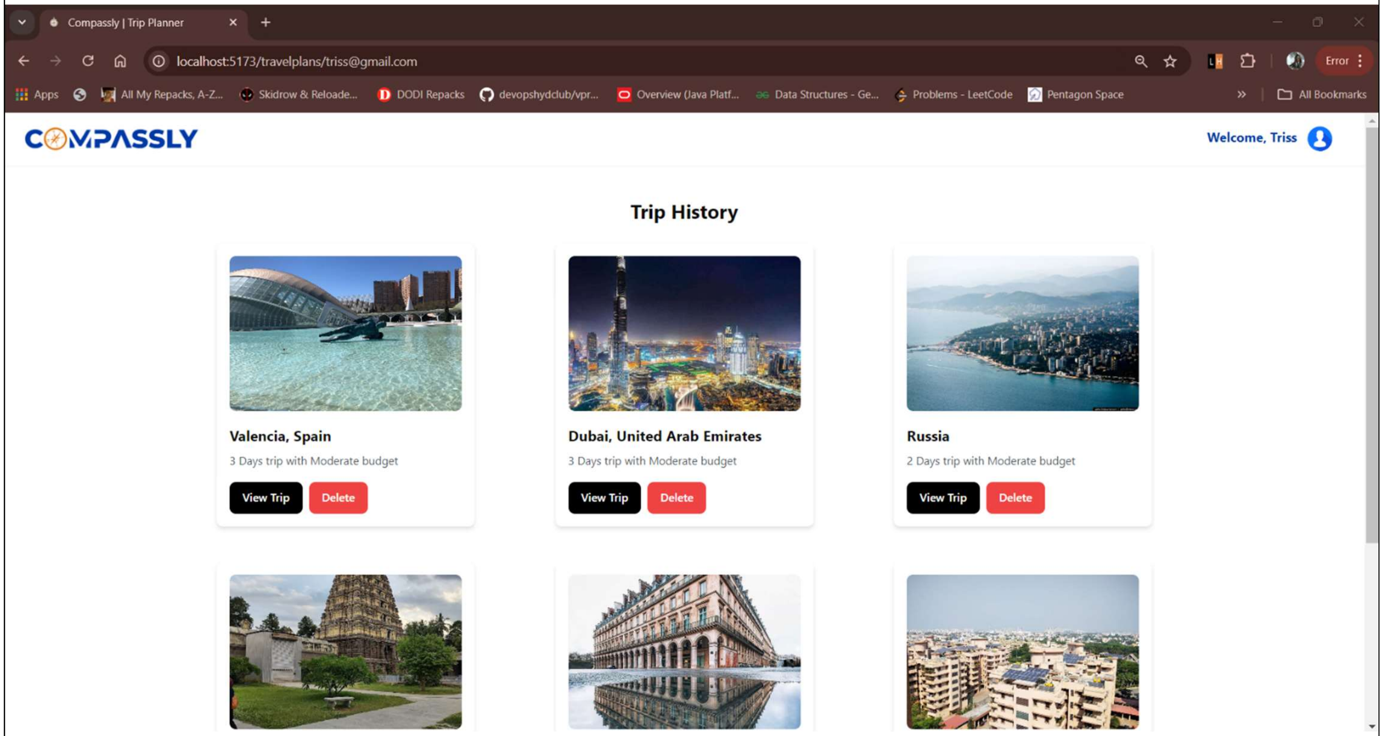
Hotel Recommendations

The Westin Pune Koregaon Park
Mundhwa Road, Koregaon Park, Pune, Maharashtra 411001
₹5,000 - ₹8,000
⭐ 4.5

Hyatt Regency Pune
Weikfield IT Park, Nagar Road, Viman Nagar, Pune, Maharashtra 411014
₹4,500 - ₹7,000
⭐ 4.3

The Orchid, Pune
48/1/A, Off S B Road, Opp. Pune University, Ganeshkhind, Pune, Maharashtra 411007
₹3,500 - ₹6,000
⭐ 4.0

Fig- 11: Trip Result



Trip History

Valencia, Spain

3 Days trip with Moderate budget

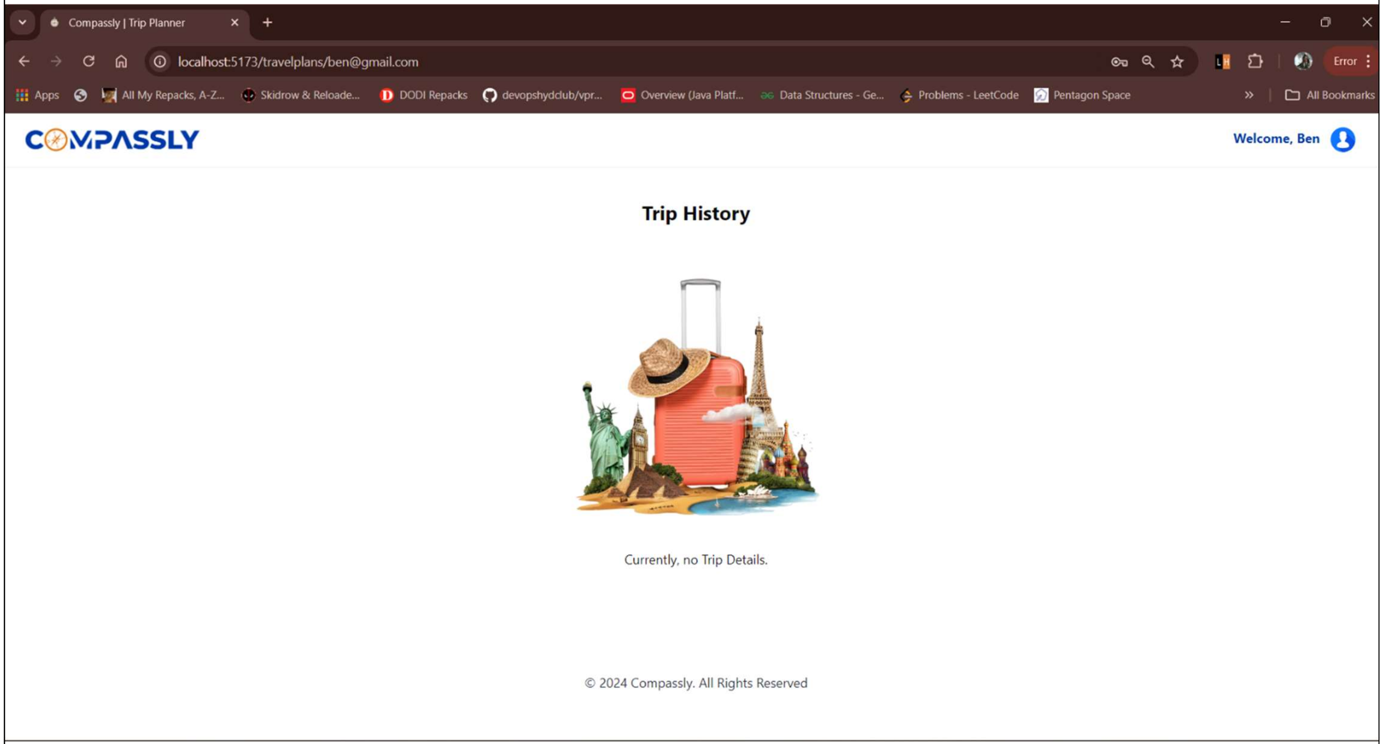
Dubai, United Arab Emirates

3 Days trip with Moderate budget

Russia

2 Days trip with Moderate budget

Fig- 12: Trip History



Trip History

Currently, no Trip Details.

© 2024 Compassly. All Rights Reserved

Fig- 13: Empty Trip History

B) Use case Diagram

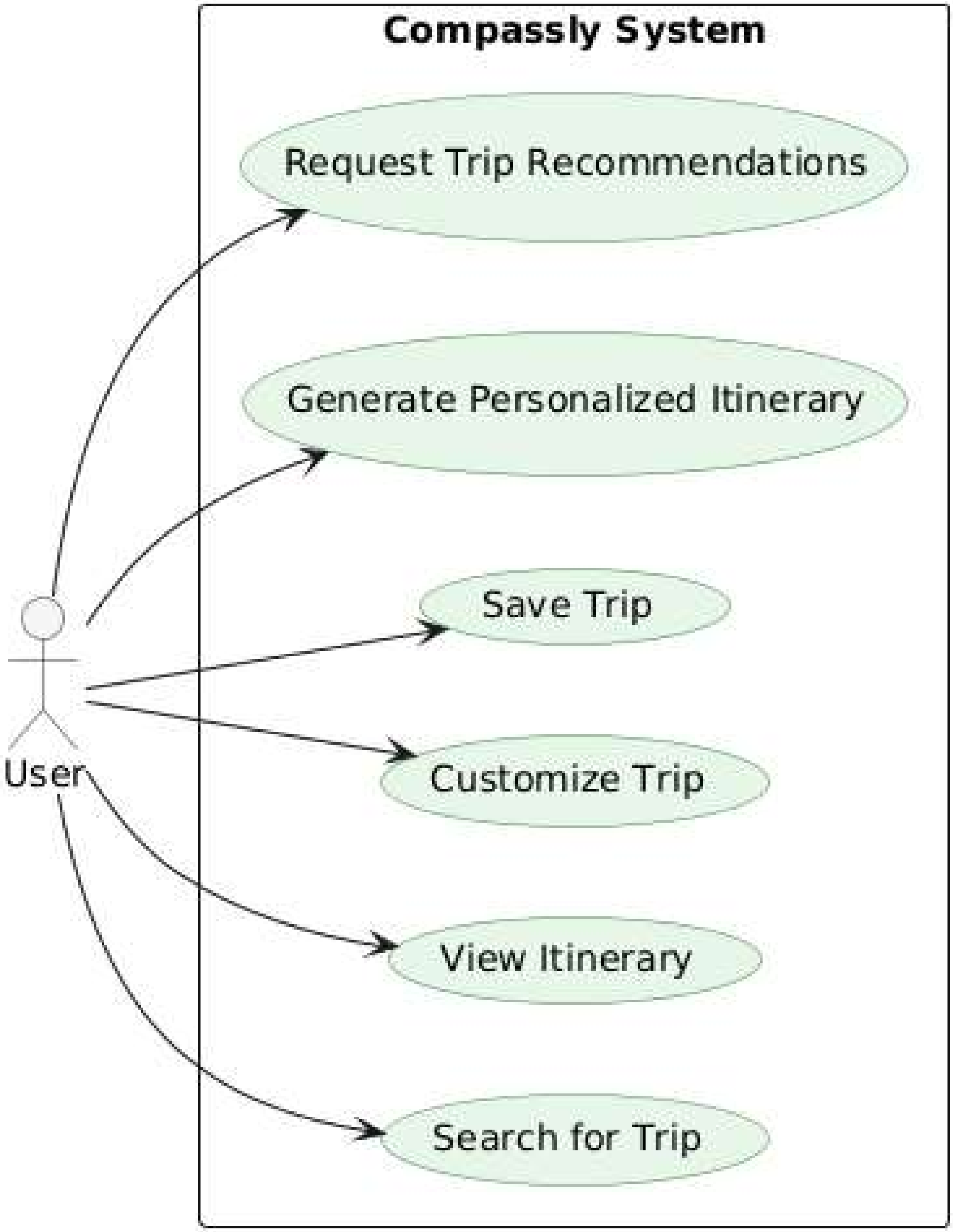


Fig- 14: Use Case Diagram

7. Future Scope for Compassly (AI Trip Planner)

1. Enhanced Personalization Features: Future updates could include advanced personalization algorithms to offer even more tailored trip itineraries based on user preferences, past behavior, and real-time data. Incorporating machine learning models to predict user preferences and suggest destinations dynamically could significantly enhance the user experience.

2. Integration with Additional APIs: Expanding the application's capabilities by integrating with more APIs could provide users with additional services such as real-time weather updates, local event recommendations, and transportation options. This would create a more comprehensive travel planning experience.

3. Multi-Language Support: To reach a broader audience, future versions of the application could include multi-language support. Implementing internationalization and localization features would allow users from different regions to interact with the platform in their preferred languages.

4. Mobile Application Development: Developing a mobile app version of Compassly could provide users with on-the-go access to their trip itineraries and recommendations. A mobile app could leverage device-specific features such as GPS for real-time location tracking and notifications for timely updates.

5. Enhanced AI and ML Capabilities: Integrating more sophisticated AI and machine learning algorithms could improve the accuracy and relevance of recommendations. Features like predictive analytics for travel trends and sentiment analysis based on user feedback could offer deeper insights and enhance the application's value.

6. Social Sharing Features: Adding social sharing capabilities would allow users to share their trip plans, itineraries, and experiences on social media platforms. This feature could facilitate user engagement and attract new users through word-of-mouth and social proof.

7. User-Generated Content: Incorporating user-generated content such as reviews, photos, and travel tips could enrich the application's database and provide more authentic and diverse travel recommendations. This could be achieved through community features and review systems.

8. Integration with Augmented Reality (AR): Implementing AR features could enhance the travel planning experience by allowing users to visualize destinations, landmarks, and accommodations in a more immersive way. This could be particularly useful for exploring new locations and planning activities.

9. Advanced Data Analytics: Future iterations could include advanced data analytics tools to provide users with insights into travel trends, cost estimates, and other valuable metrics. This would help users make informed decisions and optimize their travel plans.

10. Partnership Opportunities: Collaborating with travel agencies, airlines, and accommodation providers could open up opportunities for exclusive deals and offers. Partnering with industry players could also provide users with additional services and enhance the overall travel planning experience.

8. Limitations of Compassly (AI Trip Planner)

1. Limited API Integrations: Currently, the project integrates with a specific set of APIs (Gemini API, Google Places API, Google Image API). Future enhancements could benefit from additional integrations to provide more comprehensive travel-related information and services.

2. Dependence on Internet Connectivity: The application relies on real-time data and API calls, which necessitates a stable internet connection. Users with intermittent or poor connectivity may experience limited functionality or slower performance.

3. Data Privacy Concerns: Handling user data for personalized recommendations and trip planning raises privacy concerns. Ensuring robust data protection and compliance with data privacy regulations is essential but can be challenging.

4. Scalability Challenges: While the application is built with scalable technologies, the growing number of users and data may pose scalability challenges. Continuous performance monitoring and optimization may be required to maintain responsiveness as the user base expands.

5. Limited Offline Functionality: The current implementation may offer limited offline capabilities. Users might not have access to their travel plans or real-time updates when offline, which could affect the overall user experience.

6. Customization Constraints: The project provides a generalized travel planning experience. Highly specific or niche travel preferences may not be fully addressed by the current features and customization options.

7. API Rate Limits and Quotas: External APIs used in the project, such as the Google Places API, often have usage limits and quotas. Exceeding these limits may affect the availability and responsiveness of certain features.

8. User Interface Complexity: Although designed for a user-friendly experience, the integration of multiple features and data sources could lead to a complex interface. Ensuring intuitive navigation and ease of use is crucial but may require ongoing refinements.

9. Real-Time Data Accuracy: The accuracy of real-time data provided by external APIs (e.g., hotel availability, activity recommendations) may vary. Inaccurate or outdated information could impact user trust and satisfaction.

10. Performance Overheads: The combination of various technologies and APIs may introduce performance overheads. Optimizing the application to handle multiple concurrent users and ensure fast response times is essential but challenging.

9. Conclusion

In conclusion, the Compassly (AI Trip Planner) project exemplifies the successful integration of advanced technologies to offer a sophisticated and intuitive travel planning solution. By utilizing React JS, ViteJs, React Router, and Shadcn for a dynamic user interface, coupled with Spring Boot and MongoDB for a reliable backend, Compassly ensures a seamless and engaging experience for users. The incorporation of Gemini API for personalized itineraries and Google APIs for enhanced functionality underscores the project's commitment to providing a comprehensive travel planning tool. With a future-oriented vision that includes potential enhancements such as advanced personalization, mobile app development, and AR integration, Compassly is well-positioned to evolve and adapt to emerging trends in the travel industry. This project not only delivers a cutting-edge solution but also sets the stage for continuous innovation and growth in the travel planning domain

10. References

1. **React JS**: https://reactjs.org/
2. **ViteJs**: https://vitejs.dev/
3. **React Router**: https://reactrouter.com/
4. **Shadcn**: https://shadcn.dev/
5. **Tailwind CSS**: https://tailwindcss.com/
6. **Spring Boot**: https://spring.io/projects/spring-boot
7. **MongoDB**: https://www.mongodb.com/
8. **Gemini API (Generative AI & Prompt Engineering)**: https://gemini.google.com/app?hl=en-IN
9. **Google Places API**: https://developers.google.com/maps/documentation/places
10. **Google Image API**: https://developers.google.com/custom-search/v1/overview*)*
11. **HTML and CSS**: https://developer.mozilla.org/en-US/docs/Web/HTML and https://developer.mozilla.org/en-US/docs/Web/CSS
12. **Axios**: https://axios-http.com/
13. **Git:** https://git-scm.com/