Travel Advisor Website

By Austin Philip, Han Chen, Jinyan Jiang, Rain Zhang

GitHub Repository Link:
https://github.com/f4ncy1zach/group10-finalproject

Video Demo Link:
https://drive.google.com/file/d/1XvrvX8FSCE4AEr4jEBwm7ABnHE8nOTMh/view?usp=sharing

Deployed Website Link:
https://travel-advisor-project.vercel.app/

Success Analysis:

Our Travel Advisor project was mostly successful in meeting user needs based on feedback from peers who tested it during the peer testing session. The users liked the appearance and interactive components of the application, and they considered the functionality useful and easy to use. However, there were some problems to fix. The chat did not update automatically with new messages from the AI, so users had to manually refresh it. Users also could not easily read AI responses that included bullet points because there was no formatting. Some testers commented that they could not tell when the AI was typing a response since there was no "typing" cue to show that the AI was thinking. This feedback tells us that even though our core travel planning feature worked seamlessly, we still need to work on the details of this application such as user experience for the AI chat feature. Adding simple status indicators and enhancing the formatting would significantly enhance the experience for users trying to plan trips, thus pushing this website to a higher level.

SDLC Model Analysis:

Agile Methodology

Our group used the Agile methodology for the Travel Advisor project and it was a productive approach to our development. As computing science students working on a team project that requires effective cooperation, Agile provided the flexibility and iterative approach that we needed.

The following practices that we utilized in our development consists of:

Sprint Planning: We divided our work into 2-week sprints with well-defined goals and deliverables.

Weekly Stand-ups: Brief one-hour meetings to discuss progress, blockers, and what we are going to accomplish in the upcoming week.

Sprint Reviews: We presented completed features at the end of each sprint and gathered feedback.

Sprint Retrospectives: We talked about what went well and how to improve it for the next sprint.

User Stories: The requirements were broken down into user stories, since we wanted to focus on user experience.

The following changes are made during our development:

Changed Sprint Duration: We initially planned 1-week sprints, however, we underestimated the complexity and amount of work required for each milestone.

Therefore, we switched to 2-week sprints, which provided a good balance between thorough testing and development time.
Frequency Change for Weekly Stand-ups: During the last two weeks of our development, we increased the frequency of our meetings from weekly to at least twice per week, due to the high amount of bug fixes and testing required to accomplish.

API Features Description:

OpenAI API Integration:

The OpenAI API (ChatGPT 4o Mini) is the core AI engine of our Travel Advisor AI functionalities.

1. Destination Recommendation:
   - Processes user passport and visa information to suggest destinations
   - Considers visa requirements, travel dates, and number of travelers
   - Returns country and city suggestions with error handling and smart-correction for invalid inputs and typos

2. Itinerary Generation:
   - Produces day-to-day travel plans by destination and trip duration
   - Includes places to visit and things to do per day
   - Implements retry logic for handling API failures
   - Transitions from planning: Included more descriptive activity titles and location-specific information

3. General Information Retrieval for Destinations:
   - Produces short descriptions of destinations in case of TripAdvisor data unavailability
   - Limits responses to 50-60 words for consistent UI display
   - Changes from planning: Originally planned to use only TripAdvisor data for descriptions only, but added this as a fall-back since there is a possibility of TripAdvisor API failing to return destination descriptions

4. Interactive Chat Assistant:
   - Answers travel-related questions in real-time
   - Provides personalized suggestions based on user questions
   - Changes from planning: Added more travel-specific context requirement to make chatbot responses more relevant

TripAdvisor API Integration

The TripAdvisor API provides data to enhance the website's recommendations. No change was made for this API after the planning stage.

1. Location Search and Details:
   - Searches for destinations by either user input (when user chooses not to use AI recommendation)
   - Retrieves city information, including descriptions and photos

2. Hotels, Restaurants, and Attractions Data:
   - Retrieves and displays accommodation with rating, price, and features
   - Provides restaurant recommendations with cuisine and reviews
   - Presents popular attractions with descriptions and features
   - Changes from planning: We removed the price comparison feature but we added $ icons to show how expensive the hotel is with more $ meaning more expensive.

3. Photo Retrieval:
   - Presents destination and location images throughout the category listings
   - Changes from planning: We could not get comments for TripAdvisor as we did not have the exclusive TripAdvisor Content API and there was no ethical way to scrape the comments either so we ended up switching to finding images

CI/CD Pipeline Description & Overview:

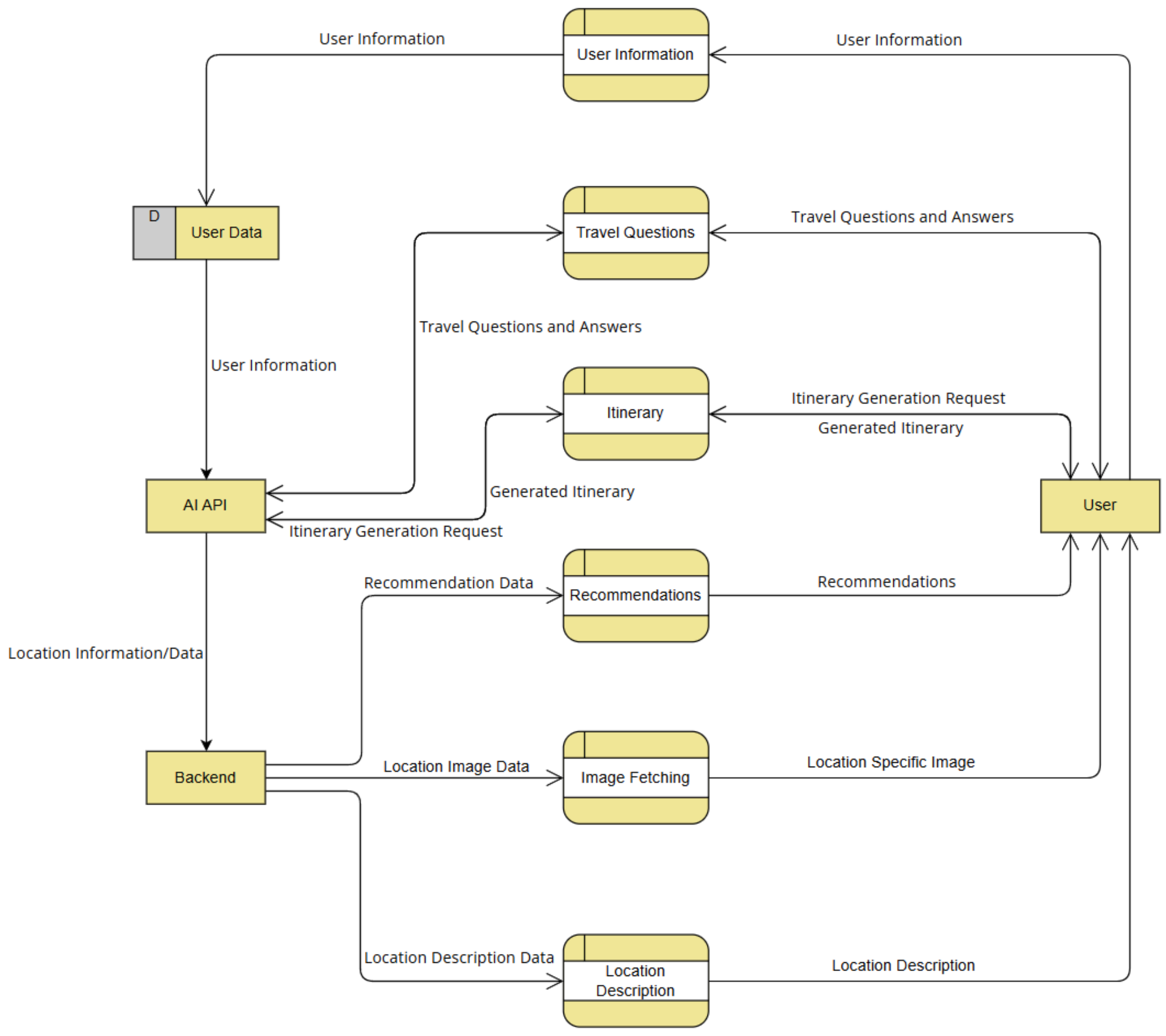
In our project, the CI/CD pipeline was implemented by using GitHub Actions to facilitate the development workflow as well as ensure all functionalities still work as expected with every update. On every push and pull request to the main branch, the pipeline will be triggered automatically to provide instructions to start testing. For such testing, we used Cypress to do so. The process of how everything works is using Github Actions to install dependencies, build the application, and then run Cypress to validate API responses, page navigation, user inputs, as well as overall functionality of all of the features. Once it is deployed, we check Actions on Github to make sure to monitor the results of the tests and record down any issues we find and if we can figure out anyways to solve them.

Testing Strategy Description:

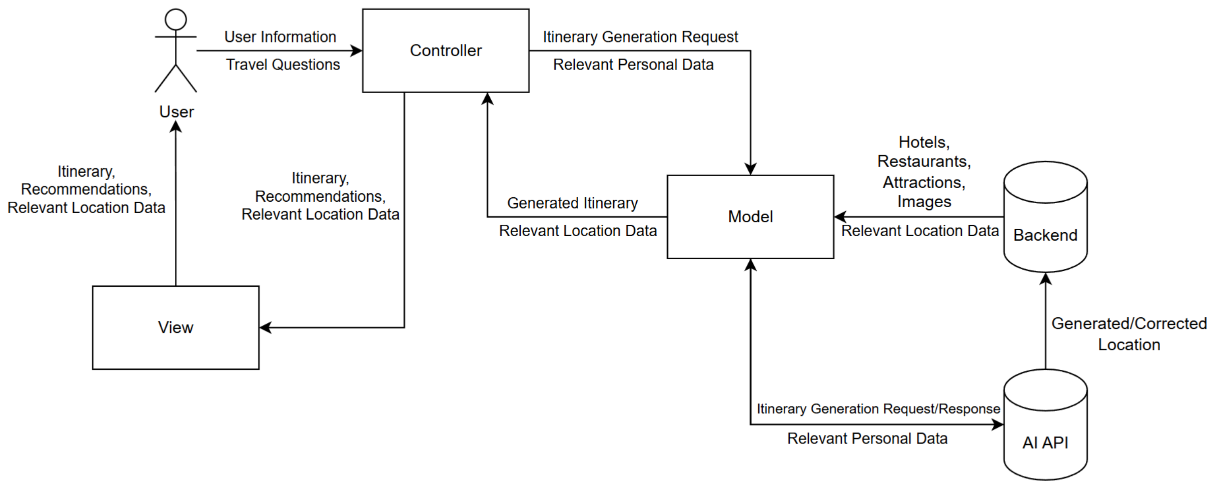
Our testing strategy focused on functionality and reliability across our application. We used Cypress' end-to-end testing to allow us to simulate real user interactions with our application as well as validate API calls, navigation, and form submissions. One test just for the AI chatbot feature, one for choosing your own destination and all of the features and one for where the AI chooses one for you based on your information. Running all three tests, we are able to see all of the functionalities of our application in action along with the API calls and to see if we are receiving all of the correct information.

Project Architecture Diagram:

Updated Level-1 DFD:



Updated MVC Model:



User
User Information
Travel Questions
Controller
Itinerary Generation Request
Relevant Personal Data
Itinerary, Recommendations, Relevant Location Data
Itinerary, Recommendations, Relevant Location Data
Generated Itinerary
Relevant Location Data
Model
Hotels, Restaurants, Attractions, Images
Relevant Location Data
Backend
View
Generated/Corrected Location
Itinerary Generation Request/Response
Relevant Personal Data
AI API

List of Known Bugs and Issues:

Bug 1	**Incorrect Listings With Other Destinations Being Displayed**
Description	When the website sends requests to the TripAdvisor API for the location listings, the website occasionally displays results from entirely different destinations. Although the name of the listing may include the name of the correct destination city or location, the actual physical location of the listing does not match. For example, if the destination is Vancouver, and a restaurant called “Vancouver Foods” might show up, with the physical location of this restaurant being in Toronto.
Cause	This issue is caused by the inconsistency of the TripAdvisor API. Our team configured the API request to search with respect to the physical location, some listings still might be shown according to the destination’s name.
Steps to Reproduce	1. Run the project using “npm run dev”. 2. Click the start button, then follow the question flow, until the result page shows. The method of choosing a destination or being recommended one by our AI is not important. 3. In the category tab of the result page, browse through the hotel, attractions, and restaurants listings until a location with incorrect location is found.

	4. Since the result of each destination is different, the user will have to go through step 1 to 3 again if no location was found with incorrect location shown since this bug does not always occur.
Severity	**Moderate** - This can mislead the user on choosing their desired locations during the trip. However, it won't lead to big consequences since the user will notice the error when searching for the address, and users can still choose other locations that are recommended.
Link to Issue	https://github.com/f4ncy1zach/group10-finalproject/issues/91

Bug 2	**Hotels, Attractions, and Restaurants Fail to Load**
Description	Sometimes, the listings of the hotels, attractions, and restaurants might fail to load, requiring the user to manually click the reload button to load the missing information.
Cause	This issue appears to be caused by inconsistencies in the TripAdvisor API. Although our team has managed to send the API request at a consistent point in the code, since it is sent immediately when the relevant page loads. The API sometimes still fails to return the data.
Steps to Reproduce	1. Run the project by using "npm run dev". 2. Click the start button, then follow the question flow until the results page. The method of choosing a destination or being recommended one by our AI is not important. 3. If any of the three categories (hotels, attractions, and restaurants) are displayed as "failed to retrieve information", then this issue is reproduced.

	4. Since this issue only has a small chance of happening, if it does not happen, the user will have to repeat steps 1 to 3 until this issue is found.
Severity	**Moderate** - Even though this bug can affect the user's experience, this bug does not happen often and the reload button is always available for use. However, when this problem does appear, it affects the user's experience and the overall reliability of our website.
Link to Issue	https://github.com/f4ncy1zach/group10-finalproject/issues/92

Bug 3	**Terminal Warning Message for API Requests**
Description	At certain times, the terminal displays warning messages related to too many requests being made to the backend folder "travel_advisor_guidance" when sending requests to the TripAdvisor API. These messages indicate that multiple API calls are being sent within a certain time period.
Cause	This issue appears to be caused by the inconsistencies in the TripAdvisor API. The frequency of requests within a short period could be overloading the API or triggering rate limits. As a result, this violates the final project requirements which specifies that a clean console with no warning messages are required.
Steps to Reproduce	1. cd to the directory of "travel_advisor_guidance". 2. Run the file using "npm run dev" so it runs on a local port. 3. Run the travel advisor project using "npm run dev". 4. Follow through the questionnaires until the result page, then refresh the category pages (hotels, attractions, and restaurants) multiple times, until a warning message that says "too many requests" appears in the terminal that runs "travel_advisor_guidance".

Severity	**Low** - According to our team's observation through testing this problem thoroughly, this bug does not affect the user's experience as the page does not fail to load when the message appears. However, it is still a violation to the CMPT 276 final project's requirement.
Link to Issue	https://github.com/f4ncy1zach/group10-finalproject/issues/93

Bug 4	**Missing Image From Certain Category Listings**
Description	Some hotels, attractions, and restaurant listings do not display images on the website. This occurs because the TripAdvisor API does not provide image data for certain locations. As a result, users viewing these listings are unable to see a visual reference, which can affect the overall user experience.
Cause	This issue is due to missing image fields in the data returned by the TripAdvisor API. For some listings, the API simply does not include an image, even though the request is structured correctly on the frontend.
Steps to Reproduce	1. Run the project using "npm run dev". 2. Follow the questionnaires until the result page, ideally choosing an unpopular destination city. 3. Scroll down to the category listings section, and observe the listing(s) without image, if any. If no listings are found, steps 1 and 2 will need to be repeated.
Severity	**Moderate** – This affects the user interface and experience by removing visual context for some listings, but it does not break functionality or cause application errors.
Link to Issue	https://github.com/f4ncy1zach/group10-finalproject/issues/94

Future Work and Potential Improvements:

User Authentication and Profiles

- Implementation of user accounts to retain travel plans and likes
- Profile support to have passport and visa information ready for future trips
- Social aspects to let users share itineraries with travel companions
- Technical Approach: Can be implemented with user account system, where each user has their own Travel Advisor account for login

Enhanced Destination Intelligence

- Weather forecasts integration to provide climate information for travel periods
- Exchange rates for currency to help in financial planning
- Local festivities and events during the travel planned in the attractions category
- Technical Approach: Can use OpenWeatherMap API for climate data, RestCountries APIs for currency, and Eventbrite API for festivities and events

Mobile Application Development

- Native apps for iOS and Android through React Native
- Offline availability of itineraries without internet connectivity
- Push alerts for reminder notifications and alerts while traveling
- Technical Approach: Replicate the current React web app to React Native with suitable mobile-specific modules

Map Features

- Interactive maps showing daily routes and transportation methods
- Time-based scheduling with estimated travel duration between sites
- Technical approach: Can use Google Maps API for mapping features

AI Enhancements

- Personalized recommendations based on user preference and trip history (Prerequisite: User Account System)
- Natural language processing for trip planning in more conversational manner
- Image recognition to recognize landmarks in user photographs
- Technical approach: Configure more advanced OpenAI models and specialized AI services

Lesson Learned & Project Takeaway:

The development of the Travel Advisor project was a valuable experience for all our team members. We acquired technical proficiency in React, modern JavaScript, integrating APIs, responsive UI/UX, continuous integration and deployment, along with automated testing . Dealing with asynchronous operations and handling errors gave us a good background for building websites.

Collaboration enabled us to effectively use GitHub, where we enhanced our skills in code review and learned the need for clear documentations throughout our development process. We also enhanced our skills on time and task management, setting priorities and changing timelines if needed.

Visible issues like unpredictable API responses, complexity of state, and conflicting schedules prompted us to review our methodologies. These were resolved by adding wrapper functions, reorganizing components, and improving our team communication. We also bridged our skill gaps by effectively splitting the works.

Some of the most important learnings that we achieved in this project were the importance of starting with a minimal viable product, maintaining good group communication, writing documentations thoroughly, and emphasizing early testing. These learnings impacted our methodology and made the project a major milestone in our development journey as Computing Science students.

Group Members and Detailed Contributions to the project:

- **Austin Philip**: Took the lead in developing the initial website design, laying the foundation for the overall user interface, which was later refined by Rain. Implemented the ChatGPT API integration and designed the logic to effectively coordinate multiple APIs across various React components, ensuring seamless functionality throughout the application.
- **Han Chen**: Focused on CI/CD by creating tests using Cypress to validate API, test overall navigation, and ensure core functionalities. These tests significantly improved our website's stability and caught bugs that were not present with manual testing. These tests were also automated by Github actions to test everytime we committed something. Found bugs in the functionalities of the website and fixed them with the help of Austin. Implemented the Chatbot that used the ChatGPT API and formatted how the response of the AI looked as well as took in real user testing responses from the in-class session to implement

auto-scrolling for the AI Response and with the help of Rain, AI typing indication was also added.

- **Jinyan Jiang**: Focus on TripAdvisor API integration and data processing. I display dynamic content by obtaining multi-dimensional information such as hotels, locations, and images. Using React to build the interface call logic, I developed the search function based on the user's input of city and time to realize the real-time query and display of hotel information. At the same time, I encapsulate the API request process for location search, hotel list and image resources to solve the problem of cross-domain restriction of the official API, and use Node.js to build a relay server to ensure stable access to data in the local development and deployment environments.
- **Rain Zhang**: Focusing on UI/UX design and implementation, creating a visually engaging travel planning experience. Designed and developed the entire user interface from scratch, including the multi-step question flow, animated background elements, responsive cards, AI chatbot functionalities, and interactive categories sections. Built all UI components using React, with careful attention to details and user experience.

Changelog table including any revisions since previous milestones:

- Backend Implementation:
    - We created a backend layer to handle API requests for TripAdvisor, since the API did not accept requests from a **DOM environment**.
    - Backend Repository: https://github.com/f4ncy1zach/cmpt276-group10-backend
- Switched AI Model:
    - We replaced **DeepSeek** with **ChatGPT** for improved conversational capabilities and reliable data generation.
- User Interface Overhaul:
    - We reworked the **User Interface** for a more intuitive and streamlined user experience.
- Questionnaire System Update
    - We modified the structure and flow of the **questionnaire**.
    - We do not ask users for their budget and current city to simplify the user input process.
- Fallback Mechanism for Trip Data
    - Initially we planned on having TripAdvisor generate every information related to the location. We later found out that TripAdvisor don't provide enough data for lesser known places.

    - For this reason we integrated **ChatGPT** as a fallback source when **Tripadvisor API** lacks to provide sufficient data
- Removed Safety Index:
    - We completely removed the **Safety Index** feature, as this information is not provided by **Tripadvisor** and could not be reliably sourced.
- Hotel Pricing Update:
    - We replaced direct **hotel price** listings with a **Hotel Price Index** to give a broader cost perspective and because tripadvisor does not provide the relevant data.

Peer Testing Feedback Form & Feedback Results:

Peer Testing Feedback Form Link:
https://docs.google.com/forms/d/e/1FAIpQLScm1e4lyucAJbwY8v22WK5lsOWUJ0VewNGME5EfVN8PVzatnQ/viewform?usp=dialog

Our list of feedback from the peer testing was:

- The chat not automatically updating to its latest from the chatbot UI
- The formatting can be better for the AI chat, as it was hard to read answers that were given in bullet points
- There was no indication to show that the AI was typing. If this could be implemented or a message to show the user that the AI is typing/thinking/done would be nice

All of these issues were taken in count for and we implemented fixes for all 3 issues

Any Additional Diagrams, Charts, and Tables: