**Step Two Project Proposal**

**1. What goal will your website be designed to achieve?**

Our website's main aim is to make travel planning easier and more enjoyable. Instead of just looking at the cost, we want to help travelers consider another important factor: the weather at their destination. By giving users flight details and historical weather info, we hope to help travelers make smarter travel decisions for a better trip.

**2. What kind of users will visit your site? In other words, what is the demographic of your users?**

The primary users of the website are travelers of all demographics, including individuals, families, and business travelers. The demographic of users varies in terms of age, travel frequency, and travel purposes.

**3. What data do you plan on using? You may have not picked your actual API yet, which is fine, just outline what kind of data you would like it to contain**.

Flight Data: The project plans to utilize the Amadeus Flight API (https://developers.amadeus.com/) or something similar to fetch real time flight information, including flight schedules, availability, and pricing.

Weather Data: Our website will use the Tomorrow.io Weather API (https://www.tomorrow.io/weatherapi/) to access a range of weather information. This versatile API will provide historical weather data for longterm trip planning, current conditions for immediate travel, and forecasts for upcoming trips. The displayed data will depend on the selected travel dates, ensuring users receive the most relevant and timely weather information for their trip planning needs.

**4. In brief, outline your approach to creating your project (knowing that you may not know everything in advance and that these details might change later). Answer questions like the ones below, but feel free to add more information:**

**a. What does your database schema look like?**

The database schema will primarily include tables for storing flight information and weather data. Depending on workload and complexity, it may or may not include a user table.

Flight Table:

- flight\_id (Primary Key): A unique identifier for each flight

- departure\_city: The starting city for the flight

- arrival\_city (Foreign Key): A reference to the destination city's location\_id in the Weather Table

- departure\_date: The date of flight departure

- arrival\_date: The date of flight arrival

- flight\_number: A unique identifier for the flight

- Optional: Additional details about the flight, such as airline, ticket price, and flight duration

Weather Table:

- location\_id (Primary Key): A unique code for each location (city or airport)

- date: The date when weather information is recorded

- temperature: The recorded temperature on that date

- precipitation: Details regarding any precipitation, like rainfall or snowfall

- wind\_speed: Data indicating the speed of the wind

- Optional: Other weather-related data for the city

**b. What kinds of issues might you run into with your API?**

Potential issues with the APIs might include rate limiting, API key management, and ensuring data accuracy.

**c. Is there any sensitive information you need to secure?**

For the weather and flight application, there isn't really any sensitive user-specific information that needs to be secured. However, if user accounts are incorporated in the future, there would be a need to secure sensitive user data such as usernames, emails, and passwords, as well as any user sessions for data protection. Additionally, if the application has a feature that stores user preferences or current city information in session or cookies, then that data would also need to be handled securely.

**d. What functionality will your app include?**

The app will allow users to search for flights by specifying their departure and destination locations, travel dates, and preferences. Simultaneously, it will fetch weather data for the destination.

**e. What will the user flow look like?**

Users will start by entering their travel details on the homepage. After submitting the search, they will be presented with a list of available flights and the weather forecast at their destination.

**f. What features make your site more than CRUD? Do you have any stretch goals?**

The website will offer more than CRUD operations by providing real time flight information and real time or historical weather information, making it a valuable tool for travelers. A stretch goal may include implementing a user feature that allows users to save their trip searches and preferences for a more personalized experience.