# Assignment 3

## 1. Team member (Team 77)

* Sugeily Valentina Angulo-Limon
* Aiden Hai

## 2. Assignment description

### Project Title

* Travel Assistant Web Application

### Project Description

* Travel Assistant Web App is a simple (or complicated) digital tool designed to help travelers plan and organize their trips in a single, easy-to-use platform. It will enable users to create detailed itineraries, check real-time weather, set reminders (maybe based on location and/or time), and access location-based recommendations and tips. These are all essential travel planning features. By combining them, this app aims to streamline the travel experience, making it more manageable, enjoyable and stress-free.

### Project features

* Core features
  + Itinerary Planner: we are going to create an itinerary card collection like a to-do list, separated by date, where users can add, edit, delete items (Using React.js may help uniform the style). Clicking the destination, a pop-out detail card will show up. There will be different cards for various types of trip elements
    - Trip Events (in detail card): Includes details like destination, location, city, website link, contact info, start & end time, and usernotes.
    - Transportation: Displays departure and arrival locations, transport means, duration, and alternatives that can switch.
    - Dining, Accommodation, and Shopping: These are a part of the trip events, so the trip events should be categorized for a more organized experience, supporting filter.
  + Weather integration: Integrate a Weather API to provide real-time or forecasted weather for each destination, helping users plan for outdoor activities. This information will be displayed within the itinerary event’s detail cards.
  + Responsive Design: The app will be optimized for both desktop and mobile use, adjusting displays across different devices to ensure usability and accessibility.
  + Local storage: Data will be stored locally so users can access their plans even if they close or refresh the page. (HOW?)
  + A current time and/or calendar: Display current time and date on the main page to help users track time differences when planning across different time zones. This can also be used to differentiate the current, past and future trip events
* Intermediate features
  + Reminders: Set notifications for upcoming events (important events especially, where users can set priorities in the card). Users can receive time-based or location-based reminders.
  + Location based tips & recommendations: Provide suggestions for nearby attractions, dining, or shopping options based on the user’s current or upcoming destinations. These can be automatically added to the itinerary (detail card) for easy access.
  + Four tabs: Users can view events in four tabs: All, Past, Current, and Future, with options for further filtering by category (dining, shopping, transportation).
* Advanced features
  + User authentication (maybe): allow users to log in with their google account for easier account setup and log-in, helping with personalized access and saving across devices
  + Calendar view (maybe): Integrate a Calendar API, offer an intuitive calendar view where users can see all trip events laid out in a monthly or weekly format.

## 3. HCI perspective

### Target users

* Incoming travelers, who need a detailed trip plan to organize and manage their trips. This includes solo travelers, families, or friends planning trips together, and any kind of trip can be benefitted. However, a multi-day trip is preferred, especially a trip to another country, since they need a detailed trip plan to cover as many destinations as possible, and a robust trip management is necessary for a trip into a new place.

### A simple user story

* Lin, a 24-year-old student from China, is planning a 10-day solo trip to the U.S. for her summer vacation. She wants an easy way to organize her itinerary, check the weather, and find local recommendations.
* Scenario:

1. Planning: Lin uses the Travel Assistant app to create a daily itinerary. She adds the hotel, flights, and sightseeing spots, like Times Square and the Golden Gate Bridge.
2. Weather Updates: Before each city visit, she checks the weather forecast through the app to plan the activities and pack accordingly.
3. Reminders: The app sends Lin reminders for important events, like catching his train or arriving at the airport on time.
4. Location Recommendations: While exploring, she uses the app to find nearby dining and attractions, making it easier to explore new places confidently.

### HCI requirements

* Error prevention:
  + Use interactive calendar and clock components to prevent user input errors, especially for setting dates and times.
* Error recovery
  + Include a checklist for marking event items as completed, which automatically moves them to the Past tab but allows users to revert them to Current or Future if needed.
  + Deleted items can be sent to a trash bin, where they can be restored if deleted by mistake.
  + Auto save when entering the details of the item (but may be hard to achieve)
  + A cancel button to close the pop-out detail card.
* Simple UI
  + Focus on a clean, intuitive interface that minimizes the learning curve, especially for new users accessing the app during travel.
  + Use common icons (notification with a bell, Deleted items with a trash bin, and so on)
* Cross-Platform Usability
  + Design with responsive elements to ensure the app works seamlessly on both desktop and mobile platforms.
* Accessibility Concerns about Robust:
  + Full screen-reader support may be challenging to implement because it will probably be difficult to have a perfect explanation of the trip events card, but we will focus on clear labels and a simple layout to support basic accessibility needs.

## 4. Template & Pseudocode

### Template

* Top bar
  + Website
    - Left: user icon to go to the top, or a list icon/calendar icon to change between <List View> or <Calendar View>
    - Mid-right: notification icon, account setup & login
    - Right: <My trips> for more than 1 trip, and create a new trip (must be better if opening a new page, but probably time consuming)
  + Mobile: not a mobile app, so the overall will not be changed too much
* Middle top (or place it to the left, vertically placed)
  + Filter icon: all, checkbox for each categories
  + Four tabs: ALL, CURRENT, FUTURE, PAST
  + “➕”: add a trip item to the trip, should be obvious. On mobile one it can be placed at the bottom right
  + Trash can icon: save deleted items
* Middle middle - MAIN
  + Date based, date for each card-collection (can be opened and closed -> need a > / v icon at the left of the date)
  + For each item card, from left to right
    - Icon / checklist icon
    - Destination / Self-created one-line description, can be clicked to the details card (or the whole card can be clicked to enter the details card)
    - Start & end time
    - Location: can be clicked for weather [or can be put only to details card)
    - Delete button
  + For transportation card
    - Not a card like
    - Two destinations, one for each side, connection line with the traffic mean and the duration time
    - Whole card can be clicked for edit
    - Left - a switch button to allow transportation alternatives
  + Details card - clicked and pop out, glass opacities for the background
    - Destination at the top
    - Location: can be clicked for weather, call weather API with the location’s coordinates.
    - Start & end time
    - Phone number, email address, website link
    - User note
    - Bottom icons: edit (click for edit), delete (a pop-up delete message & double check will appear)

### Simple pseudocode for MAIN

Function createEventItem(date, title, location, startTime, endTime, category, notes):

Initialize an item object with given parameters

Assign a unique ID to the item

Save the item to local storage

Return the newly created itinerary item

Function editItineraryItem(itemID, updatedFields):

Find the itinerary item by itemID

Update item fields based on updatedFields

Save changes to local storage

Return the updated item

Function deleteItineraryItem(itemID):

Find the itinerary item by itemID

Remove item from the itinerary list

Move item to the trash bin

Update local storage

Function displayItinerary():

For each date in the itinerary:

Create a date section

Fetch all items for the date

For each item:

Display item details in the card

Add event listener for clicking to enter details card

Function toggleDetailsCard(itemID):

Find the itinerary item by itemID

Open the popup detail card of the item

Allow user to edit, delete, or close the details card

### Code for fetchWeather.js with API implemented

Please check the code in my [github repository](https://github.com/AiiiiDannn/Trip-Assistant-Web-App/blob/main/fetchWeather.js).

## 5. Feedback from the instructional team

* Is it possible to achieve everything, especially given this less than one-month time frame?
* What are the most important features & what can be discarded if time is not enough?
* Are there any suggestions for improving the layout or navigation of the app?