Travel itinerary

* Project name

Atlas Itinerary

* Team member names

List each person’s name.

Gabriel Taylor, Dustin Zook, Sorelle Djuissi, Gabriyel Sorensen

* Abstract

This project proposes a personal travel itinerary planner that helps people design detailed, time slotted trips without the pressure of booking. After a user creates an account, they can build multiple trip plans, each focusing on a chosen destination. Within a trip the user can select their preferred lodging, restaurants, activities, and attractions, then schedules them into specific days and times to create a daily flow on their trip.

* Description

The purpose of Atlas Itinerary is to give people a simple but effective tool to organize their trips without the pressure of booking. Most people begin trip planning with scattered notes, random websites, and fluttering ideas, but have the problem of not knowing how to fit all the pieces of their trip together. A user might know which hotel they want, which restaurants they want to try, and which attractions are a must see, but without a daily structure these plans can feel overwhelming or incomplete. Atlas Itinerary addresses this by offering a central place where users can design, customize, and save detailed trip schedules that show what they can do and when they can do it.

The usefulness of this project reaches a wide range of users. Casual vacationers can use this system to draft itineraries so that they can balance leisure time with sightseeing. Families can plan days that work around meal time, nap times, and attractions that appeal to different ages of the family. Even students getting ready for study abroad trips or other types of group travel can use Atlas Itinerary to create their day by bay plans that can be adjusted as they go. This system offers clarity and flexibility, helping the user reduce stress that can come when planning and managing multiple trip details.

There are existing systems for trip planning, but those systems lack in important ways. Systems like TipIt and Wanderlog rely on importing reservations and booking confirmations, while that makes it good for organizing purchased travel, it lacks possibility for brainstorming and having flexibility while planning. While other options like physical notes or note apps have the space for full lists that allows for creativity, but lacks time scheduling, and visual choices during the planning process. Google Trips used to offer a similar solution to this problem but was discontinued in 2019 leaving users without a solution or tool for their trip planning. Because of the lack of tools, users use a combination of documents, notebooks, calendars, and others that leads to confusion during the trip planning.

Atlas Itinerary sets itself apart from competitors by focusing strictly on the trip planning rather than purchasing and booking. Without the need to purchase allows users to create and experiment with multiple trip plans before committing to reservations and bookings. One of the main features of the Atlas Itinerary is the use of time slots. Instead of just listing an assortment of attractions, restaurants, hotels, etc., the user decides when an activity or event fits into the day. Lodging is assigned with nights, meals are in breakfast, lunch, or dinner windows, and the attractions and events will have specific time frames set by the users, as this creates a daily timeline that the user can follow. As users can create multiple trips under one account, the users can compare itineraries, duplicate itineraries to change elements.

The main productivity of flexibility, organization, and time based structure, Atlas Itinerary offers a clear advantage over other systems. It gives the user a creative way to imagine their trips without the pressure of booking and the freedom to adjust and refine the plans. Atlas Itinerary is a travel planning tool that is not just about where to go, but how to make the most out of each day of the trip.

* Feature list

A list of features that will be completed by the end of the semester

1. User Accounts: users can sign up, login, save their trip itineraries
2. Trip Management: users can create multiple trips. Can also edit, delete, save, and view their trips.
3. Lodging Selection: users can add hotels to a trip and assign the number of nights.
4. Dining options: users can add restaurants and pick the specific meal of the day (breakfast, lunch, dinner).
5. Attractions and Activities: users can add events, attractions, and activities and choose the time frame for each.
6. Daily Timeline View: trips are displayed in a day by day schedule showing the hotels, meals, and activities the user has chosen.
7. Mobile Access: the system will be accessible on mobile so users can use all the features on the go.

A list of features that will be completed if there is time

1. Trip Duplication: users can duplicate an previous trip to modify it as a new trip.
2. Comparison View: users can compare two itineraries side by side.
3. Drag and Drop: users can rearrange items in their itinerary by dragging them into new slots.

A list of features you would like to implement, but cannot be completed this semester.

1. Interactive Map: users can view locations of hotels, restaurants, and attractions on an interactive map.
2. Plan Sharing: users can share their itinerary with other users and contribute to the plans.
3. Suggested Itineraries: recommend preset itineraries based on users trip destination.

The list should contain the core features of the project.

What does it do to accomplish the project?

What will the user be able to do?

How will the system help the user?

Do not focus on how the project will be built. Do not mention coding solutions, development tools, or specific technology.

* Initial set of technologies

List all technologies the team believes will be needed to develop and run the project. Explain why the team will need each technology.

The following is a non-exhaustive list of technologies that may need to be mentioned:

Platform, e.g., desktop, phone, web, console, etc.

Operating system, e.g., Windows, Mac, Android, etc.

IDE, e.g., IntelliJ, Visual Studio Code, NetBeans, etc.

Languages, e.g., Java, TypeScript, HTML, etc.

3rd party frameworks, libraries, and tools, e.g., Bootstrap, Tailwind, PDF.js, Toast UI Calendar, etc.

Server software, e.g., React, Vue, Flutter, etc.

Communication software, e.g., Discord, Slack, etc.

AI platform, e.g., ChatGPT, Claude, Perplexity, etc.

For AI, also indicate how the team plans to use AI.

Which plan will you be using? Free, Pro, etc. All team members must use the same plan.

If the team does not plan to use AI, indicate that too.

Note:

All team members must use the same platforms

Can use more than one AI platform

* Server information

If you intend to implement a server component, your project must run a service provider. Examples of internet services previous students have used include Linode, Firebase, Supabase, and Vercel. Any service may be used, but the instructor must approve them first. Your instructor may be able to point you to free services depending on your project.

Server components may not be hosted on laptops, lab computers, or home computers.

Do not select AWS, Azure, or GCP unless the team has significant experience with these platforms. They are large and complex.

* Data sources
  + TripAdvisor Content API https://www.tripadvisor.com/developers
  + Google Maps platform API https://developers.google.com/maps/apis-by-platform
  + OpenStreetMap API https://www.openstreetmap.org/about

These API’s will allow for access to a dataplace of locations and attractions with descriptions and photos.

Some projects may rely on external data sources.

Identify the data and an initial source or set of sources from which you will get the data.

Include information about or links to licenses for each data source.

* Team members’ backgrounds.
  + Gabriel Taylor: Background in Computer Systems. Languages like C#, C++, JavaScript, Python, and SQL as well as base knowledge on other other programs like Racket and Haskel. Used to generating base ideas for endeavors and helping to push projects along.
  + Dustin Zook: Experience with Java, JavaFX, HTML, CSS, bootstrap, Javascript, API’s, JSON, and SpringBoot. Databases I have done work in are MySql, Microsoft SQL Server, and Azure.
  + Gabriyel Sorensen: Experience with Java, JavaFX, C#, HTML, CSS, JavaScript, Python. Used Microsoft SQL.

Provide a short background of each team member’s familiarity levels with the technology used on the project.

Not everyone needs experience with every technology used in the project.

There may be technology that no one on the team has experience with. That is ok, as identifying this means that there is a risk that needs to be considered and managed.

Include descriptions of each member’s primary responsibilities in the project.

* Dependencies, limitations, and risks

Large projects carry many risks. Successful projects begin by identifying as many risks as possible before the team starts any design or development.

List any dependencies between tasks and how they may put the project at risk. For example, suppose you are making a client/server application. In that case, the majority of the server code may have to be implemented before you can start working on sending messages back and forth between clients and the server.

List any limitations that may put the project at risk. For example, you may be using a service that limits how many requests you can make each day. Another example would be discussing team members who may not have the background required for their portion of the project.

List any other risks that may prevent you from completing the project

Explain how the team will handle the issue for each dependency, limitation, and risk.

* Timeline

Provide a weekly timeline that outlines the work that each person will need to complete each week on the project from now to the end of the semester. Base your schedule on the deadlines listed in the introductory notes.