## Project Name

Atlas Itinerary

## Team Member Names

Gabriel Taylor, Dustin Zook, Sorelle Djuissi, Gabriyel Sorensen

## Abstract

Atlas Itinerary is a personal travel planning web app that lets users design detailed time-slotted trips without the pressure of booking. After creating an account, users can plan multiple trips, search destinations, and add lodging, restaurants, attractions, and activities. A defining characteristic is the timeline where every item can be placed on a specific day and time, so the schedule is easy to see and follow.

The app focuses on straightforward trip management, allowing users to create, edit, and save itineraries, assign lodging to nights, and meals to time slots, and add items to a daily schedule. The goal is to simplify the planning by keeping choices and timing in one place. When it's time to book, the plan is already organized. The web app also works well on phones for quick updates while traveling.

Tools & Technologies

* Platform: Web. Atlas Itinerary is a web-based program. Most of our group knows web-based program functionality, so by making one, we are more prepared for the task.
* Browser: Firefox. Firefox is the search engine on which we will make our web-based program. Our group is more familiar with programming web applications on Firefox.
* Operating System: Windows. All of our members have Windows operating systems by default. Instead of trying to learn how to make the application on Apple through a VM, we have elected to use Windows.
* IDE: VSCode. Useful for coding in HTML, CSS, JavaScript, etc. Incredibly versatile.
* Languages: HTML, CSS, JavaScript, TypeScript, and React. Necessary for designing web pages and understanding how users will interact with the project..
* Server Software: Vercel. Able to link with the Supabase Postgres database.
* Database: Supabase. Postgres database to use with Vercel.
* Communication Software: Discord. Allows for communication outside of class. Non-professional contact method to encourage bonding and familiarity among members.
* AI: not to use for coding, but to ask basic questions.

## Requirements list

1. Sign-Up Form
   1. Upon accessing the domain, users will be greeted with a sign-up screen
   2. The signup page offers three boxes:
      1. One for username
      2. One for email
      3. One for password.
   3. If the username is already in use, below the username box, the user will be told that the username already exists
   4. Passwords will have several requirements:
      1. To be between 8 and 16 characters
      2. At least one uppercase letter
      3. At least one lowercase letter
      4. At least one special character
      5. At least one number
   5. On the signup page, there is an option to sign up through the user's Google account.
   6. Login hyperlink at the bottom
2. Sign-in Form
   1. Two boxes, one for username, one for password
   2. If the login is not recognized, a message appears below the username and password box letting users know it is not recognized.
   3. Need help option for recovering the user name and/or password
      1. Two recovery options will be provided:
         1. One for retrieving the username
         2. one to reset the password
   4. Option to sign in using a Google account
3. Home Page will be rendered once a user logs in
   1. About Section
4. Navigation Bar
   1. Create Itineraries
      1. Once clicked, the user will be presented a page where they can create itineraries
      2. There will be a label for their Travel Destination
      3. There will be an input box for their Travel Destination
      4. The application will be able to detect the user’s current location and generate suggestions for their Travel Destination
         1. Once the destination searched for is found, the user will be presented with a sidebar with several categories
            1. Lodging is the Default category selected

Hotel

Bed and breakfast

Specialty lodging

* + - * 1. Dining

Can be sorted by Subcategories

* + - * 1. Attractions

Subcategories

Filters for attractions include

* + - * 1. A drop-down menu will be above where locations are displayed, containing a variety of sub-categories to filter locations
        2. Once a category is selected, the locations of the category will be displayed as a list of cards

Cards will consist of:

image of the location

the name of the location

* + - * 1. There will be a paging system so that the maximum number of cards will be displayed on each page
        2. Cards will be ordered by the rating of each location
        3. When a card is clicked, a modal is displayed

Images of the location are displayed

A Description of the location is displayed

Ratings are displayed

The modal will have a button to add to an itinerary they’re building

When attempting to add a location to an itinerary, users will be presented with a popup requesting the date and time they will go to that location

An input box for the start date

An input box for the end date

An input box for the start time

An input box for the end time

The modal will have a button to cancel, closing the modal

* + - 1. Once a location is added to an itinerary, a sidebar on the right page will appear
         1. Sidebar will display the current itinerary being built

Displayed as a calendar

Saved locations are displayed within this calendar as bullet points for each day

Days may be expanded

Locations/events are listed in chronological order

Locations/event may be selected

When selected, the location/event timeslot may be moved around

When selected, the location/event may be deleted

Days may be selected

When selected, that day may be deleted

* + - * 1. Save As Button

When clicked, user is prompted for the name of the current itinerary

Once named, the built itinerary is saved

* 1. View Itineraries
     1. Once clicked, the user will be presented a page with all their built itineraries
        1. Itineraries will be displayed as a list of Cards
           1. Cards will consist of the name of the itinerary, the start date, and the end date of the itinerary
        2. Once you click on the card, the card expands into a more detailed view
           1. The view is a calendar view that has bullet points for each event that will occur on each day

Clicking on the calendar day expands it more, and the events scheduled for that day are listed in chronological order.

The calendar day starts in view mode

The calendar day will have an “edit mode” button that will allow users to move into edit mode

Edit mode will allow users to select events

Selecting an event will allow users to delete that event.

If a user chooses to delete an event, a warning alerting the user that they are about to delete the event will pop up.

The alert will offer a chance to continue and delete the event.

The alert will also allow users to change their minds and keep the event.

Selecting events will also allow users to move events

If an event is not available at that set time, a warning alert will be provided

* + - * 1. Edit mode for calendar that allows for selecting a day, deleting, and moving around of days
        2. Selecting a day will allow users to delete the event’s day.
        3. If a user chooses to delete a day, a warning alerting the user that they are about to delete the events for the day will pop up.

The alert will offer a chance to continue and delete the day’s events.

The alert will also allow users to change their minds and keep the day’s events.

* + - * 1. Selecting a day will also allow users to move a day’s events to another day.

If a day’s events are not available at the set times on the day they are being moved to, a warning alert will be provided telling the user which events are not available on the day they are being moved to

* + - 1. A deleted option will be provided for each itinerary
         1. Once clicked, a warning message will be displayed asking if the user is sure they want to delete the itinerary

Will default to no

If the user clicks yes, the itinerary will be deleted

* 1. Account
     1. Once clicked, the user will be presented an account page
        1. This page will display user’s current information:
           1. Their username
           2. Their current email
           3. Their current location
        2. User is presented with some options:
           1. Change password

When clicked, user must type in a new password and re-type to confirm

Once confirmed, the password for the account is updated

* + - * 1. Sign-out

Once clicked, the user is signed out of their account

## Timeline

1. Week 2 (Sep 1-5):
   1. Draft and submit the project proposal (all members)
2. Week 3 (Sep 8-12):
   1. Project requirements document (all members)
   2. Slideshow presentation (all members)
3. Week 4 (Sep 15-19):
   1. Finalize requirements document
   2. Present the slideshow on features
4. Week 5 (Sep 22-26):
   1. Set up project structure(all members)
   2. Set up server on Vercel and database on Supabase(Dustin)
   3. Educating team members on how Vercel and Supabase work and aiding in database setup by pitching ideas for database information(Taylor)
   4. Begin basic account creation and login system (front-end: design and style login UI) (Sorensen)
   5. Implementing Google OAuth
   6. Creating basic navbar (Home, create, view, account)
   7. Build simple API calls in a test to fetch hotels or restaurants by location(Dustin)
5. Week 6 (Sep 29-Oct 3):
   1. Create category sidebar (lodging default) with filter option
   2. View itineraries and create the itineraries page (cards with info for locations)
   3. Implement trip management-CRUD operations (Sorensen)
   4. Develop trip management UI components (create, update, delete trip pages)(Taylor)
   5. Implement user storage for the database(Dustin)
6. Week 7 (Oct 6-10):
   1. Right-side “building” sidebar with a removal option
   2. Implement the lodging selection of adding hotels and assigning nights(Taylor)
   3. Implement dining options of selecting restaurants and the meal assignment (Sorensen)
   4. Integrate API’s into lodging and dining features(Dustin)
   5. Build lodging/dining UI screens and integrate with API results (Sorelle)
   6. Save itinerary to the account feature
7. Week 8 (Oct 13-17):
   1. Implement attractions and activities features with time slots(Taylor)
   2. Add trip data handling, such as trip creation, lodging nights, restaurants, and activities (Sorensen)
   3. Integrate API for attractions and activities lookup(Dustin)
   4. Develop attractions/activities UI and calendar input fields (Sorelle)
8. Week 9 (Oct 20-24):
   1. Implement a timeline view, day-by-day schedule
   2. Connect the timeline view to the database records (Sorensen)
   3. Connect API data to the timeline view(Dustin)
   4. Build interactive timeline/day-by-day itinerary view UI (Sorelle)
9. Week 10 (Oct 27-31):
   1. Review the core features and make adjustments if necessary(all members)
   2. Test database with multiple users and trips(Dustin)
   3. Sorelle: Refine UI/UX consistency (colors, layout, navigation)
10. Week 11 (Nov 3-7):
    1. Implement the extra features if we have time(all members)
    2. Optimize API calls to reduce duplicate calls(Dustin)
    3. Assist with UI integration for extra features (Sorelle)
11. Week 12 (Nov 10-14):
    1. Conduct full system tests(all members)
    2. Write a technical report(all members)
    3. Support usability testing and document UI design choices (Sorelle)
12. Week 13 (Nov 17-21):
    1. Finalize implementation/make sure all features are implemented and working(all members)
    2. Polish the frontend and user experience (Sorelle)
13. Week 14 (Nov 24-28):
    1. Final round of testing(all members)
14. Week 15 (Dec 1-5):
    1. Final presentation(all members)