

RoadTrip Runner

Emily Gerst

(<https://github.com/emmykg38>)

Melanie Gunn

(<https://github.com/mgunnhawkins>)

Valerie Tonsor

(<https://github.com/valerietonsor>)

Krystal White

(<https://github.com/krystalwhite>)

Description

RoadTrip Runner provides users with the tools to create carefully-curated and memorable road trips. Authorized users interact with a clear and easy-to-use interface in which users enter their starting and ending locations, and then the application renders a map with those locations as well as all National Parks along their route. Users can select National Parks to add to their trip and then save the trip for future reference. RoadTrip Runner allows the everyday user, young or old, to plan their ideal National Park tour.



Features

Authentication and Authorization: Individuals must register with the app, using a complex password, and their login information is saved to the database. Users login to utilize the app.

Autocomplete: When planning a trip, users are aided by the Google Maps Autocomplete API when entering in starting locations, destinations, and choosing National Parks to add to their trip.

Informative and Interactive Map: When planning a trip, users see an interactive map that displays the starting and ending locations, the most efficient driving pathway connecting the two locations, and all National Parks within 100 miles of their route. Each National Park is pinned with a marker and a comprehensive information window to assist the user in determining which National Park(s) would be best suited for them. Both the Google Maps API and the National Park Service API feed into the map, so the information is guaranteed to be up-to-date at all times.

Trip Repository: Users may save their trips, thereby linking the user to the trip within our database. The user may view their named trip at a future point.

Demo Video: Guests and users can view a demonstration video that details the features of the app.



Planning - User Stories

As part of our planning process, we brainstormed all the features we wished a roadtrip app would have and then we endeavored to make each of those features a reality.

Users enter in starting and ending locations and the Google Maps API renders the most efficient driving directions between the two spots and the National Parks API provides information about all the national parks within 100 miles of the most efficient route.



Planning - Database

The RoadTrip Runner database has tables for User and Trip. Each User is connected in a 1:Many relationship with the Trips, so that an individual user can save and display multiple trips within the app.

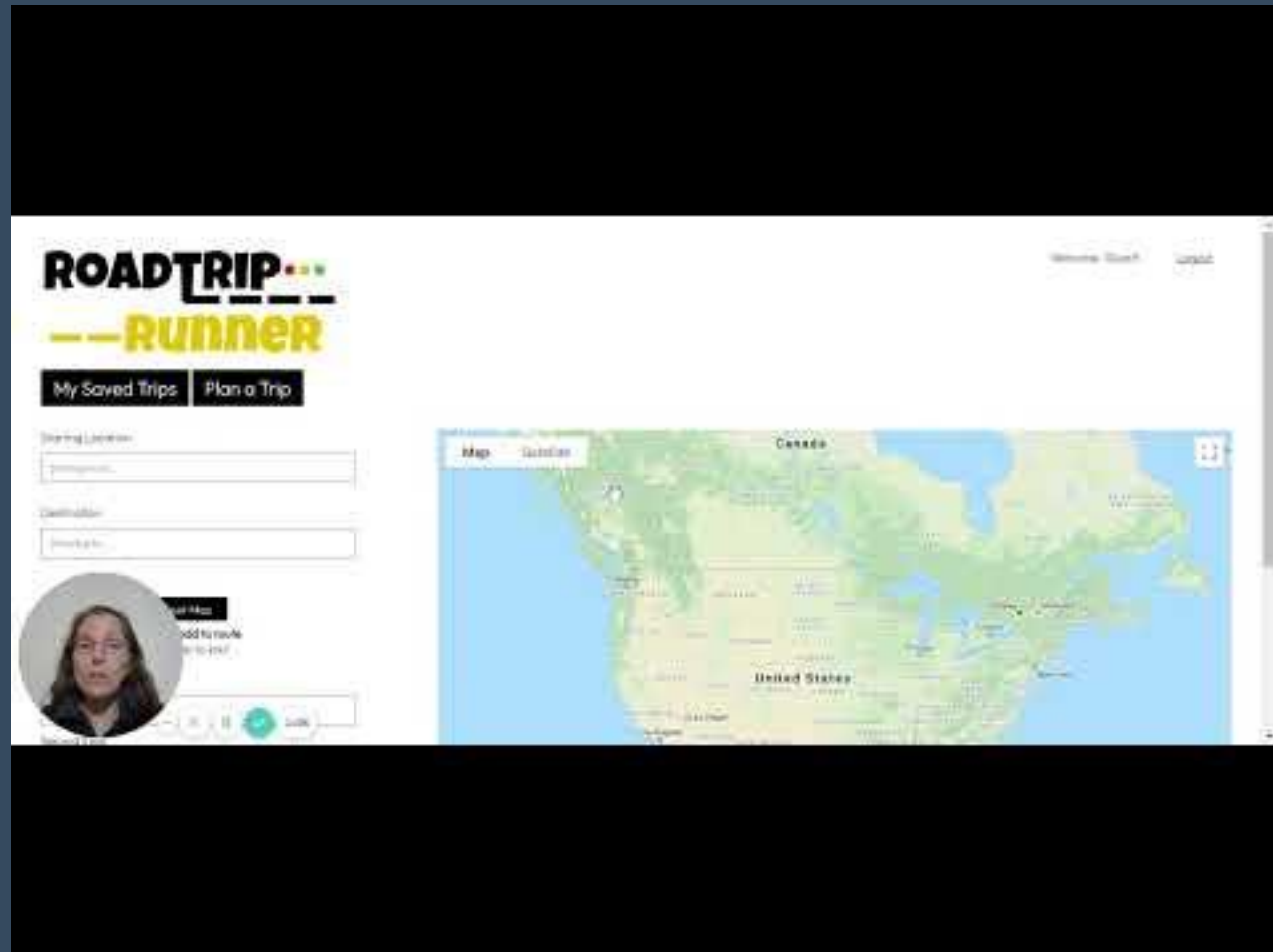


Technology Stack

- JavaScript, Java
- Thymeleaf
- SpringBoot
- MySQL
- Google Maps API, National Park Service API



Demo



What I Learned

- I learned an incredible amount about the Google Maps API and the National Parks API, as I was the primary person responsible for those areas for a significant portion of the project.
- I also learned the importance of brainstorming ideas with my teammates and pair-programming. Two brains always created better results than one.
- We should have used the Java Client for Google Maps.



What's Next

- Restructure the coding approach to use more of the Java Client for Google Maps Services.
- Have the user add National Parks to their trip in an easier way. Ideas include clicking on the marker or on checkboxes of options.
- Use the Google Places API or other APIs to include other possible locations to add to the user's route. Filter locations by price point, category, etc.
- Allow the user to suggest possible locations to be included in the location database for other users to select. Incorporate administration access to approve these user suggested places.
- Have the user have the ability to choose the detour radius.
- Have an itinerary page that easily allows the user to print the trip.
- Style the project with Angular rather than Thymeleaf.

