Destination Explorer: A Road Traveler's Adobe

Prajwal Nadagouda, Raviteja Gannarapu, Chirag Arora, Shlok Patel

Department of Software Engineering, San Jose State University

San Jose, CA

prajwal.nadagouda@sjsu.edu, raviteja.gannarapu@sjsu.edu,
chirag.arora01@sjsu.edu, shlokpankajkumar.patel@sjsu.edu

Abstract — Build an application that enables travelers to plan trips conveniently by removing all the innate hassle. Everyone enjoys traveling to new places to experience different activities. This application will personalize the travel plan and assist users from start to finish. The primary features of this application are destination recommendations, planning travel routes and itineraries, searching hotels, and packing suggestions.

I. INTRODUCTION

Destination Explorer is a tool that helps travelers with finding nearby events and locations based on their preferences. The application provides options for the users to set their preferences, get real time nearby events and connect with other users with similar interests.

II. ARCHITECTURE

There are two major components: Backend for database queries and user interface for the application.

<INSERT HERE>

Figure 1. Project Architecture

The frontend of the project was written in HTML,CSS and Javascript. The backend was written in Python using the Flask Framework and PHP. The database used in this application is MySQL which was hosted on the AWS Relational Database Server. The frontend and backend were each in their own AWS Instance.
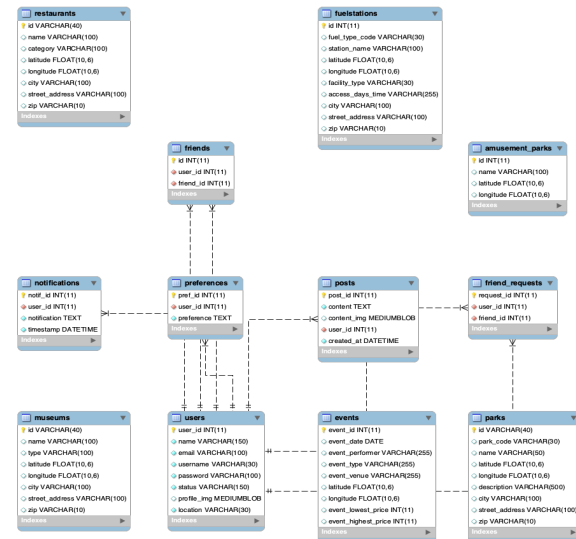
III. ER Diagram

Figure 2:ER Diagram

III. BACKEND

The backend of this application used the Flask framework. Database calls were handled in the backend of this application.

APIs were used for the purpose of real time data fetching and making recommendations

based on what the user wanted to be displayed on the feed.

A major API used in the project is

A. SeatGeek's API:

The SeatGeek API maintains comprehensive data about the current and live events taking place across most of North America. All responses can be served in JSON, JSONP or XML. The API makes extensive use of query strings to retrieve resources based around various search criteria.

The First major step is to create a database to store all the data about the events and locations that are to fed to the user ahead and during the trip. For that multiple SQL queries are executed to create tables: Events, Amusement parks, Fuel Stations, National Parks,etc.

Data Cleaning from the data received by SeatGeek API is done so as to make the data organized and to store that to the created tables.

Once the start and end data are fetched from the frontend part of the application several API calls are made to fetch data about the Live ongoing going events during the given time Frame, Fetching the Map and route data from the Leaflet API, mining data about the fuel stations, Electric Charging Stations, Parks, Museums and popular Food Places.

B. Leaflet Library:

Leaflet is the leading open-source JavaScript library for mobile-friendly interactive maps. We use Leaflet library to display the map on the application with the routes the user needs to take in order to travel to the desired locations. We also use it to display the recommendations for events and places on the maps as markers between the two locations. It would show a detailed navigation on the route with complete directional guidance using the database on the project.

V. CLIENT-Frontend

In the first phase of the project, we created designs of the application using Figma to get a better idea about the frontend implementation.

The application includes a home page, Post Feed page, Map page, User Modification page, signup page, a login page, Contact Us page and a Testimonials page. The home-page has an option to get a trial of the application with limited features. When a user attempts to search a location, a pop-up appears where the user can set the importance of various aspects of a markers and the map and routes are displayed based on what the user wants during his route.

The map is with detailed routes and recommendations along the way. The user can change the filters on the results page and can get a newly extracted map with the relevant markers. The feed page serves as a .
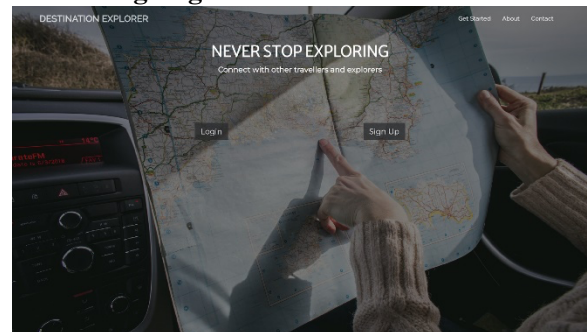
A. Landing Page

Figure 3. Landing page of the application.

The top right corner has the options to get to Account sign up, Learn more about the intiative and to contact Us. The main contents of the page include the options to login and signup.
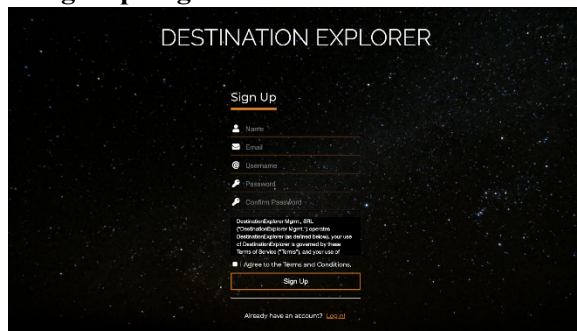
B. Sign Up Page

Figure 4. Sign up page of the application.

A user can enter the credentials as required to create an account and get started with the application. There is email validation and the terms of agreement to comply with the legalities of the application.
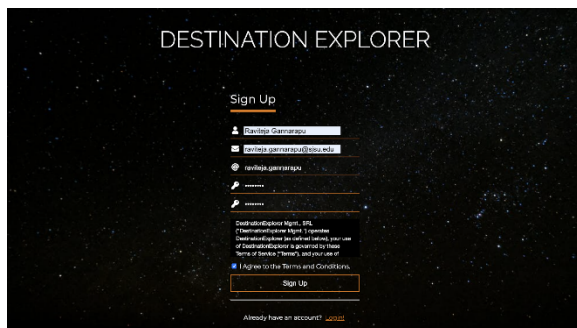

Figure 5. Trial user input sign up.

Figure 5 shows a user that is trying to create an account on the application.
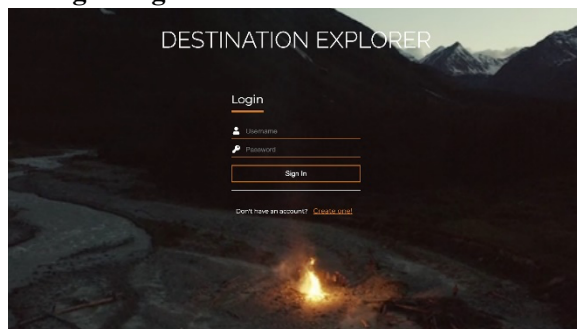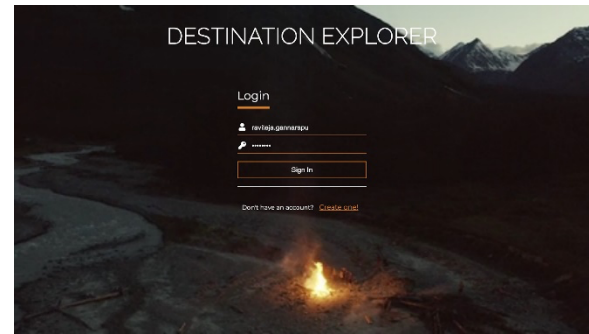
C. Login Page

Figure 7. Login Page interface

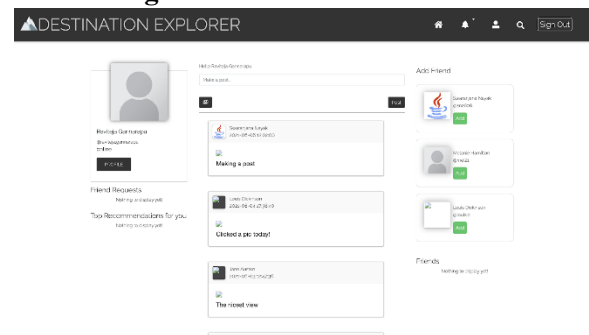**Figure 8.** Trial user input login

D. Feed Page

Figure 9. User Feed page.

Figure 9 displays the results for a trial user. Once the user is logged in, the user can access the interface similar to a social media application where there is a profile section which can be modified including Status, Display picture and contact information. The second part is the actual feed where a user is able to see the posts posted by people on his friend group similar to an Instagram feed. Then the user's Friend list is also displayed where one can send friend requests to new users or remove someone from their network.
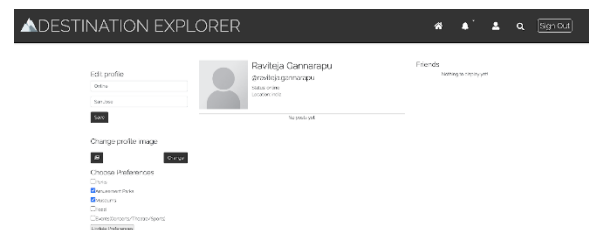

Figure 10. Update recommendations .

Figure 10 displays the results for a logged in user. A user can select the types of events and places he/she would like to explore and the type of content he would like to see on his feed.
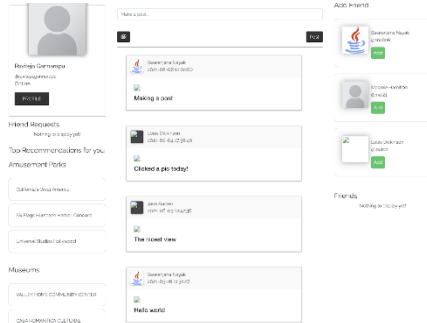

Figure 11. Updated recommendations in feed.

Figure 11 displays the results once the user has updated the content preferences on the application.
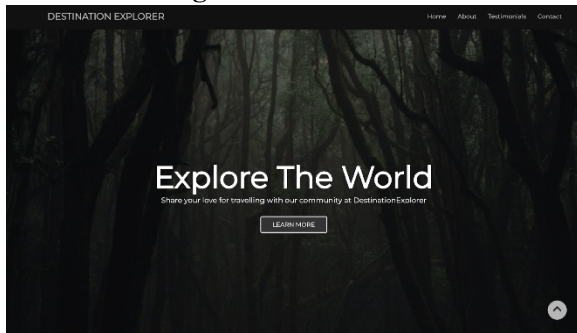
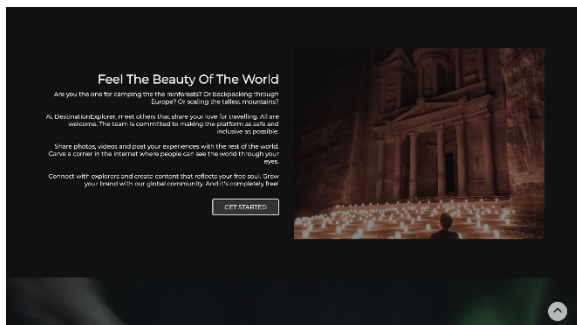### E. About Us Page

Figure 12: About us Page, Landing

Figure 13: About us Page, Extended
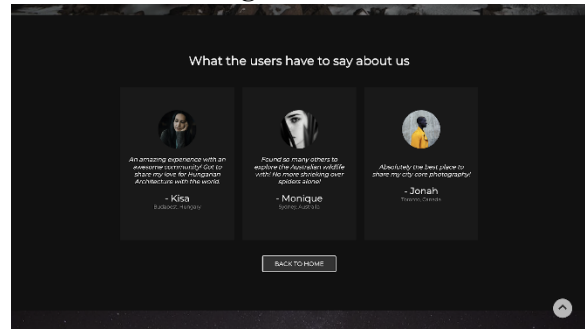
F. Testimonials Page

Figure 14: Past user Testimonials
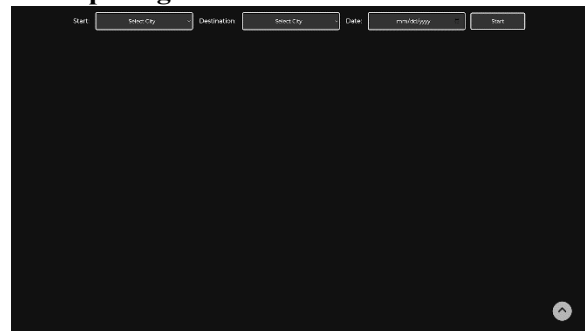
G. Maps Page

Figure 15: Getting Route and Recommendations

Figure 15 displays the Search boxes for the logged in user in order to select the locations where he/she needs to travel to and from and also the date of travel.
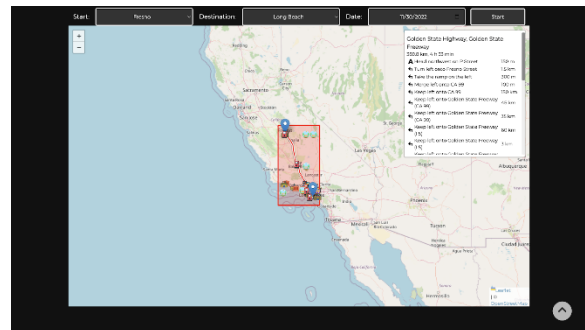

Figure 16: Map Output for desired location

Figure 16 displays the results once the user has entered the location and date of travel. This will make a call to the Leaflet library and a map will be displayed on the page. Then a call will be made to the SeatGeek API and data will be fetched and processed into the database and the recommendations for the particular route will

be displayed on the Map in the form of different markers once they are fetched from the database.

VI. EXPERIMENTS AND RESULTS

We conducted a series of experiments to test our tool against multiple testcases to verify the reliability of the tool and to measure the performance of the tool as the following:

A. Fresno to Long Beach

For the given locations, tool displays a wider view which we can see once zoomed with markers for just type of events based on the user Preferences.
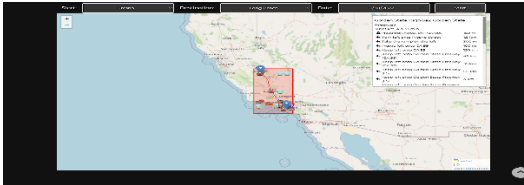

Figure 17: Route 1

B. San Diego to Oakland

Here the user has input the location from San Diego to Oakland for the date 2nd December,2022. As we can see there are various markers on the route displaying various events, food Places, National Parks and museums along the way. We can also see the pop head displaying the exact details about the location when hovered over.
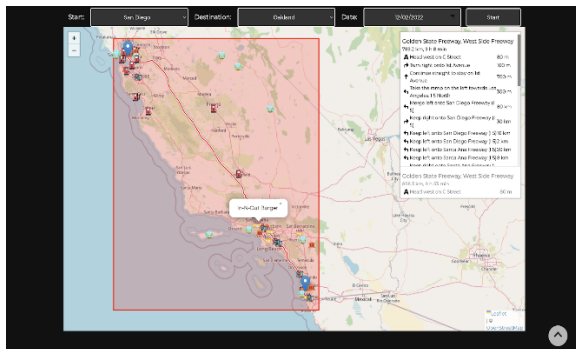

Figure 18: Route 2

C. Anaheim to Bakersfield
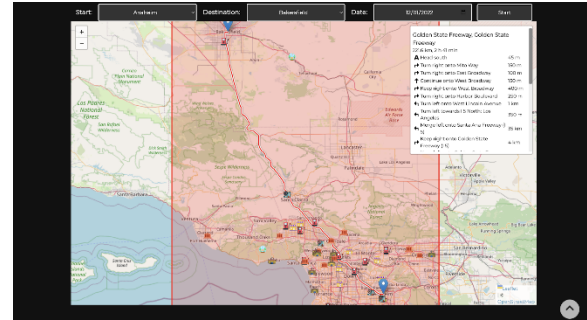Date:1st December 2022

Figure 19: Route 3

VIII. CONCLUSION

Destination Explorer helps travelers to take complete control of a trip and effectively plan out their travels and recommends the best places to explore based on their preferences. This helps them make an informed choice while planning their next road trips. It also gives users the ability to connect with other like-minded travelers and view their journey for the same route they are planning to go on.

VII. FURTHER IMPROVEMENTS

A. Additional Features
- The ability for users to send personal messages to other users using a Web-RTC or XMPP messaging protocol.
- Allow the user to search for more locations using the same preferences they initially set.

B. Optimizing SeatGeek API usage

We can optimize the SeatGeek API usage by storing information about a particular location in the database and scaling the database accordingly. Thus next time when a user searches for the same location, that information can be

retrieved from the database and minimally updating the database to fetch the latest events.

References:

- Events.
  https://api.seatgeek.com/2/events?per_page=50&venue.state=CA&client_id=MzAxNTQ4MTV8MTY2NzY4MzY2Ni41NjMyMjg4

- Fuel Stations.
  https://developer.nrel.gov/api/alt-fuel-stations/v1.json?fuel_type=all&status_code=E&access_code=public&restricted_access=false&state=CA&api_key=

- National Parks
  https://developer.nps.gov/api/v1/parks?stateCode=ca&limit=2000&api_key=

- Amusement Parks
  https://queue-times.com/en-US/parks.json

- Museums.
  https://www.kaggle.com/datasets/imls/museum-directory?resource=download

- Restaurants.
  https://www.kaggle.com/datasets/khushishahh/fast-food-restaurants-across-us