

# NPSAdvisor

## **Description:**

NPSAdvisor is a web application that will make it easier for people to plan out their visits to different national parks. It will serve as a resource for accessing useful up-to-date information about all parks managed by the National Park Service using data provided by the NPS API. Users will be able to find out more about every park's location, upcoming events, and popular activities all in one convenient place. The website will also provide detailed information about parking locations and availability at each park location to address the common issue of not being able to find parking spots. By building NPSAdvisor, we hope to encourage more tourism to US National Parks.

## **Usefulness:**

With its simple and intuitive user interface, NPSAdvisor will make it easier to plan out travel logistics while visiting different national parks. Unlike TripAdvisor, another popular travel website, NPSAdvisor will be geared more towards people interested in learning more about what to do at national parks. In general, other competitor sites tend to be cumbersome and archaic in design. On the other hand, NPSAdvisor will provide information in a manner that is more streamlined for ease of access and cuts out frivolous complicating features.

## **Realness**

Building this web application will require general information about the national parks as well as specific details about recreation, available amenities, upcoming events, and parking lot capacity. Using the National Park Services API, a service that provides comprehensive National Park Service data about parks with this designation, all the necessary data can be sourced. Prior to loading the database, several calls will be made to this API to retrieve it as several JSON files. The API will specifically make calls to `/parks`, `/thingstodo`, `/parkinglots`, `/events`, and `/amenities/parksplaces`.

## **Functionality**

The data is from various calls made to the National Park Services API. It consists of data related to information about the parks themselves, things to do, parking lots, events, and various amenities. Each of the API calls will ultimately represent one table in the SQL database being used for the backend of NPSAdvisor. The final SQL database will be primarily composed of string variables.

The landing page will list all parks that are a part of the National Park system along with specific details about each park, such as title, description, operating hours, entrance fee costs, and a link to the park website. From there, users can filter the list by state and what activities they'd like to participate in (e.g. Skiing, Astronomy, etc.). After the user selects a park they're visiting, they can click on it to learn more specific details about the activities and amenities available. Users can also navigate to the Events Calendar page to find out more about upcoming events happening at all national parks. Similar to the homepage, users can filter by state; in addition, there will also be date filters and event category filters. Lastly, the Parking Map page will display

the parking lots of all national parks on a map based on the Lat/Long coordinates retrieved from the NPS API. Once again, users will be able to filter by state and specific national park.

For our creative function, we will make a park description generator that takes all of the 'park description' data fields as input to create random fake park descriptions at the click of a button. We will use PyTorch and the Transformers library to use a quick pre-trained AI model. If this is too hard, we can simplify it by making a park name generator that will be smaller and requires less overhead. This function could be helpful for developing ideas for new national (or private) nature parks.

### **UI Mockup**

See "UI Mockup.pdf"

### **Project Work Distribution**

Since this project is centered around database design, we decided to use Flask, a lightweight Python framework that is suitable for developing small web applications. In terms of backend systems, Michael and Alex will be responsible for retrieving the data from the NPS API and cleaning the resulting JSON files. Shreya will set up the SQL schema and load the JSON data into the SQL database server. Frank will work on establishing the connection between the database server and the Flask application. For the implementation of specific features, Frank will be responsible for the home page, Shreya will be responsible for the Parking Lot and Things to Do page, Michael will be responsible for the amenities and Events page, and Alex will be responsible for the Events and the corresponding Events/Calendar page.