ParkScape Tech Report

Motivation

Our idea behind the website was to help people in planning their future trips. Parks, airports and cities come in together very often, as most people have to fly in order to visit a specific park, and most of the time they have to fly to a city first. In addition, people who are already visiting parks/cities may want to visit other parks/cities nearby. Our website essentially connects these three models and shows a good amount of information and multimedia of them. In addition, we plan to implement searching/sorting, which will make trip planning even easier.

User Stories

Phase 1

User Stories Written by Us

- Splash page
    - *“As a website user, I want an aesthetically pleasing splash page. I want it to explain the purpose of the website and guide me through how to use it. The splash page should direct me to the other pages where information is stored.”*
        - Response: This has been completed and added to the website.
- Search page for models
    - *“Implement search page for all three instance models. This should be done using HTML and Bootstrap. Instances need to be presented in cards.”*
        - Response: This has been completed and added to the website.
- About page
    - *“As a user, I want to learn about the website and how it was created. More specifically, I'd like to know what tools were used (any pre-existing platforms and APIs). I also want to know who the members of the team are, and what they contributed through GitLab statistics.”*
        - Response: This has been completed and added to the website.
- Create instance pages for cities
    - *“I am traveling to Denver next month and I want to learn a bit more about the city before I go. As a website user, it would be helpful to have some basic information about the city, such as costs, nearby airports, climate, etc. I also would like a map and picture of the city.”*
        - Response: This has been completed and added to the website.
- Create instance pages for parks
    - *"As the website user, I'd like to see not only the different nearby parks, but also information about them. More specifically, I'd like to know the state in which they're located, the amenities they provide, the activities available, and the cost of entry per person. Additional information like days of operation would also be helpful.”*
        - Response: This has been completed and added to the website.
- Create instance pages for airports

    - *“When using the website, I would like to see information about different airports in the area. I would like to know information about the airport, such as where it’s located and different attributes. I would also like to see visuals, such as a map.”*
        - Response: This has been completed and added to the website.

User Stories Written by Our Customer

- Filter cities based on money range
    - *“I'm traveling to California next month with my family of six and am unsure about which cities to visit. We're limited on money but are still interested in traveling for my children. I think it would be easier for us to use your site if I could filter cities based on a certain range of money it may cost to go to that city.”*
        - Response: This is outside the scope of this phase of the project, but it will be a feature added later.
- Language support
    - *“Hello, I am a foreigner from France planning on visiting America. I would like to use your website to plan my visit around New York and Florida, but my English is not very good. Are there plans to provide language support for languages that is not English like French?”*
        - Response: This feature is out of our scope for this phase 1 of the project, but could be considered in the future.
- City accessibility for disabilities
    - *“Hi, I have a disability that affects my movement and stumbled upon your site while looking for travel planning apps. I'm in need of an app that could tell me some sort of idea of how accessible cities are for people like me. Is that something that you guys are planning to implement or could implement?”*
        - Response: Hi, this functionality is out of the scope of this Phase of the project, as our data source for cities doesn't include accessibility information. In the future, we will try to implement this functionality and help you in planning your trips.
- Park image
    - *“I'm a visual learner and want to learn more about specific parks. I'd love to see an image of the parks that are available on your website. It'll give me more insight into the parks I'm considering traveling to.”*
        - Response: That's a great idea. We have added a picture for each of the parks when you view the Parks page and each park has more pictures when you click Learn More. Hope that helps!
- Airport ratings
    - *“I'm a frequent flier and want to make sure that I go to the best airport. I'd like to see ratings on your website of the best airports so I know which ones I should go to.”*
        - Response: Currently, our data for the airports does not include ratings, but we can definitely look into that in the future and add the information to the website.

User Stories Written by Us to Our Developer

- Filter campsites by amenity
    - *“I want to go camping next weekend, but I am new to camping and am not confident in my abilities. It would be easier for me if I had a campsite with running water and electricity. I would like to filter the campsite search to include only those that include both of these two features.”*
        - Response: Thanks for the feedback! We agree that this would be a helpful and practical feature and hope to implement it in a future phase. This type of information is actually included under the amenities section of the campgrounds API so it is definitely a good idea. However, not every campground might have detailed information about water and electricity so this might be a challenge.
- Deploy website on AWS/GCP
    - *“I'm a user who's trying to access your website in order to plan my incoming trip. However, I couldn't find your website anywhere on the web. Could make sure that the website is deployed so that I can access it?”*
        - Response: This is a great idea! While we did not have a website before, at the time of writing this we have deployed and hosted our website at https://www.re-park-able.me/ so that anyone can access it for an upcoming trip!
- Sorting trails by activity
    - *“Depending on my mood, sometimes I'd like to go on biking trails and other times I'd like walking or hiking trails. However, not all trails are equally biking or running friendly. As a user, I'd like some way of differentiating which trails are more popular/suitable for biking, running, walking, or hiking.”*
        - Response: This seems like a helpful feature for users! While for the current phase it is out of scope, we will implement it in Phase III. The APIs have some helpful and interesting information about trails that could be helpful in sorting or filtering them to the above criteria. This might be challenging since all trails might not have the appropriate information, but it will be interesting to implement!
- Sort national parks by state/city/distance
    - *“Sometimes I am traveling to a certain state and want to visit a national park. However, it would not be feasible to travel far from where I am staying to visit the national park. I would like to be able to sort the national parks based on location (or possibly even distance from certain cities) so that I can determine which parks I can visit.”*
        - Response: This is a great idea! We have the latitude and longitude of parks along with addresses, so we could implement a feature that does this. For the current phase that we are in, however, this feature is out of our scope. It will be implemented in Phase III.
- Filter/Sort campsites by weather overview
    - *“I want to visit a campsite soon but while viewing the different campsites I see a lot of campsites with poor weather overview. It would be easier for me to find a*

campsite that has a good weather overview if I could sort them or filter them based on good weather."

- Response: This is definitely a helpful feature! With the current API we are using, we can pull information about a general weather overview for a campsite that users will find helpful. However, this is currently out of scope for Phase I and will be implemented in Phase III.

Phase 2

User Stories Written by Our Customer

- Light/Dark mode
  - *"Hi, I'm a user with sensitive eyes. I want to use your website in Dark mode so as to not hurt my sensitive eyes. I think it'll help others as well with accustoming your webpage to my system settings in terms of light/dark mode."*
    - Response: This is a great suggestion! We will see if we can implement it in the upcoming phases.
- Budget Scale
  - *"I'm planning a trip with my friends over Spring Break, and we're using your site to decide which city to visit. Our collective budget overlaps the Medium and Medium High ranges; we would rather set the low and high bound ourselves. Implement a sliding scale for city budgets to filter on a user-provided numerical input."*
    - Response: This is outside the scope of this phase, but will be considered when we implement sorting/filtering in Phase 3.
- Responsive Design
  - *"Hi, I'm a travel influencer so I'm always on the go and I frequently use my phone. The design of your website is not very friendly for mobile devices; for example, the navigation bar is not fully visible. I would like a responsive design to adapt your site to be compatible with different screen sizes and devices."*
    - Response: Hi, thanks for the suggestion. Our navigation bar now has a drop down menu and the other components scale accordingly for easier access on mobile devices with smaller screens.
- Flights from my location to airport
  - *"Hi, I'm planning a vacation for me and my girlfriend and I'm looking through the airports. I thought it would be a nice feature if I could see how much flights from my nearest airport to the one I'm currently viewing cost, so I don't have to go to another website to find out."*
    - Response: Unfortunately the API we are using for airport data does not include any information on airport cost.
- Git text on home page
  - *"I think your website looks very nice, but I think it could be cleaner and more professional. Currently, there appears to be git text on the home page which makes it appear quite messy. I suggest that this should be fixed for the future."*
    - Response: Hello, could you please specify what the git text is displaying and where? I'm not seeing it on our end. Thanks!

        - Followup: I can't seem to find any git text anymore on your website. Perhaps it was updated within the past five days? Or maybe I could be mistaken. Either way, this doesn't seem to be an issue anymore.

User Stories Written by Us to Our Developer

- Search function
    - *"I am planning a trip to Big Bend National Park and want to see what trails I can visit. I don't want to have to go through the parks list to find this park manually. Can you add a search function that will allow me to search the park by name?"*
        - Response: This is a great suggestion, however, it is currently out of our scope. This is a feature that we will implement in Phase III of development. Having a search function will be extremely helpful to people who are using our website.
- Embedded Google Maps
    - *"As a tourist, I like looking at maps to have a visual idea of where locations are before I look more into visiting them. Although the pages for each park and campground have URLs to the directions, I'd like to have an embedded map on each of those pages - similarly to the pages for each trail. More specifically, embedded Google Maps would be helpful so that I can easily see an overview of the place's reviews, ratings, address, as well as click on embedded links to see specifics (like directions)."*
        - Response: Thank you for the suggestion! We have gone ahead and implemented this feature. Once you find a card of information that interests you, click on the "Learn More" button. You will then be led to a page that contains more information about which site you are interested in. There you should see an embedded map that shows what you are looking for.
- Filter/Sort campsites by the fee amount
    - *"As a tourist, I would love to be able to filter relevant campsites by the fees associated with them in order to best pick which option suits me. Sometimes I might need a cheap option, and it can be useful to sort the campsites either by high to low or low to high in terms of fees."*
        - Response: Thanks for your suggestion! This is a feature we plan to implement since many users would be taking pricing into account when deciding which campsite to stay at. However, this feature is currently out of our scope and will be implemented during Phase III.
- National Parks
    - *"Hey, I'm a user who was trying to use your website for planning a future trip. However, your website doesn't include all the national parks in the US. Could you please add the rest of the national parks?"*
        - Response: Thanks for your suggestion! We have gone ahead and added this feature to our website. Once you click on the parks page you should be able to see all of the National Parks. Each card contains basic information about the park that should help you find a location that suits

your interests. From there, you can click on the card to get more details about the national park.

- More Trails
    - *“I like hiking on different trails. However, your website only has a small amount of trails in a specific amount of places. Would you be able to add more trails to the website?”*
        - Response: Thanks for the suggestion! We have taken your request into consideration and went ahead and added many more trails to the website. In the trails tabs, you should see a plethora of cards that lead to different trails all around the country. They also have some relevant information that should help you quickly pick which trails might interest you. Additionally, you can click on a card to learn more about that specific trail.

Phase 3

User Stories Written by Our Customer

- Social platform presence
    - *“Hello, I really want to share your website with my peers on social platforms, such as Facebook, Twitter, Discord, etc., but it doesn't display very nicely. I think utilizing the Open Graph protocol would be a great addition that would help your site look more pleasing when sharing. I'll include a screenshot of how your site currently appears on Discord, and I hope you can consider the idea.”*
        - Response: Thank you for this suggestion! We also think it'd be more user-friendly for us to implement, but as it is not in our current scope, we will consider adding it for the next phase.
- Attributes in url
    - *“Hello, I like to ask my friends for their opinions on my selections on your site. Typically, this involves me sharing the url with my friends, telling them what attributes I filtered by, and them reapplying those attributes on their own. I was wondering if it's possible if the url I share with my friends could include the filters/settings I've selected on my page so that way I don't have to tell them what I selected, and they don't have to manually apply those settings.”*
        - Response: We have implemented sorting, searching and filtering by attributes for parks, cities and airports. You can now share the URL with your friends.
        - Followup: Perhaps there was a misunderstanding. I meant putting attributes in the url for the frontend, not the API. I'm not trying to use the API directly. I just want to use your website. For example, when I go to https://www.parkscape.me/parks and apply filters/sorts, I'd like to see those in the url, so I can share that with my friend without having to explicitly tell them what I selected. For example, if I go to https://www.parkscape.me/parks?phone=yes, the phone filter should be applied with 'yes'. Based on the commits so far, this doesn't seem to be implemented.

        - Followup Response: Due to how React renders the components, the page url does not change when filters are applied, but we can look into this for the next phase (phase 4).
- Add favicon
    - *“Hello, your website looks very clean, but the current favicon doesn't seem to be very closely related to your website. It seems that it's the generic React favicon. I think adding a favicon would help make your site standout from others.”*
        - Response: Thank you for the suggestion! We have now updated both our favicon, as per your suggestion, as well as a title and description to match. [Time Tracking] Estimated time: 30 minutes; Actual time: 10 minutes
- Narrowing down results
    - *“Hello, when I visit a page, such as parks, I would like to narrow down the results to fit some criteria. I noticed you have attributes I could filter by, but when I select some options and click Apply, the results don't change. Will this feature be implemented?”*
        - Response: We have implemented searching, sorting and filtering by many attributes for all cities, parks and airports. You can now narrow down results as you are using our website to plan your next trip.
- Loading content
    - *“Hello, I'm using your site and at times, the content does not load. I've inspected the page, and it seems that there are some errors that occur involving your API. I'll include a screenshot as an example, but I hope there are plans to fix this.”*
        - Response: Hi, we have updated our API calls, so this issue should be fixed now!

User Stories Written by Us to Our Developer

- Sort/filter hiking trails
    - *“Hey, I'm trying to an easy hiking trail that is close to me. However, your website doesn't allow me to filter out all the trails which aren't in my state, nor does it allow me to sort them by duration. Could you please implement this functionality?”*
        - Response: N/A
- Sort parks by entry fee
    - *“I want to go on a road trip next week, but I don't want to spend a lot of money. It would be helpful to be able to sort parks by increasing entry fee. This way I can find free and cheap parks more easily.”*
        - Response: N/A
- Filter campgrounds by internet access
    - *“I love camping and love the list of campgrounds available. However, sometimes when I go camping, I want access to the internet. I would like if the list of campgrounds was filtered by whether there was internet access.”*
        - Response: N/A
- (Sitewide) search bar

        - *“As a user, I'd like to keyword search for any given city, park, or airport I've heard of straight from your homepage. It is a little cumbersome to go to each respective model page to search for a specific place, as sometimes I'm not sure what exactly I'm searching for. If I could just quickly search from anywhere on your website (maybe a search bar on the navigation), it'd be very convenient for such situations.”*
            - Response: N/A
- Clickable Cards
    - *“I noticed while surfing through your website that whenever I went to a model page, the cards had learn more buttons. I think the learn more buttons are great but as a user, I am used to being able to click on the entire card and have it navigate me to the correct instance page. I have seen this pattern in many other websites so as a User it would be great if the entire card was clickable and not just the learn more button.”*
        - Response: N/A

RESTful API

API Documentation
Endpoints:

- Single Park
    - GET api.parkscape.me/park/:parkID
    - Returns information about a park
        - website - the URL of the park website
        - activities - an array of activities available in the park
        - fee - the cost of entry for the park
        - description - a short description of the park
        - photos - the URL of photos of the park
        - nearest_airports - an array of three airports closest to the park
        - topics - an array of topics commonly associated with the park
        - weekdays - the hours of the park for each day of the week
        - longitude - longitude coordinate of the park
        - id - the ID for accessing the park with the API
        - states - the US state(s) in which the park is located
        - phone - contact phone number of the park
        - latitude - latitude coordinate of the park
        - nearest_cities - an array of three cities located closest to the park
        - email - contact email for the park
        - name - full name of the park
- Single City
    - GET api.parkscape.me/city/:cityID
    - Returns information about a city
        - rating - the user rating of the city
        - airbnb_listings - URL to the AirBnB listings in the city

        - short_name - name of the city
        - photo - URL to a photo of the city
        - walkability - URL to the city walkability score
        - nearest_parks - an array of three parks located closest to the city
        - longitude - longitude coordinate of the city
        - long_name - name of the city, name of the state, name of the country
        - cost - cost score of the city
        - safety - safety score of the city
        - hiking_trails - URL to Alltrails hiking trail search in the city
        - nearest_airports - an array of three airports closest to the city
        - latitude - latitude coordinate of the city
        - population - population of the city
        - id - the ID for accessing the city with the API
        - name - name of the city, abbreviation of the state
- Single Airport
    - GET api.parkscape.me/airport/:airportID
    - Returns information about about an airport
        - website - the URL of the airport website
        - city - city in which the airport is located
        - name - full name of the airport
        - icao_code - ICAO airport code
        - nearest_parks - an array of three parks closest to the airport
        - longitude - longitude coordinate of the airport
        - iata_code - IATA airport code
        - phone - contact phone number of the airport
        - state - the US state in which the airport is located
        - address - airport address
        - latitude - latitude coordinate of the airport
        - nearest_cities - an array of three cities located closest to the airport
        - id - the ID for accessing the airport with the API
        - zip_code - zip code of the airport
- List of Parks
    - GET api.parkscape.me/parks
    - Params
        - page - Return given page of parks. Each page contains 12 results.
        - sort - Sort list of parks by attribute. Values ending in "_asc" indicate ascending order and "_desc" indicates descending order. Supported attributes: name, fee
        - states - Filter parks by state.
        - activities - Filter parks by activities.
        - topics - Filter parks by topics.
    - Ex. api.parkscape.me/parks?page=1&sort=name_asc&states=TX&activities=Astronomy&topics=Animals

- List of Cities
    - GET api.parkscape.me/cities
    - Params
        - page - Return given page of cities. Each page contains 12 results.
        - sort - Sort list of cities by attribute. Values ending in "_asc" indicate ascending order and "_desc" indicates descending order. Supported attributes: name, population
        - state - Filter cities by state.
        - cost - Filter cities by max cost budget value on a scale from 1-10, 10 being most expensive.
        - safety - Filter cities by min safety score on a scale from 1-5, 5 being most safe.
    - Ex. api.parkscape.me/cities?page=1&sort=name_asc&state=TX&cost=7&safety=3
- List of Airports
    - GET api.parkscape.me/airports
    - Params
        - page - Return given page of airports. Each page contains 12 results.
        - sort - Sort list of airports by attribute. Values ending in "_asc" indicate ascending order and "_desc" indicates descending order. Supported attributes: name, iata_code
        - state - Filter airports by state.
        - website - Filter airports by whether they have a website.
        - phone - Filter airports by whether they have a phone.
    - Ex. api.parkscape.me/airports?page=1&sort=name_asc&state=Texas&website=yes&phone=yes

Single objects can be obtained by their unique IDs. Lists can be filtered using parameters.

Models

- Cities
    - One of our models revolves around cities. Users will be able to browse and compare cities by many attributes. Some of these attributes include population, average rating, travel cost, safety score, and COVID score. Users will also be able to easily access AirBnB listings, city walkability reports, and browse walking trails through provided external links. Users will also be presented with information regarding the nearest airports and parks. To retrieve information about Cities we are using the RoadGoat Cities API, which Synthesizes information to provide the richest and most accurate context about over 4.3+ million destinations.
- Airports

    - Another one of our models is airports. Users will be able to browse and compare airports. Each airport instance will contain information regarding the name, location, phone number, and ICAO code. Users will be able to see nearby cities and parks to plan out their travel if they know which airport they are going to. To retrieve information about Airports we are using the Airport Info API, which is an extensive database of airport codes and data, including address, phone number, website and more.
- Parks
    - Our last model is parks. Users will be able to browse and compare parks according to their opening hours, activities, topics, fees, and location. They will also be presented with contact information such as phone number and email and will be able easily access nearby cities and airports to plan their travel. To retrieve information about Parks we are using the National Park Service
    - API, which provides authoritative National Park Service (NPS) data and content about parks and their facilities, events, news, alerts, and more.

Tools

- GitLab
    - We used GitLab for our git repository and CI/CD pipeline, making sure to utilize GitLab's issues feature to keep track of our tasks and user stories. Our edits were pushed to a 'dev' branch before being merged to the 'main' branch of the repository.
    - We also used GitLab's API for the dynamically fetched repository data on the about page
- Amazon Web Services
    - We used:
        - Amplify to host our website's frontend
        - Lightsail to host our website's backend
        - RDS to host our database
- Visual Studio Code
    - We used VSCode as our main IDE for creating and modifying our files (html, json, js, makefile).
- React
    - We used React to build our frontend web pages/GUI components.
- Bootstrap
    - We used Bootstrap's css framework in our React js files to create a sleek and cohesive look for our website.
- MySQL
    - We used MySQL to store our instance data into a database for faster rendering on our website.
- SQLAlchemy
    - We used SQLAlchemy to interact with our AWS RDS from the backend server.
- Flask

    - We used Flask to provide endpoints for our API.
- Docker
    - We used Docker to create both our frontend and backend containers.
- Postman
    - We used Postman to document an API of our calls to each of our models' APIs which we scraped our data from (see RESTful API section).

Hosting

The website is hosted on AWS Amplify. There's essentially no build scheme for now, since the website is static. We have also created a hosted zone on AWS Route 53 in order to redirect users from the domain (www.parkscape.me) to our Amplify application. The name servers from AWS have also been added to NameCheap, the hosting service that we obtained our domain from. Lastly, we used AWS IAM in order to ensure the highest standard of security for our website.

Database

The data was initially scraped using a Python script in scrapper.py. The script executed the needed HTTP requests to the APIs that we were using to obtain the data. After that, the data was stored in JSON format. The next step was to convert the JSON data into Python objects, which was done in db.py. In the same file, we used the SQLAlchemy framework in Python in order to connect to our database and store the Python objects as rows. The database was implemented using MySQL, and had 3 tables (cities, parks and airports). The database was hosted on AWS, using the Relational Database (RDS) service. In app.py, our backend server makes the required query to the database depending on the invoked endpoint. We provide endpoints for all instances of a model, one specific instance (by id) and all instances for a page in our frontend server.

Pagination

Our pagination was implemented by a custom React component created that makes use of our API endpoints. An API request to any of the models (for example: GET api.parkscape.me/parks) returns the data associated with 12 instances for that particular model. At any particular time only 12 instances are displayed. When the back or next button is clicked, we make a request to our API for the next 12 instances using the "current page" as an id for which batch of 12 new instances we want to display. Once we receive the new 12 instances we are able to render new model cards based on the updated data.

Sorting

We implemented sorting on the backend by adding a "sort" argument to the API calls. The parameter would follow the form attribute_asc/dsc in order to specify which attribute to sort the

data by (population, name, state…) and whether to sort it in ascending or descending order. This argument would then be passed on to our SQLAlchemy query.

Based on user input on what type of sorting they wanted (as indicated on the sort option in the front end), we use axios to call the api to get a new response reflecting the assortment of model objects in their requested order. Using the response from the endpoint provided by the backend, we update the data in the frontend and rerender to show the user.

Filtering

We implemented filtering by having a filters list in our schema for each model. Inside app.py, we search through the filter list and check whether each filter is passed as an argument to the API call. If it is, we add a predicate to the SQLAlchemy query, equivalent to the filter. When it came to activities, topics and states in the park model, the process was a bit more complex. A simple equals predicate wasn't enough, since the column entries were JSON objects.

Based on the user input on the different types of filters, we use axios and call the api endpoint with the corresponding optional filter populated in the url. From this, we get the new set of data that corresponds to the chosen filters by the user. We then use this response and update the data shown on the frontend. This includes not only the instance cards and the total "instances" shown, but also the pagination widget and how many "pages" can be clicked through.

Searching

We implemented searching by adding a "search" argument to the API call. We would then parse the search argument and add predicates to our SQLAlchemy query. We add a predicate that checks whether a specific column entry for a row contains the search parameter, and we do this for all columns that may contain the given search parameter.

We added useEffects to re-render the cards shown based on the search input whenever something is searched. Using the user's input we use the api endpoint provided to us by the backend and get a new response with the new data and rerender the frontend. We also update the instance cards. the total "instances" shown, and the pagination widget. We also added a parameter to our existing model components to facilitate site wide search, to be able to easily showcase all three models on a universal search.

How it Works

ParkScape compiles information about United States airports, nearby cities, and their local state/national parks by scraping data from existing APIs for each of these models.

Challenges

Phase One:
Perhaps the biggest challenge we encountered was navigating and learning the tools as we worked, since some of us were rather unfamiliar with web development. For the “About” page specifically, we hadn’t used APIs, Postman, or Bootstrap, and had little experience with Javascript prior, so figuring out where to start was a challenge. To overcome this struggle, we broke down the larger goals into smaller subtasks after lots of research about each tool – starting with the simpler Bootstrap and HTML design before moving onto using Postman and calling the GitLab API in Javascript. A smaller challenge we faced was finding free APIs with sufficient data that would fit our needs.

Phase Two:
- Our API has issues where it tends to go down occasionally, which breaks our model and instance pages. This also causes our Selenium and Postman tests to fail. It seems to be a problem on NameCheap's end. (We have discussed this with our TA).
- Implementing pagination and making calls with axios proved to be difficult as we had limited experience with using axios to not only receive information from our API but to also use that information and update the React pages dynamically.
- Getting our jest test suites to work in our environment was difficult as we kept encountering the system limit for file watchers error.
- Figuring out how to use React was initially difficult. We had to spend some time figuring out how to convert the HTML files to Javascript to incorporate the multiple instances of each file.

Phase Three:
- Our API has issues where it tends to go down occasionally, which breaks our model and instance pages. This also causes our Selenium and Postman tests to fail. It seems to be a problem on NameCheap's end. (We have discussed this with our TA).
- Writing Jest tests on components that use axios proved to be a challenge because it seems that Jest does not interface well with it. It seems that we must somehow “mock” axios in order for jest to be able to correctly test components using Jest.
- Implementing site wide search proved to be a challenge because at first we did not reuse our already existing components for Cities, Parks, and Airports.
- With our previous backend code structure, we would’ve had to write a separate method for each filter/sort, since each one would need its own SQL query. This would have required a very large amount of code to implement these features. We had to refactor our backend code to avoid this.