Travelwise Technical Report
11/4/2020

Motivation

Travelwise is a web app that enables consumers to make ideal travel plans. Through the use of various RESTful APIs, Travelwise is an ideal platform for users to compare destinations by safety, affordability, and more. A feature of Travelwise is providing users with up-to-date statistics on COVID in a given region. Travelwise streamlines the process of making travel plans, where users can search for destinations, corresponding flights, and hotels all in the same place.

Customer Stories Phase 4

11/23/2020

Customer Story 1:

- Story:
  - As a customer, I would love it if clicking the title on the navbar led to the home page. I would also love it if there was a logo to go with your website's name. Adding a logo may make your brand more recognizable and help customers like me associate better with your brand.
- Response:
  - Done, we've added our logo from Phase 1 back in.

Customer Story 2:

- Story:
  - As a customer viewing the about page I think it would be nice if some more visuals were added such as icons for each link. As of now, each API has a red circle with the first letter of the API, and I think this is a bit repetitive. With icons, maybe even one for each technology, the about page would look much more visually appealing by adding something to look at other than text.
- Response:
  - This was extremely tedious to do, but it's done, and I think it was worth it! The page looks much nicer now.

Customer Story 3:

- Story:
  - As a customer of Plant Planet I would like to feel like I'm using a completed website. The use of the word "model" on each categorical page makes me feel like the website still has yet to be completed. It would be cleaner if some other header than "Plant Model," "Location Model," and "Animal Model," were to be used.
- Response:
  - Good call, we've removed the word "Model" from the model pages.

Customer Story 4:

- Story:
  - As a user of theplantpla.net, I would like to have a more informative landing page. It is a nice landing page, but I feel having a brief description of the site's purpose would make the experience better.
- Response:
  - We added a little snippet underneath the big text on the splash page.

Customer Story 5:

- Story:
  - As a customer, right now it seems to me that the website is lacking a bit of personality. It would be great if the website had a more distinct and unique theme, for right now it seems a little generic to me; a good theme can really make the website memorable and stand out while also complement the website as a whole.
- Response:
  - This is a great point and definitely something we could improve further upon. However, for the time being, we've made an effort to make more of our graphics fit a more pastel color theme, and our visualizations (especially our bubble chart) to align with this as well.

Customer Stories Phase 3

11/4/2020

Customer Story 1:

- Story:
  - As a customer I would like to easily filter information by specific criteria. As it stands, filtering works, but when I try to switch from one filter to another, it instead combines multiple filters at a time. This is unintuitive as most filtering does not work this way, and when it does it's done with more visual clarity.
- Response:
  - We added the capability to deselect filter values! Now you can select a filter, and click on the menu item again to remove it, so you can have only one filter apply at a time.

Customer Story 2:

- Story:
  - As a customer, I would like to be able to search for items. I can only find plants by going through all the pages and manually finding it. I would like to see a search field that will return corresponding items.
- Response:
  - We've added this! We now have the ability to search on the model pages and the search page.

Customer Story 3:

- Story:

    - As a customer, if I was researching things about plants for a project and using this website, I'd like to be able to recognize the website's tab amongst my other open tabs. As of now, the icon is a default google chrome gray globe with the title "React App". It would be much easier to find this website's tab and also look more well-done if the website had a custom tab icon and tab title.
- Response:
    - We added a favicon! Our build files had to be modified to make it show up

Customer Story 4:

- Story:
    - As a customer, I would love to see the interface of the website become both more organized and visually appealing. Right now, some of the information or data appear scattered across the pages, and sometimes images such as the state flags appear distorted, so it would be really nice if the visuals of the website could be cleaned up and refined.
- Response:
    - We organized our information so everything is now on the instances pages, and the model pages only display limited information.

Customer Story 5:

- Story:
    - I would love to see more pictures on the animal and plant models. As a customer I like seeing pretty pictures of animals and plants and would love to see pictures of most of the plants/animals on your website. If possible, I would love the pictures to be somewhat representative of the actual animal/plant.
- Response:
    - We've refined our pictures more-- because our images came from Google Images, we had to go in and double check to make sure the images were appropriate, which we've done, to an extent.

Customer Stories Phase 2

10/20/2020

Customer Story 1:

- Story:
    - As a customer of Plant Planet, I would like to see one or more "featured" locations on the landing page. It's apparent in the header tabs that locations are a part of your site, but since there aren't any on the landing page, I might be inclined to think that it isn't worth it to check out locations. Inclusion of this feature would make me as a customer feel like your service is more fleshed out, placing a similar amount of importance on all 3 models, rather than just 2.
- Response:
    - This user story was provided using an old version of our website, so our design has changed. However, the point about it not being clear that locations are

important is valid. We will make sure to highlight the location model on our splash page.

Customer Story 2:

- Story:
    - As a customer, I would like to be able to recognize invasive plants and know how to play my part in preventing their spread. This would make the mission of revitalizing and maintaining the environment much more accessible. I'd like to go to my region or area and be given a list of invasive species, so I can start combating them.
    -
- Response:
    - Although including invasive species is a bit out of the scope of our website, we did our best to make sure that native plants were clear and the location pages do not accidentally promote showing invasive species.

Customer Story 3:

- Story:
    - I would love to see more instances of each model. As a customer I have interests in a variety of plants and specifically I want to see more American native plant instances. If possible, I would like to see which plants belong to which state also.
    -
- Response:
    - We are working on bringing more instances of each model in this cycle, so this time there should be more than 3 instances per model on our website.

Customer Story 4:

- Story:
    - As a customer of PlantPlanet, I would like to see icons that match their description. As it is now, the icons don't match their descriptions, for example, Postman has a potted plant as its icon! I feel like updating the icons would give a very pleasant aesthetic and a more enjoyable user experience.
    -
- Response:
    - Unfortunately, we are getting rid of the icons. We have started using more tools than we have anticipated, and ran out of icons. By having no icons, we are attempting a neater approach to listing all the tools we have used so far.

Customer Story 5:

- Story:
    - As a customer of PlantPlanet, I would like to be able to have the ability to view information for more specific regions, such as for individual states or even cities. Average information for an entire country will often vary drastically from local

    - regions across the country, so it would be nice if the locations are less generalized. It would also be nice to have the ability to easily access additional information such as what the neighboring regions are, in order to understand what the surrounding areas are like as well.
    -
- Response:
    - We found an API that has geographical information at the state level, whereas we had it at the country level before. Our website reflects this now.

Customer Stories Phase 1

9/29/2020

Customer Story 1:

- Story:
    - As a customer, I would love to be able to search up my current location and see facts about it. Specifically I would love to be able to search up an area and get data on the native plants in it. If possible I would also like to see data on which animals are in the area.
- Response:
    - Didn’t respond

Customer Story 2:

- Story:
    - As a property manager, given a location, I want to know what plants and animals are native to that location. In other words, I want to know what lives in a specific place. This would allow me to grow non-invasive plants, as well as determine which animals I need to keep in mind when making changes to my property.
- Response:
    - No response

Customer Story 3:

- Story:
    - As a customer of Plant Planet, with underlying medical conditions, it would be greatly helpful to include any potential medical dangers from these plants. Many people can have mild to severe allergic reactions to these plants which they might not be aware of if they attempt to handle these plants, so it would be very helpful to include any potential dangers that might be associated with coming in contact with these plants.
- Response:
    - none

Customer Story 4:

- Story:

  - As a customer, I'd like to know how to help my local ecosystem such as being able to search for what native plants around my location need more support. What can I do to help? What should I plant? How do I plant them? This would make it much easier to get started on actively improving my local ecosystem!
- Response:
  - none

Customer Story 5:

- Story:
  - As a customer of Plant Planet, given a particular plant, I would want to know the wildlife that depends on it. Basically, I would like to be able to search for a plant and then from that plant page be able to get associated wildlife pages. This type of flow would make it very easy to connect plants to the wildlife that depends on them!
- Response:
  - none

User Stories Phase 4

11/23/2020

User Story 1:

- “On the about page of travelwise's website the unit test counts for each team member and the total unit test count is 0. As a customer of travelwise, this makes it seem like there is no testing for this website. I would like these numbers to be updated with a more accurate amount.”
- Approach:
  - Simple fix since we were using placeholder numbers.
- Estimated Time:
  - 15 minutes
- Implementation:
  - Implemented
- Actual implementation time:
  - 15 minutes

User Story 2:

- “Model page sort/filter/search options. While it doesn't look bad, I'd like to see the search/sort filter options arranged a bit better.”
- Approach:
  - We agreed and made the filter/sort options look a lot neater. We did this by playing around with bootstrap’s many classes.
- Estimated Time:
  - 30 minutes
- Implementation:
  - Implemented

- Actual implementation time:
    - 20 minutes

User Story 3:

- “As a user of TravelWise, I would like to have more info on the "Provider Visuals" page. I think the visuals look clean but I'm not sure what I'm supposed to gain from animals by their orders or plants by their families. Maybe provide some additional text or a question mark button with an explanation.”
- Approach:
    - We fixed this by adding a caption that links to our provider website where the user may find more information.
- Estimated Time:
    - 10 minutes
- Implementation:
    - Implemented
- Actual implementation time:
    - 15 minutes

User Story 4:

- “Some of the pictures on the city pages are really big and it makes it difficult to see all the information at one time. It would be nice if the pictures were smaller so that the information on the page is more digestible. An example would be Tijuana.”
- Approach:
    - Figured out weird css styling cause sometimes bootstrap does weird things, fixed it by specifying CSS directly instead of using bootstraps classes.
- Estimated Time:
    - 2 hours
- Implementation:
    - Implemented
- Actual implementation time:
    - 3 hours

User Story 5:

- “As a customer of TravelWise, I find it strange that I am given the option to sort Cities by 'null.' Some people may get very confused when they see that. Even if it does not confuse people, it looks out of place. I would like it to be removed or renamed.”
- Approach:
    - We removed the error in our data for sorts, and removed the issue entirely.
- Estimated Time:
    - 10 minutes
- Implementation:
    - Implemented
- Actual implementation time:
    - 10 minutes

User Stories Phase 3

11/4/2020

User Story 1:

- *“As a customer of Travelwise, I would like to have links to the technologies listed on your about page. This could take the form of either hyperlinking the technology websites to their names or providing a clickable company icon for each technology. This is important so I can easily investigate more into the technology used to create your website.”*
- Approach:
    - We have hyperlinked the names of each technology to their main website or a location to learn the most about them.
- Estimated Time:
    - 10 mins
- Implementation:
    - Implemented
- Actual implementation time:
    - 10 mins

User Story 2:

- *“As a first-time user of TravelWise, I would like to have a title, or something similar, at the top of each page letting me know which page I am looking at. When navigating to the cities page from the home page, I was not immediately aware I was looking at the cities page. Adding a title or some other form of indication will improve the clarity of the website. Something similar to what the About page already has would be sufficient.”*

- Approach:
    - Titles have been added to each page to make them more similar to the About page.
- Estimated Time:
    - 20 minutes
- Implementation:
    - Implemented
- Actual implementation time:
    - 20 minutes

User Story 3:

- *“As a user of TravelWise, it would be nice to have an option to choose how many rows to display per model page. This allows for a better user experience so that users aren't overwhelmed with results. Furthermore, if a user was on a phone with a smaller screen, they could choose to see less results per page.”*

- Approach:
    - We added a dropdown menu to select from 3, 9, 18, 36, and 72 items per page
- Estimated Time:
    - 1 hours
- Implementation:
    - Implemented

- Actual implementation time:
    - 1 hour

User Story 4:

- *"I am glad to see that your website already includes sorting on the model page. However, I think the look of it should should change by the final phase. Instead of having a stack of buttons for sorting, typically sorting tables are done by clicking the column title. If you could implement that, the browsing experience would be more streamlined."*
- Approach:
    - We added a dropdown menu instead which is just about as streamlined as clicking on the columns
- Estimated Time:
    - 10 hours
- Implementation:
    - Implemented
- Actual implementation time:
    - 2 days

User Story 5:

- *"Rather than see all the sort options in a list type view, it would be nice to have them in some kind of dropdown. This allows for a better user experience. Additionally, the user isn't overwhelmed with results when they don't want to see them."*
- Approach:
    - We implemented both sorting and filtering in dropdown menus, as well as options to sort either through ascending or descending order.
- Estimated Time:
    - 10 hours
- Implementation:
    - Implemented
- Actual implementation time:
    - 2 days

User Stories Phase 2

10/20/2020

User Story 1:

- *"As a user of Travelwise, I would like to see a footer on the website. It would make the website look a little more visually appealing. It would also make it slightly easier to navigate."*
- Approach:
    - We have added a footer, but we decided it would be repetitive to have the same functions as the header.
- Estimated Time:
    - 30 mins
- Implementation:

    - Implemented
- Actual implementation time:
    - 30 mins

User Story 2:

- *“As a customer of Travelwise, I would like to see a more visually appealing landing/home page. Since this is the first page users will see when they go to Travelwise, I think it’s important to make it professional to establish a level of credibility. I think it should include some sort of media whether it’s an image or slideshow.”*

- Approach:
    - Added pictures to the landing page with a bright turquoise and purple theme.
- Estimated Time:
    - 30 minutes- a few days for the entire website redesign
- Implementation:
    - Implemented
- Actual implementation time:
    - 30 minutes

User Story 3:

- *“As a user of TravelWise, when I am looking at flights, I would like to know the city that corresponds with the airport that is displayed. At the moment, the flights display the abbreviations of the airports. When flights arrive or depart from more obscure locations, it will be difficult for me to tell what city corresponds to what airport.”*

- Approach:
    - Our current models display the city correlated with the airport instance, and each city/airport model is searchable to find the other.
- Estimated Time:
    - 2 days
- Implementation:
    - Implemented
- Actual implementation time:
    - 1 day

User Story 4:

- *“As a customer of TravelWise I would like to know more information about the different country scores. Currently, the website lists different numbers for different categories for each country. However, I don't really know what those numbers mean in the big picture. I can't tell what is a relatively good score and what is a relatively bad score. If you could add some clarification then the information would be more digestible.*
- Approach:

    - Scores are threat levels out of 100 against certain populations such as women or rating of dangers (high medical score means bad medical care, high theft score means a high amount of robbing). Ratings are done by several grading organizations (is what the API creator says). We plan to create a visualization that explains everything in the last phase.
- Estimated Time:
    - 1 hr, once figure out how we want to lay everything out, we can add a description area/pop-up
- Implementation:
    - Not implemented yet
- Actual implementation time:
    - NA

User Story 5:

- *"I would like to see a bit more on the instance pages for cities. Add some more media and information other than what is presented on the model page."*

- Approach:
    - We added unsplash for high quality pictures for each model and a map for each country and city.
- Estimated Time:
    - 2 weeks
- Implementation:
    - Implemented
- Actual implementation time:
    - 2 weeks

User Stories Phase 1

9/29/2020

User Story 1:

- *"As a customer of Travel Wise I would want to know the number of COVID cases in the country of my destination. This is because my travel plans are now affected by the number of people infected in my destination. The effects of a greater infection rate could mean more restrictions on places I'm allowed to go as well as a higher risk of catching the disease and then having to spend 2 more unplanned weeks at my destination."*

- Approach:
    - We have destination city instance pages link to associated COVID stats instance pages of the country/region.
- Estimated Time:
    - 3 days
- Implementation:
    - Each city now has a link to it's COVID stats showing things such as number of new cases.
- Actual implementation time:

    - 3 days

User Story 2:

- *“As a user of Travelwise who may have a flight with layovers, I would like to know how much time there is in between connections. Information about the layovers for a flight is helpful because I may want to avoid a flight if it has layovers that are too short or too long. This information will help me weigh the pros and cons of the different flights listed on TravelWise and will improve my browsing experience.”*

- Approach:
    - Travelwise has been revised to be an accessory service to provide general safety information to be used in conjunction with other services such as Trivago.
- Estimated Time:
    - NA
- Implementation:
    - NA
- Actual implementation time:
    - NA

User Story 3:

- *“As someone who has had to travel on a tight budget before, I think in addition to having hotel and flight prices sortable, it would be nice to filter them by range. Other services like Google Flights have this feature and it would make sense to include this functionality on your website. Overall, this would make browsing easier.”*

- Approach:
    - Travelwise has been revised to be an accessory service to provide general safety information to be used in conjunction with other services such as Trivago.
- Estimated Time:
    - NA
- Implementation:
    - NA
- Actual implementation time:
    - NA

User Story 4:

- *“Since safety seems to be important to the application, on top of covid stats at the destination, if possible, I would like to see the covid procedures implemented by the flight company that I end up choosing. Companies that have very loose procedures may not be seeing high customer traffic right now. By providing the user this information, it covers one more area of concern regarding COVID.”*

- Approach:

    - Travelwise has been revised to be an accessory service to provide general safety information to be used in conjunction with other services such as Trivago.
- Estimated Time:
    - NA
- Implementation:
    - There is not really a feasible way to find the COVID procedures of every flight corporation. Also the models have been redesigned to only airports, covid, and cities, but the safety scores for each city will still be there to provide secure travels.
- Actual implementation time:
    - NA

User Story 5:

- *“As a customer of Travelwise, I am unsure of the functionality of the dropdown under the search bar on each model page. I believe the intention is for the selected field to be the column each table is sorted by, although this is a guess. It would be beneficial to users of this site if the purpose of the dropdown was labeled, or if the default text in the dropdown indicated its purpose, such as "Order columns by:"*

- Approach:
    - We’ve default text to be more descriptive in the dropdown menus.
- Estimated Time:
    - 1 day
- Implementation:
    - We created dropdown menus for each category
- Actual implementation time:
    - 1 day

RESTful API

For destination information (cities), we use GeoDB cities API:

- https://rapidapi.com/wirefreethought/api/geodb-cities?endpoint=5990a0b4e4b075a0d1d6da26

For destination safety scores, we use Amadeus's Safe Place API:

- https://developers.amadeus.com/self-service/category/destination-content/api-doc/safe-place-api/api-reference

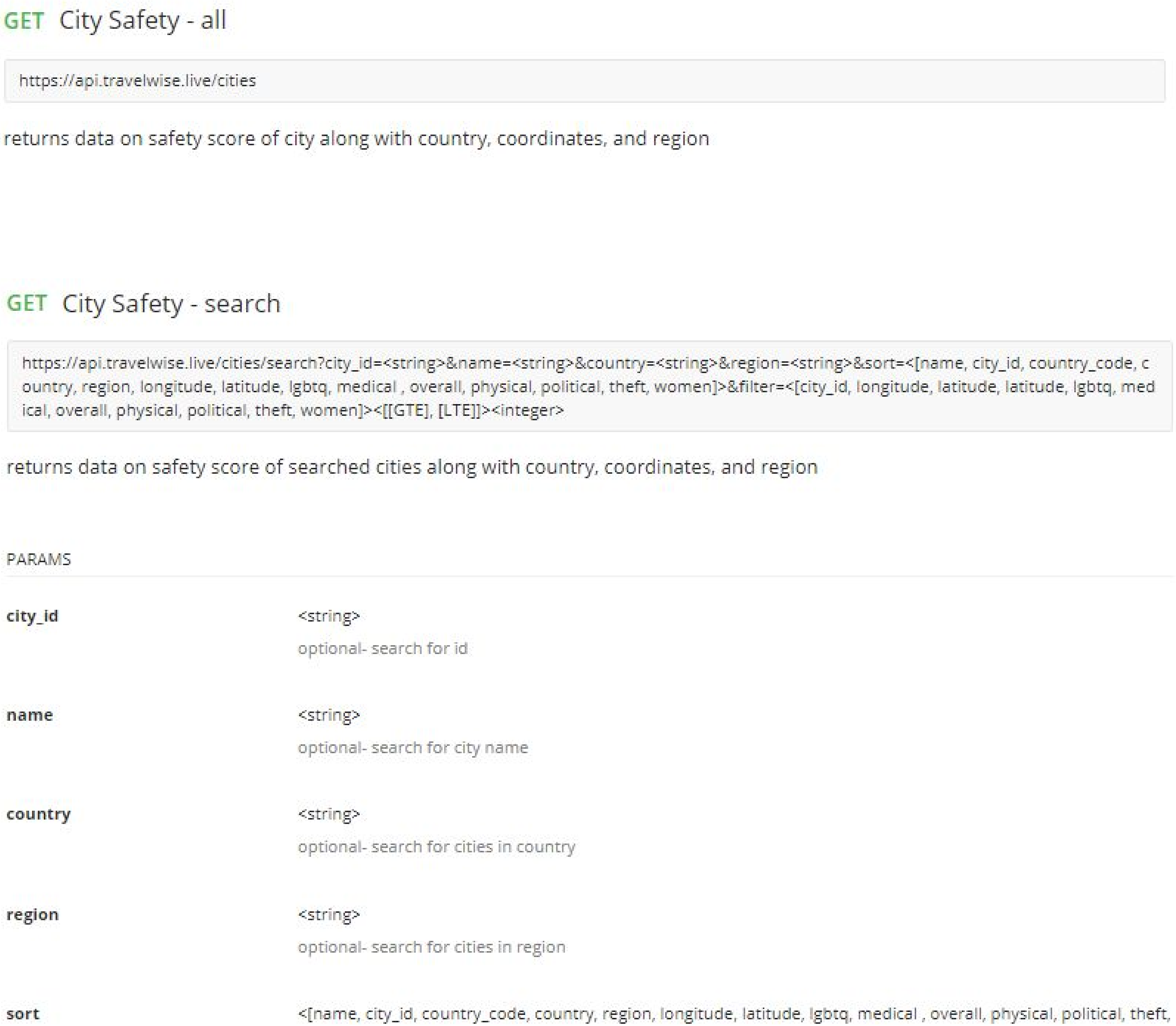
For COVID statistics on given regions, we use COVID19 API:

- https://documenter.getpostman.com/view/10808728/SzS8rjbc

For airports, we use Amadeus's Flight Searcher API:

- https://developers.amadeus.com/self-service/category/air/api-doc/flight-offers-search/api-reference

Travelwise API



GET City Safety - all

```
https://api.travelwise.live/cities
```

returns data on safety score of city along with country, coordinates, and region

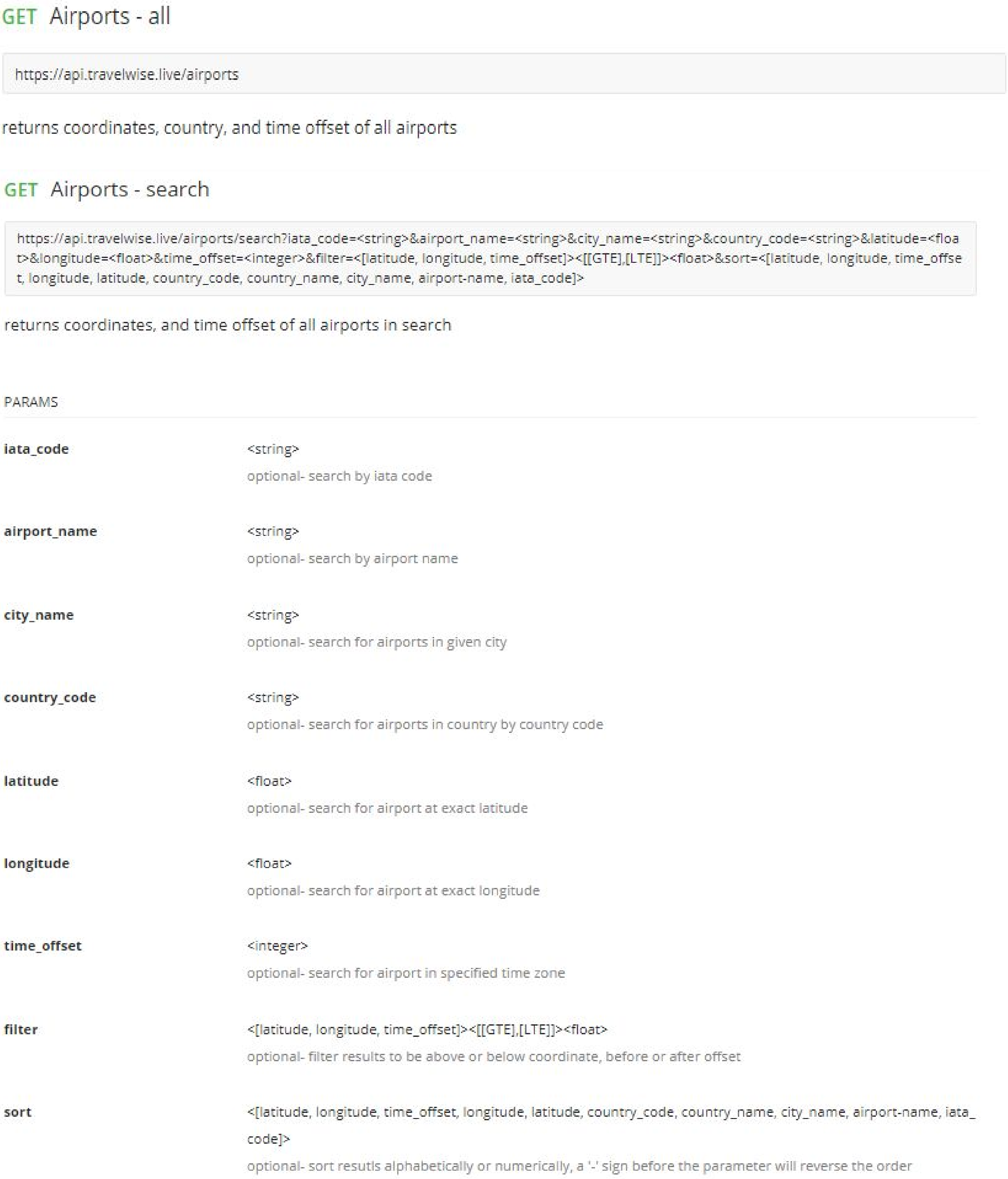
GET City Safety - search

```
https://api.travelwise.live/cities/search?city_id=<string>&name=<string>&country=<string>&region=<string>&sort=<[name, city_id, country_code, country, region, longitude, latitude, lgbtq, medical , overall, physical, political, theft, women]>&filter=<[city_id, longitude, latitude, latitude, lgbtq, medical, overall, physical, political, theft, women]><[[GTE], [LTE]]><integer>
```

returns data on safety score of searched cities along with country, coordinates, and region

PARAMS

city_id	<string> optional- search for id
name	<string> optional- search for city name
country	<string> optional- search for cities in country
region	<string> optional- search for cities in region
sort	<[name, city_id, country_code, country, region, longitude, latitude, lgbtq, medical , overall, physical, political, theft,



GET Airports - all

```
https://api.travelwise.live/airports
```

returns coordinates, country, and time offset of all airports

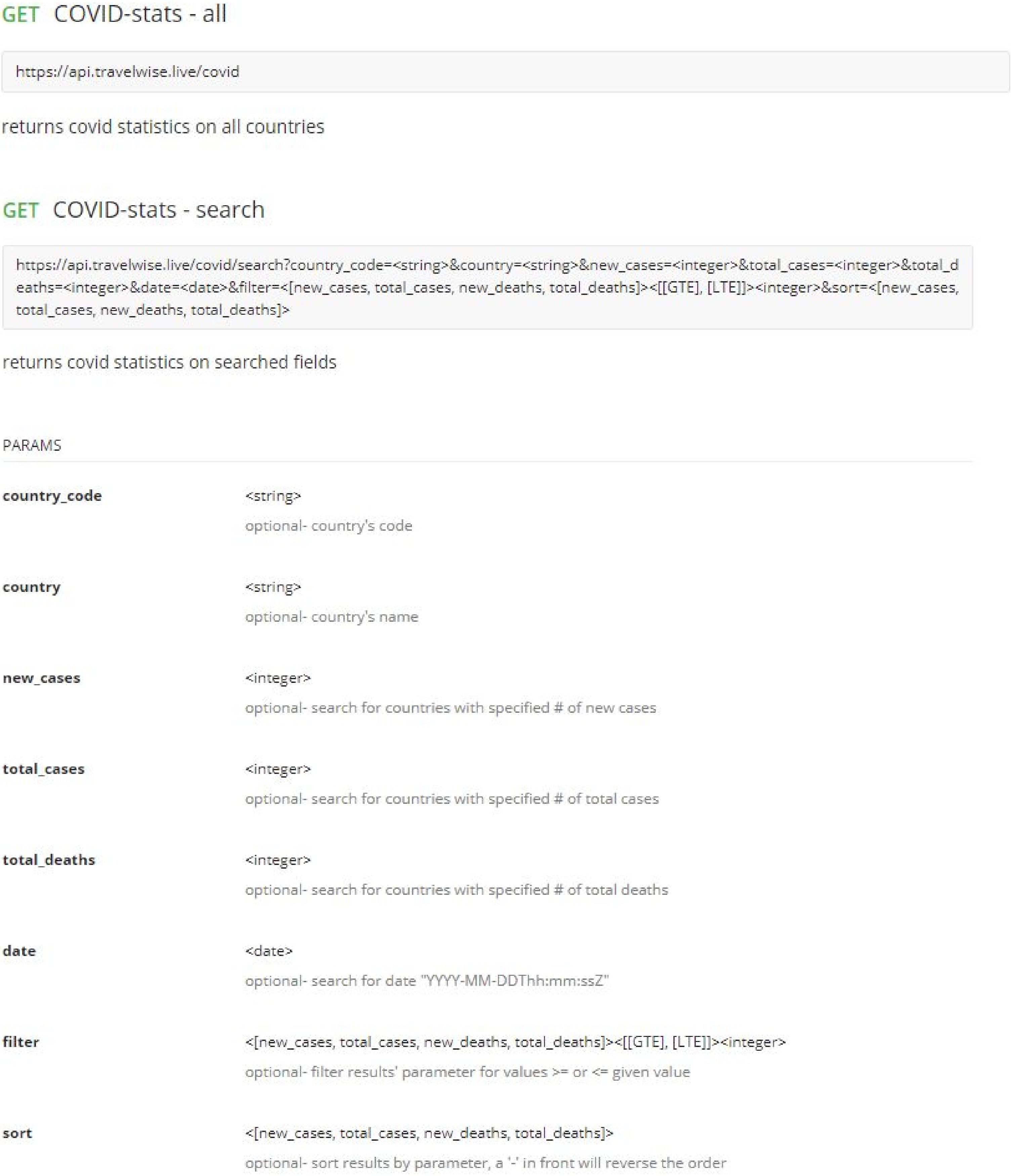
GET Airports - search

```
https://api.travelwise.live/airports/search?iata_code=<string>&airport_name=<string>&city_name=<string>&country_code=<string>&latitude=<float>&longitude=<float>&time_offset=<integer>&filter=<[latitude, longitude, time_offset]><[[GTE],[LTE]]><float>&sort=<[latitude, longitude, time_offset, longitude, latitude, country_code, country_name, city_name, airport-name, iata_code]>
```

returns coordinates, and time offset of all airports in search

PARAMS

iata_code	<string> optional- search by iata code
airport_name	<string> optional- search by airport name
city_name	<string> optional- search for airports in given city
country_code	<string> optional- search for airports in country by country code
latitude	<float> optional- search for airport at exact latitude
longitude	<float> optional- search for airport at exact longitude
time_offset	<integer> optional- search for airport in specified time zone
filter	<[latitude, longitude, time_offset]><[[GTE],[LTE]]><float> optional- filter results to be above or below coordinate, before or after offset
sort	<[latitude, longitude, time_offset, longitude, latitude, country_code, country_name, city_name, airport-name, iata_code]> optional- sort resutls alphabetically or numerically, a '-' sign before the parameter will reverse the order



GET COVID-stats - all

```
https://api.travelwise.live/covid
```

returns covid statistics on all countries

GET COVID-stats - search

```
https://api.travelwise.live/covid/search?country_code=<string>&country=<string>&new_cases=<integer>&total_cases=<integer>&total_deaths=<integer>&date=<date>&filter=<[new_cases, total_cases, new_deaths, total_deaths]><[[GTE], [LTE]]><integer>&sort=<[new_cases, total_cases, new_deaths, total_deaths]>
```

returns covid statistics on searched fields

PARAMS

country_code	<string> optional- country's code
country	<string> optional- country's name
new_cases	<integer> optional- search for countries with specified # of new cases
total_cases	<integer> optional- search for countries with specified # of total cases
total_deaths	<integer> optional- search for countries with specified # of total deaths
date	<date> optional- search for date "YYYY-MM-DDThh:mm:ssZ"
filter	<[new_cases, total_cases, new_deaths, total_deaths]><[[GTE], [LTE]]><integer> optional- filter results' parameter for values >= or <= given value
sort	<[new_cases, total_cases, new_deaths, total_deaths]> optional- sort results by parameter, a '-' in front will reverse the order

Models

For this second phase, our three models are Cities, Airports, and COVID regions.

Lower safety scores correspond to a safer environment. Scores are 1-100, 1 being unlikely to suffer from the metric, to 100 being very likely to suffer from the metric.

- **Destination Cities**
  - City ID
  - Name
  - Country
  - Country Code
  - Region
  - Longitude
  - Latitude
  - LGBTQ Safety Score
  - Medical Safety Score
  - Overall Safety Score
  - Physical Harm Safety Score
  - Political Freedom Safety Score
  - Theft Safety Score
  - Women Safety Score
- **Airports**
  - Iata code
  - Name
  - City Name
  - Country Code
  - Country Name
  - Time Offset
  - Longitude
  - Latitude
- **COVID Statistics**
  - Country Code
  - Country
  - Total Cases
  - Total Deaths
  - New Cases (daily)
  - New Deaths (daily)

Tools

We are using a variety of tools:

- **Docker:** We will use docker to create a docker image that packages our tool-chain and dependencies into one convenient container.
- **Postman:** We have used Postman to scrape our initial instances of our models. We have used Postman to create a shared workspace, making it easy to collaborate with team members when working with our RESTful APIs. We also have used Postman to design Travelwise's API. We have used Postman to design and implement our API and build tests for it.
- **React.js:** We use React.js for our front end.
- **Bootstrap:** We use Bootstrap as our primary CSS framework.
- **Yarn:** We use Yarn, a package manager, for our React App. It makes documenting and installing our dependencies easy and streamlined.

- **Visual Studio Code (LiveShare)**: We utilize Visual Studio Code in developing our React App. We also utilize the LiveShare extension, allowing real-time code collaboration and pair programming.
- **GitLab:** We utilize Gitlab for our version control.
- **Slack:** We've integrated slack to our GitLab repo, for communication purposes and issue tracking.

Hosting

- **Namecheap:** We used namecheap to buy our domains because they support whois protection and have cheap domain names.
- **AWS Amplify:** AWS Amplify allows us to host websites by simply committing and pushing code to our Git Repository. The service allows us to focus on simply writing code and not having to directly interact with our build server/terminal.
- **AWS Beanstalk:** Used to deploy our flask backend server. Used in conjunction with our PostgreSQL database to feed data that our front-end consumes
- **AWS RDS:** Used to host the database we built for Travelwise.
- **Create-React-App:** The Create-React-App allows us to have a modern build setup with pre-set configurations that make our react app compatible with most technologies. It is officially supported by Facebook, the creators of React so it is extremely reliable.

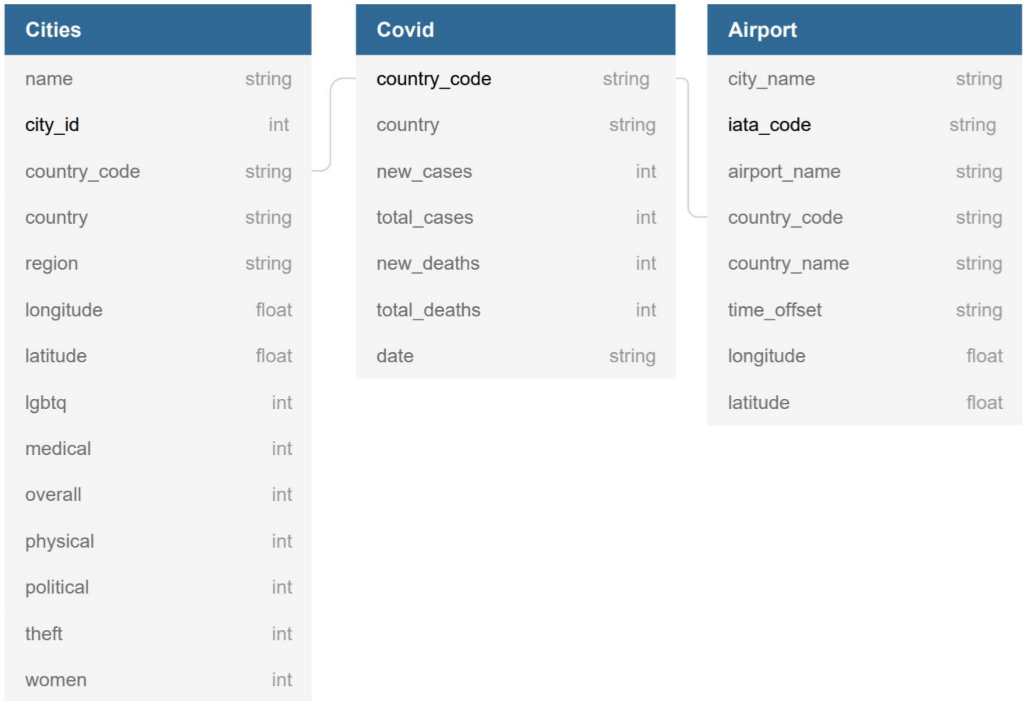
Database

- **Postman:** documentation of our api: https://travelwiselive.postman.co/collections/12799472-e9e5eb99-d6fe-49df-9435-6f8fc6c7e0d6?version=latest&workspace=4e45581b-ddf8-4fd5-82b8-31c31b43a9b5
- **Flask-restless:** Used to connect database to server
- **Flask-SQLAlchemy:** Used to operate on our database in flask
- **PostgreSQL:** Our relational database management system
- **pgAdmin4:** The GUI we used to construct the database.
- **PlantUML:** Used for UML diagram
- **Tables:** Cities, airports, countries
  - Cities
    - Primary key: city_id
    - Foreign key: country_code -> Covid
  - Covid
    - Primary key: country_code
  - Airport
    - Primary key: iata_code
    - Foreign key: country_code
  - Cities 1…* -- 1 Covid
  - Airport 0…* -- 1 Cities
  - Airport 0…* -- 1 Covid

- **NOTE**: We weren't very familiar with SQL datatypes going into this, and we discovered (before we had enough time to fix it) that the way we stored strings into the database were actually arrays of strings. What this meant is that when we processed the data as a json, we had to peel off the array characters to properly display the content. This is shown on every line with a string surrounded by the following expression: [“”]. We have to rescrape all the API’s to fix our data, so we plan to save that for phase III of the assignment.

- city_id: 3100212
- country: ["France"]
- country_code: ["FR"]
- latitude: 48.8412
- lgbtq: 0
- longitude: 2.3003
- medical: 0
- overall: 41
- physical: 0
- political: 28
- region: ["Île-de-France"]
- theft: 0
- women: 0

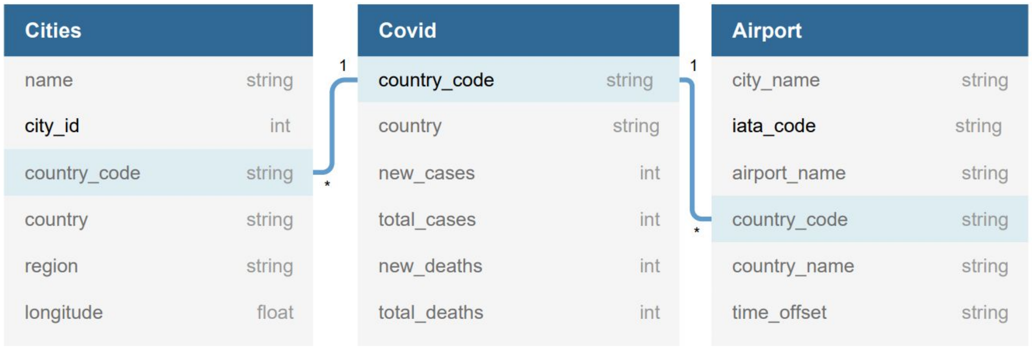
Database Diagram



Cities	
name	string
city_id	int
country_code	string
country	string
region	string
longitude	float
latitude	float
lgbtq	int
medical	int
overall	int
physical	int
political	int
theft	int
women	int

Covid	
country_code	string
country	string
new_cases	int
total_cases	int
new_deaths	int
total_deaths	int
date	string

Airport	
city_name	string
iata_code	string
airport_name	string
country_code	string
country_name	string
time_offset	string
longitude	float
latitude	float



Cities	
name	string
city_id	int
country_code	string
country	string
region	string
longitude	float

Covid	
country_code	string
country	string
new_cases	int
total_cases	int
new_deaths	int
total_deaths	int

Airport	
city_name	string
iata_code	string
airport_name	string
country_code	string
country_name	string
time_offset	string

Back-end

- **Flask:** Backend framework
- **AWS Elastic Beanstalk:** Used to deploy flask server
- **AWS RDS:** Used to host our PostgresSQL database
- **Database:** Info above.

Testing

- **Newman:** Used for Postman API Tests, automates tests for each request to API
    - In Gitlab Postman_tests CI pipeline
- **Jest:** Used for Typescript testing, tests if components rendered correctly
    - Not in Pipeline, not configured because CI incompatibility issues src/App.test.tsx
    - Located in src/App.test.tsx
- **Selenium:** Used for GUI testing, tests user interaction
    - Not in Pipeline, requires browser driver, incompatibility issues with gitlab CI
    - Located in travelwiseBackend/test/GUITests.py
- **Unittest:** Python extension used to test backend in yml pipeline
    - In Gitlab Backend CI pipeline
    - Located travelwiseBackend/TravelwiseBackendUnitTests.py
- **Docker:** Used for CI
    - Frontend Image “mitchellwatkins125/travelwise”
    - Backend image “mitchellwatkins125/travelwiseflask”
    - Postman image “postman/newman_alpine33”

Pagination

React-Pagination: Handles the logic of rendering a pagination component, since our databases are currently small, we make one fetch to the back-end and paginate the data when someone loads up one of our model pages.
Bootstrap: Handles the design of our pagination component, might customize later with our own CSS

Filtering, Sorting, Searching

Filtering: Fetch data, then parse data so that the data can be filtered according to the selected attributes; selected filters stored in array, where looping through, if current data matches the current attribute including as part of current state, otherwise is excluded.
Sorting: Fetch data, then parse data; afterwards, check the selected sort options and whether it is ascending or descending, then sort data based on the options in the respective model and set state.

Searching: Fetch and parse data, and then after receiving input for desired search, compare with all data attributes after calling toString() for data to check for all possible matches, include any potential matches and set state to only include updated matching results; search also includes a live highlighting function where every time new input is detected, the site renders with the matching attributes highlighted.

Visualizations

D3: Our group used d3 for most of our visualizations and the technology allows us to manipulate DOM elements in an efficient way allowing for easy creation of visuals for both our data and plant planet's data.
Recharts: Although recharts is somewhat outdated, the technology made creating the pie chart visuals with D3 a lot more streamlined and allowed us to easily animate the visual, allowing for stunning User Experience.

Animations

Framer-Motion: We used frame motion to animate some of our components. The framer motion library is vast and easy to use so it was perfect for our needs.

Refactoring

Functional Components: Every single page in our code base was split into functional components and anything that involved heavy management of state, we decided to leave as class components. Functional components are streamlined and allowed us to make beautiful concise code.
Front-end file system: Every directory inside our front end file “src” has no subdirectory since our website is relatively small scale and imports are very common. We made sure to use directories to organize different types of files but other than that, there was no real hierarchy to our file system since too much nesting would have made front-end imports efficient.
Back-end database: We converted every necessary column from a varchar[] to text in the Amazon database. Also, instead of re-scraping the API's all we had to do was change some properties via pgAdmin4, and provide Amazon Elastic Beanstalk a new image.
Back-end routing: Split up our single file application into many files, each handling routes for a specific table. Also refactored code that was previously operating on the contents of a varchar[], a string instead.

Presentation

Power-Point: We will use a powerpoint to demonstrate some of the website features and also compile it as a pdf for submission. Which slideshow technology we will be using is undecided.
Technical video: We will be making a technical video demonstrating our UI/UX, which video editor we will be using is undecided.