

Project Overview

The software was built with the base of Django for backend and whereas React and Bootstrap was designed to handle the frontend operation. The software highlights all the countries in the world map that matches with the search query.

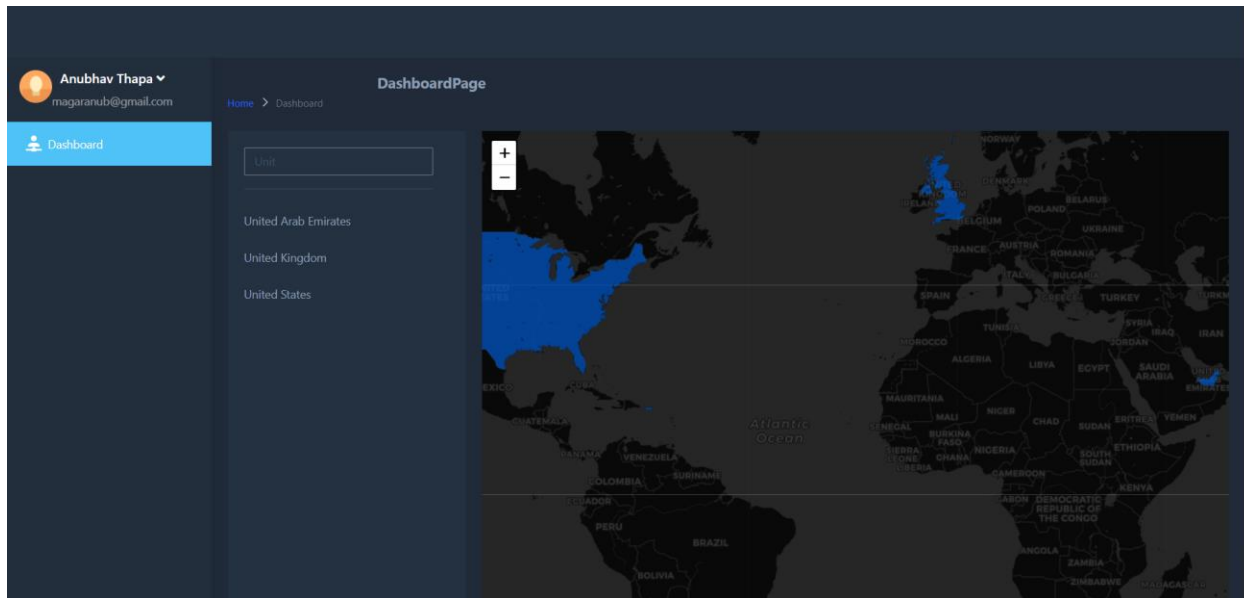


Figure: Overall view of the project

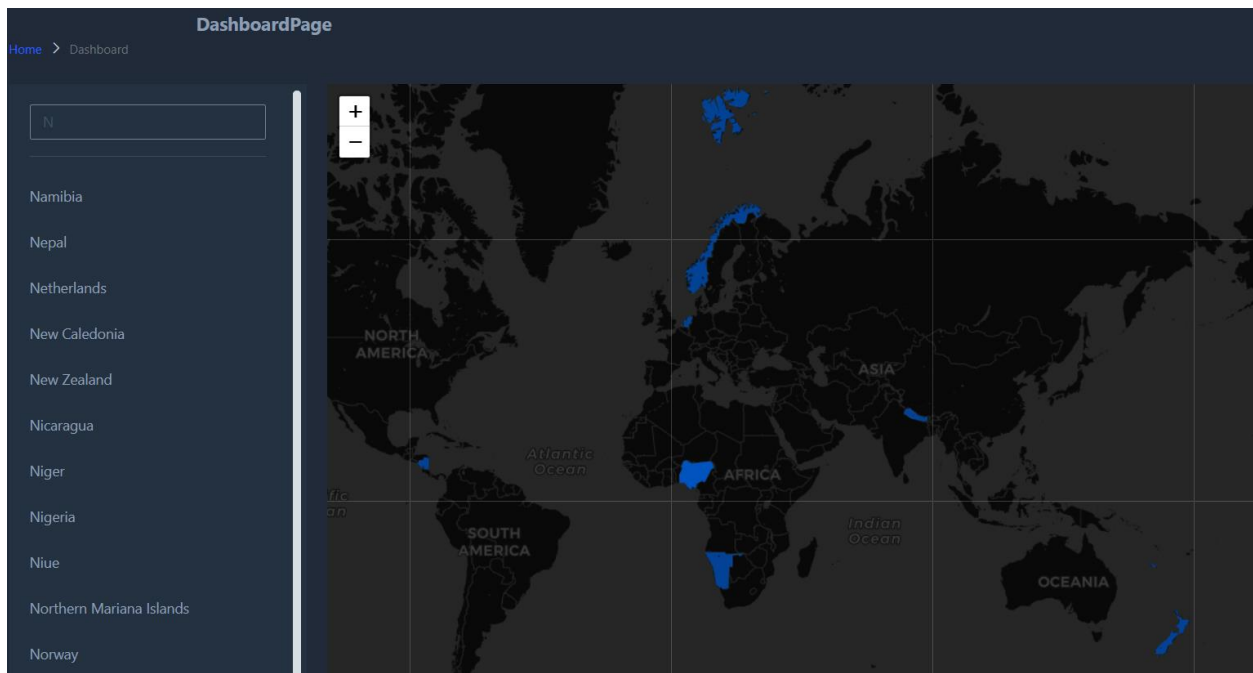


Figure: Search query and mapping function of the Application

Development Overview:

The project was developed on two phases Backend and Frontend.

Backend Development:

Since in this project Django was used to only structure the APIs for the frontend rather than performing web development activities in the Django. So Django Rest Framework was preferred as it make it efficient and easier to serialize and filter data while also providing OAuth support for the APIS

Phases of Backend Development:

- First a virtual environment was created in order to limit the scope of the project within the boundary of the environment
- Required dependencies such as Django, Django Rest Framework, Psycopg, pycountry, geopy was installed
- On the basis of mapped out architecture models and serializes were created in order to create table and serialize the data of those tables
- In the third step, all the required views and routes for the APIs was created so that the backend could interact with the frontend
- In order to get the country name and their geolocation the data were scrapped of from the internet in a json format
- The Oauth permissions was updated to only enable the Read features from the frontend as the program does not require Write permission from the Frontend on the present situation
- After the Frontend was developed all the static files were collected and Templates present in the settings file were updated to render the files of the build folder

Frontend Development:

For the Frontend Development React JS was preferred as it has a good support base community for developers and the components are rendered at very fast rate which was very essential for this project as the software needs to map all the geo location on the basis of search query and it also provides the ability to reuse component which was very beneficial in the project as most of components such as country name and geolocation was efficiently reused through this feature

Phases of Front End Development:

- First react project was initated and dependencies such as axios, react dom, react leaflet was installed through npm command
- In the Second phase, The frond end design was structured through the use of JSX, Bootstrap, CSS and HTML

- In order to provide appealing visual effects font icon and images were also integrated
- After that, Axios was setup and configured so that the frontend could interact with the backend through the API
- The combination of Openmap, Leaflet and React- Leaflet was used to render out all the geo locations on the world map which was provided by the API based on the search query
- At last step all the Frontend components were compressed into a build order in order to improve the app efficiency as well to reduce the memory storage during the production phase

Challenges:

Since the project was done on a short duration. The challenges faced were often quite low but still I had faced some challenges such as:

JsonField :

Problem:

One of the major obstacle of Django is that its built in model has no JsonField support for Sqlite Database. Since the geo location of all the countries had to be stored in a Json format in order to improve the efficiency when the data is stored ion the Backend and rendered at the Frontend as often times geo location of the countries have mixtures of variables such as String, Integer, Array and Float

Solution:

In order to solve these problem, PostgreSQL was used since it was more secured and the Django had the built in JsonField model for PostgreSQL hence making the stroing and retrieving process more efficient and effective

Leaflet:

Problem:

Since one of the requirement of the Frontend Design was to present it in a dark mode and this became quite obstacle when rendering the world map as the newer version of react-leaflet ignored any property changes made on the background

Solution:

In order to solve this restriction problem the Layer present in react was entirely changed through rendering the layer from another open source website and the layer fitted perfectly with the dark theme website

Future Improvements:

Since the project was only done on a shortly basis so there are lot the room for the improvement that could be performed such as:

- Locating and Marking the country with a popup marker that locates the country when a user clicks on the country name during a search operation
- Integration of admin, moderator and user panel
- Graph chart that displays the most searched country
- Availability of both the dark and light theme based on user preferences
- Integration of kelper library for more visual effects and animations on world map
- Detailed information such as the population, GDP and other economical factors of the searched country
- Based on the present situation active corona cases of the searched country could also be highlighted through the use of choropleth (darker color representing high active cases in relation to the population of the country) on the world map
- Vaccination tab where user can check on which city in their country the vaccination program is about to take place to achieve this feature the software could be programmed to send Google map location with direction based on the current user location

Guide to use the project:

- In order to use the app the user install all the required pip dependencies through the provided requirements.txt

pip3 install -r requirements.txt

- User must import the provided PostgreSql file(worldmap.sql):

worldmap	6/9/2021 10:39 AM	File folder	
Pipfile	6/8/2021 3:58 PM	File	1 KB
worldmap	6/9/2021 11:19 AM	Microsoft SQL Ser...	1,219 KB

- User must update the Password filed which in present in the DATABASE section at settings.py file located in worldmap folder

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql',  
        'NAME': 'worldmap',  
        'USER': 'postgres',  
        'PASSWORD': os.environ.get('POSTGRESS_PASSWORD'),  
        'HOST': '127.0.0.1',  
        'PORT': '5432',  
    }  
    # 'default': {  
    #     'ENGINE': 'django.db.backends.sqlite3',  
    #     'NAME': BASE_DIR / 'db.sqlite3',  
    # }  
}
```