PROJECT PLAN DOCUMENT

Project number	4
Project Title	Quantifying carbon footprinting for logistic hubs
Document	Project Plan
Creation date	22nd March
Created By	Kushagra Agarwal, Dama Sravani, Sriharshitha Bondugula, Shreeya Pahune
Client	Name of the Client : Gowtham Gollapalli Name of the organisation: Kaiinos

Brief problem statement

Quantifying carbon footprint using geospatial data

Global climate change is adversely affected by the increasing amounts of greenhouse gases being released. Carbon footprint is a measure of the amount of greenhouse gases produced which can be in different sectors such as Agriculture, Industrial, Transportation etc. In this project our goal is to make a person handling a logistics hub to quantitatively understand the carbon footprint due to the truck movement in the hub. Using this information then suggest alternative routes using a pgRouting server to minimize this emission.

Team Members

Kushagra Agarwal
Shreeya Pahune
Dama Sravani
Sriharshitha Bondugula

Team Communication

While on campus, we had weekly meetings with the client usually on Tuesdays at 5.30 pm. Due to the current situations team meetings and client meetings through regular Hangouts calls with client and MS Teams or Hangouts for the team.
MOMs are up to date and documents sharing through Google docs and common gitlab repository that includes the Client.
Apart from the meetings we are in regular communication with the TA on messenger and the client and the team on respective Whatsapp groups.

Development Environment

- Javascript(React JS) and OpenLayers (for front end)

- Flask (for backend development)
- PostgreSQL- with PostGIS extension and pgAdminIV (maintaining databases)
- Pgrouting extension (for routing and other network analysis functionality like spatial queries)

Milestone Schedule

Milestone	Release	Deliverable?
Reviewing carbon footprint literature	R1	Yes
Finalising requirements	R1	Yes
Taking input data	R2	Yes
Setting up pgrouting server	R1	Yes
Importing the London osm data	R1	Yes
Writing pgrouting queries on the imported data	R1	Yes
Visualising the data using OpenLayers	R1	Yes
Login / Logout	R1	Yes
Visualising pgrouting output using OpenLayers	R2	Yes
Find the route with least CFP	R2	Yes
Find the shortest path between 2 nodes.	R1	Yes
Calculate CFP of the company	R2	Yes
Display the hotspots in the city	R2	Yes
Final GUI for the webapp	R2	Yes
Use Satellite Data to find hotspot hubs with high carbon emissions(extension of the project)	R2	No

Feature description

Sprint 1 and Sprint 2 :
We have used these 2 sprints for requirement analysis.

1. Sprint 1 included studying about the carbon footprint calculator to **perfectly understand the problem statement** given by the client , reading up on new different tools that we are going to use **(Open Layers, Pgrouting, London data, POST GIS** etc) .
2. Sprint 2 included **setting up the tools and getting familiar with them.**

Sprint 3:
In this sprint we worked on setting up the server and creating databases. It included the following tasks.

1. **Setting Up pgrouting server :** pgRouting extends the PostGIS / PostgreSQL geospatial database to provide geospatial routing functionality. We have installed it and went through the algorithms that it supported. We have also learnt how to import osm data into it and tried to work with different pgrouting queries.
2. **Designing Spatial DB :** Creating spatial database in postgres.
3. **Create spatial data for London :** Creating a DB and importing london data to the database created.
4. **Setting up pgAdmin4**
5. **GUI prototyping :** We have made a GUI prototype using a prototyping tool 'evolus pencil'.

Sprint 4:

1. **User Registration:**
    - This feature is for a new user to register.
    - We used React-js to implement this feature.
    - There are 2 types of users to this feature Manager and Executive Officer.
    - User details are stored in postgre-SQL.
2. **User Login:**
    - The users who have registered can login and use the application.
    - Based on the type of the user (Manager or Executive officer), the corresponding dashboard is opened allowing the user to make use of the corresponding functionalities.
    - We have used Reactjs and postgre-SQL to implement this feature and there are 2 types of users to this use case too.
3. **Display the shortest path between the nodes:**
    - This is the feature used by Manager only.
    - Manager is allowed to select 2 nodes and then the shortest path between them is supposed to be displayed.
    - We have implemented selection of 2 nodes and sending them for the query.
4. **Display the map generated:**
    - This feature is to display a map to visualise the result and is the use case of both the users. Visualising the shortest path for a manager and visualising the hotspots for an executive officer.
    - Once data is uploaded onto the postgreSQL with postGIS extension(done in sprint 2), we loaded DB into QGIS and visualised it on the desktop that included integration of different layers.
    - The same OSM data is loaded as json into OpenLayers.
    - Adding dynamic functionalities to the map that is displayed on the browser using OpenLayers(Mouse positioning,Zoom in and Zoom out,Drag to span the map).

Sprint 5:

1. **Route-wise CFP calculation:**
    - This is the use case of a Manager.

- This feature enables him to know the CFPs along different routes between 2 nodes.
- This includes the task of writing pgrouting queries.

2. **Upload data:**
    - This is the use case of both the users and enables them to upload data upon which the queries are run.
    - This includes the task of writing DB connections.
    - (pending) ER Model of the database that is to be uploaded by the user is to be made.
3. **Display shortest path between two nodes (Re):**
    - This includes the task of running spatial queries and attribute queries.
    - We wrote the query to retrieve the shortest path when 2 nodes are given.
    - For this we used pgr_djikstra.
4. **User registration (Re):**
    - Different errors are handled.
    - Registration is allowed only with a strong password.
    - Password hashing is done.
5. **User Login (Re):**
    - Authentication using Google and Facebook is implemented.
    - Creation of jwt tokens is done for authentication after registration.
    - Forgot password is to be implemented.

Sprint 6:

1. **Display the map generated (Re):**
    - We are allowing only some nodes to be selected. So we are supposed to display all the nodes that are allowed for selection on the map as a Layer.
    - We are doing it using the GeoJSON object in OL.
    - We have to add GeoJSON Result (pgrouting for shortest path) into OL as Layer to visualize in browser.
    - We have to Visualize data into QGIS to understand its bounds and enable import into DB
2. **Upload data (Re):**
    - Apparently, as we are dealing with static data, we have to integrate databases for calculation of CFP along different paths so that we can display the path with least CFP.
3. **Display the shortest path (Re):**
    - The result of the pgrouting query should be visualised.
    - That is,we are converting the pgrouting data obtained via querying into a GeoJSON object so that it can be later loaded onto a map using openlayers in the browser.
4. **City-wise CFP calculation:**
    - This is the use case of an Executive officer. This feature is to calculate the CFP of the whole city.

- We have to integrate the DB with area-wise data so that area-wise calculation for CFP hot spots can be done.
- We should also import the area-wise data into postgreSQL (PostGIS) using QGIS so that it can be integrated.

5. **Display path with least CFP:**
    - The result of the pgrouting query should be visualised with changes cost for this case.
    - That is,we are converting the pgrouting data obtained via querying into a GeoJSON object so that it can be later loaded onto a map using openlayers in the browser.

Sprint 7:

1. **Display the map generated (Re):**
    - This includes the task of running spatial queries for the executive officer to see the map.
    - Using the GeoJSON format of the required Layer, we should add onto OL so that it can be displayed onto the main map.
    - We should add area-wise data as Layer on London Data using OL so hot spots can be easily labeled.
2. **Display hotspots with high carbon emission:**
    - This is the use case of an executive officer.
    - Net CFP of the city is calculated and the major areas or hotspots with more CFP should be displayed.
    - This includes the task of running attribute queries.
    - It also includes visualizing the hot spots on Map (color code them) based on the area-wise CFP calculation done.
    - We have to calculate the area-wise CFP so that hot spots can be found and labelled.
3. **Display company's total CFP:**
    - This is the use case of a Manager.
    - According to the data uploaded, the company's total CFP has to be calculated and displayed.
    - This allows the manager to take necessary decisions about the carbon credits of the company.
    - It includes the task of writing attribute queries.
4. **Pre-emptive carbon emission:**
    - This is the use case of a manager.
    - This allows the manager to know how much the CFP is going to be for certain data that he uploads.
    - So, based on the pre-emptive calculation he can take the decision of either changing the route to reduce his CFP or continuing with the same routes.
    - We should run attribute queries to implement this feature.

Sprint 8:

1. **Display the map for EE:**
    - This is the use case of an executive officer.
    - This is to display the city-wise map with CFP quantified.
    - Data from 2002-2011 was imported for various vehicles. The EE can select his choice of year and vehicle for which he wants his visualisation.
2. **Integration :**
    - In all the above sprints we have implemented different features by dividing them into tasks.
    - We have also implemented all the tasks separately.
    - This includes integrating all the components that are built till now.
    - It also includes designing and finalising the GUI of the app.
3. **Compiling,testing and error handling:**
    - We also have to test the integrated components.
    - We should test the different use cases for different types of data uploaded.
    - We should make sure that different errors are handled.
4. **Documentation:**
    - There were many changes the project and its requirements have gone through.
    - There were many challenges that we have faced during the development.
    - This includes making the documentation of all these and changes in the previous documents.

FINAL USE CASES IMPLEMENTED

1. **User Registration** - Users should be able to register specifying their usertype.
2. **Login** - Registered users should be able to login to make use of the app.
3. **Upload Data** - Users should be able to Upload the data necessary to make the calculations.
4. **Shortest path between nodes** - Given source and destination as input,find the shortest path between these two and display in the map
5. **Path with least CFP** - Given the number of trucks , source and destination as input,find the path with least CFP and display in the map.
6. **Route-wise CFP calculation** - Based on input provided (Number of trucks) and path chosen, net CFP for that path is printed.
7. **Net CFP of the company** - To calculate the net carbon footprint of the company in order to take the necessary measures.

8. **Choose nodes from a Map** - Allow the user to choose nodes from the map.
9. **Log or history** - Storing all the paths chosen and printing the log.
10. **Identifying hotspots with high carbon emission** - Based on the input data, display the hotspots with respect to the selected parameters.
11. **Identifying hotspots with high carbon emission** - Based on the input data, display the hotspots with respect to the selected parameters.
12. **Preemptive CFP calculation** - Based on input provided (Number of trucks) and path chosen, net CFP for that path is printed. If the input is not in the correct range, throw an error.