Paper- 3.2a

Project Report on

Kaise Diya

An Effective Cost Management System for Agricultural Produce for Farmers of India

Submitted By:

DARIVEMULA PREETHAM- Mtech Geoinformatics

NIRWAN- Mtech Urban And Regional Studies Department

LANDGE OMKAR PRAKASH- Mtech Water resources Department

Date: 08-07-2022

Submitted To:

SHIVA REDDY KOTI

Scientist/Engineer-SE
Department Of Geoinformatics, IIRS, Dehradun

INDIAN INSTITUTE OF REMOTE SENSING (IIRS)

INDIAN SPACE RESEARCH ORGANISATION (ISRO)

DEPT OF SPACE, GOVERNMENT OF INDIA

Introduction:

The farmers of a country are the torchbearers for carrying out farming to produce the most necessary commodities for better sustainable of the society. Hence, farmers play an important role in promoting the farming practices for a good agricultural produce for a country. A farmer must sell his/her agricultural produce yield to the markets to reach different communities, business players of the society. A farmer is paid as per the produce or yield sold by him/her and the selling price of a farmer or the cost price of the agricultural commodities are revised by the Union Government on a daily basis. This approach proposes a system for the farmers and the vendors whereby helping farmers to sell his/her agricultural produce for a price demanded by abiding to the Government's revised prices to ease the mode of selling the agricultural yield and thereby benefiting the farmer by eliminating the extra expenditure spent by him/her. This system is designed to help farmers choose the appropriate buyer of his/her agricultural produce meeting the expected price. This system can be made available as a web-portal which handles the requests of both the farmers and the vendors. Vendors on the other hand can make use of this web-portal for checking the details of the farmers offering different types of agricultural produce, their location and also the expected cost of selling proposed by the farmers which helps vendors in decision-making for buying the agricultural produce.

Objective-

To develop an web-portal which provides an effective mode and buyer for selling the agricultural produce of farmers therefore assisting farmers and vendors for effective decision making.

Dataset and data sources used:

The following are the data sources for this project:

- Current Daily Prices Of Various Commodities From Various Markets (Mandi)
  Link: https://data.gov.in/catalog/current-daily-price-various-commodities-various-markets-mandi

- Year-Wise Average Prices Of Indian Basket Of Crude Oil And Petrol, Diesel And Domestic LPG Retail Selling Price At Delhi From 2016-17 To 2021-22
  Link: https://data.gov.in/resource/year-wise-average-price-indian-basket-crude-oil-and-petrol-diesel-and-domestic-lpg-retail

Methodology and Tools used:

To develop a complete application, the following tools and programming language was used,

- Django-rest framework using python programming language was used for developing the backend for the application for handling the requests and responses in client-server interaction and for integrating databases with the application.
- PostgreSQL database for storing the non-spatial and the spatial information.
- Python packages like folium, geopy, OSMnx, are used for Modelling, handing, supporting Geo-spatial data in Django. Also, the python package Tkinter was used for developing GUI for a portion of our project.
- For the front-end Visualizing part HTML with JS and CSS were used with Django framework.
- For Geocoding and routing part between locations and for visualizing geodata on the map MapQuest API was used.

Workflow of the Project:

The following Business Process Model and Notation (BPMN) diagram shows the different components of the project and the interactions between them.

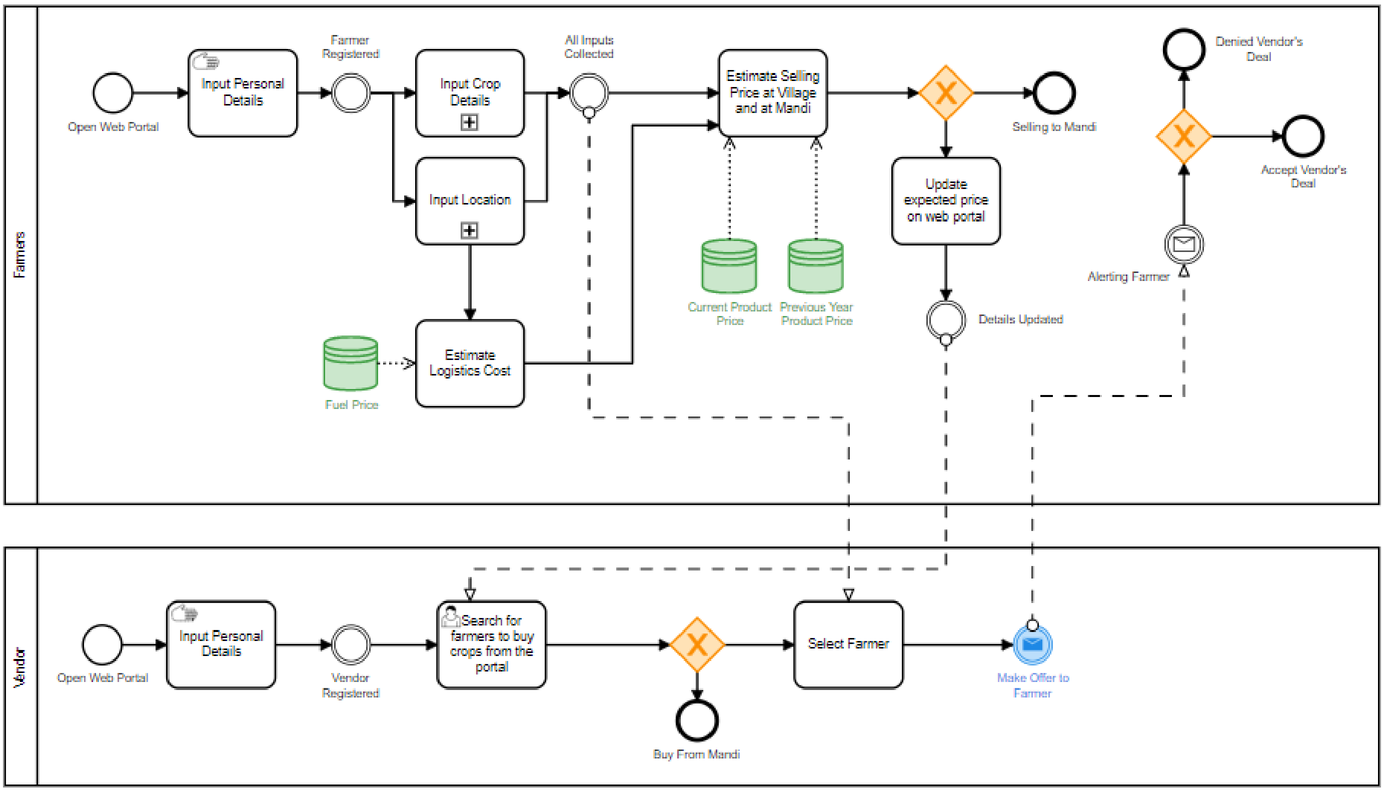


Figure 1 showing the BPMN model of the project with it's various components.

Procedure:

The procedure for the application Development in this work is Described in the following stages:

1. Configuring apps and developing models in Django
2. Migrating the models and Database connection
3. Using the templates HTML design for rendering the content onto the screen

Configuring apps and developing models in Django:

The first thing to do is to install Django framework library and it's dependencies in an python virtual environment.

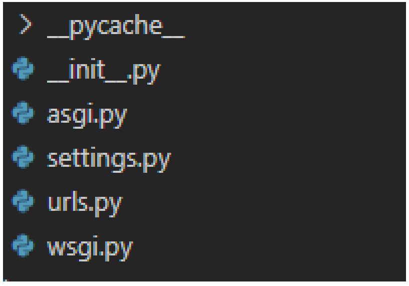
Next, Open a working directory to store all the project files and start Django project in it using the command

```
>> Django-admin startproject project-name
```

This command creates a main project directory with the name specified by the user at the project-name.

The main project directory contains the following folders and files.

- A folder named with the main project name and it consists the following files,



```
> __pycache__
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
```

The settings.py file is the file which sets all the necessary settings and configuration for the Django project.

- A manage.py file for handling the migrations and to run the server.

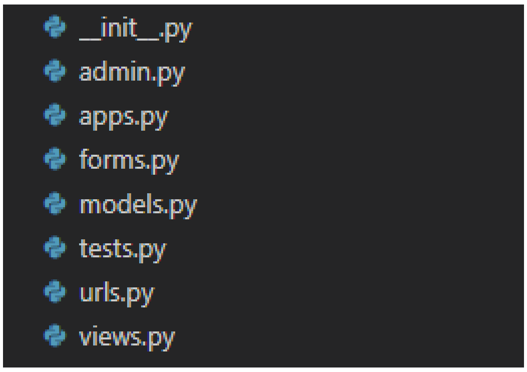
Next the main thing is to create apps. An App is a subset of the Django project and the application as a whole can be sub-divided into number of apps. Each app has its own functionality defined and is used to render different components of an application.

In a Django project, an app can be created using the following command,

```
>> python manage.py startapp app-name
```

The above command will create an app with the name specified by the user at app-name.

The django app created consists of the following files,



```
__init__.py
admin.py
apps.py
forms.py
models.py
tests.py
urls.py
views.py
```

The models.py file contains the models which are created in a Django project. A model in an Django project typically creates a table to in an relational database management system.

The views.py file contains the functions defined to return the responses to the user based on functions defined.

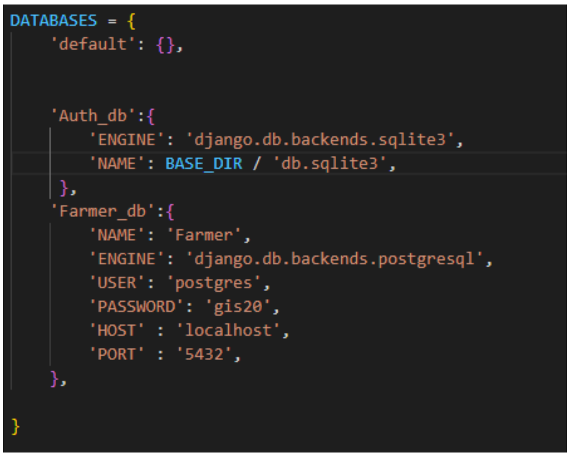
The urls.py file contains the url's for routing pages i.e. accessing the different functionalities using url's specified.

Migrating models and configuring the databases:

Next, the models must be specified within the Django app to be used to create the tables in an RDBMS.

Starting with the registration form, the form details are specified using a Django model. This model creates a table in the database with the name specified to the model.

After specifying the model, make sure that the database is properly connected by checking the settings of the Django app. This project uses a PostgreSQL database and the connection settings is shown below,



```
DATABASES = {
    'default': {},


    'Auth_db':{
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
     },
    'Farmer_db':{
        'NAME': 'Farmer',
        'ENGINE': 'django.db.backends.postgresql',
        'USER': 'postgres',
        'PASSWORD': 'gis20',
        'HOST' : 'localhost',
        'PORT' : '5432',
    },

}
```

The following command setsup the connection with the database specified in the settings,

```
>> python manage.py makemigrations
```

After making migrations the following command pushes the data into the database,

```
>>python manage.py migrate --database= Farmer_db
```

Thus, using the command above the data is pushed into the database and the data can be queried from the particular database.

Using the templates HTML design for rendering the content:

Finally, in-order to facilitate the user for inputs and for content rendering HTML based pages should be provided and all the necessary HTML pages for the project are mentioned in the templates.

And these templates are called in the views.py file of the app to render the content to the user.

Geocoding and routing services:

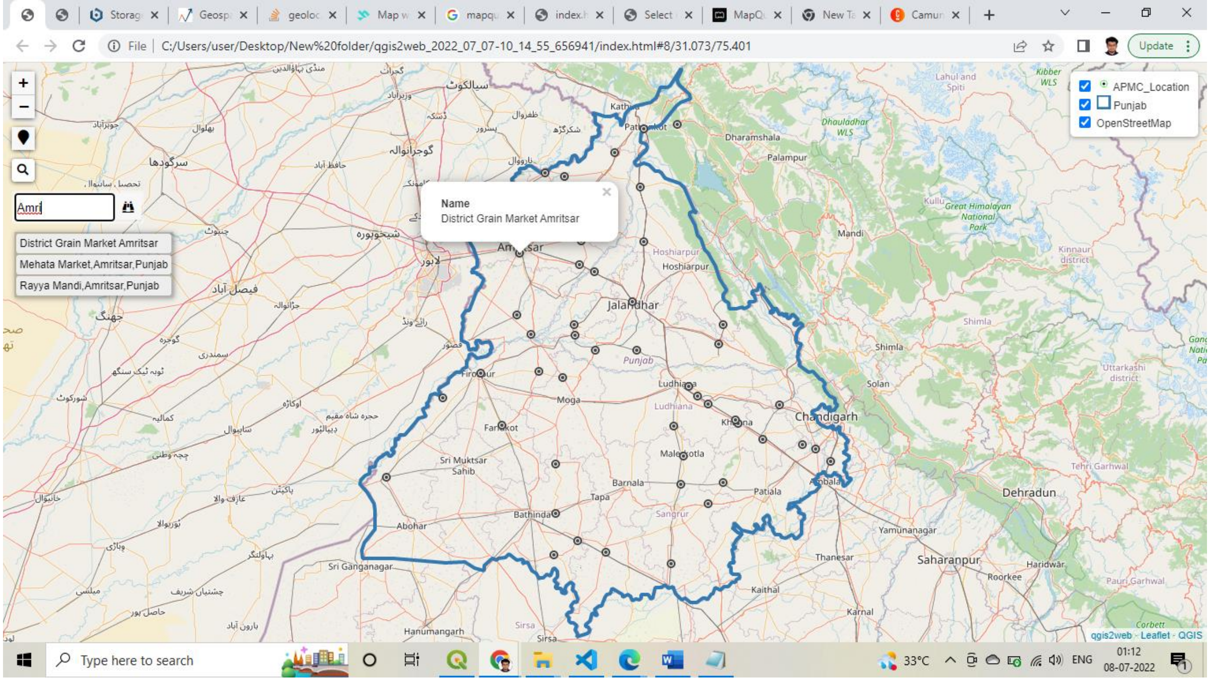
The geocoding and routing between two input locations are made possible using MapQuest API. Thus was the spatial data handled using Django.

Result:

The application was developed and tested for the performance. The figure below gives an idea of the location-based service and the response is shown to the user.

Estimating the logistics cost through map service:

The following image shows the locations of Mandis within the State Punjab. From these the user can select one destination.



C:/Users/user/Desktop/New%20folder/qgis2web_2022_07_07-10_14_55_656941/index.html#8/31.073/75.401

Amri

District Grain Market Amritsar

Mehata Market,Amritsar,Punjab

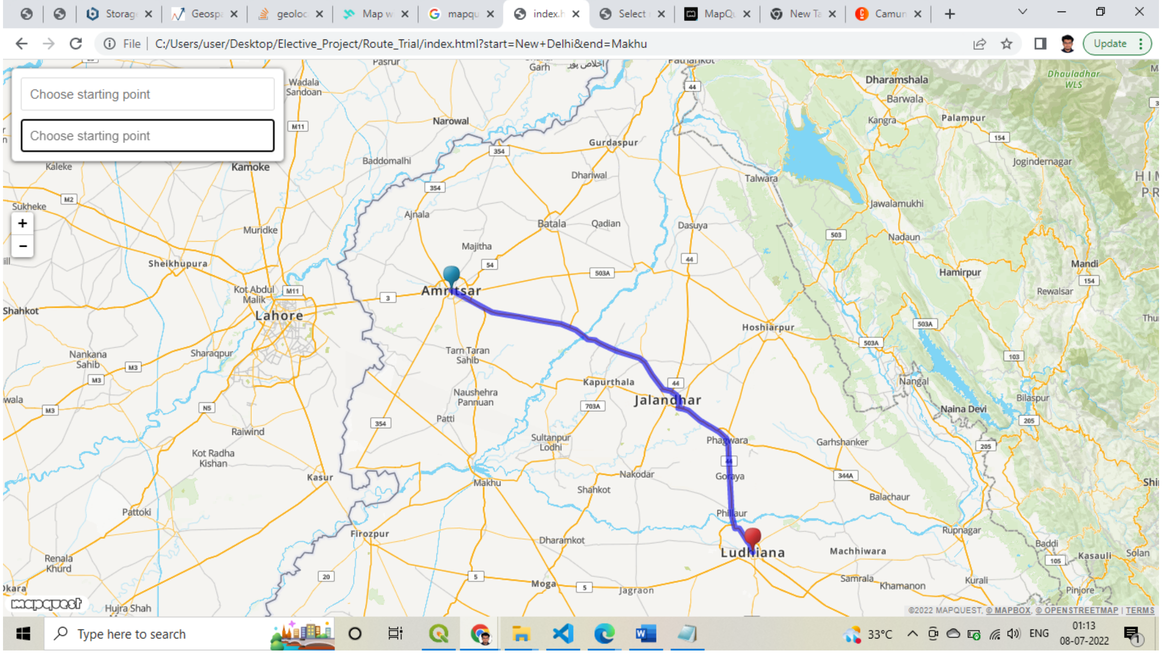
Rayya Mandi,Amritsar,Punjab

Name
District Grain Market Amritsar

☑ APMC_Location
☑ Punjab
☑ OpenStreetMap

Figure 2 Map showing the locations of mandis within the state of Punjab.

The location of the using the IP address. The destination is entered by the user.



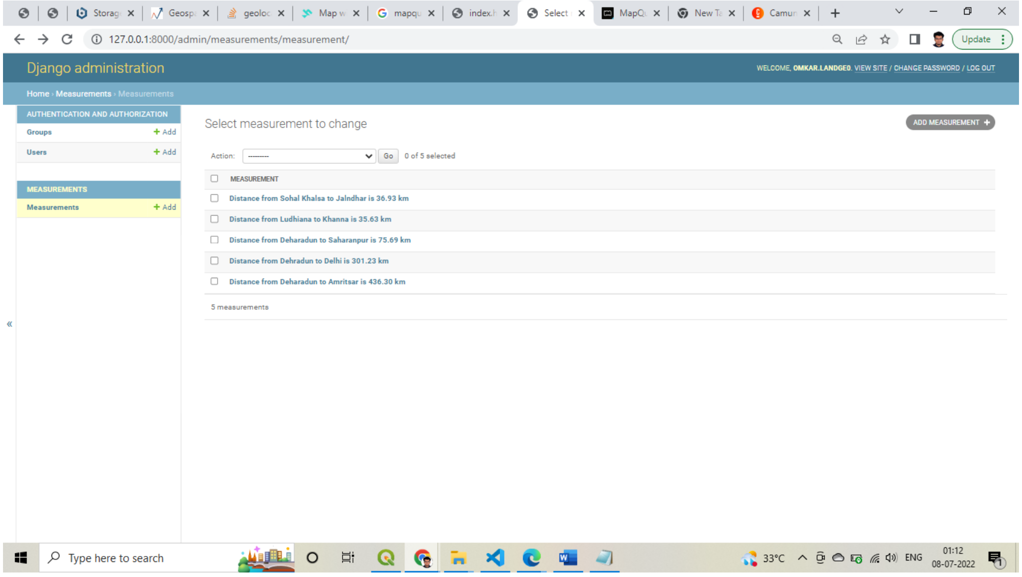
C:/Users/user/Desktop/Elective_Project/Route_Trial/index.html?start=New+Delhi&end=Makhu

Choose starting point

Choose starting point

Figure 3 Showing the route for estimating the logistics cost.

The following data is stored in the database and can be retrieved by Querying.


127.0.0.1:8000/admin/measurements/measurement/
Django administration
WELCOME, OMKAR.LANDGE0. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Measurements › Measurements
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
MEASUREMENTS
Measurements + Add
ADD MEASUREMENT +
Action: --------- Go
Type here to search
33°C ENG
01:12
08-07-2022

Select measurement to change

0 of 5 selected

MEASUREMENT
Distance from Sohal Khalsa to Jalndhar is 36.93 km
Distance from Ludhiana to Khanna is 35.63 km
Distance from Deharadun to Saharanpur is 75.69 km
Distance from Dehradun to Delhi is 301.23 km
Distance from Deharadun to Amritsar is 436.30 km

5 measurements

Figure 4 Figure showing the database records.

Conclusion:

The application is developed and is tested. This application provides the efficient means to sell the agricultural produce of a farmer by assisting him/her to make efficient decisions.

The code for this project is made available in the GitHub repository link: https://github.com/D-PREETHAM/Django-Project-Paper-3.2a.git