

Project Report

on

GPS Toll - based System Simulation using Python

Prepared by: **Shourja Dutta**

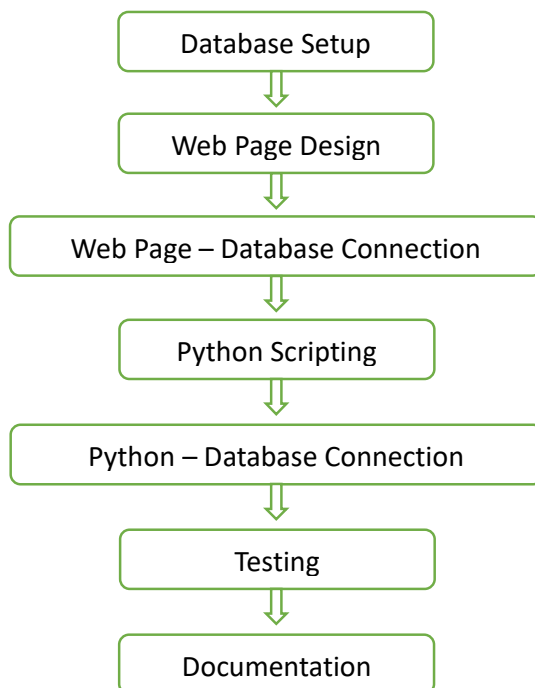
Introduction

Toll collection using Global Positioning System (GPS) ensures accurate distance-based toll calculations rather than fixed charge based on number of toll-road usage. Vehicle number plate / on-board tracking device is scanned by cameras installed at strategic locations along the toll-roads confirming presence of a vehicle at those GPS coordinates and enabling software algorithms to determine toll charges based on the distance travelled on these toll-roads.

Problem Statement

PS - 06 GPS Toll - based System Simulation using Python

Flowchart



Software used

1. Python3.12
2. Hypertext Markup Language (HTML5)
3. Cascading Style Sheets (CSS)
4. JavaScript (JS)
5. Hypertext Preprocessor (PHP)
6. Structured Query Language (SQL)

Project Components

1. database.txt

This file contains the code to create the database **toll_system** with tables '**users**' and '**transactions**' in the backend.

2. db.php

This file is used to connect the web pages to the database.

3. index.php

The web page consists of a text that welcomes the user to the website. It consists of two hyperlinks 'register' and 'login' which redirects the user to the respective pages.

4. register.php

The web page consists of two text boxes assigned for entering username and password and a button for the user to register. On clicking the 'Register' button, the username and password gets inserted as an entry into **users** table of **toll_system** database in the back-end.

5. login.php

The web page consists of two text boxes assigned for entering username and password and a button for the user to login. On clicking the 'Login' button, the program checks whether the credentials are available in any of the entries in the **users** table of **toll_system** database. Once validated, the user is redirected to **balance.php**.

6. balance.php

The web page consists of a text that welcomes the user and shows the user id which it derives from the **users** table of **toll_system** database. The current balance of the user is also derived from the same table and displayed. The transaction history is derived from the **transactions** table of **toll_system** database and displayed. The page

then has a text box for entering an amount by which the user wants to recharge. On clicking the 'Recharge' button, the amount in the text box gets added to the existing balance in the **users** table corresponding to the user's entry. The 'Logout' button redirects the user back to the **login.php** page.

7. recharge.php

This file contains the code to access the balance of the logged in user from the **users** table of the **toll_system** database and add the value entered in the text box in the **balance.php** page to the existing balance.

8. logout.php

This file contains the code to log the user out and redirect the user back to the login page.

9. toll_calc.py

Six places (New York, Atlantic City, New Jersey, Pennsylvania, Philadelphia, Washington DC) have been defined using latitudes and longitudes. A map centred around New York is created and the six places are marked on it. Toll roads are defined connecting each of the places with the others. Parallel non-toll roads are also defined. The map is saved to a file named **map.html**.

Great circle distance between two places are calculated using Haversine formula. Toll amount is calculated with a constant rate per km. The user must enter the user id to validate whether the user is valid. Once validated, the user is required to enter the source and destination and whether he/she would prefer to use toll road or non-toll road. Then the toll amount is printed along with the previous and new balances. The toll amount is added to the **transactions** table and the new balance is updated in the **users** table of the **toll_system** database in the back-end.

10. map.html

This web page contains the map with the six places and the toll and non-toll roads connecting them marked on it. It is generated once the **toll_calc.py** is run.

- **index.php**, **register.php**, **login.php** and **balance.php** are styled using CSS.
- All data that is supposed to be collected via GPS tracking during real-time application is taken as user input in this simulation as hardware implementation is not possible.

Conclusion

This software may be further developed by real-time interface with GPS based cameras to obviate need for manual data entry for source and destination. Actual distance travelled along toll-road can be calculated rather than considering least

distance along straight line. Linking with banking networks will initiate actual transactions for recharge and toll charge.

----- XXXXX -----