Title: MM 802 – Visualization Mini-project

Date: March 18, 2018

Names: Aman Singh (Dinghao Liu)

Project Title: Car share by Visualization of Geolocations using Google API

Abstract:

The visualization of Geographical location and information is crucial in the research as it is not only helpful in defining, explaining and navigating through the world but it is also helpful in providing the context to knowledge like demography, geology and history. In the mini project our aim is to visualize Geographical Locations using Google Maps API and implement it as a complete web application. Our website, allows the user to view the Geographical locations, route information by indicating their destinations and search for trips from departure point to destination point. It also allows users to make their own trips by registering on the website. This allows users to share their resources and reduce the pollution in the environment.

Introduction

Application Domain

In our mini project, we provide a visualization of geographical data. This project help those who want share their resources such as cars, bikes, buses, etc. to reach from one point to another. It help people manage and organize their trips. Our website is designed to help travellers to customize their trips and provide variety of travelling routes to them.

Motivation

This Geographical data visualization project is designed to help the users to figure out some simple geographic questions pertaining to places of interest. Being able to answer to all these questions, the users can easily plan their time and organize their trips. Such questions include:

- Where is this place located? What is its extent?
- What is the Start location and destination location?
- How many trips available from start to destination?
- Is it a Regular trip or one-off trip ?
- If one off trip then what is the date of trip ?
- How many seats are available per trip ?
- What is the price seat ?
- What driver details are present for the user to interact?
- What types of transports are available ?

Data

In this project we first render our own map using GeoJSON and d3.js and later on replace it with Google Map API, which can provide a better and detailed view of geographic locations. The map we rendered ourselves uses the Geospatial Data Abstraction Library(GDAL) to manipulate the Natural Earth shape files and Google Map API uses map data provided by Google. We used Geo Encoders to find the place located, start location and destination location. We have used twitter bootstrap for the creation of forms, few ajax scripts for Google API , some specific sort of fonts from font awesome and few other scripts and templates to style the pages of our website.

Existing work

Uberpool, Ola cabs, zipcars etc are some of the car sharing websites and application that are build on the same methodology.

References

https://fonts.googleapis.com
https://ajax.googleapis.com (for uploading the map)
https://maps.googleapis.com (populating the map with places and surrounding of world)
bootstrap.min.js (Downloaded bootstrap file)
https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.0.0/css/bootstrap-grid.css

Project Status

At the first place we considered rendering our own map using GeoJson and Google Map API (Rendering spherical global map using GeoJson and d3.js and get detailed location through Google Map API), in this application user enter the locations and locates them on a spherical geometry map and shows connections between two locations. However, our previous work does not provide detailed information due to scale of the original map data and lack of important statistic data like route, satellite view and other geographical details. Therefore, we re-designed our project and tried showing geographical information of interest on Google map.

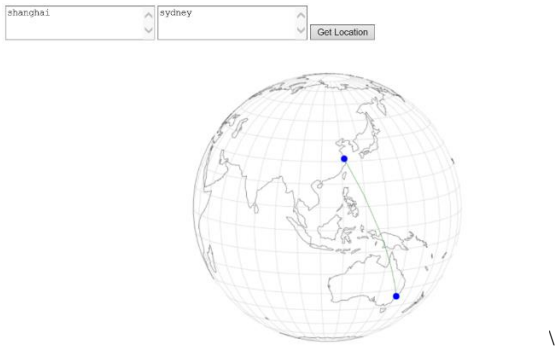


Fig1: Rendering spherical global map using GeoJson

At the present state, we have developed a web application which includes PHP and Javascript files that allows users to register and login to user account and provides users with geographic data visualization with Google API .To make the project more interesting we implemented this complete structured visualization on a car sharing application where users can pool cars for reaching from one place to another. The application is presently available for the users to access and test our work through

http://asingh.thecompletewebhosting.com/CarShareFinal

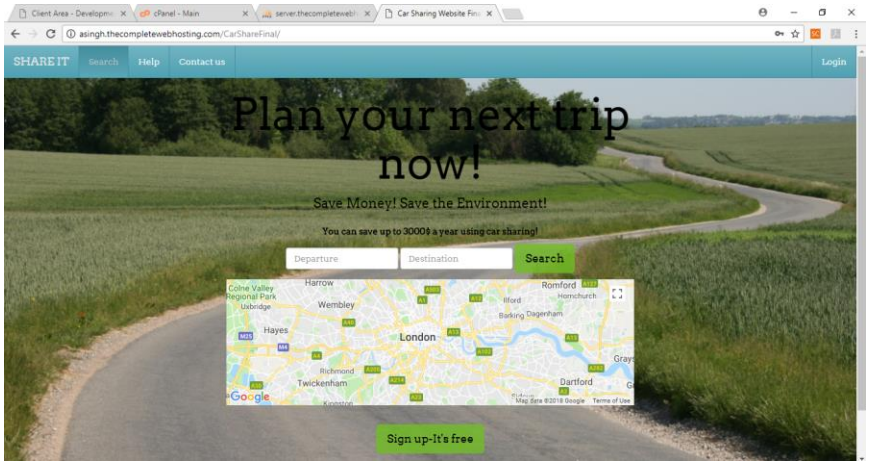


Fig2: Default page when website is loaded

Name	Work
Aman Singh	Registration, login and other forms(add trip, user details, etc.) creation, building the framework of the website and Implementation of functionality of website using ajax , php and javascript , Created the database using Mysql and deployed the website on a live server using fireFTP.
Dinghao Liu	Map rendering, Google Javascript API, Styling of website, setting up server using Xampp

Development Environment:

MySQL: It is used to create the most important part of any website, using MySQL a developer creates the database for the various aspects of the website. It is an open source relational database management system which allows the user to create and delete databases as and when required. Without a database the developers cannot store the information of various aspects of the website.

Ajax: It is widely used in our project because our project require to update a web page without reloading the page, request data from a server after the page has loaded, receive data from a server after the page has loaded and send data to a server in the background. Ajax is a set of Web

development techniques using many Web technologies on the client side to create asynchronous Web applications. With Ajax, Web applications can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behavior of the existing page. By decoupling the data interchange layer from the presentation layer, Ajax allows Web pages, and by extension Web applications, to change content dynamically without the need to reload the entire page.

FireFtp: FireFtp is used to deploy the web application live on the website so that users can access and test the application. FireFTP is a free, open-source, cross-platform FTP client for Mozilla Firefox. It supports FTP, FTPS, and SFTP.

Xampp: Xampp is a free and open source cross-platform web server solution package developed by Apache Friends. Xampp provides everything needed to setup a web server from server application to database and makes it extremely easy for developers to create a local web server for testing.

PHP: Our server-side webpage is developed through PHP – a widely used open source general-purpose scripting language that is especially suited for web developments and can be embedded into HTML. Different from client side script language, the PHP code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was.

Javascript: Javascript is a scripting or programming language that is used to create things like dynamically updating content, control multimedia, animate images.

CSS: Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language. It is used to set the visual style of web pages and user interfaces written in HTML and XHTML.
Google Map Javascript API:
Google Maps API can add maps based on Google Maps data to your application. The API automatically handles access to Goggle Maps servers, data downloading, map display, and response to map gestures. It also allows developers to add markers, polygons, and overlays to a basic map and to change the user's view of a particular map area.

GeoJSON: GeoJSON is a JSON based, open standard format designed for representing simple geographical features, along with their non-spatial attributes. The features include points (therefore addresses and locations), line strings (therefore streets, highways and boundaries), polygons (countries, provinces, tracts of land), and multi-part collections of these types.

D3.js: D3 (or D3.js) is a JavaScript library for visualizing data using web standards. D3 is capable of visualize variety of data using SVG, Canvas and HTML. D3 combines powerful visualization and interaction techniques with a data-driven approach to DOM manipulation.

Development Work

In the development of the car sharing web application we will first explain the basic setup of generating the Google API and then go toward the overview of few important files used in the development.

Basic setup

Go to Google map developer website then go to Web section. Click on Google map javascript key, Click on Get key, Create a new project , Create and enable api key and then save the key.

Overview of files in project

- We started by creating a php file named **index.php** . This particular file contains the links and scripts of all the files that are being used in this particular project because when the user tries to run the project either on local server or on a live webserver then this is the first file that is being executed, so it is very necessary that index.php must contain all the necessary details. In our project the details of various php files such as connection, logout, remember, navigation bar etc are present, in addition to these files it contains the details of various forms that are being created. All the Googleapis and bootstrap scripts should be present in this file for correct visualization of the data.
- **Styling.css** is a css file which is used for styling of the website. It gives the look and feel to the website. It is used for setting the background color and different font structure to make the website presentable.
- **Connection.php** is a very simple but very important file without this file a connection between various web pages cannot be established in this file the details of the server, user , password and database is provided this file create links between the webpages.
- **Signup.php** contains the signup form which the users have to fill in order to register on the website . It contains the logic to enter the user details and insert those details in the database at the end by executing the MySQL query at the time when the user hits the submit button. It also contains the logic generate a unique activation code using bin2hex(openssl_random_pseudo_bytes(16)) method and send an email to the user with a link to activate.php with their email and activation code.

Figure 3: Sign-up page code

- **Activate.php** is the file to which the user is re-directed after clicking on the activation link. It conatin two GET parameters namely email and activation key.
- **Activatenewemail.php** is the file to which contains logic that the email address used is the new email address. The link contains three GET parameters which are email, new email and activation key. If any of these three parameter is missing then an error is show otherwise the values are stored in three variables. After that the variables are prepared for the query, a query is executed to update the list of emails in the database.

Figure 4: Activatenewemail.php

- **Login.php** is a file which contains code for checking user login details, getting email address and password and showing error messages if the specific information is missing. It uses mysqli_real_escape_string() to escape special characters in a string for use in an SQL statement. It contains the logic for checking the combination of password and email address. This file also creates two variables suthentificator1 and authentificator2 to store the email address and password in rememberme table so that user doesnot have to remember the credentials everytime the user logins.
- **Mainpageloggedin.php** is the file which gets loaded when user is successfully logged in. It contain reference to various links , scripts and php files. It contains the details of various forms that are used in the application such as add trip, edit trip and Navigation Bar. It is the landing page after the user is logged in the application.

Figure 5: Mainpageloggedin.php

- **Addtrips.php** is the file which contains the code for the visualization of all the parameters that are required for adding the trip in the list of trips already available. First we started the session and made a connection, then we defined all the error messages that will appear if the user does not mentions mandatory information, the coordinates are checked by using GeoEncoders to provide the right location for both departure and destination location. All the logic for the details such as departure, destination , price , available seats , type of trip is coded in this file and at last a query is executed to store these all the values in database provided by the user.

Figure 6 : Addtrips.php

- **Gettripdetail.php** is file which is used to fetch the details of the trips that are created by different users it uses mysqli_query() and mysqli_fetch_array() method to fetch the results.
- **Mytrips.js** is the heart of our development work. It contains all the necessary logic for visualization. It creates a geocoder object to use the geocodes, logic to add trip form, hide/ show input depending on whether the trip is a regular or one-off, edit trip form, logic for the working of create trip button, edit trip button, delete button for deleting the trip, calendar input for the date, month and year details, getAddtripDepartureCoordinates(), getAddTripDestinationCoordinates(), submitAddTripRequest(), getTrips() and other very important functions.

Figure 7: Mytrips.js

- **Javascript.js** is the file which is defining the functionality of the web application. We started by creating a geocoder object to use the geocode, it contains the code for the ajax call for the sign up form, login form, forget password form, search form and some other forms once the form is submitted. It has two very important functions getSearchTripDepartureCoordinates() and getSearchTripDestinationCoordinates() which uses geocoder.geocode() to get the longitude and latitude of the departure and destination location.

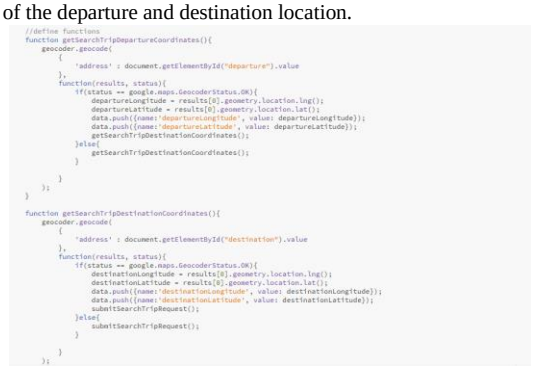


Figure 8: Javascript.js

- **Map.js** is used to set the map options. In this firstly we define the default latitude and longitude on the google map which will be load a default location at the beginning. It contains the logic for autocomplete departure and destination location, directionservice object to use the route method and get a result for our request, initialize map in the googleMap div and calculate route.

Figure 9: Map.js

There are other important scripts and php files which provide other feature set in our application which can be seen in the project folder. All the code is having comments in it for proper and better understanding of the system.

How to use

For local Servers using Xampp

- Create a Database in phpMyAdmin(using file notes.txt from project)
- Provide the details of the database in connection.php

- Run index.php on localhost after enabling apache server and MySQL in XAMPP.

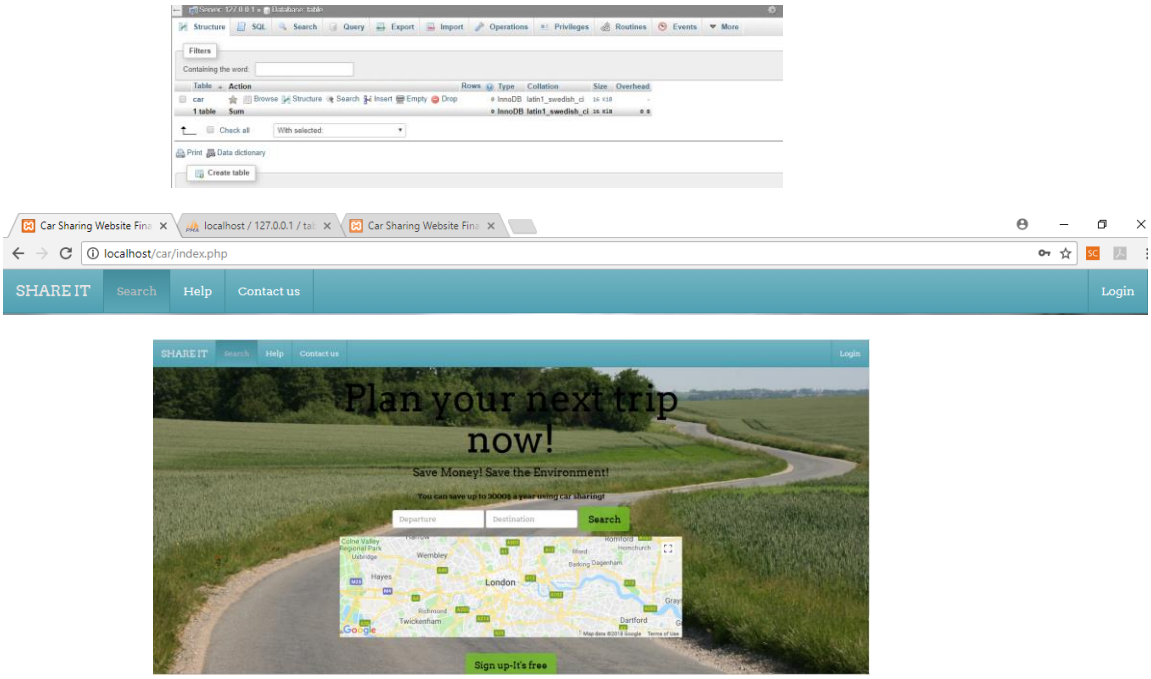


Figure 10: Local server Implementation

For Deploying the code on live server

- Sign up on a web hosting website and make a database(using file notes.txt) on that web hosting website.
- Add FireFTP on Mozilla Firefox and add an account on the FireFTP using the account details of the web hosting website.
- Upload Files from your computer to FireFTP.
- Using the domain provided below users can access and visit our website
  http://asingh.thecompletewebhosting.com/CarShareFinal/

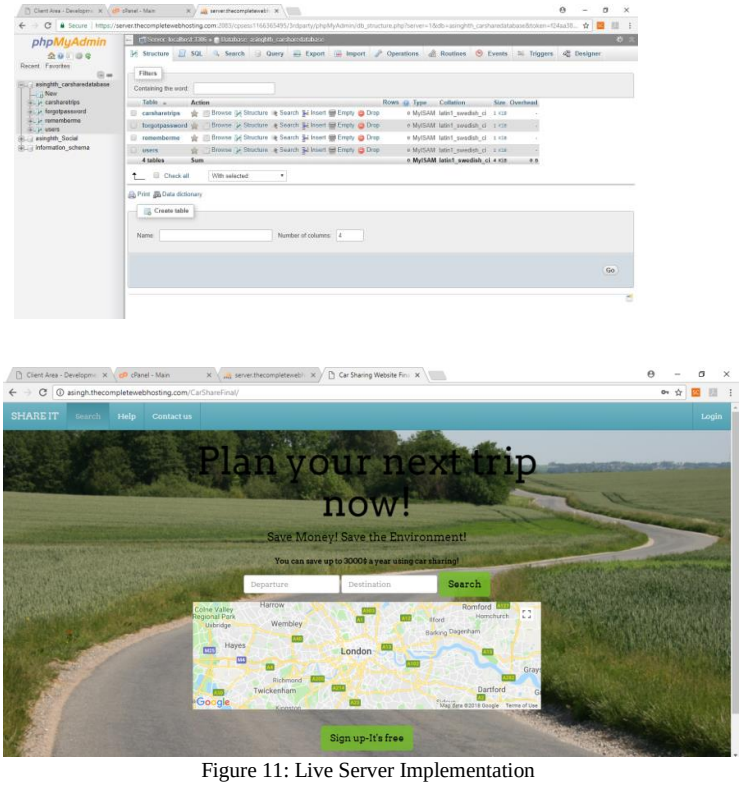


Figure 11: Live Server Implementation

Rest of the process is same for local server and on live server

1. **User Registration**

Press the green button at bottom of the page to sign up and register. Users need to fill-in the form below.

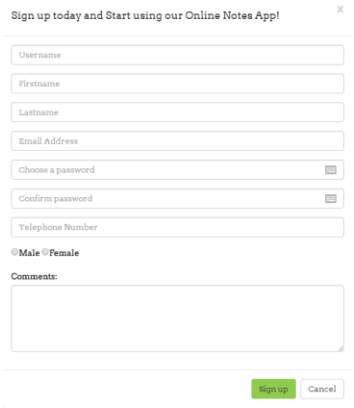


Figure 12: Registration Form

3. **Login**

To login, press the Login button at the upper right corner of the page. User password and email address will be checked for authentication for further login

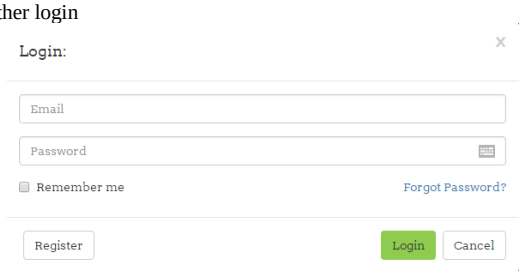


Figure 13: Login Form

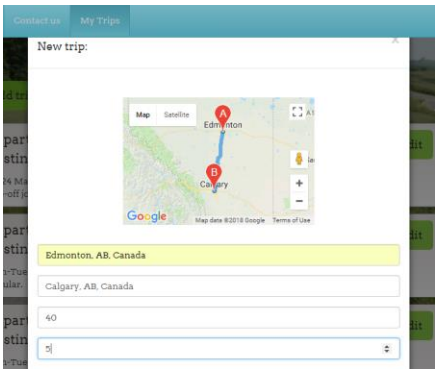
4. **Search**

To search for geolocation information, the user should enter their departure and destination in corresponding text area. And by pressing the search button on the right, the map below will update automatically.

Figure 14: Route

5. **Add trips**

To add the trips, user should first login to the website and then click on the Add trip button and then provide the necessary details in order to add the trips to the system.



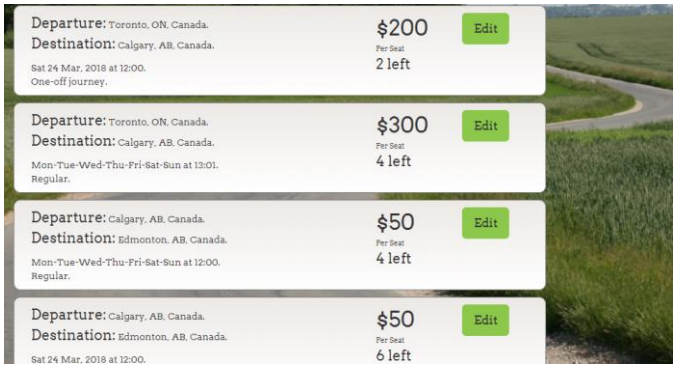


Figure 15: Add trips and Created Trips

Limitations:

- This system is inaccurate for the unpopular location within a city or province, probable reason can be the working of Google API
- The system does not provide multiple routes from departure to destination.

Concluding Remarks:

- We tried to implement a car sharing system which uses Google API and other state of the art technology. We accomplished a reliable car sharing application which has a state of the art security level.
- There is a tremendous requirement for the system like this because we are destroying our environment by over- utilizing the resources. We as a team learned a lot in this process, we explored various implementation environments and visualized various models and structures in order to make this application.
- Possible future extensions of this project are:
    - Provision of multiple routes in order to select the best possible route by the users
    - Implementation of Estimated Time of Arrival logic in visualization of data in order to save the time of users.
    - Implementation of traffic congestion algorithms in order to avoid high traffic areas.
    - Inclusion of unpopular areas of cities in the map to widen the area of visualization.