Assignment – 1

Building an EV database

1. Assignment Information

Course: MSCBD

Stage / Year: 1

Module: Cloud Computing

Semester: 2

Assignment: 1 of 3

Student Name: Suvajit Dulal Jana

Student ID: 3001336

1. Overview

This assignment was about building a PaaS based cloud application where users can manipulate the datastore. That is, we have created an application where we have created an Electric vehicle database with No SQL database and any logged in user may add, edit or delete the vehicle information add reviews for a vehicle rate them and can compare them side by side. The application is built with python 2.7 and ndb database.

This assignment depends on app.yaml which holds information about the platform. The information like the version info and libraries to be used throughout the platform and directory information. The assignment consists of more python file which carries out the functionality of the application.

1. Methods in Code
2. MainPage: - This method is the first method to load when the application is been started. It renders the main.html page where it checks whether the user is logged in or else provide with a login link. Only if the user is logged in the page will show links like add, edit user information, edit vehicle information, compare vehicles, search vehicles and add a review for the vehicles. The page also shows a logout button if the user is logged in. This page also shows a table contains all the cars in the datastore with a link leads to the detail information about the car with recent ratings. The python file also defines/declare the methods to access throughout the application at the end.
3. Edit: - This method renders the edit.html page where a logged in user can edit the user information. The page retrieves the user’s information by getting the current user’s id and update the changed information on the user id.
4. EvAdd: - This method renders the evadd.html where only a logged user can add the new vehicle. The method also checks if the vehicle with the same information exists it won’t add the vehicle. The page on top shows all the vehicle information available in the datastore. The table also contains links which will leads to the edit vehicle information page where user may edit or delete the vehicle information from data store.
5. EvEdit: - This method renders the evedit.html where a logged in user can edit vehicle information. If noticed the link for the page the is passed with vehicle information that is id for that vehicle act as a global variable. Which helps us to retrieve a particular vehicle information and edit that particular information. This page won’t load up if the user in not logged in.
6. EvSearch: - This method renders the evsearch.html which load a page with a form asking this criterial or a search query. The textbox on the page look for a particular keyword matches the keyword form the datastore the category is for the range search finding the vehicles falls on the particular range between minimum value and maximum value. If nothing is filled the page will show all the vehicle in the datastore.
7. EvDetail: - This method renders the evdetail.html. Same like EvEdit the link for the page is passed with vehicle information that is vehicle id that act as a global variable. The page shows the details for the same vehicle with average rating and all the recent reviews on this page this page can be access by any user even if the user is not logged in.
8. EvCompare:- This method renders multiple pages one is evcompare.html and error.html in the evcompare page the user have to select 2 different vehicle from the dropdown list and the page will show the comparison between those two vehicles highlight the positive things like recent year, high battery size, high WLTP range, less cost, high power and high average rating with green or else red. The name of the vehicle in the comparison table consist of like which will lead to details of that vehicle. If there are any errors like compare the same vehicle or did not select any vehicle then method will render the error.html page display the error.
9. EvReview: - This method renders the page review.html. This page will load only If the user is logged in. Only a logged in user may add a review for a selected vehicle. The review page consists on a form where the user has to select a car want to review add some feedback for the vehicle and rate the vehicle. The method will store the review with the user id and a time strap.

There are 3 more methods used to create and declare database or datastore those are

1. MyUser: - It declares the data structure of the user. The data datastore contains the email address, name and age for the user
2. ElecVel: - It declares the data structure for the vehicle information. This data datastore contains the name of the vehicle, it’s manufacturer, year of manufacture, battery size wltp range, cost and power.
3. EvReview: - it declares a datastore to store reviews and rating for the vehicle it stores the user’s id of the user writing the review, car’s id for which the review is been written, name of the car, feedback, rating and time at which the review have been add.

We have use ndb datastore to declare the database. It is a NoSQL database. It is a collection of entities. The ndb database generates it own unique key based on time. The key can be user define but if the user does add and key the datastore will create its own key. The key is like primary key and it will help to retrieve or identify the data using that key. The only difference is that the key is not the property of the data that is been stored. The ndb datastore uses structureProperty class to have a nested structure. It is integrated with automatic caching, which help the application to respond faster in inexpensive manner.

We have also use jinja2 library which helps us to load multiple environment in a web application. The jinja2 help us to get or post data on the webpage and run some loop and conditions directly on webpage. It also helps us to control the white page that is without loading the whole page show the results like ajax.

We have also used webapp2 library with help the application to handle the HTTP request and respond to it. Basically, help us to interact with the web application. It is light weight frame for python to handle HTTP request and responses. And is easy to use for developer who have just started learning the google app engine. The webapp2 can also be used as normal web application responder outside google app engine.

The User interface design is kept default. As per now the application page greets the user as a guest as the user information is updated the application greets the user by its name and bellow that all the useful links will be displayed. This User interface was created by keeping in mind to show all the useful information by one click and tried to show all information under one page.