**CHAPTER 1**

1. **INTRODUCTION**

**1.1 SYSTEM STUDY**

The purpose of this project is to build an “vehicle reservation system”, a place for enterprise rental company to maintain the vehicle details and to alert expiry of vehicle insurance, adherence towards scheduled maintenance and notification towards ageing vehicles. The system consists in a web-portal where registered users on logging in they can able to register new vehicle details, view registered vehicle details, update vehicle details and search vehicle details. Registration of users is preceded by a “admin registration”, after successful registration they can able to login to the system can manipulate all the vehicle details. The system is realized with model-view-controller pattern. The view which consists of all the user interface modules sends updates to the controller, the controller updates the model, and view gets updated directly from the model.

**1.2 OBJECTIVE**

The primary aim of the project is to trigger Email notification to admin to notify –Expiry of Insurance in next 15 days or Service due in next 15 months. Then, the admin can able to manipulate and maintain all the vehicle details. Admin as to keep on updating the last service date of the serviced vehicle, service due date of the vehicle and updated insurance expiry date of the vehicle. Admin on entering into the system he/she can able to search the vehicle details and details will be displayed in red if the insurance expiry date falls within 15days else the details will be displayed normally. Admin can able update vehicle details based on the selecting the criteria of the vehicle fields and also the vehicle search is based on the selecting the criteria of the vehicle fields.

Salient features of the project :

1. Paperless vehicle details reservation system.
2. Mail remainder to admin regarding insurance expiry and service due date.
3. Reliable admin validation & checking.

**1.2 SCOPE**

"Vehicle Reservation System" is an online vehicle details reservation system for enterprise rental company. An all-inclusive vehicle rental management tool is necessary for seamless operations of an enterprise rental company. Centralized inventory management is required to alert expiry of vehicle insurance, adherence towards scheduled maintenance & notification towards ageing vehicles This also helps to oversee vehicle in use at different branches of the company and be supplied with revised models of the vehicles from the central inventory to the branch. This also helps to avert non-availability of vehicles at a branch store at any given time.

The solution developed will address the objective in a holistic manner and will have all the features and functionalities which shall let the fleet management company be able to record & update the vehicle details along with their insurance details. The application will also run batch jobs to alert the admin on the expiry of the vehicle insurance to ensure the Vehicles moving out are 100% checked and approved.

Online application of the whole system helps easy access anywhere in the organization. The time taken for process completion is now largely reduced. The database is managed by MySQL, so data will be easily accessed and retrieved. Data cannot be viewed or edited by unauthorized personnel. So, this automated and computerized system is safe, fast and user friendly. The front end designed using java server page that is connected to the database using servlet.

**CHAPTER 2**

**2. GENERAL DESCRIPTION**

**2.1 PRODUCT PERSPECTIVE**

The proposed computerized "Vehicle Reservation System" site has made vehicle details reservation process simple. The only pre-condition is that the user must register and authenticate before he/she can take part in the maintenance and manipulation of the vehicle details.

Maintaining all the registered vehicle details and it will be manipulated by the registered admins. Admin has access to update the vehicle details and an insurance remainder mail will be automatically triggered from the server to the admin when the insurance expiry date falls with 15 days or service due date falls within 15 months.

Admin if logged into the system, he/she can able view the status of all the vehicles and they can also update all the vehicle details.

**2.2 USER CHARACTERISTICS**

Here admin is the only user of the system. The system is user friendly; hence the user should not need any software and hardware knowledge. They are assumed to have basic knowledge of computers. Friendly user interface, user guide must be sufficient to educate the users on how to use this product without any problems or difficulties.

**2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS**

* The information of admin and vehicle details gets stored in the database.
* Apache Tomcat 7.0 Web Server.
* It can be operated at any time.
* Trigger an mail if insurance expiry date falls with 15 days or service due date falls within 15 months only when the server starts.

**CHAPTER 3**

**3. REQUIREMENTS**

**3.1 FUNCTIONAL REQUIREMENTS**

**3.1.1 User Module**

**Prerequisite (**user signed in) for all requirements below

**Requirement ID** Req\_1.1

**Title** Login in module

**Description** This action is done to login to the site

**Requirement ID** Req\_1.2

**Title** New User Register

**Description** This action is done to enter the details of new user.

**Requirement ID** Req\_2.1

**Title**  Update vehicle details

**Description** This action is done to update existing vehicle details.

**Requirement ID** Req\_2.2

**Title** Create new vehicle details

**Description** This action is done to create new vehicle details.

**Requirement ID** Req\_3.1

**Title** Search vehicle details

**Description** This action is done to search vehicles.

**Requirement ID** Req\_3.2

**Title** Search criteria

**Description** This action is done to search vehicle based on criteria.

**Requirement ID** Req\_3.3

**Title** Display search vehicle

**Description** This action is done to display vehicle details in red if the If Expiry

of Insurance or Service due falls in next 15 days.

**Requirement ID** Req\_4.1

**Title** Notify Admin

**Description** This action is done to trigger Email notification to admin to notify –

Expiry of Insurance in next 15 days or Service due in next 15 months.

**3.2 NON-FUNCTIONAL REQUIREMENTS**

**3.2.1 Error Handling**

* Vehicle reservation system will handle expected and non-expected errors in ways that prevent loss in information.

**3.2.2 Performance Requirements**

* The system accommodates high number of vehicle information.
* Responses to view information shall take no longer than 5 seconds to display.
* It will take 10 seconds to trigger the remainder mail when the server starts.

**3.2.3 Safety Requirements**

* Maintainability
* Reliability.

**3.2.4 Security Requirements**

* System will use secured database.
* Vehicle details are updated only with admin’s knowledge.

**3.3 USER INTERFACE**

**3.3.1 Hardware Requirements**

Hard disk: 250 GB or above

RAM: 2 GB or above

Processor: i3 or above

**3.3.2 Software Requirements**

Operating system: windows 7 or above

Web server: APACHE TOMCAT-7 or above

RDBMS: MySQL

**3.3.3 Technologies Used**

**JSP**

Java Server Pages (JSP) is a Java standard technology that enables you to write dynamic, data-driven pages for your Java web applications. JSP is built on top of the Java Servlet specification. The two technologies typically work together, especially in older Java web applications. From a coding perspective, the most obvious difference between them is that with servlets you write Java code and then embed client-side markup (like HTML) into that code, whereas with JSP you start with the client-side script or markup, then embed JSP tags to connect your page to the Java backend.

JSP is also closely related to JSF (JavaServer Faces), a Java specification for building MVC (model-view-controller) web applications. JSP is a relatively simpler and older technology than JSF, which is the standard for Java web frameworks like Eclipse Mojarra, MyFaces, and PrimeFaces. While it is not uncommon to see JSP used as the frontend for older JSF applications, Facelets is the preferred view technology for modern JSF implementations.

While JSP may not be your first choice for building dynamic web pages, it is a core Java web technology. JSP pages are relatively quick and easy to build, and they interact seamlessly with Java servlets in a servlet container like Tomcat. You will encounter JSP in older Java web applications, and from time to time you may find it useful for building simple, dynamic Java web pages. As a Java developer, you should at least be familiar with JSP.

This article will be a quick introduction to JavaServer Pages, including the JSP Standard Tag Library (JSTL). Examples show you how to write a simple HTML page, embed JSP tags to connect to a Java servlet, and run the page in a servlet container.

**SERVLET**

A **servlet** is a Java programming language class that is used to extend the capabilities of servers that host applications accessed by means of a request-response programming model. Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by web servers. For such applications, Java Servlet technology defines HTTP-specific servlet classes.

The javax.servlet and javax.servlet.http packages provide interfaces and classes for writing servlets. All servlets must implement the Servlet interface, which defines life-cycle methods. When implementing a generic service, you can use or extend the Generic Servlet class provided with the Java Servlet API. The HttpServlet class provides methods, such as doGet and doPost, for handling HTTP-specific services.

**BOOTSTRAP**

Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website. It is absolutely free to download and use. It is a front-end framework used for easier and faster web development. It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others. It can also use JavaScript plug-ins. It facilitates you to create responsive designs. It is very easy to use. Anybody having basic knowledge of HTML and CSS can use Bootstrap. It facilitates users to develop a responsive website. It is compatible on most of browsers like Chrome, Firefox, Internet Explorer, Safari and Opera etc.

**ECLIPSE**

Eclipse and MyEclipse for the programming part of the project we used Eclipse, that is a very powerful open source integrated development environment (IDE). This IDE offers several services to the developer: it has compiler aware editing; syntax errors are highlighted when they are made, as are simple semantic errors such as missing declarations. Eclipse supports method completion, shows class interfaces concisely in graphical notation and supports interactive exploration of a program, through features such as fly-over name resolution. In addition, Eclipse supports Software Engineering principles such as packaging, debugging, testing, refactoring and versioning [2]. Another very important advantage of using Eclipse is that several plug-ins can be installed on it, like MyEclipse, that is particularly useful when working with Enterprise JavaBeans.

**Database and JDBC**

As database for the project we used PostgreSQL, that is an open-source relational database. A relational database is a type of database management system (DBMS) that stores data in the form of related tables. Relational databases are powerful because they require few assumptions about how data is related or how it will be extracted from the database. As a result, the same database can be viewed in many different ways. The most common options when choosing a DBMS are Oracle [9], MySQL [7] and PostgreSQL [10].

**CHAPTER 4**

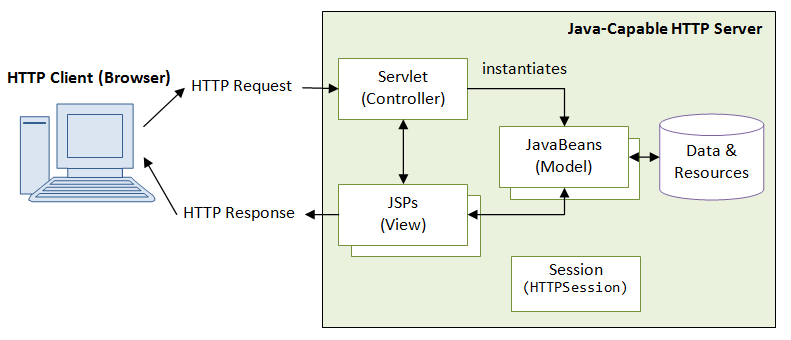
**4. DETAILED DESIGN**

**4.1 ARCHITECTURAL DESIGN**

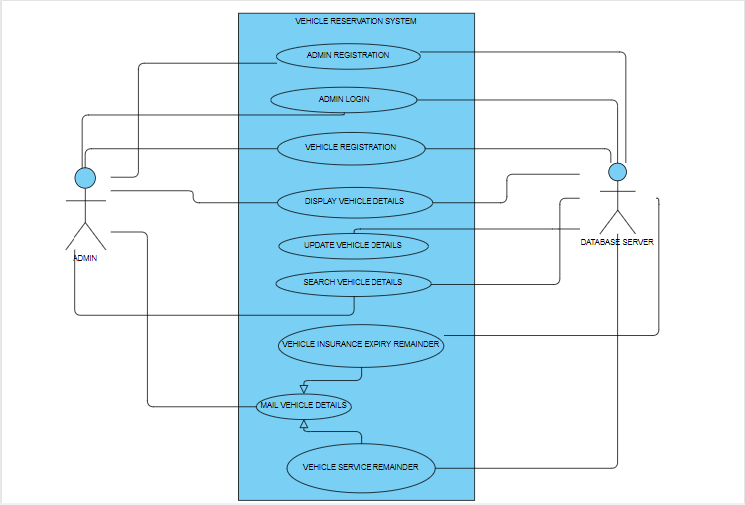
Model View Controller or **MVC** as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts −

* **Model** − The lowest level of the pattern which is responsible for maintaining data.
* **View** − This is responsible for displaying all or a portion of the data to the user.
* **Controller** − Software Code that controls the interactions between the Model and View.

MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response.

****

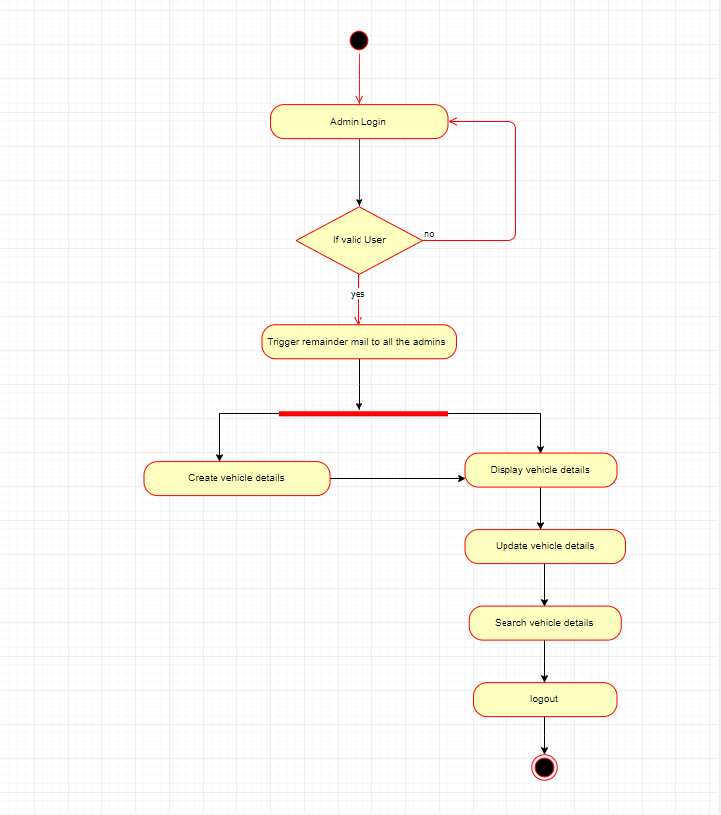
**4.2. USECASE DIAGRAM**



Fig

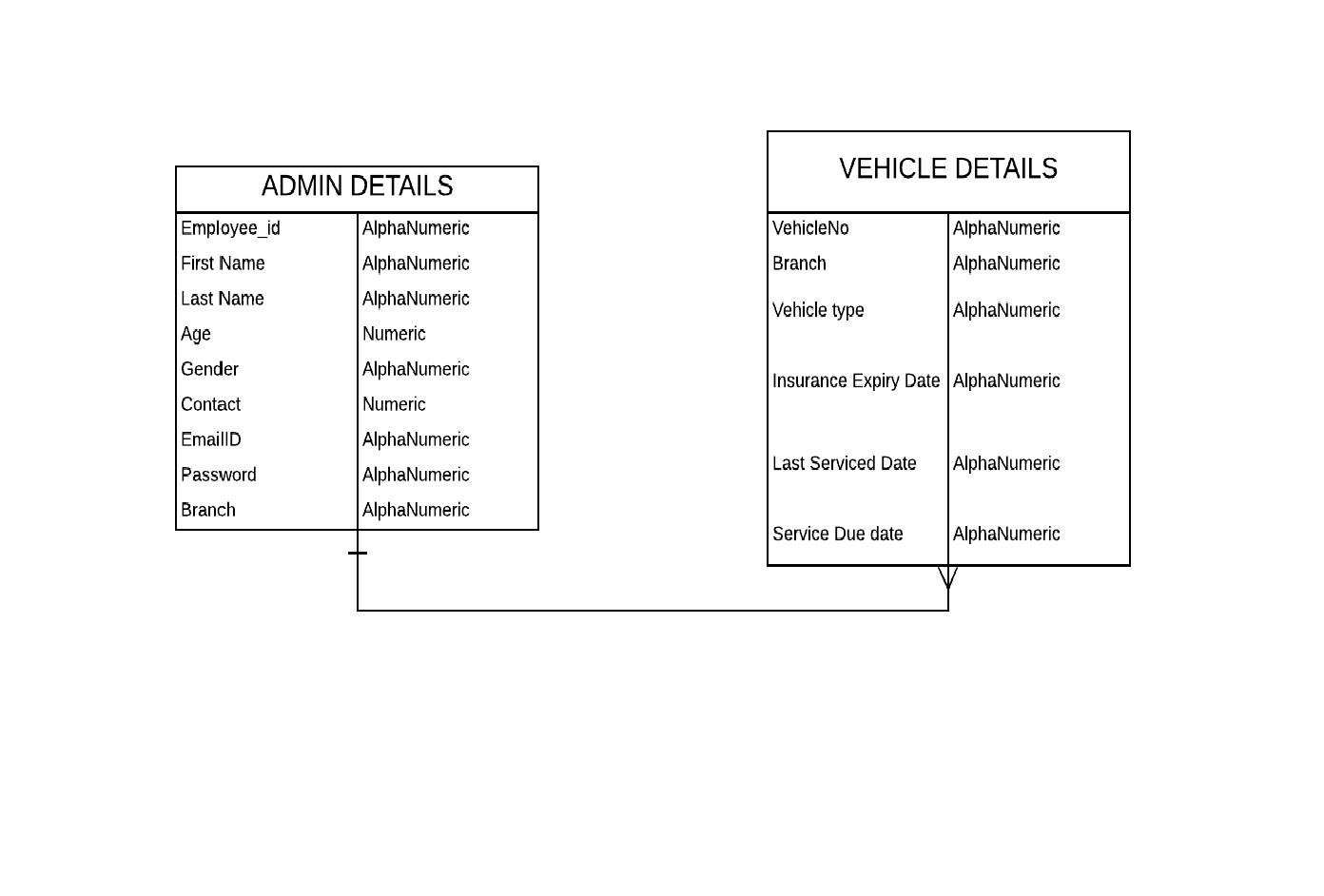
* ADMIN : Once the admin registered their details and if they entered into system they can be able to access all the vehicle details and can able to manipulate all the vehicle details that are present in the database server.
* DATABASE SERVER: When the server starts and admin logged in into the system, the database server will automatically trigger Email notification to admin to notify –Expiry of Insurance in next 15 days or Service due in next 15 months.

**4.3. ACTIVITY DIAGRAM**



Fig

**4.4 DATABASE DESIGN**



The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated whole. DBMS allow data to be protected and organized separately from other resources. Database is an integrated collection of data. The most significant form of data as seen by the programmers is data as stored on the direct access storage devices. This is the difference between logical and physical data.

Database files are the key source of information into the system. It is the process of designing database files, which are the key source of information to the system. The files should be properly designed and planned for collection, accumulation, editing and retrieving the required information.

The organization of data in database aims to achieve three major objectives: -

* Data integration.
* Data integrity.
* Data independence

The proposed system stores the information relevant for processing in the MS SQL SERVER 2000 database. This database contains tables, where each table corresponds to one particular type of information. Each piece of information in table is called a field or column. A table also contains records, which is a set of fields. All records in a table have the same set of fields with different information. There are primary key fields that uniquely identify a record in a table. There are also fields that contain primary key from another table called foreign keys.

Database design is the organization of data according to database model. The database

design consists of database tables

1. User Details table
2. Vehicle Details table

**4.5. OUTPUT DESIGN**

The output of vehicle reservation system consists of the output in favor to search and update vehicle details. User can able to create new vehicle details. The status of all the vehicles are viewed by the administrator.



**CHAPTER 5**

**5. TESTING**

**5.1. UNIT TESTING**

A Unit corresponds to a screen /form in the package. Unit testing focuses on verification of the corresponding class or Screen. This testing includes testing of control paths, interfaces, local data structures, logical decisions, boundary conditions, and error handling. Unit testing may use Test Drivers, which are control programs to co-ordinate test case inputs and outputs, and Test stubs, which replace low-level modules. A stub is a dummy subprogram.

**5.2. REGRESSION TESTING**

Each modification in software impacts unmodified areas, which results serious injuries to that software. So the process of re-testing for rectification of errors due to modification is known as regression testing.

**Installation and Delivery:**

Installation and Delivery is the process of delivering the developed and tested software to the customer. Refer the support procedures

**Acceptance and Project Closure:**

Acceptance is the part of the project by which the customer accepts the product. This will be done as per the Project Closure, once the customer accepts the product; closure of the project is started. This includes metrics collection, PCD, etc.

**5.3 VALIDATION**

Validation refers to the process of using the new software for the developed system in a live environment i.e., new software inside the organization, in order to find out the errors. This application is validated using java script. The validation phase reveals the failures and the bugs in the developed system. It will be come to know about the practical difficulties the system faces when operated in the true environment.

By testing the code of the implemented software, the logic of the program can be examined. A specification test is conducted to check whether the specifications stating the program are performing under various conditions. Apart from these tests, there are some special tests conducted which are given below:

Peak Load Tests: This determines whether the new system will handle the volume of activities when the system is at the peak of its processing demand. The test has revealed that the new software for the agency is capable of handling the demands at the peak time.

Storage Testing: This determines the capacity of the new system to store transaction data on a disk or on other files. The proposed software has the required storage space available, because of the use of a number of hard disks.

Performance Time Testing: This test determines the length of the time used by the system to process transaction data.

**CHAPTER 6**

**6. RESULTS AND DISCUSSIONS**

This project enables admin to have accurate data and to maintain each and every details securely. This work has been successfully completed and has been capable of maintaining all the admin and vehicle details. Furthermore, it would allow all the admins to have access to real-time reporting of events.

The solution developed will address the objective in a holistic manner and will have all the features and functionalities which shall let the admin be able to record & update the vehicle details along with their insurance details. The application will also run batch jobs to alert the admin on the expiry of the vehicle insurance to ensure the Vehicles moving out are 100% checked and approved.

**CHAPTER 7**

**7. CONCLUSION AND FUTURE WORK**

In this project, we addressed the problem of erroneous amount of vehicle details to be recorded manually. One problem is the existing work was, human errors and accurate results cannot be computed due to lack of knowledge at some unintentional cases. Thus our work provided an optimal solution that can provide a platform to proceed easy creation, updation, searching of vehicle details and automatic mail trigger when the insurance expiry of the vehicle falls in next 15 days or service due date in next 15 months. When this system proves to be flexible to the users and achieves to be user friendly. The future work may comprise of running the server fortnightly to trigger email notification and text message notification to all the registered admins. Future work concerns deeper analysis of particular mechanism, new proposals to try different methods or simply curiosity. For further better improvements it relies on user’s needs and technical advisor’s optimized level of thinking.

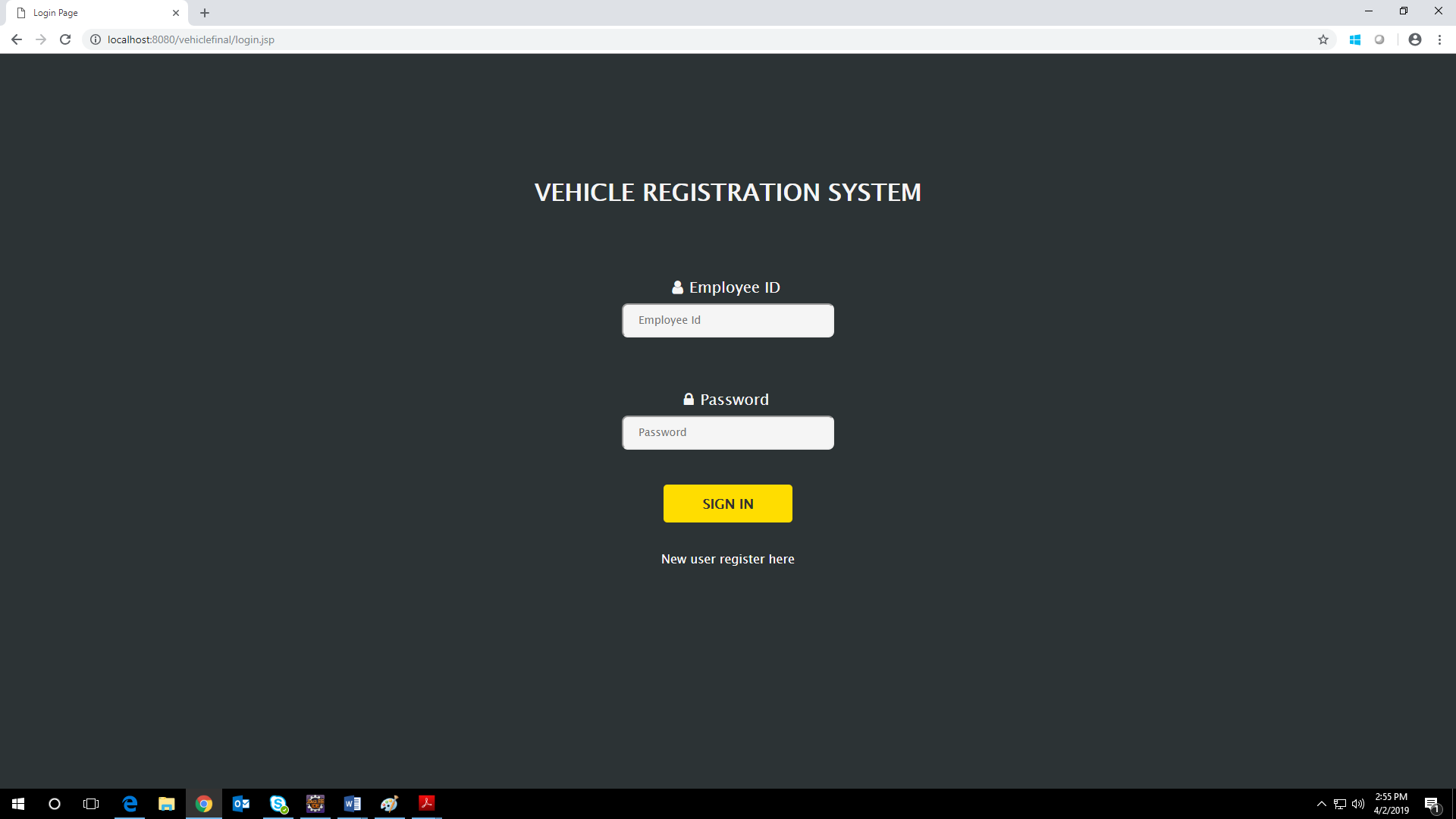
**CHAPTER 8**

**8. APPEDICES**

**8.2 SCREENSHOTS**

**8.2.1 LOGIN MODULE**

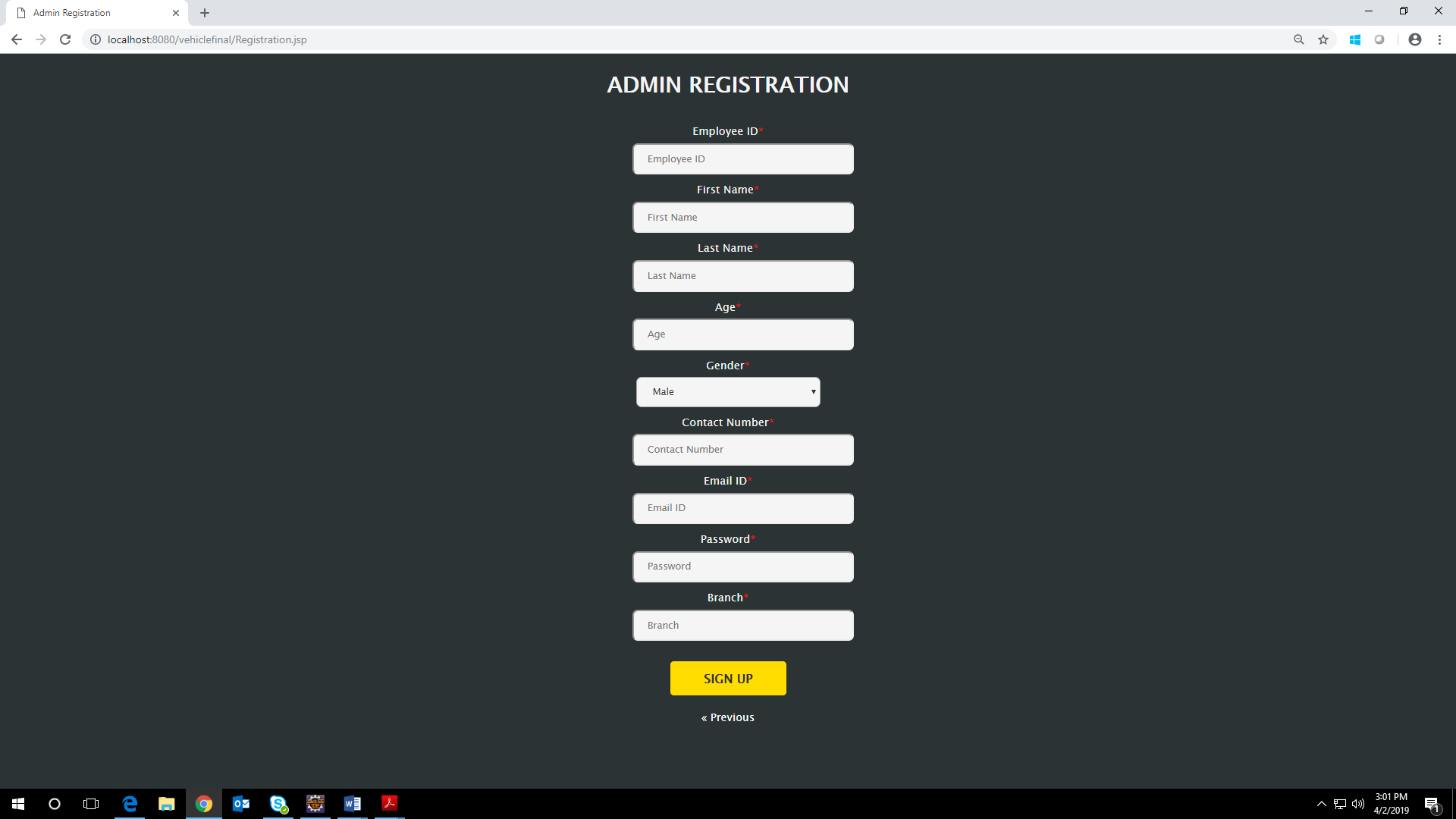
User can login with his/her employee id and password and check all the functionality from the menu..



**FIG 8.2.1 LOGIN MODULE**

**8.2.2 NEW USER REGISTRATION**

New user can be enrolled to the application by clicking signupbutton. User will be guided to fill registration form. User has to fill in all the mandatory field, not adhering to which lead to error message dialog to pop up. User can cancel the pop up by clicking okin the dialog box and fill in the required field. When user leaves User name field blank, error dialog pops up. If the user enters all the correct credentials and click submit button user will redirected to login page.

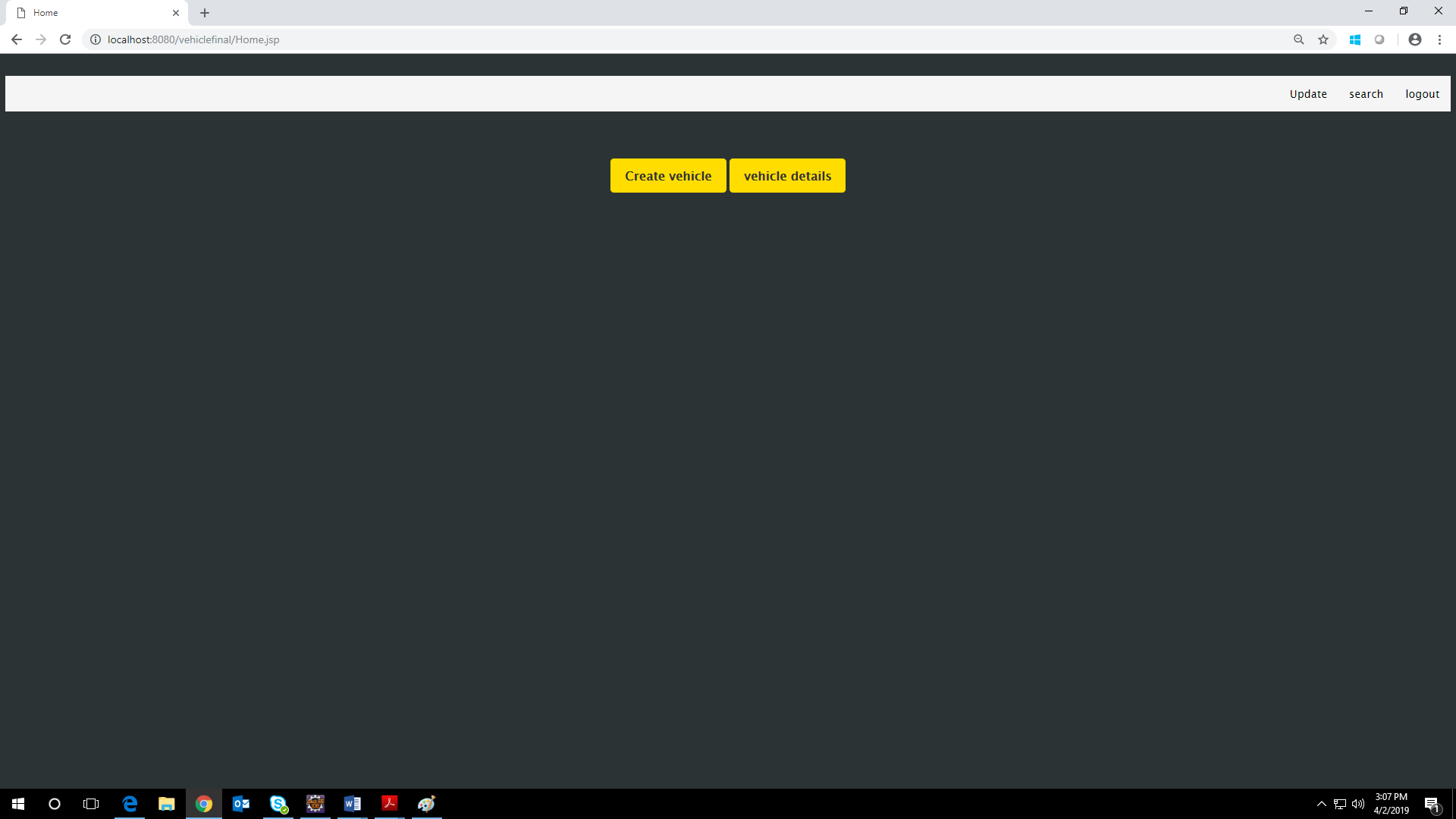
****

**FIG 8.2.2 ADMIN REGISTRATION**

**8.2.3 HOME PAGE**

It will consists of create new vehicle and display vehicle button. If the admin click’s

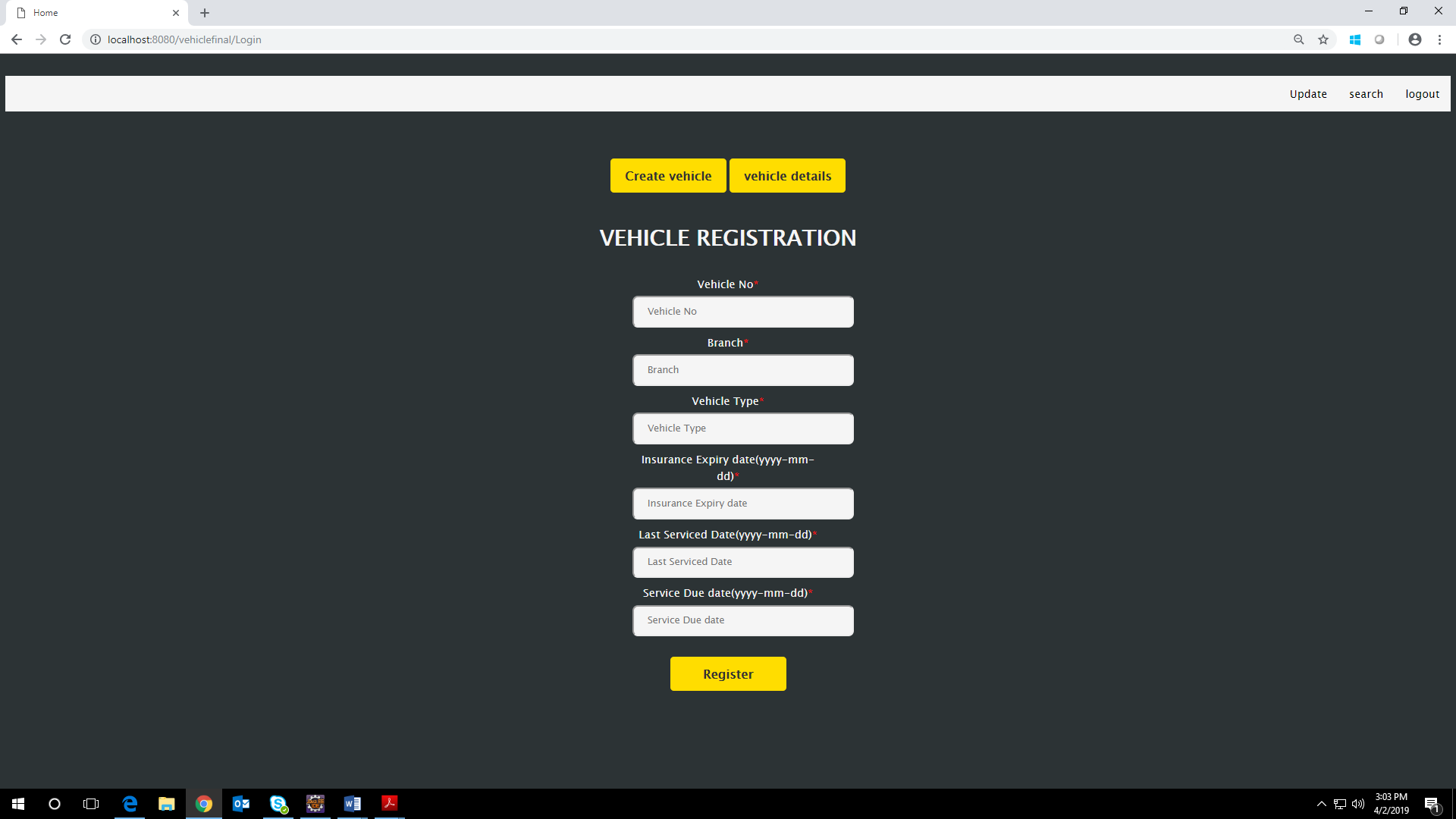
anyone of the button the appropriate pages will be displayed in the same page using ajax.

****

**FIG 8.2.3 HOME PAGE**

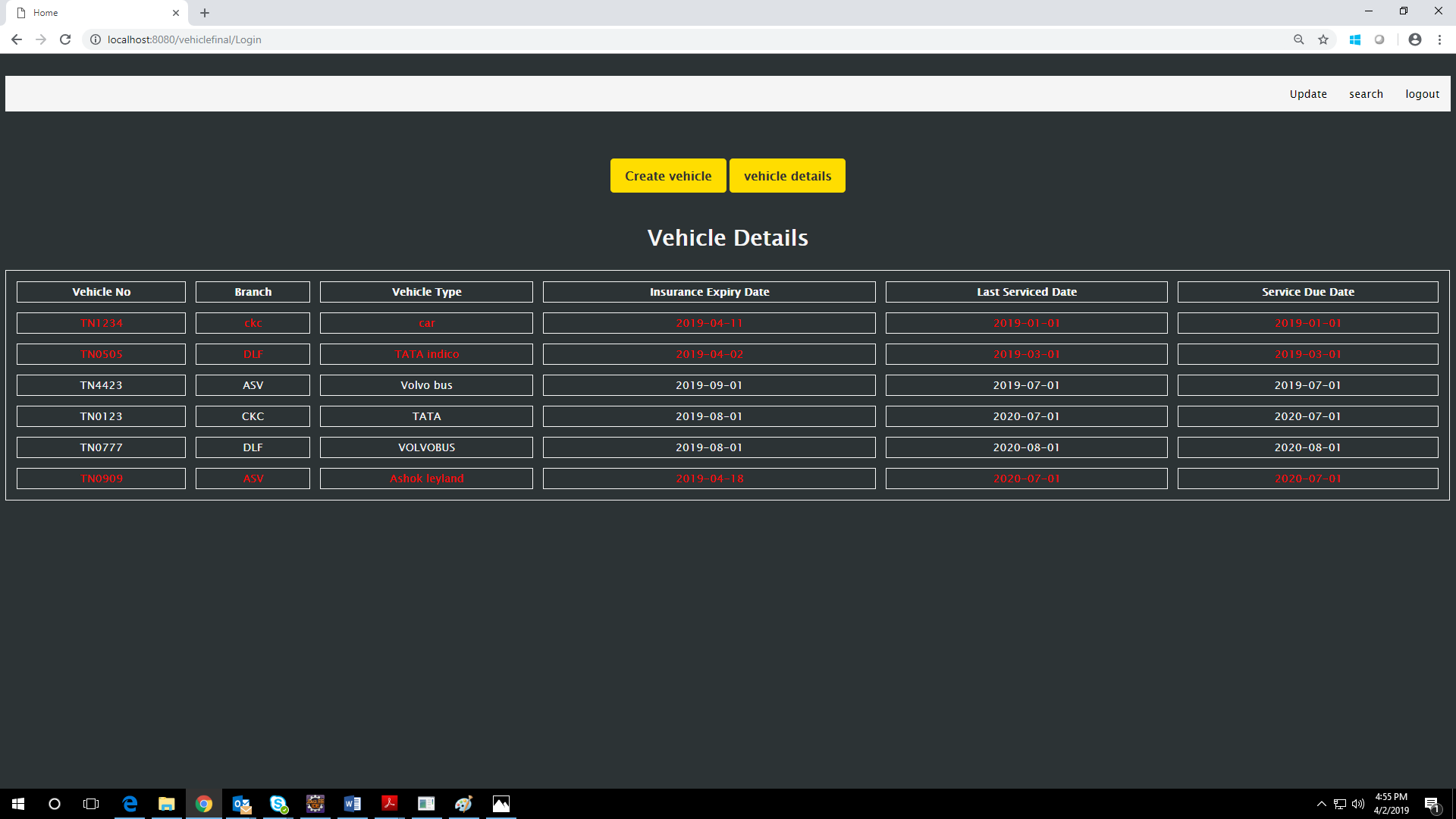
**8.2.4 CREATE NEW VEHICLE DETAILS**

In this page the admin can able to register new vehicle details.



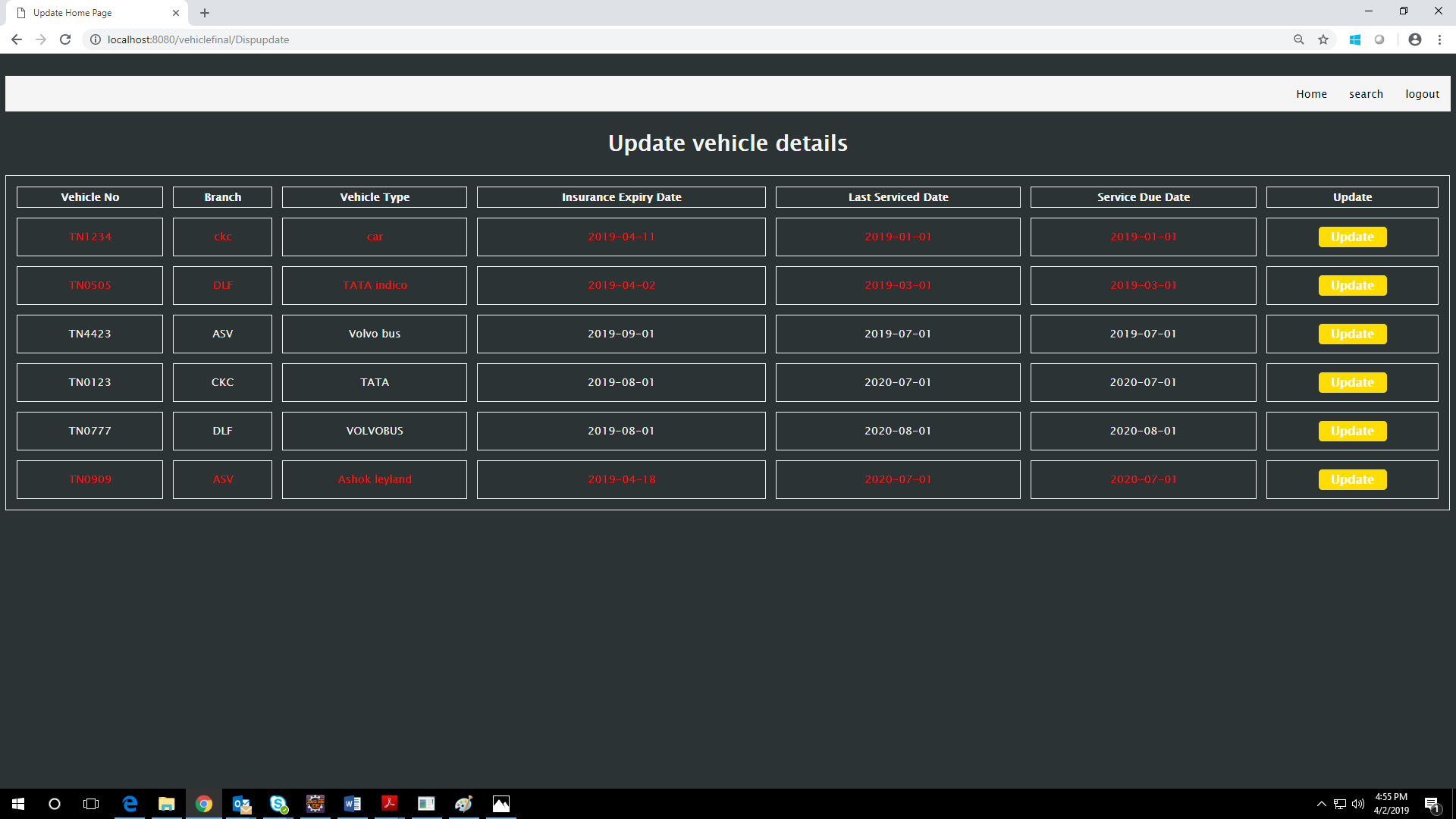
**FIG 8.2.4 CREATE NEW VEHICLE**

**8.2.5 DISPLAY VEHICLE DETAILS**



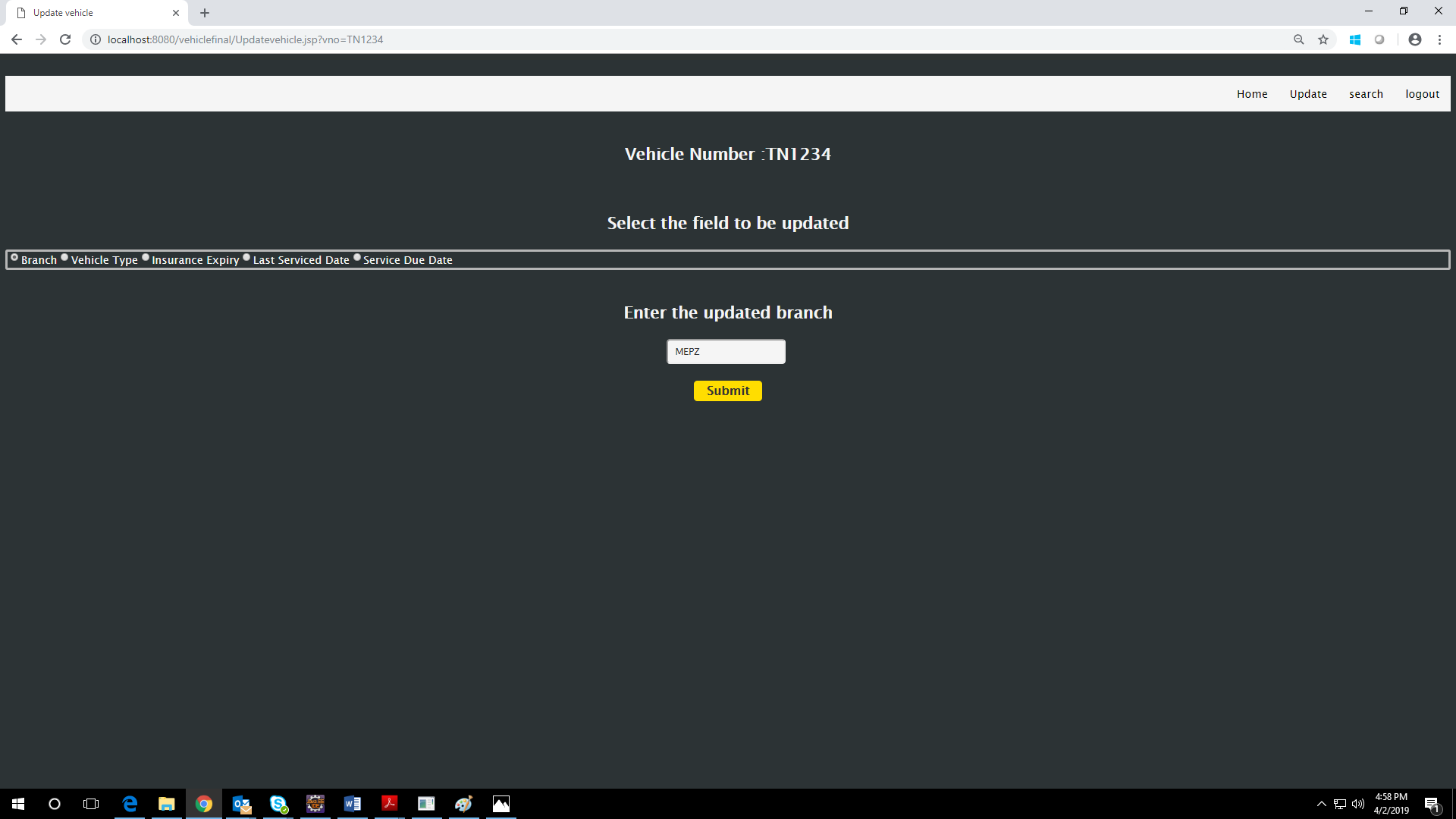
**FIG 8.2.5 DISPLAY VEHICLE DETAILS**

**8.2.6 UPDATE HOME PAGE**

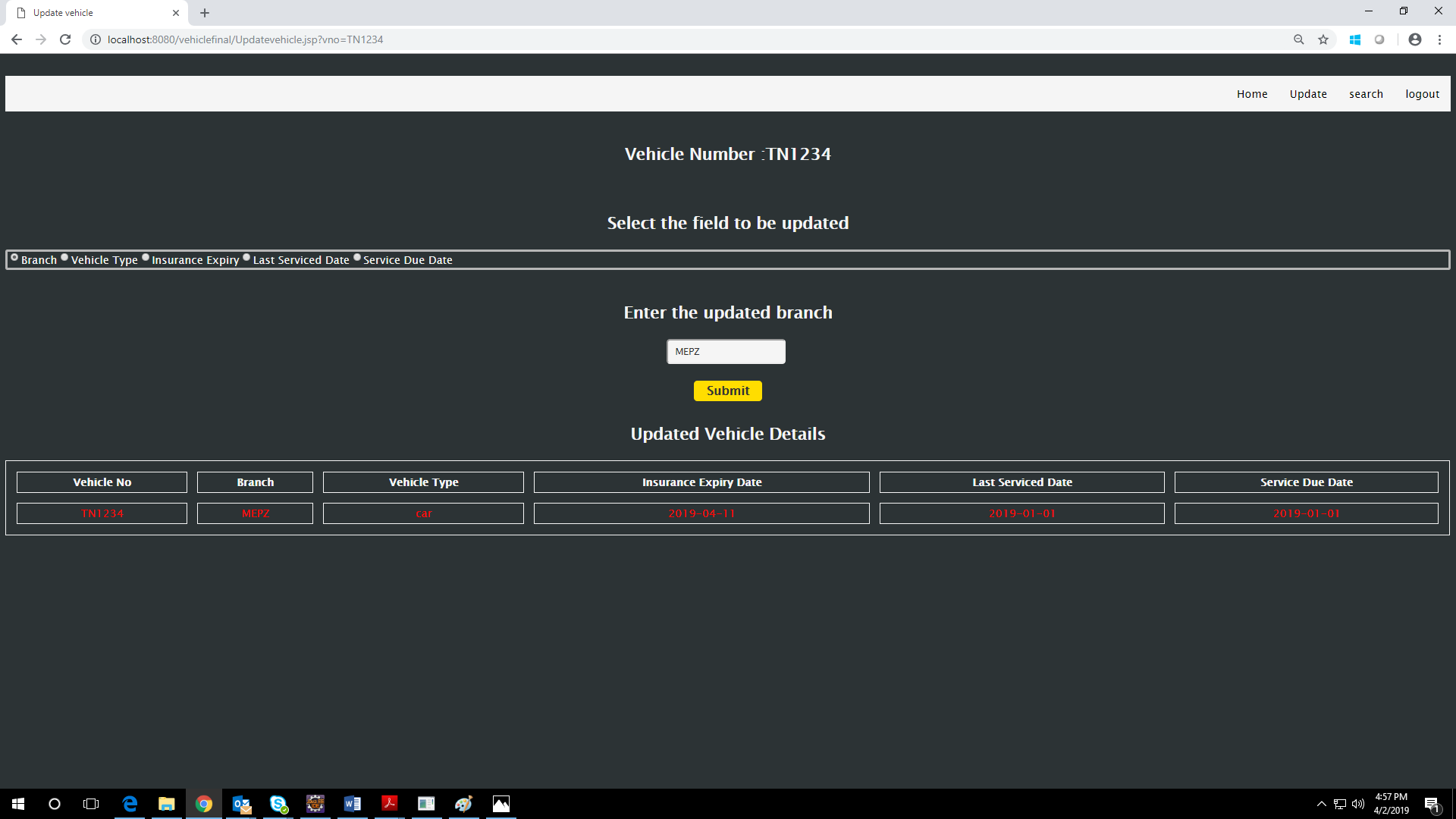


**FIG 8.2.6 UPDATE HOME PAGE**

**8.2.7 UPDATE VEHICLE DETAILS**

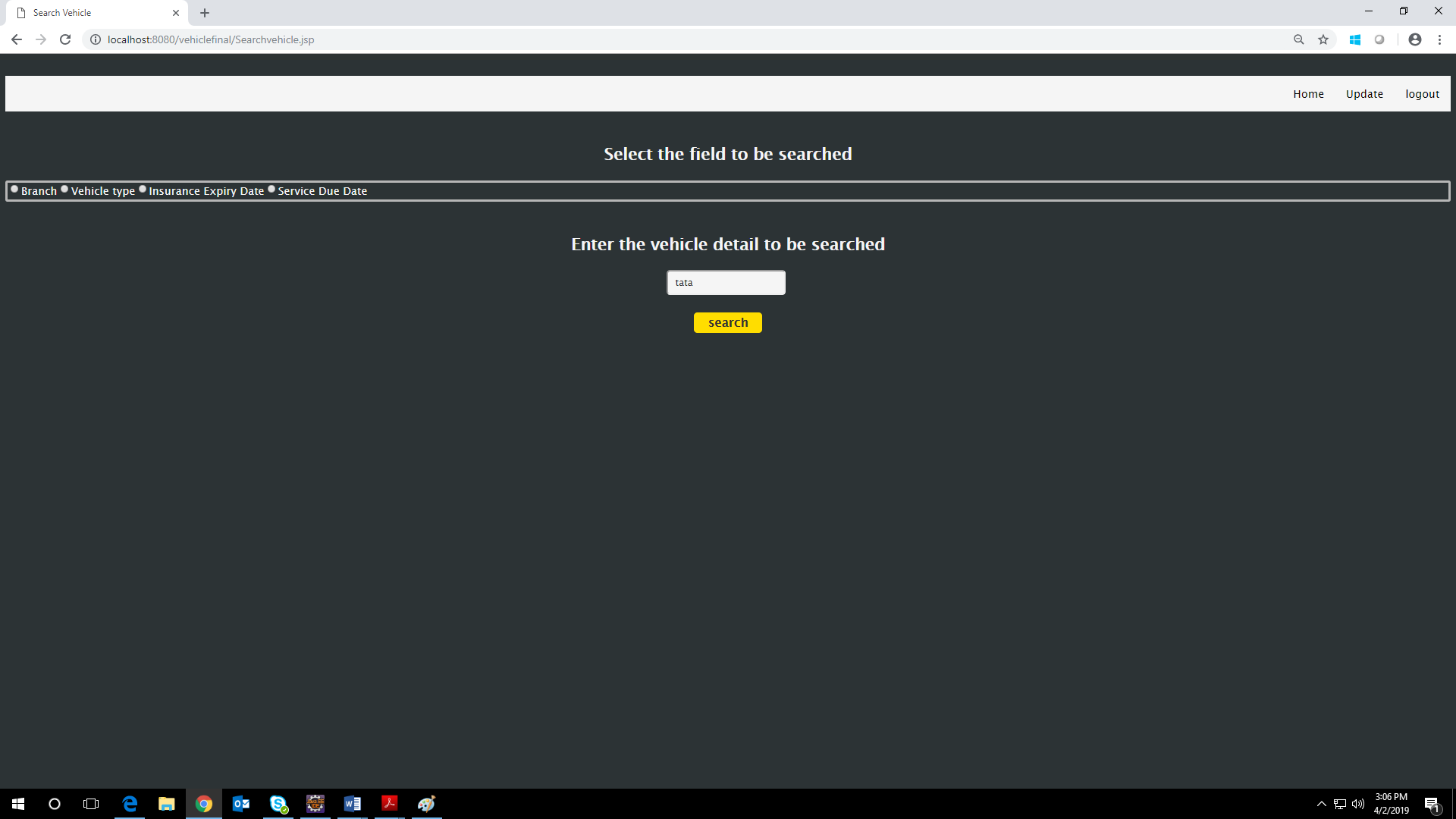


**FIG 8.2.7 UPDATE VEHICLE DETAIL**

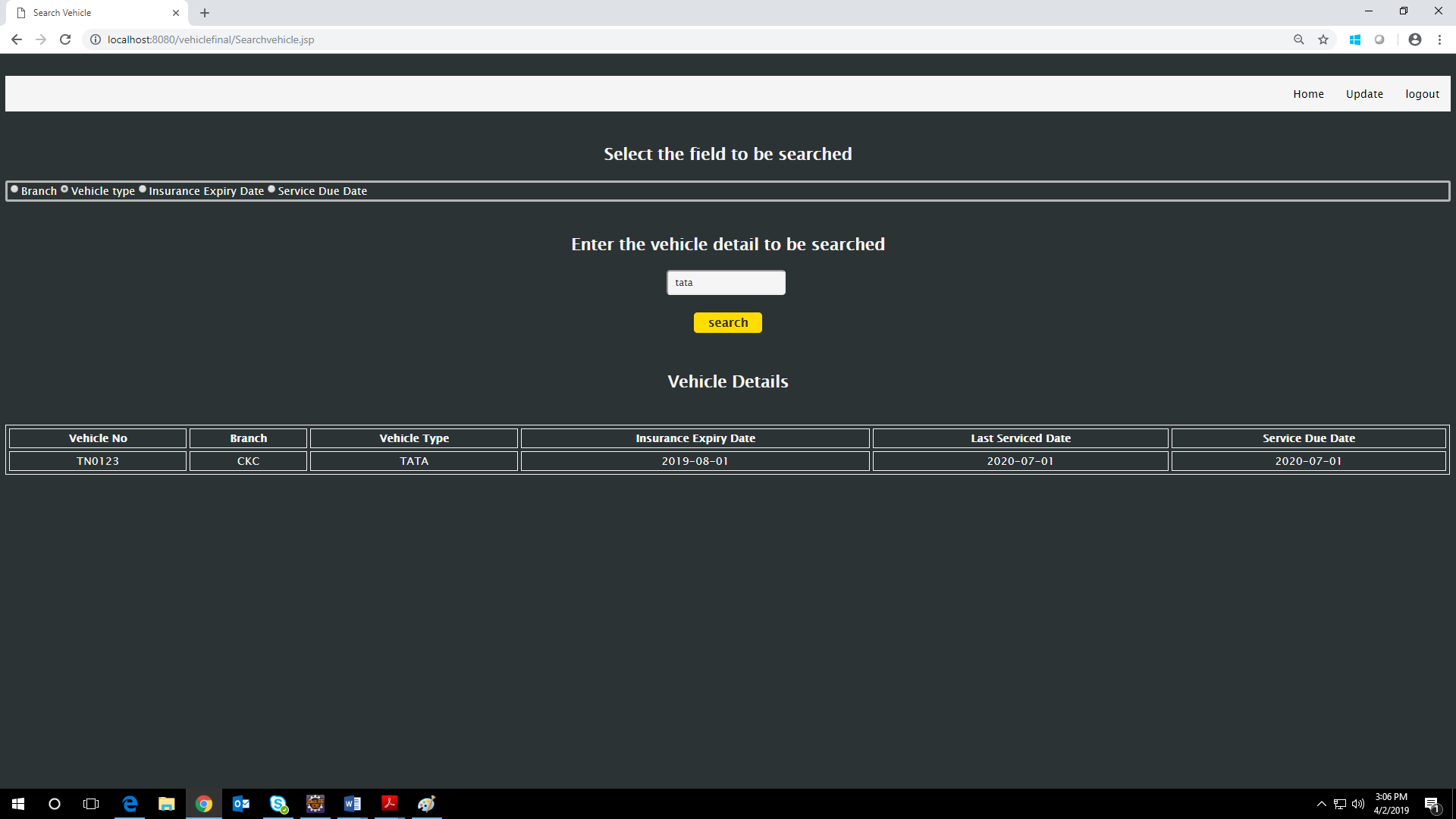
**8.2.8 UPDATED VEHICLE DETAILS** 

**FIG 8.2.8 UPDATED VEHICLE DETAILS**

**8.2.9 SEARCH VEHICLE DETAILS**

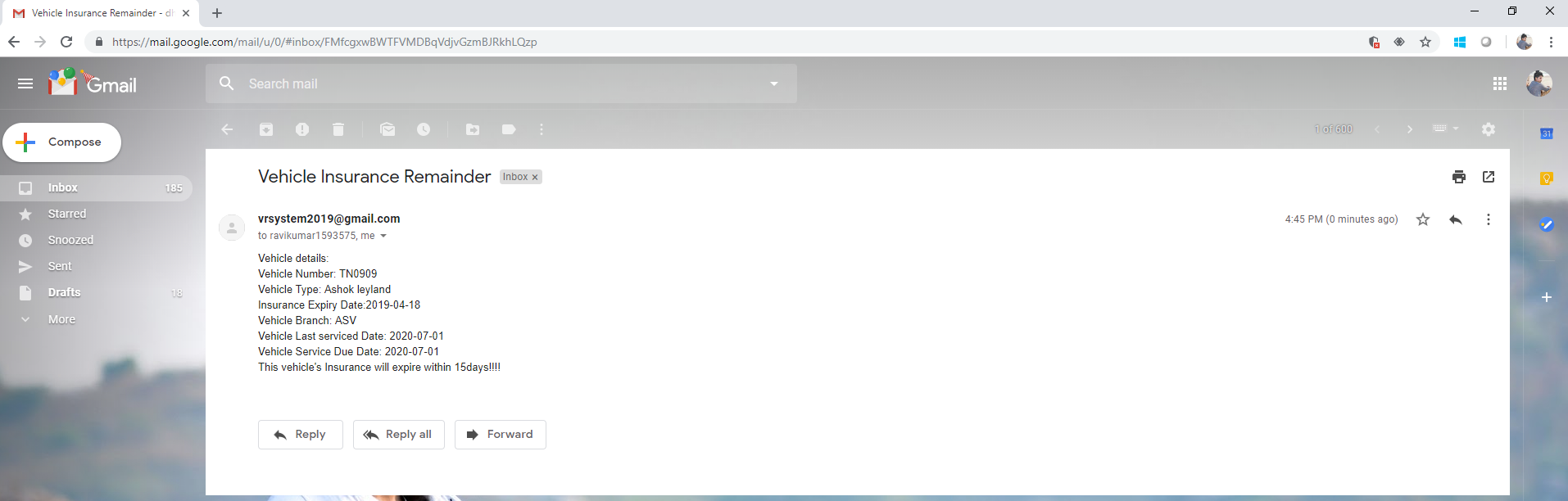


**FIG 8.2.9 SEARCH VEHICLE DETAILS**

**8.2.9 SEARCHED VEHICLE DETAILS** 

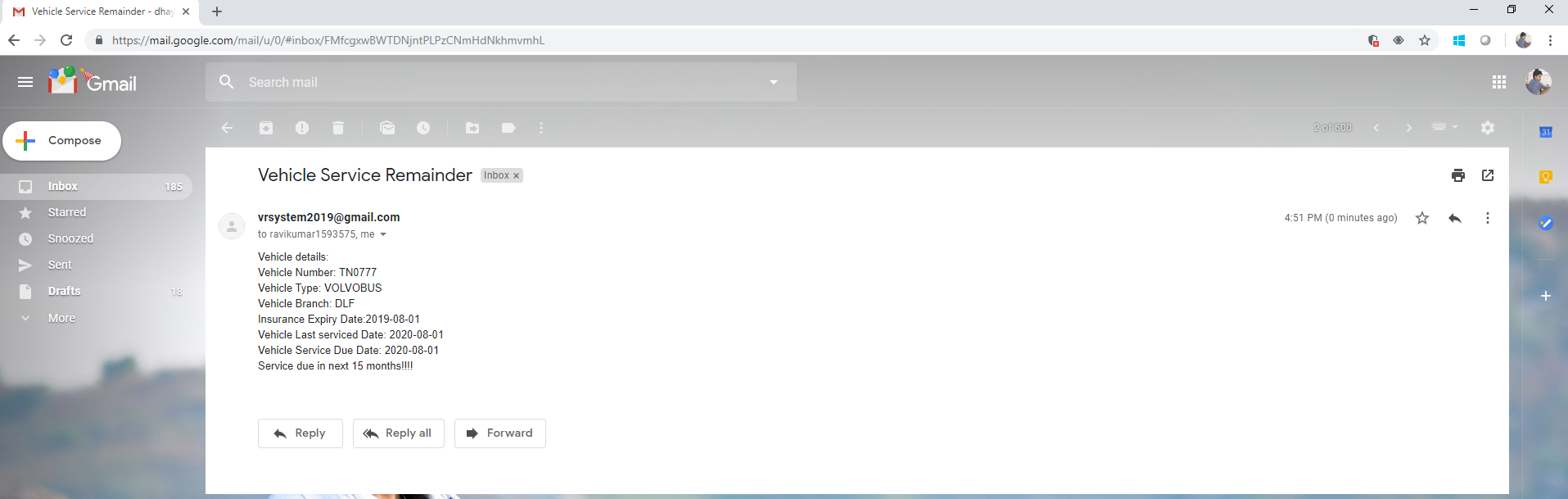
**FIG 8.2.9 SEARCHED VEHICLE DETAILS**

**8.2.10 INSURANCE REMAINDER MAIL**

****

**FIG 8.2.10 INSURANCE REMAINDER MAIL**

**8.2.11 VEHICLE SERVICE REMAINDER MAIL**

****

**FIG 8.2.11 VEHICLE SERVICE REMAINDER MAIL**

**CHAPTER 9**

**9. REFRENCES**

RESOURCE : <https://www.w3schools.com/>

STYLE SHEETS : <https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css>