­­­­­



INTERNATIONAL SCHOOL OF MANAGEMENT & TECHNOLOGY

|  |  |  |  |
| --- | --- | --- | --- |
| STUDENT DETAILS | | | |
|  | | | |
| Student ID |  | **Reg No.** |  |
| Family Name | yadav | **Given Name** | Rupesh kumar |
| Enrolment Year | 2022 | **Section** | c |
| Semester | I | **Email** | Kif22.rky@ismt.edu.np |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UNIT DETAILS | | | | |
| Unit Title | **Database design & development** | **Unit Code** | H/615/1622 | |
| Assessor Name | Bikul raj Koirala | **Issued Date** | 3 Nov,2022 | |
| Assignment Title | **Toyota vehicle management system** | | | |
| Assignment No | 01 | **Submission Date** | | 31 Dec,2022 |
| Qualification |  | **Campus** | | **ISMT** |





**STUDENT ASSESSMENT SUBMISSION AND DECLARATION**

When submitting evidence for assessment, each student must sign a declaration confirming that the work is their own.

|  |  |  |  |
| --- | --- | --- | --- |
| Student Name | Rupesh kumar yadav | Assessor Name | Bikul raj Koirala |
| Issue Date | 3 Nov,2022 | Submission Date | 31 Dec,2022 |
| Programme | BTEC HND in Computing | | |
| Unit Name | Database Design & Development | | |
| Assignment Title | Toyota vehicle management system | | |

**Plagiarism**

Plagiarism is a particular form of cheating. Plagiarism must be avoided at all costs and students who break the rules, however innocently, may be penalized. It is your responsibility to ensure that you understand correct referencing practices. As a university level student, you are expected to use appropriate references throughout and keep carefully detailed notes of all your sources of materials for material you have used in your work, including any material downloaded from the Internet. Please consult the relevant unit lecturer or your course tutor if you need any further advice.

**Student Declaration**

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

Student signature: Date:

 Pearson Education 2018  
 Higher Education Qualification

Table of Contents

[Introduction 10](#_Toc123478381)

[Tools used for the database development 10](#_Toc123478382)

[Visual Studio Code 10](#_Toc123478383)

[XAMPP 11](#_Toc123478384)

[MYSQL 12](#_Toc123478385)

[DATABASE MANAGEMENT SYSTEM 12](#_Toc123478386)

[Types Of database 13](#_Toc123478387)

[Hierarchical database 13](#_Toc123478388)

[Network database 13](#_Toc123478389)

[Relational database 13](#_Toc123478390)

[Object-Oriented Database 13](#_Toc123478391)

[An Entity Relationship Diagram (ER Diagram) 14](#_Toc123478392)

[Entity: 15](#_Toc123478393)

[Relationship 17](#_Toc123478394)

[Recursive relationship: 17](#_Toc123478395)

[Attribute 17](#_Toc123478396)

[Schema diagram 18](#_Toc123478397)

[Schema diagram in my scenario: 19](#_Toc123478398)

[Data Dictionary 19](#_Toc123478399)

[Data dictionary in my Scenario. 20](#_Toc123478400)

[Data dictionary of vehicle. 20](#_Toc123478401)

[Data dictionary of branch 20](#_Toc123478402)

[Data dictionary of color 20](#_Toc123478403)

[Data dictionary of users 21](#_Toc123478404)

[Data dictionary of customer 21](#_Toc123478405)

[Data validation 22](#_Toc123478406)

[Significant of Data validation 22](#_Toc123478407)

[Normalization: 22](#_Toc123478408)

[First Normal Form(1NF) 23](#_Toc123478409)

[alex@email.com 23](#_Toc123478410)

[shomesh@email.com 23](#_Toc123478411)

[milan@email.com 23](#_Toc123478412)

[richa@email.com 23](#_Toc123478413)

[bibek@email.com 23](#_Toc123478414)

[alex@email.com 24](#_Toc123478415)

[alex@email.com 24](#_Toc123478416)

[shomesh@email.com 24](#_Toc123478417)

[milan@email.com 24](#_Toc123478418)

[milan@email.com 24](#_Toc123478419)

[richa@email.com 24](#_Toc123478420)

[bibek@email.com 24](#_Toc123478421)

[bibel@email.com 24](#_Toc123478422)

[Second Normal form(2NF) 24](#_Toc123478423)

[alex@email.com 24](#_Toc123478424)

[milan@email.com 25](#_Toc123478425)

[richa@email.com 25](#_Toc123478426)

[bibek@email.com 25](#_Toc123478427)

[Third Normal Form(3NF) 26](#_Toc123478428)

[Output design of Toyota management system: 26](#_Toc123478429)

[Effectiveness of design: 28](#_Toc123478430)

[User requirement and system requirement: 28](#_Toc123478431)

[Conclusion 29](#_Toc123478432)

[P2 29](#_Toc123478433)

[Developing the database system 29](#_Toc123478434)

[Introduction: 29](#_Toc123478435)

[Types of SQL Constraints: 29](#_Toc123478436)

[Primary Key: 29](#_Toc123478437)

[Foreign Key: 30](#_Toc123478438)

[Unique key: 30](#_Toc123478439)

[Not Null: 30](#_Toc123478440)

[Relationships: 30](#_Toc123478441)

[Joins: 32](#_Toc123478442)

[User interface: 33](#_Toc123478443)

[Dashboard: 33](#_Toc123478444)

[LOGIN: 33](#_Toc123478445)

[HOME: 34](#_Toc123478446)

[Customers: 34](#_Toc123478447)

[Customer entry: 35](#_Toc123478448)

[New customer output: 36](#_Toc123478449)

[Update customer: 36](#_Toc123478450)

[Update customer output: 37](#_Toc123478451)

[Delete customer: 37](#_Toc123478452)

[Delete customer output: 38](#_Toc123478453)

[Branch: 38](#_Toc123478454)

[Branch entry: 39](#_Toc123478455)

[Branch entry output: 39](#_Toc123478456)

[Branch update 40](#_Toc123478457)

[Branch update output: 41](#_Toc123478458)

[Delete branch: 41](#_Toc123478459)

[Delete branch output: 42](#_Toc123478460)

[Vehicle: 43](#_Toc123478461)

[Vehicle entry: 43](#_Toc123478462)

[Vehicle entry output: 43](#_Toc123478463)

[Vehicle update: 44](#_Toc123478464)

[Vehicle update output: 45](#_Toc123478465)

[Vehicle delete: 45](#_Toc123478466)

[Vehicle delete output: 46](#_Toc123478467)

[Data Validation: 46](#_Toc123478468)

[Conclusion: 48](#_Toc123478469)

[P3 48](#_Toc123478470)

[Introduction 48](#_Toc123478471)

[DDL (Data Definition Language): 48](#_Toc123478472)

[DML (Data Manipulation Language) 49](#_Toc123478473)

[Create database: 49](#_Toc123478474)

[Create table (customer): 49](#_Toc123478475)

[Insert : 49](#_Toc123478476)

[Update: 50](#_Toc123478477)

[Select: 50](#_Toc123478478)

[Delete: 50](#_Toc123478479)

[Create table (Branch): 50](#_Toc123478480)

[Insert 50](#_Toc123478481)

[Update: 51](#_Toc123478482)

[Select: 51](#_Toc123478483)

[Delete: 51](#_Toc123478484)

[Create table (vehicles): 51](#_Toc123478485)

[Insert: 52](#_Toc123478486)

[Update: 52](#_Toc123478487)

[Select: 52](#_Toc123478488)

[Delete: 52](#_Toc123478489)

[Create table (colors) 53](#_Toc123478490)

[Insert: 53](#_Toc123478491)

[Select: 53](#_Toc123478492)

[Drop table: 53](#_Toc123478493)

[Truncate table: 53](#_Toc123478494)

[Alter statement: 54](#_Toc123478495)

[Conclusion: 54](#_Toc123478496)

[P4: 54](#_Toc123478497)

[Introduction: 54](#_Toc123478498)

[Testing: 55](#_Toc123478499)

[Software testing: 55](#_Toc123478500)

[Database testing: 55](#_Toc123478501)

[Testing Methodologies: 56](#_Toc123478502)

[Black box testing: 56](#_Toc123478503)

[White box testing: 56](#_Toc123478504)

[Types of testing: 56](#_Toc123478505)

[Unit testing: 56](#_Toc123478506)

[Integration Testing: 58](#_Toc123478507)

[Stress testing: 58](#_Toc123478508)

[Review: 59](#_Toc123478509)

[Software Review: 59](#_Toc123478510)

[Conclusion: 59](#_Toc123478511)

[P5: 59](#_Toc123478512)

[Introduction: 59](#_Toc123478513)

[User documentation: 59](#_Toc123478514)

[Constraint/Risks: 60](#_Toc123478515)

[Assumption: 60](#_Toc123478516)

[User Manual: 60](#_Toc123478517)

[Login dashboard: 60](#_Toc123478518)

[Home dashboard: 61](#_Toc123478519)

[Customer dashboard: 61](#_Toc123478520)

[Branch dashboard: 62](#_Toc123478521)

[Vehicle dashboard: 62](#_Toc123478522)

[Conclusion: 63](#_Toc123478523)

[References: 63](#_Toc123478524)

**Part 1**

# Introduction

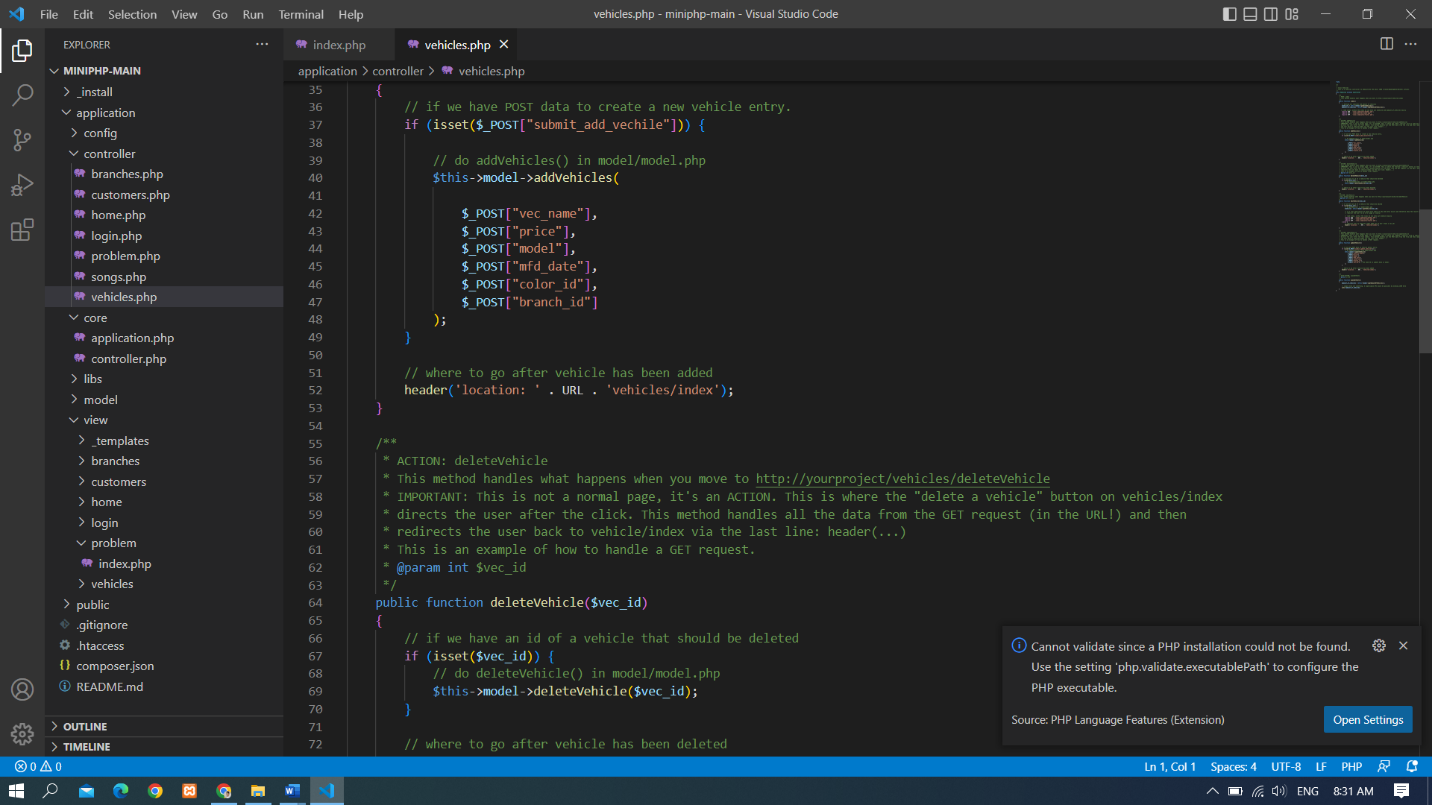
A database is a group of connected data that has a regular structure and is organized so that a computer can quickly identify the needed data. A database may store, modify, and extract data with the use of a group of tools called the Database Management System (DBMS). In addition to storing data in the form of tables and the relationships between those tables, the Relational Database Management System (RDBMS) is a database management system. Database diagrams, entity connection diagrams, and data dictionaries are made to show the link between tables, constraints, and the kind and length of the data stored in them. Normalization is the process of removing duplication from a relationship or collection of relationships. Validation of data is a technique for testing the accuracy and consistency of the data. To design and build a database, I used all the above methods.

# Tools used for the database development

As a database developer I used some tools for the maintain and manage database. I used tools desktop application (VS code, XAMPP), programming language (MYSQL, PHP,CSS). Every company requires a well-designed database management tool. Finding a well-organized, functional system that allows team to easily perform data-related tasks is critical to our company's success. There are numerous tools, frameworks, and platforms available to assist our company in managing its data, and selecting the best one for our needs can be difficult. To assist us, we asked Bikul Raj Koirala which database management tools is easy to use as well as beneficial to our company. Here's what they had to say about it.

## Visual Studio Code

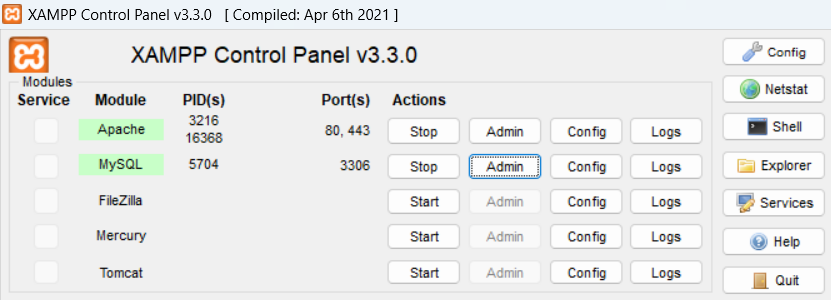
Visual Studio Code (also known as VS Code) is a free opensource text editor developed by Microsoft. Although the editor is lightweight, it includes some powerful features that have helped VS Code become one of the most popular development environment tools in recent years. It has Connections to most database types, excellent intellisense, incredibly simple formatting, cross-platform support, customizable layouts, and can built-in Git! We used VS code for Edit the program as well as insert data. (Anon., 2021) Visual studio code looks like given screenshoot.



*Screenshot 1: visual studio code*

## XAMPP

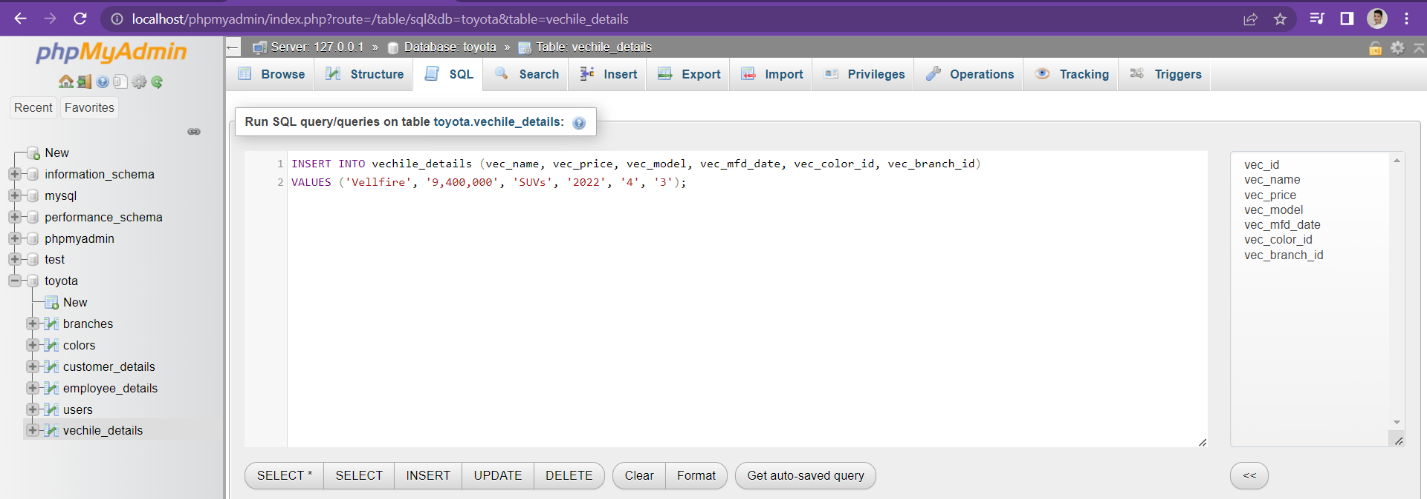
XAMPP is an open-source web solution package that includes the Apache distribution for many servers and command-line executables, as well as modules like Apache server, MariaDB, PHP, and Perl. XAMPP is an acronym that stands for Cross-Platform, Apache, MySQL, and PHP and Perl. Before releasing a website or client to the main server, a local host or server can use XAMPP to test it on computers and laptops. It is a best platform that provides a suitable environment for testing and verifying the operation of projects based on Apache, Perl, MySQL databases, and PHP via the host's system. (educba , September 09, 2021) Xampp looks like given screenshot.



*Screenshot 2: xampp*

## MYSQL

MySQL is in the XAMPP which is one of the most useful database management tools and this open-source relational database is simple to use and is already integrated into a number of popular web applications such as WordPress. MySQL is used by major corporations such as Google, LinkedIn, YouTube, PayPal, and Twitter, etc. If we only have time to learn one database, MySQL is a good choice due to its popularity and easy to use. (Anon., n.d.)

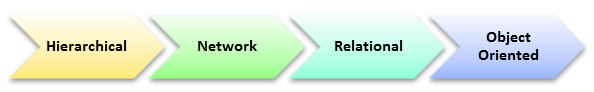


Screenshot 3: SQL

# DATABASE MANAGEMENT SYSTEM

A database management system (DBMS) is software that stores and retrieves data for users while taking appropriate security precautions. It is made up of a collection of programs that manipulate the database. The DBMS accepts data requests from applications and instructs the operating system to provide the requested data. A database management system (DBMS) assists users and third-party software in storing and retrieving data in large systems. DBMS allows users to design their own databases based on their specific needs. The term "DBMS" refers to both the database user and other application programs. It acts as a bridge between the data and the software application. We must define the different types of data elements to be stored in each record in order to specify the structure of the records in each file. A coding scheme can also be used to represent the values of a data item. There are different types of database but we mostly used relational database (RDBMS). (Anon., June 17, 2022) Here I discussed about types of database.

## Types Of database



The main Four Types of Database Management Systems are:

1. Hierarchical database
2. Network database
3. Relational database
4. Object-Oriented database

## Hierarchical database

Model data is organized in a tree-like structure in a Hierarchical database. Data is organized hierarchically (top-down or bottom-up). A parent-child relationship is used to represent the data. Parents in Hierarchical DBMS can have many children, but children only have one parent.

## Network database

Each child can have multiple parents thanks to the network database model. It enables you to address the need to model more complex relationships, such as the many-to-many relationship between orders and parts. Entities in this model are organized in a graph that can be accessed via several paths.

## Relational database

A relational database is a type of database that stores and makes data points related to one another available. Relational databases are based on the relational model, which is a simple and intuitive way of representing data in tables. Each row in a relational database is a record with a unique ID called the key. The columns of the table hold data attributes, and each record typically has a value for each attribute, making it simple to establish relationships between data points.

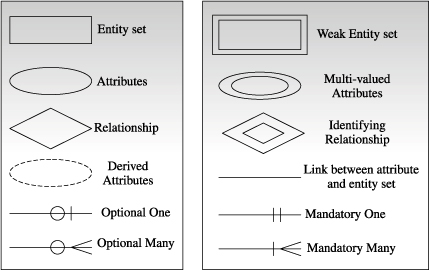
## Object-Oriented Database

Objects are used to store data in the Object-oriented Model. The structure is known as classes, and it contains data. It is one of the DBMS components that defines a database as a collection of objects that stores both the values and operations of data members.

# An **Entity Relationship Diagram (ER Diagram)**

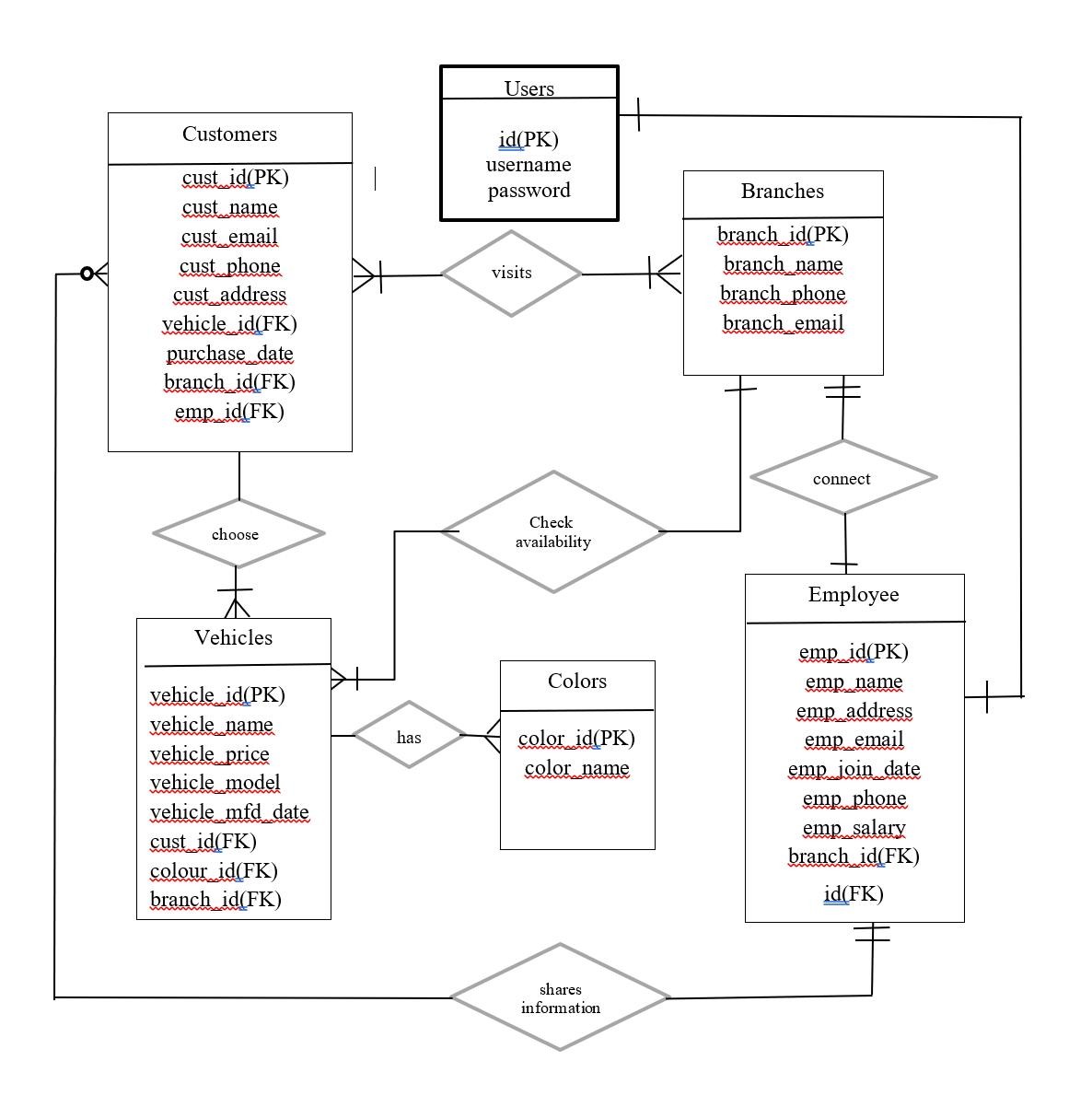
Entity Relationship (ER) Diagram is a type of flowchart that depicts how "entities" such as people, objects, or concepts interact within a system. ER Diagrams are most commonly used in software engineering, business information systems, education, and research to design or debug relational databases. They are also known as ERDs or ER Models because they use a predefined set of symbols such as rectangles, diamonds, ovals, and connecting lines to depict the interconnectedness of entities, relationships, and their attributes. They mirror grammatical structure, with entities acting as nouns and relationships acting as verbs. (Anon., n.d.)

Symbols used in ER- Diagram



*Figure 1: symbols used in ER- diagram*

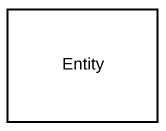
**ER diagram in my project.**



*Figure 2: ER-diagram*

# Entity:

A definable thing such as a customer, branch, concept or event—that can have data stored about it. Think of entities as nouns. (Anon., n.d.) Examples: a customer, branch, vehicles. Typically shown as a rectangle.

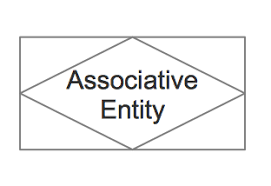


*Figure 3: entity*

Entity type**:**A group of definable things, such as students or athletes, whereas the entity would be the specific student or athlete. Other examples: customers, cars or products.

Entity set**:** Same as an entity type, but defined at a particular point in time, such as students enrolled in a class on the first day. Other examples: Customers who purchased last month, cars currently registered in Florida. A related term is instance, in which the specific person or car would be an instance of the entity set.

Entity categories**:** Entities are categorized as strong, weak or associative. A strong entity can be defined solely by its own attributes, while a weak entity cannot. An associative entity associates entities (or elements) within an entity set.



*Figure 4 : Weak entity and Associative entity*

Entity keys:  Refers to an attribute that uniquely defines an entity in an entity set. Entity keys can be super, candidate or primary. **Super key:**A set of attributes (one or more) that together define an entity in an entity set. **Candidate key:**A minimal super key, meaning it has the least possible number of attributes to still be a super key. An entity set may have more than one candidate key. **Primary key:**A candidate key chosen by the database designer to uniquely identify the entity set. **Foreign key:**Identifies the relationship between entities.

# Relationship

How entities act upon each other or are associated with each other. Think of relationships as verbs. For example, the named student might register for a course. The two entities would be the student and the course, and the relationship depicted is the act of enrolling, connecting the two entities in that way. Relationships are typically shown as diamonds or labels directly on the connecting lines (Anon., n.d.)

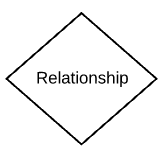
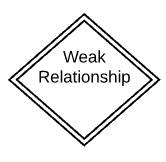
.  ****

Figure 6: Relationship and weak relationship

## ****Recursive relationship:****

The same entity participates more than once in the relationship.

# Attribute

A property or characteristic of an entity. Often shown as an oval or circle. (Anon., n.d.)

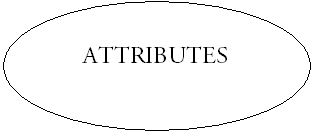


Figure 7: Attributes

Descriptive attribute:A property or characteristic of a relationship (versus of an entity.)

Attribute categories:Attributes are categorized as simple, composite, derived, as well as single-value or multi-value. Simple: Means the attribute value is atomic and can’t be further divided, such as a phone number.

Composite: Sub-attributes spring from an attribute. Derived: Attributed is calculated or otherwise derived from another attribute, such as age from a birthdate.

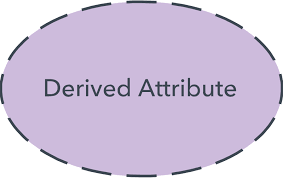
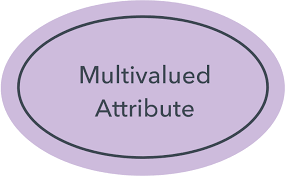


Figure 8: derived attribute

Multi-value:More than one attribute value is denoted, such as multiple phone numbers for a person.



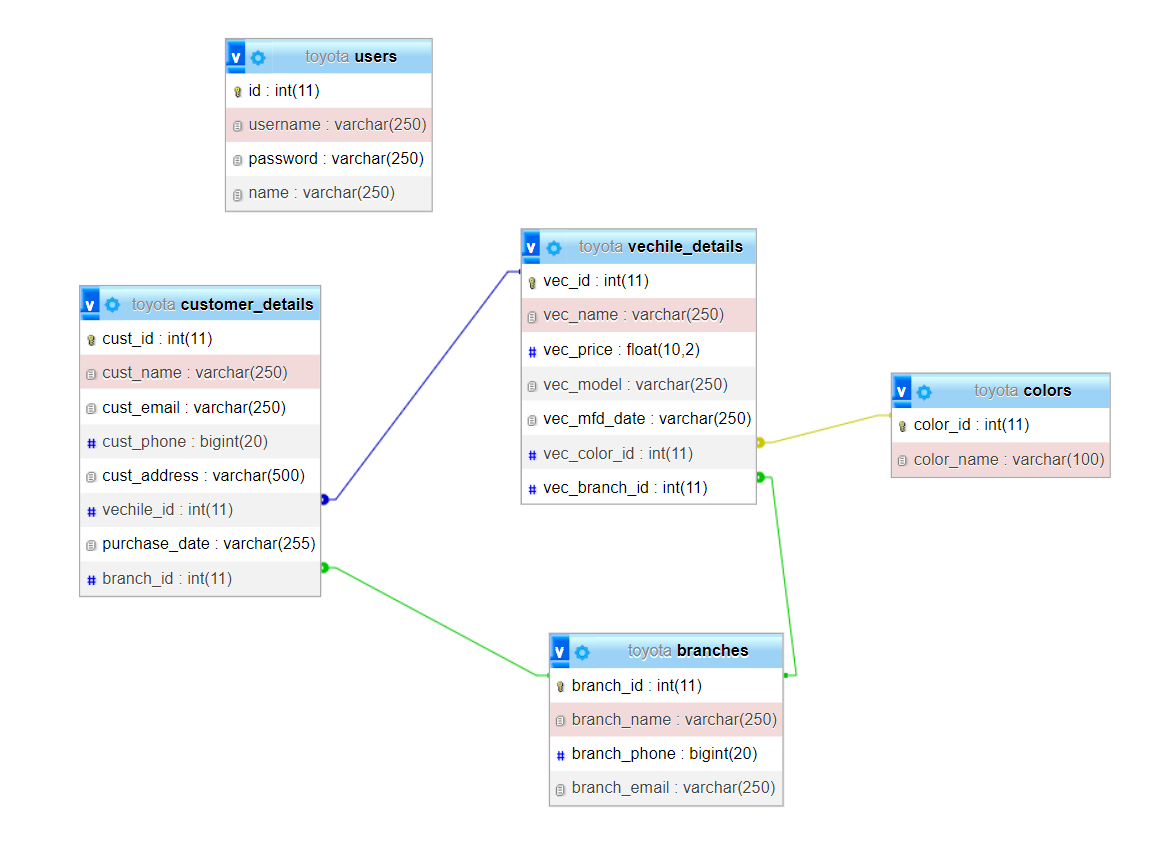
*Figure 9: multivalued attribute*

Single-value: Just one attribute value. The types can be combined, such as: simple single-value attributes or composite multi-value attributes.

# Schema diagram

“In database terms, a schema (pronounced “skee-muh” or “skee-mah”) is the organization and structure of a database. Both schemas and schemata can be used as plural forms. A schema contains schema objects, which could be tables, columns, data types, views, stored procedures, relationships, primary keys, foreign keys, etc. database schema can be represented in a visual diagram, which shows the database objects and their relationship with each other.” (IAN, 2016).

# Schema diagram in my scenario:



*Figure 10: schema diagram*

# Data Dictionary

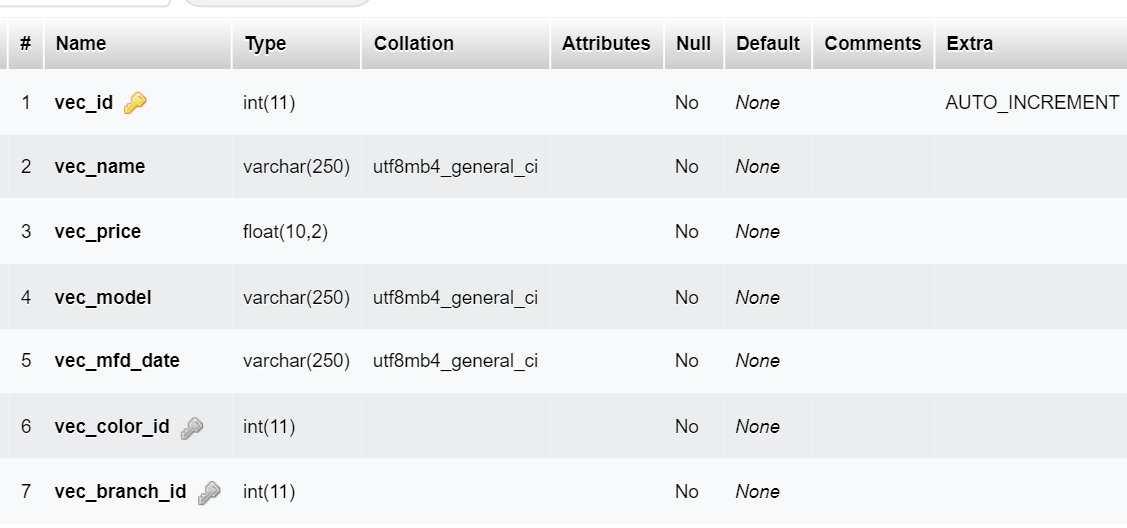
A data dictionary contains metadata, or information about a database. The data dictionary is critical because it contains information such as what is in the database, who has access to it, where the database is physically stored, and so on. The data dictionary is normally not accessed by database users; it is only accessed by database administrators. In general, the data dictionary contains information like:

Names of all the database tables and their schemas.

* Details about all the tables in the database, such as their owners, their security constraints, when they were created etc.
* Physical information about the tables such as where they are stored and how.
* Table constraints such as primary key attributes, foreign key information etc.

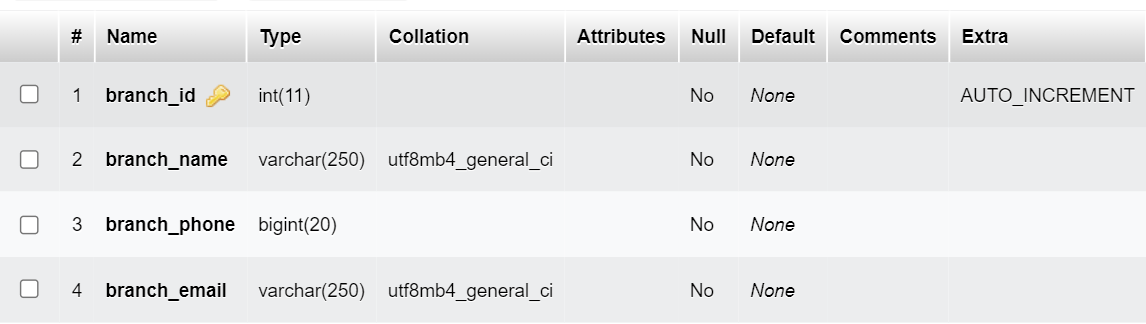
# Data dictionary in my Scenario.

## Data dictionary of vehicle.



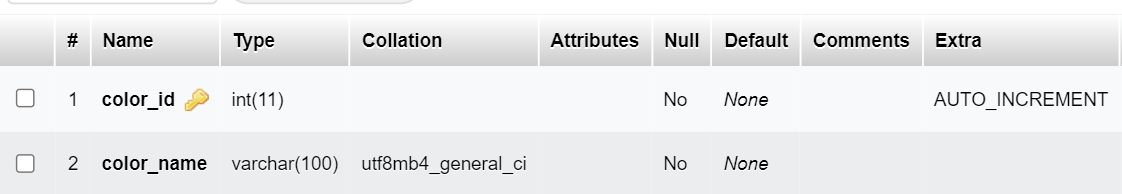
*Screenshot 4: Data dictionary of vehicle*

## Data dictionary of branch



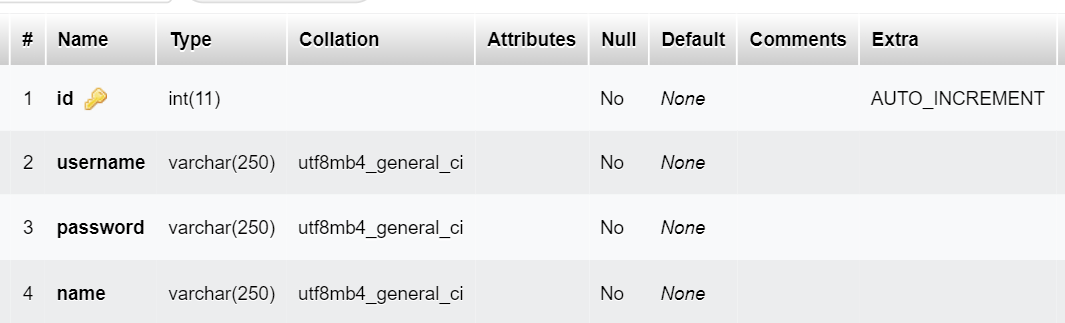
*Screenshot 5: Data dictionary of branch*

## Data dictionary of color



*Screenshot 6: Data dictionary of color*

## Data dictionary of users



*Screenshot 7: Data dictionary of color*

## Data dictionary of customer



*Screenshot 8: Data dictionary of customer*

According to the given table, it stores the data of table like table name, datatype, data length, null and description and relationship features that help to store and manage data. Therefore, I include various tables, such as user information with username and password (password should be stored in encrypted format), customer details with customer name, address, phone, email, purchase date, branch id, vehicle id. Every table is filled up like this way. For the security, I include windows authentication/SQL authentication and user login with encrypted password.

# Data validation

The process of verifying and validating data before it is used is known as data validation. To ensure accurate results, any type of data handling task, whether it is gathering data, analyzing it, or structuring it for presentation, must include data validation. It can be tempting to avoid validation because it takes time. However, it is a necessary step in achieving the best results possible. A system includes several checks to ensure that the data being entered and stored is logically consistent. Data validation is now much faster thanks to technological advancements. The majority of data integration platforms incorporate and automate the data validation step, making it an inherent step in the overall workflow rather than an additional one. There is little need for human intervention in such automated systems. Data validation becomes necessary because poor-quality data causes problems downstream, and cleansing data later in the process incurs higher costs. Within organizations that deal with data and its collection, processing, and analysis, the data validation process has grown in importance. It is regarded as the foundation for effective data management because it enables analytics based on meaningful and valid datasets. (Anon., n.d.)

# Significant of Data validation

Data validation ensures the dataset's accuracy, cleanliness, and completeness by eliminating data errors from any project to ensure that the data is not corrupted. While any data can be validated, including data within a single application. End-users may lose trust in data if it is inaccurate or incomplete. Data validation is a component of the ETL (Extract, Transform, and Load) process, which involves migrating the source database to the target data warehouse. In order to increase the value of the data warehouse and the information stored in it, data validation must be performed.

# Normalization:

Normalization is a database design technique that eliminates undesirable characteristics such as Insertion, Update, and Deletion Anomalies. Normalization rules divide larger tables into smaller tables and use relationships to connect them. The goal of SQL normalization is to remove redundant (repetitive) data and ensure that data is stored logically. Without any normalization in database, all information is stored in one table so we used normalization method up to 3NF. (Anon., n.d.)It has some forms and we used that normalization formed are described below:

## First Normal Form(1NF)

The first normal form is the normal database form, which requires that the data not contain repeating groups. I had to change a multi-value attribute in the relationship to a single value.

* Each table cell should contain a single value.
* Each record needs to be unique.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Email | Phone | Address | Purchase date | Vehicle | Color |
| Alex Koirala | [alex@email.com](mailto:alex@email.com) | 9823602111 | Biratnagar | 11/01/2021 /  13/032022 | Camry / Avalon | Red / Black |
| Sumesh | [shomesh@email.com](mailto:shomesh@email.com) | 9807782992 | Kathmandu | 01/01/2019 | Corolla | Black |
| Milan | [milan@email.com](mailto:milan@email.com) | 9823607515 | Dharan | 01/02/2020  01/05/2018 | Yaris /  Fortuner | Yellow / White |
| Richa | [richa@email.com](mailto:richa@email.com) | 9801111114 | Illam | 03/04/2018 | Corolla | Red |
| Bibek | [bibek@email.com](mailto:bibek@email.com) | 9801111115 | Itahari | 11/03/2019 /  11/10/2019 | Fortuner /  Corolla | Black / Red |

Table 1: random data

In the above example, three customers have purchased different vehicles in different dates & rest of the customers have only purchased once. Since, the normalization has not been implemented we have multiple purchase records attached in a same row. Let’s break the data into 1NF system according to the assignment scope.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Cust name | Cust email | Cust phone | Cust address | Purchase date | Vehicle name | colors |
| Alex Koirala | [alex@email.com](mailto:alex@email.com) | 9801111111 | Biratnagar | 01/01/2018 | Camry | Red |
| Alex Koirala | [alex@email.com](mailto:alex@email.com) | 9801111111 | Biratnagar | 13/01/2018 | Avalon | Black |
| Sumesh | [shomesh@email.com](mailto:shomesh@email.com) | 9801111112 | Kathmandu | 01/01/2018 | Corolla | Black |
| Milan | [milan@email.com](mailto:milan@email.com) | 9801111113 | Dharan | 01/02/2018 | Yaris | Yellow |
| Milan | [milan@email.com](mailto:milan@email.com) | 9801111113 | Dharan | 01/05/2018 | Fortuner | White |
| Richa | [richa@email.com](mailto:richa@email.com) | 9801111114 | Illam | 03/04/2018 | Corolla | Red |
| Bibek | [bibek@email.com](mailto:bibek@email.com) | 9801111115 | Itahari | 11/03/2019 | Fortuner | Black |
| Bibek | [bibel@email.com](mailto:bibel@email.com) | 9801111115 | Ithari | 11/10/2019 | Corolla | Red |

Table : normalized data according to 1NF rule

## Second Normal form(2NF)

* Rule 1- Be in 1NF
* Rule 2- Single Column Primary Key that does not functionally dependent on any subset of candidate key relation.

There should be no partial dependencies in the table. A non-prime attribute is determined by the appropriate subset of candidate keys in a partial dependency. So, on the 2NF form, I divided the Borrow table into three sections: vehicle table, branch table and color table. I filled out the vehicle table with vehicle details, the color table with color details and branch table with branch details. In the Borrow table, I added the column vehicle Id, color Id and branch id which contain the Primary Keys of the vehicle table, color tables and branch table respectively.

|  |  |  |  |
| --- | --- | --- | --- |
| Cust id | Cust name | Cust email | Cust address |
| 1 |  |  |  |
| 2 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Cust id | Cust name | Cust email | Cust address |
| 1 | Alex | [alex@email.com](mailto:alex@email.com) | Biratnagar |
| 2 | Sumesh | sumesh@gmail.com | Kathmandu |
| 3 | Milan | [milan@email.com](mailto:milan@email.com) | Dharan |
| 4 | Richard | [richa@email.com](mailto:richa@email.com) | Illam |
| 5 | Bill | [bibek@email.com](mailto:bibek@email.com) | Itahari |

Table : customer data according to 2NF rule

|  |  |
| --- | --- |
| Vehicle id | Vehicle name |
| 1 | Fortuner |
| 2 | Yaris |
| 3 | Corolla |
| 4 | Camry |
| 5 | Avalon |

Table 4: vehicle data

|  |  |
| --- | --- |
| Color id | Color name |
| 1 | Red |
| 2 | Black |
| 3 | Yellow |
| 4 | White |

*Table 5: color data*

## Third Normal Form(3NF)

* Rule 1- Be in 2NF
* Rule 2- Has no transitive functional dependencies.

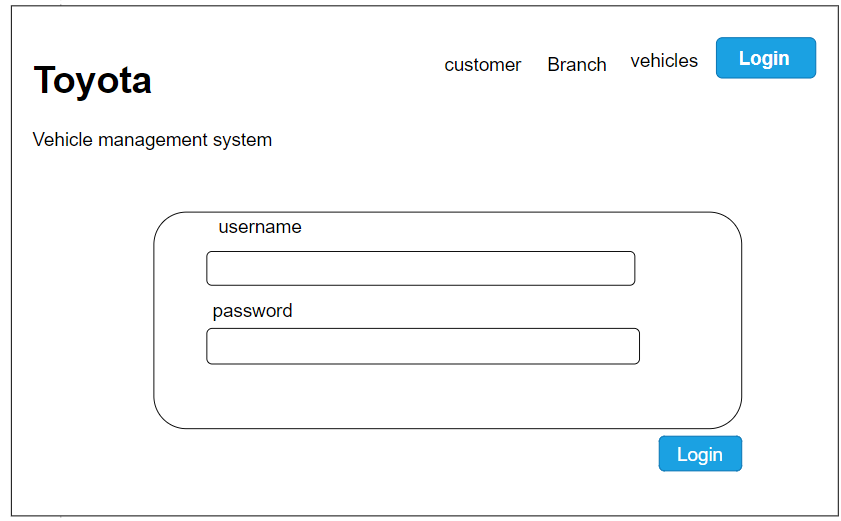
Non-prime attributes should not be transitively dependent. That is, non-prime attributes (those that do not form a candidate key) should not rely on other non-prime attributes. So, on the 3NF form, I divided the vehicle table into color table and branch tables. I created an vehicle table for vehicles details, color table for color details. Then I removed column vehicle details, color details, branch details from the borrow table. In the vehicle column, color column, branch column, I added the vehicle id, color id, and branch id respectively, which contain the primary keys respectively. To move our 2NF table into 3NF, we again need to again divide our table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cust id | Cust phone | Purchase date | Vehicle id | Color id |
| 1 | 9801111111 | 01/01/2018 | 4 | 1 |
| 1 | 9801111111 | 13/01/2018 | 5 | 2 |
| 2 | 9801111112 | 01/01/2018 | 3 | 2 |
| 3 | 9801111113 | 01/02/2018 | 2 | 3 |
| 3 | 9801111113 | 01/05/2018 | 1 | 4 |
| 4 | 9801111114 | 03/04/2018 | 3 | 1 |
| 5 | 9801111115 | 11/03/2019 | 1 | 2 |
| 5 | 9801111115 | 11/10/2019 | 3 | 1 |

Table 5: customer data according to 3NF rule

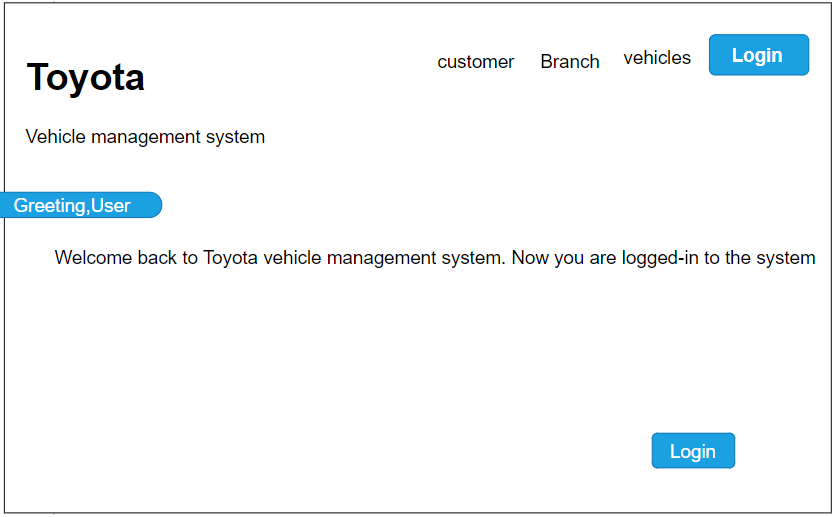
# Output design of Toyota management system:

Before login you will see our dashboard that is shown below, for the login we will have to put our correct username and password for the login.



*Figure 11: Output design of dashboard before login.*

When we login then our dashboard will show like this,



*Figure 12: output design of dashboard after login*

# Effectiveness of design:

Depending on the situation, I use the appropriate tool to create a relational database in order to create a Toyota management system. To properly store and handle all of the data, I created a branch table, a color table, a branch table, and a borrow table. I logically created an entity relationship diagram to show the relationship between various tables, as well as normalized the database to store data with minimal data redundancy. The data dictionary and relationship diagram provide primary constraints, identity property, and foreign key for tables with ID in relation to other tables. On the encryption user table, I create columns that store information in the database based on the scenario, such as username and password. Encryption of a protected password into an unreadable format that those in the password column cannot use. I created SQL injection-proof applications using the stored procedure on the login form. To make various types of decisions, I use data validation and also create designs for various types of reports. When the application is fully completed, which means that the user and device specifications have been met, I build various forms in Visual Studio code. It is very useful for databases because it has data encapsulation and data integration, which ensure data protection and SQL binding.

# User requirement and system requirement:

User and system requirement: For this project there consist of some requirement in order to make it in proper way. So, for that there are something which are require by system and user and they are shown below:

|  |  |
| --- | --- |
| User requirement | System requirement |
| * Customer entry * Vehicle entry * Branch entry * Colors entry | * User login * Xampp * Visual studio code * Mysql |

Table 7: user requirement and system require

# Conclusion

I used all of the design tools to define the entity relationship diagram, database diagram, and data dictionary, which is an important part of the database for developers and users to understand. It uses data normalization and stored procedures to store information without duplication and makes the database available for the retrieval of various types of reports. This is how I completely meet all of the Toyota management system's requirements.

# P2

# Developing the database system

## Introduction:

In order to complete this task, I need to implement all the tables by applying all the primary key, foreign key, unique, not null etc. with the evidence of my database design with user interface, output and data validations, and query across multiple tables using different joins. Constraints are rules and regulations that govern the type of information that can be added/updated/deleted from a table. The primary goal of SQL constraints is to maintain data integrity. Validation verifies the accuracy and completeness of data entered into various applications and their components. System security refers to the safeguarding of databases and applications against unauthorized access. Data backup is the process of creating and storing copies of data that can be used to protect organizations from data loss. Testing is an operation that determines whether the actual results match the predicted outcomes and ensures that the software system is defect-free. Clauses are important SQL features that are used to select tables in various ways.

# Types of SQL Constraints:

## Primary Key:

A primary key is a key in a database table that uniquely identifies each row/record. It must be unique and cannot contain any null values. A table can only have one primary key, which can be made up of one or more fields. In my database, I use the primary key with identity auto increment in each ID column, allowing us to insert data into each table in a unique way. (Anon., n.d.)

## Foreign Key:

In a database table, a primary key is a key that uniquely identifies each row/record. It must be distinct and cannot include any null values. A table can have only one primary key, which can be one or more fields. In my database, I use the primary key with identity auto increment in each ID column, which allows us to insert data into each table in a distinct manner. (Anon., n.d.)

## Unique key:

A unique key is a key that ensures that each column or group of columns has unique values. If a column has a specific restriction, it means that no duplicate values exist in a table for that column. This key is used in the User table's username column because each username must be unique and cannot be null. (Anon., n.d.)

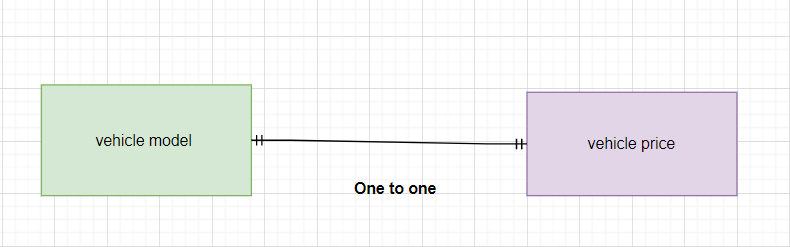
## Not Null:

In SQL, the condition not null is used to check for non-values. A null value is not accepted. If we put a null in any column, we must fill in any values, so we can't switch to another column without first adding values. Non-null constraints are used in most of the columns in my database because they are important and database information would be incomplete without them. (Anon., n.d.)

# Relationships:

“A relationship, in the context of databases, is a situation that exists between two relational database tables when one table has a foreign key that references the primary key of the other table. Relationships allow relational databases to split and store data in different tables, while linking disparate data items.” (Anon., n.d.) There are mainly three relationship and they are:

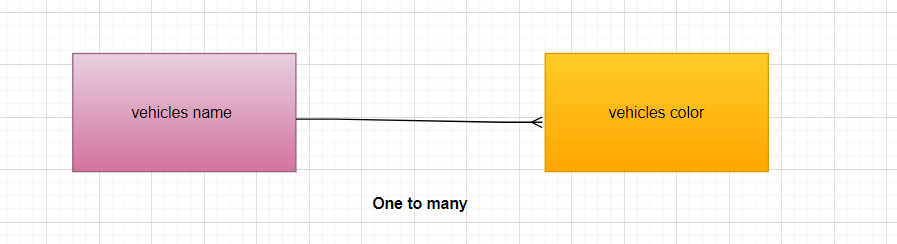
One to One Relationship:

Relation between one row table with another one table is known as one to one relations. 

*Figure 13: one to one relationship*

One to Many relationship:

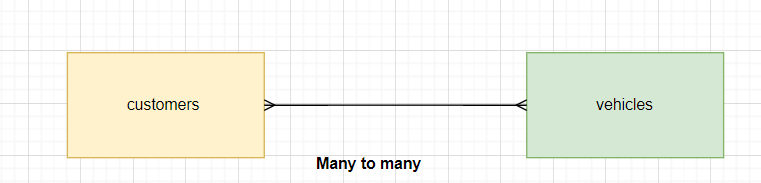
“In relational databases, a one-to-many relationship occurs when a parent record in one table can potentially reference several child records in another table. In a one-to-many relationship, the parent is not required to have child records; therefore, the one-to-many relationship allows zero child records, a single child record or multiple child records. The important thing is that the child cannot have more than one parent record.”(Anon., n.d.)

****

*Figure 14: one to many relationship*

Many to Many relationship:

When a multiple tables consisting of records associates with another same multiple table then many to many relation occurs. They are quiet tricky relationship to be presented.



*Figure 15: many to many relationship*

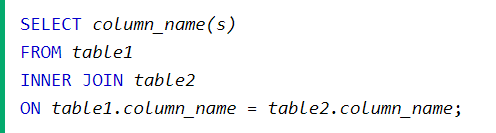
# **Joins**:

The main objective of join is to join tables from each other. There are different types of join and they are listed below:

Inner join:

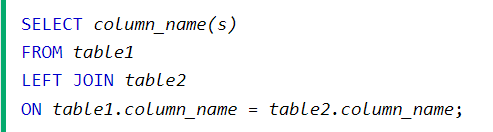
“It is the most common join rather than other joins. The inner join creates a new result table by combining column values of two tables (table1 and table2) based upon the join-predicate. The query compares each row of table1 with each row of table2 to find all pairs of rows which satisfy the join predicate. When the join-predicate is satisfied, column values for each matched pair of rows of A and B are combined into a result row.”(Anon., n.d.)

Syntax:



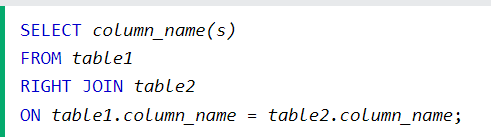
Left Join: The Left join returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side, if there is no match.

Syntax:



Right Join: “The right join keyword returns all records from the right table (table2), and the matched records from the left table (table1). The result is NULL from the left side, when there is no match.”(Anon., 1999)

Syntax:



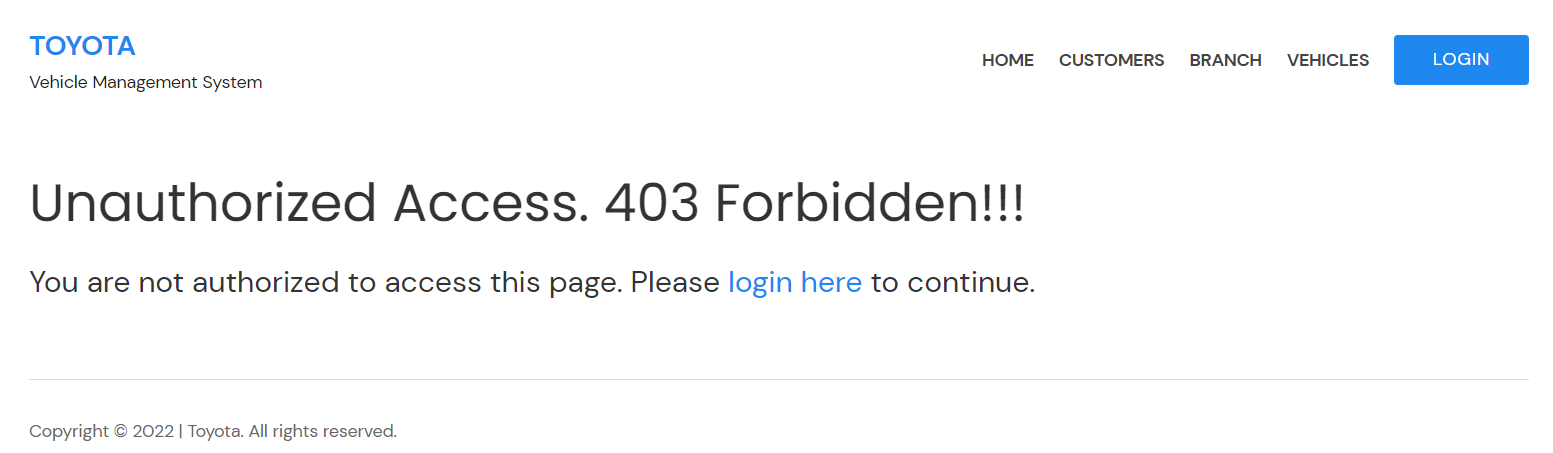
# User interface:

“A user interface is the view of the database interface that is seen by the user. User interface are often graphical or at least partly graphical (GUI) Graphic User Interface constructed and offers tools which make the interaction with database easier.”(Anon., 2016)

The project on which I have been working is ready and I am going to show it with screenshots of it.

## Dashboard:

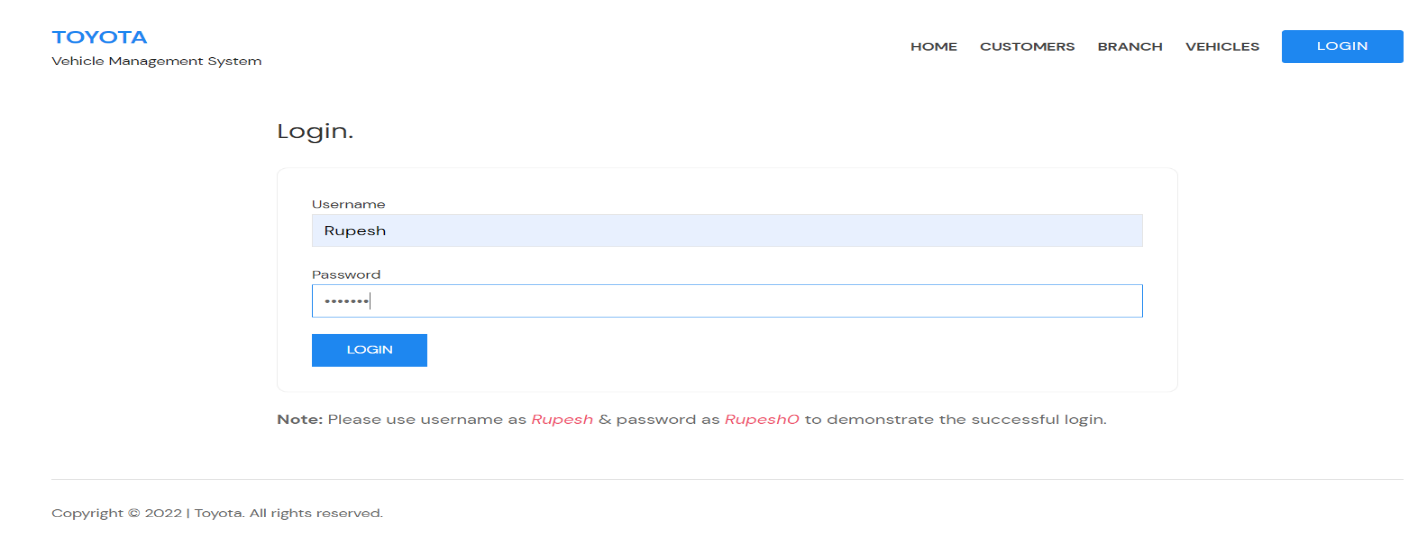
Before login dashboard look like this:



## LOGIN:

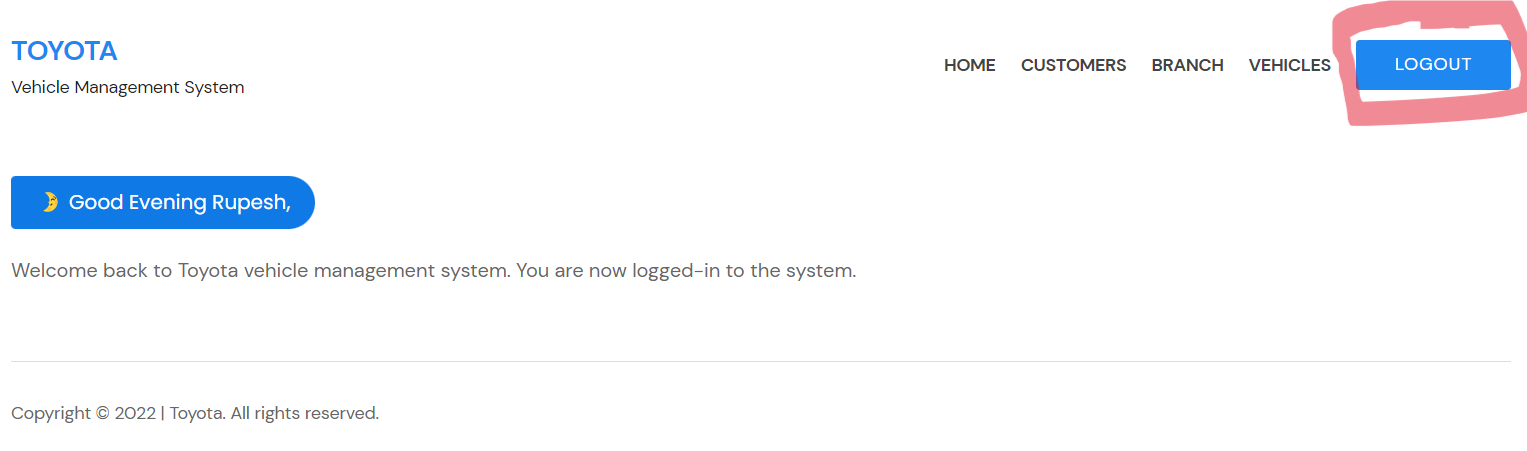
When we tap on login bottom then interface looks like screenshot given below.

There should be give username, password and then tap on login bottom for login.



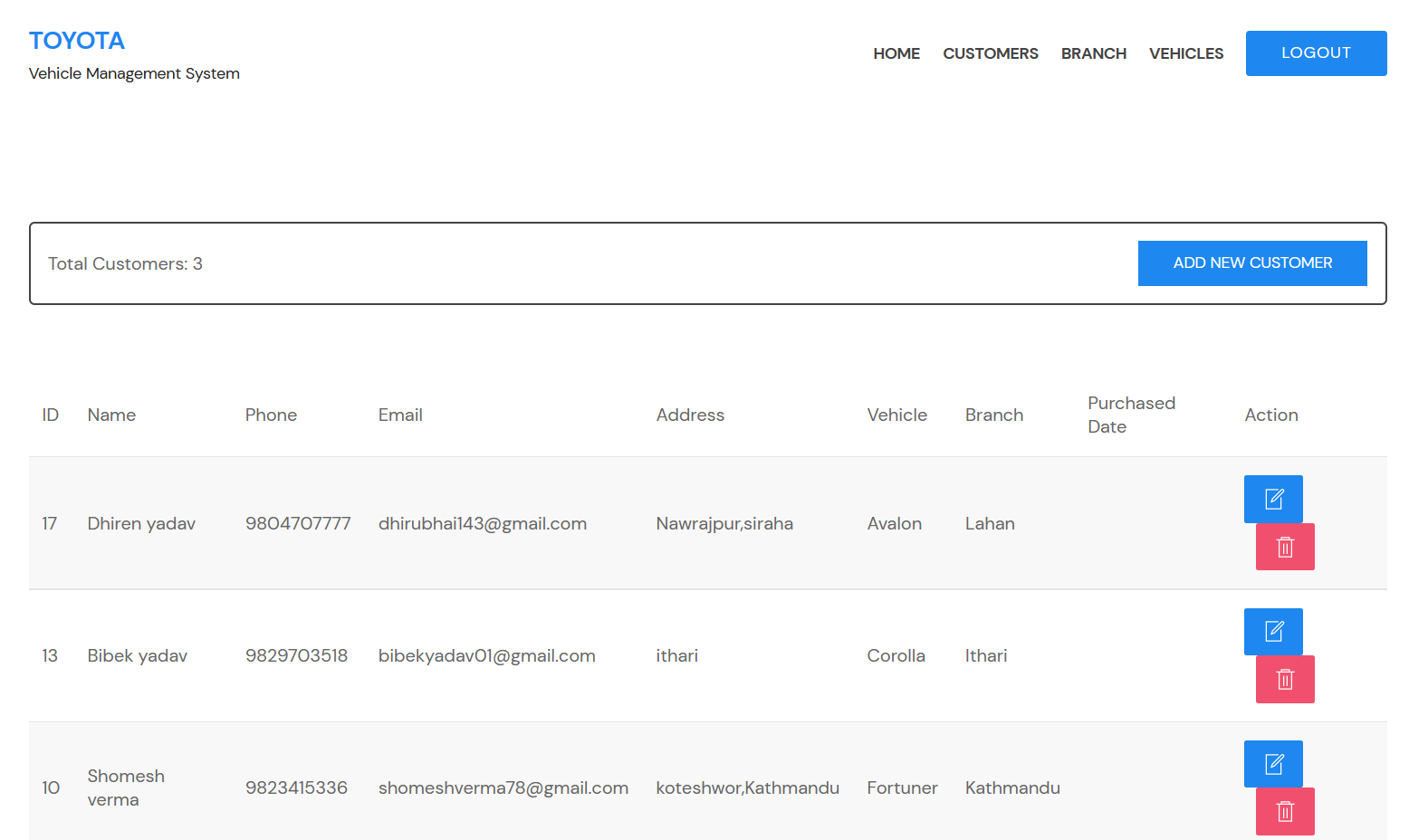
## HOME:

When you complete login then dashboard interface looks like or when you click on home bottom (screenshot is given below). There is menu as home, customer, branch, vehicle and logout bottom in right corner. Every bottom have different function I will discuss about every bottom separately. when you click on logout bottom then it suddenly logout your dashboard.



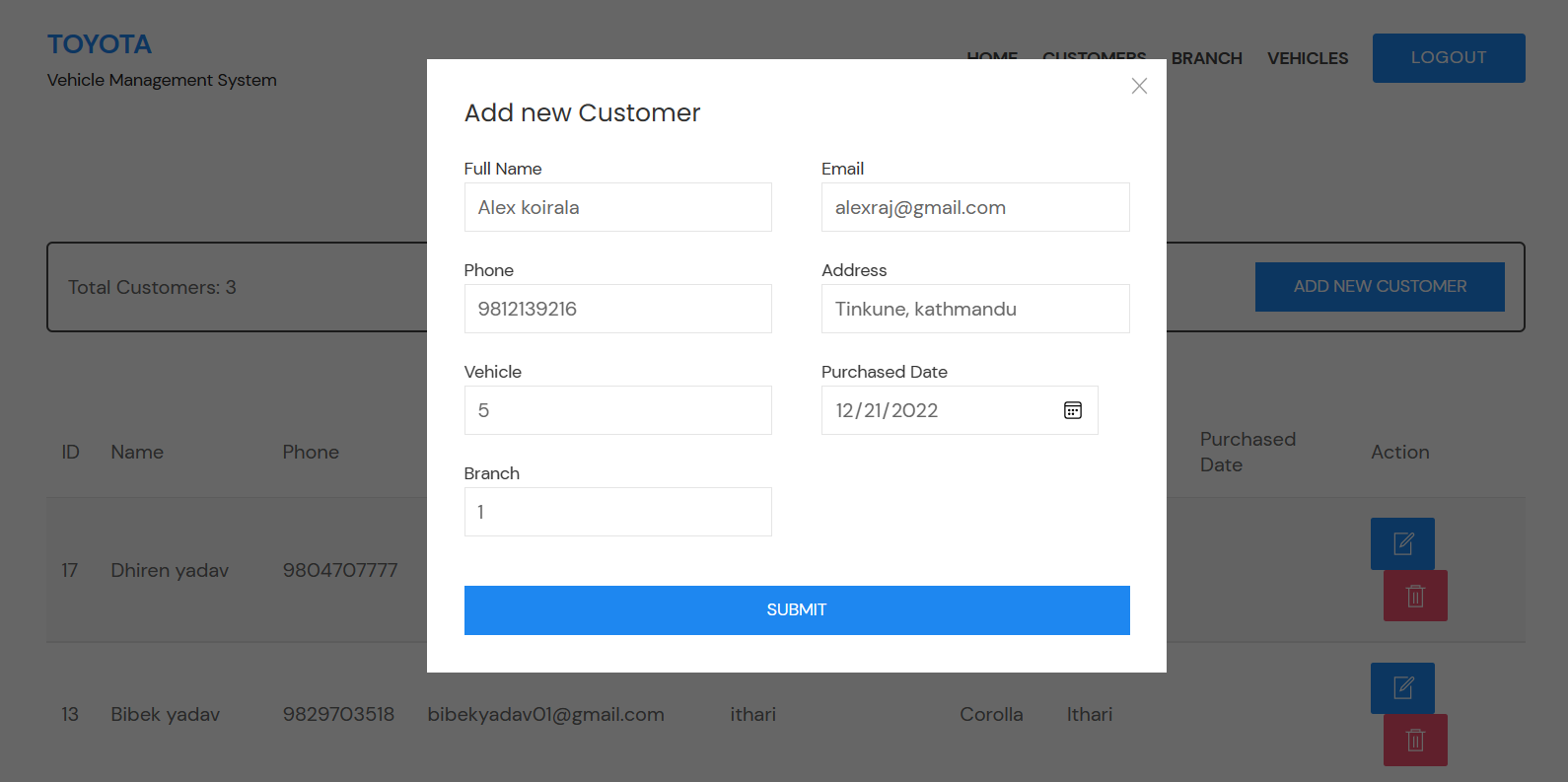
## Customers:

When you click on customers its interface looks like as screenshot given below. List of customer and details of customers including name, phone, email, Address, vehicles, branch, purchase date and two action bottom update and delete bottom appear.



### Customer entry:

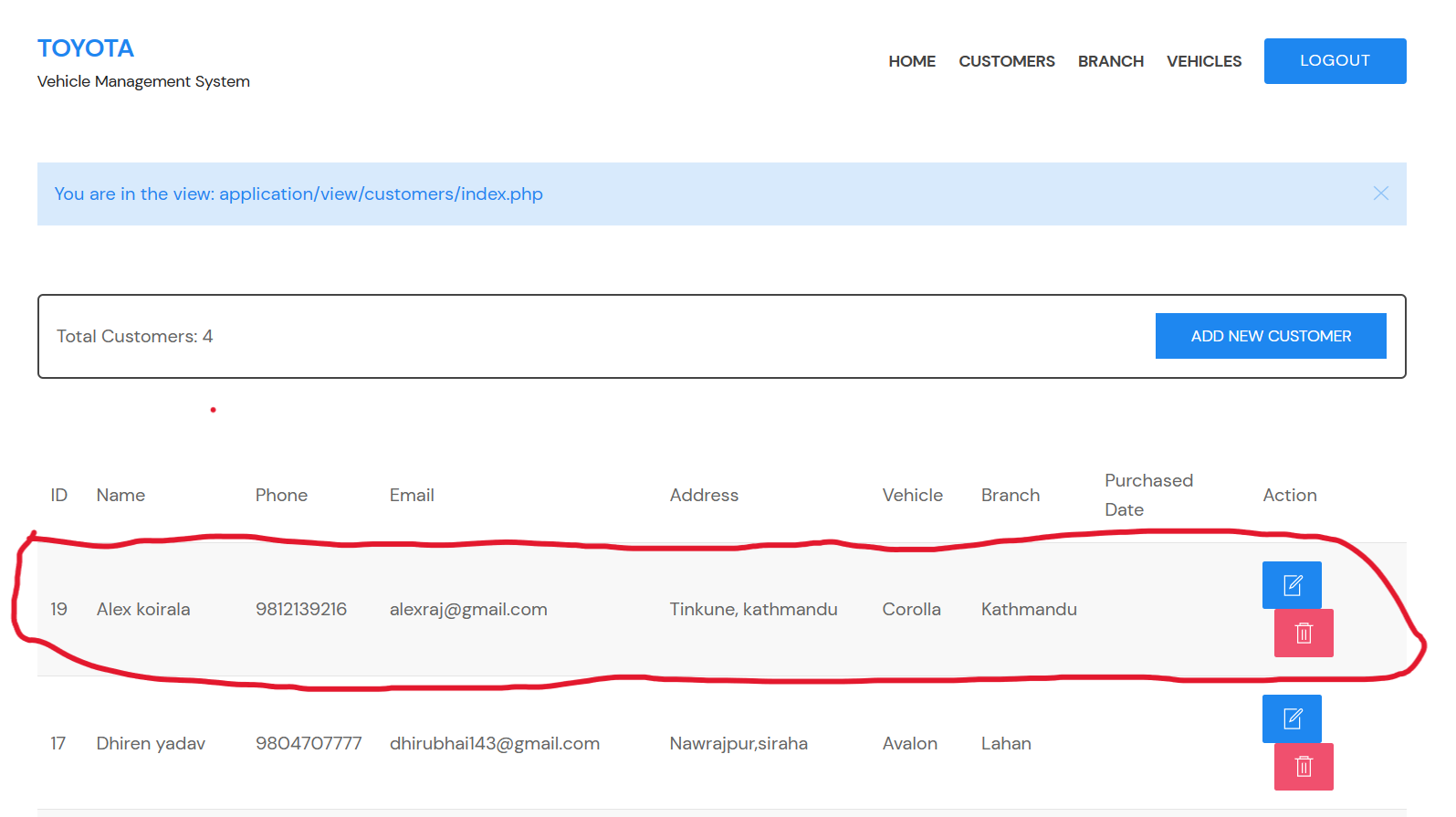
If you need to add new customer then click on add new customer and fill up the required data. We can entry new customer and there details too details like phone, email, address, vehicles, branch and purchase date. Customer entry interface looks as give below screenshot.



When you filled up required data then click on submit bottom. Your customer entry completed you can see your new customer details.

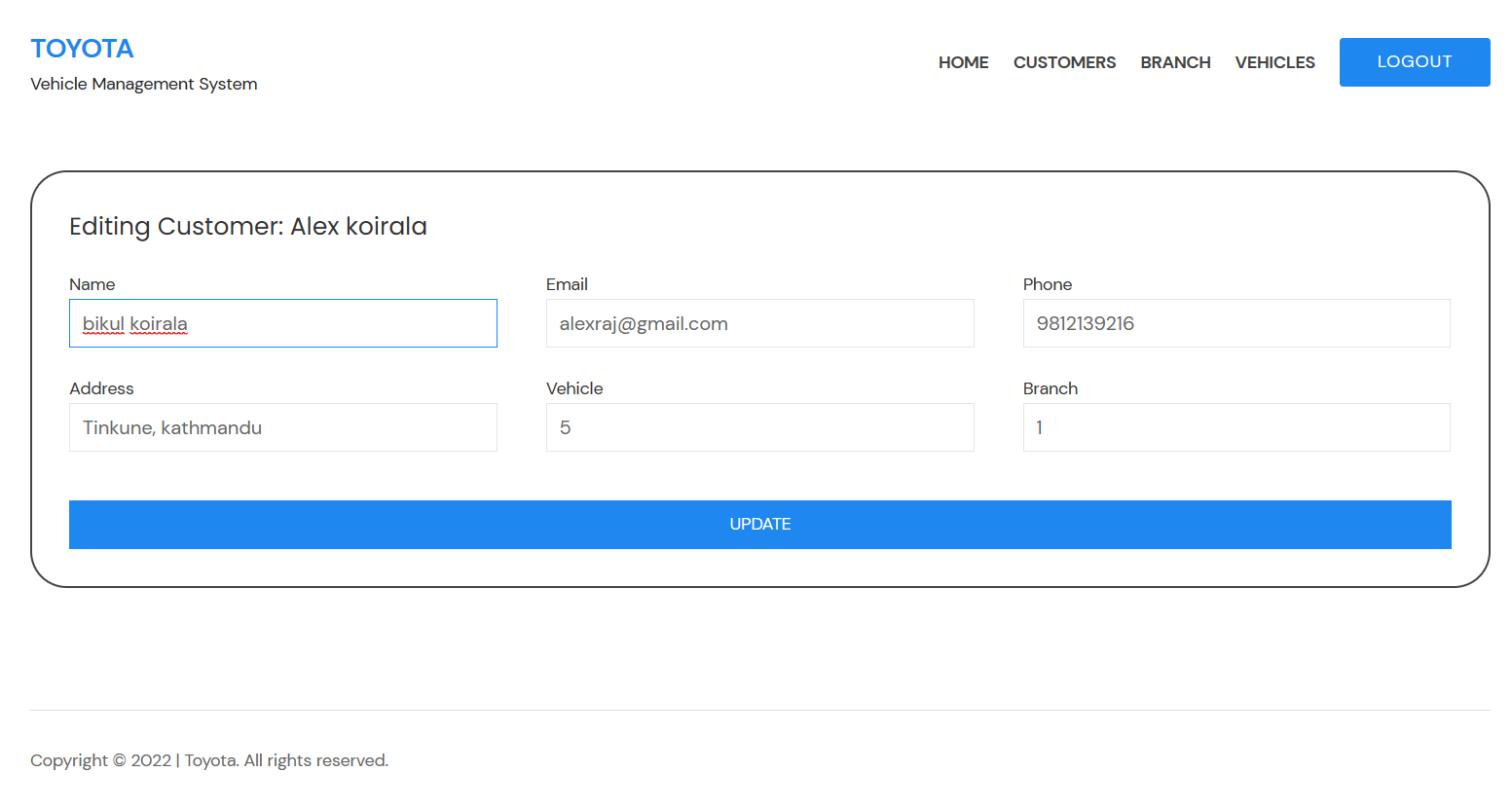
### New customer output:

Your new customer is shown which is marked up with red color in below screenshot below.



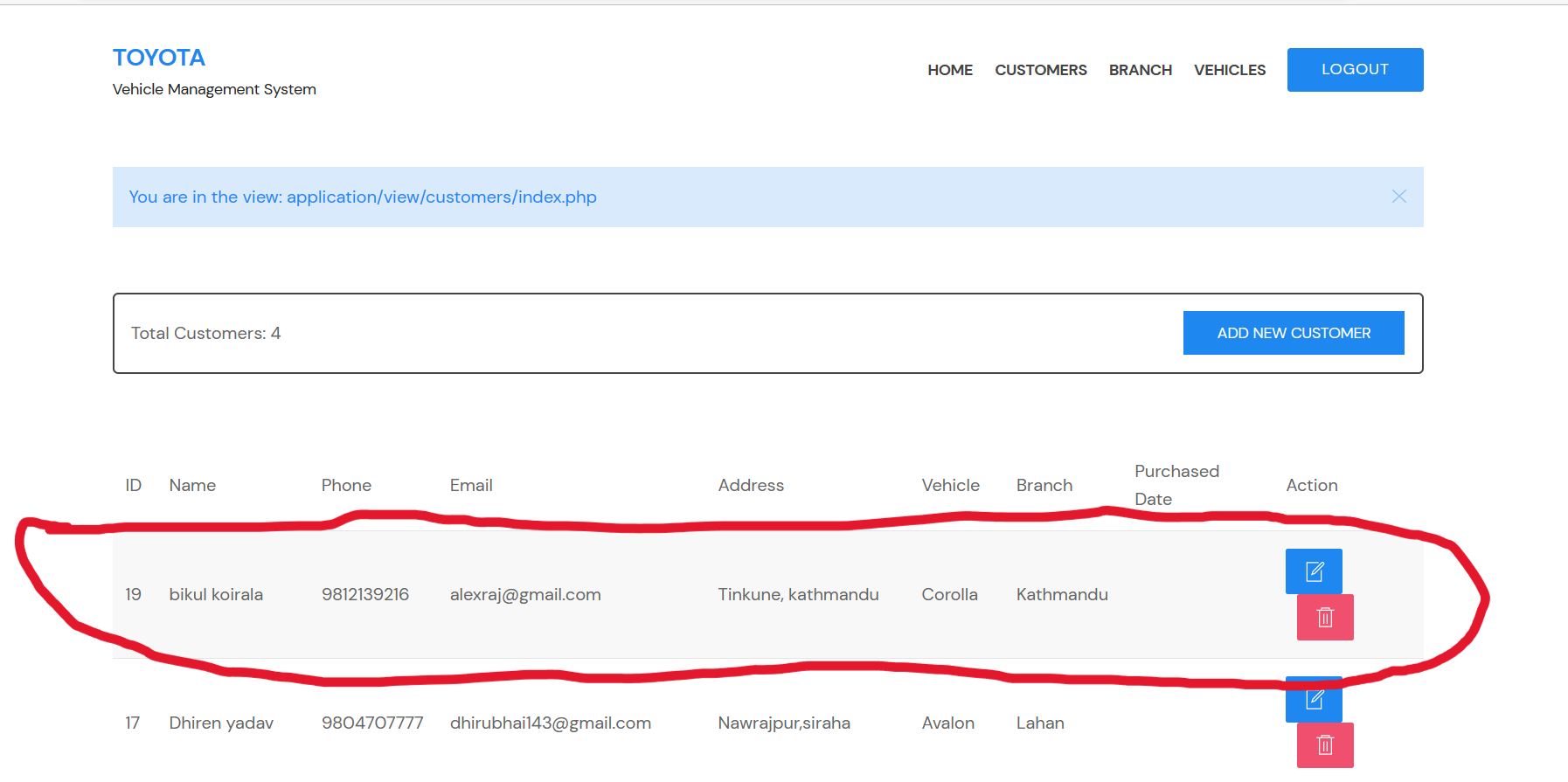
## Update customer:

Click on edit action bottom, editing customer interface will appear. Edit customer details then click update bottom. Example; I update name, alex Koirala to bikul Koirala.



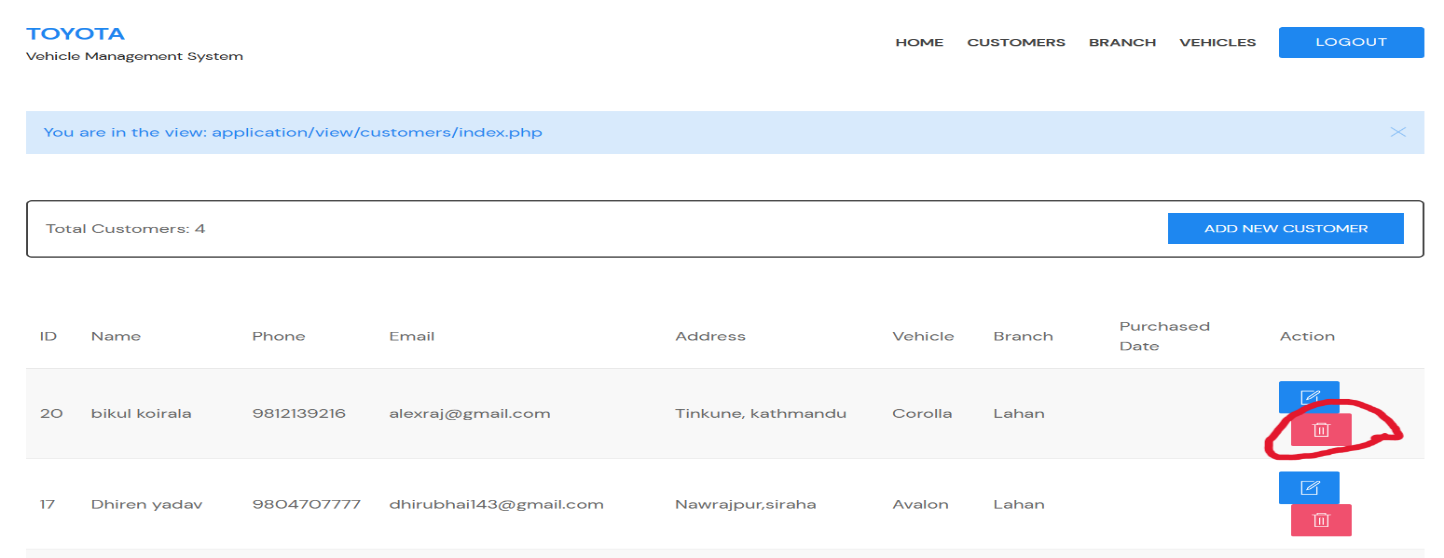
## Update customer output:

Customer name is updated from alex Koirala to bikul Koirala.



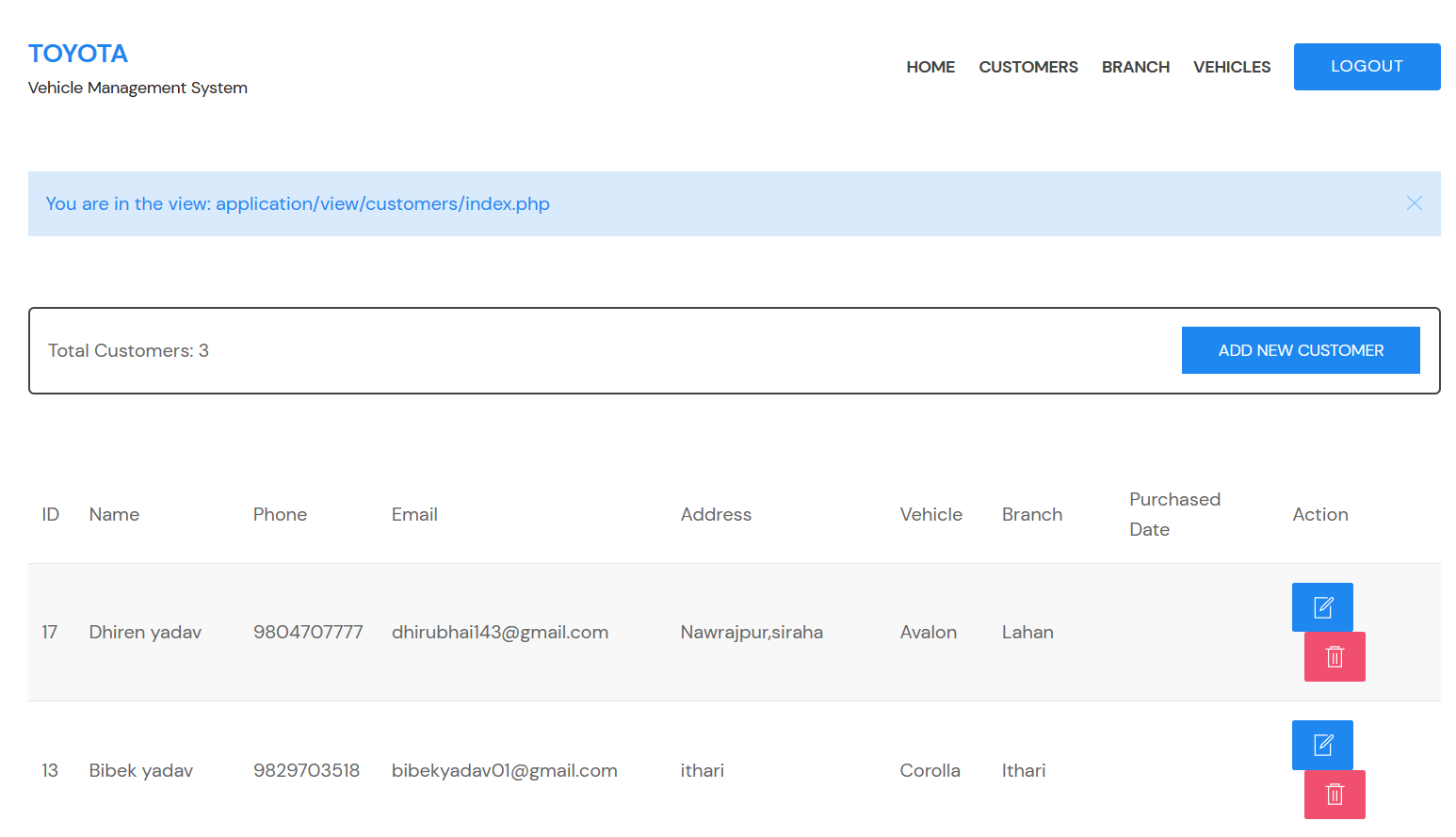
## Delete customer:

if you need to delete a customer form customer list then Click on delete action bottom that delete bottom can see which is markup in given screenshot.



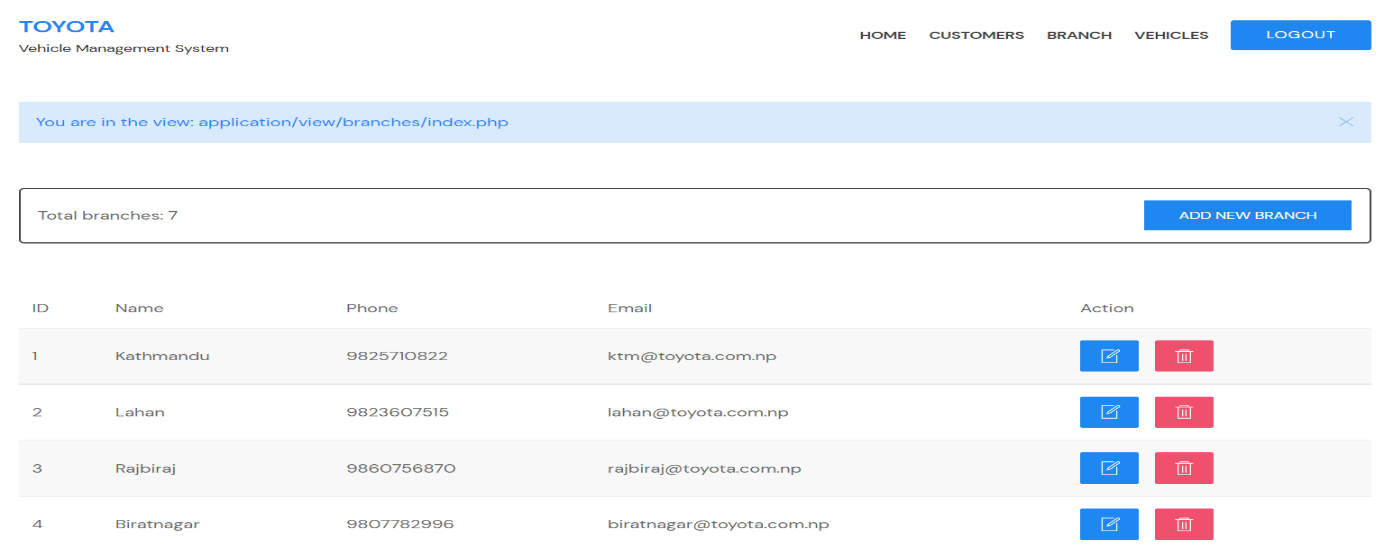
## Delete customer output:

You can see customer bikul Koirala is deleted from customer list.



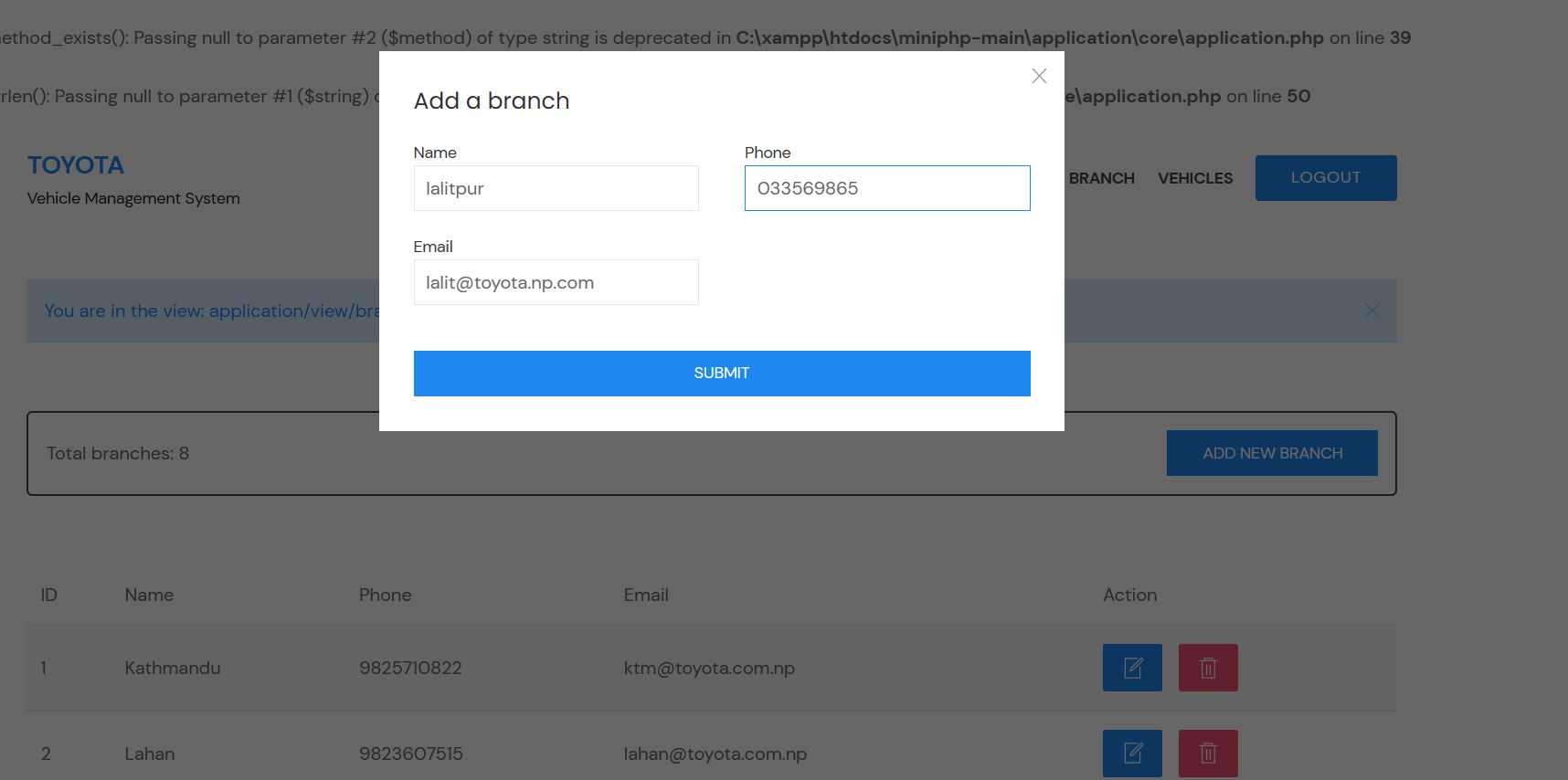
## Branch:

Click on branch bottom it show List of branch and details of branch including branch name, phone, email. Screenshot is given below:



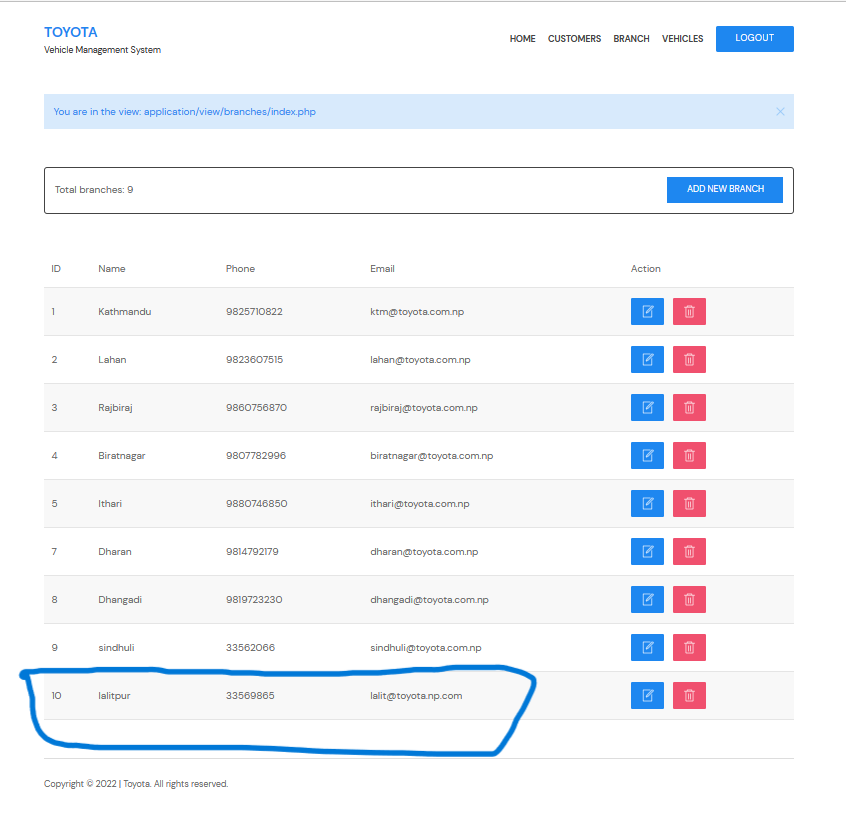
### Branch entry:

Click on add branch bottom to add new branch and their details and fill up the data as needed.



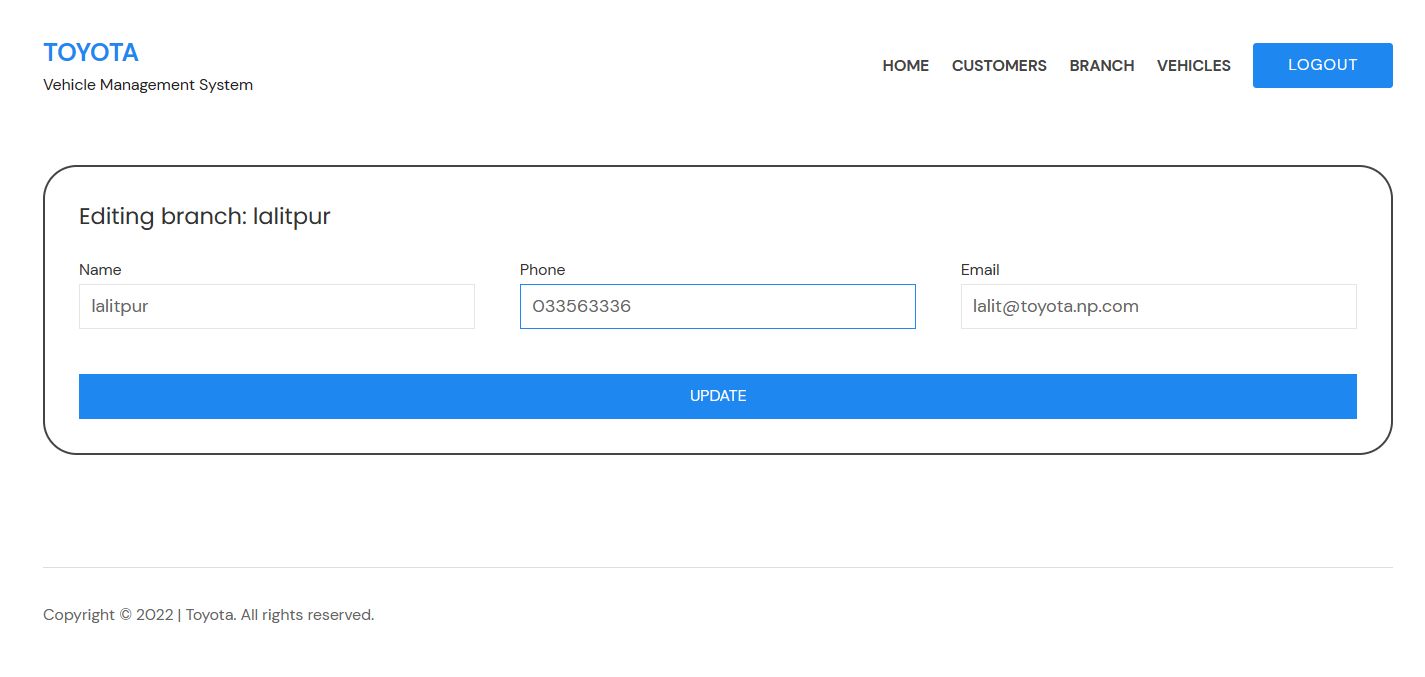
### Branch entry output:

You can see Lalitpur branch that I added, screenshot is given below.



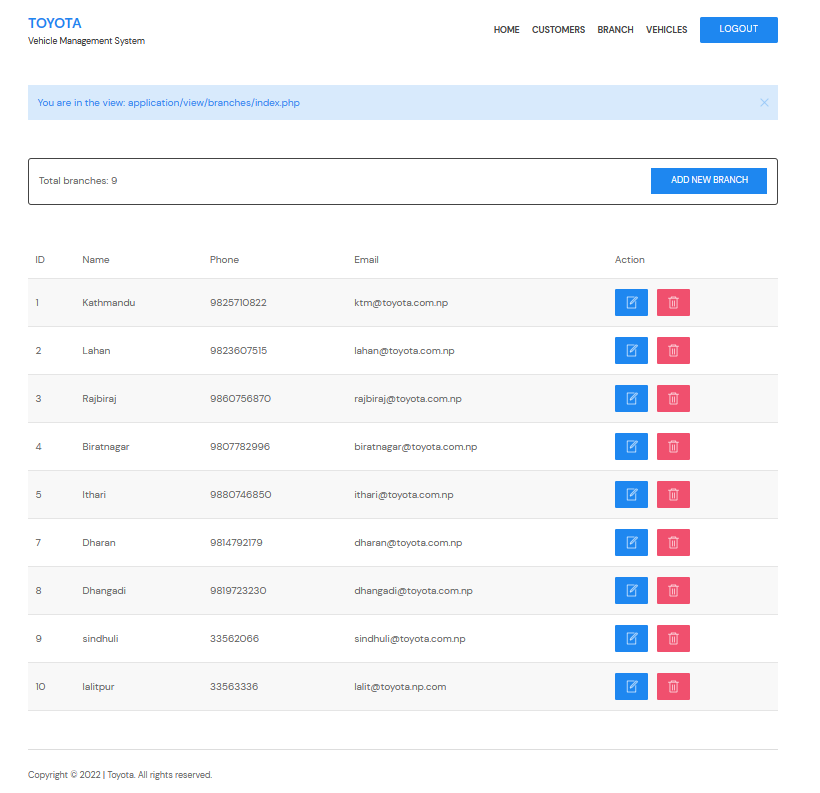
### Branch update

Click on edit action bottom, editing branch interface will appear. Edit branch details then click update bottom. Example; I update phone, 33569865 to 033563336.



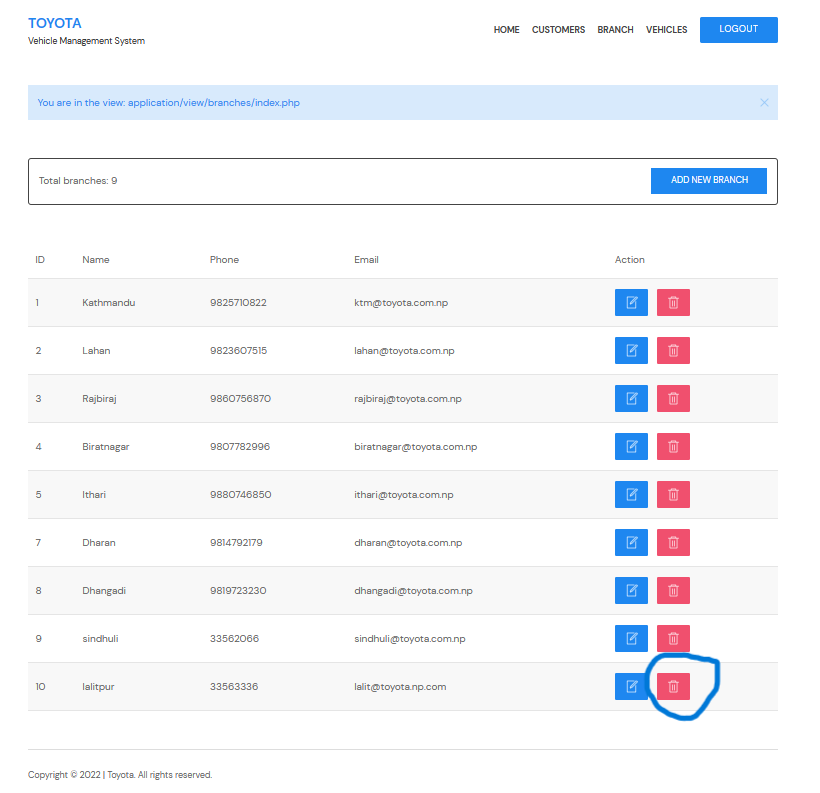
### Branch update output:

You can see number 10, phone of Lalitpur branch updated.



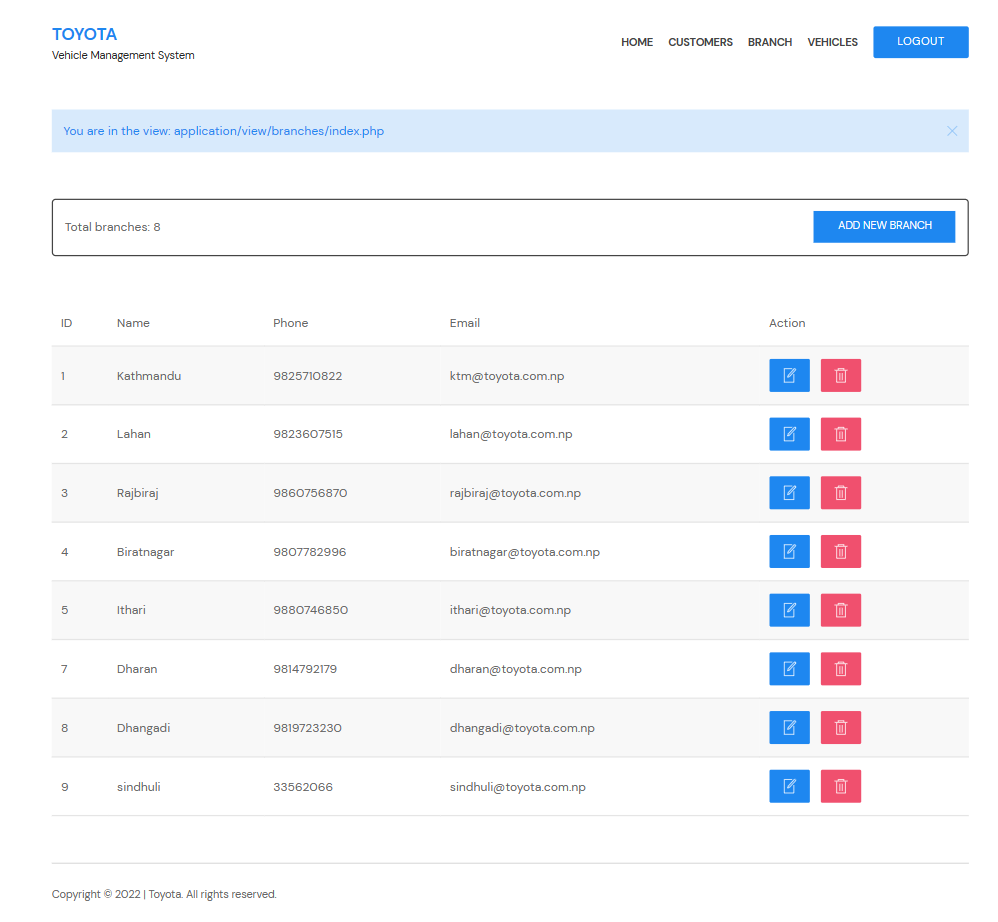
### Delete branch:

Click on delete bottom that delete bottom can see which is markup in given screenshot. Once you click here it will delete a branch.



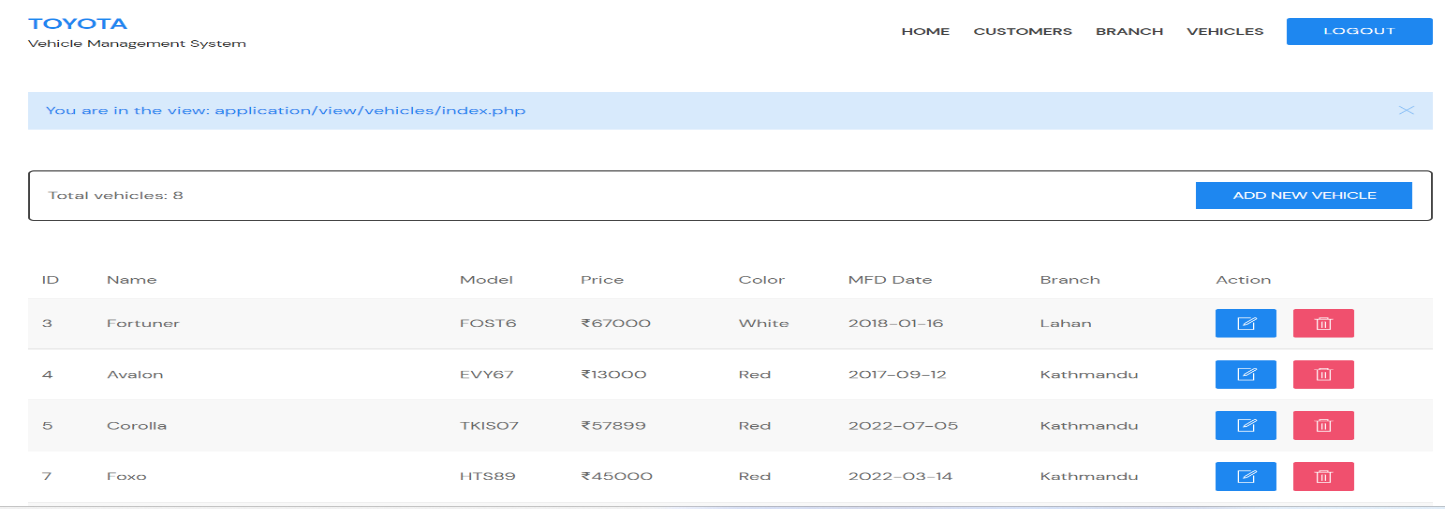
### Delete branch output:

You can see there is no Lalitpur branch, Lalitpur branch is deleted. Screenshot is given below.

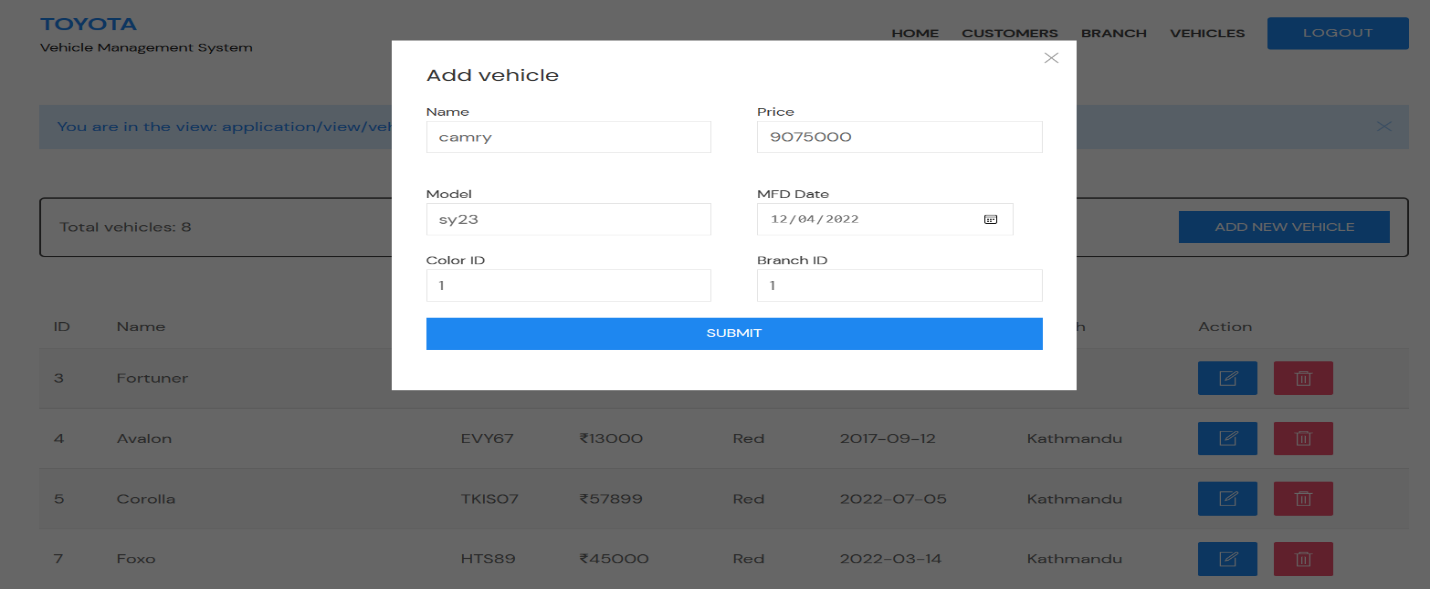


## Vehicle:

Click on vehicle bottom it show List of branch and details of branch including branch name, model, price, color, MFD date, branch, and action bottom Screenshot is given below:

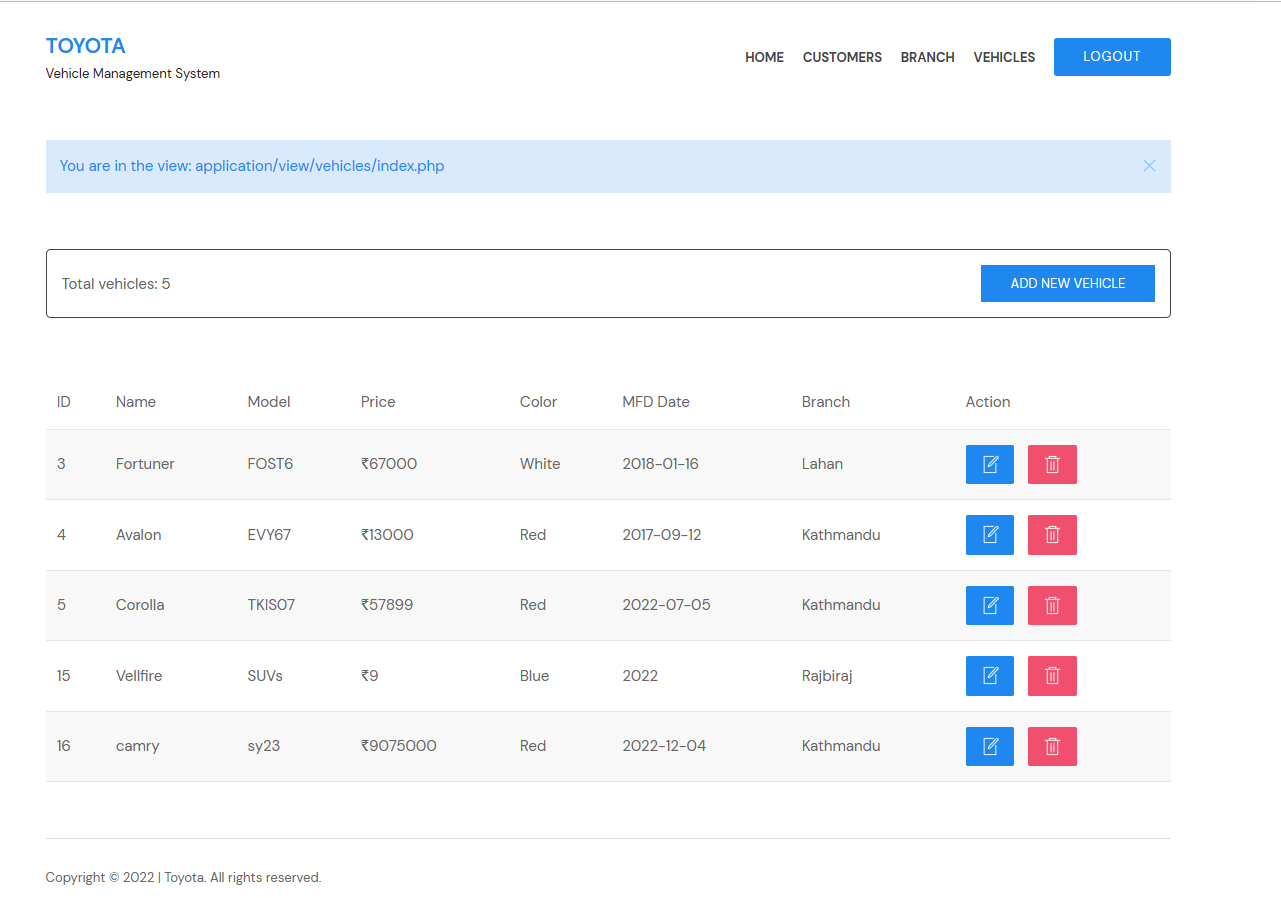


### Vehicle entry:

Click on add new vehicle bottom then entry needed data then click on submit bottom. you can add vehicles and their details including model, color, price, MFD date, and branch. Screenshot is given below.

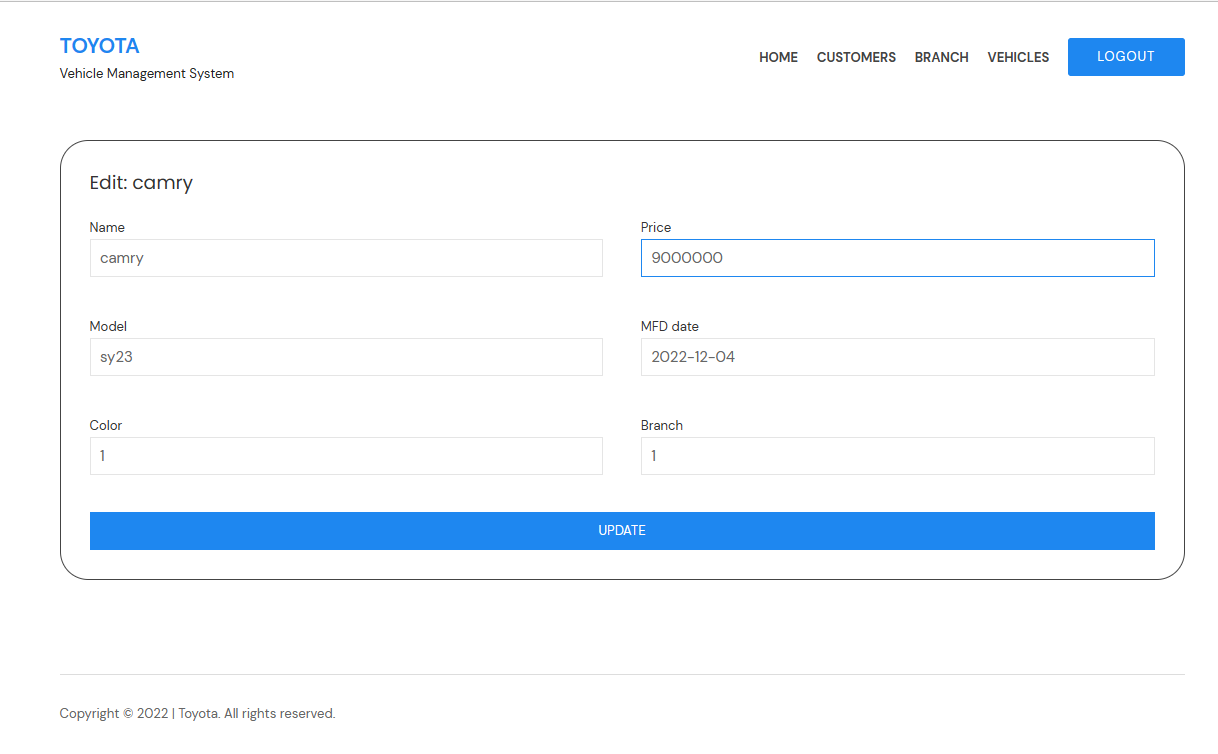
### Vehicle entry output:

Here vehicles which was added that show screenshot is given below.



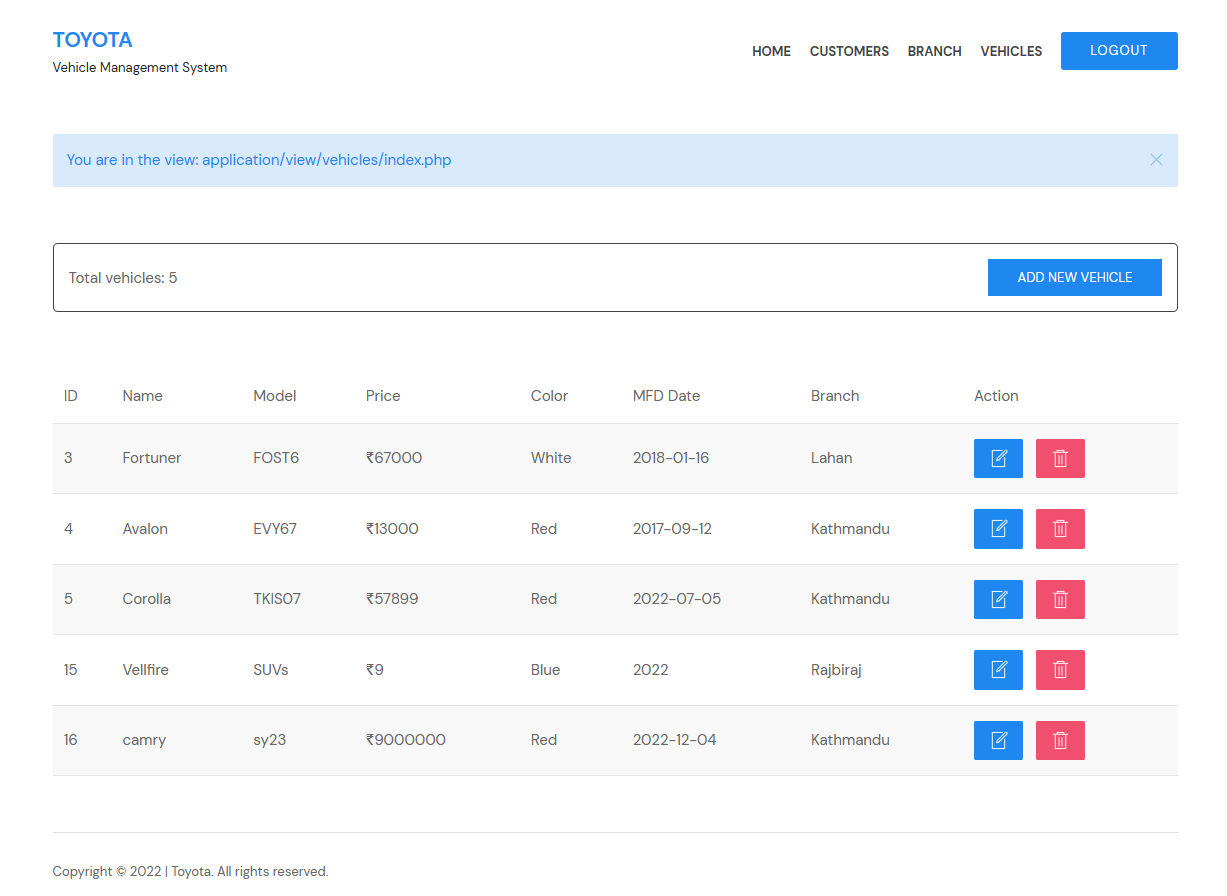
### Vehicle update:

Click on edit action bottom, editing vehicles interface will appear. Edit vehicles details then click update bottom.



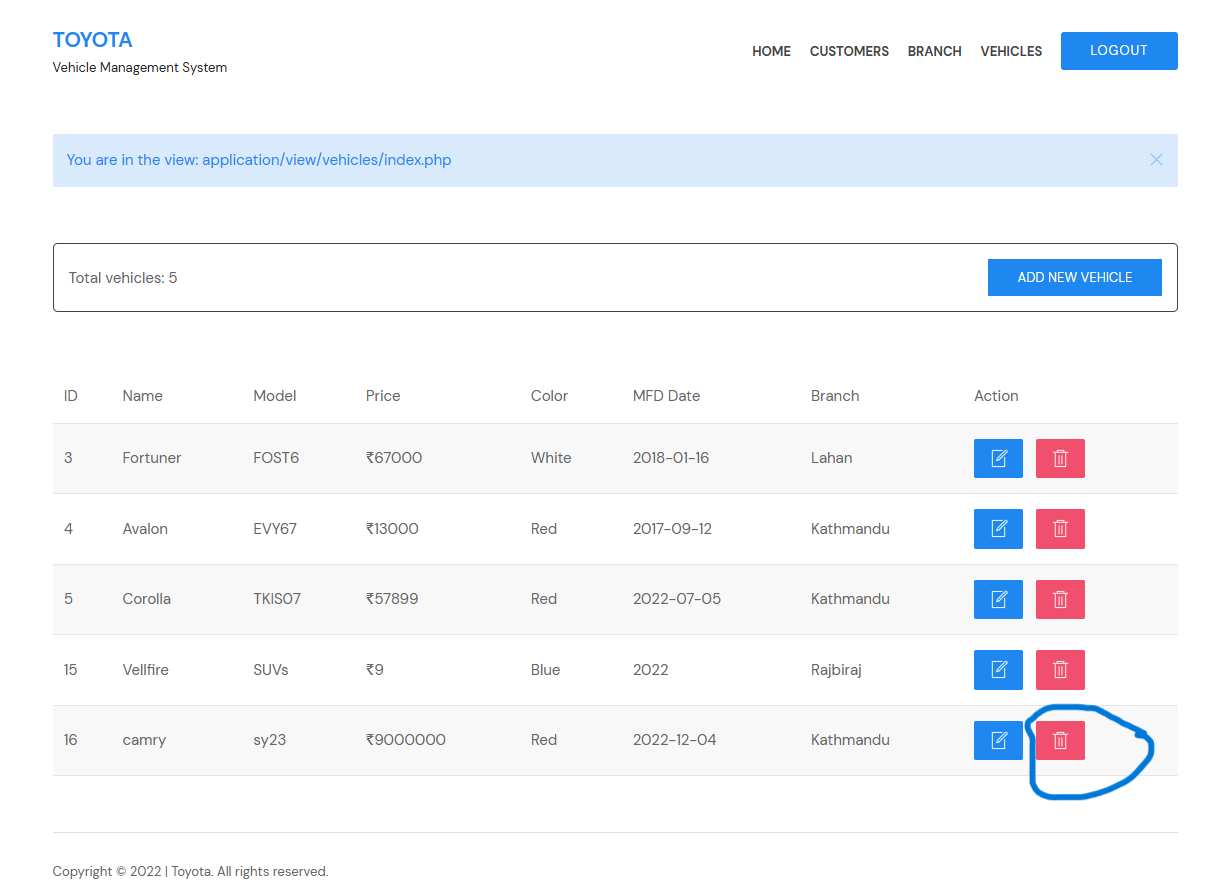
### Vehicle update output:

I update price of camry vehicle from 9075000 to 9000000. Screenshot is given below:



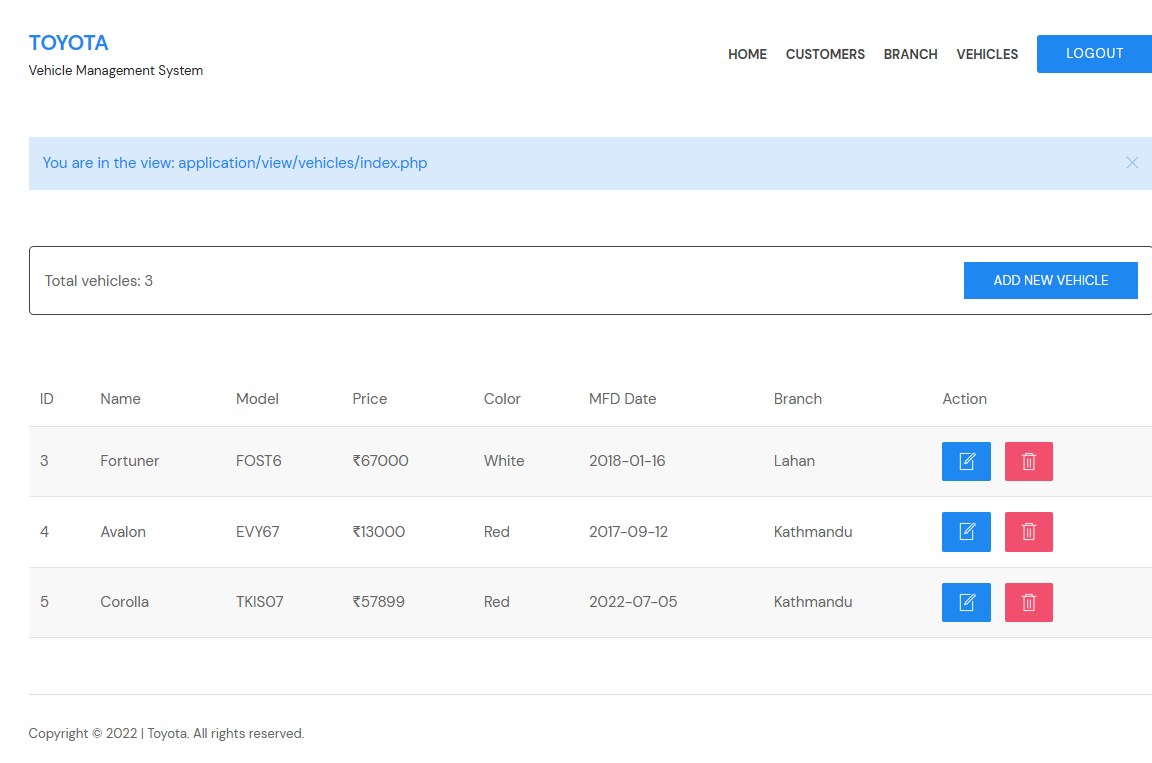
### Vehicle delete:

Click on delete bottom. vehicle will delete from list. I delete camry vehicle from list. (delete bottom is markup in screenshot)



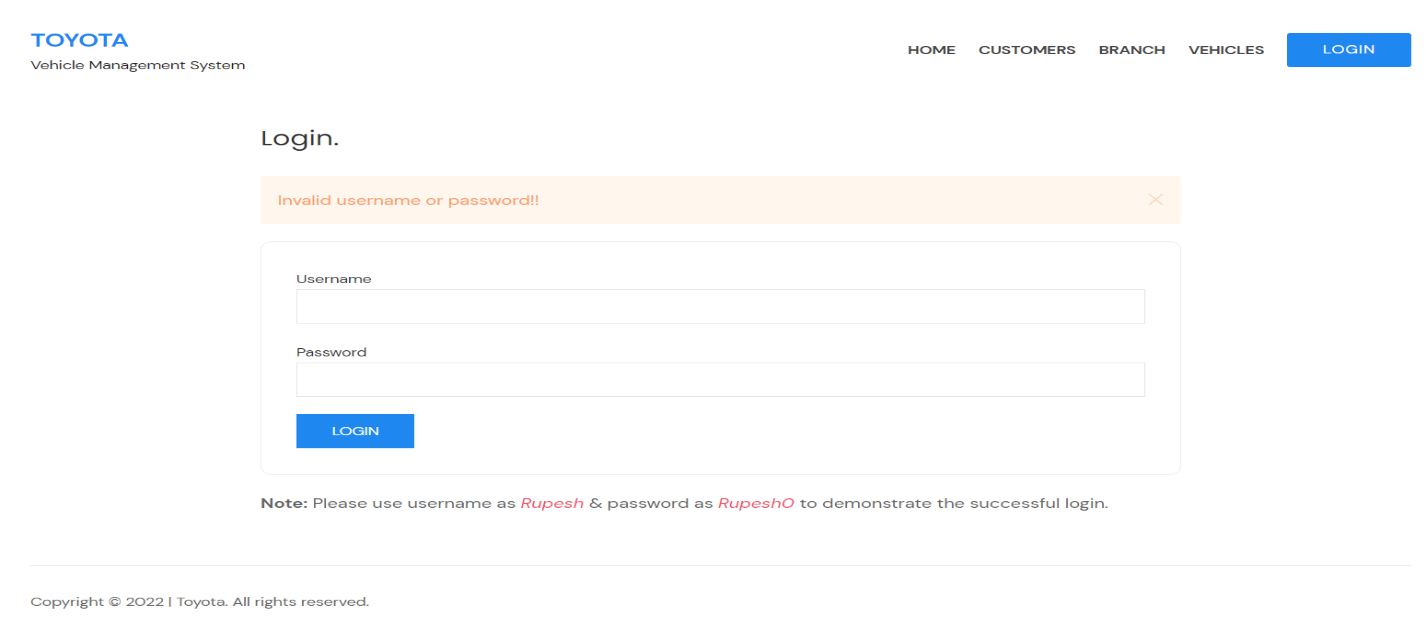
### Vehicle delete output:

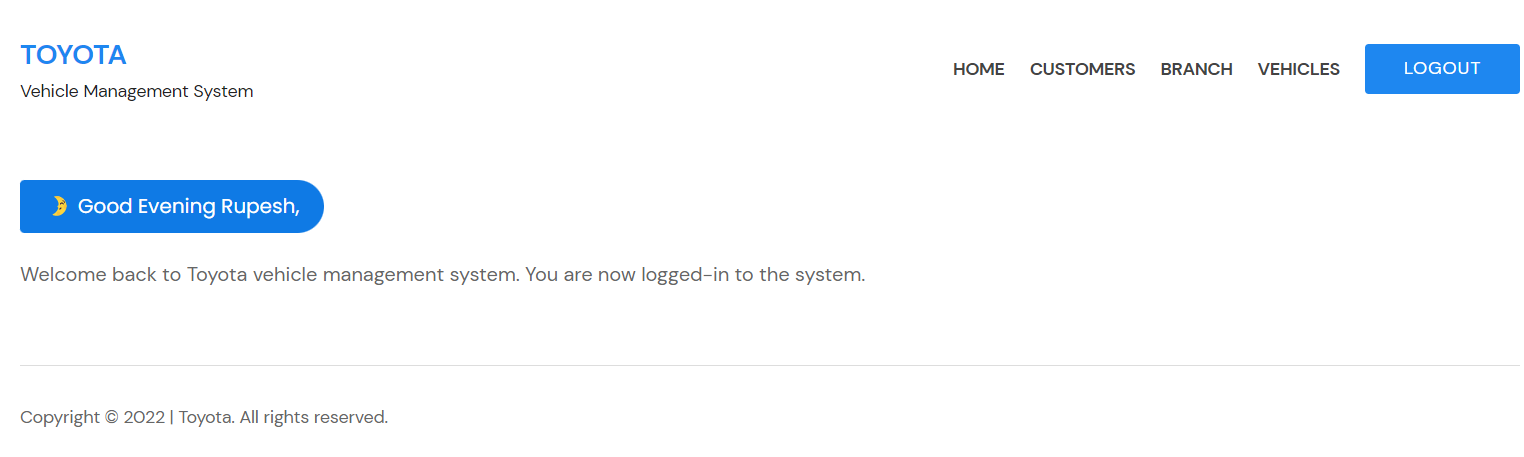
Here camry vehicles deleted from list.



# Data Validation:

“Data validation is a process that ensures the delivery of clean and clear data to the programs, applications and services using it. It checks for the integrity and validity of data that is being inputted to different software and its components. Data validation ensures that the data complies with the requirements and quality benchmarks. Data validation is also known as input validation.” (Anon., n.d.) Data validation is like main part of the project. It makes sure that data is placed correctly in application. If data is not provided correctly, then it will so the message box like (invalid username or password) screenshot is shown below.



Without the proper data we cannot enter in the home page and cannot get access to Toyota vehicle management system application. Database consist of all table where we entered all of data related to consist of all table. Home page of Toyota vehicle management system is given below:

# Conclusion:

I've added constraints to every table and created designs based on data with user interfaces that produced results. In this task, joins with different have also been provided. I had all the relevant constraints primary key, foreign key, unique, not null, etc. Applied when I had implemented the tables. I had created documentation to support my suggested database design in terms of user interface, output, and data validations. I had also implemented searching across several tables using various join types.

# P3

## Introduction

I create some basic SQL statements for insert, update, delete, and select data both in the front end and the back end in order to complete my project (Toyota management system).

## DDL (Data Definition Language):

"A DDL is a language used to create and edit data structures. For instance, tables in a database can be added, removed, or modified using DDL commands. DDLs are regarded as a subset of SQL, or Structured Query Language, which is utilized in database applications. However, a DDL may also specify different kinds of data, like XML.

The syntax for describing data in a data definition language is predetermined. The CREATE command is used to create a new table, for instance, and is then followed by parameters for the table name and column definitions. Each column's name and related data type can also be specified in the DDL. The ALTER command can be used to change a table after it has been created. The DROP command can be used to remove a table from a database if it is no longer required.

DDL is a subset of SQL, therefore it does not contain all of its commands. Commands like SELECT and INSERT, for instance, are categorized as belonging to the Data Manipulation Language (DML), whereas access commands like CONNECT and EXECUTE are categorized as belonging to the Data Control Language (DCL). The majority of SQL's allowed commands are included in the DDL, DML, and DCL languages. (Anon., 2015)

## DML (Data Manipulation Language)

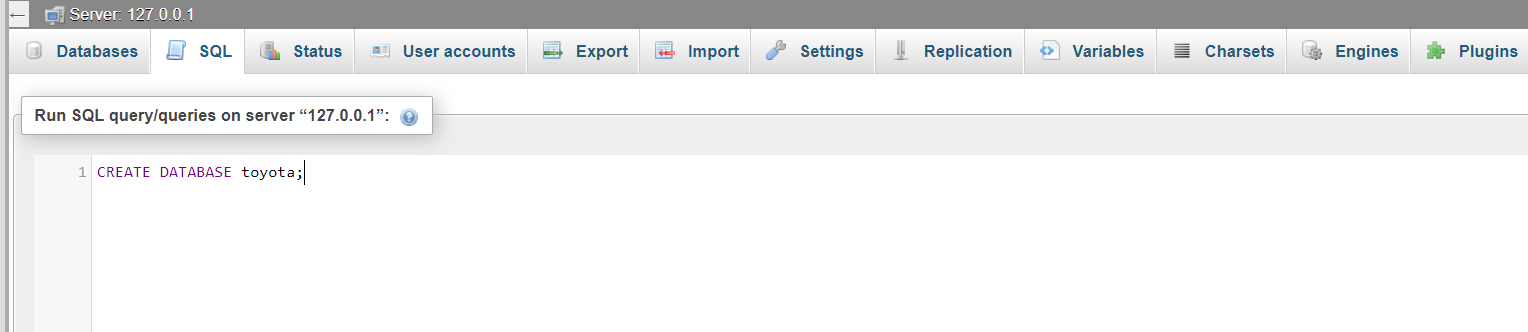
Users can access and manipulate data using the Data Manipulation Language (DML). Retrieval, insertion, deletion, and alteration of data or information kept in the database are all considered forms of data manipulation. The main objective is to achieve effective human–system interaction. DML is similar to mostly used as a Structured Query Language (SQL) for information retrieval and processing, easy English language. (Anon., 2011)

In my project there are five Module, my module are customer, vehicles, colors, branch and users. Query of insert, update, and delete of every module are shown below:

At first creating database, before creating table we have to create database by query like this:

### Create database:

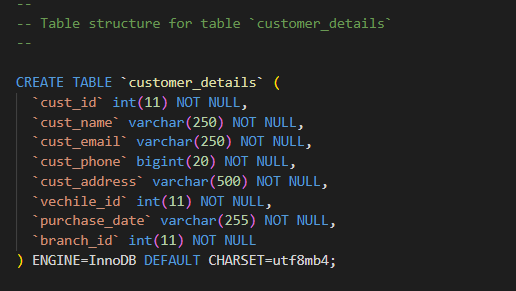
Query for creating database.



After creating database I create four table like branch, customer, vehicles and users. Now I am showing query of create, select, update, and delete for every table.

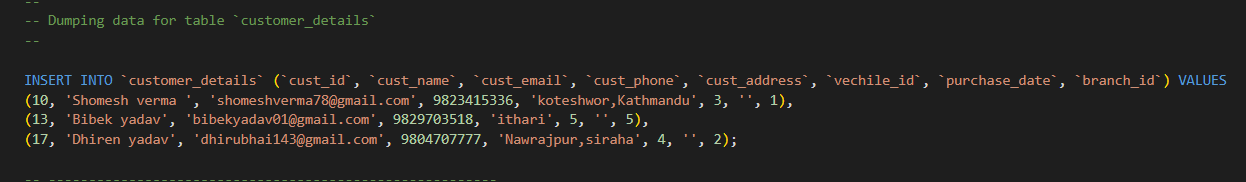
### Create table (customer):

Query for creating customer table.



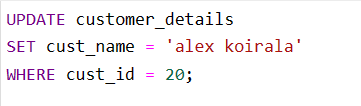
### Insert :

Query for insert data into customers table.

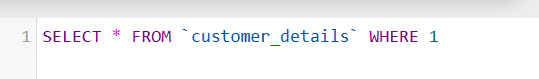


### Update:

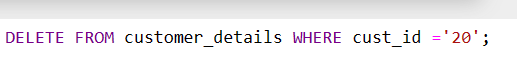
Query for update



### Select:

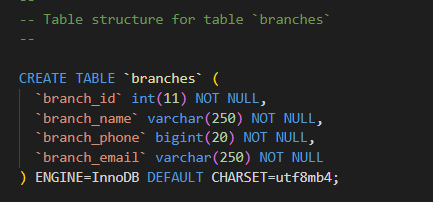


### Delete:



### Create table (Branch):

Query for creating branch table.



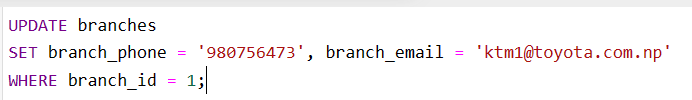
### Insert

Query for insert data into branch table.



### Update:

Query for update data of branch table.



### Select:

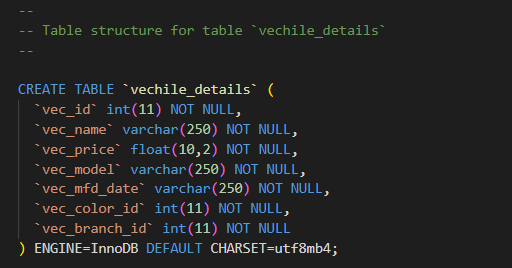


### Delete:

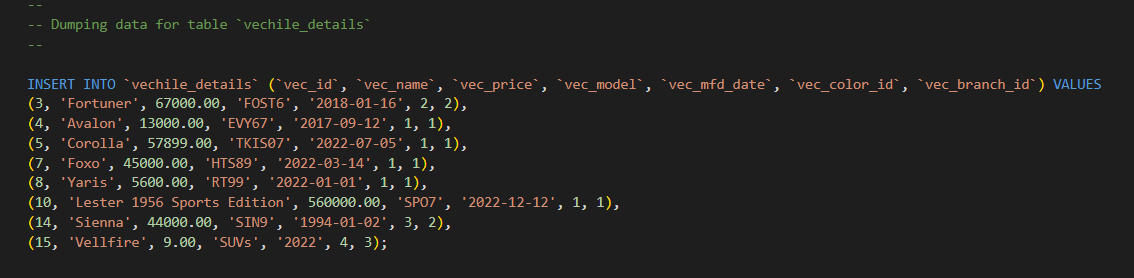


### Create table (vehicles):

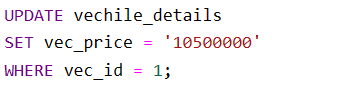
Query for creating vehicles table.



### Insert:



### Update:



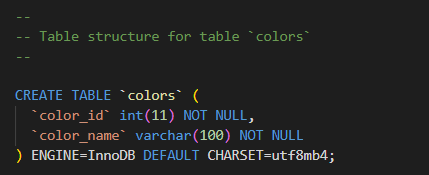
### Select:



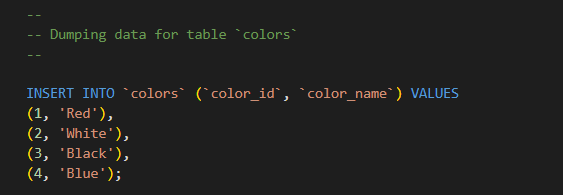
### Delete:



### Create table (colors)



### Insert:

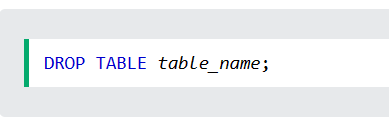


### Select:



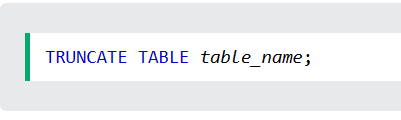
### Drop table:

Sometime if you need to delete some table the you can use syntax given below.



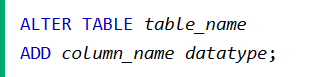
### Truncate table:

When you nee to delete all data from table then you can use syntax given below:

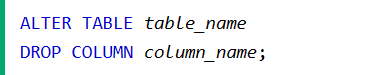


### Alter statement:

When you need column then you can add column by using this syntax is given below.



When you need to delete column of table then uh can use syntax given below.



# Conclusion:

I have mention some basic SQL statement as create, insert, update, select, and delete. I discussed step by step of every SQL statement and query which I was implemented in my project. I discussed that SQL statement that I was implement into the relational database.

# P4:

Test the system against user and system requirements.

# Introduction:

As a database developer I will test the system against user and system requirement too. I will performing test like unit test, integration and stress test for my project for Toyota vehicle management system.

# Testing:

"Database testing entails using values that a desktop or web application has retrieved from the database. Records recorded in the database should match the data on the user interface.” (Anon., n.d.) It consists two types of testing which are listed below:

## Software testing:

"Software testing is the process or technique of identifying errors in a software application or program so that the application functions in accordance with the end user's requirements. Software testing is the process of examining a system to find any flaws, gaps, or unmet requirements in comparison to the real requirements. Functional testing and non-functional testing are the two main categories of software testing. When testing efforts should begin: To minimize the time and money needed to fix bugs and generate software that is bug-free so that it can be given to the client, testing should begin as soon as possible. Testing, however, can begin in the requirements gathering phase of the Software Development Life Cycle (SDLC) and continue until the software is in use in production. Additionally, it relies on the development model being applied. For instance, testing begins in the testing phase of the w

aterfall model, which is much lower in the tree, but testing occurs concurrently with development in the V-model. When testing should end: A software program can never be completely bug-free. (Anon., 2018)

## Database testing:

The collection of connected files known as a database. Databases may be heterogeneous, meaning they may not all be of the same type across multiple servers. Database testing is essentially a multi-layered process. Database systems typically have four layers: the database itself, the business layer, the data access layer, and the user interface (UI) layer. It's crucial to test at each of these levels for a reliable database system. Web software frequently uses different backbends to access the database, or heterogeneous databases. The main testing that involves examining tables, performing queries, and writing processes is database testing. Testing can be carried out on desktop or web applications that use databases like SQL or Oracle. Numerous initiatives, including those in banking, finance, and health insurance, call for thorough database testing. (Anon., n.d.) Database testing consist these type.

1. Database validation testing
2. Database integration testing
3. Performance related to database
4. Testing of procedure, triggers and functions

# Testing Methodologies:

## Black box testing:

"Black-box testing, commonly referred to as functional testing, treats the software being tested as a black box without knowing anything about its internal workings. Software interfaces are used in tests to check that they function as expected. Tests ought to pass even if internals are modified as long as interface functionality stays the same. The tester is aware of what the software ought to accomplish, but he or she is unaware of how it actually does it. In traditional organizations with distinct departments for testers, black-box testing is the sort of testing that is most frequently utilized. This is especially true when the testers lack coding expertise and find it difficult to comprehend the code. It offers an outside viewpoint of the software that is being tested. (Dijkstra, n.d.)

## White box testing:

White box testing is the examination of the internal infrastructure and coding of a software system. The main areas of concentration are enhancing security, streamlining inputs and outputs inside the program, and enhancing design and usability. There are several other names for white box testing, including clear box, open box, structural, transparent, code-based, and glass. It is one of two components of the software testing strategy known as "box testing." Black box testing, which is its opposite, entails testing from an outside or end-user perspective. White box testing, on the other hand, focuses on internal testing and is based on an application's internal operations. (Anon., n.d.)

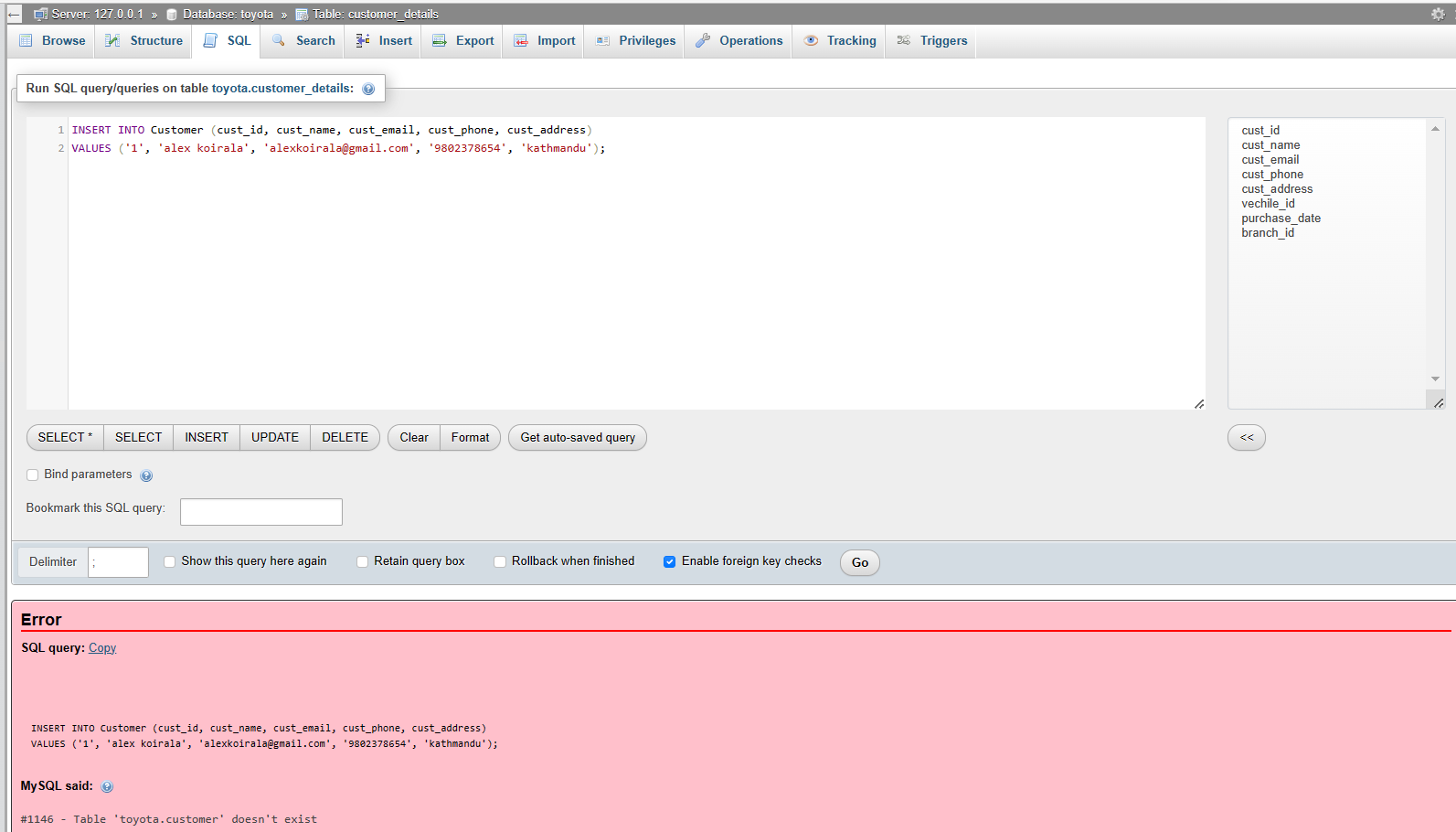
# Types of testing:

## Unit testing:

"Unit testing is a type of software testing where only specific program units or components are tested. The goal is to confirm that each piece of software operates as intended. The smallest testable component of any software is called a unit. It typically has one or more inputs and one output. An individual program, function, process, etc. can all be considered units in procedural programming. The smallest unit in object-oriented programming is a method, which may be a part of a base/super class, abstract class, or derived/child class. (Some treat an application module as a unit. This should be avoided because that module presumably contains a lot of different units.) Unit testing is aided by the usage of drivers, stubs, mock/fake objects, and unit testing frameworks.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Test date | What to test | Expected output | Actual output |
| 1 | 2022/11/18 | Repetition of primary key twice, | Primary key must not exist as same | Data not insert. |

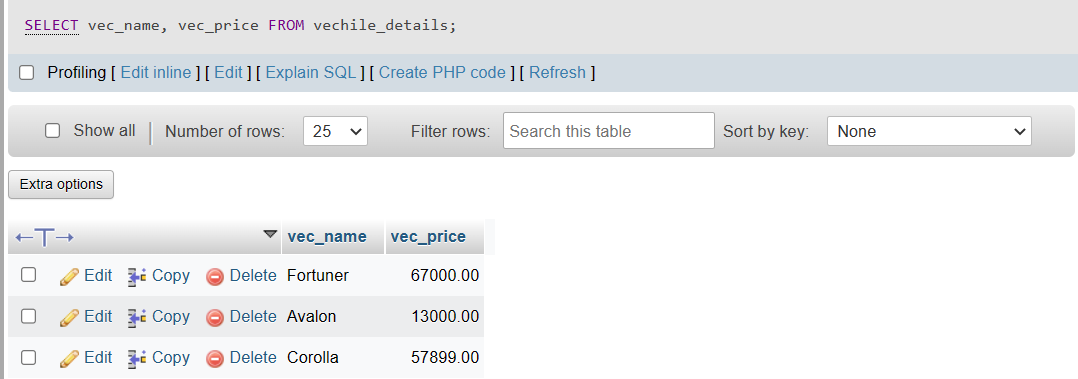
Screenshot of test



The entered id does not accept, that entered id is already exists so it does not accept twice.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 2022/11/18 | Selecting column from tables | Selected data will show | Only selected data  Shown. |

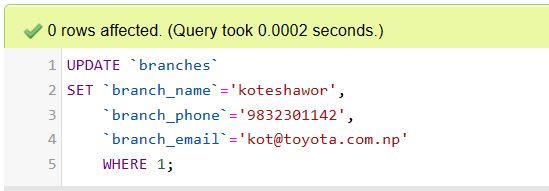
Screenshot of test



Selected data only show that was expected.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | 2022/11/18 | Updating data in the table | Data will change/update | Data updated |

Screenshot of test.



Data updated not any rows affected.

## Integration Testing:

Software testing at the level of integration testing involves the combination and group testing of separate components. This level of testing seeks to identify issues with how integrated units interact with one another. Integration testing is aided by the usage of test drivers and test stubs. (Anon., n.d.)

## Stress testing:

"Stress testing is the process of ascertaining if a computer, network, software, or device can continue to function effectively under challenging circumstances. The procedure may entail quantitative lab tests, including counting the number of mistakes or system failures. The phrase also describes the qualitative assessment of elements like accessibility or defense against denial-of-service (DOS) assaults. Performance testing, which is a more broad approach, is frequently conducted in conjunction with stress testing. (Rouse, 2007)

# Review:

I have designed this Toyota vehicle management system by placing the requirements like customers, branch, vehicles, color, and user To make this project I used many applications like Visual Studio, Xampp, MySQL, draw.io.

# Software Review:

Database management software, usually referred to as database software, can be divided into specialized divisions such relational and non-relational databases. Both database types are frequently used to store client information, product details, financial data, and other business information. (Anon., n.d.)

Conclusion:

I've completed many test kinds for this work, and I've provided screenshots as proof. I've carried out several test activities, including unit tests, integration tests, and stress tests. My test report was supported by a test log that recorded the date, the subject of the test, the expected results, and screenshots of the test's execution.

# P5:

# Introduction:

In this task, I am going to produce a technical and user documentation. I discuss about introduction of documentation, constraints, assumption for user manual.

# **User documentation:**

"In the development of computer hardware and software products, documentation is the data that informs the user of the product. It includes of online resources and product technical manuals (including online versions of the technical manuals and help facility descriptions). The phrase is occasionally also used to refer to the primary sources of product knowledge, such as design documents, in-depth code comments, white papers, and blackboard session notes.(Rouse, 2007) As a database developer I have developed a database design web GUI (Toyota vehicle management system ).

# Constraint/Risks:

Many kinds of errors can happen when developing applications. Some errors are unforeseen, and we must be concerned for the elderly. Coding is like the biggest risk we have to deal with because it might cause the entire data set to fail.

# **Assumption:**

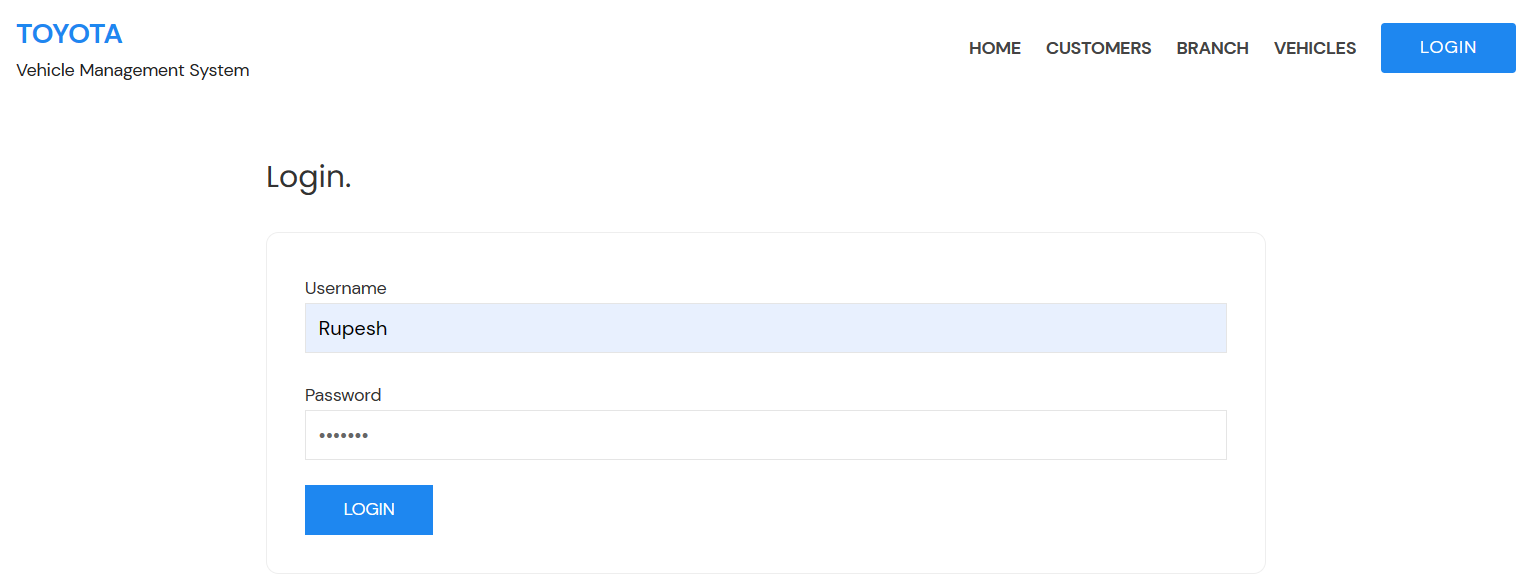
Any issue could arise while the application is being developed. As a result, we cannot be certain of the project. Because they might be useful and we must constantly be prepared for solutions, it is necessary that we are familiar with the tools we utilized to construct the application.

# **User Manual:**

It functions similarly to a map or guide that is simple to understand and apply. It includes all relevant information on application using step-by-step techniques. This is the user guide I made for my application to make it simple to use.

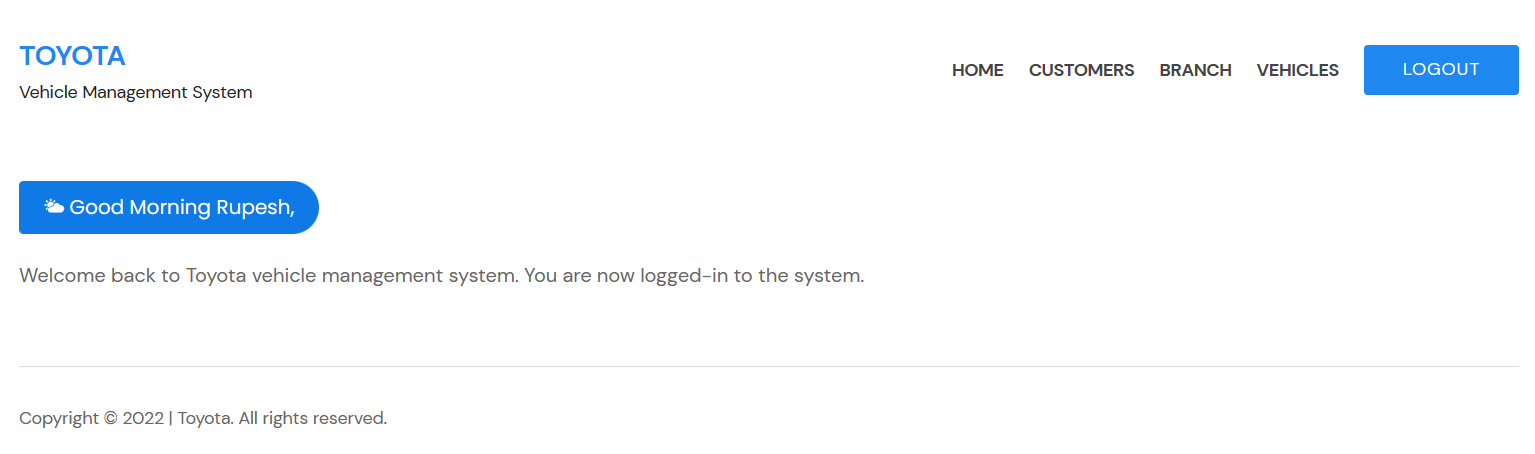
## Login dashboard:

Please enter correct username and password then click on login bottom for login the dashboard.



## Home dashboard:

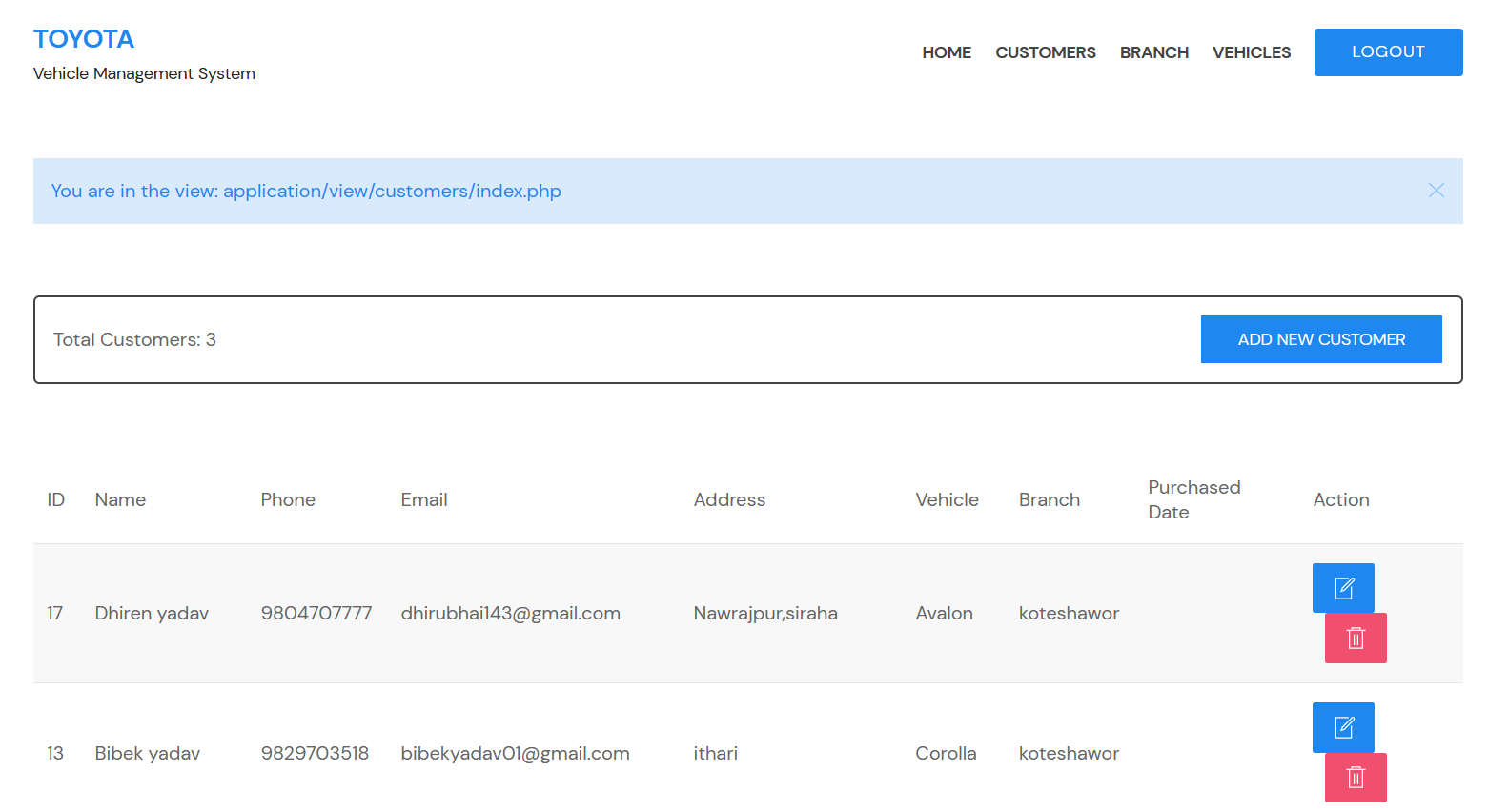
In this page, you can find numerous tables for entering the data, displays once you log in. You can find home, customer, branch, vehicles and logout bottom. When you will complete your task then you can logout your dashboard. For logout your dash board click on logout bottom.



## **Customer dashboard:**

In the customer dashboard, you can see customer list with their details. you can get some action bottom like add new customer, edit and delete.

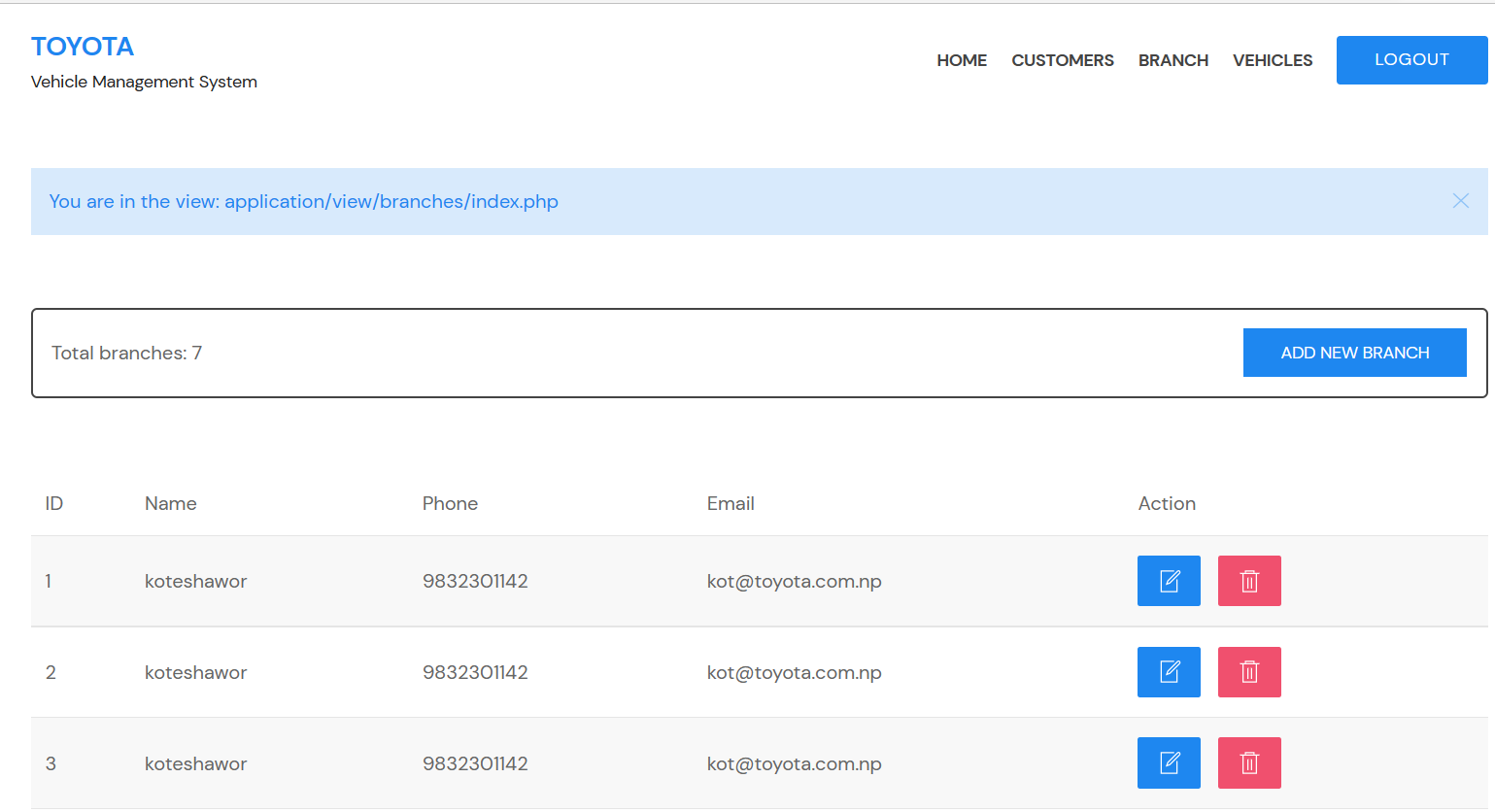
* To add new customer click on add new customer.
* To edit or update data click on edit action bottom.
* To delete click on delete action bottom.



## Branch dashboard:

In the branch dashboard, you can see branch list with their details. you can get some action bottom like add new branch, edit and delete.

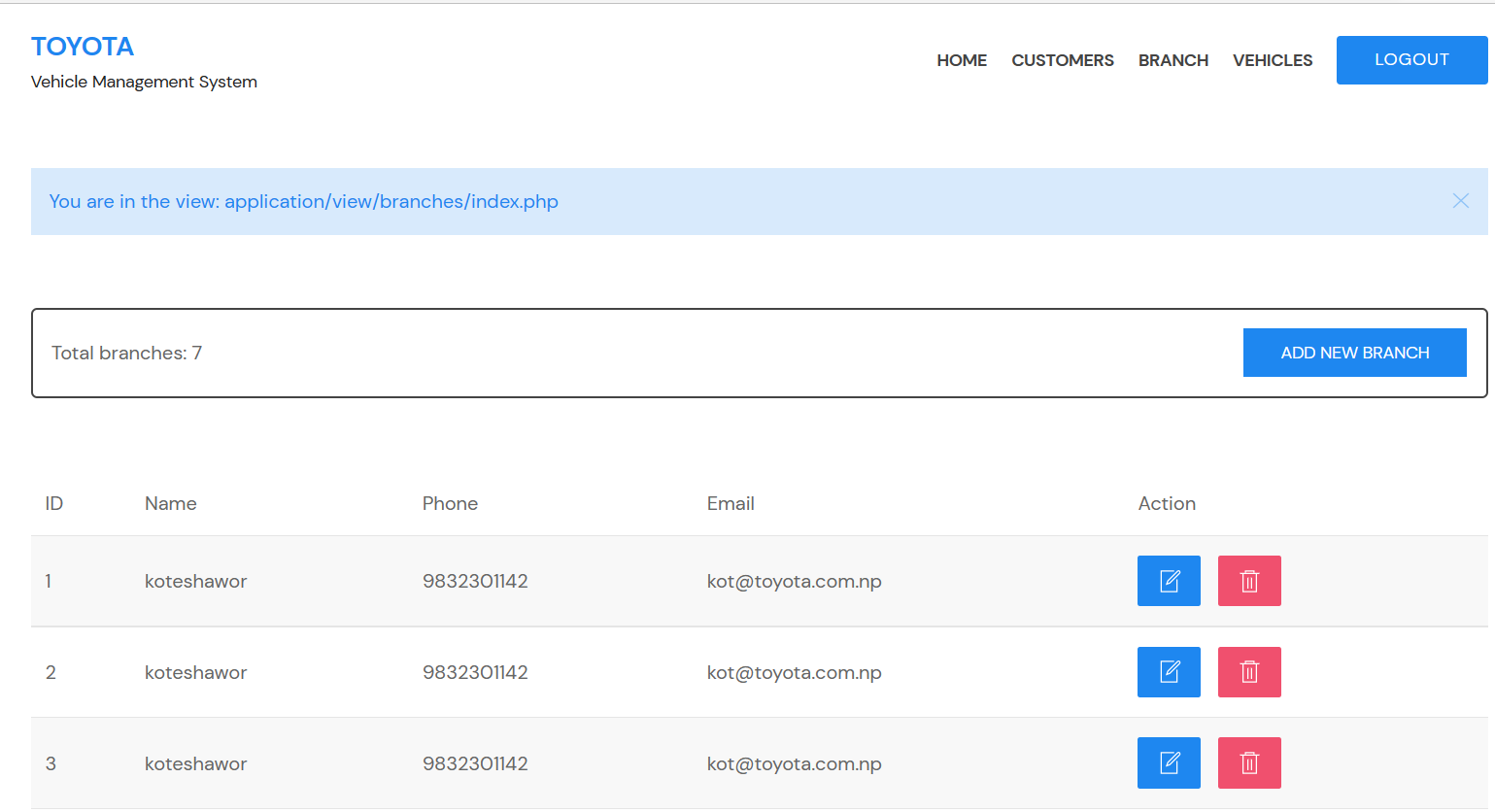
* For add new branch click on add new branch bottom.
* For edit or update data click on edit action bottom.
* For delete click on delete action bottom.



## Vehicle dashboard:

In the vehicle dashboard, you can see vehicle list with their details. you can get some action bottom like add new branch, edit and delete.

* For add new vehicle click on add new vehicle bottom.
* For edit or update data click on edit action bottom.
* For delete click on delete action bottom.



# **Conclusion:**

I produced a documentation in this manner, which includes an introduction, an overview, limitations, and an assumption. I had written a user guide to aid in using the web GUI. I had discussed everything above.

# References:

Microsoft (2021) *Visual studio code - code editing. redefined*, *RSS*. Microsoft. Available at: https://code.visualstudio.com/ (Accessed: December 31, 2022).

*What is XAMPP? Complete guide to what is XAMPP* (2021) *EDUCBA*. Available at: https://www.educba.com/what-is-xampp/ (Accessed: December 31, 2022).

*Creating mysql database with XAMPP - javatpoint* (no date) *www.javatpoint.com*. Available at: https://www.javatpoint.com/creating-mysql-database-with-xampp (Accessed: December 31, 2022).

*What is a database management system: DBMS: Monitoring* (2022) *AppDynamics*. AppDynamics. Available at: https://www.appdynamics.com/topics/database-management-systems (Accessed: December 31, 2022).

*What is an entity relationship diagram (ERD)?* (no date) *Lucidchart*. Available at: https://www.lucidchart.com/pages/er-diagrams (Accessed: December 31, 2022).

Manisha Jena • April 7th, Integration, M.G.on D. and Manisha Jena on Database Management Systems (2022) *What is data validation?: It's working and importance simplified 101 - learn*, *Hevo*. Available at: https://hevodata.com/learn/data-validation/ (Accessed: December 31, 2022).

Peterson, R. (2022) *What is normalization in DBMS (SQL)? 1NF, 2NF, 3NF example*, *Guru99*. Available at: https://www.guru99.com/database-normalization.html (Accessed: December 31, 2022).

*DB - primary key: A primary key is a field in a table which uniquely identifies each row/record in a database table. primary keys must contain unique: Course hero* (no date) *db -contraints*. Available at: https://www.coursehero.com/file/18658886/db/ (Accessed: December 31, 2022).

*Data definition language: How to create, Alter,Drop Command in DDL?* (2021) *EDUCBA*. Available at: https://www.educba.com/data-definition-language/ (Accessed: December 31, 2022).

*Data manipulation language: A quick glance of DML & DML commands* (2021) *EDUCBA*. Available at: https://www.educba.com/data-manipulation-language/ (Accessed: December 31, 2022).

(no date) *SQL drop,tuncate,alter statement*. Available at: https://www.w3schools.com/sql/sql\_drop,truncate,alter\_table.asp (Accessed: December 31, 2022).

*Database testing* (no date) *Tutorials Point*. Available at: https://www.tutorialspoint.com/software\_testing\_dictionary/database\_testing.htm (Accessed: December 31, 2022).