## 1.1 ABOUT THE PROJECT

The project named Online Two-wheeler Booking Platform is to create a fully computerized system implemented to get booking through online, store customers, staff, vendors details and also to reduce the manual process as online is more efficient. The customers can book their bikes by making booking through the company website after creating an account. Then the customers can view details of the bike that they are planning to book either by choosing from the categories displayed on the home page or browsing over to the all productions button from the navigation bar. It also allows the customer to book one or more bikes where total payment and total booking can be viewed. If the customer wishes to book a bike, they can book their desired bike by paying the booking fee amount either through debit or credit card by entering the card details. Later the details of the booked bike along with the details of the customer would be forwarded to the respective brand's vendor's dashboard and the vendor will contact the customer to proceed with the necessary documentation work required. The platform stores data about the customers and the vendors on the database. The main aim of the project is to provide an effective working platform to computerize and simplify the whole Two-Wheeler booking experience for customers.

The proposed project on Online Two-Wheeler booking platform is an effort to solve the various problems in purchasing the bikes directly from the showroom. The implementation of the project establishes a systematic and reliable service along with a well-maintained management of the booking records. Using this system, one can book bikes through internet.

The project is computerizing the hectic process of going to a showroom and spending days to book a bike and deals with the necessary booking transaction. The customer can avoid delay by using the system. It is very smooth, efficient and fast system. The current states of the system can be understood by using the report form which is very helpful for the administrator. For the staff, they can handle their dealings very easily. The manual system had many limitations, however through the process of computerization, the manual system overcomes those limitations.

**Advantages for the Project**

1. The system accepts customer booking for bikes through online.
2. It allows the customers to create their own account and stores their details in the database.
3. Once the customer book a bike successfully the details of the customer including the booking amount will be transacted to the respective brand vendor.
4. It makes the customers easy to know the details about the various vehicle options available to them and choose one hassle free, all withing the comfort of one's home.
5. The system is very fast and efficient application for the working of Two-Wheeler booking.

## 2.1 SYSTEM STUDY

### 2.1.1 Existing system

In existing system, to manage the bike booking we use a manual system. In the present system all the activities are performed manually. All data entry is performed by writing data into the bookings, paper documents. The record is prepared manually, so there is a chance for occurring errors and the calculations are not so accurate. Various information such as staff details, vendor details, bike details, payment details are handled manually. When there is need for retrieving details searching is unavoidable, this is a difficult task searching the records manually.

The existing system uses manual method for whole process. This requires a lot of hard work and time consumption to complete the task. This may include human errors. In this existing system, it is difficult to retrieve some particular information. Also, all the records are stored manually and it is tedious task. As a result, the security of these records is always a challenging task. Hence, the computerization of the system of record maintenance is the only solution to reduce the shortcomings of existing system.

**Disadvantages of existing system**

- Time consuming.
- Lots of travelling required.
- Less interactive.
- Requires more man power.
- High maintenance costs.

### 2.1.2 Proposed System

The proposed system is interactive, highly user friendly and designed exclusively for the Online Two-Wheeler booking. The Two-Wheeler booking Management System is a database system used to store the information regarding staff details, vehicle details, vendor details, customer details, payment details, etc.

All the operations and activities related to The Online Two-Wheeler booking Platform can be carried out efficiently. The system manages a well-organized database for storing the resources that they are providing by the Online Two-Wheeler booking. This help us to eliminate the entering of invalid data. Most problems of manual system can be solved by this system.

The computerization of the system allows the easy maintenance of the details. Large amount of data can be stored easily. In addition, updating and other changes can be done easily. The information can be retrieved with high speed and accuracy. The use of GUI oriented software makes the system user friendly.

**Advantages of Proposed System**

- High processing speed.
- Easy to retrieve old records from respective dashboards for the management team.
- We can analyse staff details, vehicle details, payment details.
- Minimal errors.
- Greater portability.
- User friendly.
- Reduced workload.

### 2.1.3 Feasibility Study

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, time and effort spent on it.

Study lets the developers to see the future of the project and its usefulness. Finding out whether a new system is required or not.

The study is carried out to the best system that meet performance requirement.

This entails identification, description and evaluation of candidate system and selection of the best system for the job. It simply identifies whether the proposed system is feasible to the organization or not.

There are three aspects in the feasibility study portion of the preliminary investigation

i) Technical feasibility
ii) Economic feasibility
iii) Operational feasibility

#### 2.1.3.1 Technical Feasibility

The Online Two-Wheeler booking Management System must be evaluated from technical view point first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedure having identified an outline system, the investigation must go on to suggest the type of equipment, required method of developing the system, method of running the system once it has been designed. The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed with latest technology. There are only minimal constraints involved in this project.

### 2.1.3.2 Economic Feasibility

Here an evaluation of development cost is weighted against the ultimate income or benefit derived from the developed system. The cost for the development of the project has been evaluated and we want to check that the cost does not exceed beneficial cost of the system. The economic and financial analysis is used for evaluating the effectiveness of the candidate system. This project also undergone economic feasibility study and found that it is feasible. So, the cost for development does not exceed its beneficial cost. This brought to as the conclusion that the system is economically feasible in the context.

### 2.1.3.3 Operational Feasibility

In operational feasibility the entire application is checked whether the system will be used if it is developed and implemented. Also, it is checked whether there will be resistance from user that may undermine the possible application benefits. There is no barrier for implementing the system. The system also helps to access the information immediately as need arises. Thus, the system is found to be operational feasible.

**2.2 User Characteristics**

This software has three users

1. Administrator
2. Staff
3. Customer
4. Vendor

**Administrator:**

The Administrator is the primary user who has the most or maximum control over the software. Administrator administrates over the entire activities of the system and has full control over what all happens in the system. He/she is the only user who can add, view, change or delete details of an existing staff, category, subcategory. However, they have no rights over the customer details as it is added by the customer itself.

**Staff:**

Staff is the secondary user. He/she has limited privileges when compared to the administrator. They have the functions like managing vendors and vehicles, booking etc. General interaction with the system is done with the help of the staff.

**Customer:**

The customer will have little privileges when compared to both the administrator and staff. They can place booking for the vehicle and book them. They can also find their desired vehicle using the system and get all details about the vehicle.

## 2.3 SYSTEM SPECIFICATION

### 2.3.1 Hardware Specification

The selection of hardware and software configuration is very important task related to system development.

| | |
|---|---|
| Processor | Intel Pentium IV (3.0 GHz) or above |
| RAM | 1 GB |
| Hard Disk | 80 GB and above |
| Key Board | Normal or multimedia |
| Monitor | 15''CRT or LCD monitor |
| Mouse | Compatible Mouse |

### 2.3.2 Software Specification

| | |
|---|---|
| Operating System | Windows |
| Front | PHP |
| Back End | SQL Server 2008 |

### 2.3.3 About The Software Tools

FRONT END SPECIFICATION: PHP

PHP is a powerful server-side scripting language for creating dynamic and interactive websites. PHP is the widely-used, free, and efficient alternative to competitors such as Microsoft's ASP. PHP is perfectly suited for Web development and can be embedded directly into the HTML code. The PHP syntax is very similar to Perl and C. PHP is often used together with Apache (web server) on various operating systems. It also supports ISAPI and can be used with Microsoft's IIS on Windows.PHP is an embedded scripting language that is excellent for creating dynamic Web sites based on database content or different characteristics of browsers. It is available when you have a Departmental (Web Central) Publishing account, a Faculty Publishing account, a Student Organization Publishing account, or if you use Custom Web Publishing.

**Features**

- Allows you to build templates to ease site maintenance
- Serve different content to users based on their browser, IP address, date and time, or numerous other characteristics
- Enables connections with databases such as MySQL
- Build discussion forums or Web-based email programs

**Characteristics of PHP**

- Objected Oriented
- Complied and Interpreted
- Portable
- Distributed
- Secure

BACK END SPECIFICATION: SQL Server 2008

SQL Server 2008 is an integrated database management system and analysis solution that delivers increased security, scalability and availability to enterprise data and analytical applications, while making them easier to build, deploy and manage. It is comprehensive software that enables to reliably manage mission – critical information and confidently run today's increasingly complex business applications. SQL Server 2008 allows gaining greater insight and achieving faster results for a competitive advantage. The key capabilities of SQL Server 2008 are the following:

High Availability: Ensures business continuity with the highest levels of system availability through technologies that protect data against costly human errors and minimize disaster recovery downtime.

Performance and Scalability: Deliver an infrastructure that has proven record in handling today's large amounts of data and critical enterprise workloads.

Security: Provides a secure environment to address privacy and compliance requirements with built in features that protect data against unauthorized access.

Manageability: Manages infrastructure with automated diagnostics, tuning and configuration to reduce operational costs while reducing maintenance and easily managing very large amounts of data.

Developer Roomivity: Build and Deploy critical business ready applications more quickly by improving developer roomivity and reducing project lifestyle times.

Business Intelligence: Gain deeper insight into the business with integrated comprehensive analysis and reporting for enhanced decision making.

**Introduction**

The most creative and challenging phase of the system development is system design. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Design goes through the logical and physical stages of development.

In designing a new system, the system analyst must have a clear understanding of the objectives, which the design is aiming to fulfil. The first step is to determine how the output is to be produced and in what format. Second input data and master files have to be designed to meet the requirements of the proposed output. The operational phases are handled through program construction and testing. The point is to choose such an environment in which we will be able to operate with in a convenient and easy way. The most creative and challenging phase of the system development is system design. It provides the understanding and the procedural details necessary for implementing the system recommended in the feasibility study. The analyst should understand the requirements of the user and develop the system accordingly. Design goes through the logical and physical stages of development. In designing a new system, the system analyst must have a clear understanding of the objectives, which the design is aiming to fulfil. The application program as an interface between the users and the database should be an accurate reflection of the database on the screen; hence a well analyzed and defined structure is needed. The user interface should be easy to understand and operate on for the users. The first step is to determine how the output is to be produced and in what format it has to be produced. Second, input data along with the master files have to be designed to meet the requirements of the proposed output.

The analyst must ensure that the interaction between the user and the interface is simple to understand. To ensure that everything works properly and as it has been expected, test performances have to be done upon the system functionality. Testing plays an important role in identifying any minor errors after system design and it will be corrected.

## 3.1 MODULES AND DESCRIPTION

Online Two-Wheeler booking Platform is a web-based shopping system which provides us facilities to manage the activities taking place in Online vehicle booking. This system is developed to manage the work flow in an Online vehicle booking. There are 7 modules in this project. They are:

1.0 Staff Management

2.0 Customer Management

3.0 Vendor Management

4.0 Product Management

- 4.1 Category Management
- 4.2 Sub-category Management
- 4.3 Brand Management
- 4.4 Vehicle Management

5.0 Cart Management

6.0 Booking Management

7.0 Payment Management

**1.0 Staff Management**

This module deals with the managing the staffs working in the System. Staff Management module also deals with adding new staff to the system, updating the details of the existing staff, maintaining the status of the staff being registered to the system.

**2.0 Customer Management**

Customer Registration Module stores the details of all the customers who register in the webpage. Customers enter the system after registration and their details get stored in this module. They can edit and view their details when they need to.

**3.0 Vendor Management**

Vendor management module stores the details of all the showrooms who supply vehicles to the webpage. Vendor Detail can be added, edited or viewed only by the Administrator or staff**.**

**4.0 Product Management**

The main aim of this module is to manage all the products in an organized manner. This module stores the details of products that are displayed in the webpage for the customers.

**4.1 Category Management**

Different vehicles have different categories. The Vehicle Management Module is used to store the details of vehicles regarding their category. The examples for categories are Riding Style, Cubic Capacity, etc.

**4.2 Sub-Category Management**

Vehicle Sub-Category Management Module is used to store the details of two-wheeler in sub-category level. The sub-categories include Motorbikes, Scooters etc.

**4.3 Brand Management**

The Brand Management Module stores the details of two-wheeler on the basis of their brand.

**4.4. Vehicle Management**

This module keeps record off all the two-wheeler on the basis of category, sub-category and brand. This module stores the two-wheeler into the item table by providing the product details.

**5.0 Cart Management**

This module coordinates cart information with customer details. A vehicle can be added to the cart before purchase. This cart can be later edited or viewed only by the customer. Cart management also have 2 tables:

• Cart Master: Contains the details of customer after reference to customer table.

• Cart Child: When a customer adds the same vehicle into the cart twice the cart child refers item table and cart master table and stores a cart child id.

**6.0 Booking Management**

This module manages the list of vehicles booked by the customers. This module keeps track of all the bookings placed by the customers and make it clear for the vendor to act upon.

**7.0 Payment Management**

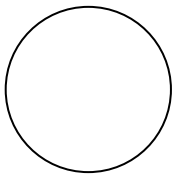
This module keeps the details of all the payments done by the customer. Here the customer can add and view their payment methods.

## 3.2 DATA FLOW DIAGRAM (DFD)

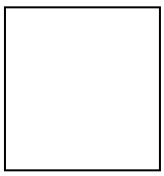
A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are central tool and the basis from which the other components are developed. The transformation of data from input to output, trough processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams.

A DFD is also known as a "bubble chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So, it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

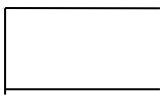
In the DFD, there are four symbols



Process that transforms data flow



Source or Destination of data



Data store



Data flow

**Rules for drawing data flow diagrams**

Rule 1: Establish the context of the data flow diagram by identifying all of the net input and output data flows.

Rule 2: Select a starting point for drawing the DFD.

Rule 3: Give meaningful labels to all data flow lines.

Rule 4: Label all processes with action verbs that relate input and output data flows.

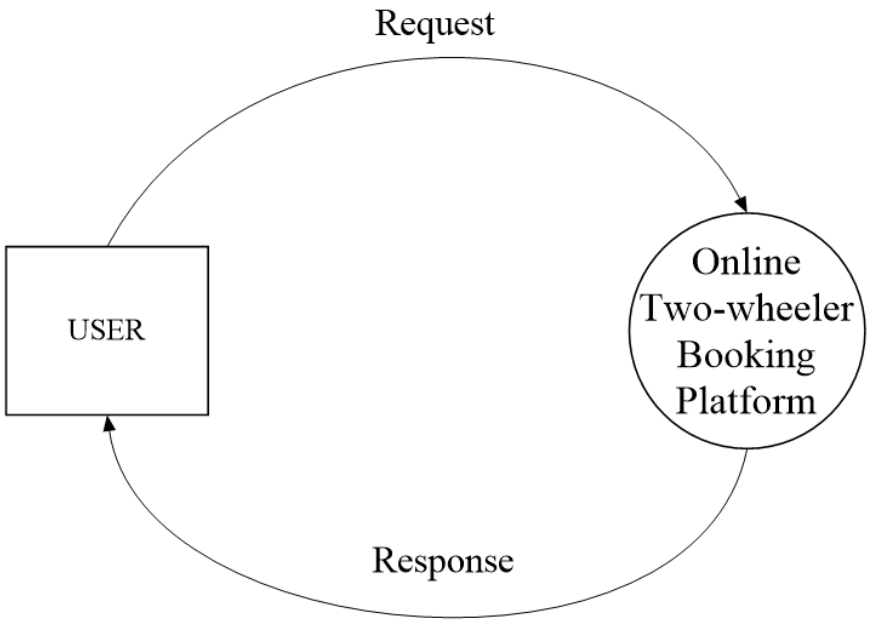
Rule 5: Omit insignificant functions routinely handled in the programming process.

Rule 6: Do not include control or flow of control information.

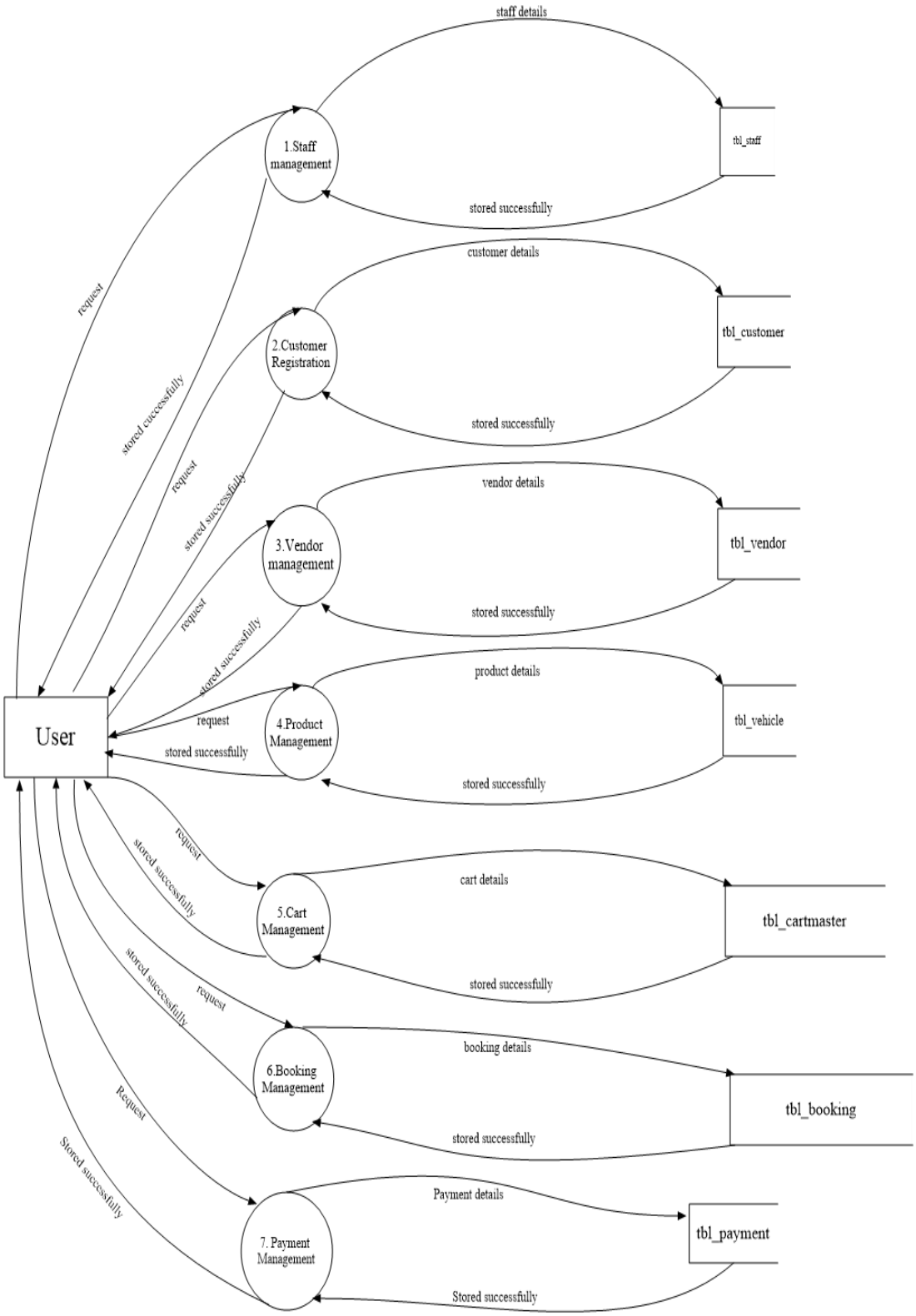
Rule 7: Do not try to put too much information in one DFD.

Rule 8: Be prepared to start over

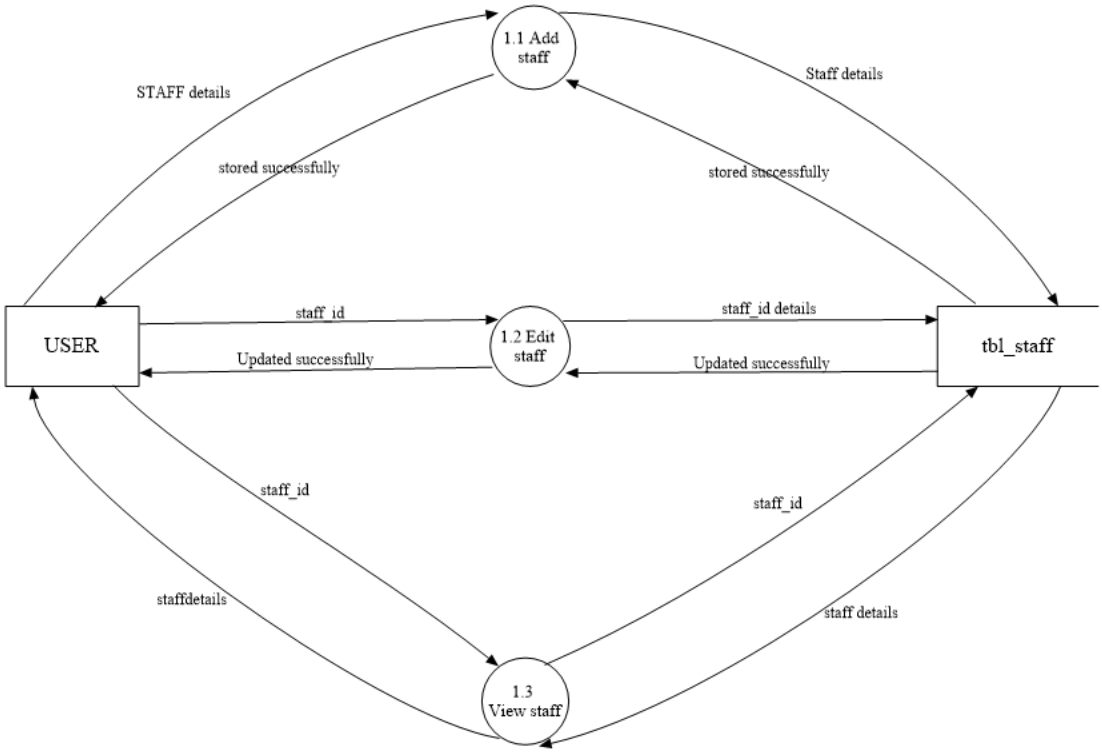
**Level 0 DFD Showing Online Two-Wheeler Booking Platform**



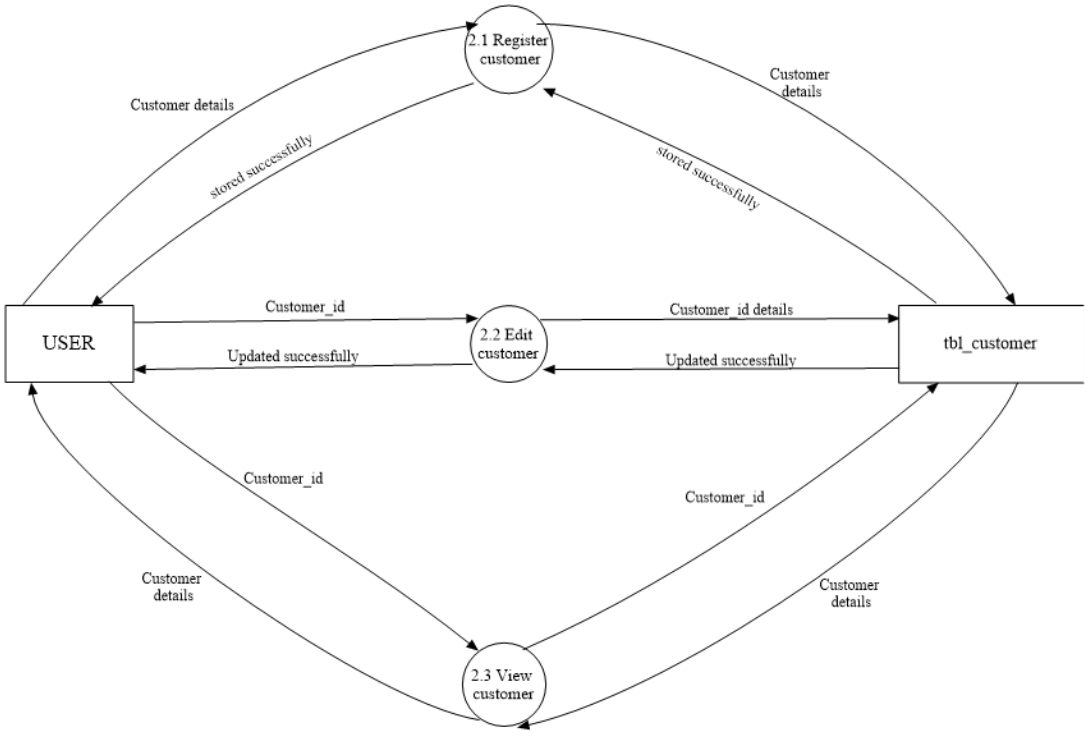
**Level 1 DFD Showing Online Two-Wheeler Booking Platform**



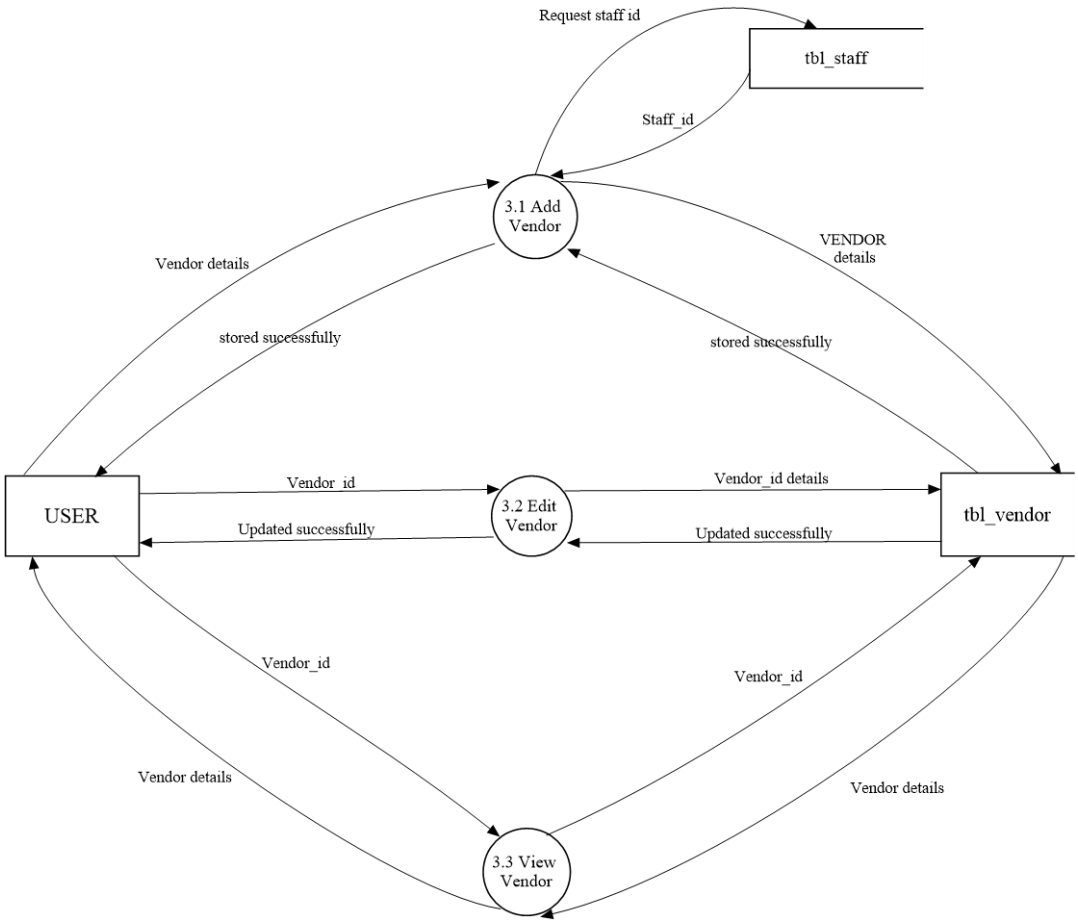
## Level 2 DFD Showing Staff Management



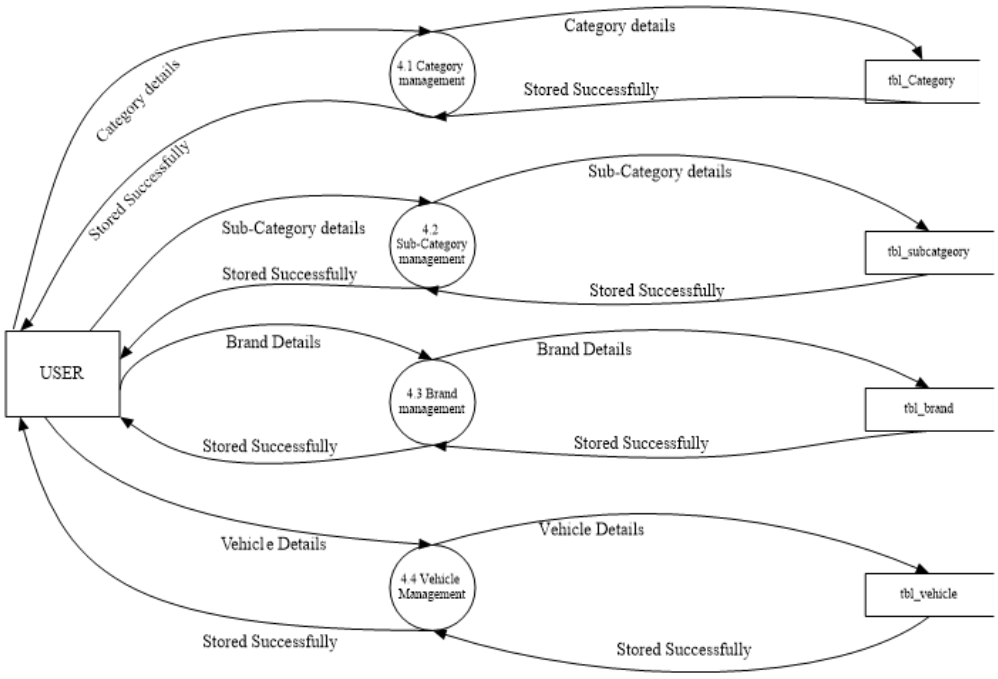

## Level 2 DFD Showing Customer Registration



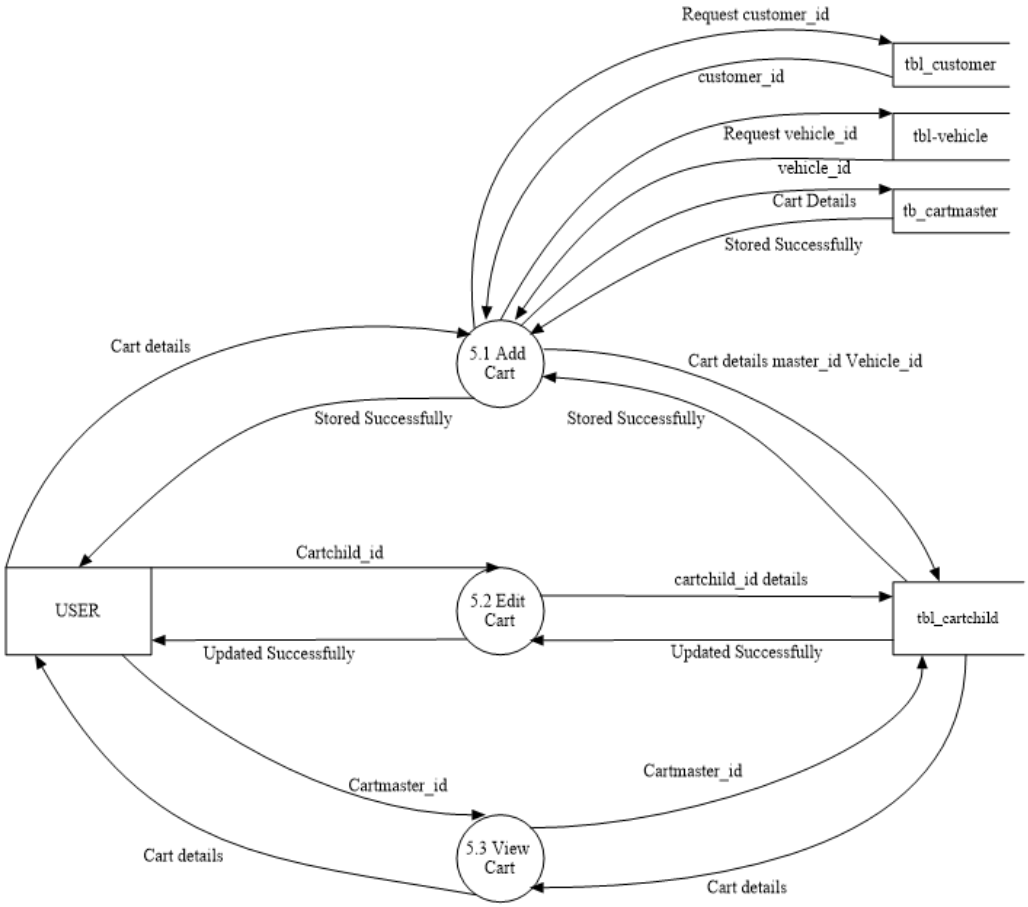
## Level 2 DFD Showing Vendor Management



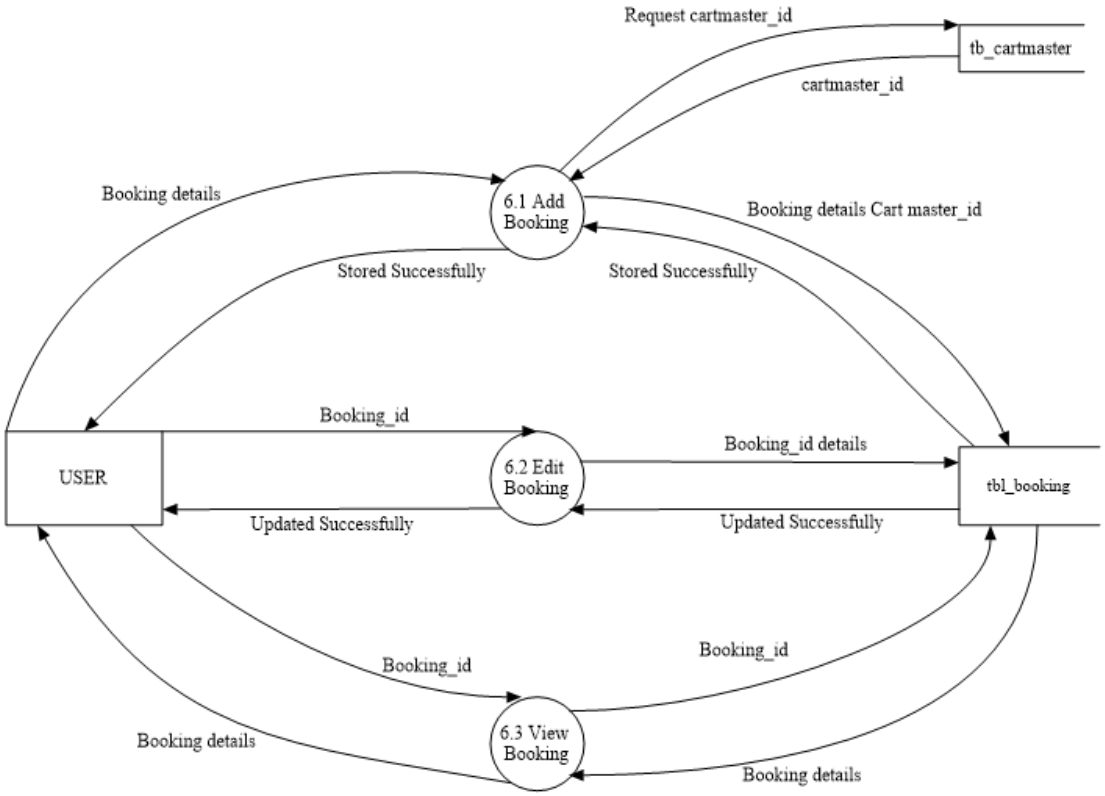
## Level 2 DFD Showing Product Management



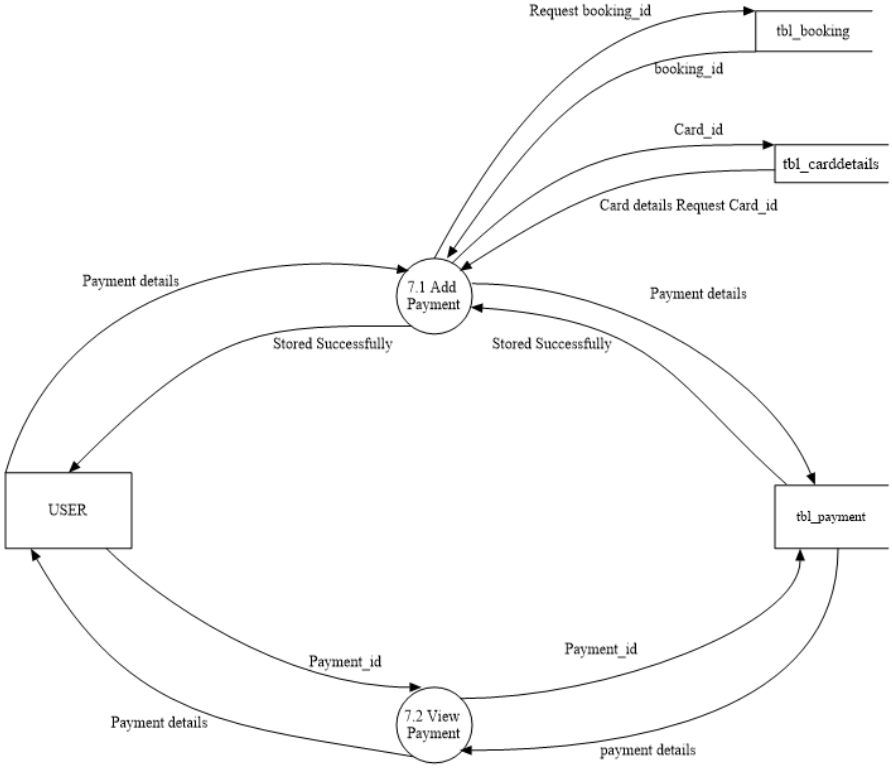

## Level 2 DFD Showing Cart Management



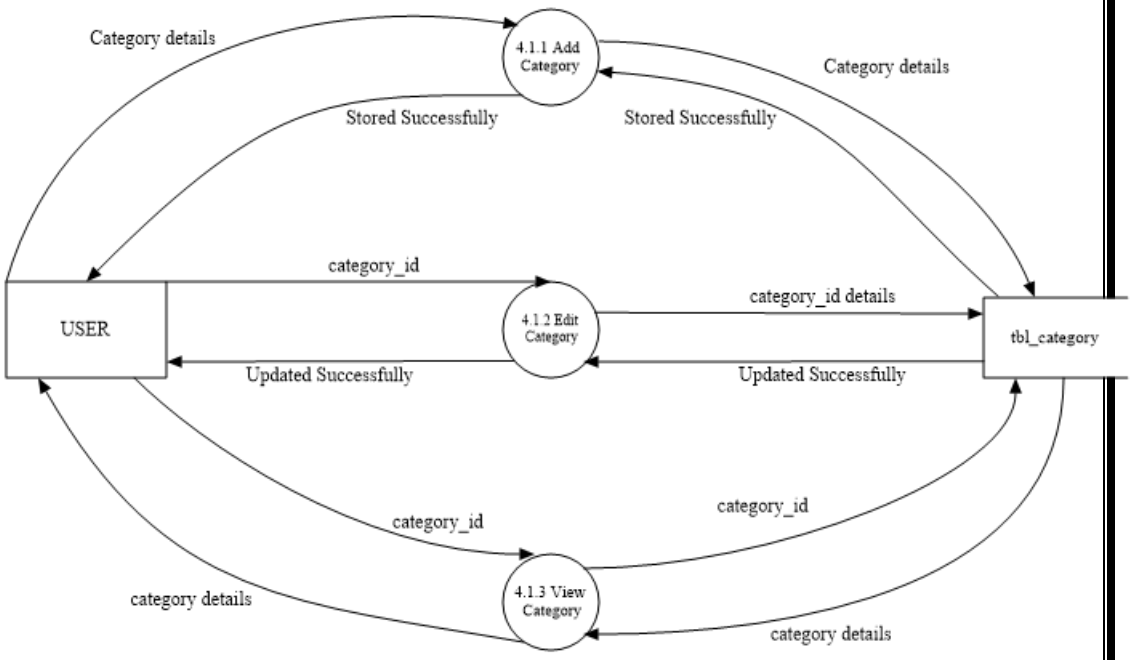
## Level 2 DFD Showing Booking Management



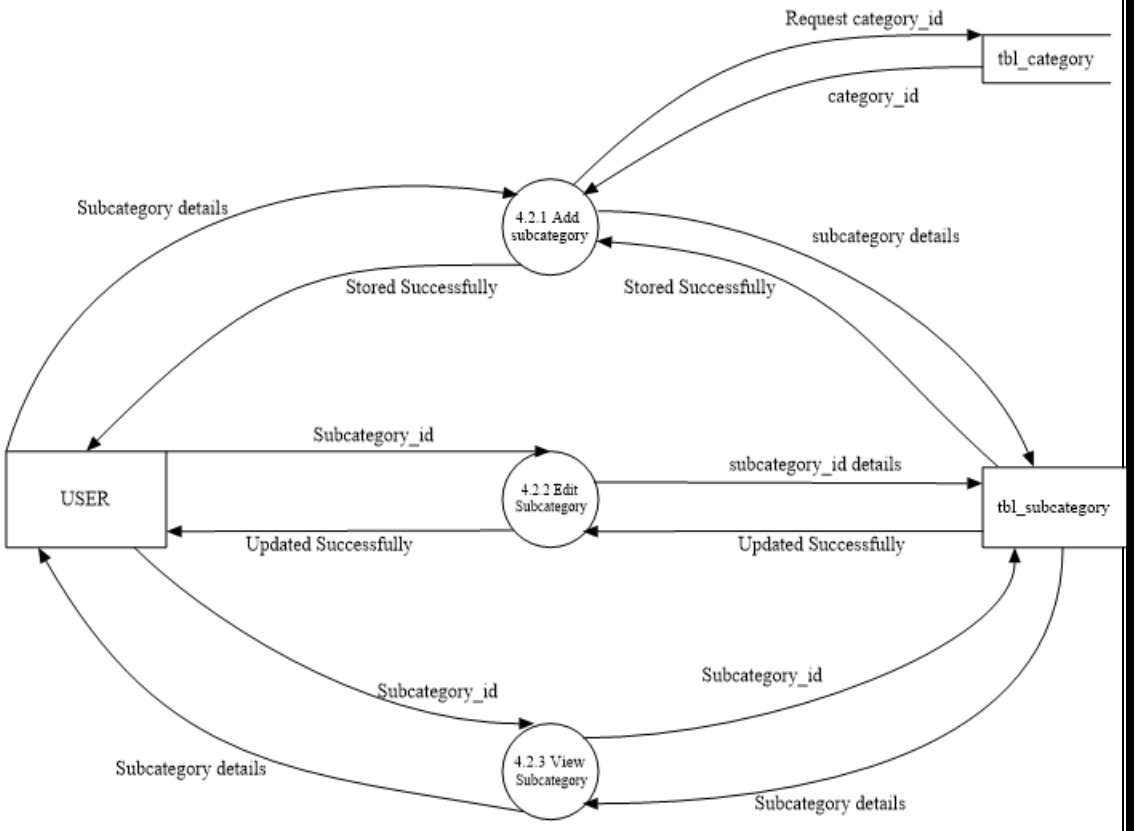

## Level 2 DFD Showing Payment Management



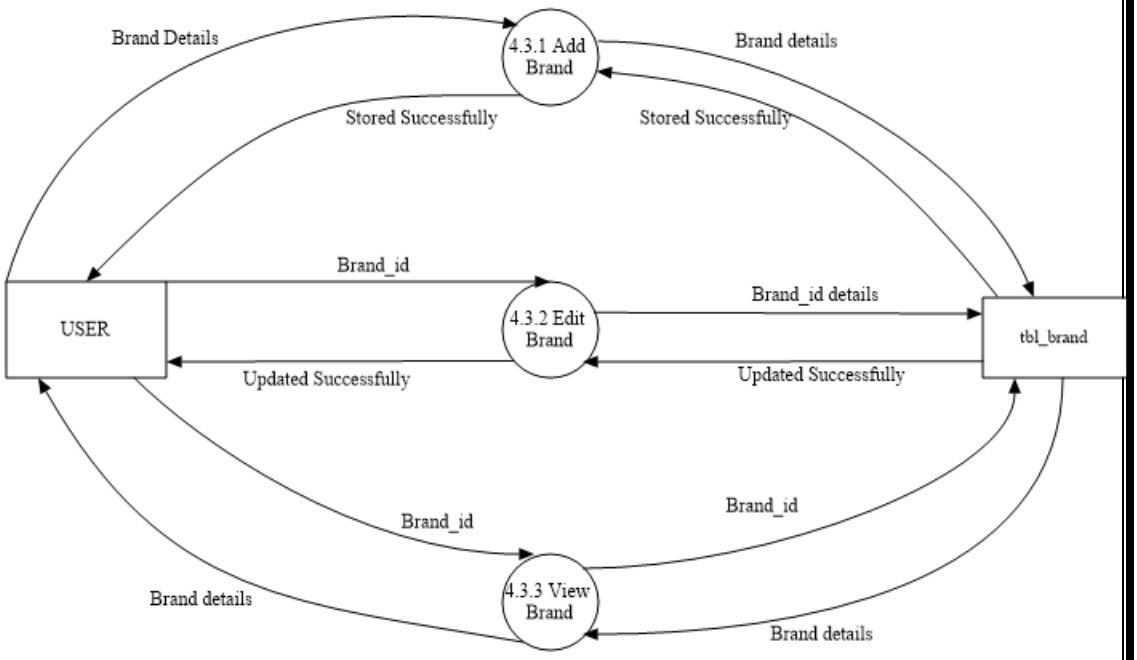
## Level 3 DFD Showing Category Management



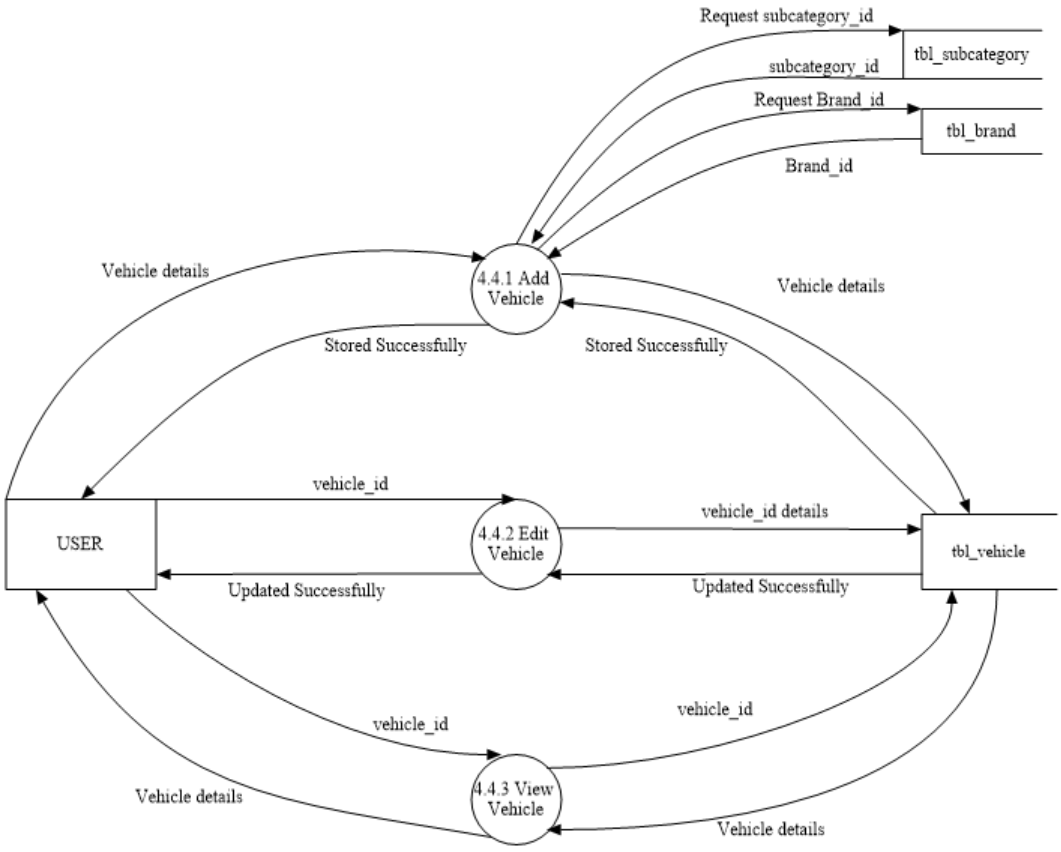

## Level 3 DFD Showing Sub-Category Management



## Level 3 DFD Showing Brand Management

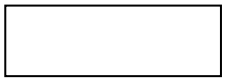



## Level 3 DFD Showing Vehicle Management

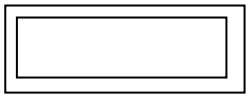


## 3.3 ENTITY RELATIONSHIP DIAGRAM

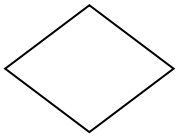
The ER model is a conceptual data model that views the real world as a construct of entities and associations or relationships between entities. A basic component of the model is the Entity-Relationship diagram, which is used to visually represent data objects. The ER modelling technique is frequently used for the conceptual design of database applications and many database applications and many database design tools employ its concepts.



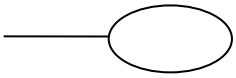
Entity Type



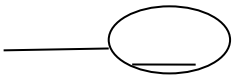
Weak Entity Type



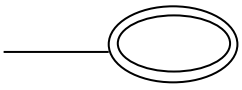
Relationship Type



Attribute

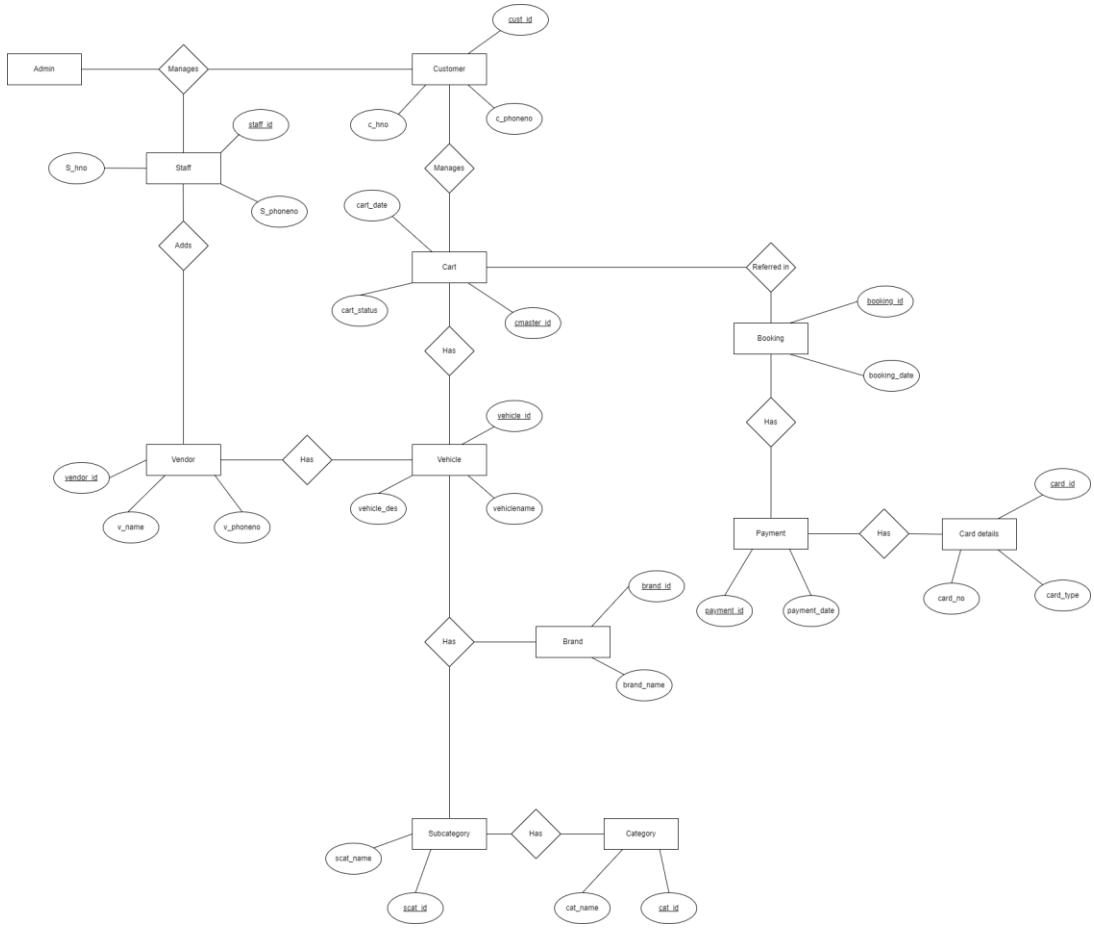


Key attribute



Multivalued Attribute

# ER DIAGRAM


cust_id
Admin
Manages
Customer
staff_id
c_hno
c_phoneno
S_hno
Staff
Manages
S_phoneno
cart_date
Adds
Cart
Referred in
booking_id
cart_status
cmaster_id
Booking
Has
booking_date
vehicle_id
Has
Vendor
Has
Vehicle
vendor_id
vehicle_des
vehiclename
v_name
v_phoneno
card_id
Payment
Has
Card details
brand_id
payment_id
payment_date
card_no
card_type
Has
Brand
brand_name
Subcategory
Has
Category
scat_name
scat_id
cat_name
cat_id

### 4.1 Input Design

Input design is the process of converting a user-oriented description of the inputs to a computer-based business system into a programmer-oriented specification. The quality of system input determines the quality of system output. Input specification describes the manner in which data enter the system for processing. Input design features can ensure the reliability of the system and produce result from accurate data or they can result in the roomion of errors. The input design also determines whether the user can interact efficiently with the system.

Input design requires consideration of the needs of the data entry operator. Three data entry considerations are:

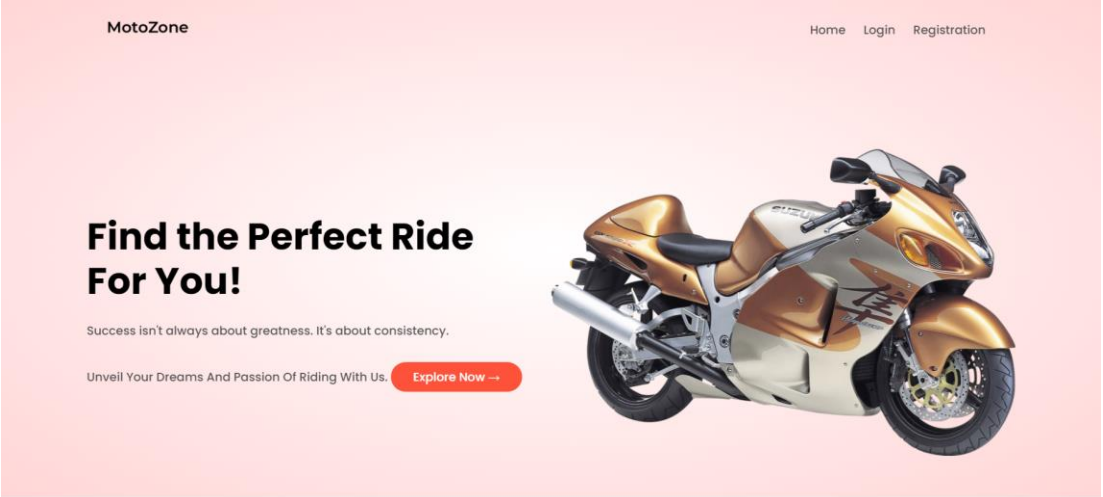
- The field length must be documented
- The sequence of fields must match the sequence of the fields on the source document.
- The data format must be identified to the data entry operator.

In our system almost all inputs are being taken from the databases. To provide adequate inputs we have to select necessary values from the databases and arrange it to the appropriate controls.

Inaccurate input data are the most common cause of errors in data processing. Errors entered by data entry can be controlled by input design. Input design is the process of converting user-oriented inputs to a computer-based format. There are three major approaches for entering data into the computer. They are menus, formatted forms and prompts. A menu is a selection list that simplifies computer data access or entry. Instead of remembering what to enter, the user choices from a list of option. A formatted form is a pre-printed form or a template that request the user to enter data in appropriate location. It is a fill-in-the-blank type form. The form is flashed on the screen as a unit. In prompt the system displays one enquiry at a time, asking the user for a response.

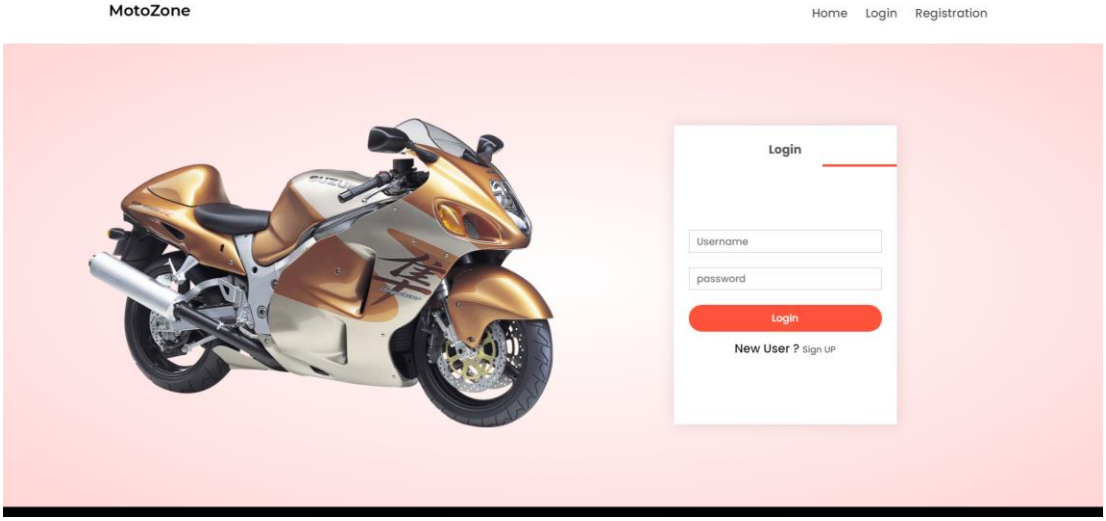
**Home Page Form**

**Description:** This form shows the Home page for all the users.



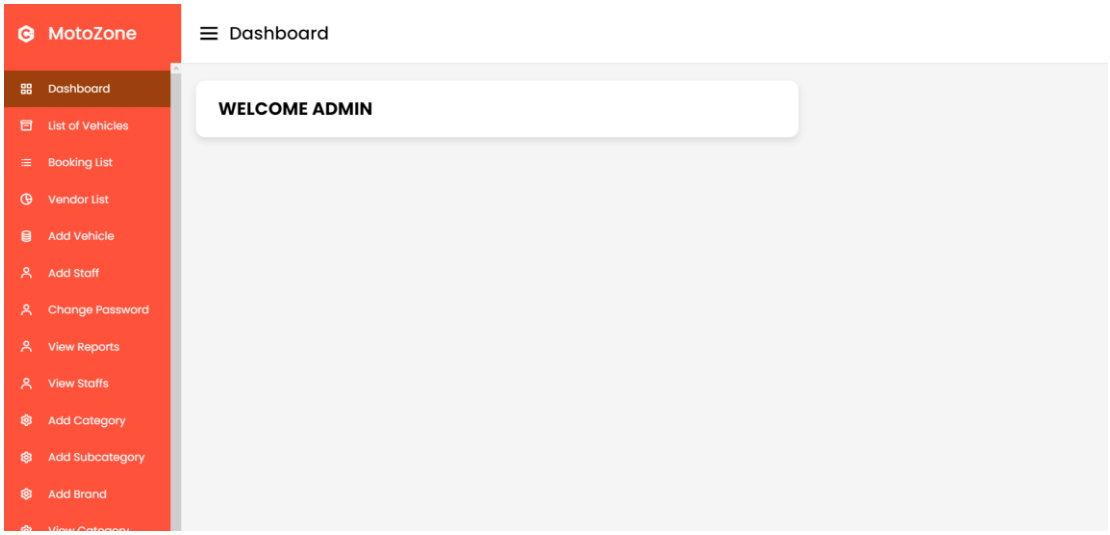

**Login Form**

**Description:** This form shows the login page for all the users.



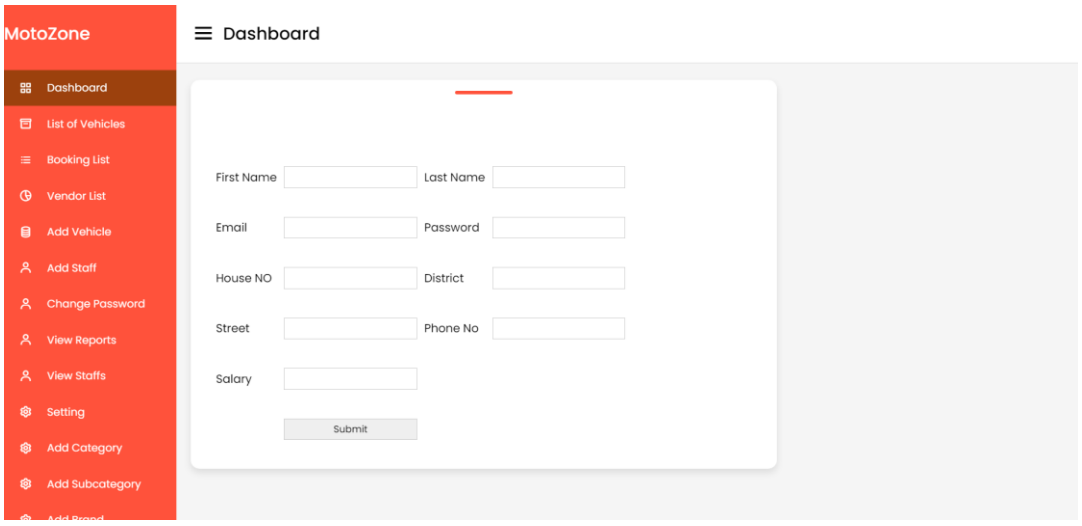
## Admin Form

**Description:** This form shows the Admin page.



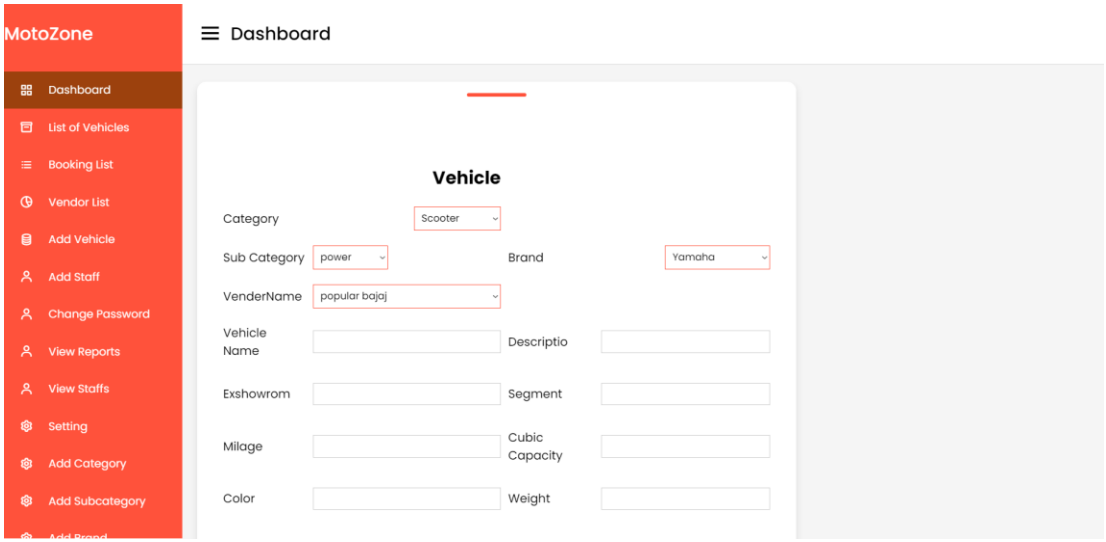

## Admin Add Staff Form

**Description:** This form is used to add new staff to the shop.



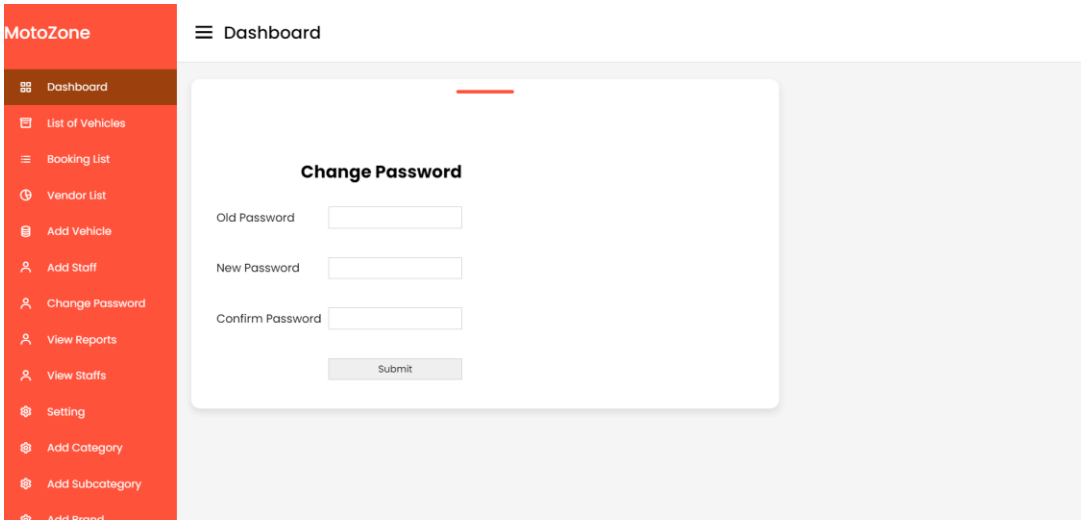
### Admin Add Vehicle Form

**Description:** This form is used to add new vehicle to the shop.



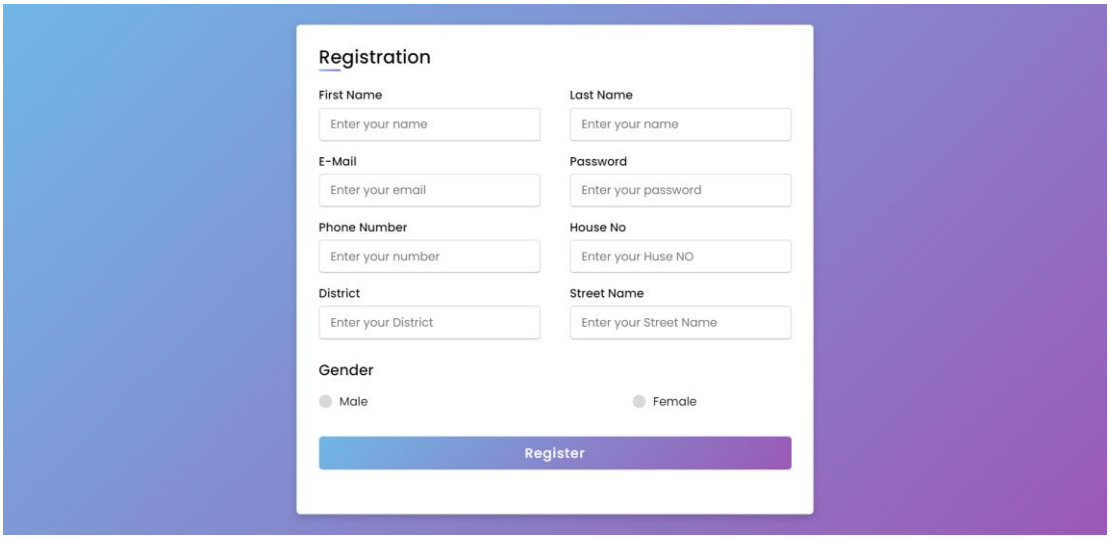

### Admin Change Password Form

**Description:** This form is used to change the password of the admin account.



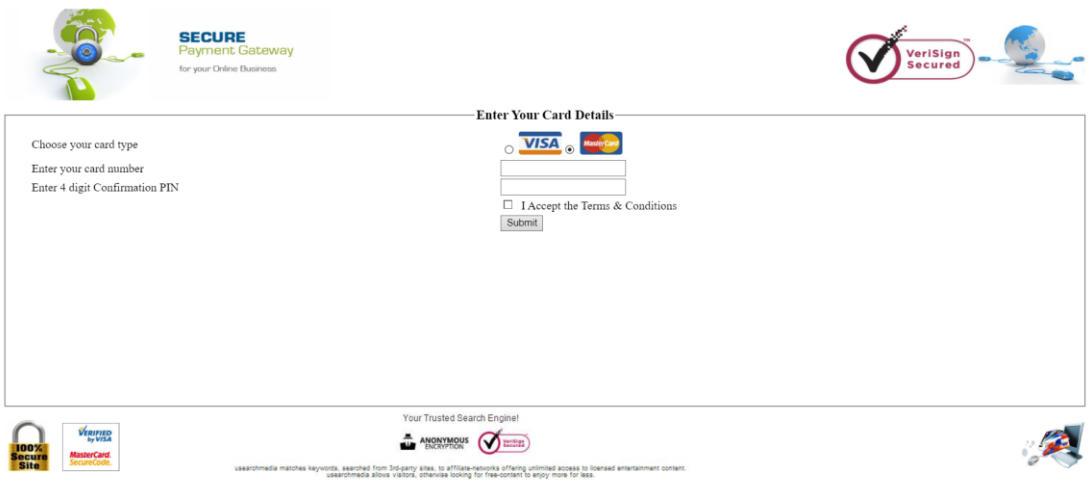
**Customer Registration Form**

**Description:** This form shows the registration page for new customers.



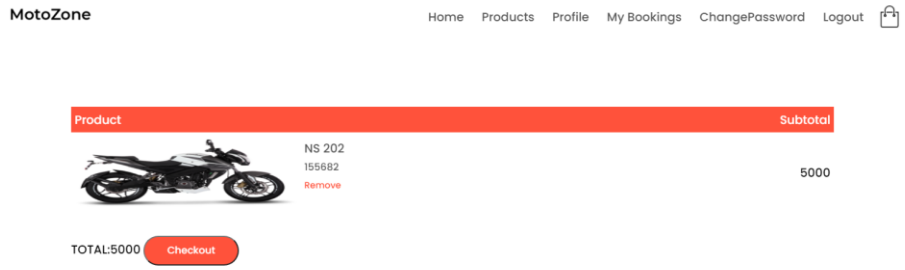

**Payment Form**

**Description:** Customer payment is done through this form.



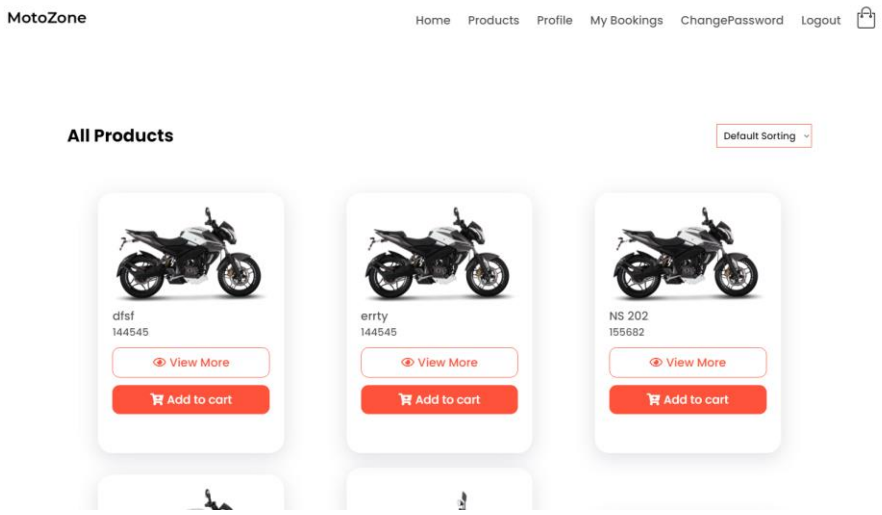
## Cart Form

**Description:** This form is used to show the cart of the user



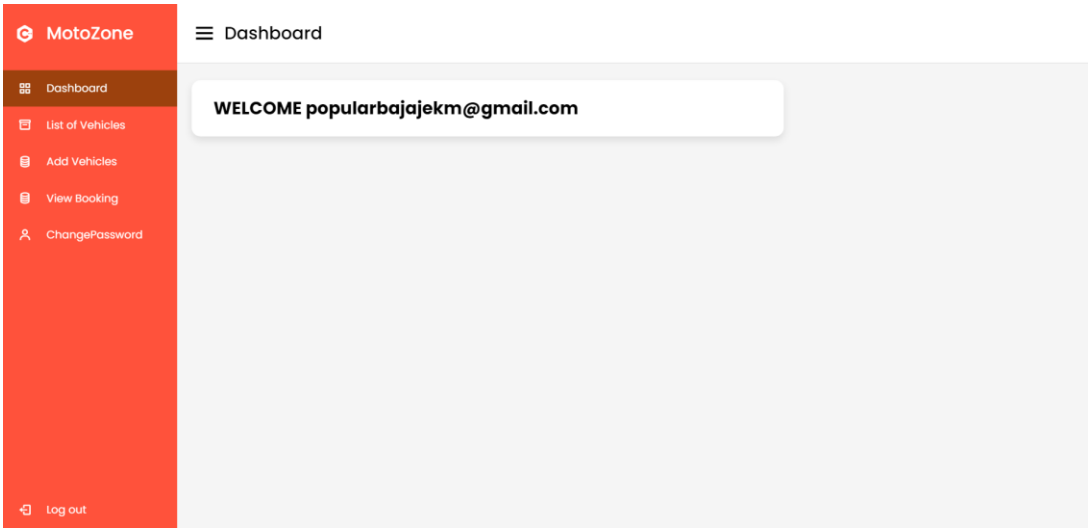

## Customer Home Form

**Description:** This form is shown to the logged in customer.



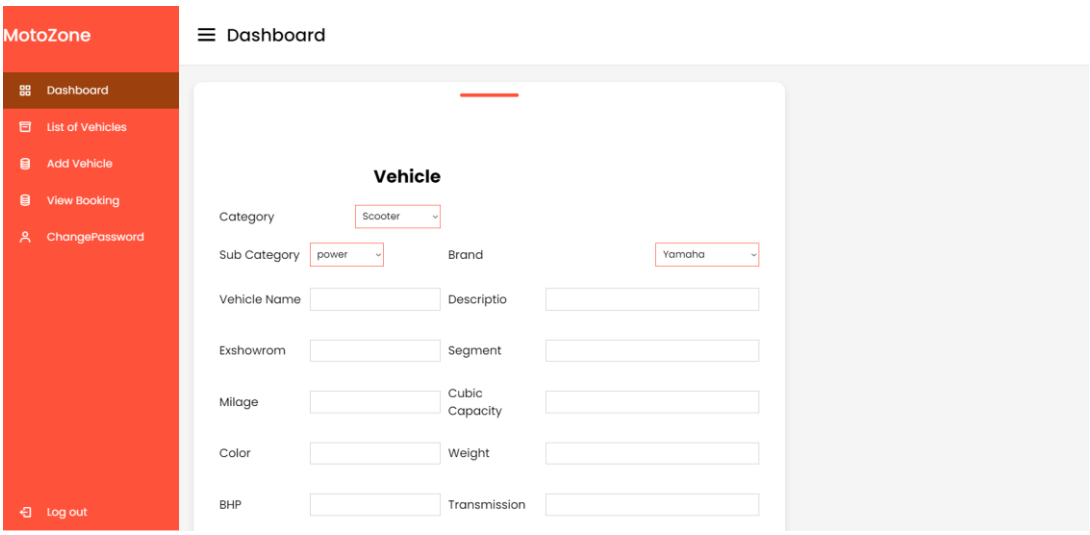
### Vendor Form

**Description:** This form is used as the dashboard for the vendor.



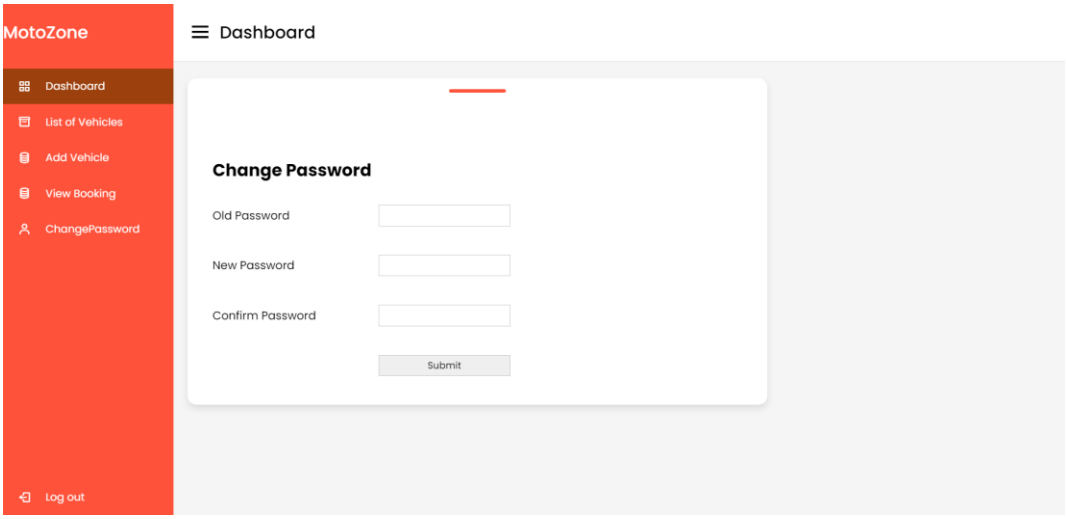

### Vendor Add Vehicle Form

**Description:** This form is used to add a new vehicle to the website as vendor.



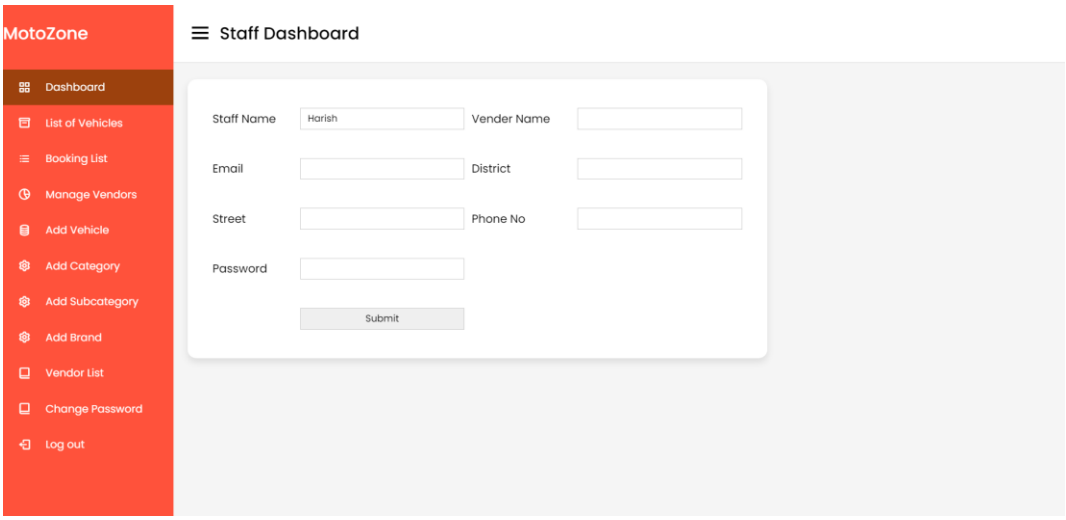
### Vendor Change Password Form

**Description:** This form is used to change the password of the current vendor.



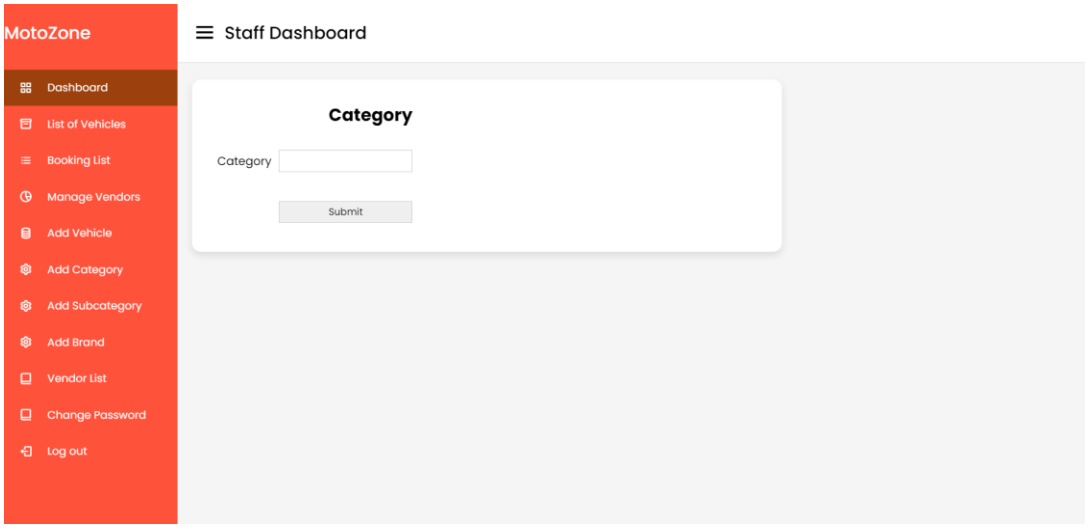

### Staff Add Vendor Form

**Description:** This form is used to add a new vendor to the website.



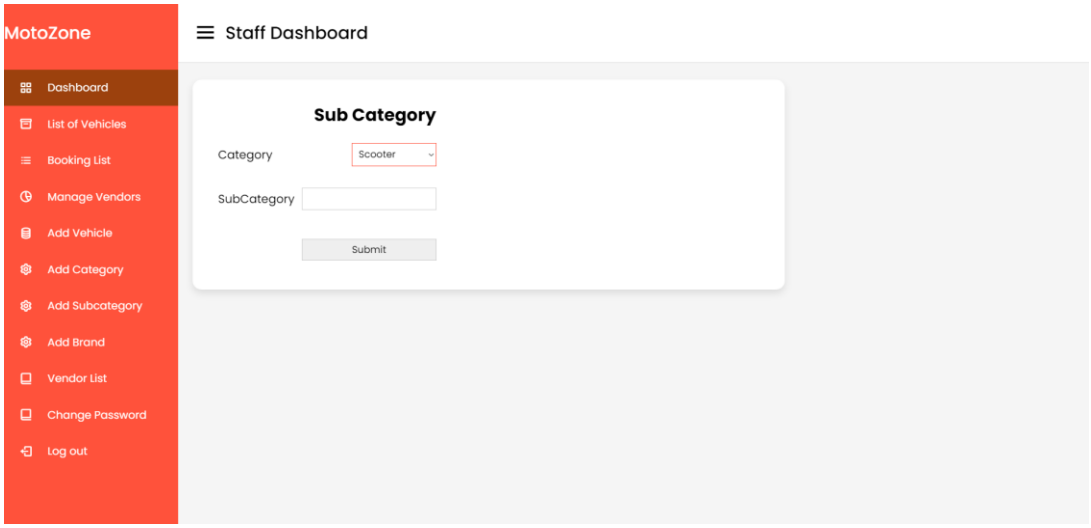
### Staff Add Category Form

**Description:** This form is used to add a new Category of vehicle to the website.



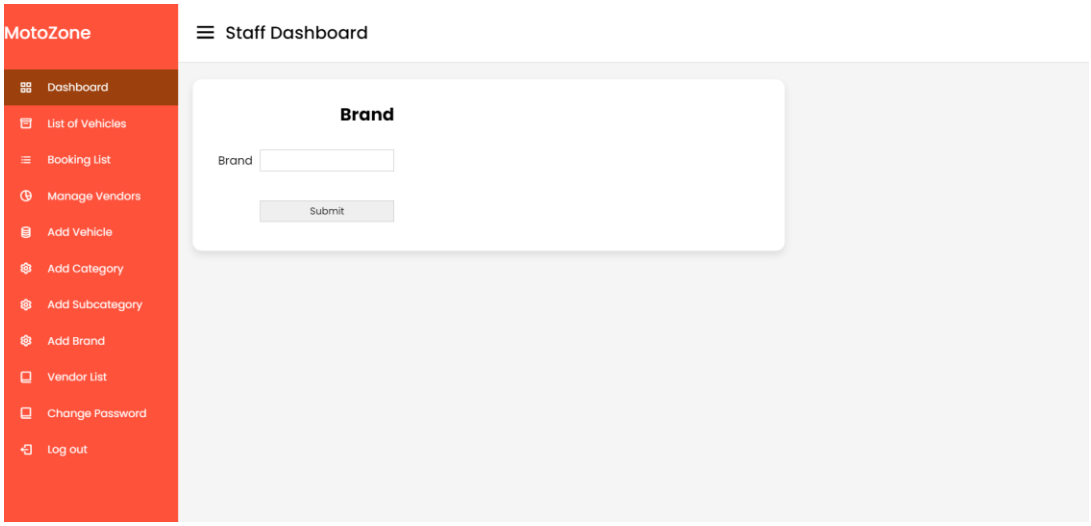

### Staff Add Sub-Category Form

**Description:** This form is used to add a new Sub-Category of vehicle to the website.



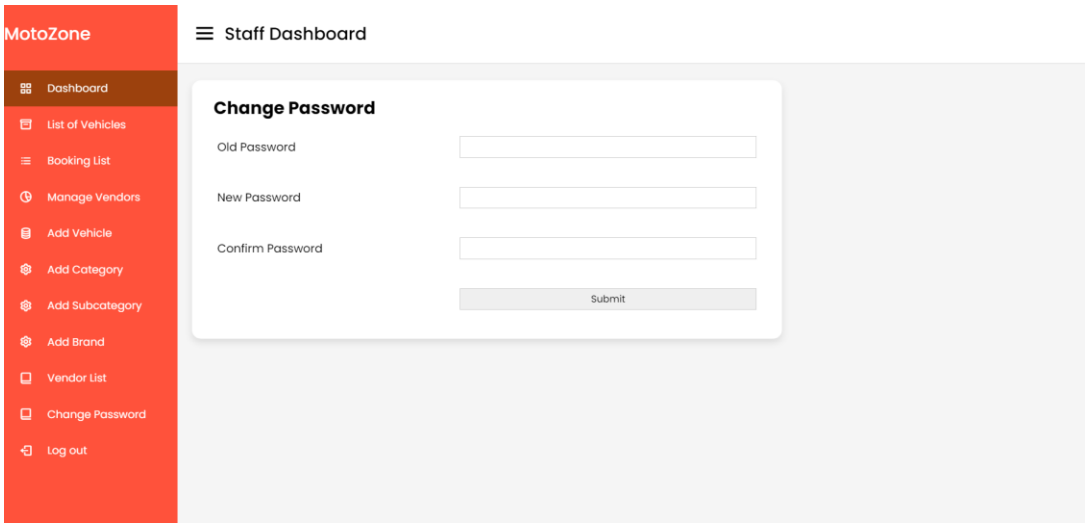
### Staff Add Brand Form

**Description:** This form is used to add a new brand of vehicle to the website.



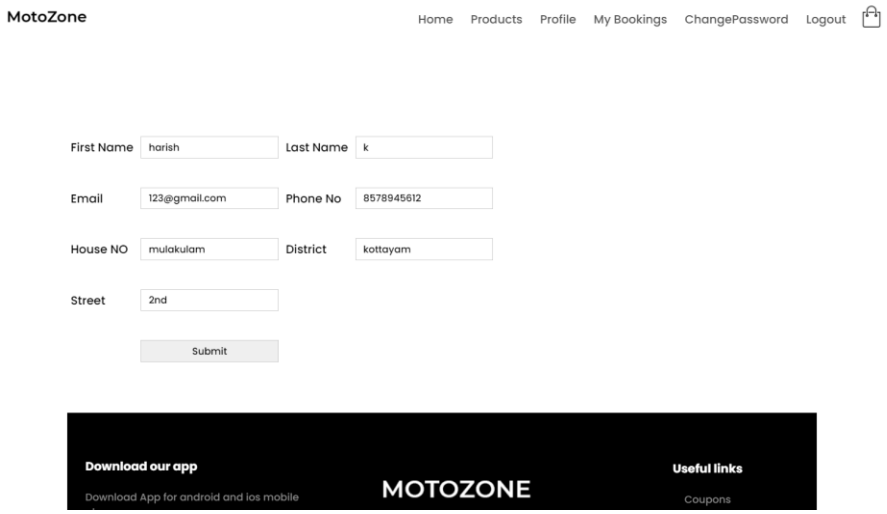

### Staff Change Password Form

**Description:** This form is used to change the password of the current staff.



**Customer Profile Form**

**Description:** This form is used to let the customer update his details if needed.



## 4.2 OUTPUT DESIGN

One of the important features of an information system for users is the output it produces. Output is the information delivered to users through the information system. Without quality output, the entire system appears to be unnecessary that users will avoid using it. Uses generally merit the system solely by its output. In book to create the most useful output possible. One works closely with the user through an interactive process, until the result is considered to be satisfactory.

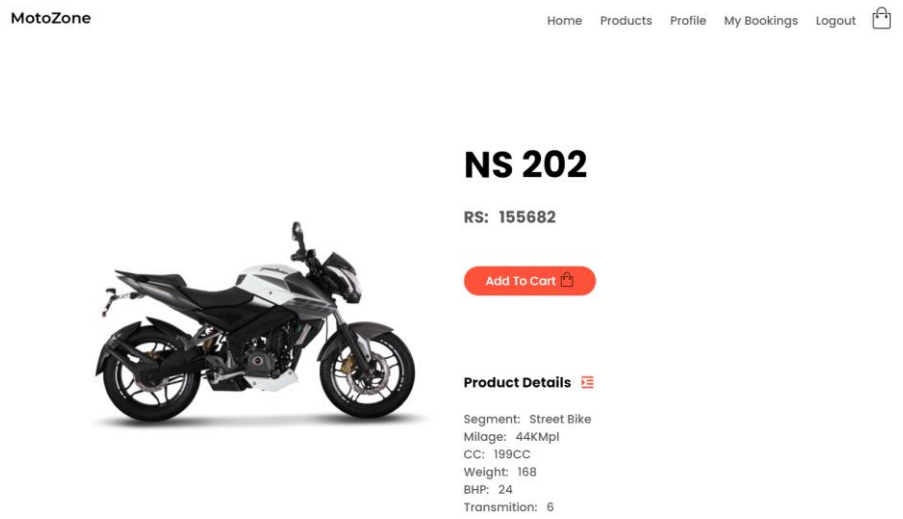
Output design has been an ongoing activity almost from the beginning of the project. In the study phase, outputs were identified and described general in the project directive. A tentative output medium was then selected and sketches made for each output. In the feasibility analysis, a “best” new system was selected; its description identified the input and output media. In the design phase the system has included an evaluation and selection of specific equipment for the system.

Output design generally deals with the results generated by the system i.e., reports. These reports can be generated from stored or calculated values. Reports are displayed either as screen window preview or printed form. Most end users will not actually operate the information system or enter data through workstation, but they will use the output from the system.

Outputs from computer systems are required primarily to communicate the results of processing to the user. They are also used to provide a permanent copy of these results for later consultation.

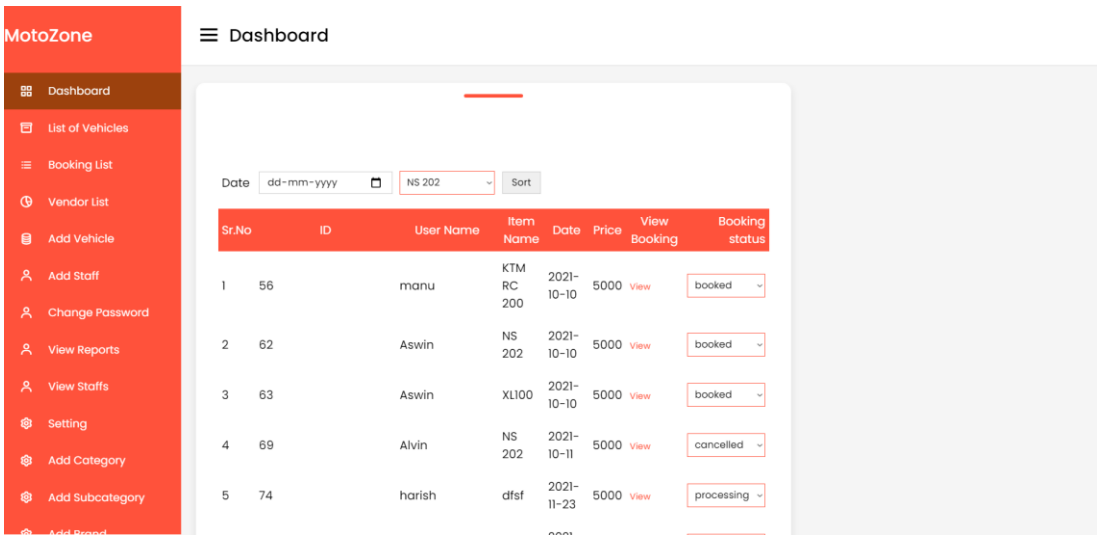
**Customer View Product Details Page**

**Description:** This page shows the details of vehicle selected



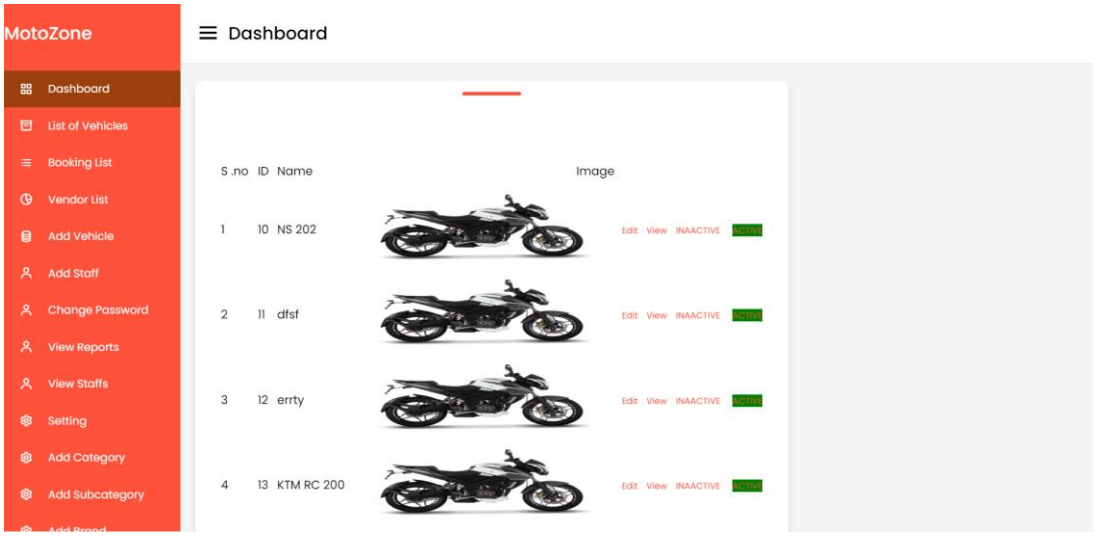

**Admin view Booking Page**

**Description:** This page shows the details of the bookings made in the system.



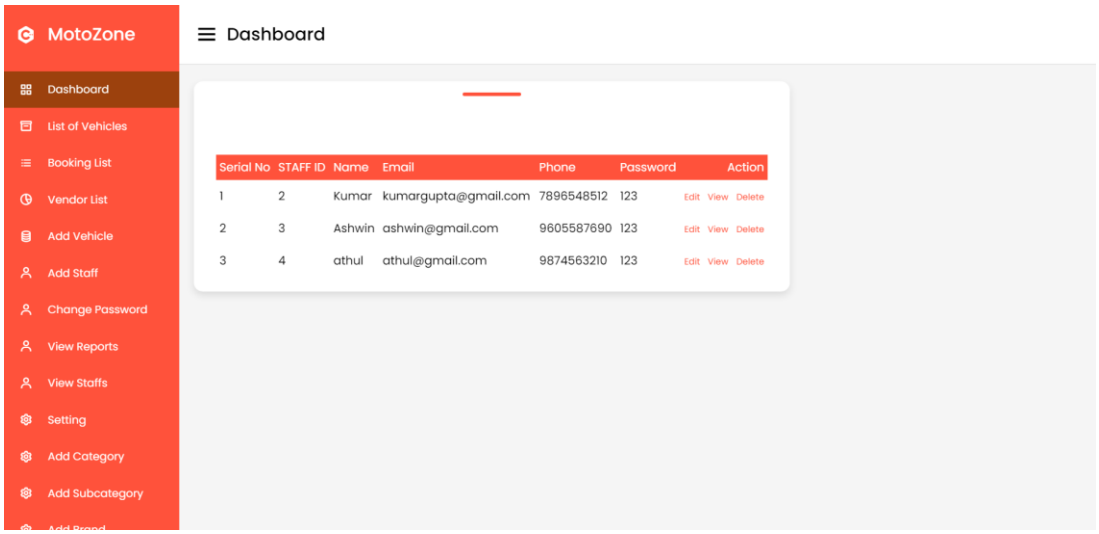
**Admin View list of vehicles page**

**Description:** This page shows the list of all vehicles added to the system.



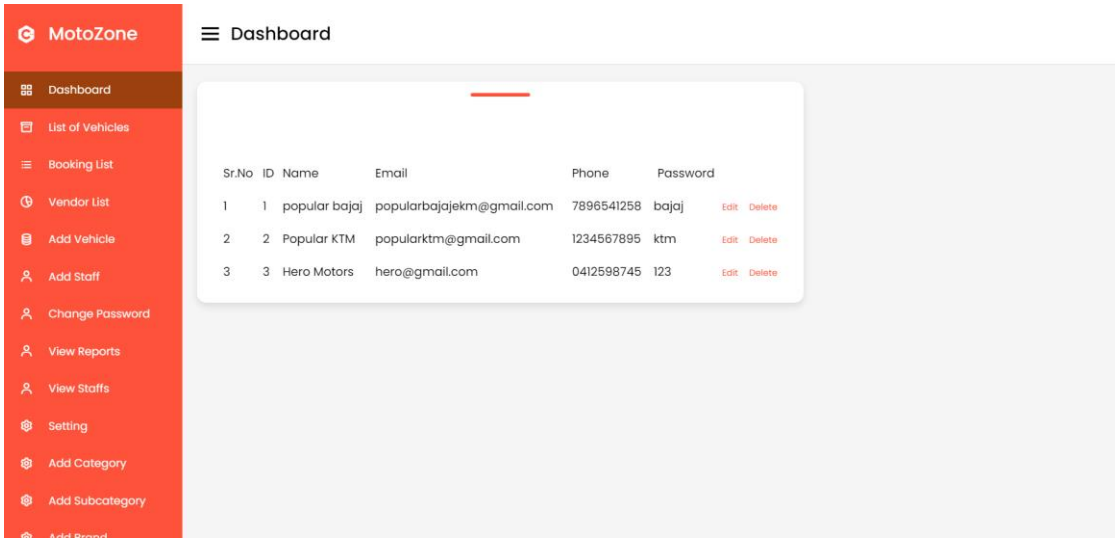

**Admin View Staff Page**

**Description:** This page shows the list of Staff working in the system.



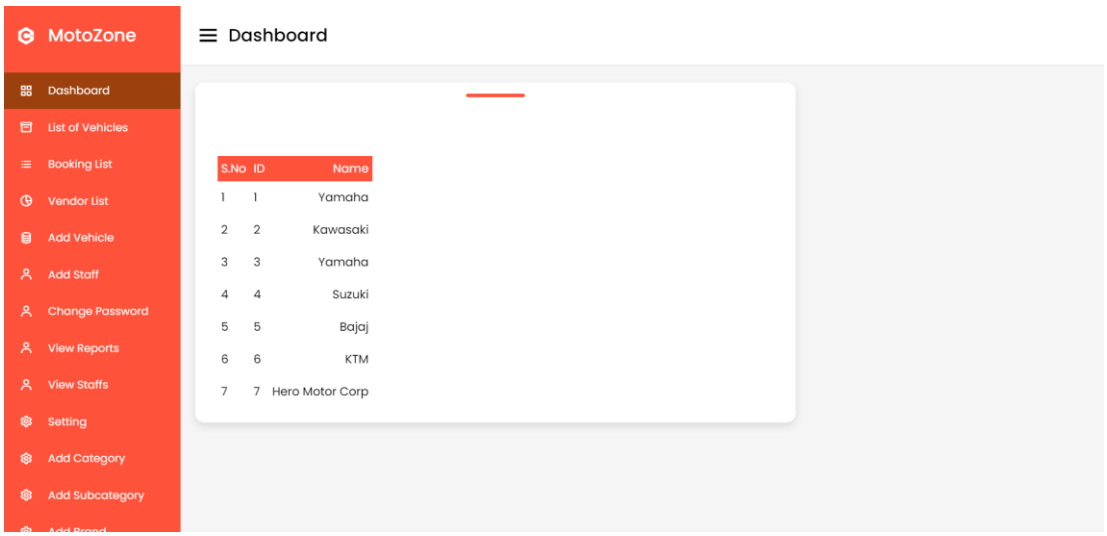
### Admin View Vendors Page

**Description:** This page shows the list of vendors in the system.



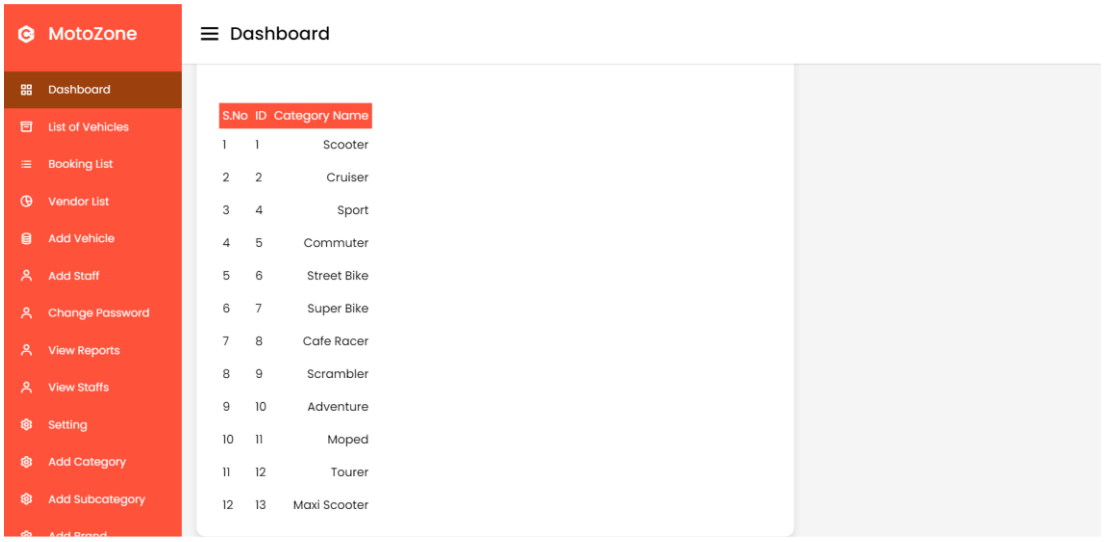

### Admin Brands list Page

**Description:** This page shows the list of brands in the system.



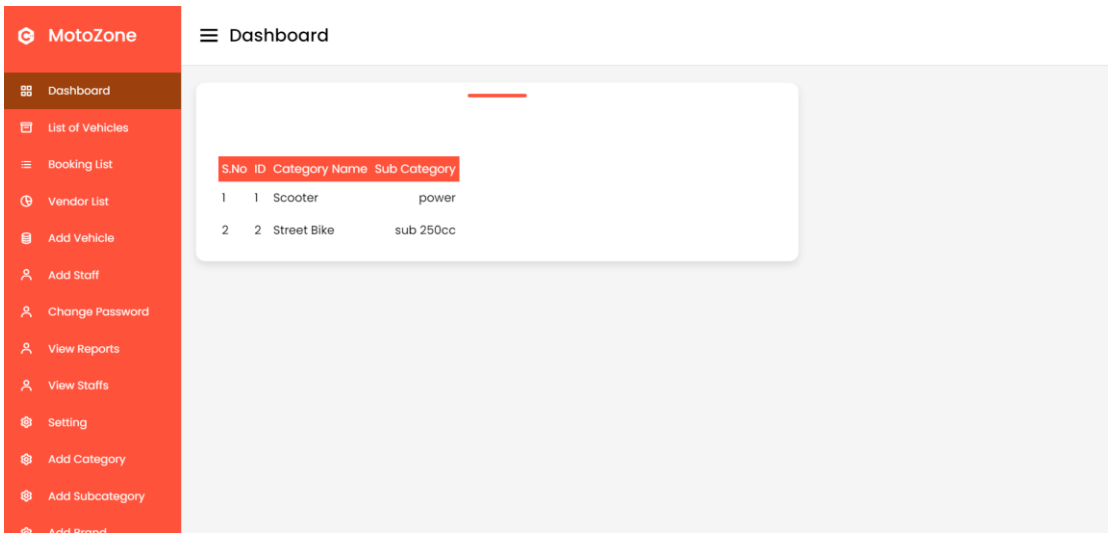
### Admin Categories list Page

**Description:** This page shows the list of categories of products.



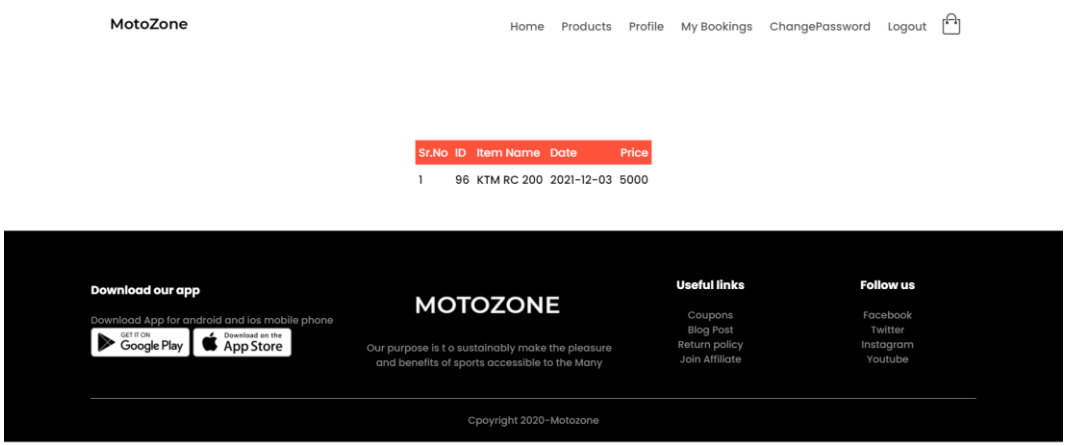

### Admin Sub-Categories List Page

**Description:** This page shows the list of Sub-categories of products.



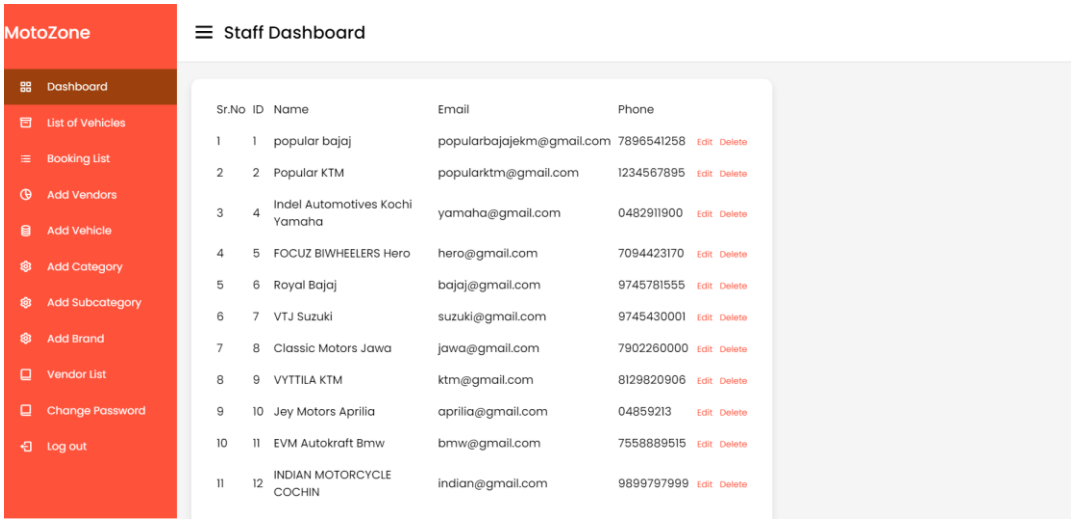
## Customer view Booking Page

**Description:** This page shows the list of bookings made by the customer.



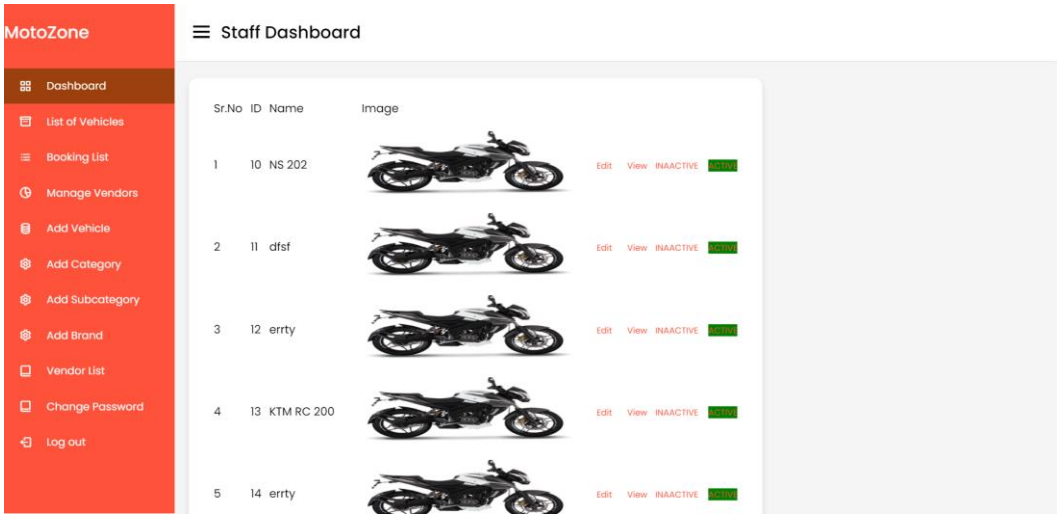

## Staff View List of Vendor Page

**Description:** This page shows the list of vendors in the website.



**Staff View List of vehicles Page**

**Description:** This page shows the list vehicles in the website.



## 4.3. DATABASE DESIGN

### 4.3.1 NORMALIZATION

Designing a database is a complex task and the normalization theory is a useful aid in this design process. The process of normalization is concerned with transformation of conceptual schema into computer representation form.

A bad database design may lead to certain undesirable situation such us,

- Repetition of information
- Inability to represent certain information
- Loss of information

To minimize these anomalies, normalization mat be used. If the database is in a normalization form, the data can be restructured and can maintain it easily. This is important that the database using that we are using may free from data redundancy and inconsistency. For this need we maintain the tables in a normalized manner.

**First Normal Form**

A relation is in first normal form (INF), if and only if all its attributes are based on single domain. The objective of normalizing a table is in to remove its repeating groups and ensure that all entries of the resulting table have at most single value.

**Second Normal Form**

A table is said to be in Second Normal Form (2NF), when it is in 1NF and every attribute in the record is functionally dependent upon the whole key, and not just a part of the key.

**Third Normal Form**

A table is in third Normal Form (3NF), when it is in 2NF and every non-key attribute is functionally dependent on just the primary key.

**Table name**: tbl_login

**Description**: Stores the login details of users that exist in the system.

| FIELD | DATATYPE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| username | Varchar(30) | Primary key | Email |
| usertype | Varchar(20) | Not null | Type of user |
| password | Varchar(30) | Not null | Password |

**Table name**: tbl_staff

**Description**: Stores the details of staff working in the store

| FIELD | DATATYPE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| staff_ id | Varchar(10) | Primary key | Staff ID |
| s_ fname | Varchar(30) | Not null | Staff First Name |
| s_ lname | Varchar(30) | Not null | Staff Last Name |
| s_ email | Varchar(30) | Unique | Staff Login Email |
| s_ password | Varchar(30) | Not null | Staff Password |
| s_ hno | Varchar(20) | Not null | Staff House Number |
| s_ district | Varchar(30) | Not null | Staff District |
| s_ street | Varchar(25) | Not null | Staff Street |
| s_ phoneno | Numeric(10) | Unique | Staff Phone Number |
| s_ salary | Numeric(15,3) | Not null | Staff Salary |

**Table name**: tbl_customer

**Description**: Stores the details of customers

| FIELD | DATATYPE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| cust_ id | Varchar(10) | Primary key | Customer ID |
| c_ fname | Varchar(30) | Not null | Customer First Name |
| c_ lname | Varchar(30) | Not null | Customer Last Name |
| c_ email | Varchar(30) | Unique | Customer Login Email |
| c_ password | Varchar(30) | Not null | Customer Password |
| c_ hno | Varchar(40) | Not null | Customer House number |
| c_ district | Varchar(30) | Not null | Customer District |
| c_ street | Varchar(25) | Not null | Customer Street |
| c_ gender | Varchar(2) | Not null | Customer Gender |
| c_ phoneno | Numeric(10) | Unique | Customer Phone Number |

**Table name**: tbl_vendor

**Description**: Table which stores vendor details

| FIELD | DATATYPE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| vendor_ id | Varchar(10) | Primary key | Vendor ID |
| staff_ id | Varchar(10) | Foreign key | Staff ID |
| v_ name | Varchar(30) | Not null | Vendor Name |
| v_ email | Varchar(30) | Unique | Vendor Email |
| v_ district | Varchar(30) | Not null | Vendor District |
| v_ street | Varchar(25) | Not null | Vendor Street |
| v_ phoneno | Numeric(10) | Unique | Vendor Phone Number |

**Table name**: tbl_ category

**Description**: Table which stores category details

| FIELD | DATATYPE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| cat_ id | Varchar(10) | Primary key | Category ID |
| cat_ name | Varchar(30) | Not null | Category Name |

**Table name**: tbl_ subcategory

**Description**: Table which stores subcategory details

| FIELD | DATATYPE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| scat_ id | Varchar(10) | Primary key | Subcategory ID |
| cat_ id | Varchar(10) | Foreign key | Category ID |
| scat_ name | Varchar(30) | Not null | Subcategory Name |

**Table name**: tbl_ brand

**Description**: Table which stores brand details

| FIELD | DATATYPE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| brand_ id | Varchar(10) | Primary key | Brand ID |
| brand_ name | Varchar(30) | Not null | Brand Name |

**Table name**: tbl_ vehicle

**Description**: Table which stores item details

| FIELD | DATATYPE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| vehicle_ id | Varchar(10) | Primary key | vehicle ID |
| scat_ id | Varchar(10) | Foreign key | Subcategory ID |
| brand_ id | Varchar(10) | Foreign key | Brand ID |
| vehiclename | Varchar(25) | Not null | Vehicle Name |
| vehicle_ des | Text | Not null | vehicle Description |
| Ex-showroom price | Decimal(10,2) | Not null | Vehicle Ex-Showroom price |
| Segment | Varchar(10) | Not null | Vehicle Segment |
| Mileage | Decimal(10,3) | Not null | Vehicle Mileage |
| Cubic_capacity | Decimal(10,2) | Not null | Vehicle cubic capacity |
| Colour | Varchar(50) | Not null | Colour of the vehicle |
| Kerb_weight | int | Not null | Vehicle weight |
| bhp | int | Not null | Vehicle bhp |
| Transmission | int | Not null | Vehicle Gearing |
| Tank_capacity | Decimal(3,1) | Not null | Tank capacity |
| Images | varbinary | Not null | Vehicle image |

**Table name**: tbl_ cartmaster

**Description**: Table which stores cartmaster details

| FIELD | DATATYPE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| cmaster_ id | Varchar(10) | Primary key | Cart Master ID |
| cust_ id | Varchar(10) | Foreign key | Customer ID |
| tot_ amount | Decimal(10,2) | Not null | Total Amount of all the Items in Cartchild |
| cart_ date | Date | Not null | Date of Adding Item |
| cart_ status | Varchar(30) | Not null | Status of cart |

**Table name**: tbl_ cartchild

**Description**: Table which stores cartchild details

| FIELD | DATATYPE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| cchild_ id | Varchar(10) | Primary key | Cart Child ID |
| cmaster_ id | Varchar(10) | Foreign key | Cart Master ID |
| item_ id | Varchar(10) | Foreign key | Item ID |
| qty | Int | Not null | Number of items |
| tot_ price | Decimal(10,2) | Not null | Item Selling price * Qty |

**Table name**: tbl_booking

**Description**: Table which stores booking details

| FIELD | DATATYPE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| booking_ id | Varchar(10) | Primary key | Booking ID |
| cmaster_ id | Varchar(10) | Foreign key | Cart Master ID |
| bookingdate | Date | Not null | Date of Booking |
| booking_status | enum | Not null | Status of booking |

**Table name**: tbl_carddetails

**Description**: Table which stores card details

| FIELD | DATATYPE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| card_id | Varchar(10) | Primary key | Card ID |
| custid | Varchar(10) | Foreign key | Customer ID |
| cardtype | Varchar(20) | Not null | Card Type |
| cardno | Numeric(20) | Unique | Card Number |
| bank | Varchar(40) | Not null | Bank Name |
| expdate | Date | Not null | Expiry Date of Card |

**Table name**: tbl_ payment

**Description**: Table which stores payment details

| FIELD | DATATYPE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| payment_ id | Varchar(10) | Primary key | Payment ID |
| booking_ id | Varchar(10) | Foreign key | Booking ID |
| card_ id | Varchar(10) | Foreign key | Card ID |
| payment_ type | Varchar(20) | Not null | Payment Type |
| payment_ date | Date | Not null | Date of Payment |
| payment_ status | Varchar(20) | Not null | Payment Status (Paid/Not paid) |

## 5.1 Introduction

Software testing can be looked upon among the many process in organization that provides the last opportunity to correct any plane in the development system. System testing includes selecting tests and test data that have more problem of finding errors. System testing is vital for the success of any software system. The system makes a logical assumption that all part of the system work efficiently and goal is achieved. The system is tested for online response, ability to store and stress recovery from failure and usability. System testing requires a test plan that consists of several key activities and steps for programming and user acceptance testing. Another benefit of system testing is its utility as a user oriented system before implementation.

**LEVELS OF TESTING**

Some of the methods of the system testing are given below.

**Unit testing**

In this test each module is tested individually before integration it to the final system. Unit test focuses verification in the smallest unit of software design in each module. This is also known as module testing. In this test each module is tested whether it is producing the desired output and if any error occurs it can be corrected easily.

**Integration testing**

It is the systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with interfacing. Thus the relationship between difference modules is checked in this testing for overall performance of testing. Thus in integration testing step, all errors uncovered are corrected for next testing steps. The objective of the test is to take althea modules such as administrator, user and modules are integrated in this testing step and then the entire program is tested.

**Validation testing**

It in where requirements established as a part of software requirements analysis is validated against the software that has been constructed. This test provides the final assurance that the software meets all functional, behavioral and performance requirements. The errors, which are uncovered during integration testing, are connected during this phase.

**Output Testing**

No system could be useful if it does not produce the required output in the specific format. Output testing is performed to ensure the correctness of the output and its format. The output generated or displayed by the system is tested asking the user about the format required by them.

**User Acceptance Testing**

The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system user at the time of developing. The testing of the software began along with the coding. The unit testing was done for each module in the software. For various inputs such that each line of code is executed at least once.

## 5.2 TEST CASES

A test plan document the strategy that will be used to verify and ensure that a room or system meets its design specification and other requirements. A test plan is usually prepared by or with significant input from test Engineers. Depending on the room and the responsibility of the organization to which the test plan applies.

**Unit Testing**

| **Form** | **Procedure** | **Expected Result** | **Actual Result** | **Status** |
|---|---|---|---|---|
| Entry Form | Choose whether to Login,About us or Developers | | | |
| Login Form | Enter valid username and password | Should validate user and provide link to user accounts | Got entry to accounts | Pass |
| Staff Form | Enter all mandatory fields | Should validate all entered fields and flash a message indicating successful registration | Message indicating successful registration is shown | Pass |
| Customer Form | Enter all mandatory fields | Should validate all entered fields and flash a message indicating | Message indicating successful registration is shown | Pass |

| | | | | |
|---|---|---|---|---|
| | | successful registration | | |
| Category Form | Enter all mandatory fields | Should validate all entered fields and flash a message indicating successful registration | Message indicating successful registration is shown | Pass |
| Subcategory Form | Enter all mandatory fields | Should validate all entered fields and flash a message indicating successful registration | Message indicating successful registration is shown | Pass |
| Vehicle Form | Enter all mandatory fields | Should validate all entered fields and flash a message indicating successful registration | Message indicating successful registration is shown | Pass |
| Payment Form | Enter all mandatory fields | Should validate all entered fields and flash a message indicating successful registration | Message indicating successful registration is shown | Pass |

**Integration Testing**

| Form | Expected Result | Actual Result | Status |
|---|---|---|---|
| Login and user account forms | Get entry to appropriate user page | Appropriate user page is displayed | Pass |
| Staff Form | Must add staff details successfully | Insertion is successful | Pass |
| Customer Form | Must add customer details successfully | Insertion is successful | Pass |
| Vendor Form | Must add Vendor details successfully | Insertion is successful | Pass |
| Category Form | Must add category details successfully | Insertion is successful | Pass |
| Payment Form | Must update the specified entry in the database | Specified entry updated | Pass |

**Validation Testing**

| **Form** | **Expected Result** | **Actual Result** | **Status** |
|---|---|---|---|
| Create user | Check all mandatory fields and validate all entered data fields | If any error found display message and the same screen is displayed else record saved and confirmed | Pass |
| Edit User | Edit the row corresponding to the value entered | If the value entered is invalid error message is thrown otherwise message indicating successful deletion is flashed | Pass |

## 6.1 Introduction

Implementation is that state in the project plan where the theoretical design is put into real test. All the theoretical and practical works are now implemented as a working system. This is the most crucial stage in the life cycle of a project; the project may be accepted or rejected depending on how it gathers confidence among the users. If the user has achieved satisfaction with the new project, then the project can be termed as successful and then onwards its maintenance and other subsequent works can be commenced. The system goes for implementation only after passing through some rigorous testing, especially when it comes to operating system and other system software, the testing and implementation phase assumes greater significance.

The implementation stage involves following tasks

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the change cover.
- Evaluation of the changeover method.

## 6.2 INSTALLATION PROCEDURE

Installation of software refers to the final installation of the package in the real environment, to the satisfaction of the intended users and the successful operation of the system. In many organizations, those who commission the software development project will not be the one to operate them. In the initial stage, the person who is not sure that the software will make the jobs easier will doubt about the software. But we have to ensure that the resistance does not build one makes sure that

- The active user must be aware of the benefits of using the system
- Their confidence in the software is built up
- Proper guidance is imparted to the user so that he is comfortable in using the application

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage, the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned and controlled, it can cause confusion.

Implementation includes all those activities that take place to convert from the old system to the new one. Proper implementation is essential to provide a reliable system to meet the organizational requirements. Successful implementation may guarantee improvement in the organization using the new system, but improper installation will prevent it. The process of putting the developed system into actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specification of the system.

## 6.3 IMPLEMENTATION PLAN

Implementation is the most crucial stage in achieving a successful system and for us it is the processing of bringing "Online Two-Wheeler Booking Platform" into operational use and training it over to the user. Implementation includes all those activities that take place to convert from the old system to new one.

The basic requirements for implementing the proposed system are already mentioned above. This software provides total security for the operations. That is it prevents any unauthorized access. After successful login the user can go to the form according to the situation.

After completion of the Online Two-Wheeler Booking Platform System design and coding, the analyst, the user and the management evaluates the system to ensure that it fulfill all its goals. Thus the implementation of the project where the critical design is turned into a working system. System implementation plan is concerned with writing program, creating databases, testing programs and operational plans.

## 7.1 FUTURE ENHANCEMENT

The system has been developed with flexibility in mind. The requirement of the company is bound to change as and when new operations are included. Keeping in view advancements that are being made in technology it is necessary that the system be able to cope up with the changes that are bound to happen.

So in today's world of mobile technology the software “Online Two-Wheeler Booking Platform” if integrated with the mobile will be an added advantage. The mobile users will get instant alerts from this site. The software if we create a mobile app or an alert system for more interaction with the user and also widening the reach of the system to its users.

The system entitled “Online Two-Wheeler Booking Platform” provides maximum user interaction and flexibility. The system users stored procedures on the database. This also can be enhanced in the future.

## BIBLIOGRAPHY

Website References

- www.stackoverflow.com - https://stackoverflow.com/en-us/library/aa228849(v=vs.60).aspx
- www.coderewind.com -http://www.coderewind.com/article/jqueryplugins/
- www.studentsprojectsguide.com - http://studentsprojectsguide.in/15698264-2?show=86
- www.dreamincode.net - http://www.dreamincode.net/forums/topic/132258-editing-msflexgrid-by-users/
- www.w3schools.com - http://www.w3schools.com/jqueryvalidations.html

## APPENDIX A

```
<?php
include "connect.php";
session_start();
$_SESSION["uname"]="";
$_SESSION["utype"]=null;
?>
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>MotoZone | Find The Right Bike</title>
    <link rel="stylesheet" href="style.css" />
    <link rel="preconnect" href="https://fonts.googleapis.com" />
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
    <link

href="https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,300;0,500;0,6
00;0,700;1,400&display=swap"
      rel="stylesheet"
    />
    <link
      rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-
free@5.15.3/css/fontawesome.min.css"
    />
  </head>
  <body>
    <div class="header">
```

```
        <div class="container">
      <div class="navbar">
       <div class="logo">
        <a href="#"><img src="images/logo.png" width="125px" /></a>
       </div>
       <nav>
        <ul id="MenuItems">
        <li><a href="index.php">Home</a></li>
       <li><a href="account.php">Login</a></li>
          <li><a href="registration.php">Sign Up</a></li>
        </ul>
       </nav>
       <!-- <a href="cart.html"
        ><img src="images/cart.png" width="30px" height="30px"
       /></a> -->
       <!-- <img src="images/menu.png" class="menu-icon"
onclick="menutoggle()" /> -->
      </div>
      <div class="row">
       <div class="col-2">
        <h1>
         Find the Perfect Ride <br />
         For You!
        </h1>
        <p>
         Success isn't always about greatness. It's about consistency.
         <br />
         Unveil Your Dreams And Passion Of Riding With Us.
         <a href="account.php" class="btn">Explore Now &#8594;</a>
        </p>
       </div>
```

```
          <div class="col-2">
          <img src="images/image1.png" alt="" />
        </div>
      </div>
    </div>
  </div>

  <!-------- featured categories ------------------>
  <div class="categories">
    <div class="small-container">
     <table class="table">

    <?php
    $qry="Select * from tbl_category";
    $res=mysqli_query($con,$qry);
    $ccnt=0;
    while($data=mysqli_fetch_assoc($res))
    {
      if($ccnt==0)
      echo '<tr>';
        ?>
      <td>
    <a href="products.php?cid=<?php echo $data["cat_id"]; ?>"> <?php echo
$data["cat_name"]; ?></a>
    </td>

    <?php



      $ccnt++;
      if($ccnt==6)
```

```
        {
     echo '</tr>';
     $ccnt=0;
     }
    }


    ?>


    </table>
    </div>
   </div>
   <!-------- featured vehicles ------------------>
   <div class="small-container">
    <h2 class="title">Featured Vehicles</h2>
    <div class="row">




      <div class="col-9">
       <p>We are happy to help you achieve your dreams of riding with us. With our help you can now book your favourite dream bike in the comfort of your home.

We provide 100% customer satisfaction and a hassle free experience for your purchase.

We provide Brand new motorcycles to our customers.

As an aspiring brand we value your trust and considers the customer relationship our top priority.

All the motorcycles booked through us will undergo a set of inspections from our experienced staff to ensure that you get a perfectly functioning ride without any flaws.
    </p>
      </div>
    </div>
```

```
    </div>
  <!-------- offer -------->
  <div class="offer">
   <div class="small-container">
    <div class="row">
     <div class="col-2">
      <img src="images/r1.png" class="offer-img" />
     </div>
     <div class="col-2">
      <p>Exclusively available on MotoZone</p>
      <h1>Yamaha R1</h1>
      <small>YZF-R1 is a lot like its track brother YZR-M1. With pure MotoGP
blood, a crossplane engine, short wheelbase chassis and high-tech electronics, the
YZF-R1 is ready to take your riding experience to a whole new level.</small
      >
      <a href="" class="btn">Buy Now &#8594;</a>
     </div>
    </div>
   </div>
  </div>
  <!-------- Testimonial -------->
  <div class="testimonial">
   <div class="small-container">
    <div class="row">
     <div class="col-3">
      <i class="fa fa-quote-left"></i>
      <p>


      </p>
      <div class="rating">
       <i class="fa fa-star"></i>
       <i class="fa fa-star"></i>
```

```
  <i class="fa fa-star"></i>
    <i class="fa fa-star"></i>
    <i class="fa fa-star-half-o"></i>
  </div>
  <img src="images/user-1.png" />
  <h3>Sean Parker</h3>
</div>
<div class="col-3">
  <i class="fa fa-quote-left"></i>
  <p>

  </p>
  <div class="rating">
    <i class="fa fa-star"></i>
    <i class="fa fa-star"></i>
    <i class="fa fa-star"></i>
    <i class="fa fa-star"></i>
    <i class="fa fa-star-half-o"></i>
  </div>
  <img src="images/user-2.png" />
  <h3>Mike Smith</h3>
</div>
<div class="col-3">
  <i class="fa fa-quote-left"></i>
  <p>

  </p>
  <div class="rating">
    <i class="fa fa-star"></i>
    <i class="fa fa-star"></i>
    <i class="fa fa-star"></i>
```

```
        <i class="fa fa-star"></i>
          <i class="fa fa-star-half-o"></i>
        </div>
        <img src="images/user-3.png" />
        <h3>Mabel Joe</h3>
      </div>
    </div>
  </div>
</div>
<!-------- brands ---------->
<div class="brands">
  <div class="small-container">
    <div class="row">
      <div class="col-5">
        <img src="images/logo-BMW.png" />
      </div>
      <div class="col-5">
        <img src="images/logo-kawasaki.png" />
      </div>
      <div class="col-5">
        <img src="images/logo-triumph.png" />
      </div>
      <div class="col-5">
        <img src="images/logo-yamaha.png" />
      </div>
      <div class="col-5">
        <img src="images/logo-ducati.png" />
      </div>
    </div>
  </div>
</div>
```

```
<!------ Footer ----->

<div class="footer">
  <div class="container">
    <div class="row">
      <div class="footer-col-1">
        <h3>Download our app</h3>
        <p>Download App for android and ios mobile phone</p>
        <div class="app-logo">
          <img src="images/play-store.png" />
          <img src="images/app-store.png" />
        </div>
      </div>
      <div class="footer-col-2">
        <img src="images/logo-white.png" />
        <p>
          Our purpose is t o sustainably make the pleasure and benefits of
          sports accessible to the Many
        </p>
      </div>
      <div class="footer-col-3">
        <h3>Useful links</h3>
        <ul>
          <li>Coupons</li>
          <li>Blog Post</li>
          <li>Return policy</li>
          <li>Join Affiliate</li>
        </ul>
      </div>
      <div class="footer-col-4">
        <h3>Follow us</h3>
```

```
          <ul>
            <li>Facebook</li>
            <li>Twitter</li>
            <li>Instagram</li>
            <li>Youtube</li>
          </ul>
        </div>
      </div>
      <hr />
      <p class="copyright">Cpoyright 2020-Motozone</p>
    </div>
  </div>
  <!--js for toggle menu--->
  <script>
    var MenuItems = document.getElementById("MenuItems");

    MenuItems.style.maxHeight = "0px";

    function menutoggle() {
      if (MenuItems.style.maxHeight == "0px") {
        MenuItems.style.maxHeight = "200px";
      } else {
        MenuItems.style.maxHeight = "0px";
      }
    }
  </script>
 </body>
</html>
```

## APPENDIX B

**Acronyms**

SQL - Structured Query Language

DFD - Data Flow Diagram

ERD - Entity Relationship Diagram

IDE - Integrated Development Environment

OS - Operating System