American International University-Bangladesh



**Documentation Book of** **Bike Ecommerce Store**

Submitted By:

|  |  |
| --- | --- |
| **Name** | **ID** |
| **Md. Reaz Uddin Chowdhury** | **21-92155-3** |
| **Ummey Soaiba Yeasmin** | **21-92158-3** |
| **Akibur Rahman Akib** | **22-92573-2** |
| **Sheuly Akter** | **22-92609-2** |
| **Md. Rifat-Uz-Zaman** | **22-92610-2** |

**Course:** Software Documentation and Tools

**Course Teacher:** Mohammad Samawat Ullah

**Section:** [MScCS] A

**Semester:** Summer 2021-2022

**Date of submitting:** 23, July, 2022

Contents

[**OVERVIEW** 5](#_Toc109815871)

[**Software Requirements Specification** 6](#_Toc109815872)

[**Introduction** 6](#_Toc109815873)

[**Purpose** 6](#_Toc109815874)

[**Scope** 6](#_Toc109815875)

[**Definitions, Acronyms, and Abbreviations** 6](#_Toc109815876)

[**References** 7](#_Toc109815877)

[**The Overall Description** 7](#_Toc109815878)

[**Product Perspective** 7](#_Toc109815879)

[**System Interfaces** 7](#_Toc109815880)

[**Interfaces** 7](#_Toc109815881)

[**Software Interfaces** 7](#_Toc109815882)

[**Communications Interfaces** 7](#_Toc109815884)

[**Memory Constraints** 8](#_Toc109815885)

[**Operations** 8](#_Toc109815886)

[**Site Adaptation Requirements** 8](#_Toc109815887)

[**Product Functions** 8](#_Toc109815888)

[**User Characteristics** 8](#_Toc109815889)

[**Constraints** 9](#_Toc109815890)

[**Assumptions and Dependencies** 9](#_Toc109815891)

[**Specific Requirements** 9](#_Toc109815896)

[**External Interfaces** 9](#_Toc109815897)

[**Functions** 10](#_Toc109815898)

[The functionality of an admin is as follows: 10](#_Toc109815899)

[**Performance Requirements** 10](#_Toc109815900)

[**Logical Database Requirements** 11](#_Toc109815901)

[**Design Constraints** 12](#_Toc109815902)

[**Software System Attributes** 12](#_Toc109815903)

[**Organizing the Specific Requirements** 13](#_Toc109815908)

[**Change Management Process** 16](#_Toc109815909)

[**Document Approvals** 17](#_Toc109815911)

[**Supporting Information** 18](#_Toc109815912)

[**Software Design Document (SDD)** 19](#_Toc109815913)

[**Introduction** 19](#_Toc109815914)

[**Purpose** 19](#_Toc109815915)

[**Scope** 19](#_Toc109815916)

[**Overview** 19](#_Toc109815917)

[**Reference Material** 20](#_Toc109815918)

[**Definitions and Acronyms** 20](#_Toc109815919)

[**System Overview** 20](#_Toc109815920)

[**System Interfaces** 20](#_Toc109815921)

[**Hardware Interfaces** 20](#_Toc109815922)

[**Software Interfaces** 21](#_Toc109815923)

[**Communications Interfaces** 21](#_Toc109815924)

[**Memory Constraints** 21](#_Toc109815925)

[**Operations** 21](#_Toc109815926)

[**System Architecture** 21](#_Toc109815927)

[**Architectural Design** 21](#_Toc109815928)

[**Decomposition Description** 23](#_Toc109815929)

[**Design Rationale** 23](#_Toc109815930)

[**Data Design** 23](#_Toc109815931)

[**Data Description** 23](#_Toc109815932)

[**Human Interface Design** 25](#_Toc109815933)

[**Overview of User Interface** 25](#_Toc109815934)

[**Screen Images** 25](#_Toc109815935)

[**Software Development** 29](#_Toc109815938)

[**Overview** 29](#_Toc109815939)

[**Unit’s functionality** 29](#_Toc109815940)

[**Development methodologies** 29](#_Toc109815941)

[**Source code listing** 29](#_Toc109815942)

[**Controlled libraries/directories/tables** 30](#_Toc109815943)

[**Unit test results and analysis** 30](#_Toc109815944)

[**Roles and Responsibilities** 31](#_Toc109815945)

[**Process** 32](#_Toc109815946)

[**Requirements –** 32](#_Toc109815947)

[**Functionality –** 32](#_Toc109815948)

[**Source Code –** 35](#_Toc109815949)

[**Libraries, Directories, and Tables –** 35](#_Toc109815950)

[**Development Methodologies –** 36](#_Toc109815951)

[**Test Data –** 36](#_Toc109815952)

[**Test Analysis –** 36](#_Toc109815953)

[**Sign-Off –** 39](#_Toc109815954)

[**System Developer** 39](#_Toc109815955)

[**Software Testing** 40](#_Toc109815956)

[**Introduction** 40](#_Toc109815957)

[**Purpose of the test plan document** 40](#_Toc109815958)

[**Functional Testing** 40](#_Toc109815959)

[**Test Risk** 40](#_Toc109815960)

[**Items to be tested/not tested** 40](#_Toc109815961)

[**Test Approach’s** 41](#_Toc109815962)

[**Test Regulatory** 42](#_Toc109815963)

[**Test Pass/Fail Criteria** 42](#_Toc109815964)

[**Test Entry/Exit Criteria** 42](#_Toc109815965)

[**Test Deliverables** 42](#_Toc109815966)

[**Load Testing** 42](#_Toc109815968)

[**Test Risk** 42](#_Toc109815969)

[**Items to be tested/not tested** 42](#_Toc109815970)

[**Test Approach’s** 43](#_Toc109815971)

[**Test Regulatory** 43](#_Toc109815972)

[**Test Pass/Fail Criteria** 43](#_Toc109815973)

[**Test Entry/Exit Criteria** 43](#_Toc109815974)

[**Test Deliverables** 43](#_Toc109815975)

[**Training Needs** 43](#_Toc109815976)

[**Unit Testing** 43](#_Toc109815977)

[**Test Risk** 43](#_Toc109815979)

[**System Testing** 45](#_Toc109815980)

[**User Acceptance Testing** 46](#_Toc109815981)

[**Test Risk** 46](#_Toc109815982)

[**User Manual** 52](#_Toc109815983)

[**Bike E-comers System Dashboard** 52](#_Toc109815984)

[**Bike E-comers Support System** 52](#_Toc109815985)

[**Login System** 53](#_Toc109815986)

[**profile management** 54](#_Toc109815987)

[**update profile** 54](#_Toc109815988)

[**customer feature** 55](#_Toc109815989)

[**Bike details** 56](#_Toc109815990)

[**Order system** 56](#_Toc109815992)

[**Add New product** 57](#_Toc109815993)

[**Logout** 57](#_Toc109815994)

[**Tools Demonstration Screenshots** 58](#_Toc109815995)

# **OVERVIEW**

The bike E-Commerce store is an E-Commerce platform merchants can use to display and sell their bikes online. They will be able to upload information about their bikes and customers will be able to buy the bikes using the platform.

The E-commerce store will have two user classes

1.     Customers

2.     Admin

The customer will be able to browse through the different bikes, login/register and order a bike that he likes

The admin will be able to login, manage the different product information, view and manage the orders and create other admin accounts.

The system will allow users to quickly and easily view the technical specifications of each bike and make a purchase from the comforts of their homes. They don’t need to go to a showroom to view and buy the bikes. This saves the customer time and at the same time, the platform’s owner can serve their customers 24/7.

The Ecommerce platform will have a backend API and a React-based frontend which will consume the API and provide the functionality to the software. A no SQL database called mongo DB will be used as the database.

# **Software Requirements Specification**

**Introduction**

The advent of the internet-based ecommerce has created a revolution in the field of retail commerce. Ecommerce allows different companies and individuals to buy and sell their services or products through the internet. So, this Software Requirements Specifications (SRS) document is written for an e-commerce system named Rangers-bikes. This software will be used for selling bikes through a react based modern ecommerce store. This SRS document outlines the requirements that the system has which will help the design and development team during development.

## **Purpose**

This document is intended for all the stakeholders of the system so that everyone is on the same page and clear about the requirements of the system before development can start. The requirements should reduce ambiguity as much as possible for the project to be successful. The stakeholders include the product owner, the various users, the project manager, the software engineers and designers.

## **Scope**

The E-commerce store will have two user classes

Customers

Admin

The customer will be able to browse through the different bikes, login/register and order a bike that he likes. The admin will be able to login, manage the different product information, view and manage the orders and create other admins accounts.  
The system will allow users to quickly and easily view the technical specifications of each bike and make a purchase in the comforts of their home. They don’t need to go to a show room to view and buy the bikes. This saves the customer time and the owner of the system to serve more customers without hiring more sales reps.

The Ecommerce platform will have an API and a react based frontend which will consume the API and provide functionality to the software.

## **Definitions, Acronyms, and Abbreviations**

1. API- API stands for Application Programming Interface
2. React- A JavaScript library for building user interfaces
3. NodeJS A JavaScript library for building user interfaces
4. Mongo DB – Schema less database

## **References**

1. https://reactjs.org/
2. <https://www.mongodb.com/>
3. <https://sendgrid.com/solutions/email-api/>  
   <https://www.sendinblue.com/api/>
4. <https://www.twilio.com/docs/sms/api>  
   <https://rapidapi.com/blog/sms-apis-send-texts/>

## **The Overall Description**

### **Product Perspective**

The software product is a standalone cloud-based software which will be accessible via web browsers. Dependencies include the SMS API used for OTP during registration and login and email API used to send marketing and other confirmation emails.

### **System Interfaces**

The system must integrate with a third part SMS API to provide OTP and other SMS notification to the users. Some of the options for an SMS API is given below: <https://www.twilio.com/docs/sms/api>  
<https://rapidapi.com/blog/sms-apis-send-texts/>

The system should also integrate with an email API to send invoices and other email notifications to the users. So third part Email APIs and given below:

<https://sendgrid.com/solutions/email-api/>  
<https://www.sendinblue.com/api/>

### **Interfaces**

The software application will be a cloud-based platform where the customers will access a frontend application using a web or mobile browser and the admin will also be able to access the backend dashboard using a browser.

### **Software Interfaces**

### The customer currently does not have any software that needs to be interfaced with. So, the system will be standalone API and a frontend interface. We only have to interact with the third-party email and SMS APIs to provide SMS and email functionality.

### **Communications Interfaces**

Our web application will interface with the backend REST API using the standard TCP/IP over the internet. A basic overview of TCP/IP is linked below:

<https://www.fortinet.com/resources/cyberglossary/tcp-ip>

### **Memory Constraints**

To make sure the application is accessible to as many customers as possible the devices with low end RAM and processor should be able to smoothly run the application using a browser without any performance issues. Based on our focus groups the RAM requirements are 1 GB.

### **Operations**

The typical operations of the bike ecommerce include inventory management and keeping track of the model and number of bikes currently in stock. When a new bike included in stock the admin will add the bike from the backend portal so that the new bike is visible to the users. When a customer orders a bike, the admin will view the order and will change the status of the order accordingly. There will be no payment gateway in the ecommerce store so all the orders will have cash on delivery by default.

### **Site Adaptation Requirements**

Admin should have a decent laptop with at least 4 GB RAM 3 GHz processor to operate the backend portal of the system. The Admin laptops should also have continuous electricity and a stable internet connection.

### **Product Functions**

**Customers**

* Customers can Login
* Customers can Register
* Customers can Make orders
* Customers can View their Orders
* Customers can Cancel their order
* Customers can Give reviews

**Admin**

* Admin Can Login
* Admin Can Register
* Admin Can Manage Products
* Admin Can Add Products
* Admin Can make a user an admin
* Admin Can manage Orders
* Admin Can view profile

### **User Characteristics**

The primary users are customers who will access the system online. Users have minimal technical knowledge about bikes and hence need intuitive navigation aids and simple page layouts. Administrator is responsible for maintaining the system and will be involved in software fixes, deployment and regular maintenance.

### **Constraints**

The overall system must be computationally efficient so that server costs can be kept to a minimum.

### **Assumptions and Dependencies**

We have assumed that it will be running on a properly working web server and database system with an Internet connection that allows this system to perform all communications with clients.

* The manager account’s username and password may be hard coded.
* The manager cannot be a customer.
* Any user cannot edit their account information.

## **Specific Requirements**

In the beginning, the main specific requirement of this document is to provide complete detail of this software product, its expansion, parameters, and destination. So, this document describes the whole project's destination listeners and its user interface, hardware, and software requirements also. For this reason, it seems very important to finish the project by covering all of its requirements successfully. Therefore, the pros and cons of this project or software can introduce before layout precious time and money among it by using probability recitation. Now, a feasibility study is known as an analysis that takes all of the project's incidental factors into account including economic, technical, legal, and also scheduling judgments. Hence by using liability or feasibility lessons, the total principal, labor, and time will be required for the whole project can be aware, which is essential to understand as capital, exertion, and time are scarce resources. Consequently, with this information, it can be introduced if enough labor is available or more wants to be hired, if there have sufficient principles or if more needs to be added, and whether all of the requirements for the completion of the software are feasible too. Therefore, with this entire information, it seems to be easier to fix if these software requirements are doable, and also favor understanding which project or software to develop and which ones to abandon before investing resources in some things that ultimately show no commitment to generate revenue or cannot be fully successful.

### **External Interfaces**

In this software, the system users’ relationships are responsible for any formal interaction or intimate between the developer team and the user or customer contact or concretion. Consequently, all of the necessary interactions or communications are done through any people who are included in the team, besides that all kinds of discussions with users or customers document fairly for keeping a record. Additionally, every user can send a request for required services or configuration rank, or clause changes that are written and also approved by the project Configuration Control Board (CCB) including the project manager, which is consisted of entire group members.

### **Functions**

### The functionality of an admin is as follows:

* The Admin shall be able to login using by filling up the email and password field
* The Admin shall be able to able to create a new admin by inputting his email
* The Admin shall be able to view orders made by a customer
* The Admin shall be able to change the status of an order using a dropdown
* The Admin shall be able to add a new product by inputting the product details in a form
* The Admin shall be able to delete products by clicking the delete button on a product

The function of Customer is such as:

* The user shall be able to order a bike by filling out name phone and address input fields
* The User shall be able to change profile details by changing the text field and clicking on update
* The user shall be able to login using google login
* The user shall be able to register using email and password
* The user shall be able to view order status in his profile

### **Performance Requirements**

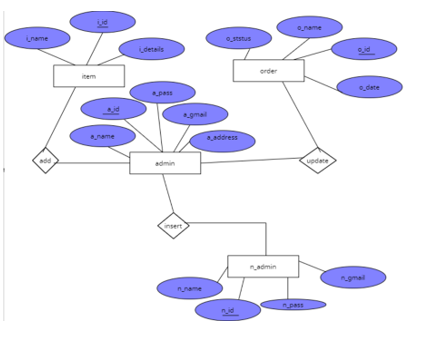
Performance requirements are playing a vital role in any software, and its Software Requirements Specifications (SRS) document because it explains how the software works with its best outcomes. Bike Ecommerce Store software is a system that provides an automatic system or process that is used by a customer for ordering bikes at favorable prices. Here, the admin can track all details, changes details if he needs, and also modify prices or details, and so on. Every option is fully customizable admin or customer can choose whatever they want and even can modify also.

* Customer can find all accurate details of his or her desire product.
* No hassle for order.
* System notify to the admin as well as customer also.
* Wireless order system.
* Eco friendly.
* Admin can manage everything smoothly with proper instructions.
* Automatic order and prices are calculated.
* Customer shall not need to visit showroom.
* Shall not need to go bank for transaction.

### **Logical Database Requirements**

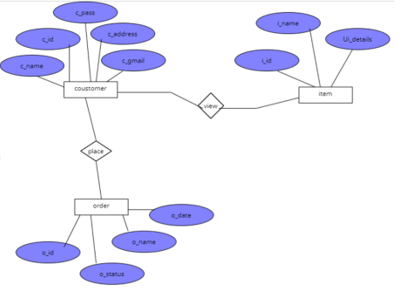
ER diagram of the database requirements is given below:

**Admin ER diagram**

****

**Figure: 1 Admin ER Diagram**

**Customer ER diagram**

****

**Figure: 2 Customer ER Diagram**

### **Design Constraints**

Like other software Bikes Ecommerce Store have some design or other constraints also. Therefore, some arrival design constraints are given below:

* Usages of outside regulation: Usually, the data passes from client to server through TCP/IP system or protocol but this software is not used any type of public-key encryption service such as Secure Sockets Layer (SSL) certificate. For this reason, this software has constraints to passing user data and causes for confidentiality, and integrity causes.
* Bandwidth limitations: This constraint may lose some server connection for a few technical mistakes (Sometimes depends on Hardware or Internet connection). For this cause, it needs to run the query afresh.
* Databases: JSON file is used in this software that user queries have more than servers’ limitations that it needs to observe and fresh table data or information. In this incident, there has a lack of DB caching.
* Communication protocols: For communication protocols, TCP/IP is used to interact with the server, and another protocol is not considered for user demand.
* Parallel operations: The parallel use of other internet applications with this software or project may hamper its bandwidth, and also may occur in the taking time for a query for slow connections.

#### **Standards Compliance**

For the standards compliance, it may cause (In case of Internet Security):

* **Confidentiality problem:** Here, sometimes the user, intended server fails “understanding” message contents and types.
* **Integrity problem:** Hither, the sender, server may fail to ensure message not altered without detection.
* **Authentication problem:** Server may not recognize/confirm actual valid user.
* **Impersonation:** Can fake (spoof) source address in packet (or any field in packet).
* **Denial of service:** Prevent service from being used by others (e.g., by overloading resources).
* **Eavesdrop:** There may be Intercept messaging, actively insert messages into connection.

### **Software System Attributes**

#### **Reliability**

Sometimes little third-party software is tried to use to build up this software or project. Therefore, these are free components; most of them are indicated as open source. This software has used a strong database component named JSON for protecting strongly, for this reason, this software will not affect by others. Moreover, this software is not used other things that indicate illegal, so it is the most reliable software to use frankly.

#### **Availability**

System’s availability requirements:

* User must have basic knowledge how to use computer or smart phone.
* Every device must have internet connection.
* The user must know English language for operate device properly, as the user   interface will be provided in English.
* Every device must have enough hardware resource for run the software.

#### **Security**

System’s security requirements:

* User authentication is required to access the system.
* A client or user must be a registered user to login to use the features of the Bike Ecommerce Store.
* Without proper authentication no one can login to the system or use the features.

#### **Maintainability**

OOP principles must be followed to make sure the system is maintainable and extendable.

### **Organizing the Specific Requirements**

#### **System Mode**

**Dashboard**

* Can view Bikes advertisement.
* Can order Bike.
* Can view reviews.
* Can find signup or sign in options.
* Can trace contract.
* Can know about this software.
* Can subscribe.

**Admin**

* Can register.
* Can login.
* Can view profile.
* Can update profile.
* Can make another admin.
* Can see order list.
* Can update order list.
* Can add products.
* Can manage products.

**Customer**

* Can register.
* Can login.
* Can view profile.
* Can update profile.
* Can view dashboard.
* Can view bikes.
* Can make order.
* Can view orders.
* Can cancel orders.
* Can give reviews.

#### **User Class**

There are two user classes. The customer who orders the bike and the admin.

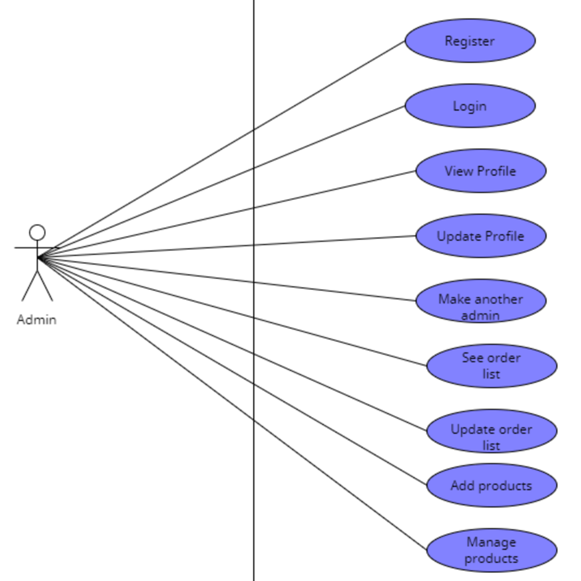


Figure: 3 Admin Class

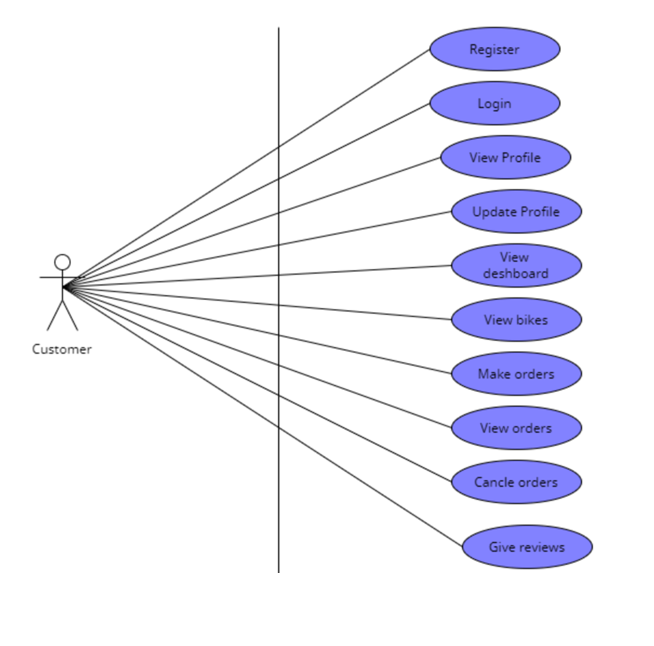


Figure: 4 Customer Class

#### **Response**

This software or application will have a graphical user interface, where the registered users can use it from any browser and so it is browser independent. The working modules:

Admin(owner)

Customer

* + The system will store the information of all registered users’ information, customer orders, and bike details, income of the software, place order information and accurate calculation.
  + As the admin, has to login to the system, it means there is an authenticated & secure login system and secure data transmission for all users.

#### **Functional Hierarchy**

The Bike Ecommerce Store software can make a functional hierarchy whenever they need extra employees as their requirements. But at present admin can make another admin if he or she wants and other things customers can add or create their ID by themselves.

### **Change Management Process**

### Change Management Process is a process that depends on system or software changes or additions. If the Bike Ecommerce Store changes its management process it will not hamper its performance and other things. It still capable,

* + To make satisfy the clients.
  + To consume time.
  + To finds brand new bikes.
  + To help reduce the usage of labor, as tasks such as having to take orders from customers is eliminated.
  + To less chances of data being lost or incorrect data entry.
  + To efficient, smooth and easy procedure of running a e-commerce site.
  + To make faster orders.

### **Document Approvals**

**LETTER OF APPROVAL**

11th June, 2022.

To,

MR. R

Associate Professor Computer science and engineering,

American International University of Bangladesh (AIUB)

Sir,

We have prepared the report on Software Requirements Specification of 'Bike E-commerce Store' for your approval. This report details the requirements we gathered for the project.

The primary purpose of this report is to summarize our findings from the work that we completed as our Software documentation and tools course project. This report includes the details of each step we followed to collect the requirements.

Sincerely Yours,

MD. REAZ UDDIN CHOWDHURY (21-92155-3)

UMMEY SOAIBA YEASMIN (21-92158-3)

AKIBUR RAHMAN AKIB (22-92573-2)

SHEULY AKTER (22-92609-2)

MD. RIFAT UZ ZAMAN (22-92610-2)

**Approved By: MR. R**

**MR.R**

**11-06-22**

### **Supporting Information**

The supporting information makes the SRS easier to use. It includes:

1. Table of Contents
2. Index
3. Appendices

The Appendices are not always considered part of the actual requirements specification and are not always necessary. They may include:

(a) Sample I/O formats, descriptions of cost analysis studies, results of user surveys

(b) Supporting or background information that can help the readers of the SRS

(c) A description of the problems to be solved by the software

(d) Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements

When Appendices are included, the SRS should explicitly state whether or not the Appendices are to be considered part of the requirements. Tables on the following pages provide alternate ways to structure section 3 on the specific requirements. You should pick the best one of these to organize section 3 requirements.

# **Software Design Document (SDD)**

## **Introduction**

### **Purpose**

This document is intended for all the stakeholders of the system especially for the development team so that everyone is on the same page and clear visualization about the requirements of the system before development can start. SDD gives a clear visualization to the user end as well as developer end. The requirements should reduce ambiguity as much as possible for the project to be successful. The stakeholders include the product owner, the various users, the project manager, the software engineers and designers. By using this user can easily make understand the concept to the developer.

### **Scope**

This software design document is intended for all the stakeholders but primarily for the development. team of the system so that everyone is on the same page and clear about the designs of the system before development can start. The design documents should reduce ambiguity as much as possible for the project to be successful. The stakeholders include the product owner, the various users, the project manager, the software engineers and designers. The E-commerce store will have two user classes

Customers

Admin

The customer will be able to browse through the different bikes, login/register and order a bike that he likes the admin will be able to login, manage the different product information, view and manage the orders and create other admins accounts.  
The system will allow users to quickly and easily view the technical specifications of each bike and make a purchase in the comforts of their home. They don’t need to go to a show room to view and buy the bikes. This saves the customer time and the owner of the system to serve more customers without hiring more sales respite Ecommerce platform will have an API and a react based frontend which will consume the API and provide functionality to the software.

### **Overview**

This SDD document is divided primarily into 6 parts. First part Introduction provides an overall functional requirement of the system written in consultation with the owner of the system. This section provides a overview of the users, how the system will be used by the various stakeholders and any constraints of the system. The other parts outline the system architecture and data design. This section will primarily be used by the development team which includes designers, developers and testers to build the final version of the system. It also shows the visualization of component design and human interface design. And finally ended with the requirement matrix.

### **Reference Material**

1. https://reactjs.org/
2. <https://www.mongodb.com/>
3. <https://sendgrid.com/solutions/email-api/>  
   <https://www.sendinblue.com/api/>
4. <https://www.twilio.com/docs/sms/api>  
   <https://rapidapi.com/blog/sms-apis-send-texts/>.

**Definitions and Acronyms**

* API- API stands for Application Programming Interface
* React- A JavaScript library for building user interfaces
* NodeJS A JavaScript library for building user interfaces
* Mongo DB – Schema less database

## **System Overview**

### **System Interfaces**

The system must integrate with a third part SMS API to provide OTP and other SMS

notification to the users. Some of the options for an SMS API is given below

https://www.twilio.com/docs/sms/api

<https://rapidapi.com/blog/sms-apis-send-texts/>

The system should also integrate with an email API to send invoices and other email

notifications to the users. So third part Email APIs answer given below

https://sendgrid.com/solutions/email-api/

https://www.sendinblue.com/api/

### **Hardware Interfaces**

The users of the system can use any JavaScript enabled modern browser on a mobile or

desktop device to access the web-based application. No other specific hardware interfaces are

required. A minimum of 1 GB RAM on the hardware device is recommended for smooth

performance.

### **Software Interfaces**

The customer currently does not have any software that needs to be interfaced with. So the

system will be standalone API and a frontend interface. We only have to interact with the

third-party email and SMS APIs to provide SMS and email functionality.

### **Communications Interfaces**

Our web application will interface with the backend REST API using the standard TCP/IP

over the internet. A basic overview of TCP/IP is linked below

<https://www.fortinet.com/resources/cyberglossary/tcp-ip>

### **Memory Constraints**

To make sure the application is accessible to as many customers as possible the devices with

low end RAM and processor should be able to smoothly run the application using a browser

without any performance issues. Based on our focus groups the RAM requirements are 1 GB.

### **Operations**

The typical operations of the bike ecommerce include inventory management and keeping

track of the model and number of bikes currently in stock. When a new bike included in

stock the admin will add the bike from the backend portal so that the new bike is visible to

the users. When a customer orders a bike, the admin will view the order and will change the

status of the order accordingly. There will be no payment gateway in the ecommerce store so

all the orders will have cash on delivery by default.

## **System Architecture**

**Architectural Design**

The primary backbone of the web app is a NodeJS based REST API which will be responsible to fetch and serve data to both the frontend app and the database server.

Two other third party APIs will be used to send SMS and email notifications The front end will be a separate app built using a JavaScript library called React. React is a very flexible library that can generate static sites with dynamic data. This will allow the site to be very responsive and have very low load times which are necessary for a high converting ecommerce store. A class diagram is provided below to demonstrate how each component of the application will interact with each other.

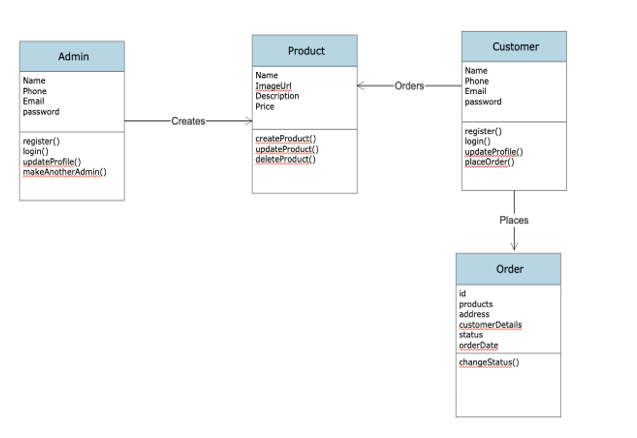


Figure: 5 Class Diagram

Use Case Diagram:

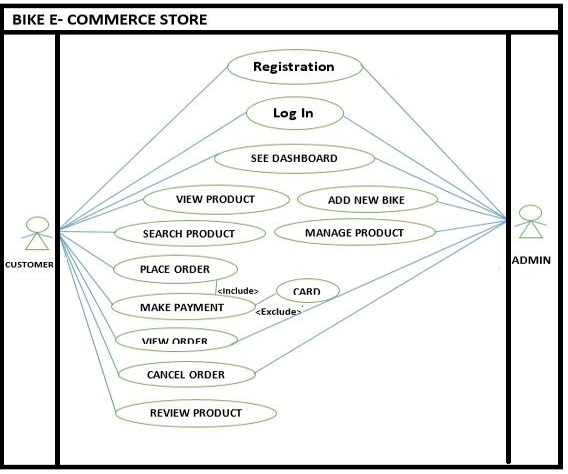


Figure: 6 Use Case Diagram

### **Decomposition Description**

The customer will order a bike after viewing and signing into the platform. A flow diagram for the customer is provided below:

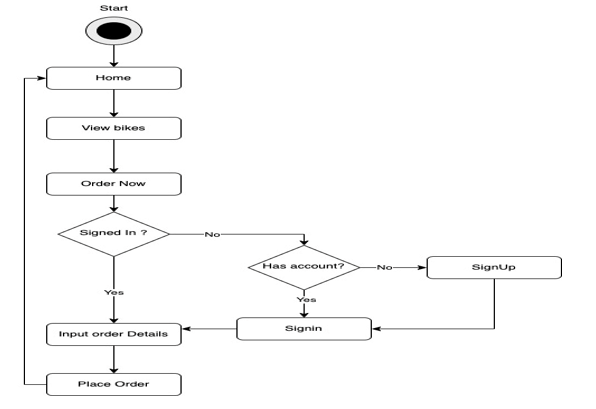


Figure: 7 Flow Diagram

### **Design Rationale**

REST API based architecture using NodeJS will allow very fast read/write operations. As it is well known node servers are not very good at compute heavy tasks but are very good at fast read and write operations. As an ecommerce store the web application will need to handle a lot of read and write requests but they won’t be very compute heavy. So, Node JS is used as the web server React is used for the frontend for its.

## **Data Design**

**Data Description**

Though we are using MongoDB which is a schema less database we will still need a database structure which will map into an ORM so that it will be easier for the development to follow the database structure. The ER diagram of the system is given bellow:

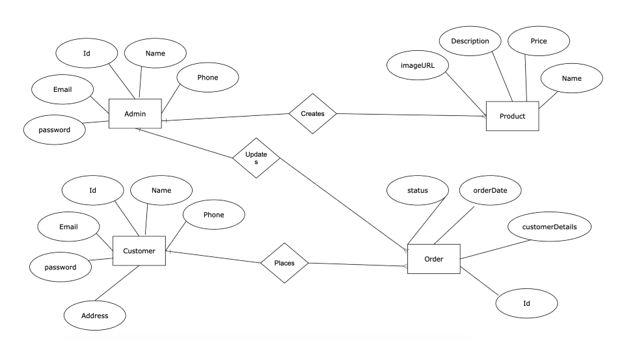
****

Figure: 8 ER Diagram

**Data Dictionary**

The data dictionary contains the datatype of each attribute based on the Entity Relation diagram and the keys that correspond to the relationship to it.

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity** | **Attribute** | **Datatype** | **Key** |
| Customer | Id | Int (32) | Primary Key |
|  | Name | Varchar (256) |  |
|  | Phone | Varchar (256) |  |
|  | password | Varchar (256) |  |
|  | email | Varchar (256) |  |
| Admin | Id | Int (32) | Primary Key |
|  | Name | Varchar (256) |  |
|  | Email | Varchar (256) |  |
|  | Password | Varchar (256) |  |
|  | Phone | Varchar (256) |  |
| Product | Id | Int (32) | Primary Key |
|  | Name | Varchar (256) |  |
|  | Price | double |  |
|  | Description | Varchar (256) |  |
|  | Image URL | Varchar (256) |  |
| Order | Id | Int (32) | Primary Key |
|  | Status | Varchar (256) |  |
|  | Items | Array (100) |  |
|  | Customer ID | Int (32) | Foreign Key |

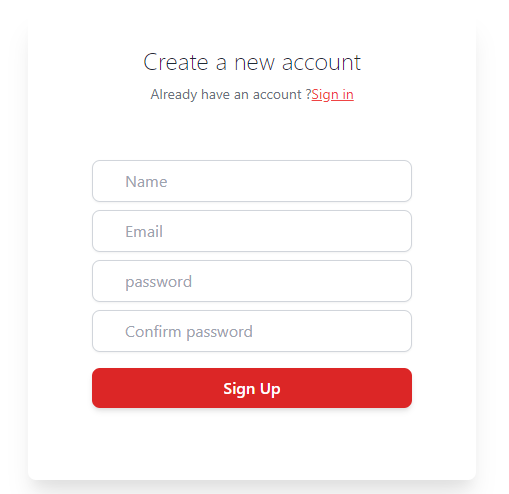
## **Human Interface Design**

**Overview of User Interface**

Describe the functionality of the system from the user’s perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user.

**Screen Images**

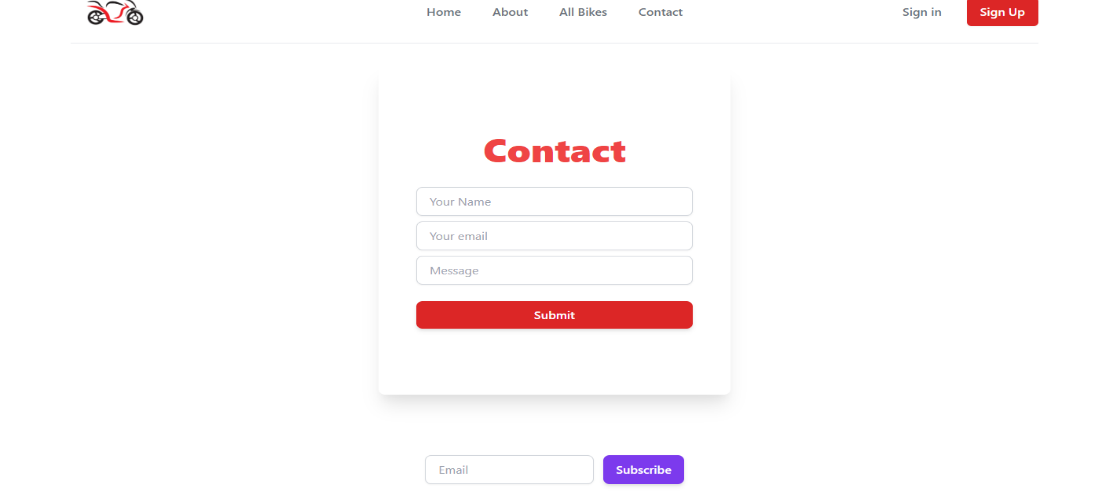
UI Wireframe

****

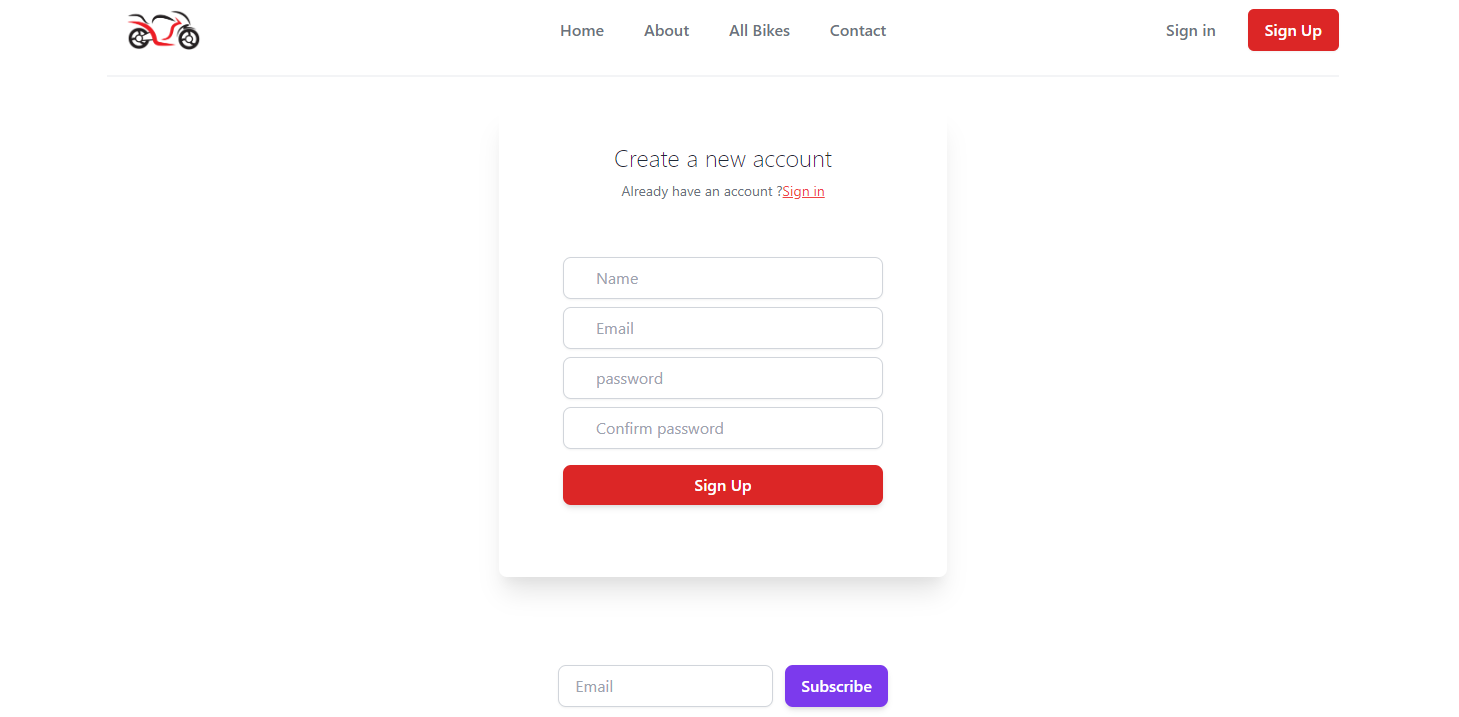
**Login interface**

## **https://lh3.googleusercontent.com/z6pBSpu29Bpo5QZbU_Q2ezXwKPhMiTBEEUAImVyYmHZ9UwWLOtvRJZ2JfkpXLTUO70Pf4oSn6iUEkuoGkrs3UoZRsJLMuwRQzj13BZcv2ssHVM0o0PGiKneA9qAALK9Q6sBqGH3OSs6E6SQxvw**

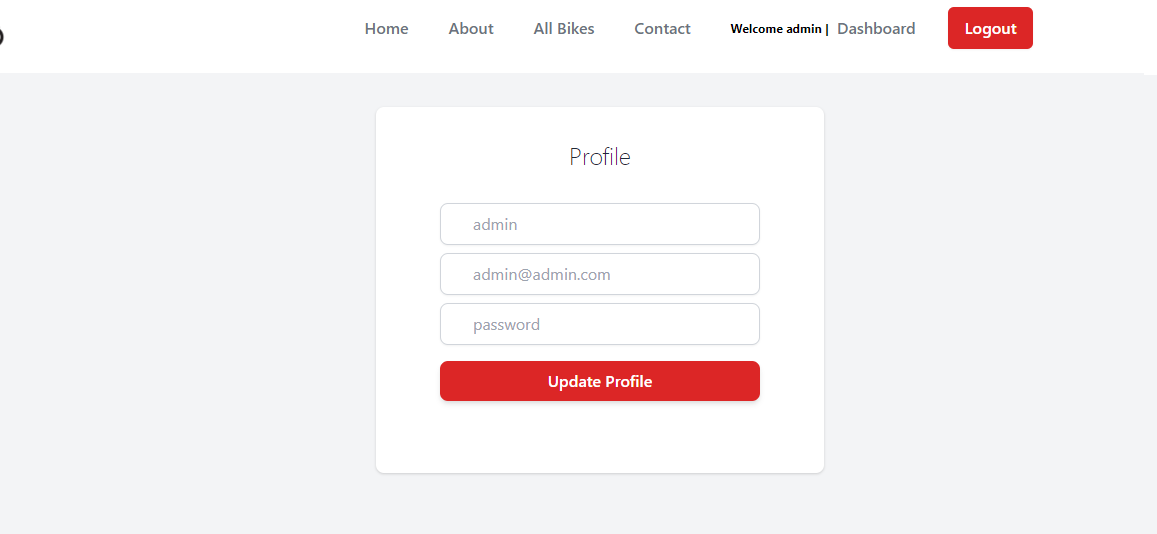
**View All Product**

****

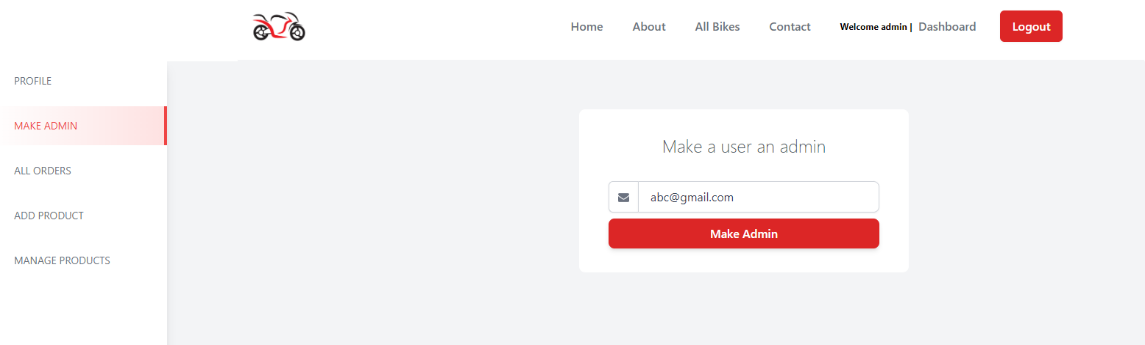
**Contact Page**

****

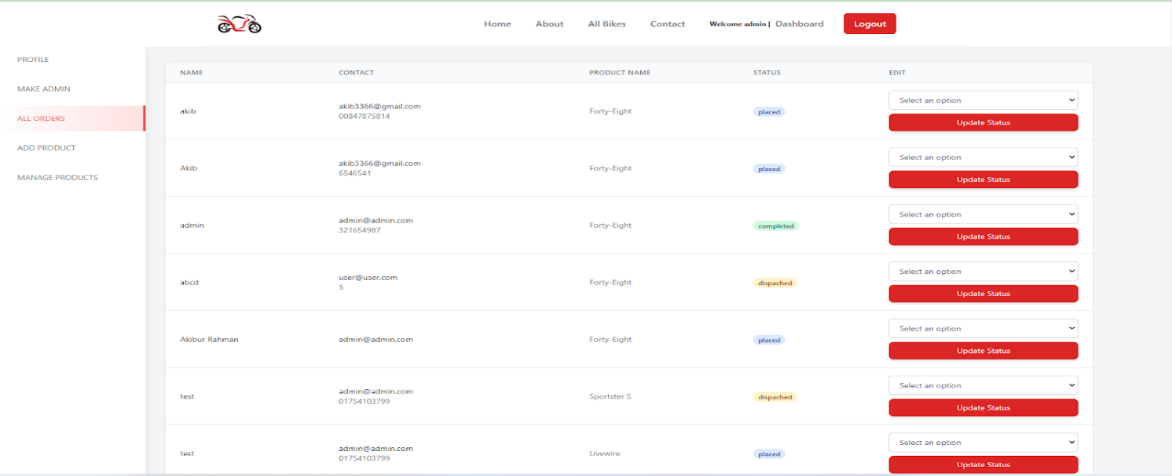
**Sign up page**

****

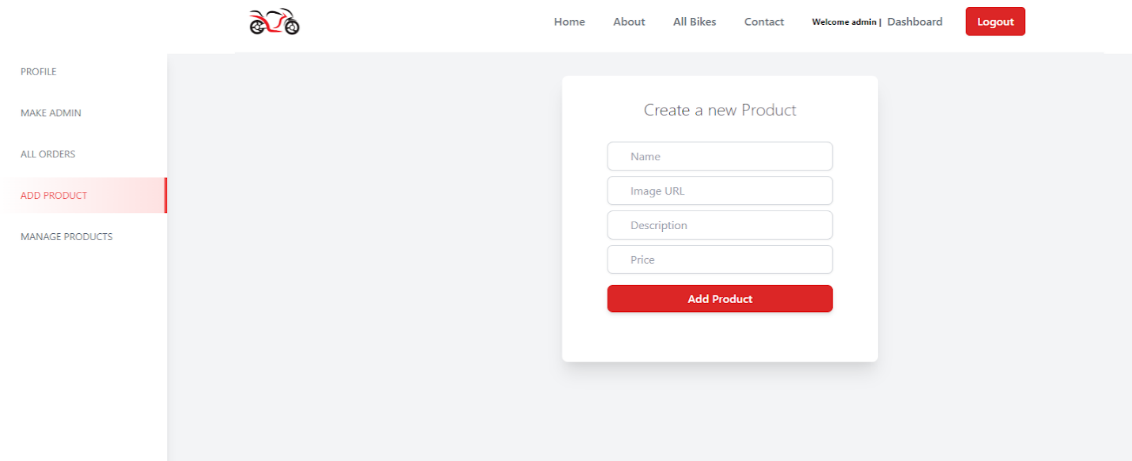
**update profile page**

****

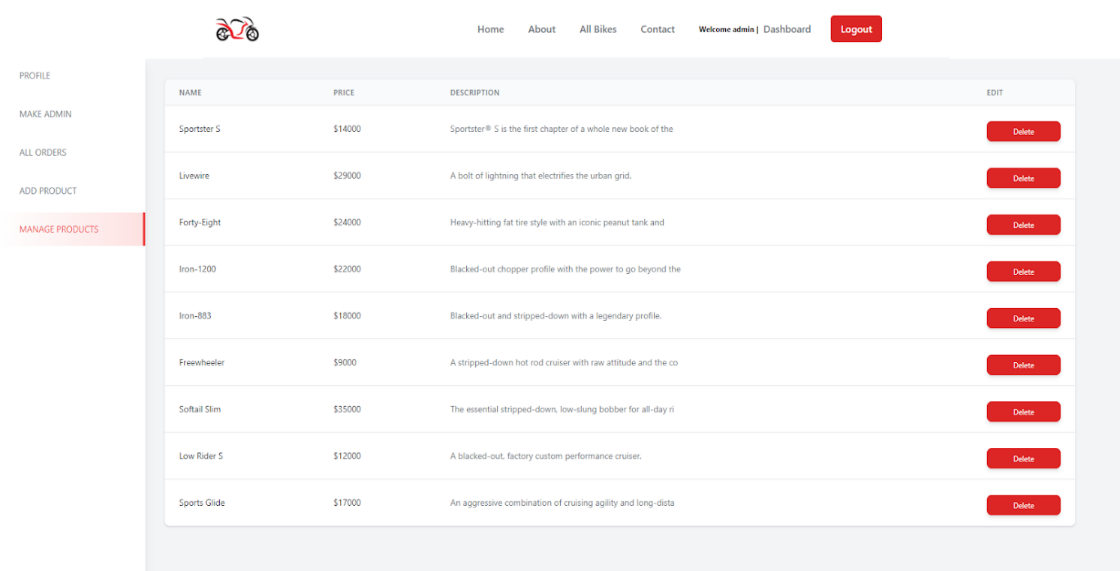
**Add user as admin**

****

**Admin can manage order**

****

**Add product**

****

**Manage product**

# **Software Development**

## **Overview**

The Software Development Document required to effectively defining the approach to be used in the document of the project’s product with the necessary information. The document is created during the Development Phase of the project. Its intended audience is the project manager, project team, and testing team. Some portions of this document may on occasion be shared with the client/user and other stakeholders whose input/approval into the developing process is needed.

### **Unit’s functionality**

Unit tests are typically automated tests written and run by software developers to ensure that a section of an application (known as the "unit") meets its design and behaves as intended. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. Therefore, for this software or site also ensure with its accurate application with best design.

### **Development methodologies**

Here, the Agile methodology and more specifically Scrum was used during the Development process where the requirements have been divided into tasks. Each sprint was one week long. It was estimated that it would require 16 sprints to complete development. Trello was used as a scrum board for collaboration and to keep track of progress.

### **Source code listing**

<https://github.com/Akibur/ranger-bike-msc>

### **Controlled libraries/directories/tables**

<https://github.com/Akibur/ranger-bike-msc>

### **Unit test results and analysis**

**st r**

**For Customer:**

**Registration:** If all customer information is filled properly according to the guidelines like email must contain @Gmail, the name will start by alphabet, etc. Then it will pass otherwise fail.

**Login:** If the login id and password match with the registered customer id and password then it will pass otherwise fail. Another way is if the forget password works with the OTP it is passed otherwise fails.

**View Profile:** If the customer login is successful then the view profile will work and if all features work properly in the profile, then it is passed otherwise fail.

**Update Profile:** If the email address is confirmed as the registered Gmail, then it passed and if it is not then failing.

**View Dashboard:** If the Customer order list is visible to the system and works regularly then it passed otherwise fails.

**Make Order:** If the ordered products are available and it moved to the purchase option then it passed it not then fail.

**View Order:** If a customer order is placed successfully then the view order is visible and passed also but if it is not then it failed.

**Cancel Order:** If the Customer filled the canceled order form with a valid reason then it will pass and move to the next and if the form is not filled then it will fail.

**Give Review:** If customer products are delivered successfully then the customer can fill the review form for admin approval if the form is filled then the phase is passed otherwise failed.

**For Admin:**

**Registration:** If all admin information is filled properly according to the guidelines like email must contain @Gmail, the name will start by alphabet, etc. Then it will pass otherwise fail.

**Login:** If the login id and password match with the registered admin id and password then it will pass otherwise fail. Another way is if the forget password works with the OTP it is passed otherwise fails.

**View Profile:** If the admin login is successful then the view profile will work and if all features work properly in the profile, then it is passed otherwise fail.

**Update Profile:** If the email address is confirmed as the registered Gmail, then it passed and if it is not then failing.

**Make Another Admin:** Registered admin can make admin by providing admin form if it is properly filled with the requirement then it will pass otherwise fail.

**View Order list:** Admin can view order list when an order is confirmed by customer and then the test will pass otherwise fail.

**Update Order list:** Admin can update order list like order condition, order time, order date etc. when an order is confirmed by customer and then the test will pass otherwise fail.

**Add New Products:** Admin can add new product by providing product information like name, id, quantity etc. and if all these fields are complete then it is passed otherwise failed.

**Manage Products:** Admin can manage product by using the product manage way if it is successful then the test is passed otherwise failed.

## **Roles and Responsibilities**

The team members have the following roles and responsibilities:

* The application developer assigned the primary responsibility for the module or unit creates a file folder for the unit, labels it according to the name of the unit, and places it in the appropriate place in the project team file cabinet.
* The application developer(s) add copies of the indicated documentation to the folder as they are created.
* The project QA representative reviews the contents of the folder for completeness, and points out discrepancies to the developer assigned primary responsibility for tile module or unit.
* The developer assigned primary responsibility for the module or unit completes the Software Development Document Check-Off sheet and arranges for the System Technical Lead review and approval when needed.
* The folder is available to all project team member for review, but if removed from the file cabinet, it must be replaced with a cheek-out card indicating who checked it out, when, and where it will be located.

## **Process**

The following sections are consisting of the Check-Off sheet:

### **Requirements –**

For the Bike Ecommerce Store, we need some specific requirements such as

* External interfaces,
* Functions (admin functionality, and customer functionality),
* Performance,
* Logical database,
* Design constraints (standards compliance),
* Software system attributes, and
* Organizing the specific

All mentioned things are described into the software requirements specification document. Here the significant thing is all those things are justified and comes with outstanding outcome.

### **Functionality –**

**For Customer:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Checkmark** | **Item to Test** | **Test Description** | **Test Date** | **Responsibility** |
|  | Registration | Must full fill all valid information and must sent OTP to assign number for confirming account. | 22/06/2022 | Developer |
|  | Login | Using valid user id and password avail to see dashboard | 22/06/2022 | Developer |
|  | View Profile | Can avail to see their given information in a tabular form that saves into the database | 24/06/2022 | Developer |
|  | Update Profile | Changing details accepted, saves into the database and gives notification | 26/06/2022 | Developer |
|  | View Dashboard | Can able to view and scroll bikes details from the system | 26/06/2022 | Developer |
|  | Make Order | Without any hassle, capable to make order and also get notify | 27/06/2022 | Developer |
|  | View Order | Avail to see place order with exact time, date, place by storing a tabular form | 29/06/2022 | Developer |
|  | Cancel Order | New order will be accepted and get notify | 29/06/2022 | Developer |
|  | Give Review | Submitting reviews store by a tabular form and admin monitor or get notify | 1/07/2022 | Developer |

**For Admin:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Checkmark** | **Item to Test** | **Test Description** | **Test Date** | **Responsibility** |
|  | Registration | Full filling valid information and OTP sent to assign number for confirming account. | 1/07/2022 | Developer |
|  | Login | Giving valid user id and password avail to see dashboard | 2/07/2022 | Developer |
|  | View Profile | Avail to see given information in a tabular form that saves into the database | 4/07/2022 | Developer |
|  | Update Profile | Newer details accepted, saves into the database and gives notification | 7/07/2022 | Developer |
|  | Make Another Admin | Capable to save new admin information and get notify both admin | 8/07/2022 | Developer |
|  | View Order list | Receiving order saves into the database with all details | 10/07/2022 | Developer |
|  | Update Order list | Capable to change place order list and notify both admin and customer | 13/07/2022 | Developer |
|  | Add New Products | Adding newly products saves into the database and shown into the dashboard | 15/07/2022 | Developer |
|  | Manage Products | With whole details managing products saves to database by a tabular form and avail to see those into the site | 18/07/2022 | Developer |

### **Source Code –**

<https://github.com/Akibur/ranger-bike-msc>

### **Libraries, Directories, and Tables –**

<https://github.com/Akibur/ranger-bike-msc>

### **Development Methodologies –**

For the Bike Ecommerce Store, here the Agile methodology and more specifically Scrum was used during the Development process where the requirements have been divided into tasks. Each sprint was one week long. It was estimated that it would require 16 sprints to complete development. Trello was used as a scrum board for collaboration and to keep track of progress.

* Agile
* Scrum

### **Test Data –**

During the testing and using all data necessary to conduct testing referenced in this software.

### **Test Analysis –**

**For Customer:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Checkmark** | **Item to Test** | **Test Description** | **Test Date** | **Responsibility** |
|  | Registration | Must full fill all valid information and must sent OTP to assign number for confirming account. | 22/06/2022 | Successfully done with outstanding outcome |
|  | Login | Using valid user id and password avail to see dashboard | 22/06/2022 | Successfully done with outstanding outcome |
|  | View Profile | Can avail to see their given information in a tabular form that saves into the database | 24/06/2022 | Some error detected and go to the developers again |
|  | Update Profile | Changing details accepted, saves into the database and gives notification | 26/06/2022 | Some error detected and go to the developers again |
|  | View Dashboard | Can able to view and scroll bikes details from the system | 26/06/2022 | Successfully done with outstanding outcome |
|  | Make Order | Without any hassle, capable to make order and also get notify | 27/06/2022 | Successfully done with outstanding outcome |
|  | View Order | Avail to see place order with exact time, date, place by storing a tabular form | 29/06/2022 | Successfully done with outstanding outcome |
|  | Cancel Order | New order will be accepted and get notify | 29/06/2022 | Some error detected and go to the developers again |
|  | Give Review | Submitting reviews store by a tabular form and admin monitor or get notify | 1/07/2022 | Successfully done with outstanding outcome |

**For Admin:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Checkmark** | **Item to Test** | **Test Description** | **Test Date** | **Responsibility** |
|  | Registration | Full filling valid information and OTP sent to assign number for confirming account. | 1/07/2022 | Successfully done with outstanding outcome |
|  | Login | Giving valid user id and password avail to see dashboard | 2/07/2022 | Some error detected and go to the developers again |
|  | View Profile | Avail to see given information in a tabular form that saves into the database | 4/07/2022 | Successfully done with outstanding outcome |
|  | Update Profile | Newer details accepted, saves into the database and gives notification | 7/07/2022 | Successfully done with outstanding outcome |
|  | Make Another Admin | Capable to save new admin information and get notify both admin | 8/07/2022 | Some error detected and go to the developers again |
|  | View Order list | Receiving order saves into the database with all details | 10/07/2022 | Successfully done with outstanding outcome |
|  | Update Order list | Capable to change place order list and notify both admin and customer | 13/07/2022 | Some error detected and go to the developers again |
|  | Add New Products | Adding newly products saves into the database and shown into the dashboard | 15/07/2022 | Successfully done with outstanding outcome |
|  | Manage Products | With whole details managing products saves to database by a tabular form and avail to see those into the site | 18/07/2022 | Successfully done with outstanding outcome |

### **Sign-Off –**

The sign off date 20-07-2022.

## **System Developer**

I certify that this software development document is complete, the unit \_1, 2 defined in this folder has successfully completed development and unit testing, and the unit is ready to be base lined and integrated into the system

Date: 20-07-2

System Developer:

1. MD. RIFAT-UZ-ZAMAN
2. SHEULY AKTER

System Technical Lead Initials: AKIBUR RAHMAN AKIB

# **Software Testing**

## **Introduction**

### **Purpose of the test plan document**

The Test Plan documents and tracks the necessary information required to effectively define the approach to be used in the testing of the project’s product. The Test Plan document is created during the Planning Phase of the project. Its intended audience is the project manager, project team, and testing team. Some portions of this document may on occasion be shared with the client/user and other stakeholders whose input/approval into the testing process is needed.

## **Functional Testing**

### **Test Risk**

Based on the framework, the testers can easily execute the tests and obtain relevant     results by generating test reports. But selecting the right tool to automate Website depends on many key parameters.  It is always important to compare the available tools based on key parameters like features, performance, extensibility, licensing cost, maintenance cost, and Training and support. Continuous changes for demands Regression so run regression test suits every day to keep track of the effects of change.

### **Items to be tested/not tested**

|  |  |  |  |
| --- | --- | --- | --- |
| **Item to Test** | **Test Description** | **Test Date** | **Responsibility** |
| Customer Register | OTP will be sent to the given mobile number and the account | 1/7/2022 | Successfully done |
| Customer log in | After successfully login by using user Id, password and can view dashboard | 1/7/2022 | Successfully done |
| Customer view profile | System shows all given information in a tabular form with all details that saved | 4/7/2022 | Fail, go to developers |
| Customer update profile | Change information is accepted and gives notification | 7/7/2022 | Successfully done |
| Customer view dashboard | Able to see their desire or wanted bikes with proper details from the site | 10/7/2022 | Successfully done |
| Customer make order | Capable to make order without any types of hassle and as well as can get notification | 13/7/2022 | Successfully done |
| Customer view order | Place order will be shown and recorded by a tabular form with date, time, and place | 15/7/2022 | Fail, go to developers |
| Customer cancel order | The new order can be done by the customer and get notification | 17/7/2022 | Successfully done |
| Customer give review | Reviews will be stored by a tabular form and after submitting that admin can monitor those reviews | 19/7/2022 | Successfully done |
| Admin register | OTP will be sent to the given mail or mobile number and also confirm the account | 21/7/2022 | Successfully done |
| Admin login | By given correct user Id and password, Admin can login to the site that is stored into the site storage | 22/7/2022 | Successfully done |
| Admin view profile | Given all information will be visible to admin those stored into the database | 23/7/2022 | Successfully done |
| Admin update profile | System update the profile and sends notification to assign mobile number or mail Id and saves details | 23/7/2022 | Successfully done |
| Admin make admin | System saves new admin details and sends notifications both admin | 25/7/2022 | Successfully done |
| Admin see order | Given entire order store to the database and sends all details to admin | 27/7/2022 | Fail, go to developers |
| Admin update order | Admin can update place order list whenever he needs and sends notification to both customer and admin | 29/7/2022 | Successfully done |
| Admin add product | Newly products will be added by admin and saves details to the database as well as shown dashboard | 31/7/2022 | Successfully done |
| Admin manage product | Admin will be able to manage products that save to site database and shows details in a tabular form | 2/8/2022 | Successfully done |

### **Test Approach’s**

A website/application works differently for different user profiles such as customers (both authorized and unauthorized), sales representatives, and online shop managers. It is, therefore, important to make sure that your website test cases cover the varied operations for all types of users such as filtering of items, adding/removing goods to the shopping cart, and so on. To offer a hassle-free experience to customers, it is pivotal to test the complete workflow of your web/mobile application that consists of login and signup options, sorting feature, search functionality, add/remove functionality in the shopping cart, check-out process, payment gateway, and payment processing among others. An increasing number of users are now using mobile devices, driving companies to reconsider their testing strategy, and take a mobile-first approach in their applications. The important aspect here is to test the responsive design of the application in mobile devices of different screen sizes.

### **Test Regulatory**

Test regulatory changes when there is a disconnect between regulatory adherence and product development. This disconnect happens when regulations are tracked by different teams, and in different systems than those used to track requirements and testing on the product side. Sometimes it’s because of an institutional gap those on the product team simply aren’t regulatory experts, and vice versa. Other times it is a tool gap, where regulations can’t be tracked within the same tool as these other processes.

### **Test Pass/Fail Criteria**

In 18 items that we have tasted,15 items will done successfully and 3 items become fail then it returns to the developers for solve

**Test Entry/Exit Criteria**

 The testing entry criteria are the Customer Register so that customers can login into the system will be sent to the given mobile number and will be confirmed by the account.

The testing exit criteria is Admin manage product that admin can manage products such as its price, details, stock and so on.

### **Test Deliverables**

### A record of the consequences of the take a look at the side of maps and graphs displaying CPU operation, reminiscence operation and database connections used over the length of the take a look at.

## **Load Testing**

### **Test Risk**

Load testing will be done on the frontend react app and on the API endpoints. We need to make sure that we don't do it on a live server after deployment as this can cause performance issues for users.

### **Items to be tested/not tested**

|  |  |  |  |
| --- | --- | --- | --- |
| **Item to Test** | **Test Description** | **Test Date** | **Responsibility** |
| API | API will be tested to check ability to serve 1000 users simultaneously | 3/7/22 | Tester |
| React Front end | Check responsiveness of the react app under 1000 user load | 3/7/22 | Tester |

### **Test Approach’s**

Load testing will be done system-wide to test the API and frontend app simultaneously. The main objective of the test will be to test the stability and responsiveness of the system under a large load. Automation tool will be used to simulate users using the system.

### **Test Regulatory**

System must be responsive even when a thousand users are using it simultaneously.

### **Test Pass/Fail Criteria**

Test will pass if the system remains responsive when a thousand users are using theSystem for 10 minutes. Test will fail if the system becomes unresponsive

### **Test Entry/Exit Criteria**

**Entry Criteria:** Test automation tool will simulate 1000 users and start performing different actions on the system.

**Exit Criteria:** If the system remains responsive for 10 minutes or if the system   becomes unresponsive

### **Test Deliverables**

A report on the results of the test along with charts and graphs showing CPU usage, memory usage and database connections used over the period of the test.

### **Training Needs**

Test will be performed on a clone of the live server so that the live application isn't   affected. A tester will use automation tools to simulate 1000 users performing standard tasks on the application like registering, login, ordering etc.

## **Unit Testing**

For Bike Ecommerce Store, unit tests will be avail to develop for testing all interfaces of this software or site all of the components on the side the low-level unit tests will be suitable to develop by testing methods of the initial classes in the components. Therefore, as a prerequisite of this software or site, the automatic unit test suite is to run by building server on a prebuild basis options. Mainly, the unit testing of software testing indicates a method that works on every individual units of source code of the software (which sets of one or more software modules with associated control data, usage of procedures, and operating procedures) are tested to determine that suitable for an application or user. Moreover, the white-box testing method is used in Unit testing and it is generally done by the programmer.

### **Test Risk**

**For Customer:**

**Registration:**

For customer registration tests risks or issues can be occurred when the tester gives any invalid information for new registration and oversight typing of code by a developer. After that low network connections, misunderstanding any option, and not storing all information properly in the database can be causes of these test risks.

**Login:**

Typing mistake of pre-registered user id or password is the major test risk or issue for the login page or feature. Moreover, the server connections, false program typing, and storing data problems are also other possible test risks or issues.

**View Profile:**

Storing duplicate data that customer gives during the registration will be the main indicated issue that will occur hassle for the customer. Lacking database functionality, and programming, not giving the right information will be shown as a problem.

**Update Profile:**

Not giving modern or new information to the customer properly, and retype the previous one while the customer wants to update the profile this issue will be found. At the same time, the storage capacity of this site or software or network connection, and mistaking compose of code will be found as a test risk or issue of update profile.

**View Dashboard:**

After successfully registration and login dashboard is found but the cause of invalid login or registration, network issue, and faux enter of codding by developer the view dashboard issue will be shown.

**Make Order:**

To buy bikes, sometimes whole details not shown properly or will not take order for network issue, and error entry of programming.

**View Order:**

Not storing order details into the database for complex network issue and lacking of coding, view order will create test issues.

**Cancel Order:**

After conforming order, sometime customer cannot cancel order because of expired date and server problem at the same time coding functionality also causes of test issues.

**Give Review:**

Could not store review in a tabular form cause of functional complexity and server maintaining at the same time submitting button sometime not working, those indicates the test risk or issues of this software or site.

**For Admin:**

**Registration:**

For admin registration tests risks or issues can be occurred when the tester gives any invalid information for new registration, used pre-defined name or information and oversight typing of code by a developer. After that low network connections, misunderstanding any option, and not storing all information properly in the database can be causes of these test risks. Must have ensure about admin registration form is correct.

**Login:**

Typing mistake of admin login pre-registered user id or password is the major test risk or issue for the login page or feature. Additionally, the server connections, false program typing, and storing data problems are also other possible test risks or issues.

**View Profile:**

Storing duplicate data that admin gives during the registration will be the main indicated issue that will occur hassle for the customer. Lacking database functionality, and programming, not giving the right information will be shown as a problem.

**Update Profile:**

Not giving modern or new information to the customer properly, and retype the previous one while the customer wants to update the profile this issue will be found. At the same time, the storage capacity of this site or software or network connection, and mistaking compose of code will be found as a test risk or issue of update profile.

**Make Another Admin:**

Making another admin, the person must be admin and full fill all requirements that store information and notify both admin but cause of sufficient storing capacity it creates problem and also misunderstanding of system requirements and error codding is also cause of this test risk

**View Order list:**

Lacking capacity of the database storing of order list, submitting button issues, and server problem will indicate view test issue or risk.

**Update Order list:**

Sometimes cause of network or server cause, mistake of programming, and time limitation can be cause of update order list test issue.

**Add New Products:**

Not avail to storing details of adding new products into the database and functional complexity creates test issue as well as matching older products information creates test issues.

**Manage Products:**

Misunderstanding of system requirements, change request for the system, and matching older version of product with the newer versions or products with complex codding or functionality of the system is causes of this test risk.

## **System Testing**

When we get to the system tests, we should have completed the unit and integration tests and has the software fully loaded in a test environment. The main purpose of system testing is to evaluate the system meets the specified requirements. The black box test method is used in system testing. System testing should be done by a professional tester. In our Bike Ecommerce Store system, we focus on the following:

* **Usability testing**: The user interfaces of the system are easy to use and easy to understand or not.
* **Documentation test** - When the customer uses the system, the system guide explains the functions as they work in real life. We will check it.
* **Functionality test**: As described in the documentation, the system behaves as expected or not.
* **Interoperability test**: Check that the system works with different operating systems, browsers, plug-in, etc. Properly or not.
* **Performance Test** - Performance Test is the most important test for our Ecommerce system. We need to verify that our system can meet the requirements for speed, responsiveness, and stability under load. And it will help ensure a more positive user experience.
* **Security testing**: Our System can check testing validation (the input and output like **User’s Name, User’s Password**) or not.

## **User Acceptance Testing**

### **Test Risk**

In software testing Risks are the possible problems that might endanger the objectives of the project stakeholders. It is the possibility of a negative or undesirable outcome. A risk is something that has not happened yet and it may never happen; it is a potential problem. In this project there have some ricks problem issue those are described below:

**For Customer:**

* **Registration:** The first way to enter to this software is to do the registration. And for that there have some specified way to complete the registration. Test ricks in registration like there must have send the OTP in the mail. The mail account must include @gmail.com. Main ricks on it that if the Gmail is not correct then there is no alternative way to register in the system. There have some problems in OTP like sometimes there have some network issue for that reason OTP send in delay. Another test risk is that the given OTP will be send only in specified Gmail only.
* **Login:** A user only can enter in the login if the registration is successful. After that using user id and password he/she can enter to the system. Ricks issue in login is that sometimes the user password is wrong and user tried several times then the I'd got blocked. So, there will be a solution that immediately the user can solve the forgetting password using that registered mail.
* **View Profile:** Test issues in the user profile is sometimes some button may not work due to network issue. As the project is on-line based so when the network has some problem the system will automatically provide the notification.
* **Update Profile:** Issues in update profile is that if the user mail given during the registration period is not same then user can't update profile. And during the update there will be send an OTP in the mail and for network issue sometimes it being late to deliver the OTP. Then the user face problem updating the profile.
* **View Dashboard:** After placing order sometimes it need more time to adjust in the dashboard. After successful update it shows in the dashboard. And for this network issue user have to wait till the dashboard update.
* **Make Order:** Due to bad network sometimes, user face issue in placing order. They press order button but it may not work due to network issue. That time the user has to order it again from the system.
* **View Order, Cancel Order:** To view order and cancel order have some technical issues like without giving the valid reason user can't cancel order and there need the admin approval. Sometimes admin is busy in some duty so that's why it delayed sometimes. And due to network issue user have some delay to browse the placed order.
* **Give Review:** User can give review of product after successful delivered to the user. Something it need some time to update the delivery status for that user can't give review. So, issues in giving review user have to wait until the product shows delivered in the system.

**For Admin:**

* **Registration:** For Admin registration the admin details is saved in the system automatically. And for that there have some specified way to complete the registration. Test ricks in registration is sometimes defined Id and Password may have some error like the mail account must include @gmail.com. Main ricks on it that if the Gmail is not correct then there is no alternative way to register in the system. There have some problems in OTP like sometimes there have some network issue for that reason OTP send in delay. Another test risk is that the given OTP will be send only in specified Gmail only.
* **Login:** An admin only can enter in the login if the system has predefined admin id and password. After that using predefined admin id and password, he/she can enter to the system. Ricks issue in login is that sometimes the admin password is wrong and admin tried several times then the I'd got blocked. So, there will be a solution that immediately the admin can solve the forgetting password using that registered mail.
* **View Profile:** Test issues in the admin view profile is sometimes some button may not work due to network issue. As the project is on-line based so when the network has some problem the system will automatically provide the notification.
* **Update Profile:** Issues in update profile is that if the admin mail given during the registration period or the predefined time is not same then admin can't update profile. And during the update there will be send an OTP in the mail and for network issue sometimes it being late to deliver the OTP. Then the admin face problem updating the profile.
* **Make Another Admin:** As admin information is predefined in the system so it is very difficult for an admin to create another admin. For that admin have to enter the predefined system to create another admin. And there have some technical ussies like admin have to be specified and have some network issues as well.
* **View Order list, Update Order list:** To view order and update order have some technical issues like admin have to enter the order list and if there is some network issue the it will not enter. Sometimes system is busy in some updating issue so that's why it delayed sometimes. And due to network issue admin have some delay to browse the placed order.
* **Add New Products, Manage Products:** Adding new product and manage them have issues like sometimes that specified product is not available or there have some stockout issue but due to bad network or because of system updating it have some delay and for that adding new product and manage product have some interruption.

**Phase: 1 Testing Deliverables:**

|  |  |
| --- | --- |
| Items Name | Date |
| Master Test Plan | May 2022 |
| System Test Results Document | June 2022 |
| Acceptance Test Results Document | July 2022 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SL No | Test Case | Condition | Expected Result | Actual Result | Status | Snapshot Link | Comment |
| 1 | Admin | Register in the system | I can login into the system | OTP will be sent to the given mail or mobile number and also confirm the account | Fail | https://drive.google.com/file/d/12bvZhoSnJQCes-m-Vga-yB6jX6KXlrGp/view?usp=sharing | N/A |
| 2 | Admin | Log in to the system | I can perform all types of my activities carefully | By given correct user Id and password Admin can login the site that are store into the site storage | Fail | https://drive.google.com/file/d/1pPYP0AYmslt3B28ZnykS9fORLnIfAWIP/view?usp=sharing | N/A |
| 3 | Admin | View Admin profile | I can see my profile | Given all information will be visible to admin those stored into the database | Fail | https://drive.google.com/file/d/1TMzJHKtiH7d9aNLBB9hQXnyCX7oQyYTl/view?usp=sharing | N/A |
| 4 | Admin | Update Admin profile | I can update my account whenever I need | System update the profile and sent notification to assign mobile number or mail Id and saves details | Pass |  | N/A |
| 5 | Admin | Make another admin | I can divide my responsibility and work properly timely | System saves new admin details and sends notifications both admin | Pass | https://drive.google.com/file/d/1Pw2Zo\_fxTFskba-Hh9WRzStcEpPF6vB2/view?usp=sharing | N/A |
| 6 | Admin | See placed order list | I can see all order list to take action and do other activities | Given entire order store to the database and sends all details to admin | Fail | https://drive.google.com/file/d/1-ssL9-KXhfo5g5B5AE0mo6X9pLvGWtoo/view?usp=sharing | N/A |
| 7 | Admin | Add new products | I can add newly comes bikes details to the site | Newly products will be added but admin and saves details to the database as well as shown dashboard | Fail | https://drive.google.com/file/d/1fI4SDnRue9NFHQz\_ckIfIfHu8UmwmuRf/view?usp=sharing | N/A |
| 8 | Admin | Manage products | I can manage my products such as its price, details, stock and so on | Admin will able to manage products that save to site database and shows details in a tabular form | Pass | https://drive.google.com/file/d/1ef9hK2CXFToknF4FH6ng6V1S1my-tnuB/view?usp=sharing | N/A |
| 9 | Admin | Update placed order list | I can change placed order list if any barrier circumstance comes | Admin can update place order list whenever he needs and sends notification to both customer and admin | Pass | https://drive.google.com/file/d/1N-qqFVoRDu31az2DH4w8ODUkjaHGMNrW/view?usp=sharing | N/A |
| 10 | Customer | Register in the system | I can login into the system | OTP will be sent to given mobile number and will be confirm the account | Pass | https://drive.google.com/file/d/12bvZhoSnJQCes-m-Vga-yB6jX6KXlrGp/view?usp=sharing | N/A |
| 11 | Customer | Log in to the system | I can able to perform my activities | After successfully login by using user Id, password and can view dashboard | Pass | https://drive.google.com/file/d/1pPYP0AYmslt3B28ZnykS9fORLnIfAWIP/view?usp=sharing | N/A |
| 12 | Customer | View Customer profile | I can see my saved or given information | System shows all given information’s in a tabular form with all details that saved | Pass | https://drive.google.com/file/d/1wR8cnO8hMONtsKjEFVJZ332zzrc4Ytc4/view?usp=sharing | N/A |
| 13 | Customer | Update Customer profile | I can update my profile whenever I need to change any information | Change information are accepted whenever it necessary for him or her and gives notification | Pass | https://drive.google.com/file/d/1Y-ONP6cuP8wLoxVQZ6f7vGqWhJFijJr3/view?usp=sharing | N/A |
| 14 | Customer | View Dashboard | I can see all bikes with their details, prices for purchase | Able to see their desire or wanted bikes with proper details from the site | Pass | https://drive.google.com/file/d/1Y-ONP6cuP8wLoxVQZ6f7vGqWhJFijJr3/view?usp=sharing | N/A |
| 15 | Customer | Make an order to the system | I can purchase my suitable bike | Capable to make order without any types of hassle and as well as can knowledge about purchasing bikes | Pass | https://drive.google.com/file/d/1IVR6oOVJDCtBHPaDrancJF8A5Zbd7181/view?usp=sharing | N/A |
| 16 | Customer | View placed order | I can ensure all kinds of ordering requirements for my satisfaction. | Place order will be shown and recorded by a tabular form with date, time, and place | Pass | https://drive.google.com/file/d/1ceXko5Y1SRVoE3r5Lz\_m5F8tkCLCJA\_K/view?usp=sharing | N/A |
| 17 | Customer | Cancel placed order | I can take confirm about ordering bikes | The new order can be able to do by the customer and get notification | Fail | https://drive.google.com/file/d/1ceXko5Y1SRVoE3r5Lz\_m5F8tkCLCJA\_K/view?usp=sharing | N/A |

**Phase: 2 Testing Deliverables:**

|  |  |
| --- | --- |
| Items Name | Date |
| System Test Results Document | July 2022 |
| Acceptance Test Results Document | August 2022 |

**Hardware components required**

* 1 Network Controller
* 6 Networked PC's (See below)
* 1 DAP Workstation
* 1 Oracle Server
* 1 HP LaserJet 4v Printer

**PC Specifications**

* The 6 PCs required for the test environment will include the following:
* 1 x P100, 1 GB SSD, and 16Mb RAM [Current Minimum Specification]
* 3 x P166, 1.5 GB SSD, and 32 Mb RAM [Current Standard Specification]
* 1 x P333, 2.5 GB SSD, and 64 Mb RAM [Current Maximum Specification]

These specifications are the various specifications currently in use in different branches. 1 x Pentium running Windows NT is also required as the Test center for controlling and executing the automated testing.

**Software required**

**Test Tool:** Selenium, postman for automated testing

**Server:** Oracle database server, MySQL Workbench, Install MySQL server and Apache server

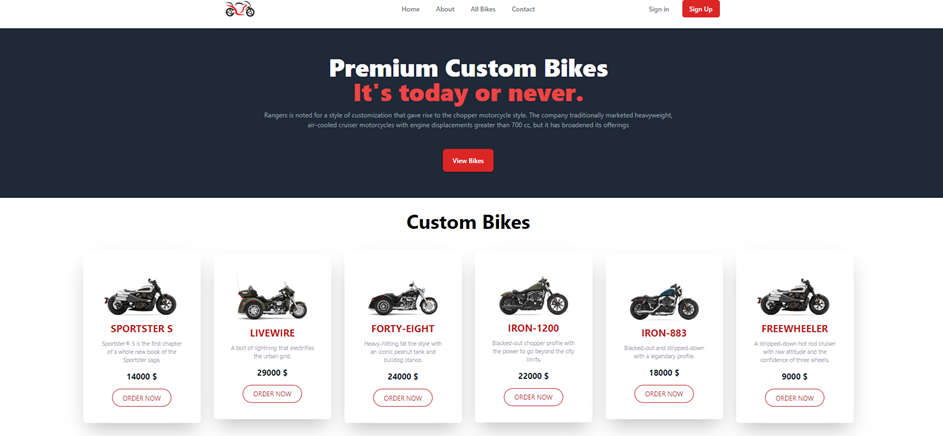
**System:** Windows 10 professional with latest Microsoft 365 access.

**Network:** Setup a LAN gigabit and 1 internet line with a least 10 MB/s

# **User Manual**

### **Bike E-comers System Dashboard**

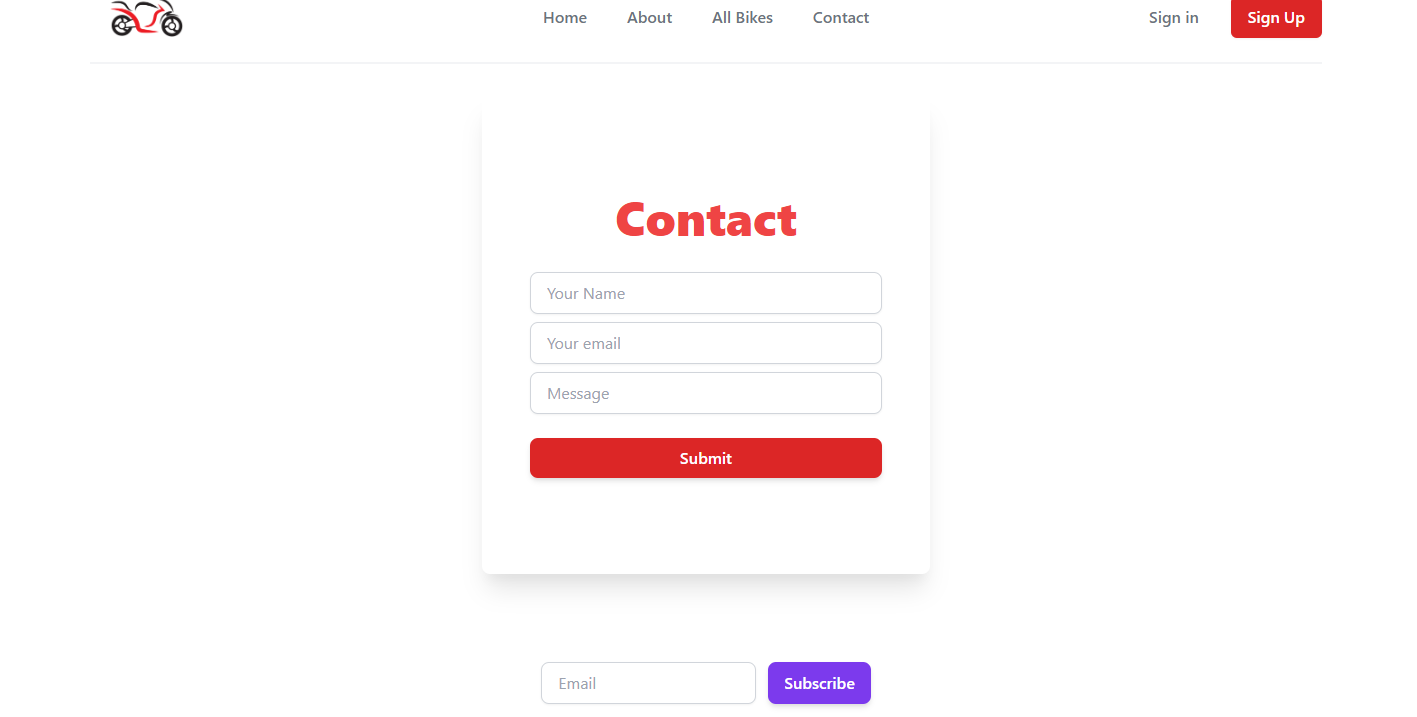
In the home page the customer will view info about the company and view some of the best bikes displayed in the home page.

****

## **Bike E-comers Support System**

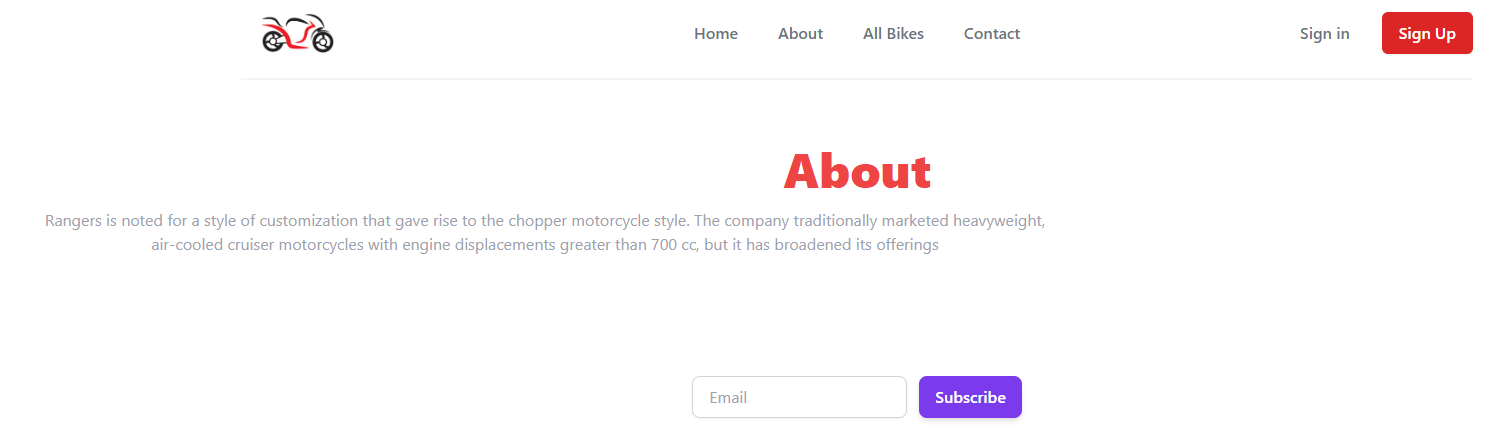
The contact page will be used to contact the owner of the platform by filling up the form.

**Contract Details**

****

**About Details**

The about page will contain details about the company

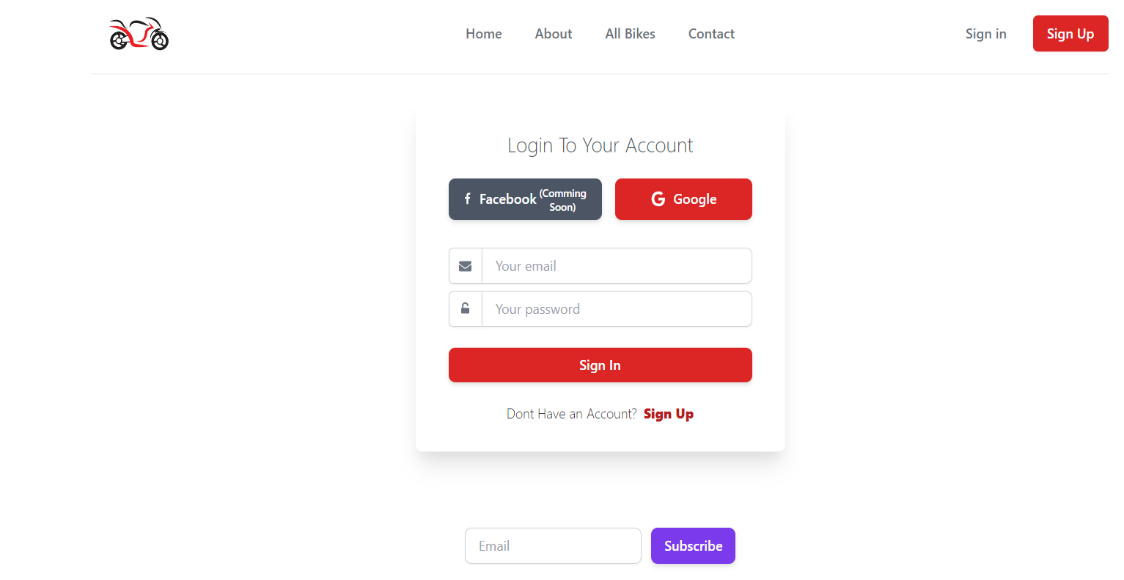
****

## **Login System**

The user will be able to login using the login form. Using his email and password. The can also login using google sign in

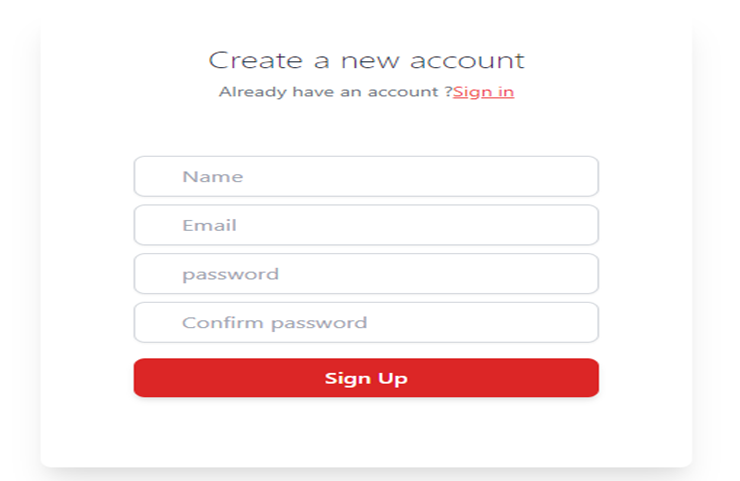
**Login Sign**

**login form**

****

**New registration**

The user will be able to register using the signup form.

****

## **profile management**

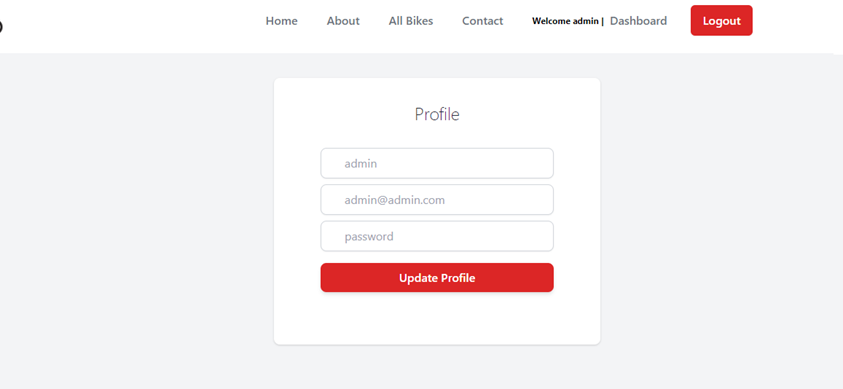
### **update profile**

After login in a user will be able to update his profile using the profile update form

**Admin feature**

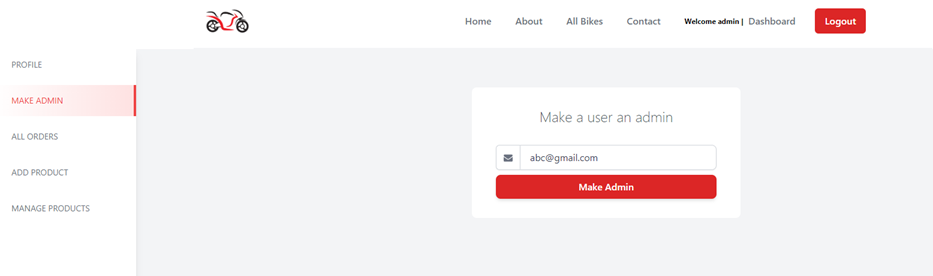
**admin profile**

After login in an admin will be able to update his profile using the profile update form

****

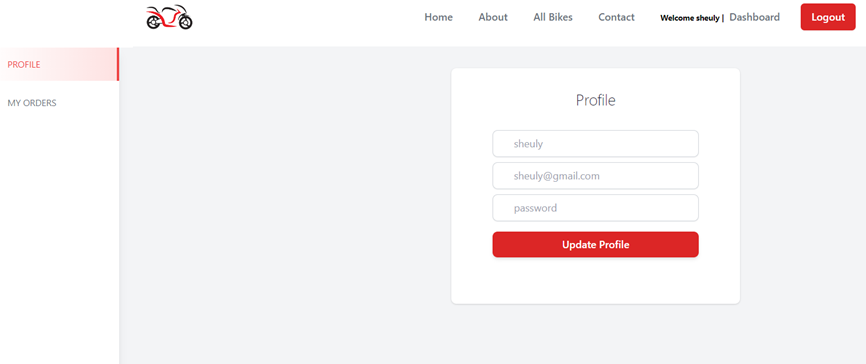
**make another admin**

An admin will be able to create another admin using his email address. Initially there will only be one system admin

****

## **customer feature**

**customer profile**

****

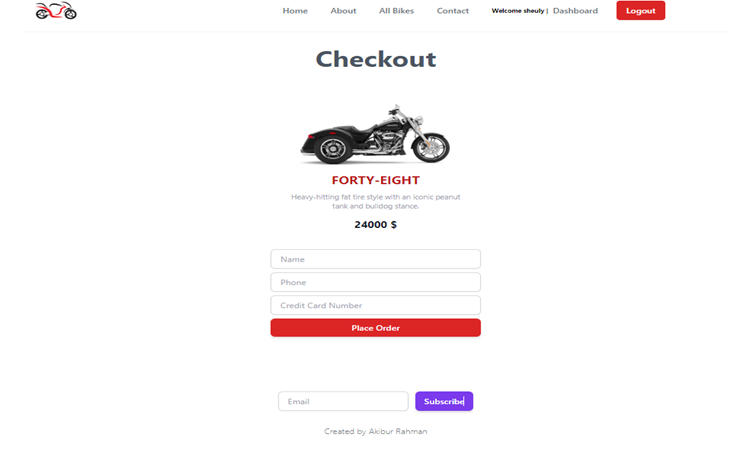
## **Bike details**

## **https://lh4.googleusercontent.com/UOON3TVisXSAyVsOHboDm5yLSQQZsRmRI0P-YYSPDU_xVE5RhUfwHVG1pYrKlQN8M7XlnzD-DOaKO6ExGzY5LkTow-k6wrlPf5mwZXEtakhfWr2kZvDmLZD2KZmajgWjYryu1fJJBtFHwgJENEt75g**

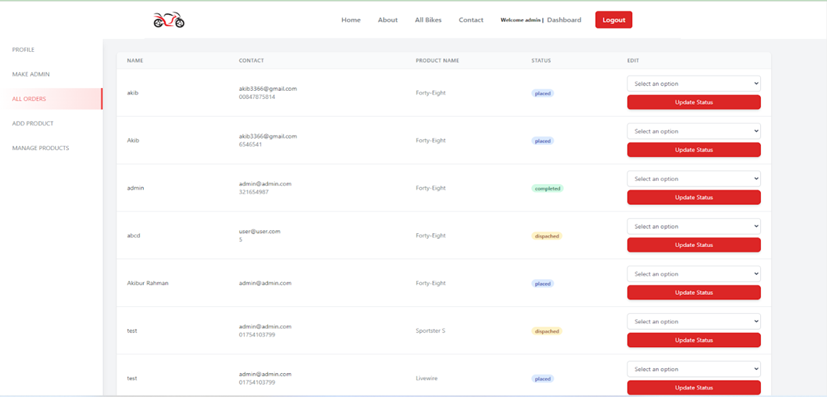
## **Order system**

The admin will be able to view and change the status of an order

**order details**

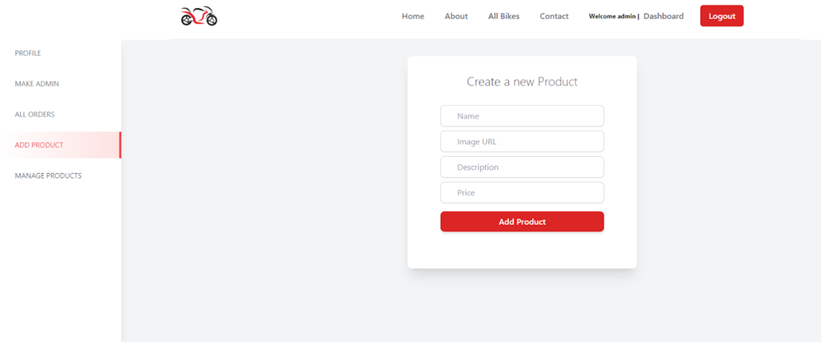
****

**order management**

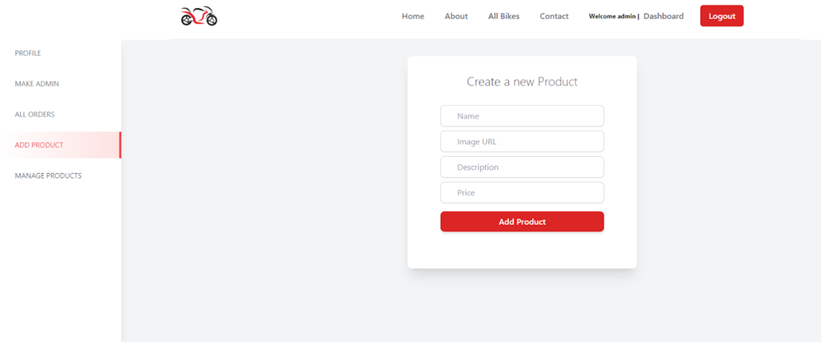
****

## **Add New product**

The admin will be able to add a new bike by filling up the add bike form.



## **Logout**

****

# **Tools Demonstration Screenshots**

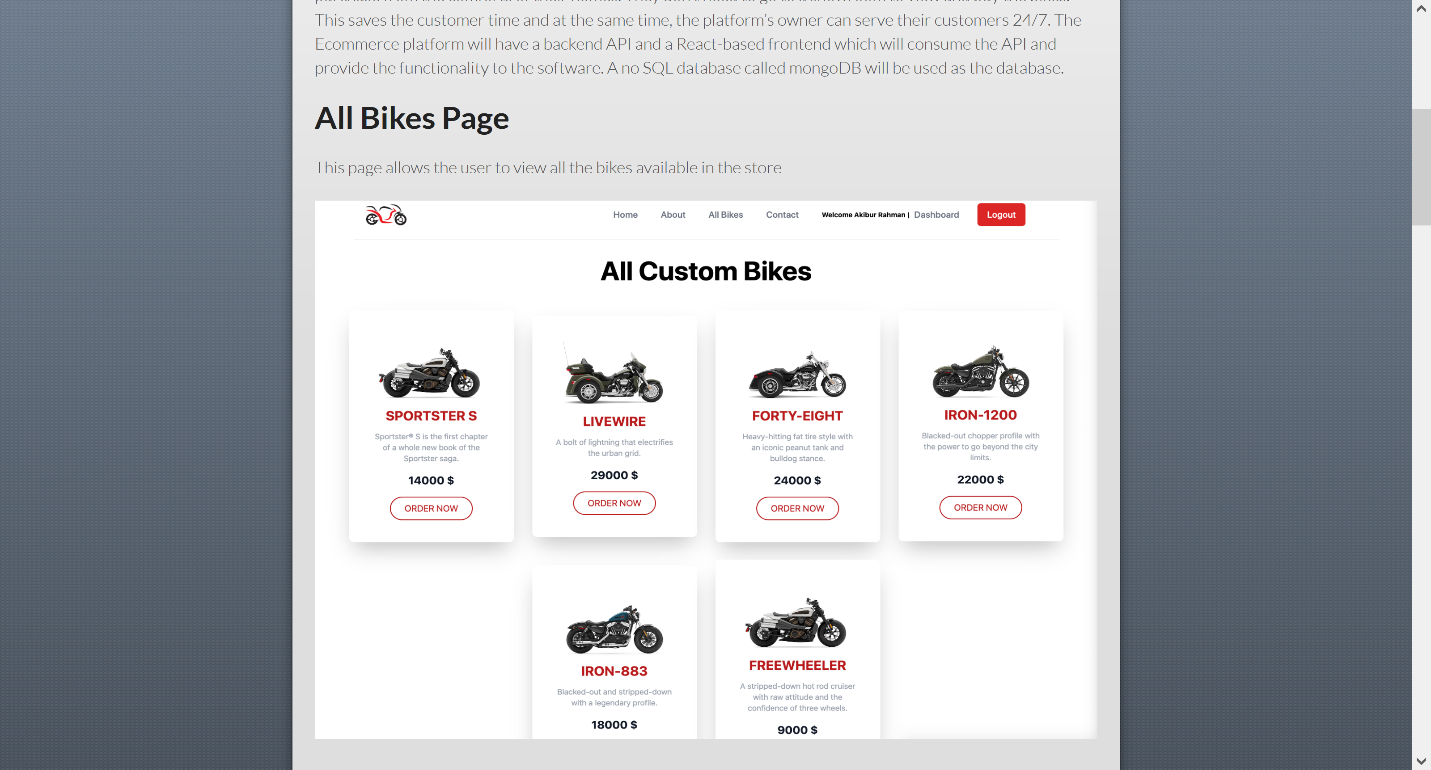
****

Fig 9: All Bike Page

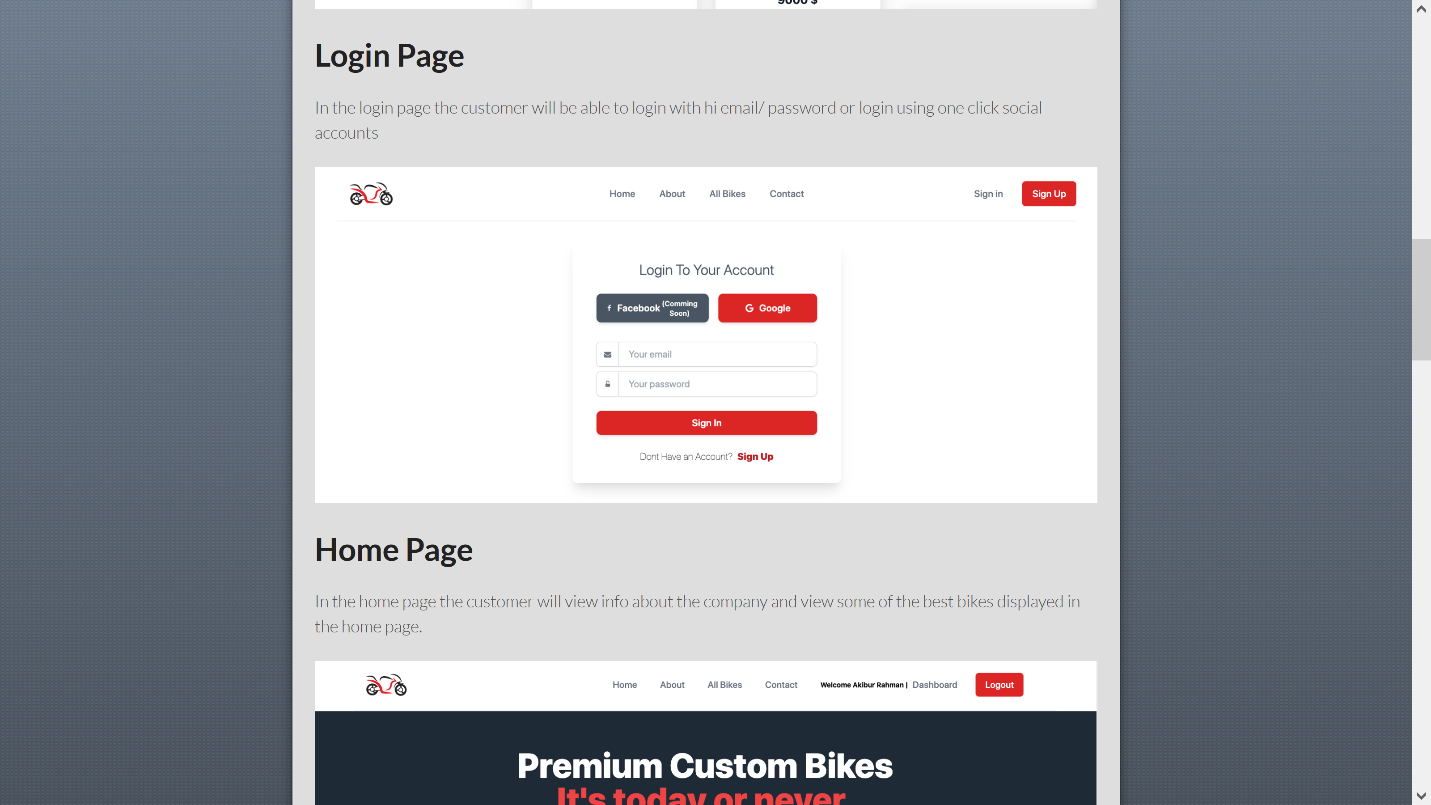
****

Fig 10: Login Page

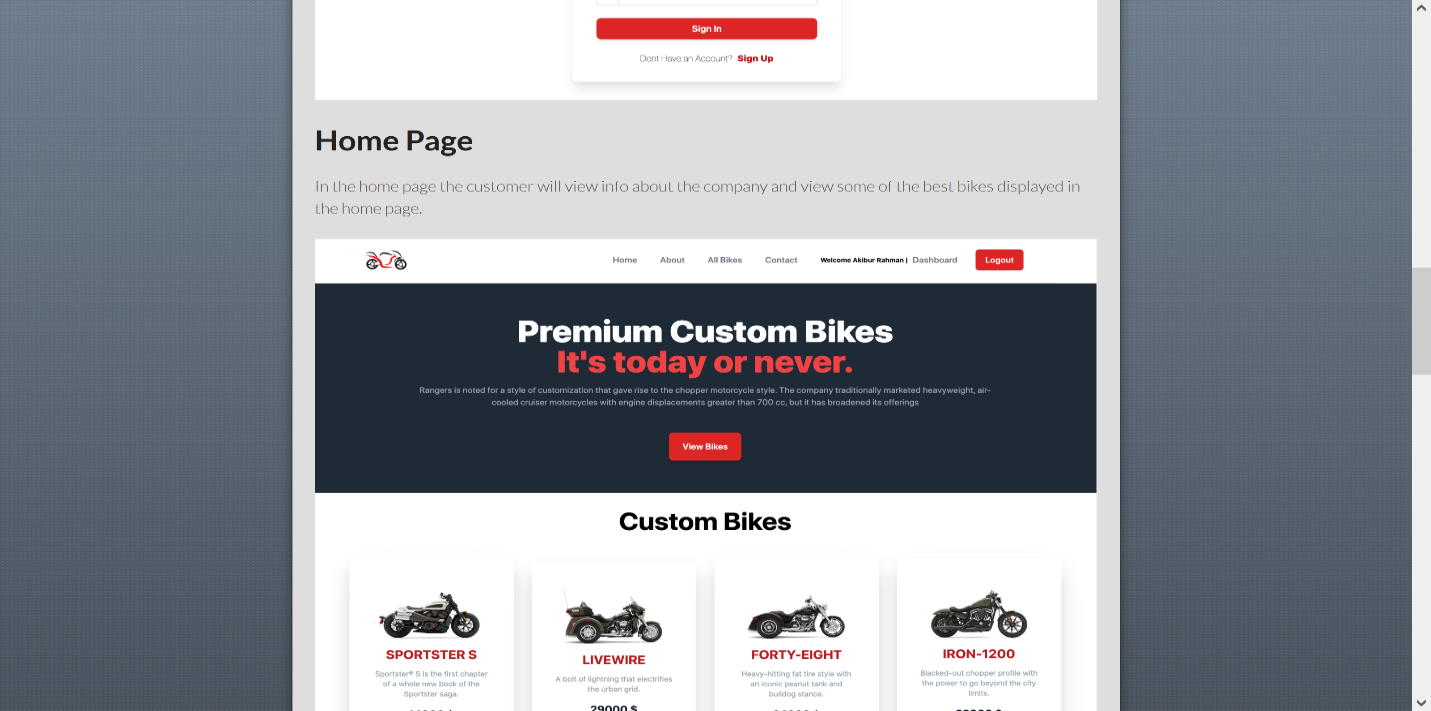
****

Fig 11: Home Page

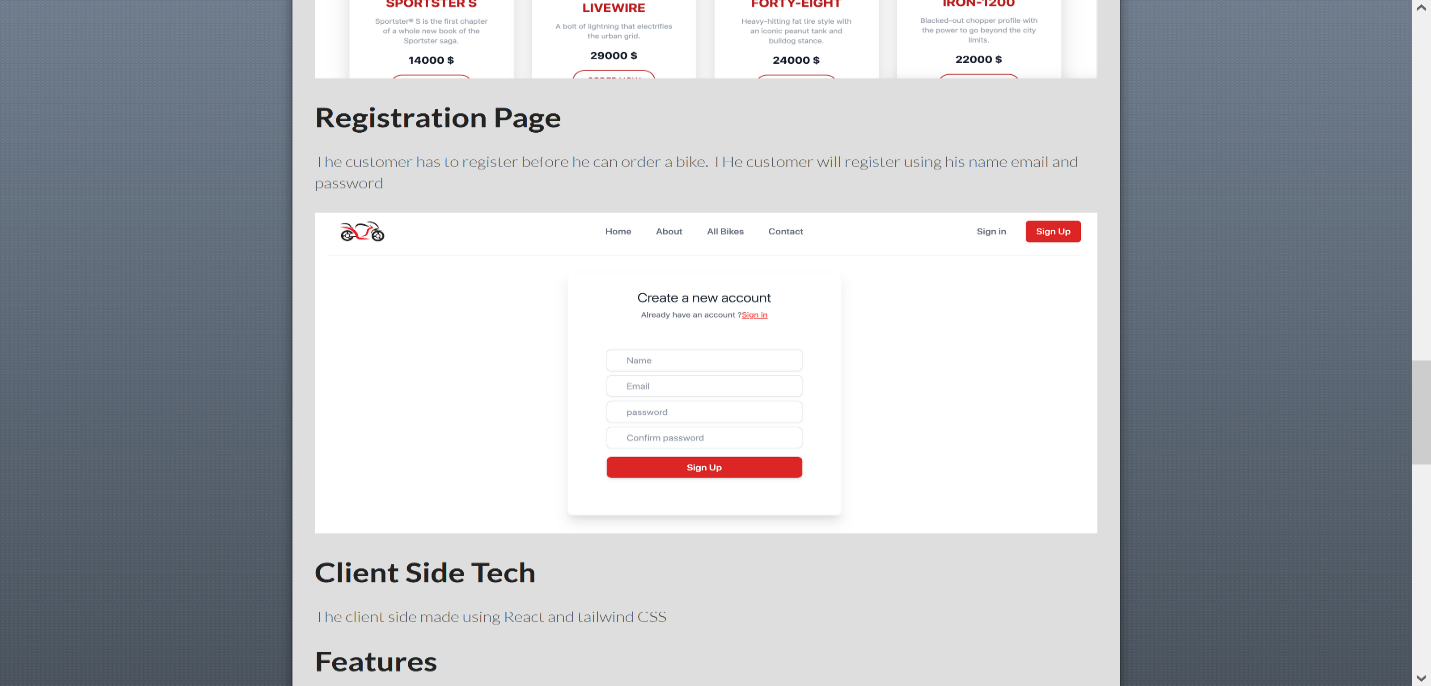
****

Fig 12: Registration Page