

# Computing Project - Ramitha 20199209 - Sameen 20198986- Akalanka 20199208 - Akila - 20197288- Yashoda 20199207 .docx

*by Sitinamaluwe Paranapalliya Guruge Ramitha Amayuru*

---

**Submission date:** 18-Jul-2022 07:25PM (UTC+0100)

**Submission ID:** 184181431

**File name:** 81883\_Sitinamaluwe\_Paranapalliya\_Guruge\_Ramitha\_Amayuru\_Computing\_Project\_-  
\_Ramitha\_20199209\_-\_Sameen\_20198986-\_Akalanka\_201992\_113711738.docx (6.78M)

**Word count:** 12170

**Character count:** 59110



# E-COMMERCE WEB SITE (ALLINONE.LK)

## NAMES OF THE STUDENTS

1. RAMITHA GURUGE.
2. YASHODHA DILSHAN.
3. AKILA JAYASANKA.
4. SAMEEN RANSARA.
5. AKALANKA HASINDU.

## GROUP NO 05

1  
A PROJECT SUBMITTED TO

INTERNATIONAL COLLEGE OF BUSINESS AND TECHNOLOGY

(ICBT)

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE

HD in Computing and Software Engineering

JULY 2022

## **ACKNOWLEDGEMENT**

Thanks to all those who helped us to complete this project created by us and to all those who gave us guidelines to complete the project. It was a great pleasure for all of us to complete this project. We all express our heartfelt gratitude to all those who enlightened us by providing the necessary guidance to complete the project through various tutorials. We would also like to express our deep gratitude to all those who have directly and indirectly guided us to complete this project.

Also, we would like to thank Deshan Sir, Upeka Miss, and Priyanga Sir who pointed out the shortcomings of our project and advised us to correct them from the day the project started till the end. Also, team members have given valuable comments, suggestions, and innovative ideas to this project which motivated us to improve our project. We would like to thank everyone who helped us complete our project.

## Table of Contents

<b>ACKNOWLEDGEMENT</b>	2
<b>1</b>	
<b>Table of figures</b>	7
<b>Table of tables</b>	8
<b>CHAPTER 01</b>	9
• <b>Introduction</b>	9
• <b>Problem statement</b>	10
• <b>Literature Review</b>	11
• <b>Proposed solution</b>	12
• <b>Justification of the Solution</b>	13
• <b>Functionalities &amp; Non functionalities of the system</b>	14
1. <b>Functional Requirements</b>	14
I. <b>Register</b>	14
II. <b>Login</b>	14
III. <b>Admin User Level Details</b>	15
1. <b>Update User details</b>	15
2. <b>Delete Users</b>	15
3. <b>View ad</b>	15
4. <b>Post ads</b>	16
5. <b>Update Ad Details</b>	16
6. <b>View Customers Feedbacks &amp; Messages</b>	16
7. <b>Delete Posted ads</b>	17
IV. <b>User Level Details</b>	17
1. <b>View ad</b>	17
2. <b>Post ads</b>	17
3. <b>Update Ad Details</b>	18
4. <b>Delete Posted ads</b>	18
5. <b>Direct chat with Admins</b>	18
6. <b>Give the feedbacks</b>	19
v. <b>Logout</b>	19
2. <b>Non-Functional Requirements</b>	20
I. <b>Security</b>	20
II. <b>Performance</b>	20
III. <b>Usability</b>	20
IV. <b>Accuracy</b>	20

V. Availability .....	20
VI. Platform Independents .....	20
• Software Development methodology.....	21
<b>CHAPTER 02.....</b>	<b>22</b>
<b>Feasibility study report.....</b>	<b>22</b>
1. Technical Feasibility:.....	22
• <i>Resources Available.</i> .....	23
• Technology.....	23
2. Economic Feasibility:.....	24
• Cost benefit analyses.....	24
3. Operational feasibility:.....	25
3. Legal feasibility:.....	25
Requirement gathering.....	26
Resource identification .....	27
Work breakdown structure.....	28
Risk Analysis.....	29
1. Technical Risk: .....	29
2. Project Management Risk:.....	29
3. Organizational Risk:.....	29
4. External Risk:.....	29
Risk Assessment.....	30
<b>1 Chapter 03.....</b>	<b>31</b>
• Design of the system.....	31
○ Database.....	31
✓ Entity relationship diagram (ER Diagram).....	31
✓ Normalized relation schema.....	32
● System Design.....	35
○ DFD.....	35
✓ Context diagram.....	35
✓ Level 1 Admin DFD.....	35
✓ Level 1 User DFD.....	36
✓ Level 2 User DFD.....	37
✓ Level 2 User DFD.....	37
● Pseudo code.....	38
1. Login.....	38

2. Add Categories.....	38
3. Add Products.....	38
4. Update Categories.....	39
5. Update Products.....	39
6. Delete Categories.....	39
7. Delete Products.....	40
8. Add To Card.....	40
9. Buy Product.....	40
10. Search.....	41
• User Interfaces / Navigations.....	42
1. User Interfaces.....	42
1. Home Page.....	42
2. Product Page.....	43
3. Vehicle Details Page.....	44
4. Computer Details Page.....	45
5. Product Payment Page.....	46
6. Post Ads.....	47
7. Admin Home Page.....	48
8. Admin Sub Category page.....	48
9. Admin User page.....	49
10. Update Category Page.....	49
11. Update User Page.....	50
12. Create Category.....	51
13. Create Sub Category.....	51
14. Add User.....	52
15. Available Stock Manage.....	53
16. Payment Page.....	53
17. Login Page.....	54
18. Register Page.....	54
Chapter 04.....	55
• Explain co-functionalities with evidence.....	55
1. Login Page.....	55
2. Register Page.....	57
3. Database Connection file.....	59
4. Post Ad.....	60
5. Create category.....	62

6. Update Category.....	63
7. Delete category.....	65
<b>Chapter 05.....</b>	<b>66</b>
1. Testing.....	66
○ Testing methodology and justification.....	66
○ Test plan.....	67
○ Test cases.....	75
○ Test conclusion.....	111
2. User Manual.....	111
Product Page.....	112
Computer Details Page.....	114
Product Payment Page.....	115
Post Ads.....	116
Admin Home Page.....	117
Update Category Page.....	121
Update User Page.....	122
Create Category.....	123
Create Subcategory.....	124
Add User.....	125
<b>Chapter 06.....</b>	<b>128</b>
• Limitations for the project.....	128
• Lesson Learned.....	129
• Future recommendation.....	130
<b>References.....</b>	<b>131</b>

## Table of figures

Figure 1 ER Diagram .....	31
Figure 2 Context Diagram .....	35
Figure 3 Level 1 Admin DFD .....	35
Figure 4 Level 1 User DFD .....	36
Figure 5 level 2 User DFD .....	37
Figure 6 Level 2 User DFD .....	37
Figure 7 Home Page .....	42
Figure 8 Product Page .....	43
Figure 9 Vehicle Details Page .....	44
Figure 10 Computer Details Page.....	45
Figure 11 Product Payment Page.....	46
Figure 12 Post Ads.....	47
Figure 13 Admin Home Page.....	48
Figure 14 Admin Sub Category page.....	48
Figure 15 Admin User Page .....	49
Figure 16 Update Category Page .....	49
Figure 17 Create Category .....	51
Figure 18 Create Sub Category .....	51
Figure 19 Add User.....	52
Figure 20 Available Stock manage .....	53
Figure 21 Payment Page .....	53
Figure 22 Login Page .....	54
Figure 23 Sign Up Page.....	54
Figure 24 HTML - Login Code .....	55
Figure 25 PHP-Login Code.....	56
Figure 26 HTML-Register Code .....	57
Figure 27PHP-Register Code .....	58
Figure 28 Database Code .....	59
Figure 29 HTML-Post Ad Code .....	60
Figure 30 PHP-Post Ad Code .....	61
Figure 31 HTML- create category.....	62
Figure 32 PHP-Create Category.....	63
Figure 33 PHP-Update Category .....	63
Figure 34 HTML-Update Category .....	64
Figure 35 PHP-Delete Category.....	65
Figure 36 SDLC MODEL .....	66

## **Table of tables**

Table 1. Economic Feasibility Report .....	24
Table 2. Work Breakdown Structure .....	28
Table 3. Risk Assessment .....	30

## CHAPTER 01

- **Introduction.**

At present, our country is in a very critical situation. As a result, not all people have an adequate source of income at present and the people have to face many oppressive situations due to the inadequate source of income. And nowadays many people use various websites to sell their vehicles, appliances, etc. at affordable prices using the internet. One of the things we noticed is that most websites charge a lot of money to publish an ad. Another factor we noted was the need to complete a number of steps to publish such an ad.

Then our team decided to build a website to easily place ads and make it easy to view those ads as well as facilitate the purchase of certain items. We also heard that a friend of ours was currently running a product marketing business using social media and that he was hoping to improve that business and create a website on the internet. So, we talked to him and decided to set up an e-commerce website for him to use his ideas and the ideas we have to carry on his business.

In this document, we discuss the external functions used to make this website easy to use for users after building it, and the specific functionalities we hope to use.

- **Problem statement.**

Our client currently uses a variety of methods, such as social media, to carry out his business activities as an intermediary, posting ads to him daily from different regions, different individuals and different organizations. As he was doing business, Covid 19 spread all over the world and due to the economic impact in Sri Lanka, a large number of people joined him to sell their various products. But he now wants to sell goods and services using the internet rather than using social media to sell goods.

Another problem that our client faces is that the advertisers do not give proper information about certain advertisements before they are published. Due to this, our client had to face many difficulties on many occasions. He also used social media to advertise, but due to the fact that the advertisements were not published properly and were not interesting, the number of customers viewing the advertisements had gradually decreased. Some advertisers do not give out their contact numbers, so after posting the ad, viewers call our client's phone and ask for details, which is another problem that our client may face.

In some cases, we receive a large number of ads per day on their social media account, but our client is not able to manage all of them properly. One of the biggest problems our clients face right now is that he has no time left for any other external activities as he has to spend a lot of time choosing all those ads and their descriptions properly.

- **Literature Review.**

Today, there are a number of websites that are affiliated with eCommerce, offering their services on products and services. Each of these sites features unique features, functionality, and unique features. Websites like Alibaba.com, lkman.lk, Patpat.lk can be identified as the most popular websites that we can identify.

If you take the lkman.lk website, the various products and services on the site are designed in such a way that users can access and use those categories in separate categories using different categories. A unique feature of the site is that visitors are able to find the product they want using the search bar. The price, images, details of the product, etc. that should be mentioned in relation to any such advertisement are also clearly stated in this website. These websites also allow customers to connect directly with the person who advertised the service. In other words, all of these sites have the same unique characteristics. When posting ads on these websites, you can easily enter the details of the relevant ad and post it. But on some websites, you have to pay a certain amount to post an advertisement.

And all of these websites use databases to keep users' data and ad data secure.

Another factor that can be found on these sites is that anyone who visits the site can also search for details about items currently for sale around his or her area. The company also offers the convenience of home delivery of the products they currently have for sale on their website when a customer needs them. Various such technologies have been used in these sites.

- **Proposed solution.**

What our client wants is an easy way to deliver goods and services using the Internet. For this we suggest that he should build an e-coms website. That is, the website we propose allows any person to visit the site and easily display their belongings, services, vehicles, land and clothing on the site. This website also enables our client to take care of all their needs. Another thing is that nowadays many people use the internet and a large number of users come daily to receive the advertisements and services included in this website as well as to buy goods.

We also recommend this website which provides solutions to all the problems that our client is currently facing. This will enable him to save a lot of our client's time and effort by using the website we suggest.

- **Justification of the Solution.**

#### Allinone.lk – Web site For Buying & Selling

We also proposed the name “**AllinOne.LK**”, which advertises a wide range of products through this e-commerce website that we propose to market our products and services, and our client also agreed to this.

After explaining our client's needs and concerns, we suggest that this site minimize his problems. This system is designed not only to advertise the products for sale but also to enable the people who visit the website to get the product they want. We hope that this website we present will help the server avoid the following problems.

- Save time - Anyone who visits this site can easily register on the site and post their ad. As a result, the time it takes for us client to post an ad is greatly reduced.
- Labor savings - Currently, our client had to put in a lot of effort when posting an ad and posting ads looking for information about that ad. This is because we call our server to find out information about certain ads. But the website we create has given us the ability to easily post ads without the intervention of any person. So, he is able to save his labor as our client expects.
- Ability to store ad details - Because we use a database here, he is able to store a large amount of information in the database and maintain the website, which is also convenient for our clients.
- Was able to group ads - Our client told us that customers are less likely to find ads because all the ads are cohesive when posted on social media. But we have created a website that allows you to post ads in many categories.

Example:  
A separate category for auto ads  
A separate category for clothing ads  
A separate category for land advertisements  
A separate category for animal advertising  
A separate category for new job vacancies

- Information provided for posting the ad –  
Another problem that our client is facing is that it does not provide true information about the ad when it is published, but this website that we create must publish the true information about the ad in order to post an ad, so our client is currently facing Problematic situations can be minimized

- **Functionalities & Non functionalities of the system.**

## **1. Functional Requirements.**

### **I. Register.**

The site we are creating has an administrator already registered in the system. Also, only an administrator currently on the system is eligible to register another administrator. A user who comes to the site to receive a service or product must register on the site. This can be done by clicking on the Register tab at the top of the website, in the navigation bar, after filling in the form correctly and successfully registering on the website.

Input: Valid User Registration Details.

Process: Checking entered data Using Database Information & Save all details in Database.

Output: Successfully Registered As user or Admin & Display Notification "User Has Registered Successfully".

### **II. Login.**

Administrators and users who are currently registered on this site that we are creating will be able to successfully access their accounts once they have entered the username and password they provided when registering on the site. But if you enter the wrong username or password, the system will indicate that it is incorrect.

Admin = Admin email And Admin Password.

User = User Email and User Password.

### ***III. Admin User Level Details.***

#### **1. Update User details.**

To change the information provided during the registration of an administrator or user who is currently registered on the Website, be sure to log in to the admin port system and click the Update button to update the information.

**Input** : Click Update Button

**Process:** Get All user details to registration Form.

**Output** : Update User Details & Display "Successfully Updated"

#### **2. Delete Users.**

An administrator has the ability to control an administrator or user who is currently registered on the site. Administrators can log in to their account and successfully delete a user by clicking the Delete User button.

**Input** : Click Delete User Button

**Process:** Get All user details from database

**Output** : Delete User & Display "Successfully Deleted"

#### **3. View ad.**

Anyone who visits the site will be able to see the ads currently on the site. When a particular ad is clicked, a separate page is loaded on the website. The full description of the ad can be found on the website.

**Input** : Click ad.

**Process:** Get a database of information about relevant ads

**Output** : Display ad full details.

#### **4. Post ads.**

All members who are currently registered on the Website will be able to display their ad on the Website. To do this, go to the add new post tab and enter the relevant information and related illustrations so that the ad can be successfully displayed on the website.

**Input** : Click Add new Item Button

**Process:** Get Insert ad details & Photos after those Details in database

**Output** : Post Ad & Display "Successfully Posted Your Ad" message.

#### **5. Update Ad Details.**

If an ad published by a user wants to modify the information provided, he can update the information by clicking on the Update add details button.

**Input** : Click Update Post Button

**Process:** Get a database of information about relevant ads

**Output** : Display "Post Successfully Updated" message

#### **6. View Customers Feedbacks & Messages.**

If any person or customer who visits the Website wants to know more about any service after the service has been provided, they can view the Message Administrator who sent us the message and reply to those messages.

**Input** : Click see Customers Feedback & Massages Tab.

**Process:** Get a database of information about relevant feedbacks & messages.

**Output** : Display all Customer feedbacks & messages.

## **7. Delete Posted ads.**

To remove an ad posted on the Website by any user, the user must log in to the Website from the User Account or the Admin Account and then successfully remove the relevant ad from the Website.

**Input** : Click Delete ad Button.

**Process:** Get a database of information about relevant ad.

**Output** : delete ad & Display "Post Successfully Deleted" Message.

## ***IV. User Level Details.***

### **1. View ad.**

User can visit the site will be able to see the ads currently on the site. When a particular ad is clicked, a separate page is loaded on the website. The full description of the ad can be found on the website.

**Input** : Click ad.

**Process:** Get a database of information about relevant ads

**Output** : Display ad full details.

### **2. Post ads.**

User currently registered on the Website will be able to display their ad on the Website. To do this, go to the add new post tab and enter the relevant information and related illustrations so that the ad can be successfully displayed on the website.

**Input** : Click Add new Item Button.

**Process:** Get Insert ad details & Photos after those Details in database.

**Output** : Post Ad & Display "Successfully Posted Your Ad" message.

### **3. Update Ad Details.**

If an ad published by a user wants to modify the information provided, he can update the information by clicking on the Update add details button.

**Input** : Click Update Post Button.

**Process:** Get a database of information about relevant ads.

**Output** : Display "Post Successfully Updated" message.

### **4. Delete Posted ads.**

To remove an ad posted on the Website by any user, the user must log in to the Website from the User Account or the Admin Account and then successfully remove the relevant ad from the Website.

**Input** : Click Delete ad Button.

**Process:** Get a database of information about relevant ad.

**Output** : delete ad & Display "Post Successfully Deleted" Message.

### **5. Direct chat with Admins.**

If any customer or user visiting the site wants to know something about the ad, the chat option can send the information to the person who posted the ad or to the right admin on the site.

**Input** : Click Chat Button.

**Process:** Inserting all messages to the database.

**Output** : Message Send.

## **6. Give the feedbacks.**

If a person who came to the website and obtained a service or product wants to share his experience with others, the give a feedback tab allows them to share their present experience so that others can see it.

**Input** : Click Give Your Feedback Button.

**Process:** Inserting all feedbacks to the database.

**Output** : Display "Thank You For the rate Us" message.

## **v. Logout.**

Once a person has logged in to the website by entering his username and password, he can successfully log out of the user account by pressing the logout button above to successfully log out from the website.

**Input** : Click Logout Button.

**Process:** Load Session Destroy page.

**Output** : Successfully Logout to home page.

## 2. Non-Functional Requirements.

Non-Functional Requirements of a system are attributes such as time limits, system security and currently used security limits.

### I. ***Security:***

The system that we create contains the user's personal data and some confidential information, so the database must be protected so that no other outsider can access that information.

### II. ***Performance:***

This function is primarily defined as how quickly a person responds after an activity has taken place.

### III. ***Usability:***

Utilities can take care of making this site easy for anyone to understand and work with this system.

### IV. ***Accuracy:***

All data entered into the Website must be accurate at all times.

### V. ***Availability:***

Any user should be allowed to visit the site at any time.

### VI. ***Platform Independents:***

This means that the website we create must be successfully implemented on any operating system. You must also be able to access this site using any web browser.

- **Software Development methodology.**

The methodology used to create this website that we are creating is the waterfall model methodology. The reason we use this waterfall model is that in the process of developing the website, the whole process of developing the system here is divided into several stages. This will make the development of the website easier and more accurate. Another reason for using this waterfall model is that once one stage is completed, the result of that stage is systematically used as the input of the next stage. For this reason, if there is a problem at any stage, it can be easily identified. For all these reasons, this waterfall model was used to complete the web.

Below are the programming languages used to develop this website.

- HTML.
- PHP.
- CSS.
- JavaScript.

The DBMS we used to create this site are:

- My SQL.

The software we used to create the website are:

- Adobe Dreamweaver.
- VS Code.
- Wamp Server.
- Database MySQL.
- Web browser (Google Chrome).
- Adobe XD.

Other Development Tools:

- Bootstrap Version 04.
- jQuery.

## CHAPTER 02.

### Feasibility study report.

Before developing a system, it is important to accurately study the impact on the organization or organization that provides the system. Thus, the potential impact of the feasibility study on the system can be positive or negative. This feasibility study can be done in two ways, namely technical feasibility and economic feasibility.

#### 1. Technical Feasibility:

Here we assessed whether the task could be done technically while creating the website. When the client was assigned to create the website, a definite plan was drawn up for each step of the operation, and preparations were made to create the website. Furthermore, the use of this technological capability provided an accurate understanding of technology, liquidity, labor, and supply processes.

Using this technology feasibility study, it was important for us to have a clear understanding of the full functionality of the website and whether we have the technical resources needed to transform the system being created into an effective and cost-effective website. This also helps to resolve any issues that may arise during the development of the site.

Focusing on the technical background of the system we are creating, the minimum requirements for implementing this system are,

- 19
  - Core i5
  - 8GB Ram
  - 1TB Hard Drive
  - Windows 10 +

- ***Resources Available.***

- ✓ Software S
- ✓ A Team
- ✓ Back End Developers
- ✓ Front End Developers
- ✓ Database Engineers
- ✓ System Analysis
- ✓ Computers.

- ***Technology.***

- ✓ Bootstrap
- ✓ jQuery
- ✓ HTML
- ✓ CSS
- ✓ JavaScript
- ✓ MySQL

## 2. Economic Feasibility:

The cost is minimal due to the use of very simple architecture with minimal requirements to facilitate the use and maintenance of a web-based application. Also, having an automated system is not helpful for all existing businesses and institutions. Therefore, after conducting cost analyzes, an analysis should be done on whether the website created is useful or not.

This method does not require a large investment of capital at the outset, so the client can reap large returns. We can also increase the financial status of the client as he can make huge profits through this website that we create. It also adds a large customer base to the site as it receives a large number of users daily. It can also increase efficiency and user satisfaction. Because of all these functions, the profit can also be greatly increased when the users get a large number of products through the website.

Table 1 2.Economic Feasibility Report

Description	Price (LKR)
MySQL	Free
Visual Studio Code	Free
Data Collection	5000.00
Hardware (Ex: PC, Router, etc.)	190000.00
Development	10000.00
Maintenance	15000.00
Hosting	10714.00
Antivirus Software	4350.00
Other	5000.00
<b>TOTAL</b>	<b><u>240064.00</u></b>

- 12  
• Cost benefit analyses.

- ✓ An Assessment of The Economic Justification for The Computer-Based System Project
- ✓ Cost-Benefit Analysis Help to Give Management A Picture of The Costs and Benefits
- ✓ It Begins by Identifying Various Costs and Benefits Elements of The System

### **3. Operational feasibility:**

Here we clearly describe the extent to which the proposed website provides solutions to client problems and how the identified opportunities can be utilized for website development and to what extent the requirements identified during the analytics phase of website development can be successfully met. This makes it possible to gain an understanding. This phase also focuses on the time it takes to develop a website, the time it takes to develop a website, and the future maintenance of the site. Here you can get a clear idea of the success of the website and the extent to which the intended plans will be successfully implemented.

The Essential Question that helps in testing the operational feasibility of this web application.

1. Does the management support the project?

- Our client assisted us in various ways while we were working on creating the website. As a result, we were able to carry out the task successfully while creating the website.

2. Are the users happy with the current business practice? if not they will welcome the change to the new system.

- At present, our client cannot be satisfied with the methodology used to sell the goods. Accordingly, once we have created the website, our client will be able to clearly identify that there is a significant growth in sales as we continue to use it to market our products.

### **3. Legal feasibility:**

We were working on creating this site and we made sure that all of this work was done in a legal manner. That is, the website is built using legally recognized software. We have also taken steps to protect the user's data by complying with the Data Protection Act, as it's the user's data stored in the database when it is released to the Internet.

## **Requirement gathering.**

Adding these requirements is the process of identifying what can be taken to make our project a success and what needs to be created to make it happen. Everyone has different assumptions about what to include in the website we create. Before we start creating a website, we need to gather requirements from project partners to fully understand how this site should be implemented. Techniques are defined in terms of how the activity is completed, taking into account the given conditions. There may be one or more approaches related to a task. At least one technique must be used to successfully complete at least one task.

The methods we use to add requirements are,

- Team Meetings.
- Interviews
- Zoom Meetings.
- Observation.

Our client wanted to add some new functionality to the website we are creating. This is to make it easier for him to continue the website at his discretion. Accordingly, the latest Requirement submitted by him is,

- At the end of the month, he wanted to find out how many items were being sold through the website.
- Availability of data on the number of users who visited the site at the end of the month.
- Once a customer has purchased a product, the description of the item and the price and quantity of the item should be arranged so that he can be notified by email.

## **Resource identification.**

In carrying out this project, we have identified the following resources in order to identify the resources needed to carry out this project and to use those resources to successfully carry out the project.

- ✓ Software S
- ✓ A Team
- ✓ Back End Developers
- ✓ Front End Developers
- ✓ Database Engineers
- ✓ System Analysis
- ✓ Computers.

22

## Work breakdown structure.

Table 2. Work Breakdown Structure

WBS #	Task / Description	Duration	Dependency
1	Development E-Commerce Web Application for Product buying and selling	48 DAYS	
1.1	Analyzing	28 days	
1.1.1	Team Building	1 day	
1.1.2	Arrange a Team meeting	1 day	1.1.1
1.1.3	Team Introduction and Sharing experiences	1 day	1.1.2
1.1.4	Obtain Approval from Supervisors.	1 day	
1.1.5	Studying Client Details and Other E - Commerce Website.	2 days	1.1.4
1.1.6	Prepare a System Report and Present to the other team members.	1 day	1.1.4 , 1.1.5
1.1.7	Requirement Gathering	2 days	1.1.5
1.1.8	Prepare questionaries for customer and client.	2 days	1.1.4
1.1.9	Resource Allocation	3 days	1.1.2,1.1.5
1.1.10	Resource Buying	1 day	1.1.9
1.2	System Designing	15 DAYS	
1.2.1	Design ER Diagram.	2 days	1.1.8
1.2.2	Create normalization relational schemas.	2 days	1.2.1
1.2.3	Design use case diagram.	2 days	1.2.2
1.2.4	Design wireframe diagram.	2 days	1.2.3
1.2.5	Designing user interfaces.	7 days	1.2.4
1.2.6	Create databases.	1 day	1.2.2
1.3	Coding Web Application.	13 DAYS	
1.3.1	Frontend.	4 days	1.2.6
1.3.2	Backend	4 days	1.3.1
1.3.3	Web application testing.	3 days	1.3.1
1.3.4	Bug fixing.	2 days	1.3.2
1.4	System Finalization.	5 DAYS	
1.4.1	Create user manual.	3 days	1.3.3
1.4.2	Client training.	2 days	1.3.2 , 1.3.3
1.4.3	Release the system.	1 days	1.4.1
1.4.4	Project End	1 day	1.4.3

## **Risk Analysis.**

Some of the major categories of potential risks in this web application management system that we create are,

### **1. Technical Risk:**

As the site was being created, there were not enough technical resources and hardware as well as software to do that. In some cases, software crashes while the site is being created, as well as interruptions caused by malicious programs.

### **2. Project Management Risk:**

Expected times and dates were lower than expected before the site was created. As the duration of this project is longer, if the time allotted for one phase is extended, it can be a risk to the entire project.

### **3. Organizational Risk:**

The cost of software, hardware, and other necessary tools and items can vary and can be more expensive than the budget plan.

### **4. External Risk:**

The new government rules and regulations may affect the website and may affect the requirements of the system.

## Risk Assessment.

Table 3.Risk Assessment

Risk	Impact Scale	Mitigation Method
1 Loss of data Due to natural disasters or Electricity failures.	Very High	Keeping an external backup in a Different place and constantly taking backups.
1 Cost can be higher than the expected cost	High	Do a proper cost feasibility before starting the project.
Expected schedule can be change	Medium	Remove unnecessary function and make the scope smaller by keeping only the most important function.
1 Hardware malfunctioning	Medium	Use proper hardware and software resources that are working properly.
Malware Attacks	Medium	Use a proper virus guard and keep it updated.

1

## Chapter 03.

- Design of the system.

- Database.

- ✓ Entity relationship diagram (ER Diagram)

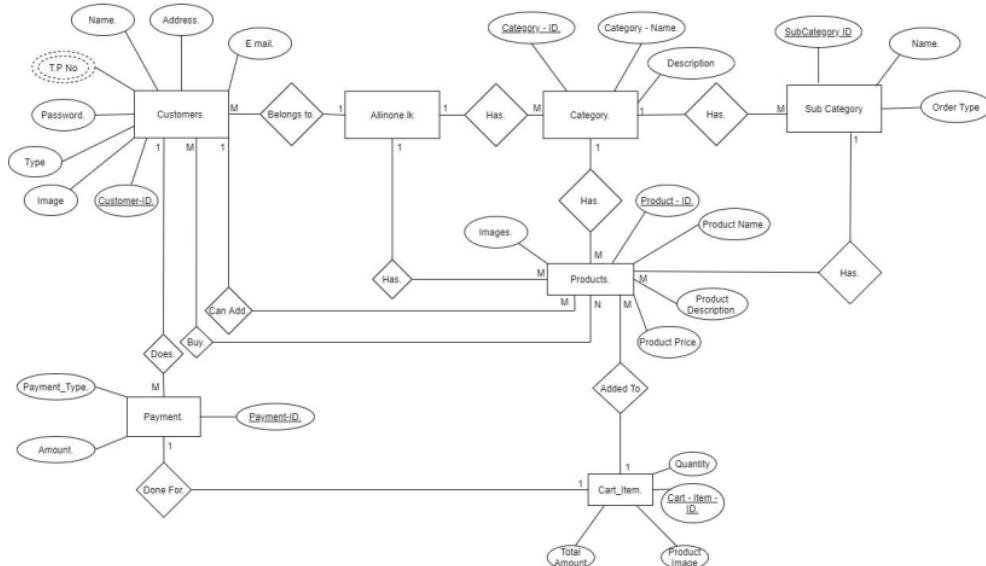


Figure 1 ER Diagram

✓ **Normalized relation schema.**

Step 01 → Map → Strong Entity

Customer (Customer ID, Name, Address, Email, Password, T.P No, Type, Image)

7 Category (Category ID, Category Name, Description)

Subcategory (Subcategory ID, Name, Order Type)

Product (Product ID, Product Name, Product Description, Product Price, Image)

Cart Item (Cart ID, Quantity, Product Image, Total Amount)

Payment (Payment ID, Payment Type, Amount)

Step 02 → Mapped One to One Relationship

Payment - Payment ID

Cart - Cart ID

Payment (Payment ID, Payment Type, Amount)

Cart Item (Cart ID, Quantity, Product Image, Total Amount, Payment ID)

### Step 03 → Mapped to One-to-Many Relationship

Product - Product ID 7

Category – Category ID

Subcategory – Subcategory ID

Category (Category ID, Category Name, Description)

Subcategory (Subcategory ID, Name, Order Type, Category ID)

---

Product (Product ID, Product Name, Product Description, Product Price, Image, Subcategory ID) 5

Subcategory (Subcategory ID, Name, Order Type, Category ID) 4

---

Category (Category ID, Category Name, Description)

Product (Product ID, Product Name, Product Description, Product Price, Image, Subcategory ID, Category ID) 4

---

Product (Product ID, Product Name, Product Description, Product Price, Image, Subcategory ID, Category ID)

Cart Item (Cart ID, Quantity, Product Image, Total Amount, Payment ID)

---

Customer (Customer ID, Name, Address, Email, Password, T.P No, Type, Image)

Payment (Payment ID, Payment Type, Amount, Customer ID)

---

Customer (Customer ID, Name, Address, Email, Password, T.P No, Type, Image)

5  
Product (Product ID, Product Name, Product Description, Product Price, Image, Subcategory ID,  
Category ID, Customer ID)

Step 04 → Mapped Many to many Relationships with Descriptive Attributes

Customer (Customer ID, Name, Address, Email, Password, T.P No, Type, Image)

5  
Product (Product ID, Product Name, Product Description, Product Price, Image, Subcategory ID,  
Category ID, Customer ID)

Customer Buy Product (Customer ID, Product ID, T.P No)

- System Design.

  - DFD.

    - ✓ *Context diagram.*

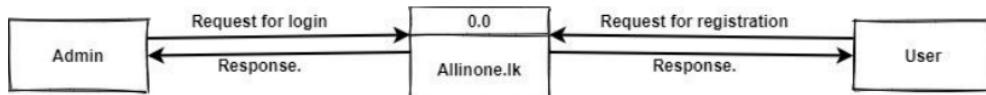
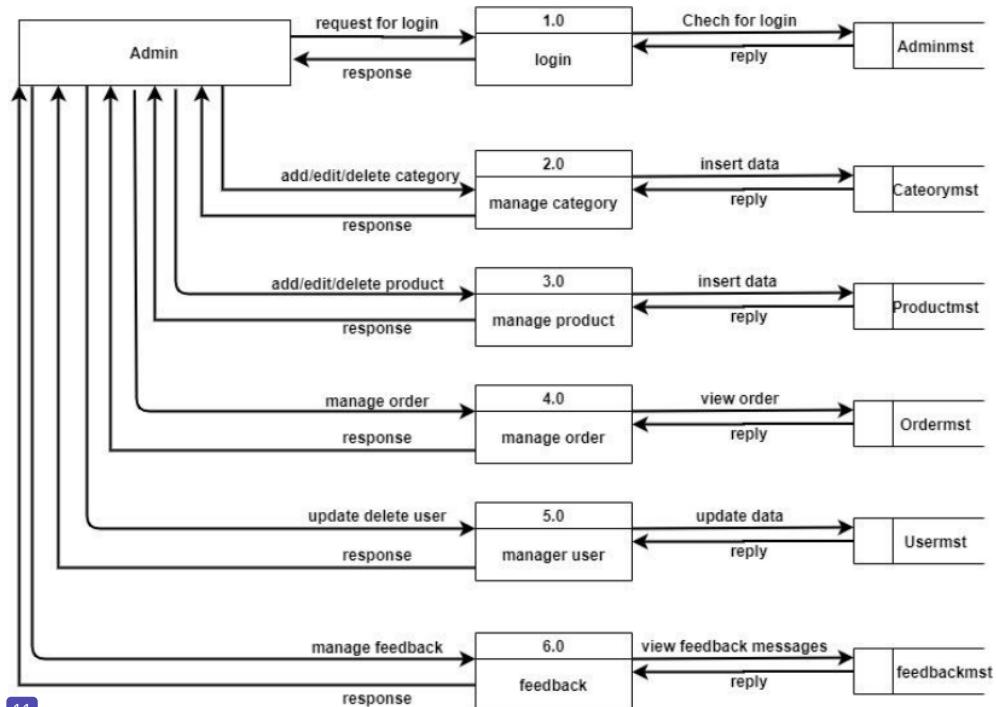


Figure 2 Context Diagram

    - ✓ *Level 1 Admin DFD.*



11

Figure 3 Level 1 Admin DFD

✓ **Level 1 User DFD.**

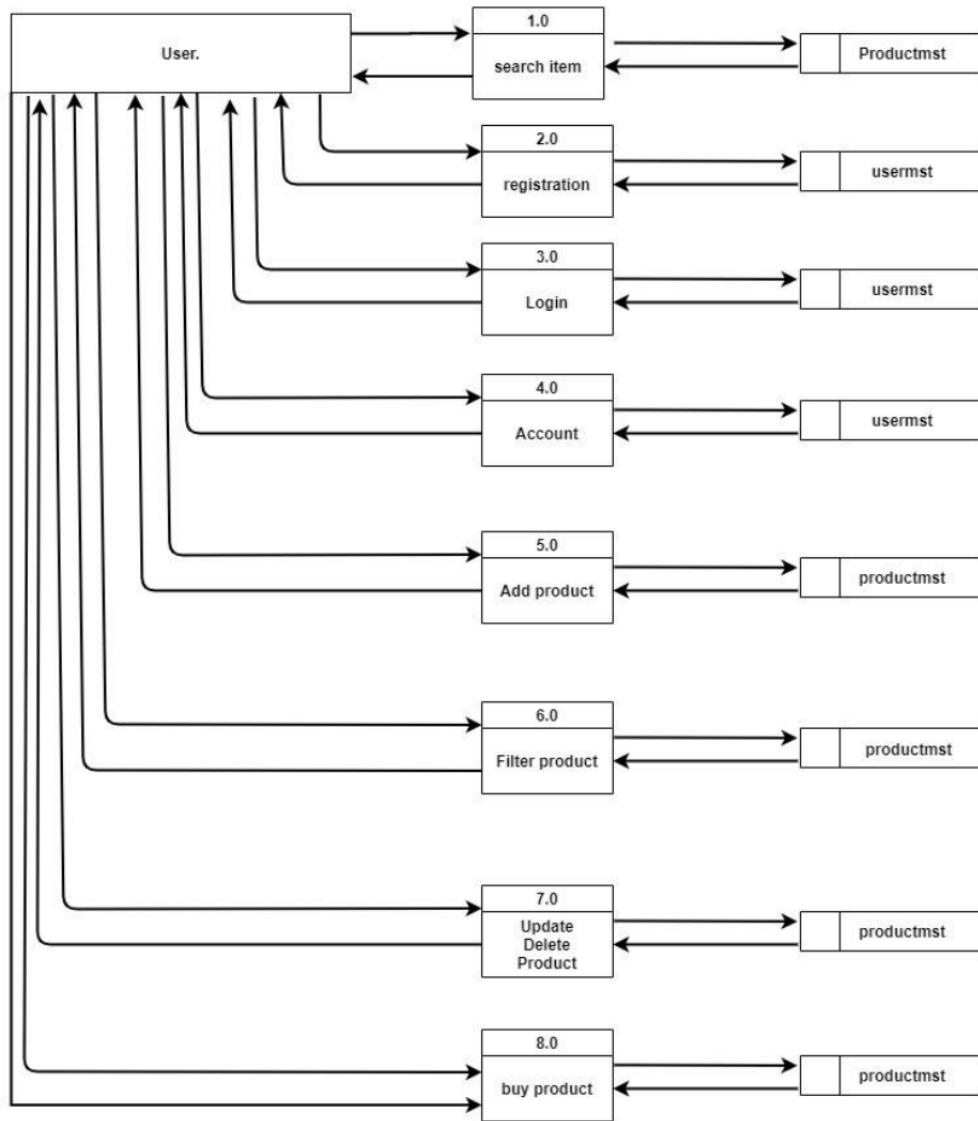


Figure 4 Level 1 User DFD.

✓ **Level 2 User DFD.**

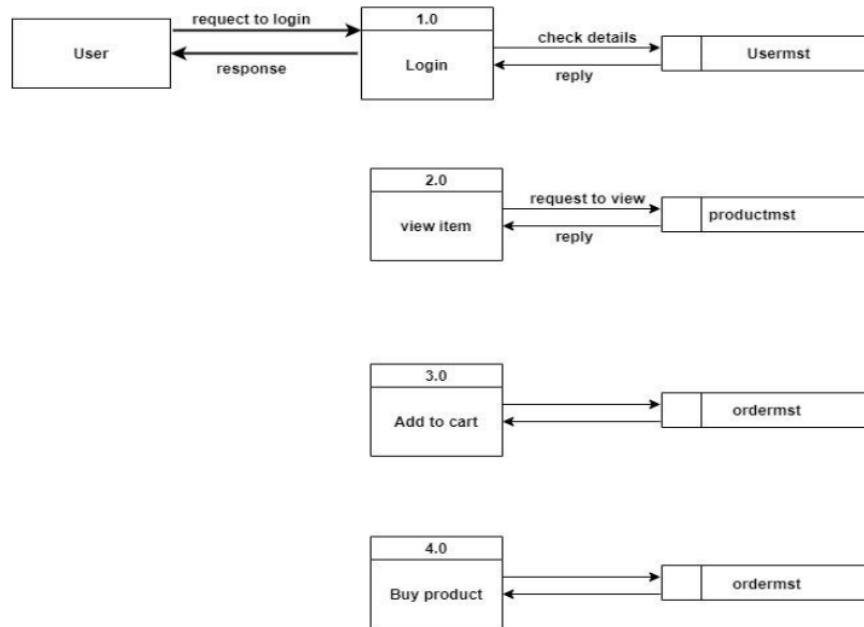


Figure 5 level 2 User DFD

✓ **Level 2 User DFD.**

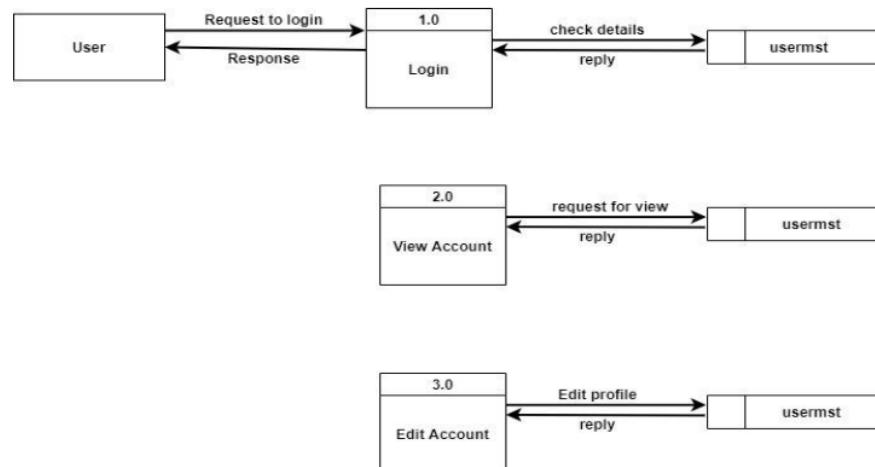


Figure 6 Level 2 User DFD.

- **Pseudo code.**

### 1. Login.

```
Begin  
GetLoginID  
Get Password  
If (LoginID == Entered Username && Password)  
    Login Successfully  
Else  
    Login Failed  
Endif  
End.
```

### 2. Add Categories.

```
Begin  
Text Category_ID, Category_Name  
Insert Category_ID, Category_Name  
Print = "Category_ID" + CID  
Print = "Category_Name" + CName  
End.
```

### 3. Add Products.

```
Begin  
Text Product_ID, Product_Name, Product_Quantity, Product_Price  
Insert Product_ID, Product_Name, Product_Quantity, Product_Price  
Print = "Product_ID" + PID  
Print = "Product_Name" + PName  
Print = "Product_Quantity" + PQuantity  
Print = "Product_Price" + PPrice  
End.
```

#### **4. Update Categories.**

```
Begin  
Text (Category Details)  
Update (Category Details)  
Get (Things to Be Update)  
Then  
Print ("Successfully Updated")  
Else  
Print ("Fail to Updated")  
End.
```

#### **5. Update Products.**

```
Begin  
Text (Product Details)  
Update (Product Details)  
Get (Things to Be Update)  
Then  
Print ("Successfully Updated")  
Else  
Print ("Fail to Updated")  
End.
```

#### **6. Delete Categories.**

```
Begin  
Text (Category Details)  
Update (Category Details)  
Get (Things to Be Delete)  
Then  
Print ("Successfully Deleted")  
Else  
Print ("Fail to Deleted")  
End.
```

## **7. Delete Products.**

```
Begin  
Text (Product Details)  
Update (Product Details)  
Get (Things to Be Delete)  
Then  
Print ("Successfully Deleted")  
Else  
Print ("Fail to Deleted")  
End.
```

## **8. Add To Card.**

```
Begin  
Text (Product Details + Category details + User Details)  
Get (product+Category Details)  
Then  
Print ("Successfully Adding Cart")  
Else  
Print ("Fail to Adding Cart")  
End.
```

## **9. Buy Product.**

```
Begin  
Buy Product  
Get Category Name, Product Name, Quantity  
Set total value = Price + Quantity  
Print Total Value, Product details  
End.
```

## 10. Search.

```
Begin
9
l=1

Found=false

While l <=n && found == false do

Being

If x== a [ l] then

Found true

Else l =i+1

End.
```

(l = Product)

- User Interfaces / Navigations.

## 1. User Interfaces.

### 1. Home Page

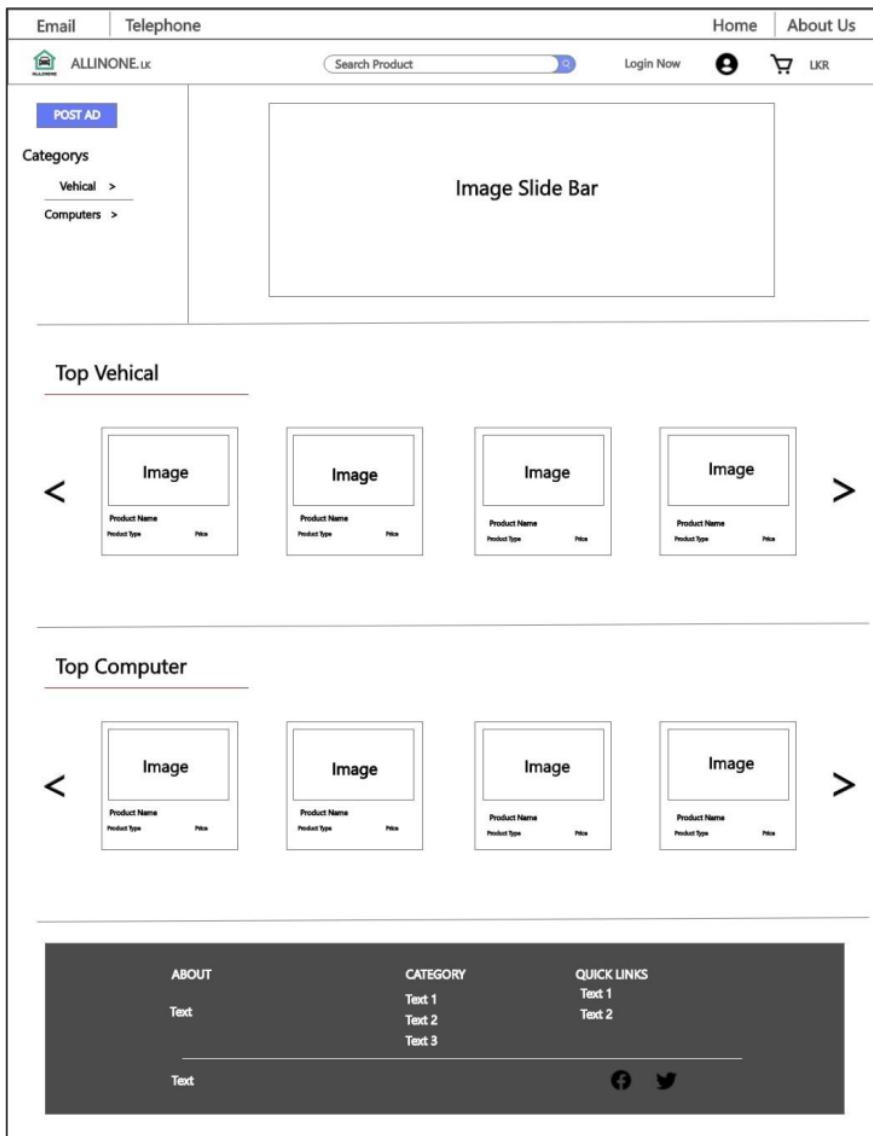


Figure 7 Home Page

**2. Product Page.**

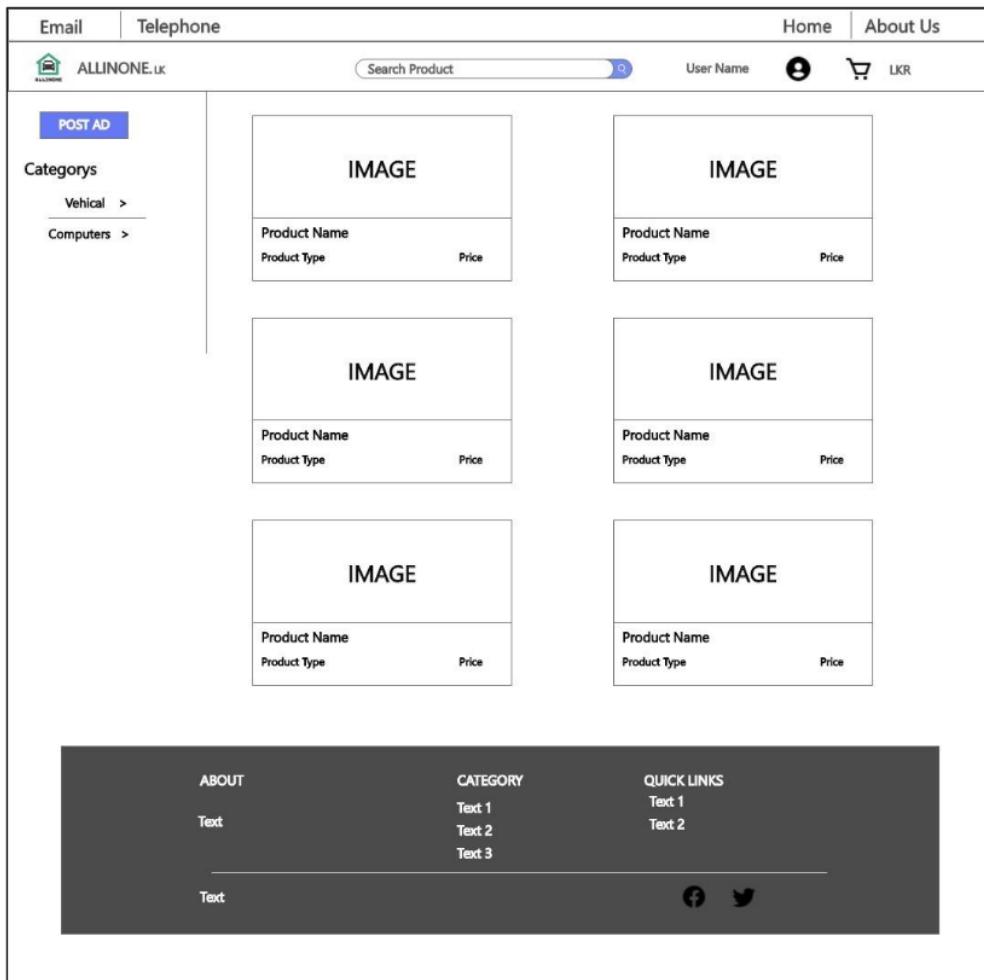


Figure 8 Product Page

**3. Vehicle Details Page.**

The wireframe illustrates the layout of the Vehicle Details Page. At the top, there is a header bar with links for Email, Telephone, Home, and About Us. Below the header is a navigation bar featuring the logo 'ALLINONE.LK' (with a house icon), a search bar labeled 'Search Product', and user account information for 'User Name' and a shopping cart icon labeled 'LKR'. The main content area contains a large placeholder box labeled 'IMAGE' and two columns of product information: 'Product Name' and 'Price'. Under 'Price', there are sub-labels for Availability, Condition, and Telephone Number. At the bottom, a dark footer bar includes sections for 'ABOUT' (with 'Text'), 'CATEGORY' (with 'Text 1', 'Text 2', and 'Text 3'), and 'QUICK LINKS' (with 'Text 1' and 'Text 2'). It also features social media icons for Facebook and Twitter, and a 'Text' placeholder at the bottom.

Figure 9 Vehicle Details Page

**4. Computer Details Page.**

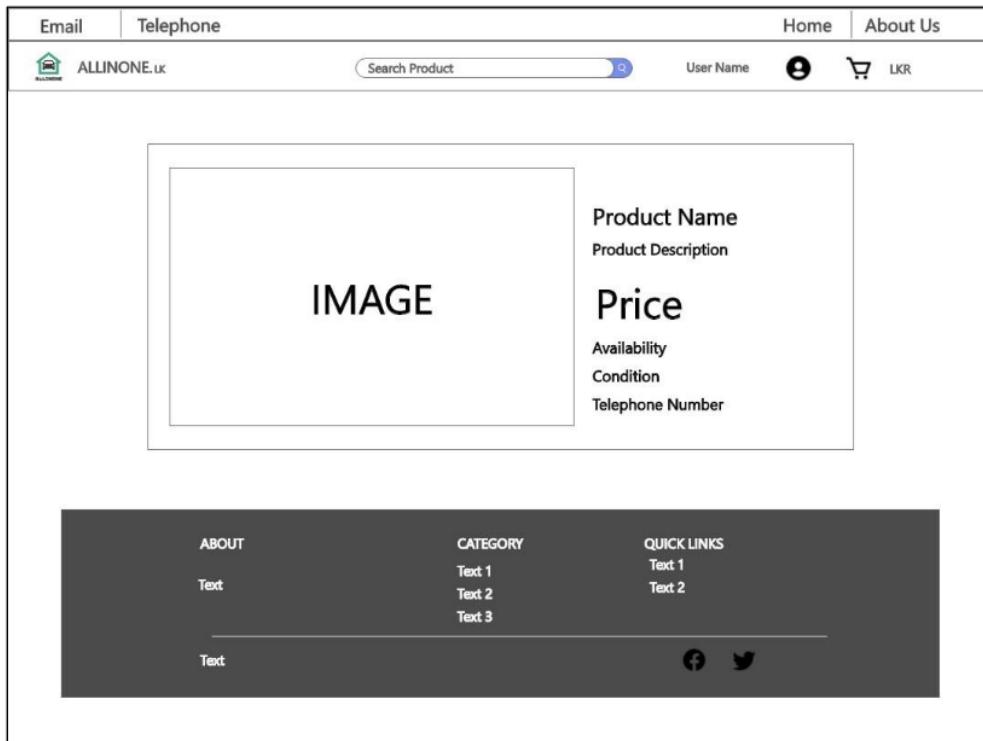


Figure 10 Computer Details Page

**5. Product Payment Page**

The image shows a user interface for a product payment page. On the left, there is a rounded rectangular section labeled "Product". It contains a placeholder "Image" box, followed by "Product Name" and "SubCategory Name" fields, "Quantity" and "Price" fields, and a "Back to Home" button with a left arrow icon. On the right, there is a grey rectangular section labeled "CARD PAYMENT". It contains fields for "Card Holder Name", "Card Number", "MM" (Month), "YYYY" (Year), and "CVV". Below these fields is a "Total Price:" label and a large blue "PAY NOW" button.

Figure 11 Product Payment Page

**6. Post Ads.**

Ads

POST AD

Select Category ▲▼

Select SubCategory ▲▼

Select Type ▲▼

Product Name

Telephone Number

Description

Quantity

Price

Choose File Picture Name

**POST AD**

The form consists of several input fields and dropdown menus. At the top, there are three dropdown menus labeled 'Select Category', 'Select SubCategory', and 'Select Type', each with up and down arrow icons. Below these are four text input fields: 'Product Name', 'Telephone Number', 'Description', and 'Quantity'. Following these are two more text input fields: 'Price' and 'Choose File Picture Name'. A large blue rectangular button at the bottom is labeled 'POST AD' in white capital letters.

Figure 12 Post Ads

6  
7. ***Admin Home Page***

The screenshot shows the Admin Home Page. On the left, there is a sidebar with a logo and a navigation menu containing links for Home, Category, SubCategory, Order, and User. The main content area is titled "Category" and contains a "Create Category" button. Below it is a table with columns: ID, Name, Description, and Button. Two rows of data are shown, each with an "ID" column, a "Category Name" column, a "Category Description" column, and two buttons: "EDIT" and "DELETE".

ID	Name	Description	Button
ID	Category Name	Category Description	<b>EDIT</b> <b>DELETE</b>
ID	Category Name	Category Description	<b>EDIT</b> <b>DELETE</b>

Figure 13 Admin Home Page

8. ***Admin Sub Category page***

The screenshot shows the Admin Sub Category page. On the left, there is a sidebar with a logo and a navigation menu containing links for Home, Category, SubCategory, Order, and User. The main content area is titled "Sub Category" and contains an "Add Sub Category" button. Below it is a table with columns: ID, Name, Category Name, Order type, and Button. Two rows of data are shown, each with an "ID" column, a "Sub Category Name" column, a "Category Name" column, an "Order type" column (containing "Details"), and two buttons: "EDIT" and "DELETE".

ID	Name	Category Name	Order type	Button
ID	Sub Category Name	Category Name	Details	<b>EDIT</b> <b>DELETE</b>
ID	Sub Category Name	Category Name	Details	<b>EDIT</b> <b>DELETE</b>

Figure 14 Admin Sub Category page

### **9. Admin User page**

ID	User Information	Type	Telephone Number	Button
ID		User Name User Email	User Type	Number <span style="background-color: #ccc; border: 1px solid black; padding: 2px;">EDIT</span> <span style="background-color: red; color: white; border: 1px solid black; padding: 2px;">DELETE</span>
ID		User Name User Email	User Type	Number <span style="background-color: #ccc; border: 1px solid black; padding: 2px;">EDIT</span> <span style="background-color: red; color: white; border: 1px solid black; padding: 2px;">DELETE</span>

Figure 15 Admin User Page

### **10. Update Category Page.**

Category

Update Category

Quantity

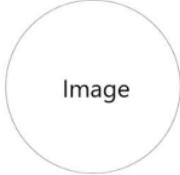
Category Name

Category Description

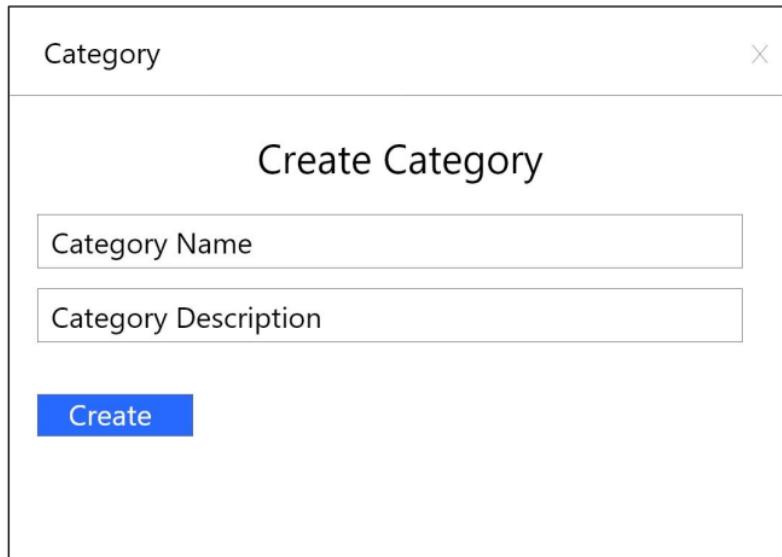
UPDATE

Figure 16 Update Category Page

**11. Update User Page**

User
<h3>Update User</h3>

Name
<input type="text" value="User Name"/>
Email
<input type="text" value="User Email"/>
Mobile Number
<input type="text" value="User Telephone Number"/>
Type
<input type="text" value="User Type"/>
Password
<input type="text" value="User Password"/>
<input type="button" value="UPDATE"/>

**12. Create Category**



The form is titled "Create Category". It contains two input fields: "Category Name" and "Category Description", both enclosed in a single horizontal input group. Below the input group is a blue "Create" button.

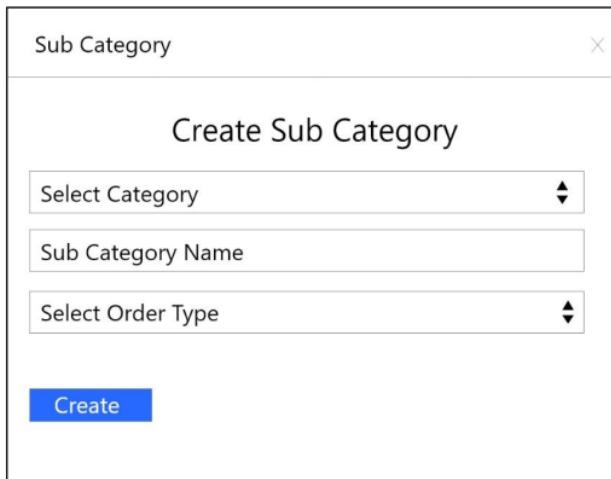
Category Name

Category Description

**Create**

Figure 17 Create Category

**13. Create Sub Category**



The form is titled "Create Sub Category". It includes three dropdown menus: "Select Category", "Sub Category Name", and "Select Order Type". Below these is a blue "Create" button.

Select Category

Sub Category Name

Select Order Type

**Create**

Figure 18 Create Sub Category

**14. Add User**

The screenshot shows a modal window titled "User" with a close button "X" in the top right corner. The main title inside the window is "Add User". Below the title are five input fields: "User Name", "Email", "Mobile Number", "Password", and "User Type" (a dropdown menu). Underneath these fields is a file upload section with "Choose File" and "Image Name" labels. At the bottom of the window is a large blue "Add User" button.

Figure 19 Add User

### **15. Available Stock Manage**

Stock				
Add Product				
Product ID	Product Information	Available	Price	Button
ID	(Image)	Product Information	Available	Price
ID	(Image)	Product Information	Available	Price

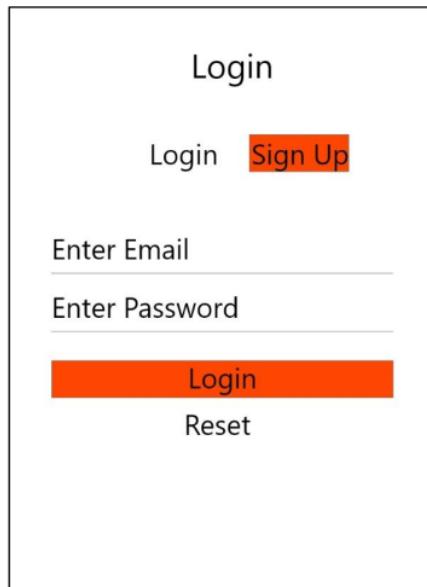
Figure 20 Available Stock manage

### **16. Payment Page**

Payment			
All Payment Total : LKR			
Payment ID	Product Name	Quantity	Total

Figure 21 Payment Page

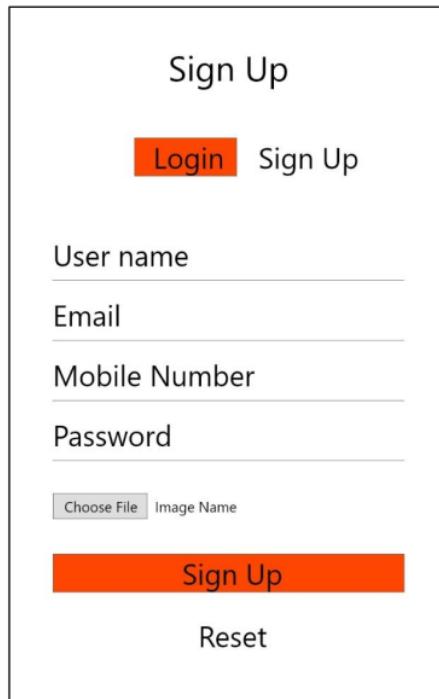
**17. Login Page**



The login page features a central title "Login". Below it are two buttons: "Login" and "Sign Up", with "Sign Up" highlighted in orange. There are two input fields: "Enter Email" and "Enter Password". Below the password field is a large orange "Login" button. To its right is a "Reset" button.

Figure 22 Login Page

**18. Register Page**



The sign up page features a central title "Sign Up". Below it are two buttons: "Login" and "Sign Up", with "Sign Up" highlighted in orange. There are four input fields: "User name", "Email", "Mobile Number", and "Password". Below the password field is a file upload section with "Choose File" and "Image Name" fields. At the bottom is a large orange "Sign Up" button. To its right is a "Reset" button.

Figure 23 Sign Up Page

## Chapter 04

- Explain co-functionalities with evidence.

### 1. Login Page

- Html code

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Login</title>
    <link rel="stylesheet" type="text/css" href="css/Login.css">

</head>

<body>
    <div class="main-box">
        <div class="tital"><center><h1>Login</h1></center></div>

        <form action="Login.php" method="post" enctype="multipart/form-data">

            <div class="box">
                <center>
                    <div class="input-box">

                        <div class="button-box">
                            <a href=""><div class="log1">Login</div></a>
                            <a href="Signup.php"><div class="sign1">Sign up</div></a>
                        </div>

                        <input type="Email" name="Email" placeholder=" Enter Email">
                        <input type="password" name="Password" placeholder=" Enter Password" >

                        <input type="submit" value="Login" name="submit" class="submitbtn">
                        <input type="reset" value="Reset" id="reset" class="resetbtn">

                    <!--</form>-->

                </div>
                </center>
            </div>
        </form>
    </div>
</body>
</html>
```

Figure 24 HTML - Login Code

The above diagram shows the html code block used to create the login page on the website. Here, the user can easily understand and log in to the website after entering his username and password. The PHP code block used for it is shown in the diagram below.

### PHP Code

```
<?php

$message="";

if(isset($_POST['submit'])) // Associate Array
{
    include("dbconnect.php");

    $Email = $_POST['Email'];
    $password = $_POST['Password'];

    if($Email == "" && $password == "")
    {
        $msg = "fill the boess";
        echo "<script type='text/javascript'>alert('$msg')</script>";
    }
    else
    {
        $sql = "SELECT * FROM users WHERE Email='".$Email' and Password='".$password'";
        $result = mysqli_query($conn, $sql);

        if(mysqli_num_rows($result) == 1 )
        {
            $row = mysqli_fetch_array($result);

            if($row["Type"] == "Admin")
            {
                $msg = "Admin Login successfull";
                echo "<script type='text/javascript'>alert('$msg')</script>";
                header("refresh:0; url=../Admin/Admin_Stock.php");
            }

            else
            {
                $msg = "user Login successfull";
                echo "<script type='text/javascript'>alert('$msg')</script>";
                header("refresh:0; url=Home.php");
            }
        }

        else
        {
            $msg = " Email or password is wrong ";
            echo "<script type='text/javascript'>alert('$msg')</script>";
        }
    }
}
```

Figure 25 PHP-Login Code

After checking whether the email address and password entered here are in the database, if the relevant information is entered successfully, the administrator provides the opportunity to enter the administrator's account and the user's account and if the entry is successful, it is displayed through a popup message.

## 2. Register Page

- **Html code**

```
<body>
  <div class="main-box">
    <div class="tital"><center><h1>Sign up</h1></center></div>

    <form action="Signup.php" method="post" enctype="multipart/form-data">

      <div class="box">
        <center>
          <div class="input-box">

            <div class="button-box">
              <a href="Login.php"><div class="sign1">Login</div></a>
              <a href=""><div class="log1">Sign up</div></a>
            </div>

            <input type="text" class="form-control" name="Uname" id="Uname" placeholder="User Name" autofocus><br/>
            <input type="Email" class="form-control" name="Email" id="Email" placeholder="Email"><br/>
            <input type="Number" class="form-control" name="Te_Number" id="Te_Number" placeholder="Telephone Number" ><br/>
            <input type="Password" class="form-control" name="Password" id="Password" placeholder="Password" ><br/>

            <input type="file" name="image" ><br/>
            <input type="submit" value="Signup" name="submit" class="submitbtn">

            <input type="reset" value="Reset" id="reset" class="resetbtn">

          <!--</form>-->
        </div>
      </center>
    </form>
  </div>
</body>
```

Figure 26 HTML-Register Code

Here, a new user entering the website is given the opportunity to successfully register in the website through this. Also, the form was prepared using html language and while preparing this, it was created in the form of creator. It also checks whether the user has filled in all the data correctly. The process that takes place here can be clearly identified by the PHP field in the diagram below.

### PHP code

```
<?php
    include("dbconnect.php");
    $message="";
    if(isset($_POST['submit'])) // associative array
    {
        $Name = $_POST['Uname'];
        $Email = $_POST['Email'];
        $T_no = $_POST['Te_Number'];
        $Pass = $_POST['Password'];

        $file_name = $_FILES['image']['name'];
        $file_type = $_FILES['image']['type'];
        $file_size = $_FILES['image']['size'];
        $temp_name = $_FILES['image']['tmp_name'];

        $upload_to = 'image/';

        move_uploaded_file($temp_name, $upload_to . $file_name);

        $url = $upload_to . $file_name;

        if($Name == "" || $Email == "" || $T_no == "" || $Pass == "")
        {
            $alert = "<script> alert('Empty text boes');</script>";
            echo $alert;
        }
        else
        {
            $sql2 = "INSERT INTO users( Uname, Email, Te_No, Type, Password, Image) VALUES('$Name', '$Email', '$T_no', 'User', '$Pass', '$url')";
            $results2 = mysqli_query($conn, $sql2);

            if(!$results2)
            {
                $alert = "<script> alert(' User Registration Fail');</script>";
                echo $alert;
            }
            else
            {
                $alert = "<script> alert('user registered successfully');</script>";
                echo $alert;
                echo '<META HTTP-EQUIV="Refresh" Content="0; URL=>login.php">';
            }
        }
    }
?
```

Figure 27 PHP-Register Code

The above code will take the data entered in the text box in the user html form and enter the data into the currently created table in the users database. If the correct entry is entered, the system will display a message as user registered successfully. Also, if the data is not successfully entered into the database, it will be displayed as User Registration Fail.

### 3. Database Connection file

```
<body>

    <?php

        $dbhost = 'localhost';
        $dbuser = 'root';
        $dbpass = '';

        $conn = mysqli_connect($dbhost, $dbuser, $dbpass);

        if(!$conn)
        {
            echo "could not connected";
        }

        echo "<br/>";

        $db = mysqli_select_db($conn, 'allinone');

        if(!$db)
        {
            echo "select database first";
        }

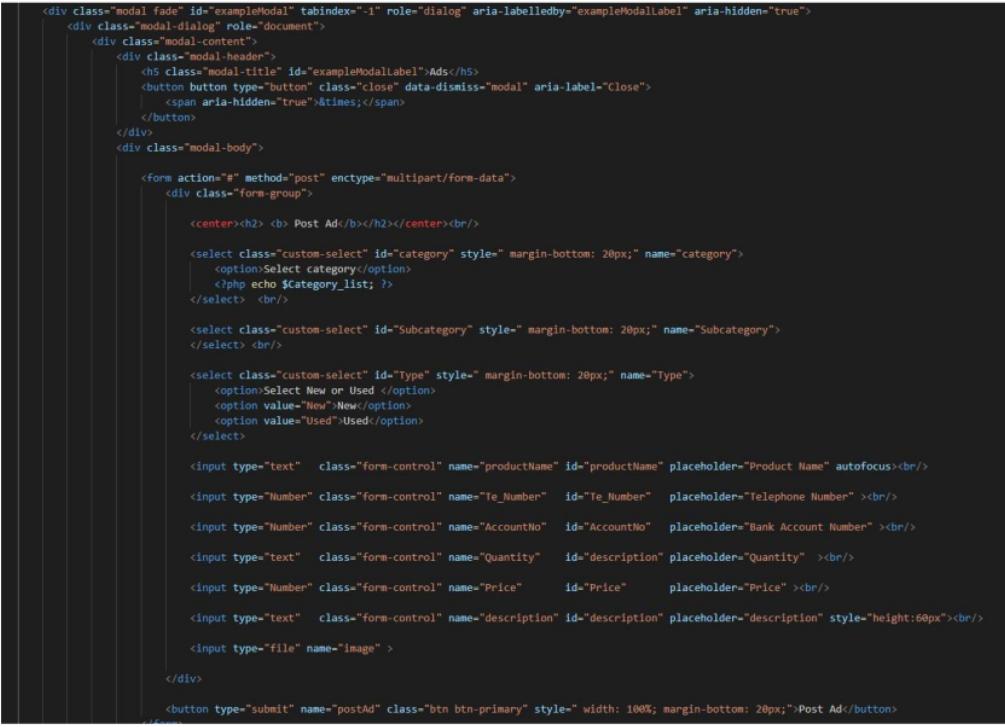
    ?>

</body>
```

Figure 28 Database Code

The above diagram shows the database used primarily to create the website. And in the above diagram "allinone" is the name of the database. Using this, the creation of the website has been done.

#### 4. Post Ad



The image shows a screenshot of an HTML file containing the code for a modal dialog titled "Post Ad". The modal includes fields for selecting a category, subcategory, type (New or Used), product name, telephone number, account number, quantity, price, and a description, along with a file input for an image and a submit button.

```
<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Ads</h5>
        <button button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        <form action="#" method="post" enctype="multipart/form-data">
          <div class="form-group">
            <center><b> Post Ad</b></center><br/>
            <select class="custom-select" id="category" style=" margin-bottom: 20px;" name="category">
              <option>Select category</option>
              <php echo $category_list; >
            </select> <br/>
            <select class="custom-select" id="Subcategory" style=" margin-bottom: 20px;" name="Subcategory">
              </select> <br/>
            <select class="custom-select" id="Type" style=" margin-bottom: 20px;" name="Type">
              <option>Select New or Used </option>
              <option value="New">New</option>
              <option value="Used">Used</option>
            </select>
            <input type="text" class="form-control" name="productName" id="productName" placeholder="Product Name" autofocus><br/>
            <input type="Number" class="form-control" name="Te_Number" id="Te_Number" placeholder="Telephone Number" ><br/>
            <input type="Number" class="form-control" name="AccountNo" id="AccountNo" placeholder="Bank Account Number" ><br/>
            <input type="text" class="form-control" name="Quantity" id="description" placeholder="Quantity" ><br/>
            <input type="Number" class="form-control" name="Price" id="Price" placeholder="Price" ><br/>
            <input type="text" class="form-control" name="description" id="description" placeholder="description" style="height:60px"><br/>
            <input type="file" name="image" >
          </div>
          <button type="submit" name="posted" class="btn btn-primary" style=" width: 100%; margin-bottom: 20px;">Post Ad</button>
        </form>
      </div>
    </div>
  </div>

```

Figure 29 HTML-Post Ad Code

The above image shows the html code section used to create the form that needs to be filled to enter a new post in the website. These are created very creatively using html codes.

- PHP code

```

$message="";
if(isset($_POST['postAd'])) // associative array
{
    $category_Id = $_POST['category'];
    $subcategory_Id = $_POST['subcategory'];
    $type = $_POST['type'];
    $productname = $_POST['productname'];
    $Stc_Number = $_POST['Stc_Number'];
    $description = $_POST['description'];
    $price = $_POST['Price'];
    $Quantity = $_POST['Quantity'];
    $AccountNo = $_POST['AccountNo'];

    $file_name = $_FILES['image']['name'];
    $file_type = $_FILES['image']['type'];
    $file_size = $_FILES['image']['size'];
    $temp_name = $_FILES['image']['tmp_name'];

    $upload_to = 'image/';
    move_uploaded_file($temp_name, $upload_to . $file_name);
    $url = $upload_to . $file_name;

    if($productname == "" || $Stc_Number == "" || $description == "" || $price == "" || $Quantity == "" || $AccountNo == "")
    {
        $alert = "<script> alert('one or more text boxes is empty');</script>";
        echo $alert;
    }
    else{
        $sql_Sub = "SELECT * FROM sub WHERE Sid = '$subcategory_Id'";
        $result_Sub = mysqli_query($conn, $sql_Sub);
        $row_Sub = mysqli_fetch_array($result_Sub);

        $ordertype = $row_Sub['Order_Type'];

        $sql_1 = "INSERT INTO product(Phone, Tell_no, Description, Price, Category_Id, SubCategory_Id, Type, image, Uid, order_type, Quantity, AccountNo) VALUES('$productname','$Stc_Number','$description','$price', '$category_Id', '$subcategory_Id', '$type', '$url', '$ordertype', '$Quantity', '$AccountNo')";
        $results_1 = mysqli_query($conn, $sql_1);

        if(!$results_1)
        {
            $alert = "<script> alert('$op Post ad fail');</script>";
            echo $alert;
        }
        else
        {
            $select = "<script> alert('Product post successful');</script>";
            echo $select;
            echo '<META HTTP-EQUIV="Refresh" Content="0; URL=index.php">';
        }
    }
}

```

Figure 30 PHP-Post Ad Code

What happens here is that the data currently entered in the input field is sent to the database called product. In addition to this, the facility required to insert an image into the relevant product is also indicated by the code in the diagram above.

## 5. Create category

- **Html Code**

```
<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Category</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">

        <form action="#" method="post">
          <div class="form-group">
            <center><b> Create Category</b></center><br/>
            <input type="text" class="form-control" name="CName" id="CName" placeholder="Category Name"><br/>
            <input type="text" class="form-control" name="description" id="description" placeholder="Category description" ><br/>
          </div>

          <button type="submit" name="Insert" class="btn btn-primary">Create</button>
        </form>
      </div>
    </div>
  </div>
</div>
```

Figure 31 HTML- create category

The html code in the above image has created the form required to enter the category. Here, the data entered in the relevant **form** by the user is sent to the database by clicking the button.

## PHP Code

```
$message="";
if(isset($_POST['Insert'])) // associative array
{
    $Name = $_POST['CName'];
    $description = $_POST['description'];

    if($Name == "" || $description == "")
    {
        $alert = "<script> alert('Empty text boxes');</script>";
        echo "$alert";
    }
    else{
        $sql = "INSERT INTO category(CName, Cdescription) VALUES('$Name', '$description')";

        $results = mysqli_query($conn, $sql);

        if(!$results)
        {
            $msg = "Failed";
        }
        else
        {
            $alert = "<script> alert('Create Category successful');</script>";
            echo "$alert";
            echo "<META HTTP-EQUIV=Refresh Content=0; URL=Admin_Category.php>";
        }
    }
}
?>
```

Figure 32 PHP-Create Category

The PHP code shown in the image above gets the data currently entered in the user html form and enters the data into the category table in the database. A "Create Category successful" message will be displayed.

## 6. Update Category

- **PHP code**

```
if(isset($_GET['Edit']))
{
    $Id     = $_GET['Id'];

    $sql2 = "SELECT * FROM category WHERE CId ='$Id'";
    $result2 = mysqli_query($conn, $sql2);
    $row2 = mysqli_fetch_array($result2);

    $cName      = $row2['CName'];
    $cdescription = $row2['Cdescription'];

}
?>
```

Figure 33 PHP-Update Category

The PHP code shown in the above diagram retrieves the information related to the selected row from the database and enters the data into the html form shown in the figure below.

- **Html code**

```
<div class="modal Fade" id="exampleModal2" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Category</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        <form action="#" method="post">
          <div class="form-group">
            <center><b> Update Category:</b></center><br/>
            <input type="text" class="form-control" name="Ucid" id="Ucid" value=<?php echo $Id; ?>" autofocus><br/>
            <input type="text" class="form-control" name="Uname" id="Name" value=<?php echo $Name; ?>"><br/>
            <input type="text" class="form-control" name="Udescription" id="Des" value=<?php echo $description; ?>"><br/>
          </div>
          <button type="submit" name="Update" class="btn btn-primary">Update</button>
        </form>
      </div>
    </div>
  </div>
</div>
```

Figure 34 HTML-Update Category

## 7. Delete category

```
<?php

if(isset($_GET['Delete']))
{
    $Id    = $_GET['Id'];

    $sql4 = "DELETE FROM category WHERE CId ='$Id'";
    $result4 = mysqli_query($conn, $sql4);

    if(!$result4)
    {
        $msg = "faield";
    }
    else
    {
        $alert = "<script> alert('Delete successful');</script>";
        echo "$alert";
        echo '<META HTTP-EQUIV="Refresh" Content="0; URL=Admin_Category.php">';
    }
}
?>
```

Figure 35 PHP-Delete Category

The html code block shown in the above diagram retrieves the information related to the selected row from the database and deletes that row from the database.

## Chapter 05

### 1. Testing.

- **Testing methodology and justification.**

Software testing methods are strategic plans and types of tests that ensure that the client's expectations are met. Each test method has a specific common purpose, test strategy and distribution.

The test methodology we used in developing this site is the **waterfall model**.

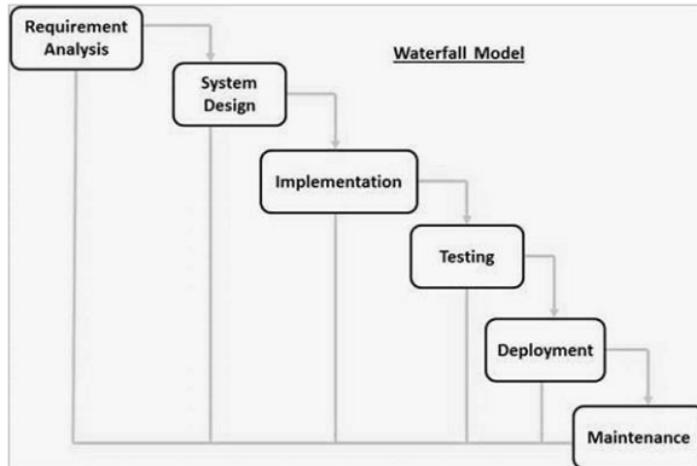


Figure 36 SDLC MODEL

This waterfall model is the most commonly used model in software engineering. This is the most popular type of model.

The main reason we use this model is because we can separate the development stages in the system development. Therefore, it is possible to carry out the work very successfully. And at some point, when something goes wrong, the process stops. It therefore has the ability to detect errors and make the system run successfully.

1  
o **Test plan.**

Test ID	Test Case Name	Scenario	Expected Result
Test 1	Logging in as an Admin	Checking that the email and password of the admin in the database are correct and providing the opportunity to log in.	If the email and password is correct, a message will be displayed as Admin login successful.
Test 2	Logging in by entering the admin's email or password incorrectly	Not providing an opportunity to log in when the admin's email or password in the database is not entered correctly.	If the <sup>2</sup> email or password is wrong, a message will be displayed as Email or password Is Wrong.
Test 3	Empty field validation ( admin )	Not giving a chance to log in without entering the admin's email or password in the database.	A message will be displayed as email or password is wrong.
Test 4	Logging in as an User	Checking that the email and password of the User in the database are correct and providing the opportunity to log in.	If the email and password is correct, a message will be displayed as User login successful.
Test 5	Logging in by entering the User email or password incorrectly	Not providing an opportunity to log in when the User email or password in the database is not entered correctly.	If the <sup>2</sup> email or password is wrong, a message will be displayed as Email or password Is Wrong.

Test 6	Empty field validation (User )	Not giving a chance to log in without entering the User email or password in the database.	A message will be displayed as email or password is wrong.
Test 7	Sign up	By completing the text field on this sign up page correctly and clicking the sign up button, a user is successfully registered in the database.	A message will be displayed as user register successfull.
Test 8	Sign up empty ( field validation )	The opportunity to add a user to the database is not given without completing the text field on this sign up page.	A message is displayed as an empty text box
Test 9	Clicking the Post add button.	After clicking the Post add button, there are two buttons as Login and Register.	After clicking the Login button and providing the correct email - password, a message will be displayed as User Login Successful.
Test 10	Subcategories (Vehicle Button)	All the details in the database related to the subcategory can be seen by clicking this details button.	All information is successfully displayed
Test 11	Subcategories (Computer Button)	All the details in the database related to the subcategory can be seen by clicking this details button.	All information is successfully displayed

Test 12	Buy now (giving wrong quantity).	In the required category, it is not possible to buy more quantity than what is available in the stoke.	A message saying Value must be less than or equal will be displayed
Test 13	Buy Now (providing correct quantity)	You can buy the desired product by giving the quantity of the product you want to buy or less and clicking the buy now button.	After giving the required quantity and clicking the buy now button, it will be taken to Card payment to pay for the required product.
Test 14	Product payment page	You can pay for the required product by filling the text field correctly on the product payment page and clicking the pay now button.	A message will be displayed as payment successful.
Test 15	Product payment page (Empty Field validation)	When paying for a product, you will not be given the opportunity to pay for the product without completing the text field included in that page	A message "No fill card details" will be displayed.
Test 16	Stoke page (admin)	Here the Admin can see the details of the product types and sizes currently in the database. Admin can click the Delete button at any time and delete the relevant product from the database.	A delete successful message will be displayed.

Test 17	Stoke page ( Admin – Add Product )	When admin adds a product to the database, click the Add products button here, fill in the text field correctly and click the post add button, then a product is successfully entered into the database.	A message will be displayed as product post successful.
Text 18	Stoke page ( Admin – Add product Empty Field validation )	After clicking the Add button, if the text field is not filled, the product will not be added to the database.	The message is displayed as an empty text box.
Test 19	Create Category Button	Here, after filling the Category Name and Category Description text boxes, it will be saved in the database and will appear under category on the home page.	A message will be displayed as Create Category Successfully.

Test 20	Empty Field validation	Not giving an opportunity to add Category without including any Category name and Category Description	Messages are displayed as empty text boxes
Test 21	Edit button	By correctly updating the information in the Update category and clicking the update button, the information in that table in the database will be updated.	A message as Update successfully will be displayed.
Test 22	Delete button	By clicking the Delete button, the relevant information in the Category menu of the database will be deleted.	A message will be displayed as Delete successfully.
Test 23	Empty Field validation (Sub Categories )	In Add Sub Category, adding a sub category does not allow to add it to the database when there is a text box or it is empty.	The message is displayed as an Empty Text box.
Test 24	Add Subcategories ( Button )	By filling the text boxes correctly in Add sub category and clicking Create button, it will be saved in the database and displayed as a sub category in the category on the home page.	A message will be displayed as Create Category Successful

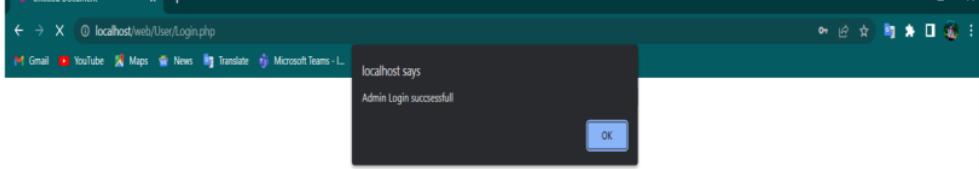
Test 25	Sub Categories (Edit button)	By correctly updating the information in the Update category and clicking the update button, the information in that table in the database will be updated.	A message as Update successfully will be displayed.
Test 26	Sub Categories(Delete button )	By clicking the Delete button, the relevant information in the Category menu of the database will be deleted.	A message will be displayed as Delete successfully.
Test 27	Add User	In Add User, fill the text box correctly, select admin or user and click Add user button, it will be saved in the database table.	A message will be displayed as Add user Successfully.
Test 28	Empty Field validation (Add User )	When the text box is empty in Add user, it does not allow to add to the database	A message is displayed as an empty text box
Test 29	Edit button (User Insert Page )		

		By correctly updating the information in the Update User Details and clicking the update button, the information in that table in the database will be updated.	A message as Update successfully will be displayed.
--	--	---	---

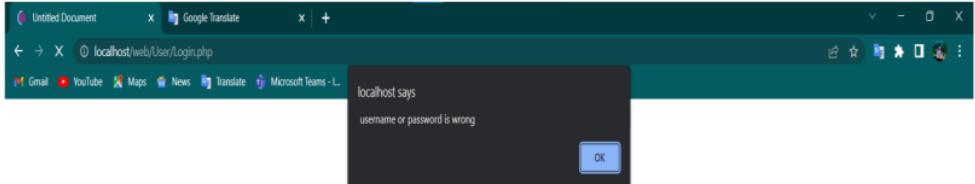
Test 30	Delete button (User Insert Page )	By clicking the Delete button, the relevant information in the User Details menu of the database will be deleted.	A message will be displayed as Delete successfully.
Test 31	User login Image	After the user logs into the user's home page and clicks on the image in front of the user name, the user can see all the Adds that he has entered the website.	My Adds login successfully
Test 32	User login Image	After logging in to the user's home page and clicking on the image in front of the user name, the user will be given the opportunity to log out of the user page.	A message will be displayed as log out successful
Test 33	User page ( post Ad )	After filling the text field in the post ad page correctly, it is saved in a table in the database. It appears as an add on the	A message will be displayed as Product post successful.

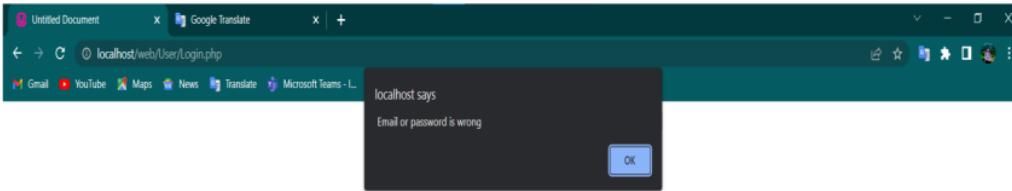
		home page in the relevant category.	
Test 34	Empty Field validation ( post Ad )	When posting an add, without completing the text field, it is not given an opportunity to enter it into the database	A message is displayed as an empty text box
Test 35	Search Bar	All products in the database can be easily found by entering the name of the product in the search bar.	Searched product is loaded successfully.

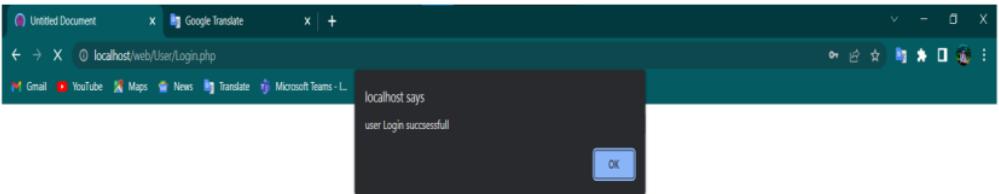
- **Test cases.**

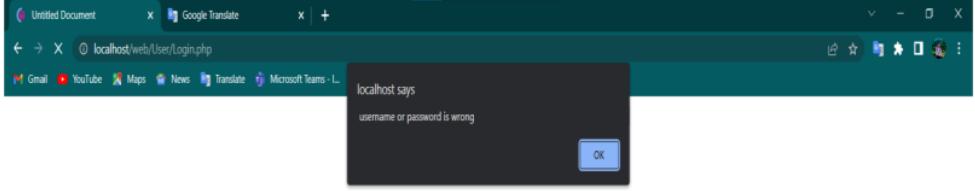
Test case	Test 1
Test Objective	Logging in as an Admin
Test Data	Checking that the email and password of the admin in the database are correct and providing the opportunity to log in.
Expected results	If the email and password is correct, a message will be displayed as Admin login successful. 1
Screenshots	
Conclusion	Functioning as intended.

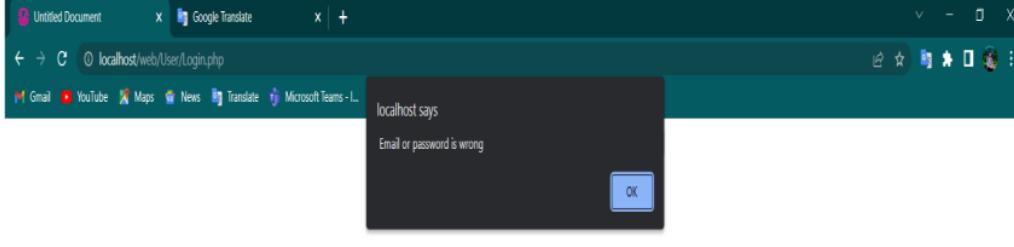


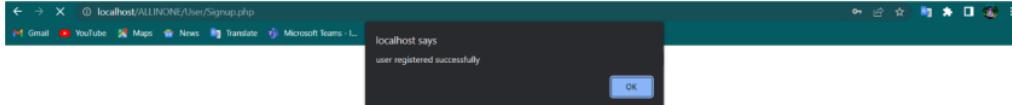
Test case	Test 2
Test	Logging in by entering the admin's email or password incorrectly
Objective	
Test Data	Not providing an opportunity to log in when the admin's email or password in the database is not entered correctly.
Expected results	If the email or password is wrong, a message will be displayed as Email or password Is Wrong.
1 Screenshots	
Conclusion	Functioning as intended.

<b>Test case</b>	Test 3
<b>Test</b>	Empty field validation
<b>Objective</b>	
<b>Test Data</b>	Not giving a chance to log in without entering the admin's email or password in the database.
<b>Expected results</b>	A message will be displayed as email or password is wrong.
<b>1 Screenshots</b>	 <p>The screenshot shows a browser window with the URL 'localhost/web/User/Login.php'. A dark gray modal dialog box is centered on the page. The dialog contains the text 'localhost says' at the top and 'Email or password is wrong' below it. In the bottom right corner of the dialog is a small blue 'OK' button. The background of the browser window is white, and there are some navigation icons and links visible at the top.</p>
<b>Conclusion</b>	Functioning as intended.

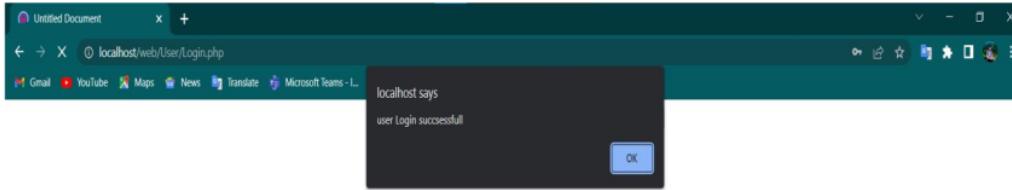
Test case	Test 4
Test	Logging in as an User
Objective	
Test Data	Checking that the email and password of the User in the database are correct and providing the opportunity to log in.
Expected results	If the email and password is correct, a message will be displayed as User login successful. 1
Screenshots	 <p>A screenshot of a web browser window titled 'Untitled Document'. The address bar shows 'localhost/web/User/Login.php'. A modal dialog box is centered on the screen with the text 'localhost says' and 'user Login successful' below it. An 'OK' button is at the bottom right of the dialog. The background of the browser shows other tabs and links like Gmail, YouTube, Maps, News, Translate, and Microsoft Teams.</p>
Conclusion	Functioning as intended.

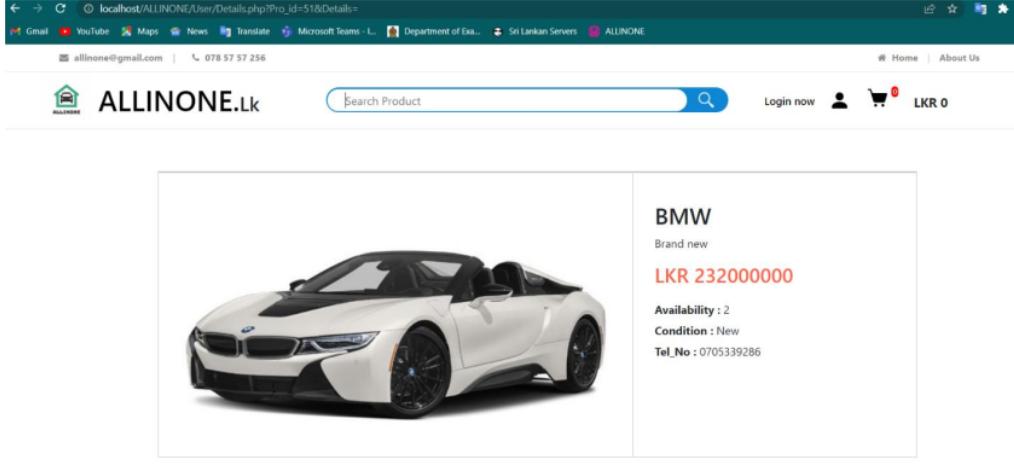
<b>Test case</b>	Test 5
<b>Test</b>	
<b>Objective</b>	Logging in by entering the User email or password incorrectly
Test Data	Not providing an opportunity to log in when the User email or password in the database is not entered correctly. 2.
Expected results 1	If the email or password is wrong, a message will be displayed as Email or password Is Wrong.
<b>Screenshots</b>	
<b>Conclusion</b>	Functioning as intended.

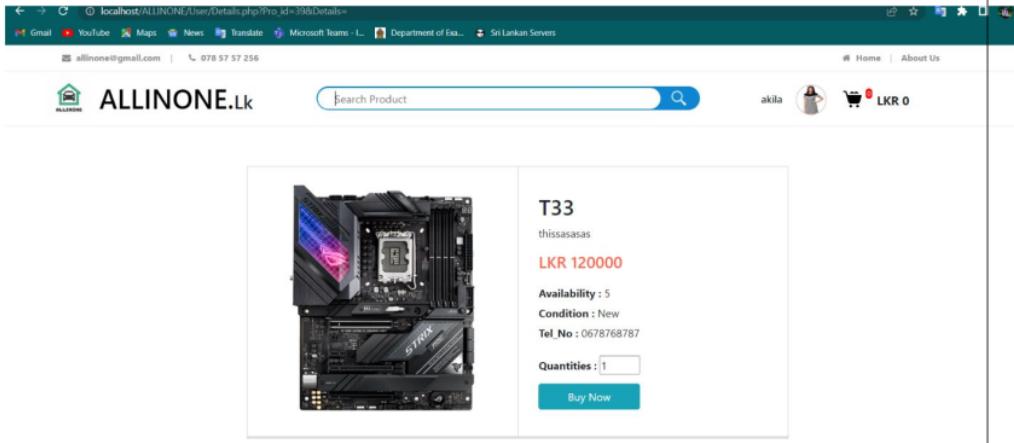
<b>Test case</b>	Test 6
Test	Empty field validation ( user )
<b>Objective</b>	
<b>Test Data</b>	Not giving a chance to log in without entering the User email or password in the database.
Expected results	A message will be displayed as email or password is wrong.
1 <b>Screenshots</b>	
<b>Conclusion</b>	Functioning as intended.

Test case	Test 7
Test Objective	Sign up
Test Data	By completing <b>the</b> text field on this sign up page correctly and clicking the sign up button, a user is successfully registered in the database.
Expected results	A message <b>will be displayed as</b> user register successful. 1
Screenshots	 A screenshot of a web browser window titled "localhost/ALLINONE/User/Signup.php". The address bar shows the URL. The main content area displays a dark gray modal dialog box with white text. The text reads "localhost says" followed by "user registered successfully" and an "OK" button at the bottom right. The browser's toolbar and various icons are visible at the top of the window.
Conclusion	Functioning as intended.

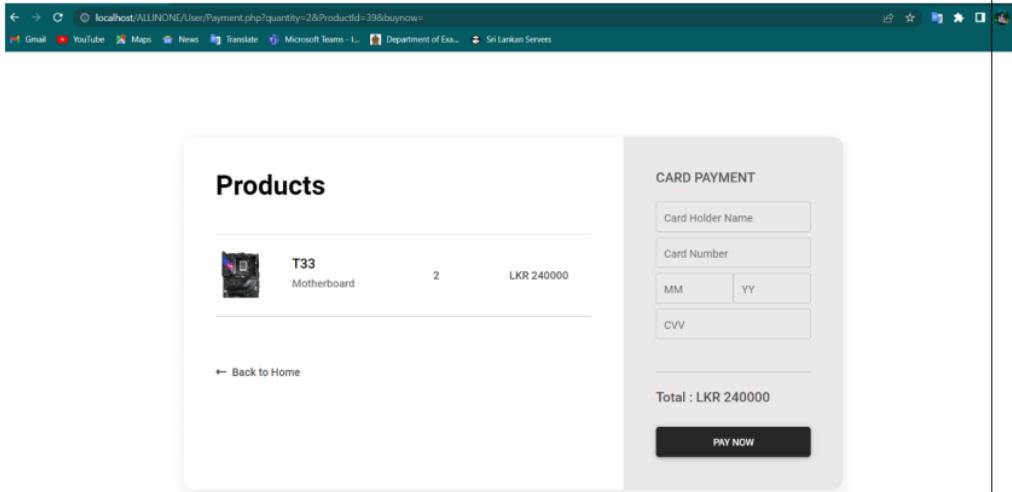
<b>Test case</b>	Test 8
<b>Test</b>	Sign up empty ( field validation )
<b>Objective</b>	
<b>Test Data</b>	The opportunity to add a user to the database is not given without completing the text field on this sign up page.
<b>Expected results</b>	A message is displayed as an empty text box
<b>1 Screenshots</b>	
<b>Conclusion</b>	Functioning as intended.

Test case	Test 9
Test	Clicking the Post add button.
Objective	
Test Data	After clicking the Post add button, there are two buttons as Login and Register.
Expected results	After clicking the Login button and providing the correct Email - password, a message will be displayed a User Login Successful.
1 Screenshots	 <p>A screenshot of a web browser window titled 'Untitled Document'. The address bar shows 'localhost/web/User/Login.php'. A dark gray modal dialog box is centered on the page with the text 'localhost says' and 'user Login successful' below it. An 'OK' button is at the bottom right of the dialog. The background of the browser shows a navigation bar with links like Gmail, YouTube, Maps, News, Translate, and Microsoft teams.</p>
Conclusion	Functioning as intended.

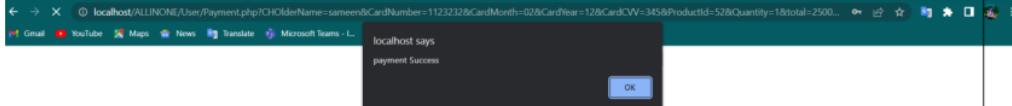
Test case	Test 10
Test	Subcategories (Vehicle Button)
Objective	
Test Data	All the details in the database related to the subcategory can be seen by clicking this details button.
Expected results	All information is successfully displayed <span style="color: red; border: 1px solid red; padding: 2px 5px;">1</span>
Screenshots	
Conclusion	Functioning as intended.

Test case	Test 11
Test	Subcategories (Computer Button)
Objective	
Test Data	All the details in the database related to the subcategory can be seen by clicking this details button.
Expected results	All information is successfully displayed
Screenshots	
Conclusion	Functioning as intended.

<b>Test case</b>	Test 12
<b>Test</b>	Buy now (giving wrong quantity).
<b>Objective</b>	
<b>Test Data</b>	In the required category, it is not possible to buy more quantity than what is available in the stoke.
<b>Expected results</b>	A message saying Value must be less than or equal <span style="background-color: #f0e6d2; border: 1px solid #ccc; padding: 2px;">will be displayed</span>
<b>Screenshots</b>	
<b>Conclusion</b>	Functioning as intended.

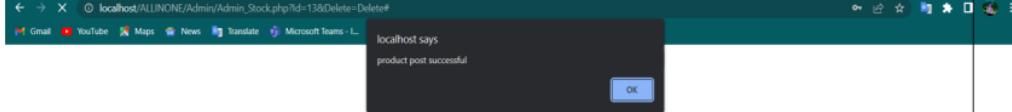
<b>Test case</b>	Test 13
<b>Test</b>	Buy Now (providing correct quantity)
<b>Objective</b>	
<b>Test Data</b>	You can buy the desired product by giving the quantity of the product you want to buy or less and clicking the buy now button.
<b>Expected results</b>	After giving the required quantity and clicking the buy now button, it will be taken to Card payment to pay for the required product.
<b>Screenshots</b>	

Conclusion	Functioning as intended.
------------	--------------------------

Test case	Test 14
Test	Product payment page
Objective	
Test Data	You can pay for the required product by filling the text field correctly on the product payment page and clicking the pay now button. 1
Expected results	A message will be displayed as payment successful.
Screenshots	 A screenshot of a browser window titled "localhost/ALLINONE/User/Payment.php?CHolderName=sameen&CardNumber=11232328&CardMonth=02&CardYear=12&CardCVV=345&ProductId=52&Quantity=1&total=2500...". The main content area displays the message "localhost says payment Success" in a black box. An "OK" button is visible at the bottom right of the message box. The browser's address bar shows the full URL. The taskbar at the bottom of the screen includes icons for Gmail, YouTube, Maps, News, Translate, Microsoft Teams, and others.
Conclusion	Functioning as intended.

Test case	Test 15
Test	Product payment page (Empty Field validation)
Objective	
Test Data	When paying for a product, you will not be given the opportunity to pay for the product without completing the text field included in that page
Expected results	A message "No fill card details" will be displayed.
Screenshots	
Conclusion	Functioning as intended.

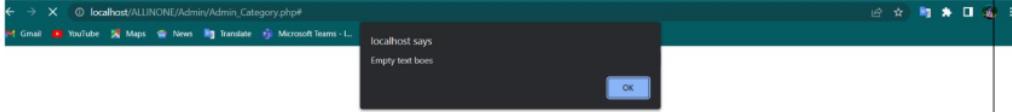
<b>Test case</b>	Test 16
<b>Test</b>	Stoke page (admin)
<b>Objective</b>	
<b>Test Data</b>	Here <b>the Admin</b> can see the details of the product types and sizes currently in the database. Admin can click the Delete button at any time and delete the relevant product from the database.
<b>Expected results</b>	A delete successful message will be displayed.
<b>Screenshots</b>	
<b>Conclusion</b>	Functioning as intended.

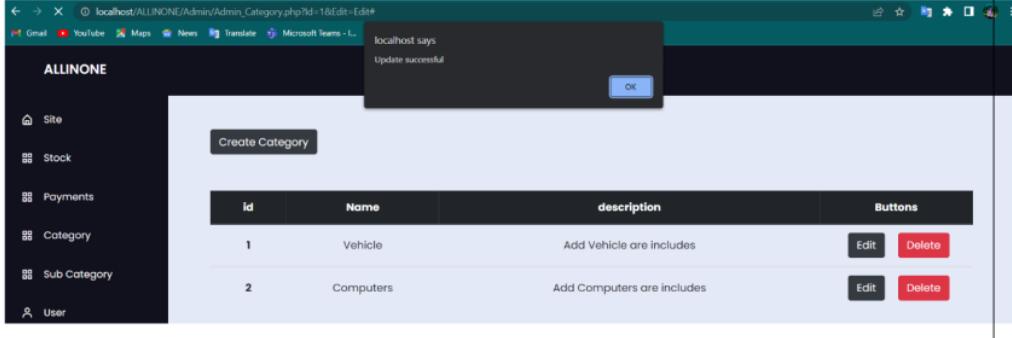
Test case	Test 17
Test	Stoke page ( Admin – Add Product )
Objective	
Test Data	When admin adds a product to the database, click the Add products button here, fill in the text field correctly and click the post add button, then a product is successfully entered into the database.
Expected results	A message will be displayed as product post successful.
Screenshots	 <p>The screenshot shows a browser window with the URL 'localhost/ALLNONE/Admin/Admin_Stock.php?id=1380Delete-Delete#'. A black modal dialog box is centered on the screen, displaying the text 'localhost says' and 'product post successful' below it. An 'OK' button is visible at the bottom right of the dialog. The browser's address bar and various icons are visible at the top.</p>
Conclusion	Functioning as intended.

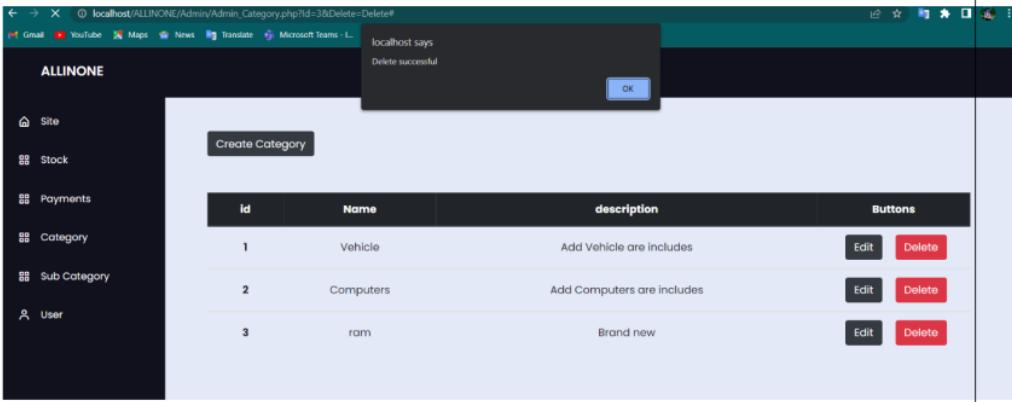
Test case	Test 18										
Test	Stoke page ( Admin – Add product Empty Field validation )										
Objective 1											
Test Data	After clicking the Add button, if the text field is not filled, the product will not be added to the database.										
Expected results	The message is displayed as an empty text box.										
Screenshots	<p>localhost says Empty text box</p> <table border="1"> <thead> <tr> <th>Product Id</th> <th>Product Infor</th> <th>Available</th> <th>Price</th> <th>Buttons</th> </tr> </thead> <tbody> <tr> <td>39</td> <td>T33 UserID - I3</td> <td>5</td> <td>120000</td> <td><a href="#">Delete</a></td> </tr> </tbody> </table>	Product Id	Product Infor	Available	Price	Buttons	39	T33 UserID - I3	5	120000	<a href="#">Delete</a>
Product Id	Product Infor	Available	Price	Buttons							
39	T33 UserID - I3	5	120000	<a href="#">Delete</a>							

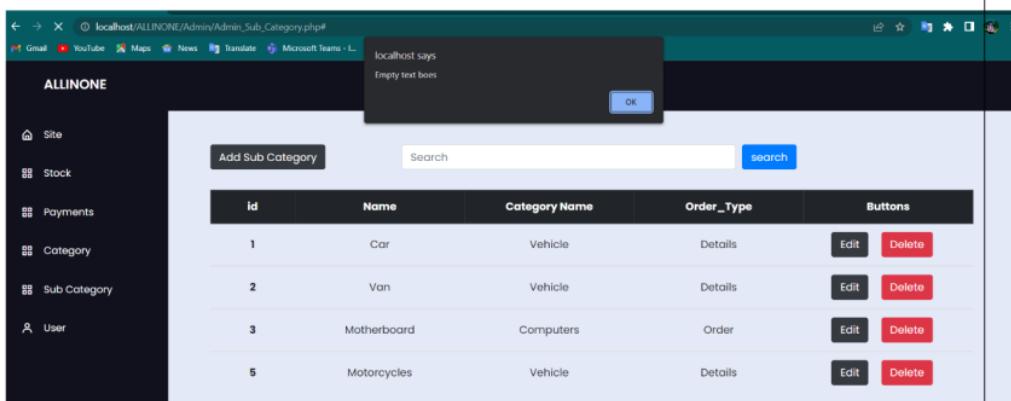
1	Conclusion	Functioning as intended.
---	------------	--------------------------

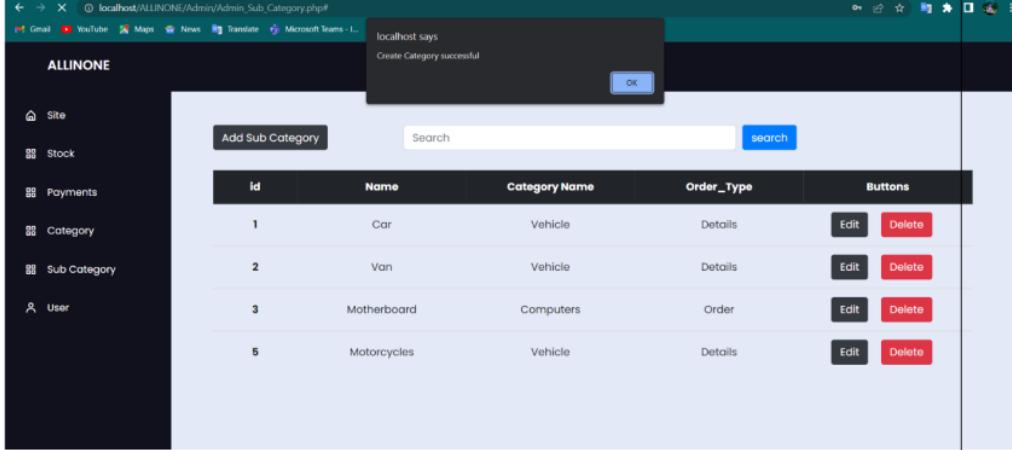
Test case	Test 19
Test	Create Category Button
Objective	
Test Data	Here, after filling the Category Name and Category Description text boxes, it will be saved in the database and will appear under category on the home page. 1
Expected results	A message will be displayed as Create Category Successfully.
Screenshots	 A screenshot of a web browser window titled "localhost:8080/Admin/Admin_Category.php". The address bar shows the URL. The main content area is mostly blank. A dark gray modal dialog box is centered, displaying the text "localhost says" and "Create Category successful" in white. An "OK" button is visible at the bottom right of the dialog. The browser's toolbar and menu bar are visible at the top.
Conclusion	Functioning as intended.

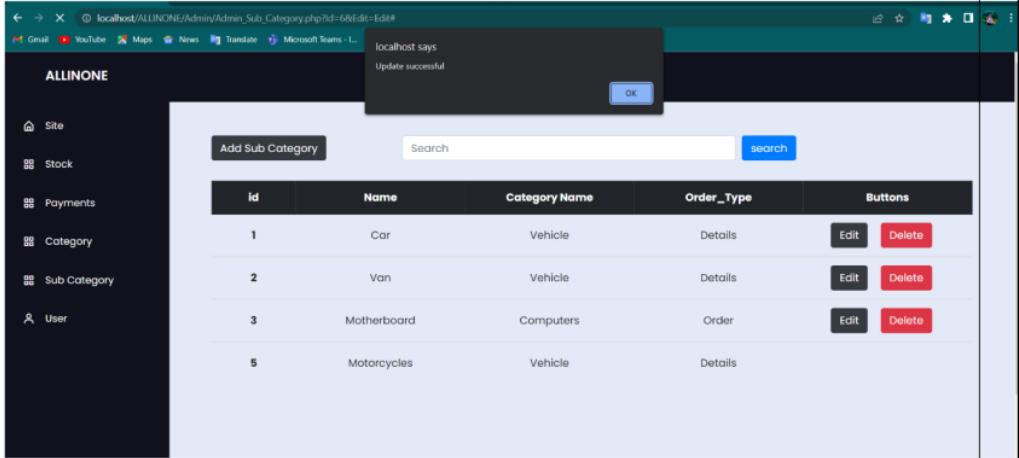
<b>Test case</b>	Test 20
<b>Test</b>	Empty Field validation
<b>Objective</b>	
<b>Test Data</b>	Not giving an opportunity to add Category without including any Category name and Category Description
<b>Expected results</b>	Messages are displayed as empty text boxes
<b>1 Screenshots</b>	
<b>Conclusion</b>	Functioning as intended.

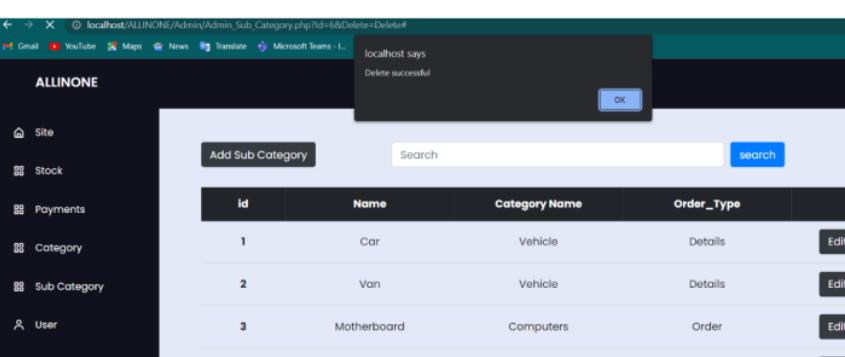
Test case	Test 21
Test	Edit button
Objective	
Test Data	By correctly updating the information in the Update category and clicking the update button, the information in that table in the database will be updated.
Expected results	A message as Update successfully will be displayed.
1 Screenshots	 <p>The screenshot shows a web browser window with the URL <code>localhost/ALLINONE/Admin/Admin_Category.php?id=1&amp;Edit=Edit</code>. A modal dialog box is open, displaying the message "localhost says Update successful" with an "OK" button. The background page shows a table titled "Create Category" with two rows of data. The first row has an ID of 1, a Name of "Vehicle", and a description of "Add Vehicle are includes". The second row has an ID of 2, a Name of "Computers", and a description of "Add Computers are includes". Both rows have "Edit" and "Delete" buttons in the "Buttons" column.</p>
Conclusion	Functioning as intended.

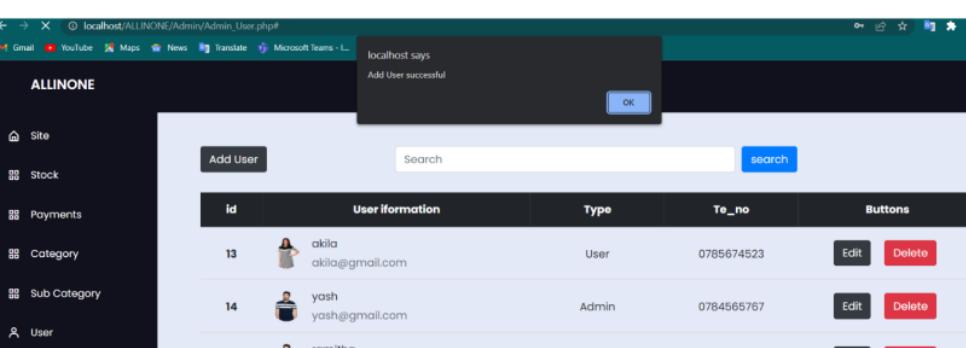
Test case	Test 22																			
Test	Delete button																			
Objective																				
Test Data	By clicking the Delete button, the relevant information in the Category menu of the database will be deleted.																			
Expected results	A message will be displayed as Delete successfully																			
Screenshots	 <p>The screenshot shows a web application interface for managing categories. On the left, there's a sidebar with links like Site, Stock, Payments, Category, Sub Category, and User. The main content area has a title 'Create Category' and a table with columns: Id, Name, description, and Buttons. The table contains three rows:</p> <table border="1"> <thead> <tr> <th data-bbox="518 1079 551 1100">Id</th> <th data-bbox="621 1079 682 1100">Name</th> <th data-bbox="910 1079 971 1100">description</th> <th data-bbox="1188 1079 1232 1100">Buttons</th> </tr> </thead> <tbody> <tr> <td data-bbox="556 1115 572 1136">1</td> <td data-bbox="643 1115 687 1136">Vehicle</td> <td data-bbox="866 1115 1008 1136">Add Vehicle are includes</td> <td data-bbox="1192 1115 1209 1136"><a href="#">Edit</a></td> <td data-bbox="1230 1115 1246 1136"><a href="#">Delete</a></td> </tr> <tr> <td data-bbox="556 1157 572 1178">2</td> <td data-bbox="643 1157 698 1178">Computers</td> <td data-bbox="866 1157 1008 1178">Add Computers are includes</td> <td data-bbox="1192 1157 1209 1178"><a href="#">Edit</a></td> <td data-bbox="1230 1157 1246 1178"><a href="#">Delete</a></td> </tr> <tr> <td data-bbox="556 1199 572 1220">3</td> <td data-bbox="643 1199 670 1220">ram</td> <td data-bbox="910 1199 954 1220">Brand new</td> <td data-bbox="1192 1199 1209 1220"><a href="#">Edit</a></td> <td data-bbox="1230 1199 1246 1220"><a href="#">Delete</a></td> </tr> </tbody> </table>	Id	Name	description	Buttons	1	Vehicle	Add Vehicle are includes	<a href="#">Edit</a>	<a href="#">Delete</a>	2	Computers	Add Computers are includes	<a href="#">Edit</a>	<a href="#">Delete</a>	3	ram	Brand new	<a href="#">Edit</a>	<a href="#">Delete</a>
Id	Name	description	Buttons																	
1	Vehicle	Add Vehicle are includes	<a href="#">Edit</a>	<a href="#">Delete</a>																
2	Computers	Add Computers are includes	<a href="#">Edit</a>	<a href="#">Delete</a>																
3	ram	Brand new	<a href="#">Edit</a>	<a href="#">Delete</a>																
Conclusion	Functioning as intended.																			

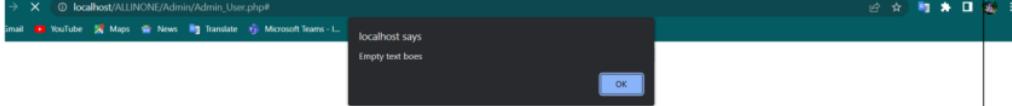
Test case	Test 23
Test	Empty Field validation (Sub Categories)
Objective	
Test Data	In Add Sub Category, adding a sub category does not allow to add it to the database when there is a text box or it is empty.
Expected results	The message is displayed as an Empty Text box.
1 Screenshots	 <p>A screenshot of a web browser displaying a modal dialog box. The dialog box has a dark background with white text that reads "localhost says" followed by "Empty text box". In the bottom right corner of the dialog box is a blue "OK" button. Behind the dialog box, a table is visible. The table has a header row with columns "id", "Name", "Category Name", "Order_Type", and "Buttons". There are five rows of data in the table. The first row contains "1", "Car", "Vehicle", "Details", and two buttons labeled "Edit" and "Delete". The second row contains "2", "Van", "Vehicle", "Details", and two buttons labeled "Edit" and "Delete". The third row contains "3", "Motherboard", "Computers", "Order", and two buttons labeled "Edit" and "Delete". The fourth row contains "5", "Motorcycles", "Vehicle", "Details", and two buttons labeled "Edit" and "Delete".</p>
Conclusion	Functioning as intended.

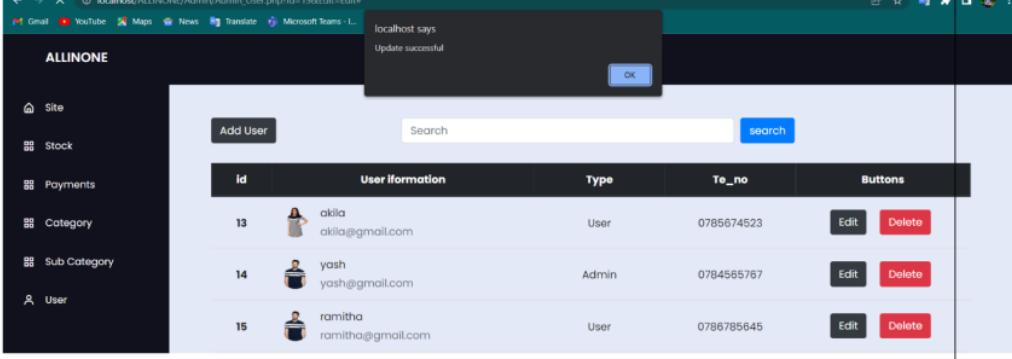
Test case	Test 24																									
Test	Add Subcategories (Button)																									
Objective																										
Test Data	By filling the text boxes correctly in Add sub category and clicking Create button, it will be saved in the database and displayed as a sub category in the category on the home page. 1																									
Expected results	A message will be displayed as Create Category Successful																									
Screenshots	 <p>The screenshot shows a web browser window with the URL 'localhost/ALLINONE/Admin/Admin_Sub_Category.php'. The page title is 'ALLINONE'. On the left, there is a sidebar with navigation links: Site, Stock, Payments, Category, Sub Category, and User. The main content area has a header 'Add Sub Category' with a search bar and a 'search' button. Below this is a table with the following data:</p> <table border="1"> <thead> <tr> <th>ID</th> <th>Name</th> <th>Category Name</th> <th>Order_Type</th> <th>Buttons</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Car</td> <td>Vehicle</td> <td>Details</td> <td>Edit Delete</td> </tr> <tr> <td>2</td> <td>Van</td> <td>Vehicle</td> <td>Details</td> <td>Edit Delete</td> </tr> <tr> <td>3</td> <td>Motherboard</td> <td>Computers</td> <td>Order</td> <td>Edit Delete</td> </tr> <tr> <td>5</td> <td>Motorcycles</td> <td>Vehicle</td> <td>Details</td> <td>Edit Delete</td> </tr> </tbody> </table> <p>A modal dialog box is overlaid on the page, displaying the message 'localhost says Create Category successful' with an 'OK' button.</p>	ID	Name	Category Name	Order_Type	Buttons	1	Car	Vehicle	Details	Edit Delete	2	Van	Vehicle	Details	Edit Delete	3	Motherboard	Computers	Order	Edit Delete	5	Motorcycles	Vehicle	Details	Edit Delete
ID	Name	Category Name	Order_Type	Buttons																						
1	Car	Vehicle	Details	Edit Delete																						
2	Van	Vehicle	Details	Edit Delete																						
3	Motherboard	Computers	Order	Edit Delete																						
5	Motorcycles	Vehicle	Details	Edit Delete																						
Conclusion	Functioning as intended.																									

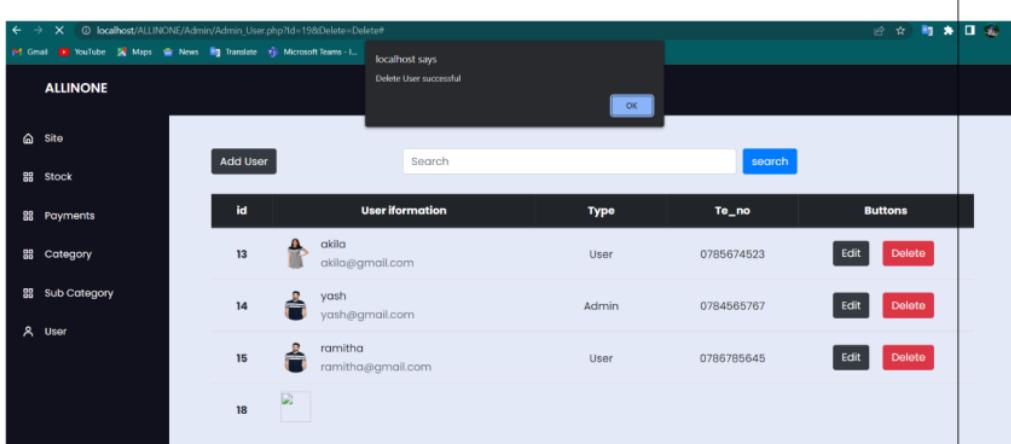
Test case	Test 25																									
Test Objective	Sub Categories (Edit button)																									
Test Data	By correctly updating the information in the Update category and clicking the update button, the information in that table in the database will be updated.																									
Expected results	A message as Update successfully will be displayed.																									
1 Screenshots	 <p>The screenshot shows a web browser window with the URL 'localhost/ALLINONE/Admin/Admin_Sub_Category.php?id=6&amp;edit=Edit'. The page title is 'ALLINONE'. On the left, there is a sidebar with links: Site, Stock, Payments, Category, Sub Category, and User. The main content area has a header 'Add Sub Category' with a search bar and a 'search' button. Below is a table with the following data:</p> <table border="1"> <thead> <tr> <th>ID</th> <th>Name</th> <th>Category Name</th> <th>Order_Type</th> <th>Buttons</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Car</td> <td>Vehicle</td> <td>Details</td> <td><button>Edit</button> <button>Delete</button></td> </tr> <tr> <td>2</td> <td>Van</td> <td>Vehicle</td> <td>Details</td> <td><button>Edit</button> <button>Delete</button></td> </tr> <tr> <td>3</td> <td>Motherboard</td> <td>Computers</td> <td>Order</td> <td><button>Edit</button> <button>Delete</button></td> </tr> <tr> <td>5</td> <td>Motorcycles</td> <td>Vehicle</td> <td>Details</td> <td><button>Edit</button> <button>Delete</button></td> </tr> </tbody> </table> <p>A modal dialog box is overlaid on the page, displaying the message 'localhost says Update successful' with an 'OK' button.</p>	ID	Name	Category Name	Order_Type	Buttons	1	Car	Vehicle	Details	<button>Edit</button> <button>Delete</button>	2	Van	Vehicle	Details	<button>Edit</button> <button>Delete</button>	3	Motherboard	Computers	Order	<button>Edit</button> <button>Delete</button>	5	Motorcycles	Vehicle	Details	<button>Edit</button> <button>Delete</button>
ID	Name	Category Name	Order_Type	Buttons																						
1	Car	Vehicle	Details	<button>Edit</button> <button>Delete</button>																						
2	Van	Vehicle	Details	<button>Edit</button> <button>Delete</button>																						
3	Motherboard	Computers	Order	<button>Edit</button> <button>Delete</button>																						
5	Motorcycles	Vehicle	Details	<button>Edit</button> <button>Delete</button>																						
Conclusion	Functioning as intended.																									

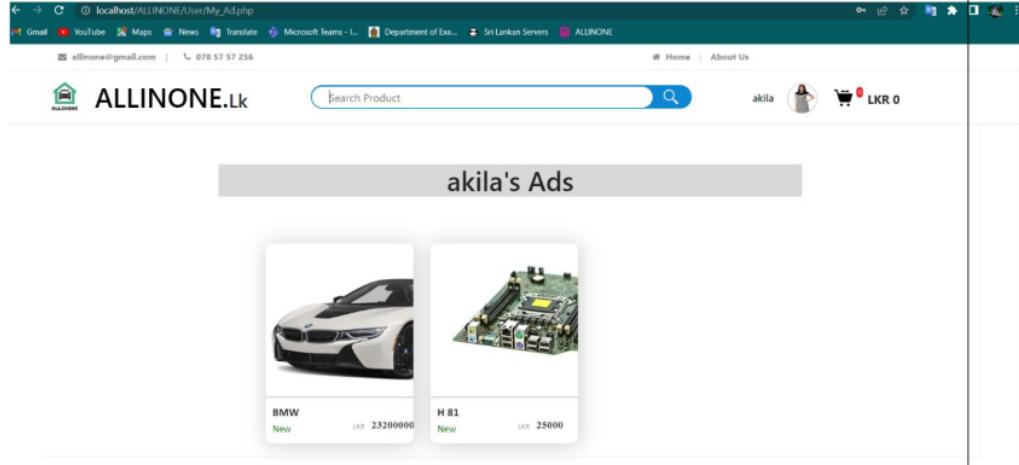
Test case	Test 26																									
Test Objective	Sub Categories (Delete button )																									
Test Data	By clicking the Delete button, the relevant information in the Category menu of the database will be deleted.																									
Expected results	A message will be displayed as Delete successfully. 1																									
Screenshots	 <p>The screenshot shows a web browser window with the URL <code>localhost/ALLINONE/Admin/Admin_Sub_Category.php?id=6&amp;Delete=Delete#</code>. The page title is "ALLINONE". On the left, there's a sidebar with navigation links: Site, Stock, Payments, Category (which is selected), Sub Category, and User. The main content area has a header with "Add Sub Category" and a search bar. Below is a table with the following data:</p> <table border="1"> <thead> <tr> <th>ID</th> <th>Name</th> <th>Category Name</th> <th>Order_Type</th> <th>Buttons</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Car</td> <td>Vehicle</td> <td>Details</td> <td><button>Edit</button> <button>Delete</button></td> </tr> <tr> <td>2</td> <td>Van</td> <td>Vehicle</td> <td>Details</td> <td><button>Edit</button> <button>Delete</button></td> </tr> <tr> <td>3</td> <td>Motherboard</td> <td>Computers</td> <td>Order</td> <td><button>Edit</button> <button>Delete</button></td> </tr> <tr> <td>5</td> <td>Motorcycles</td> <td>Vehicle</td> <td>Details</td> <td><button>Edit</button> <button>Delete</button></td> </tr> </tbody> </table> <p>A modal dialog box is open in the center of the screen, displaying the message "localhost says Delete successful" with an "OK" button.</p>	ID	Name	Category Name	Order_Type	Buttons	1	Car	Vehicle	Details	<button>Edit</button> <button>Delete</button>	2	Van	Vehicle	Details	<button>Edit</button> <button>Delete</button>	3	Motherboard	Computers	Order	<button>Edit</button> <button>Delete</button>	5	Motorcycles	Vehicle	Details	<button>Edit</button> <button>Delete</button>
ID	Name	Category Name	Order_Type	Buttons																						
1	Car	Vehicle	Details	<button>Edit</button> <button>Delete</button>																						
2	Van	Vehicle	Details	<button>Edit</button> <button>Delete</button>																						
3	Motherboard	Computers	Order	<button>Edit</button> <button>Delete</button>																						
5	Motorcycles	Vehicle	Details	<button>Edit</button> <button>Delete</button>																						
Conclusion	Functioning as intended.																									

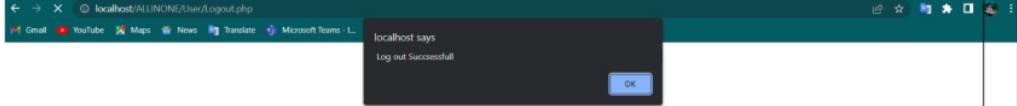
Test case	Test 27																				
Test	Add User																				
Objective																					
Test Data	In Add User, fill the text box correctly, select admin or user and click Add user button, it will be saved in the database table. 1																				
Expected results	A message will be displayed as Add user Successfully.																				
Screenshots	 <p>The screenshot shows a browser window with the URL 'localhost/ALLINONE/Admin/Admin_User.php'. The page title is 'ALLINONE'. On the left, there's a sidebar with links: Site, Stock, Payments, Category, Sub Category, and User. The main content area has a header 'Add User' with a search bar and a 'search' button. Below is a table with columns: Id, User Information, Type, Tel_no, and Buttons. The table contains three rows of data:</p> <table border="1"> <thead> <tr> <th data-bbox="538 1208 559 1216">Id</th> <th data-bbox="674 1208 757 1216">User Information</th> <th data-bbox="904 1208 926 1216">Type</th> <th data-bbox="1029 1208 1064 1216">Tel_no</th> <th data-bbox="1176 1208 1222 1216">Buttons</th> </tr> </thead> <tbody> <tr> <td data-bbox="538 1241 559 1252">13</td> <td data-bbox="620 1241 734 1269">  akila akila@gmail.com         </td> <td data-bbox="904 1241 926 1252">User</td> <td data-bbox="1029 1241 1086 1252">0785674523</td> <td data-bbox="1176 1241 1222 1252"> <a href="#">Edit</a> <a href="#">Delete</a> </td> </tr> <tr> <td data-bbox="538 1288 559 1296">14</td> <td data-bbox="620 1288 734 1313">  yash yash@gmail.com         </td> <td data-bbox="904 1288 926 1296">Admin</td> <td data-bbox="1029 1288 1086 1296">0784565767</td> <td data-bbox="1176 1288 1222 1296"> <a href="#">Edit</a> <a href="#">Delete</a> </td> </tr> <tr> <td data-bbox="538 1341 559 1349">15</td> <td data-bbox="620 1341 734 1366">  ramitha ramitha@gmail.com         </td> <td data-bbox="904 1341 926 1349">User</td> <td data-bbox="1029 1341 1086 1349">0786785645</td> <td data-bbox="1176 1341 1222 1349"> <a href="#">Edit</a> <a href="#">Delete</a> </td> </tr> </tbody> </table>	Id	User Information	Type	Tel_no	Buttons	13	 akila akila@gmail.com	User	0785674523	<a href="#">Edit</a> <a href="#">Delete</a>	14	 yash yash@gmail.com	Admin	0784565767	<a href="#">Edit</a> <a href="#">Delete</a>	15	 ramitha ramitha@gmail.com	User	0786785645	<a href="#">Edit</a> <a href="#">Delete</a>
Id	User Information	Type	Tel_no	Buttons																	
13	 akila akila@gmail.com	User	0785674523	<a href="#">Edit</a> <a href="#">Delete</a>																	
14	 yash yash@gmail.com	Admin	0784565767	<a href="#">Edit</a> <a href="#">Delete</a>																	
15	 ramitha ramitha@gmail.com	User	0786785645	<a href="#">Edit</a> <a href="#">Delete</a>																	
Conclusion	Functioning as intended.																				

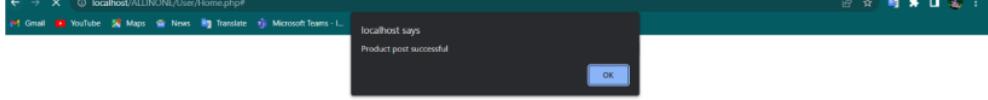
<b>Test case</b>	Test 28
<b>Test</b>	Empty Field validation (Add User )
<b>Objective</b>	
<b>Test Data</b>	When the text box is empty in Add user, it does not allow to add to the database
<b>Expected results</b>	A message is displayed as an empty text box
<b>1 Screenshots</b>	
<b>Conclusion</b>	Functioning as intended.

Test case	Test 29
Test	Edit button (User Insert Page )
Objective	
Test Data	By correctly updating the information in the Update User Details and clicking the update button, the information in that table in the database will be updated.
Expected results	A message as Update successfully will be displayed. 1
Screenshots	
Conclusion	Functioning as intended.

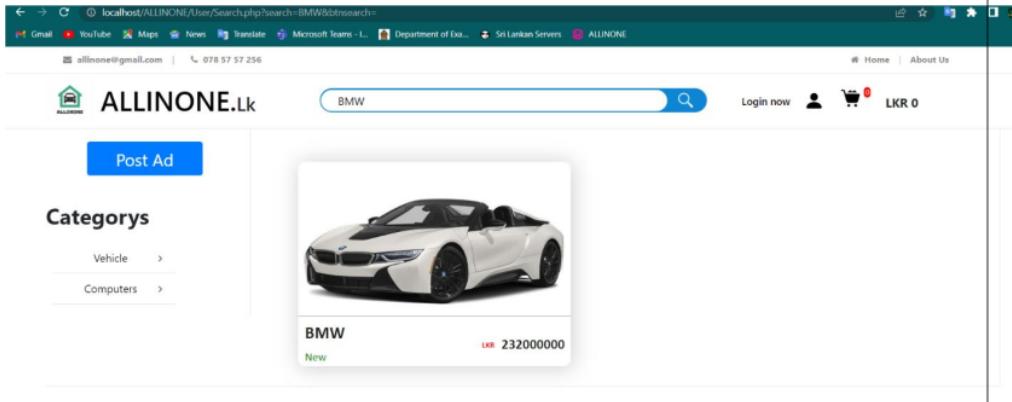
Test case	Test 30
Test	Delete button (User Insert Page )
Objective	
Test Data	By clicking the Delete button, the relevant information in the User Details menu of the database will be deleted.
Expected results	A message will be displayed as Delete successfully. 1
Screenshots	
Conclusion	Functioning as intended.

Test case	Test 31
Test	User login Image
Objective	
Test Data	After the user logs into the user's home page and clicks on the image in front of the user name, the user can see all the Adds that he has entered the website.
Expected results	My Adds login successfully
1 Screenshots	 <p>The screenshot shows the user's profile page on the ALLINONE.Lk website. At the top, there is a navigation bar with links like Gmail, YouTube, Maps, News, Translate, Microsoft Teams, Department of Exa..., Sri Lankan Servers, and ALLINONE. Below the navigation bar, the user's email (ellinone@gmail.com) and phone number (078 57 57 256) are displayed. The main content area features a header "akila's Ads". Below this, there are two ads listed: a BMW car and a computer motherboard. The BMW ad includes the text "BMW New LKR 23200000". The motherboard ad includes the text "H 81 New LKR 25000".</p>
Conclusion	Functioning as intended.

<b>Test case</b>	Test 32
<b>Test</b>	User login Image
<b>Objective</b>	
<b>Test Data</b>	After logging <b>in</b> to <b>the</b> user's home page and clicking on the image in front of the user name, the user will be given the opportunity to log out of the user page.
<b>Expected results</b>	A message <b>will be displayed as</b> log out successful  1
<b>Screenshots</b>	 A screenshot of a web browser window. The address bar shows 'localhost/ALLNONE/User/Logout.php'. The main content area displays a dark gray modal dialog box with white text. The dialog says 'localhost says' at the top, followed by 'Log out Successful' in the center, and an 'OK' button at the bottom right. The background of the browser shows other tabs and links like Gmail, YouTube, Maps, News, Translate, and Microsoft Teams.
<b>Conclusion</b>	Functioning as intended.

<b>Test case</b>	Test 33
<b>Test</b>	User page ( post Ad )
<b>Objective</b>	
<b>Test Data</b>	After filling the text field in the post ad page correctly, it is saved in a table in the database. It appears as an add on the home page in the relevant category.
<b>Expected results</b>	A message <sup>1</sup> will be displayed as Product post successful.
<b>Screenshots</b>	 <p>The screenshot shows a browser window with the URL 'localhost/ALLNONE/User/Home.php#'. A black modal dialog box is centered on the screen, displaying the text 'localhost says' and 'Product post successful' below it. An 'OK' button is visible at the bottom right of the dialog. The background of the browser shows various tabs and links.</p>
<b>Conclusion</b>	Functioning as intended.

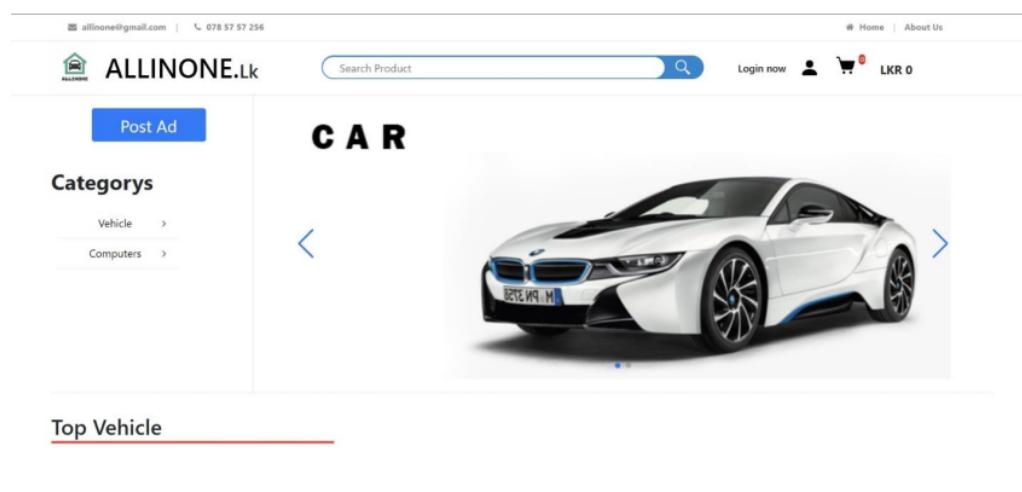
Test case	Test 34
Test	User page ( post Ad )
Objective	
Test Data	When posting an add, without completing the text field, it is not given an opportunity to enter it into the database
Expected results	A message is displayed as an empty text box
1 Screenshots	
Conclusion	Functioning as intended.

Test case	Test 35
Test	search bar
Objective	
Test Data	All products in <b>the</b> database can be easily found by entering the name of the product in the All products in the database can be easily found by entering the name of the product in the search bar.
Expected results	Searched product is loaded successfully.
Screenshots	 <p>The screenshot shows a web browser displaying the ALLINONE.LK website. The URL in the address bar is 'localhost/ALLINONE/User/Search.php?search=BMW&amp;ofnsearch='. The page header includes the ALLINONE logo, a search bar containing 'BMW', and navigation links for Home, About Us, Login now, and a shopping cart icon showing 'LKR 0'. On the left, there's a 'Post Ad' button and a 'Categories' section with links to Vehicle and Computers. The main content area displays a white BMW i8 convertible with the text 'BMW New LKR 232000000' below it.</p>
Conclusion	Functioning as intended.

- **Test conclusion.**

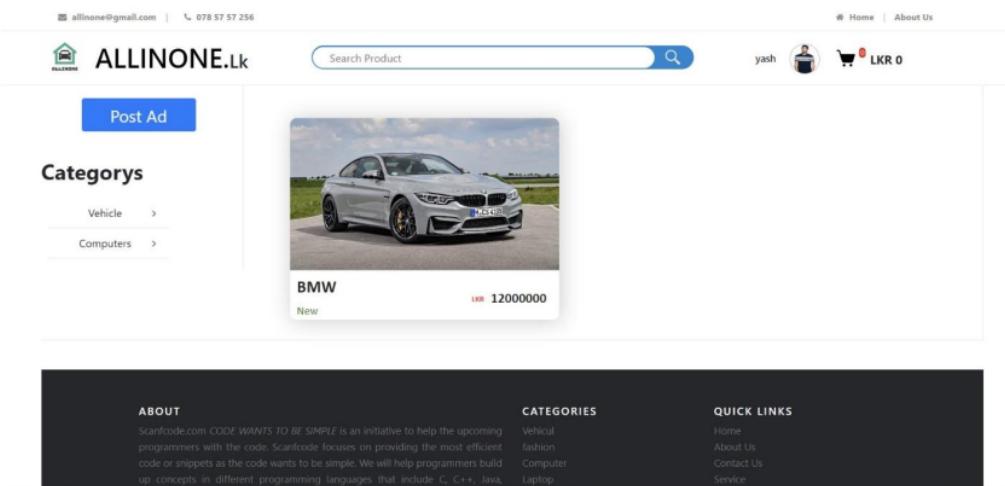
## 2. User Manual

**Home Page.**



This is our main home page, and it is designed to be easy to use for any visiting customer. Here on the left side column, we can see the current categories and a search bar has been used in such a way that it is easy for the customers who come from above to find the required products. Below you will find the vehicles and computers we have in excellent condition, and you must log in before buying the items here. The facility to log in is also provided to the left of the search bar. And the post ad option here has also provided the facility to post ads on our web page. Our e-mail address and mobile phone number are also provided at the top left so that customers can discuss their issues with us

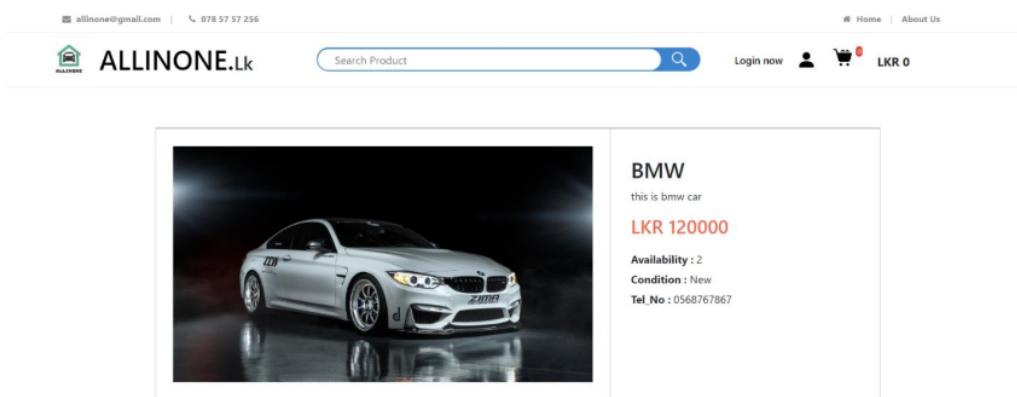
### Product Page.



Here is the product page on our website. This page shows the types of products that are currently available for sale in our system. Also, another page has been included to view all

the information about the respective item separately by using the Details button for each of these items. Information about that page is given below.

### Vehicle Details Page.



Here is the vehicle details page on our website. On this page, there is a photo of the vehicle related to the customer and a brief introduction about the current condition of the vehicle has also been made. It is also easy for the customer to get a rough idea about the vehicle through that short introduction. And on this page, the customer can see the prices for selling the vehicle, the amount of that type of vehicle we currently have, and whether the vehicle is brand new or second hand. Finally, the phone number of the person who has posted the advertisement for the sale of the vehicle has been used and thus the facility of talking about all the information and prices about the vehicle has been provided.

## Computer Details Page.

The screenshot shows a product listing for a motherboard. On the left, there is a large image of the motherboard, which is black with blue and red accents. To the right of the image, the product title is listed as "T33". Below the title is a short description: "thisisasasas". The price is prominently displayed as "LKR 120000". Underneath the price, there is additional product information: "Availability : 5", "Condition : New", and "Tel. No : 0678768787". There is also a "Quantities" input field set to "1" and a "Buy Now" button.

Here is the computer details and computer purchase page on our website. This page has a photo of the computer in question for the customer and a brief introduction about the current condition of the computer. It is easy for the customer to get a rough idea about the computer through that short introduction. And from this page, the customer can check the selling price of the computer, the quantity of that type of computer that we currently have and whether the computer is brand new or second hand. Finally, the phone number of the person who posted the computer sale ad is used, which makes it possible to talk about all the information and prices about the computer. And here the facility of selecting the number of goods to be purchased is also provided. Finally, clicking the Buy Now button will take you to a page with purchase details.

## **Product Payment Page.**

The screenshot displays a product payment interface. On the left, there's a section titled "Products" showing a single item: "T33 Motherboard" (1 unit, LKR 120000). Below this is a "Back to Home" link. On the right, there's a "CARD PAYMENT" form with fields for "Card Holder Name", "Card Number", "MM YY", and "CVV". Below the form, it says "Total : LKR 120000" and features a prominent "PAY NOW" button.

Here is the Product Payment Page of our website. On this page, the name, type, price of the product, and the number of products purchased are displayed on the left and below it, the facility to go back to the home page is provided. Here on the right side is the section provided to enter the card details of the customer purchasing the goods. Here it is allowed to enter the name of the card owner, the card number, the expiry date of the card and the secret number on the back of the card. Below these details are the total prices of the goods to be purchased. Finally, by clicking the Pay Now button, you are given the option to purchase the item.

## **Post Ads.**

**Post Ad**

Select category

Select New or Used

Product Name

Telephone Number

Bank Account Number

Quantity

Price

description

Choose File No file chosen

**Post Ad**

Here is the form that must be filled before posting a new ad on our website. Here first the category should be selected and after selecting it the relevant sub category will appear in the next column and the relevant sub category should be selected from it. After that, it is possible to enter brand new or second hand, product name, mobile phone number of the advertiser, a brief <sup>8</sup> description of the product, the quantity of the product to be sold, the selling price of the product and a picture of the relevant product. After filling all the above details, click the post ad button and the relevant ad will appear on our website.

## **Admin Home Page.**



The screenshot shows a web-based admin interface for managing products. At the top left is a button labeled "Add Products". Below it is a table with five columns: "Product Id", "Product Infor", "Available", "Price", and "Buttons". The table contains three rows of data:

Product Id	Product Infor	Available	Price	Buttons
39	 T33 UserID - 13	5	120000	<button>Delete</button>
40	 bikre UserID - 13	20	12365	<button>Delete</button>
41	 bikregv UserID - 13	59	15622	<button>Delete</button>

Here is the page that appears after logging in as admin. Here you can see the quantity of all the products we currently have. Also, there is an opportunity to add new products and if there is an unwanted product, the admin is also given the facility to delete it. And admin can logout by clicking the button on the left side.

## Admin Category Management Page.

The screenshot shows a user interface for managing categories. At the top left is a button labeled "Create Category". Below it is a table with the following data:

id	Name	description	Buttons
1	Vehicul	Add Vehicle are includes	<button data-bbox="1114 703 1155 724">Edit</button> <button data-bbox="1188 703 1228 724">Delete</button>
2	Computers	Add Computers are includes	<button data-bbox="1114 756 1155 777">Edit</button> <button data-bbox="1188 756 1228 777">Delete</button>

Here is the admin management category page. Admin can add new categories using Create Category. And you can update the categories here through the edit button. Similarly, the admin is provided with the ability to remove categories through the delete button.

## Admin User Page.

The screenshot shows the 'User' section of the Admin User Page. The left sidebar lists categories: Site, Stock, Payments, Category, Sub Category, and User. The User section has a table with the following data:

ID	User Information	Type	Tel_no	Buttons
13	akila akila@gmail.com	User	0785674523	Edit Delete
14	yash yash@gmail.com	Admin	0784565767	Edit Delete
15	ramitha ramitha@gmail.com	User	0786785645	Edit Delete

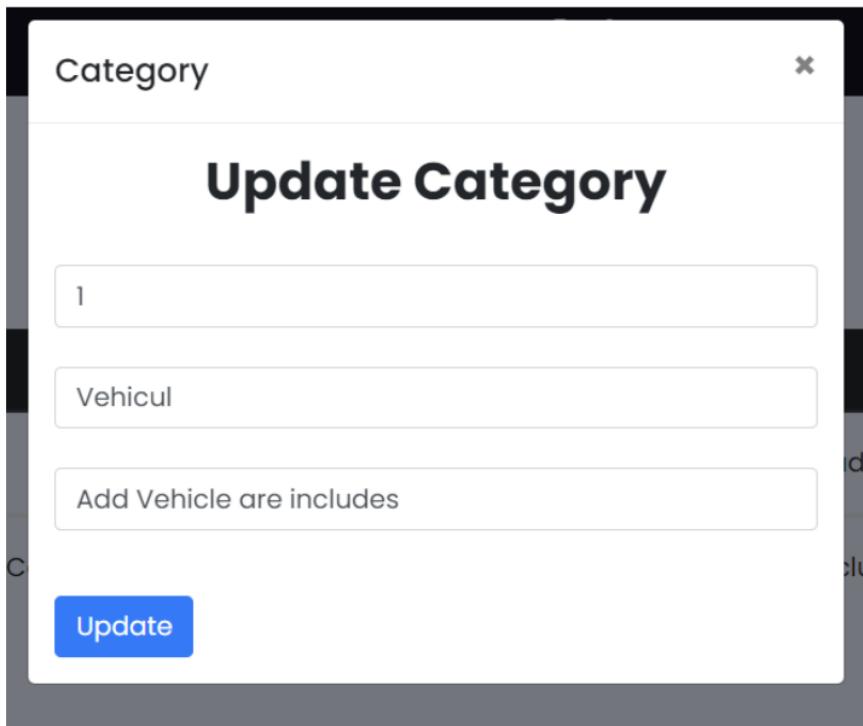
Here appears the admin's user page and here you can add a new user by adding user. Also, it is possible to update the information of a user here through the edit button. Here you can remove an unwanted user using the delete button. Here, the user's name and email address can be found in the user information. It is possible to know whether the user is an admin or a user through the user type.

## Admin Subcategory page.

Add Sub Category		Search	search	
ID	Name	Category Name	Order_Type	Buttons
1	Car	Vehicul	Details	<button>Edit</button> <button>Delete</button>
2	Van	Vehicul	Details	<button>Edit</button> <button>Delete</button>
3	Motherboard	Computers	Order	<button>Edit</button> <button>Delete</button>
5	Motobike	Vehicul	Details	<button>Edit</button> <button>Delete</button>

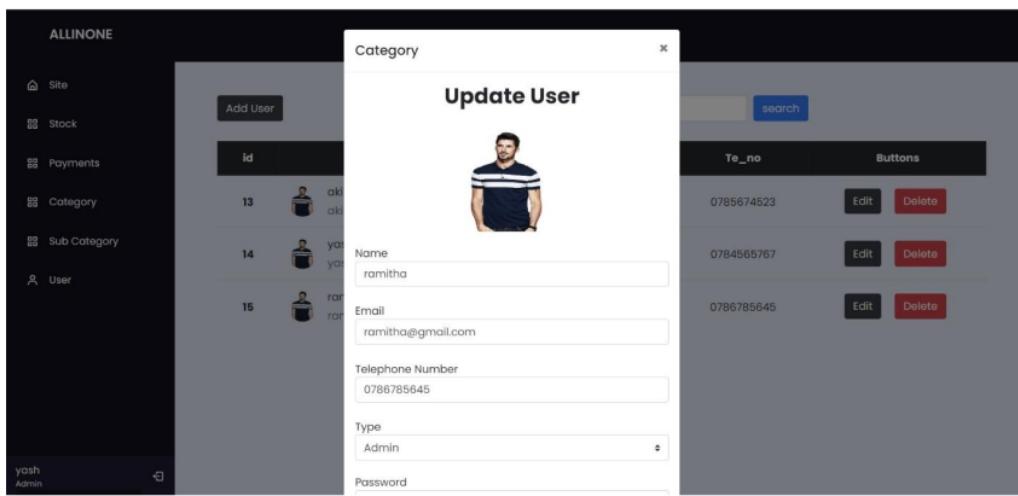
From this page you can see a summary of the products sold on their website. As well as All Payment shows the total amount of the goods sold on the website. It has made it very easy for the admin to calculate the daily profit and loss. And here, the admin can know who the customer is who buys the goods from the website from the customer's name, what goods he bought from the product name, the amount of goods bought from the content and the total profit received by the website from that customer from the total.

**Update Category Page.**



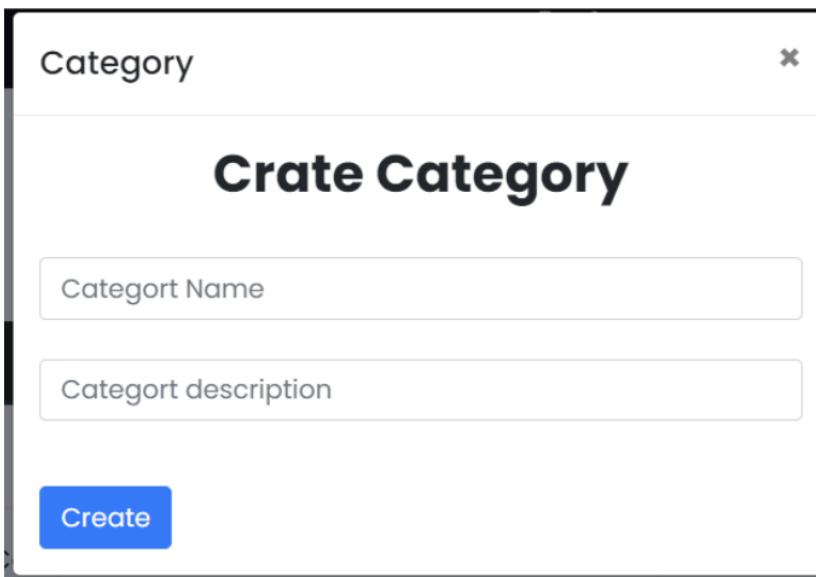
The Update Category Page will appear here, and the Category ID will be there first. After that is the category name and at the end is the category description. Here the admin can update the category by filling the above information correctly and clicking on the update button.

## Update User Page.



Here you will see the page for updating a user. Here the admin can update the information of a user. Here, the user's name, email address, phone number, user type and password are updated respectively, and the information is updated by clicking the update button.

### Create Category.



The image shows a screenshot of a 'Create Category' form. At the top left is the title 'Category' and at the top right is a close button (an 'x'). Below the title is the main heading 'Create Category' in a large, bold, black font. There are two input fields: one for 'Categori Name' and another for 'Categori description'. Both fields have a light gray background and rounded corners. At the bottom left of the form is a blue rectangular button with the word 'Create' in white text. The entire form is enclosed in a thin gray border.

Here is the Create Category page. Here, by entering the name of the category and the description of the category and clicking on the create button, the admin can add a new category to the website. 20

**Create Subcategory.**

Sub Category ×

## Add Sub Category

Select Category

Sub Categori Name

Select order Type

**Create**

Seen here is the Create Subcategory form. First select the category and after that you can enter a related subcategory. Admin can add subcategories to the website by entering the information like this and clicking on the Create button.

### Add User.

The screenshot shows a modal window titled "Add User" within a larger "Users" interface. The "Add User" window contains the following fields:

- User Name
- Email
- Telephone Number
- Password
- Admin (a dropdown menu showing "Admin")
- Choose File (with the message "No file chosen")
- A large blue "Add User" button

Here appears the form to be filled when a user enters the website. Here, the admin can add a new user to the web application by adding the user's name, email address, mobile phone number, password, user type and a picture of the user respectively and clicking on the add user button. 10

**Login Page.**

The image shows a login interface with a light gray background and a white rounded rectangular input field. At the top center, the word "Login" is written in a large, bold, black serif font. Below it, there are two buttons: "Login" in a smaller black font on a white button, and "Sign up" in white text on a red button. Inside the white input field, there are two lines for entering information. The first line is labeled "Enter Email" and the second is labeled "Enter Password", both in a small gray font. Below the input field is a red button with the word "Login" in white. At the bottom of the input field, there is a smaller button labeled "Reset" in a gray font.

Here is the form that a user or an admin must fill when entering this web page. Here first the email address must be entered and finally the user or admin can log in by entering the user's password and clicking on the login button. Also, if the information entered here is incorrect, click the reset button to correct it, all the information will be deleted, and the correct information can be entered again.

**Signup Page.**

Sign up

Login Sign up

User Name

Email

Telephone Number

Password

Choose File No file chosen

Signup

Reset

Seen here is the signup form. To sign up a new user, after entering the name, email address, phone number, password, and a photo of the user, you can add a new user to the website by clicking the signup button. If a mistake is made while entering the information here, by clicking the reset button, everything will be erased, and it can be entered correctly again.

10

## Chapter 06

- **Limitations for the project.**

While creating this project we had to work under various constraints. There, as we had imagined, it was not possible to complete the construction work properly and completely. Here are some functions that we could not add for this project. We couldn't add those functions to this project because there are big restrictions for those functions.

Accordingly, the functions we had to submit to the restrictions are given below,

**Security** - Here after the creation of our website, we have made the people entering the website to send a message to his email address that he has logged in after registering in the website, but we were not able to do that.

**When buying goods** - We have provided facilities to buy the goods available in our website, but we could not provide the necessary facility for payment through the website. This is due to the fact that permission has to be obtained from many authorized institutions such as currency Institutions for that purpose. Due to the current situation in the country, we could not do it properly.

**Email Verification** - Here we are expected to implement this facility. We were hoping to provide a facility to send a message to his email address to inform him of the details of whether any item was put up for sale on the website after visiting the website, but we were unable to do so. This is due to the fact that we do not currently have the knowledge about the mining process.

**Live Message Facility** - By implementing this, we hoped to facilitate any person visiting the website to solve his problem by chatting through the message facility that we were hoping to provide at that time if any problem arose. But due to the

fact that we currently do not have an understanding of the time and how to use it properly, we had to submit to the limitations here.

- **Lesson Learned.**

During the creation of this project, all of our team members were able to gain experience and knowledge from various sources through this project. In this way, the lessons that all of us were able to get from this project are given below.

- Start is small, but then it extends:

When all of us start working on this project, the creation of the project is started with a very simple version with very few features, interfaces, and functionality, but while continuing the creation of the project, we need to identify new features to be added and replace those features for our project. We could. That is, what we were able to learn here is that when starting any activity, it should start very simple and develop it further while continuing to develop.

- Adding logging and error handling in early:

When we start the development of this project, the first thing we do is add the necessary things for logging and error handling. What we have learned here is that when creating a system, when a bug appears, that bug should be fixed.

- Parts must be inspected before the project is completed:

Another thing we learned while working on this project was that while we were developing the system, some parts of the system were not working properly due to the use of new interfaces, different code blocks and different functions. Therefore, tests were always carried out to see if the system was working properly. It was able to identify the errors in the system early.

- Time Management:

One more thing that we were able to learn while doing this project is that all of us were able to manage the time since the development of the project was done by everyone in our group. All of us

managed the part assigned to us properly and carried out the construction work according to a time-honored plan and thus we were able to get a clear understanding about time management.

- **Future recommendation.**

We completed this project but due to some limitations we could not develop this system in our mind. That is because some functions could not be added to the system. But we hope to develop this website further. We all hope to complete the website by adding functions that we are currently unable to add to this system.

The main thing we hope to do is to get a domain for the website and release it to the internet so that anyone in the world can use it. Also, we hope to develop this website system as a **mobile phone application for mobile phones** in the future. Also, we hope to develop and complete the **payment section** that we are currently unable to include in the system. Any person who comes to the website through the internet is given the necessary space to buy the goods in the website.

And we hope to introduce **delivery methods** so that after purchasing the goods through the website, the goods can be taken to the customer's place and delivered. And in the future development of this website, we hope to introduce a way to make **live messages** into the website. We hope to develop the website in such a way that the users who come to the website can solve any problem through a message. Also, we hope to add the **SMS verification** facility to the system in the future to confirm the purchase of any item.

## **References**

**There are no sources in the current document.**

Computing Project - Ramitha 20199209 - Sameen 20198986-  
Akalanka 20199208 - Akila - 20197288- Yashoda 20199207  
.docx

---

ORIGINALITY REPORT

---



PRIMARY SOURCES

---

1	Submitted to University of Wales Institute, Cardiff Student Paper	6%
2	Submitted to University of Salford Student Paper	<1 %
3	Submitted to Lebanese International University Student Paper	<1 %
4	Submitted to NCC Education Student Paper	<1 %
5	1000projects.org Internet Source	<1 %
6	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1 %
7	Submitted to St. Patrick's College Student Paper	<1 %

---

8	Internet Source	<1 %
9	slideplayer.com Internet Source	<1 %
10	patents.patsnap.com Internet Source	<1 %
11	Submitted to University of Greenwich Student Paper	<1 %
12	documents.mx Internet Source	<1 %
13	Submitted to Majan College Student Paper	<1 %
14	Submitted to Sunway Education Group Student Paper	<1 %
15	Submitted to University Tun Hussein Onn Malaysia Student Paper	<1 %
16	Submitted to University of Sheffield Student Paper	<1 %
17	Submitted to Colorado Technical University Online Student Paper	<1 %
18	docplayer.net Internet Source	<1 %

19	ramtech.colostate.edu Internet Source	<1 %
20	ia801708.us.archive.org Internet Source	<1 %
21	www.joiku.com Internet Source	<1 %
22	www.uk-shopping-warehouse.co.uk Internet Source	<1 %

---

Exclude quotes Off

Exclude bibliography Off

Exclude matches Off