



CSE470: Software Engineering

Project Report

Project Title: Inventory Management System

Group No: 5, CSE470 Section: 02, Summer 2023	
ID	Name
20301137	Farhan Faruk
20301263	Emam Hasan
20301370	Nafisa Ahmed Progga
19301074	Sharmin Akter Keya

Table of Contents

Section No	Content	Page No
1	Introduction	2
2	Functional Requirements	3-4
3	User Manual	4-22
4	Frontend Development	23-34
5	Backend Development	35-65
6	Technology (Framework, Languages)	66
7	Github Repo Link	66
8	Individual Contribution	66-67

Introduction

In the contemporary business landscape, effective management of inventory holds paramount importance. This is where an Inventory Management System (IMS) steps in, overseeing the intricate process of monitoring, storing, and replenishing goods and resources. This facet is indispensable for businesses, as it guarantees a balance between stock availability and demand, while preventing both surplus stock and wastage. The scope of inventory management goes beyond mere item count, extending to smart reorder decisions, pacing adjustments, and resource optimization.

Our IMS offers a range of benefits. It provides a streamlined means to monitor inventory levels and their respective locations. Additionally, it tracks orders, dispatches, and deliveries. When supplies run low, the system facilitates swift reordering. Comprehensive insights into inventory, sales, and expenses are easily accessible through generated reports.

Functioning as the central hub for inventory control, our IMS encompasses a spectrum of tasks. It goes beyond mere stock monitoring by seamlessly tracing items from procurement to distribution, ensuring real-time visibility across the supply chain. This visibility mitigates the risk of stock shortages that can lead to missed sales and discontented customers. Moreover, it empowers businesses to forecast demand more accurately and fine-tune procurement strategies accordingly.

In essence, our inventory management system project assumes a pivotal role in efficient item management. It guarantees that users consistently possess the required items at the right junctures, rescuing them from surpluses and deficits. Amidst a complex network of customers and suppliers, our system enables businesses to operate effortlessly and ensure seamless operations.

Functional Requirements

1. Track:

Within the administrative dashboard, the system will offer insights into diverse metrics. These include the count of active suppliers, the number of new customers who have engaged within the last 7 days, the total active customer count, the aggregate units count, category count, product count, approved purchase count, pending purchase count, approved invoice count, and pending invoice count.

2. Suppliers Overview:

The system will provide a comprehensive display of detailed information about all registered suppliers. Furthermore, it will allow the addition of new suppliers, modification of supplier details, and the option to remove suppliers.

3. Units Overview:

This section will encompass a comprehensive depiction of all unit-related particulars. The system will have the capability to establish new units, update existing unit information, and remove specific units.

4. Category Overview:

A holistic overview of categories will be accessible through this feature. The system will enable the inclusion of fresh categories, the modification of existing ones, and the removal of categories as needed.

5. Product Overview:

A thorough exploration of product-related information will be facilitated. The system will provide the means to introduce new products, edit existing product particulars, and eliminate products when necessary.

6. Purchase Overview:

Comprehensive visibility into purchase details will be available. This entails the capacity to introduce new purchases, authorize or remove existing ones, and generate daily purchase reports in PDF format, which can be printed.

7. Order Overview:

The system will provide an all-encompassing perspective on order details. This encompasses the capability to create new orders, authorize or eliminate existing orders, aggregate an order list for printing, and generate daily order reports in PDF format, suitable for both examination and printing.

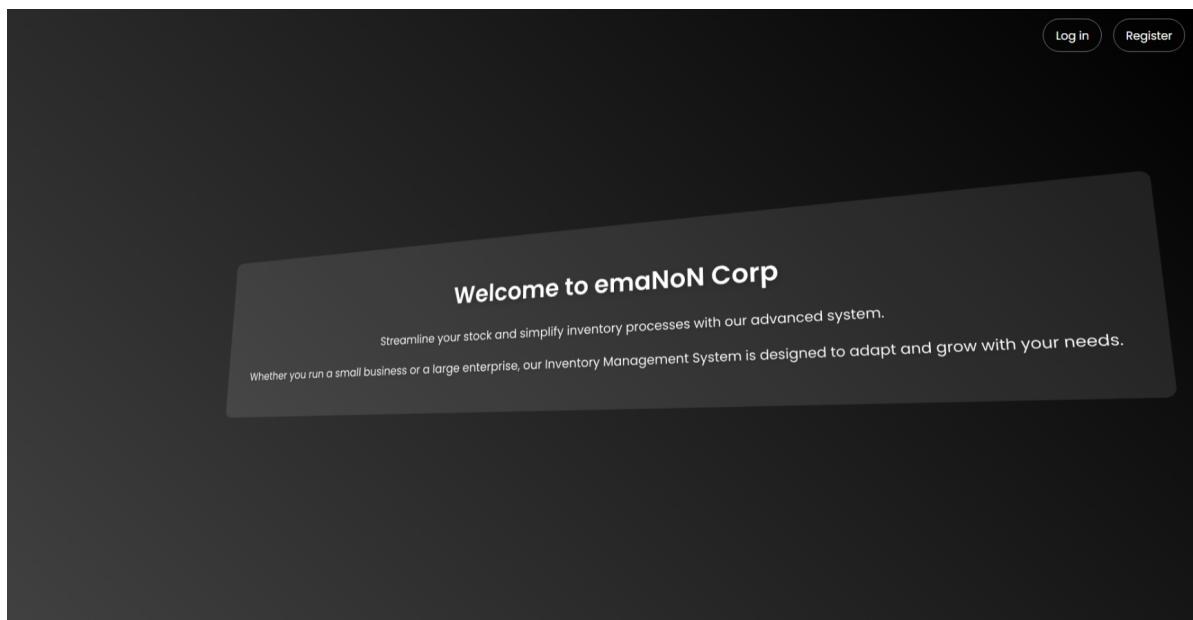
User Manual

There is one major portion of our web application which is the admin interface.

There are four major modules for the admin side.

1. Home Page:

0.0 Landing Page:



1.1. Authentication/ Registration:

- Upon initial access to the web application, the admin will be presented with the login page. In the event that the admin possesses an existing account, successful login will necessitate the input of both a username and a password.

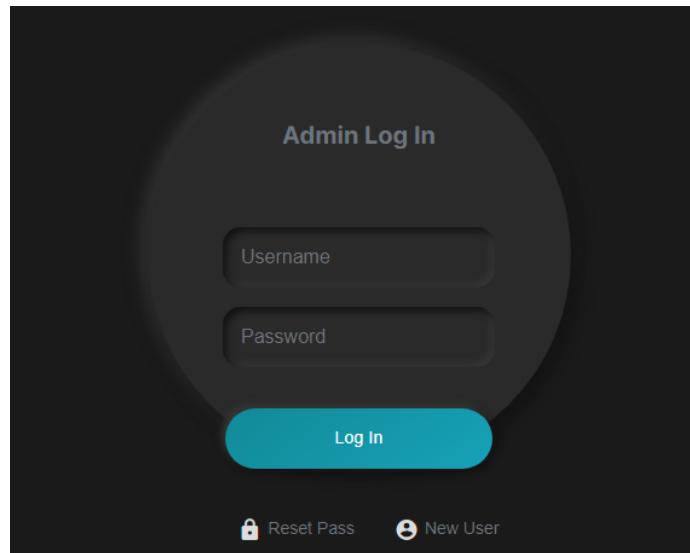


Figure 1: Admin login

- In case the admin lacks an account, there is a requirement for registration. To gain entry to the web application, the admin must select the "Create an Account" choice. The registration segment is depicted in Figure 2. This segment mandates the provision of a name, username, email, and password.

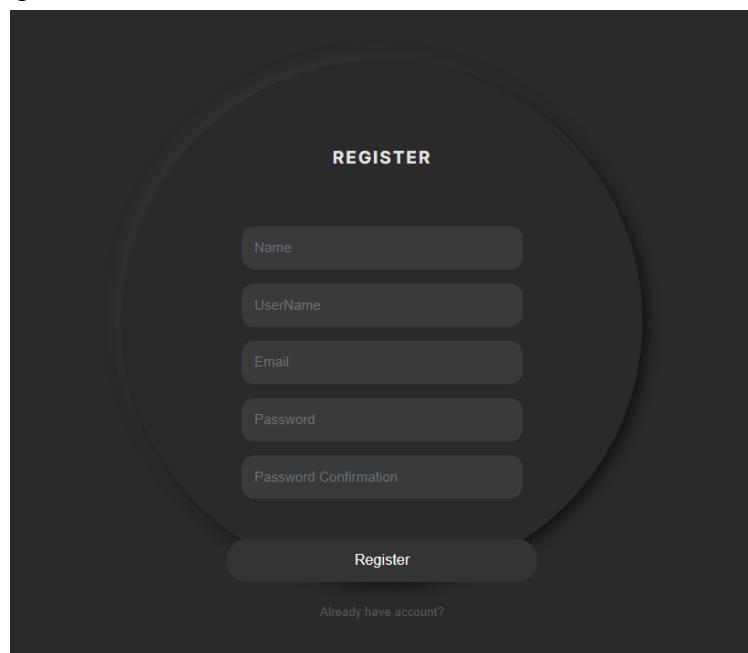


Figure 2: Registration

1.2. Track:

This page showcases specific information (as depicted in Figure 3).

- Total Suppliers: This section presents the overall count of suppliers that have been added.
- Total Categories: This segment showcases the cumulative count of established categories.
- Total Units: Lastly, you can view the total quantity of units that have been created.
- Approved Purchases: This section displays the quantity of purchases that have been given approval.

- Pending Purchases: This section indicates the number of purchases that are currently awaiting approval.
- New Customers: Within this section, the count of newly registered customers over the past week is showcased.
- Total Customers: This section provides an overview of the total number of customers that have been added.
- Pending Orders: This area displays the current count of orders that are awaiting approval.
- Approved Orders: Additionally, you can see the total number of orders that have been approved.

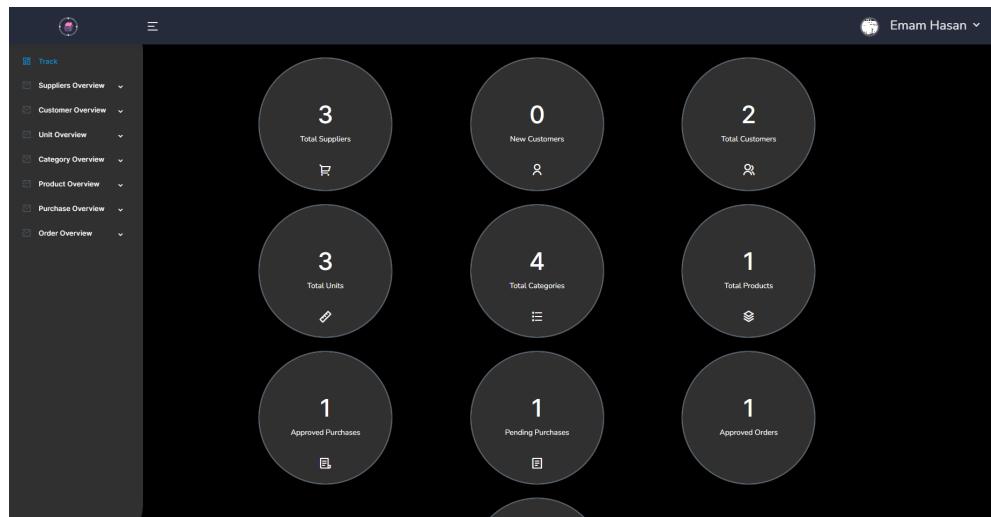


Figure 3: Track

- The admin has the option to log out by selecting the logout button as depicted in Figure 4. This button is situated at the uppermost right corner.

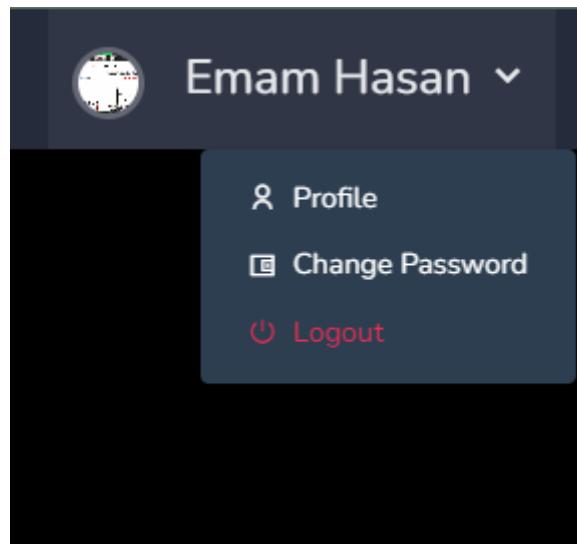


Figure 4: Logout button

- On the page, there exists a labeled choice titled "Profile." Upon choosing this option, users are directed to the admin's profile page, facilitating convenient modifications to their profile particulars. Following any adjustments, users are required to click an "update profile" button to store the modifications.

The screenshot shows the 'Edit Profile Page' interface. It features a dark-themed form with white input fields. The fields include:

- Name:** Emam Hasan
- User Email:** emamhasan716@gmail.com
- UserName:** emamhasan_
- Profile Image:** A small thumbnail image of a document or chart, with a 'Choose File' button and a message 'No file chosen'.

A prominent green 'Update Profile' button is located at the bottom left of the form area.

Figure 5: Edit Profile

- Moreover, the segment named "Change Password," visible in Figure 4, permits users to navigate to Figure 6. In Figure 6, the admin is provided with the choice to revise their password. This can be accomplished by completing the requisite form fields and subsequently activating the "Change Password" button to apply the changes.

The screenshot shows the 'Change Password Page' interface. It features a dark-themed form with white input fields. The fields include:

- Old Password:** (empty field)
- New Password:** (empty field)
- Confirm Password:** (empty field)

A prominent green 'Change Password' button is located at the bottom left of the form area.

Figure 6: Change Password Page

1.3. Supplier Overview:

Within the depicted all-supplier page, denoted as Figure 7, numerous controllers are present.

Sl	Name	Mobile Number	Email	Address	Action
1	Techland	789845	rtecg@gov.nd	Agargaon	
2	Startech	0000111	startecg@gov.nd	Agargaon	
3	BRAC Chicken Farm	404	nai@gmail.com	BRACU UB6	

Showing 1 to 3 of 3 entries

Figure 7: All Supplier

- Edit Button: This function facilitates the modification of supplier details. Upon selecting this button, the admin is directed to a page for adding a new supplier. By clicking the "update supplier" button, the admin can then proceed to revise the supplier information (Illustrated in Figure 8).

Supplier Name	Techland
Supplier Mobile	789845
Supplier Email	rtecg@gov.nd
Supplier Address	Agargaon

Figure 8: Supplier Update

- Delete Button: This option facilitates the removal of a supplier.
- Search Bar: This feature assists in locating a particular supplier (Refer to Figure 7).
- Add Supplier Button: This button directs the admin to the "Add Supplier" page.

To include a new supplier, the process involves entering the supplier's name, mobile number, email address, and supplier address. Subsequently, clicking the "Add Supplier" button will store the supplier's details.

The screenshot shows a dark-themed mobile application interface titled "Add Supplier Page". It contains four text input fields: "Supplier Name", "Supplier Mobile", "Supplier Email", and "Supplier Address", each with a corresponding white input box. Below these fields is a green rectangular button labeled "Add Supplier". The entire form is set against a dark background with rounded corners.

Figure 9: Add Supplier Page

1.4. Unit Overview:

On the all-unit page depicted in Figure 10, various controls are present.

The screenshot shows a dark-themed mobile application interface titled "UNIT ALL". At the top right is a blue button labeled "Add Unit". Below it is a section titled "Unit All Data" with a "Show 10 entries" dropdown and a "Search:" input field. The main area is a table with three rows of data:

Sl	Name	Action
1	Meter	
2	KG	
3	PCs	

At the bottom left, it says "Showing 1 to 3 of 3 entries". On the bottom right are navigation icons for back, forward, and page number (1).

Figure 10: Unit All

- **Edit Button:** This control enables the modification of a unit. Selecting this button directs the admin to the unit editing page. Clicking the "update unit" button afterward allows the admin to make updates to the unit (Depicted in Figure 10).



Figure 11: Update Unit

- Delete Button: This option facilitates the removal of a unit.
- Search Bar: This feature assists in locating a particular unit (Refer to Figure 10).
- Add Unit Button: This button directs the admin to the "Add Unit" page.
- To include a new unit, the process involves entering the unit's name (As shown in Figure 12).
- Subsequently, clicking the "Add Unit" button will store the unit's details.



Figure 12: Add Unit

2. Product & Customer:

2.1. Product Overview:

There are several controllers on the all-product page shown in Figure 13. They are-

Product All Data					
Show 10 entries	Search:				
Sl	Name	Supplier Name	Unit	Category	Action
1	Realyou LK93 Headphone	Startech	PCs	Headphone	 

Showing 1 to 1 of 1 entries

Figure 13: Product All

- Edit Button: This control facilitates the modification of product details. When selected, this button guides the admin to the product editing page. By clicking the "update unit" button, the admin can proceed to edit the product details (Illustrated in Figure 14).

Edit Product Page

Product Name	Realyou LK93 Headphone
Supplier Name	Startech
Unit Name	PCs
Category Name	Headphone

Update Product

Figure 14: Edit Product Page

- Delete Button: This option facilitates the removal of a product.
- Search Bar: This feature assists in locating a particular product (Refer to Figure 13).
- Add Product Button: This button directs the admin to the "Add Product" page.

To include a new product, the process entails entering the product name, selecting supplier name, unit name, and category name from dropdown menus (As shown in Figure 15). Subsequently, clicking the "Add Product" button will store the product's details.

Add Product Page

Product Name

Supplier Name Open this select menu

Unit Name Open this select menu

Category Name Open this select menu

Add Product

Figure 15: Add Product Page

2.1. Customer Overview:

There are several controllers on the all-customer page shown in Figure 16. They are-

CUSTOMER ALL

Add Customer

Customer All Data

Show 10 entries Search:

SL	Name	Customer Image	Email	Address	Action
1	Farhan Faruk		farhan@gmail.com	CUDS, Chittagong University	
2	Emam Hasan		emamhasan716@gmail.com	Moghazar, Dhaka	

Showing 1 to 2 of 2 entries

Figure 16: Customer All

- Edit Button: This function facilitates the modification of customer details. Upon selection, this button guides the admin to the customer editing page. By subsequently clicking the "update customer" button, the admin can proceed to revise the customer details (Depicted in Figure 17).

Edit Customer Page

Customer Name	Farhan Faruk
Customer Mobile	0111115554
Customer Email	farhan@gmail.com
Customer Address	CUDS, Chittagong University
Customer Image	<input type="button" value="Choose File"/> No file chosen 

Figure 17: Edit Customer Page

- Delete Button: This option facilitates the removal of a customer.
- Search Bar: This feature assists in locating specific customer details (Refer to Figure 16).
- Add Customer Button: This button directs the admin to the "Add Customer" page.

To include a new customer, the process entails entering the customer's name, mobile number, email, address, and optionally a picture (As shown in Figure 15). Subsequently, clicking the "Add Customer" button will store the customer's details.

Add Customer Page

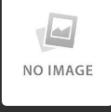
Customer Name	
Customer Mobile	
Customer Email	
Customer Address	
Customer Image	<input type="button" value="Choose File"/> No file chosen  NO IMAGE

Figure 18: Add Customer Page

2.3. Category Overview:

There are several controllers on the all-category page shown in Figure 16. They are-

CATEGORY ALL		Add Category
Category All Data		
Show 10 entries		Search: <input type="text"/>
Sl	Name	Action
1	Headphone	
2	Monitor	
3	Laptop	
4	Mobile Phone	

Showing 1 to 4 of 4 entries < 1 >

Figure 19: Category All

- Edit Button: This functionality facilitates the modification of category names. Upon selection, this button guides the admin to the category editing page. By subsequently clicking the "update category" button, the admin can proceed to alter the category name (Depicted in Figure 20).

Edit Category Page

Category Name	Headphone
Update Category	

Figure 20: Edit Category Page

- Delete Button: This option facilitates the removal of a category.
- Search Bar: This feature assists in locating specific categories (Refer to Figure 19).
- Add Category Button: This button directs the admin to the "Add Category" page.

To include a new customer, it necessitates entering the name of the category (As depicted in Figure 15). Subsequently, by clicking the "Add Category" button, the category name is stored.

The screenshot shows a dark-themed interface titled 'Add Category Page'. It features a single input field labeled 'Category Name' with a placeholder icon. Below the input field is a green rectangular button with the text 'Add Category' in white. The overall design is minimalist and modern.

Figure 21: Add Category Page

3. Purchase Overview:

3.1. All Purchases: This section presents comprehensive information about all purchase details, including fields such as SI, purchase number, date, supplier, category, quantity, product name, status, and action. On the all-purchase page depicted in Figure 22, there exist various controls:

Delete Button: This feature aids in the elimination of purchase details located in the pending section. Upon selecting the delete option, a confirmation step is required. After the user confirms, the respective purchased item will be removed.

The screenshot shows a dark-themed interface titled 'PURCHASE ALL'. At the top right is a blue button labeled 'Add Purchase'. Below it is a search bar with the placeholder 'Search:'. The main area contains a table titled 'Purchase All Data' with the following columns: Sl, Purchase No, Date, Supplier, Category, Qty, Product Name, Status, and Action. The table displays two entries:

Sl	Purchase No	Date	Supplier	Category	Qty	Product Name	Status	Action
1	664554	15-08-2023	Startech	Headphone	14	Realyou LK93 Headphone	Pending	
2	154	14-08-2023	Startech	Headphone	25	Realyou LK93 Headphone	Approved	

At the bottom left, it says 'Showing 1 to 2 of 2 entries'. At the bottom right are navigation icons for back, forward, and page number (1).

Figure 22: Purchase All

Add Purchase Button: Selecting the button leads the admin to a purchase page. For adding a product, the admin provides the date and purchase number, and then opts for choices from pre-existing menus for supplier name, category name, and product name. The "Add More" button enables the admin to input quantity, unit price, and description, with automatic

calculation of the total price. The delete button eliminates added purchases, while the purchase store button records the product details.

Category	Product Name	PSC/KG	Unit Price	Description	Total Price	Action
					0	

Figure 23: Add Purchase

3.2. All Purchases:

Within this section, the admin is presented with a comprehensive view of purchase particulars, encompassing serial numbers (SI), purchase numbers, dates, suppliers, categories, quantities, product names, status, and actions. The page dedicated to all pending purchases, depicted in Figure 24, encompasses several controls, which are as follows:

Purchase Pending: Upon selection, this button guides the admin to a purchase page. For product addition, the admin inputs the date and purchase number and selects options from pre-existing menus for supplier name, category name, and product name. The "Add More" button facilitates the input of quantity, unit price, and description, automatically computing the total price. The delete button eliminates added purchases, while the "Purchase Store" button records the product details.

Pending Button: Clicking this button permits the admin to authorize the pending purchase.

Delete Button: This control aids in the removal of pending purchase details. Upon selection, confirmation is prompted. Upon confirmation, the respective purchased item is eliminated.

PURCHASE ALL								
Purchase All Pending Data								
Show	10	entries						
Search:	<input type="text"/>							
Sl	Purchase No	Date	Supplier	Category	Qty	Product Name	Status	Action
1	664554	15-08-2023	Startech	Headphone	14	Realyou LK93 Headphone	Pending	<input checked="" type="checkbox"/>
Showing 1 to 1 of 1 entries								
< 1 >								

Figure 24: Purchase All Pending Data

3.3. Daily Purchase Report:

Within this page, the admin has the capability to access purchase reports by specifying start and end dates. Upon selecting the "Search" button, they are directed to the page displaying daily purchase reports. Additionally, a download button is provided for obtaining the report (As depicted in Figure 26).

Daily Purchase Report																																																									
Start Date				End Date				Search																																																	
dd----yyyy	<input type="button" value="Calendar"/>	dd----yyyy	<input type="button" value="Calendar"/>	Search																																																					
<input type="button" value="August 2023 ▾"/> ↑ ↓ <table border="1"> <tr> <td>Su</td><td>Mo</td><td>Tu</td><td>We</td><td>Th</td><td>Fr</td><td>Sa</td></tr> <tr> <td>30</td><td>31</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> <tr> <td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td></tr> <tr> <td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td></tr> <tr> <td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td></tr> <tr> <td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>1</td><td>2</td></tr> <tr> <td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr> </table> Clear Today									Su	Mo	Tu	We	Th	Fr	Sa	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9
Su	Mo	Tu	We	Th	Fr	Sa																																																			
30	31	1	2	3	4	5																																																			
6	7	8	9	10	11	12																																																			
13	14	15	16	17	18	19																																																			
20	21	22	23	24	25	26																																																			
27	28	29	30	31	1	2																																																			
3	4	5	6	7	8	9																																																			

Figure 25: Daily Purchase Report

DAILY PURCHASE REPORT

emaNoN Corp

emaNoN Corp:
BRACU, Dhaka
support@emaNoN.com

Purchase Report : 01-08-2023 - 31-08-2023

Sl	Purchase No	Date	Product Name	Quantity	Unit Price	Total Price
1	154	14-08-2023	Realyou LK93 Headphone	25 PCs	250	6250
					Grand Amount:	6250 Tk

Download

Figure 26: Daily Purchase Report Download

4. Order Overview::

4.1. Order History: This page shows invoice info like serial numbers (SI), customer names, invoice numbers, dates, descriptions, and amounts. The Figure 27 all-invoice page has different controllers:

ORDERS

Add Order

Order History

Show 10 entries

Search:

SL	Customer Name	Invoice No	Date	Description	Amount
1	Farhan Faruk	#1	14-08-2023	Thanks for purchasing.	660

Showing 1 to 1 of 1 entries

Figure 27: Order History

- Add Order Button: Upon selecting this button, the admin is directed to a page designed for invoice addition. The admin's initial task involves inputting the date and invoice number, followed by opting for choices from pre-existing menus for category name and product name. The stock field showcases the total count of available products. Further inputs such as

quantity, unit price, discount, and description are facilitated through the "Add More" button, which also triggers the automatic calculation of the total price. To store information, the admin can utilize the "Invoice Store" button, while the "Delete" button serves to remove data.

The screenshot shows the 'Add Invoice' interface. At the top, there are input fields for 'Inv No' (2), 'Date' (27-Aug-2023), 'Category Name' (Headphone), 'Product Name' (Realyou LK93 Headphone), 'Stock(Pic/Kg)' (23), and a ' + Add More' button. Below this is a table with columns: Category, Product Name, PSC/KG, Unit Price, Total Price, and Action. It lists one item: Headphone (Realyou LK93 Headphone) at 2 units, 2500 price, and 5000 total. There are rows for 'Discount' and 'Grand Total'. A note below says '2 headphones of Realyou'. At the bottom, there are dropdowns for 'Paid Status' (Full Paid) and 'Customer Name' (Emam Hasan - 01771408299), and a green 'Invoice Store' button.

Figure 28: Add Order

- The customer can pay in full either full due or partial payment.

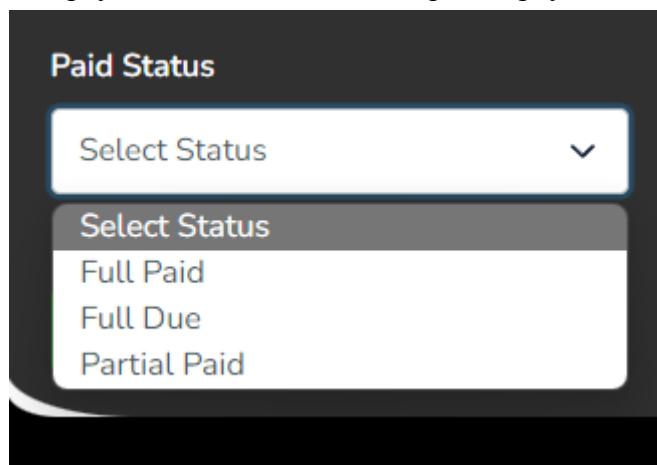


Figure 29: Payment Status Select

- From the customer name drop-down box, the admin can select the respective customer.

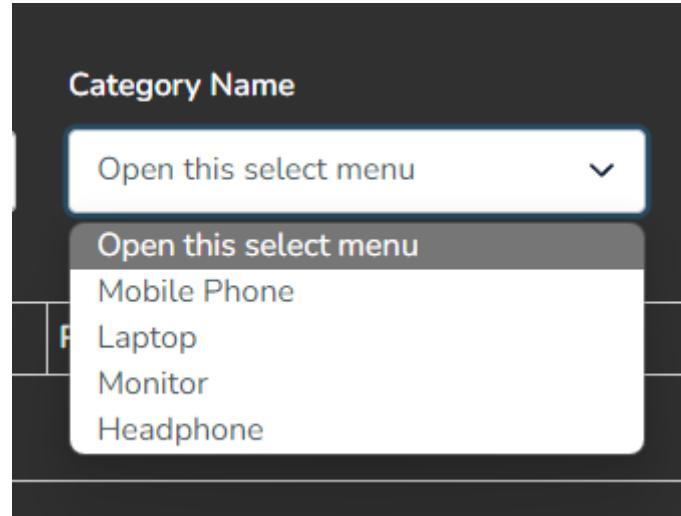


Figure 30: Customer Name Select

- Invoice store button saves the details of the invoice.

4.2. Order Approval:

This page encompasses a variety of buttons with distinct functions.

- Delete Button: This control facilitates the removal of invoice details from the database.
- Approve Button: This button serves the purpose of endorsing the invoice request.
- Add Invoice Button: This button functions in a manner analogous to the "Add Invoice" button found within all invoice sections.

ORDER APPROVAL								
Orders waiting for Approval								
Show <input type="button" value="10"/> entries <input type="text" value="Search:"/>								
Sl	Customer Name	Invoice No	Date	Desctipion	Amount	Status	Action	
1	Emam Hasan	#2	27-08-2023	2 headphones of Realyou	5000	Pending	<input checked="" type="button"/> <input type="button"/>	

Showing 1 to 1 of 1 entries

Figure 31: Order Approval

4.3. Print Order List: Within this segment, the print button facilitates the printing of specific customer invoice details.

PRINT INVOICE								
Inovice Data								
Show	10	entries						
Sl	Customer Name	Invoice No	Date	Desctipion	Amount	Action		
1	Farhan Faruk	#1	14-08-2023	Thanks for purchasing.	660			

Showing 1 to 1 of 1 entries

Figure 32: Print Invoice All

4.4 Print Interface:

INVOICE						
emaNoN Corp						Invoice No # 1
emaNoN Corp: BRACU, Dhaka support@emaNoN.com						Invoice Date: 14-08-2023
Customer Invoice						
Customer Name		Customer Mobile	Address		Description	
Farhan Faruk		0111115554	farhan@gmail.com		Thanks for purchasing.	
Sl	Category	Product Name	Current Stock	Quantity	Unit Price	Total Price
1	Headphone	Realyou LK93 Headphone	23	2	330	660
						Subtotal
						660
						Discount Amount
						0
						Paid Amount
						0
						Due Amount
						0
						Grand Amount
						660

4.5. Daily Order Report:

In this area, the admin has the capability to view invoice reports by selecting start and end dates. Upon clicking the "Search" button, they are directed to the page displaying daily invoice reports. Additionally, a download button is available in Figure 34 for obtaining the report.

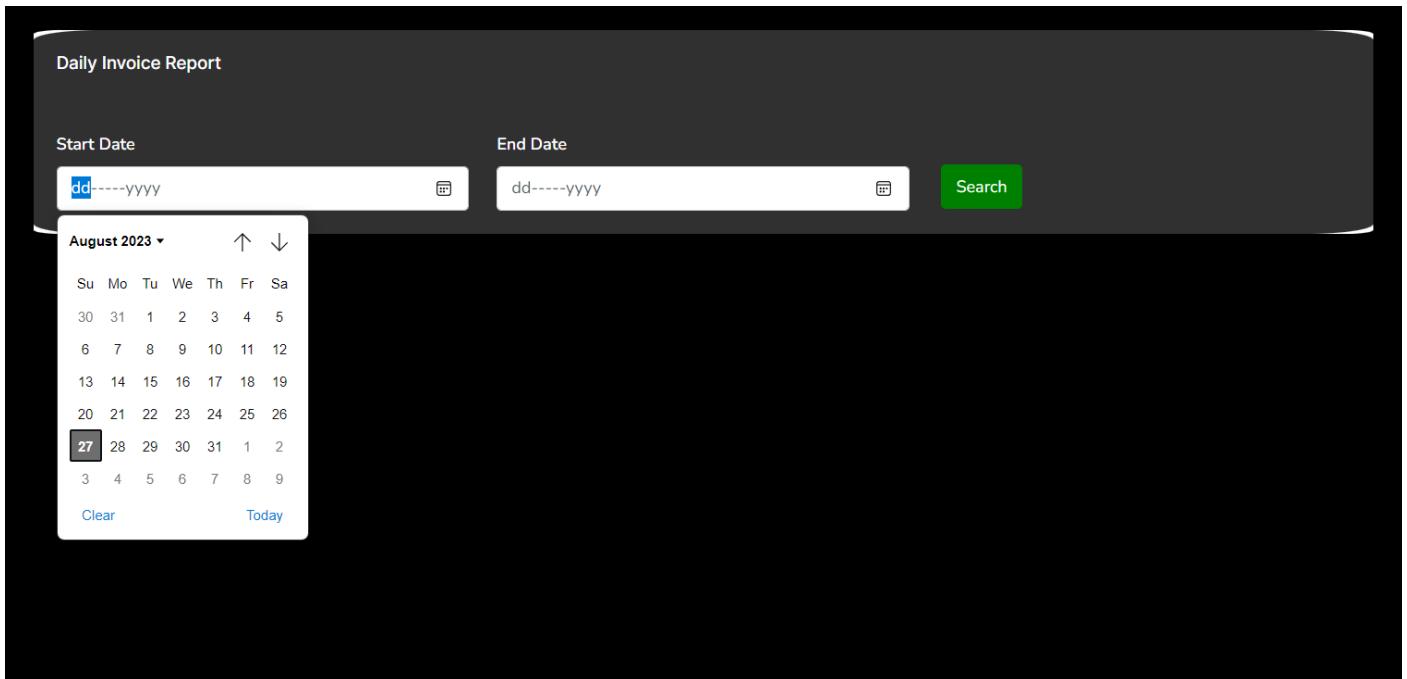


Figure 33: Daily Invoice Report

4.4. Daily Order Report:

The screenshot shows a dark-themed user interface for a 'DAILY INVOICE REPORT'. At the top left, it says 'DAILY INVOICE REPORT'. At the top right, there is a breadcrumb navigation 'Daily Invoice Report'. Below that is a logo for 'emaNoN Corp'. To the right of the logo is contact information: 'emaNoN Corp: BRACU, Dhaka support@emaNoN.com'. Below this, a green button displays the report range 'Daily Invoice Report 01-08-2023 - 31-08-2023'. The main area contains a table with the following data:

SL	Customer Name	Invoice No	Date	Description	Amount
1	Farhan Faruk	#1	14-08-2023	Thanks for purchasing.	660
				Grand Amount	\$660

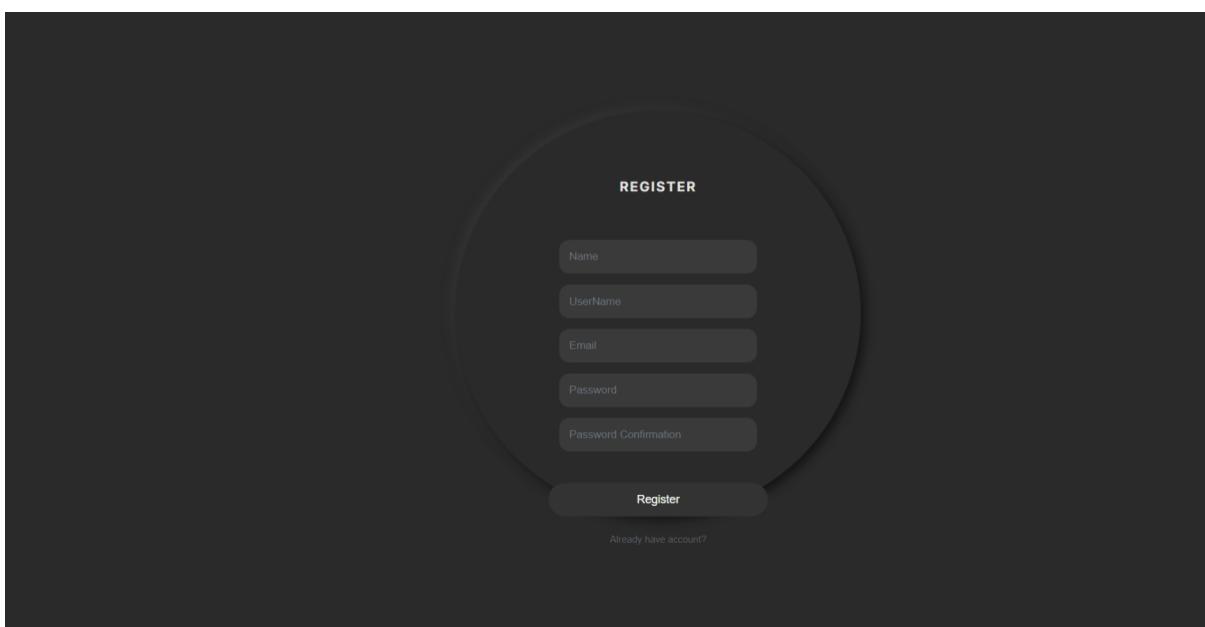
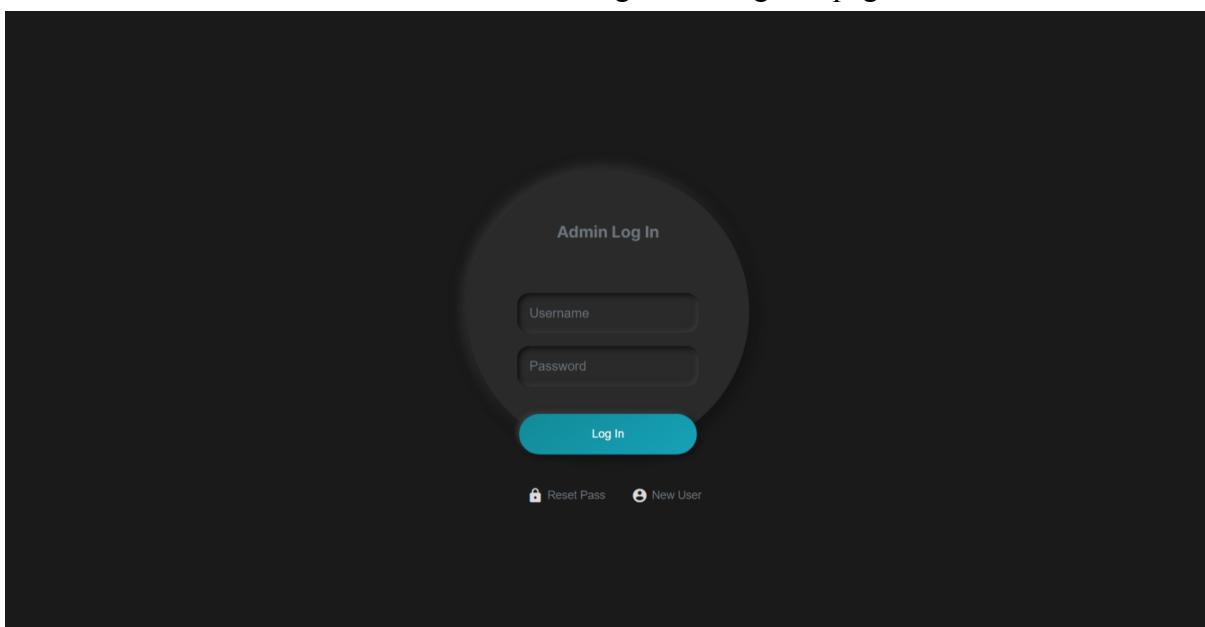
At the bottom right of the table are two buttons: a green one with a download icon and a blue one with the word 'Download'.

Figure 34: Daily Invoice Report Download

Frontend Development

Our front-end development is predominantly centered around Bootstrap, a flexible and sturdy front-end framework. This choice empowers us to create user interfaces that possess responsiveness and aesthetic appeal, seamlessly adapting to various screens and devices. By harnessing the diverse range of pre-built elements and adaptable grid system that Bootstrap offers, we ensure a consistent and user-focused experience across different platforms. Moreover, Bootstrap's modular structure allows us to efficiently customize and design components in accordance with the distinct visual and branding needs of our project.

1. This is our admin Login and Register page.



2. This is our admin profile page.

This screenshot shows the admin profile page. On the left is a sidebar menu with the following items:

- Track
- Suppliers Overview
- Customer Overview
- Unit Overview
- Category Overview
- Product Overview
- Purchase Overview
- Order Overview

The main content area displays a circular profile picture of a person. Below the picture, the user's name is listed as "Name : Emam Hasan". Underneath that, the user's email is listed as "User Email : emamhasan716@gmail.com". Further down, the user's username is listed as "User Name : emamhasan_". At the bottom of this section is a blue "EDIT PROFILE" button. At the very bottom of the page, there is a small footer note: "This Project is developed by emaNON Team".

3. This is our Track Overview.

This screenshot shows the Track Overview page. On the left is a sidebar menu with the following items:

- Track
- Suppliers Overview
- Customer Overview
- Unit Overview
- Category Overview
- Product Overview
- Purchase Overview
- Order Overview

The main content area features a grid of nine circular cards, each containing a number and a corresponding icon. The data is as follows:

Category	Value	Icon
Total Suppliers	3	Cart icon
New Customers	0	User icon
Total Customers	2	User icon
Total Units	3	Pencil icon
Total Categories	4	List icon
Total Products	1	Globe icon
Approved Purchases	1	Document icon
Pending Purchases	1	Document icon
Approved Orders	1	Document icon
Pending Orders	0	Document icon

At the bottom of the page, there is a small footer note: "This Project is developed by emaNON Team".

4. This is our all-supplier page which shows details of suppliers.

The screenshot shows a dark-themed web application interface. On the left is a sidebar with navigation links: Track, Suppliers Overview (which is currently selected), Customer Overview, Unit Overview, Category Overview, Product Overview, Purchase Overview, and Order Overview. The main content area is titled 'SUPPLIER ALL'. It displays a table with the following data:

Sl	Name	Mobile Number	Email	Address	Action
1	Techland	789845	rtecg@gov.nd	Agargaon	[Edit] [Delete]
2	Startech	0000111	startecg@gov.nd	Agargaon	[Edit] [Delete]
3	BRAC Chicken Farm	404	nai@gmail.com	BRACU U86	[Edit] [Delete]

Below the table, it says 'Showing 1 to 3 of 3 entries'. At the top right of the main content area is a blue button labeled 'Add Supplier'. The bottom right corner of the main content area contains three small circular icons with numbers 1, 2, and 3. At the very bottom of the screen, there is a footer bar with the text 'This Project is developed by emaNON Team'.

5. This is our add supplier page.

The screenshot shows a dark-themed web application interface. On the left is a sidebar with navigation links: Track, Suppliers Overview (selected), Customer Overview, Unit Overview, Category Overview, Product Overview, Purchase Overview, and Order Overview. The main content area is titled 'Add Supplier Page'. It contains four input fields with labels: 'Supplier Name', 'Supplier Mobile', 'Supplier Email', and 'Supplier Address'. Below these fields is a green button labeled 'Add Supplier'. The bottom right corner of the main content area contains three small circular icons with numbers 1, 2, and 3. At the very bottom of the screen, there is a footer bar with the text 'This Project is developed by emaNON Team'.

6. This is our all customers page where it shows the details of all customers.

The screenshot shows a dark-themed web application interface. On the left is a sidebar with navigation links: Track, Suppliers Overview, Customer Overview (selected), Customers, Unit Overview, Category Overview, Product Overview, Purchase Overview, and Order Overview. The main content area is titled "CUSTOMER ALL". It contains a table titled "Customer All Data" with two entries. The columns are Sl, Name, Customer Image, Email, Address, and Action. The first entry is Farhan Faruk (farhan@gmail.com) from CUDS, Chittagong University. The second entry is Emam Hasan (emamhasan716@gmail.com) from Moghbazar, Dhaka. Each row has a "Edit" button (green with white icon) and a "Delete" button (red with white icon). A search bar is at the top right, and a message "Showing 1 to 2 of 2 entries" is at the bottom. The footer says "This Project is developed by emaNON Team".

7. This is our add customer page.

The screenshot shows the "Add Customer Page" within the same application. The sidebar is identical to the previous screen. The main form has fields for Customer Name, Customer Mobile, Customer Email, Customer Address, and Customer Image. The Customer Image field includes a "Choose File" button and a placeholder "NO IMAGE". A green "Add Customer" button is at the bottom. The footer is the same as the previous screen.

8. This page shows all unit details.

The screenshot shows a dark-themed application interface. On the left is a sidebar with a navigation menu containing items like 'Track', 'Suppliers Overview', 'Customer Overview', 'Unit Overview' (which is currently selected), 'Category Overview', 'Product Overview', 'Purchase Overview', and 'Order Overview'. The main content area is titled 'UNIT ALL' and contains a table with three entries:

Sl	Name	Action
1	Meter	[Edit] [Delete]
2	KG	[Edit] [Delete]
3	PCs	[Edit] [Delete]

Below the table, it says 'Showing 1 to 3 of 3 entries'. At the top right of the content area, there is a blue button labeled 'Add Unit'. The bottom right corner of the screen has a footer bar with three circular icons.

9. This is our add unit page.

The screenshot shows a dark-themed application interface. On the left is a sidebar with a navigation menu containing items like 'Track', 'Suppliers Overview', 'Customer Overview', 'Unit Overview' (which is currently selected), 'Category Overview', 'Product Overview', 'Purchase Overview', and 'Order Overview'. The main content area is titled 'Add Unit Page' and contains a single input field labeled 'Unit Name' with a placeholder 'Enter Unit Name'. Below the input field is a green button labeled 'Add Unit'. The bottom right corner of the screen has a footer bar with three circular icons.

10. This is our product category page.

The screenshot shows a dark-themed web application interface. On the left, there is a sidebar with various navigation options: Track, Suppliers Overview, Customer Overview, Unit Overview, Category Overview (which is currently selected), Product Overview, Purchase Overview, and Order Overview. The main content area is titled "CATEGORY ALL" and contains a table titled "Category All Data". The table has columns for "Sl.", "Name", and "Action". It lists four entries: Headphone, Monitor, Laptop, and Mobile Phone. Each entry has a green "edit" icon and a red "delete" icon in the "Action" column. At the bottom of the table, it says "Showing 1 to 4 of 4 entries". The top right corner shows the user's name "Emam Hasan". A footer at the bottom of the screen says "This Project is developed by emaNON Team".

11. This is our add category page.

The screenshot shows the same dark-themed web application interface. The sidebar is identical to the previous screenshot. The main content area is titled "Add Category Page" and contains a form with a single input field labeled "Category Name" and a green "Add Category" button below it. The top right corner shows the user's name "Emam Hasan". A footer at the bottom of the screen says "This Project is developed by emaNON Team".

12. This page shows details of all products.

The screenshot displays a dark-themed web application interface. On the left, a vertical sidebar menu lists various sections: Track, Suppliers Overview, Customer Overview, Unit Overview, Category Overview, Product Overview (which is currently selected), Products, Purchase Overview, and Order Overview. At the top right, there is a user profile icon labeled "Emam Hasan". The main content area is titled "PRODUCT ALL" and contains a table header "Product All Data" with columns: Sl, Name, Supplier Name, Unit, Category, and Action. Below the header, a message says "Showing 1 to 1 of 1 entries". A single row is shown in the table, representing a "Realyou LK93 Headphone" from "Startech" under the "PCs" unit and "Headphone" category. The "Action" column for this row contains two icons: a green one and a red one. At the bottom of the main content area, there is a footer note: "This Project is developed by emaNON Team".

13. This is our add product page.

The screenshot shows the "Add Product Page" within the same dark-themed application. The left sidebar remains the same. The main area has a title "Add Product Page" and contains four input fields: "Product Name" (with an empty input field), "Supplier Name" (with a dropdown placeholder "Open this select menu"), "Unit Name" (with a dropdown placeholder "Open this select menu"), and "Category Name" (with a dropdown placeholder "Open this select menu"). At the bottom of this form is a green "Add Product" button. The footer note "This Project is developed by emaNON Team" is visible at the bottom of the page.

14. This is our all-purchases detail page.

The screenshot shows a dark-themed web application interface. On the left, there is a sidebar with various navigation options under categories like Track, Suppliers Overview, Customer Overview, Unit Overview, Category Overview, Product Overview, and Purchase Overview. Under Purchase Overview, 'Purchases' is selected, followed by 'Approval Purchase' and 'Daily Report'. Below these are 'Add Purchase' and 'Order Overview'. The main content area is titled 'PURCHASE ALL' and contains a table with the following data:

Sl	Purchase No	Date	Supplier	Category	Qty	Product Name	Status	Action
1	664554	15-08-2023	Startech	Headphone	14	Realyou LK93 Headphone	Pending	Pending
2	154	14-08-2023	Startech	Headphone	25	Realyou LK93 Headphone	Approved	Approved

Below the table, it says 'Showing 1 to 2 of 2 entries'. At the top right of the main content area, there is a 'Search' input field and a 'Add Purchase' button. The bottom right corner of the main content area has three small circular icons with numbers 1, 2, and 3. The footer of the page includes the text 'This Project is developed by emaNON Team'.

15. This page shows all pending purchases.

This screenshot is identical to the one above, showing the 'PURCHASE ALL' detail page with the same data in the table. It displays two pending purchase entries for Realyou LK93 Headphones from Startech. The status for the first entry is 'Pending' and for the second is 'Approved'. The rest of the interface, including the sidebar, table structure, and footer, is also identical.

16. This page is used to generate daily purchase reports.

17. This is our add purchase page.

18. This is the Order History page.

The screenshot shows a dark-themed web application interface. On the left, there is a sidebar with a navigation menu. The menu items under 'Order Overview' are 'Order', 'Order Status', 'Print Order Invoice', and 'Daily Report'. The 'Order Status' item is currently selected and highlighted in blue. The main content area is titled 'ORDERS' and contains a sub-section titled 'Order History'. A table displays one entry:

SL	Customer Name	Invoice No	Date	Description	Amount
1	Farhan Faruk	#1	14-08-2023	Thanks for purchasing.	660

Below the table, it says 'Showing 1 to 1 of 1 entries'. At the top right of the main content area, there is a button labeled 'Add Order'. The bottom right corner of the main content area has three small circular icons with numbers 1, 2, and 3. At the very bottom of the page, there is a footer bar with the text 'This Project is developed by emaNON Team'.

19. This is our Order Status page.

This screenshot shows the same dark-themed web application interface as the previous one. The sidebar navigation is identical, with 'Order Status' being the active item. The main content area is titled 'INVOICE ALL' and contains a sub-section titled 'Invoce All Data'. A table is present, but it is empty and displays the message 'No data available in table'. Below the table, it says 'Showing 0 to 0 of 0 entries'. The top right of the main content area has a 'Search' input field and three small circular icons (1, 2, 3) in the bottom right corner. The footer at the bottom of the page includes the text 'This Project is developed by emaNON Team'.

20. This is our print order invoice list page.

The screenshot shows a dark-themed web application interface. On the left, a sidebar menu includes 'Track', 'Suppliers Overview', 'Customer Overview', 'Unit Overview', 'Category Overview', 'Product Overview', 'Purchase Overview', 'Order Overview' (selected), 'Order Status', 'Print Order Invoice' (selected), and 'Daily Report'. The main content area is titled 'PRINT INVOICE' and contains a sub-section titled 'Invoice Data'. A table displays one entry:

Sl	Customer Name	Invoice No	Date	Description	Amount	Action
1	Farhan Faruk	#1	14-08-2023	Thanks for purchasing.	660	Edit

Below the table, it says 'Showing 1 to 1 of 1 entries'. At the bottom right of the content area are three small circular icons with numbers 1, 2, and 3. The footer of the page includes the text 'This Project is developed by emaNON Team'.

21. This page is used to generate invoice reports.

The screenshot shows the same dark-themed web application interface as the previous page. The sidebar menu is identical. The main content area is titled 'Daily Invoice Report' and features two input fields: 'Start Date' and 'End Date', both with placeholder text 'dd----yyyy'. To the right of these fields is a green 'Search' button. The footer of the page includes the text 'This Project is developed by emaNON Team'.

22. This is our add Order page.

23. This page shows the Invoice report of our inventory.

Backend Development

The backend of the system operates like a well-oiled machine. It adheres to a structure known as MVC, where various components are orderly placed. We start by building a model for handling data. Next, we create a view that users can see and a controller that controls operations. We employ specialized pathways that direct how data is moved around and how individuals may act in order to make everything function together. JavaScript and PHP are both used to make sure the system functions as it should.

MySQL, which functions in the background like a smart storage system, is used to store and retrieve data. We keep things orderly and simple to work on later by employing the MVC methodology. It's similar to using separate teams for various tasks, which strengthens and adapts the system. To make all of this even better, we further employ the Laravel-9 tool. The system's ingenious pathways ensure that data flows smoothly between various components, making user interaction easy and entertaining.

Track:

This is the backend code for the track.

```

resources > views > admin > index.blade.php
1  @extends('admin.admin_master')
2  @section('admin')
3
4  <div class="page-content">
5      <div class="container-fluid">
6
7          <div class="row">
8              <div class="col-xl-3 col-md-6">
9                  <div class="card">
10                     <div class="card-body">
11                         <div class="d-flex">
12                             <div class="flex-grow-1">
13                                 <p class="text-truncate font-size-14 mb-2">Approved Purchases</p>
14                                 <h4 class="mb-2">
15                                     {{ \App\Models\Purchase::where('status', 1)->count() }} <!-- Count purchases with status 1 -->
16                                 </h4>
17                                 <p class="text-muted mb-0"><span class="text-success fw-bold font-size-12 me-2">i class="ri-arrow-right-up-line me-1 align
18                                     </div>
19                                     <div class="avatar-sm">
20                                         <span class="avatar-title bg-light text-primary rounded-3">
21                                             <i class="ri-shopping-cart-2-line font-size-24"></i>
22                                         </span>
23                                     </div>
24                                 </div>
25                             </div><!-- end cardbody -->
26                         </div><!-- end card -->
27                     </div><!-- end col -->
28
29                     <!-- Pending Purchases section -->
30                     <div class="col-xl-3 col-md-6">
31                         <div class="card">
32                             <div class="card-body">
33                                 <div class="d-flex">
34                                     <div class="flex-grow-1">
35                                         <p class="text-truncate font-size-14 mb-2">Pending Purchases</p>
36                                         <h4 class="mb-2">
37                                             {{ \App\Models\Purchase::where('status', 0)->count() }} <!-- Count purchases with status 0 -->
38                                         </h4>
39                                         <p class="text-muted mb-0"><span class="text-danger fw-bold font-size-12 me-2">i class="ri-arrow-right-down-line me-1 align

```

- Here, we can see the Track page's code implementation. It'll show the number of approved purchases, pending purchases, total number of suppliers, new customers(those who are likely to

make their first purchase within the last 7 days), status of invoices(approved & pending) and total customers (all customers).

```

resources > views > admin > 📈 index.blade.php
  41 |     |     |         |     <div class="flex-grow-1">
  42 |     |     |         |     <div class="avatar-sm">
  43 |     |     |         |     <span class="avatar-title bg-light text-danger rounded-3">
  44 |     |     |         |     |     <i class="ri-shopping-cart-2-fill font-size-24"></i>
  45 |     |     |         |     </span>
  46 |     |     |     </div>
  47 |     |     |     </div><!-- end cardbody -->
  48 |     |     | </div><!-- end card -->
  49 |     |     | </div><!-- end col -->
  50 |
  51 |     |     |     <!-- ... other columns ... -->
  52 |
  53 |
  54 |     <div class="col-xl-3 col-md-6">
  55 |     |     <div class="card">
  56 |     |     |     <div class="card-body">
  57 |     |     |     |     <div class="d-flex">
  58 |     |     |     |     <div class="flex-grow-1">
  59 |     |     |     |     |     <p class="text-truncate font-size-14 mb-2">Total Suppliers</p>
  60 |     |     |     |     |     <h4 class="mb-2">
  61 |     |     |     |     |     |     {{ \App\Models\Supplier::whereNull('deleted_at')->count() }}
  62 |     |     |     |     </h4>
  63 |     |     |     |     |     <p class="text-muted mb-0"><span class="text-success fw-bold font-size-12 me-2"><i class="ri-arrow-right-up-line me-1 align-right"></i> {{ \App\Models\Supplier::whereNull('deleted_at')->count() }}</span></p>
  64 |     |     |     |     </div>
  65 |     |     |     <div class="avatar-sm">
  66 |     |     |     <span class="avatar-title bg-light text-primary rounded-3">
  67 |     |     |     |     <i class="ri-user-3-line font-size-24"></i>
  68 |     |     |     </span>
  69 |     |     |     </div>
  70 |     |     |     </div>
  71 |     |     |     </div><!-- end cardbody -->
  72 |     |     | </div><!-- end card -->
  73 |     |     | </div><!-- end col -->
  74 |
  75 |     |     |     <!-- New Customer section -->
  76 |     <div class="col-xl-3 col-md-6">
  77 |     |     <div class="card">
  78 |     |     |     <div class="card-body">
  79 |     |     |     |     <div class="d-flex">

```

```

resources > views > admin > 📈 index.blade.php
  80 |     |     |     |     <div class="flex-grow-1">
  81 |     |     |     |     |     <p class="text-truncate font-size-14 mb-2">New Customers (Last 7 Days)</p>
  82 |     |     |     |     |     <h4 class="mb-2">
  83 |     |     |     |     |     |     {{ \App\Models\Customer::where('created_at', '>=', now()->subDays(7))->count() }}
  84 |     |     |     |     </h4>
  85 |     |     |     |     |     <p class="text-muted mb-0"><span class="text-success fw-bold font-size-12 me-2"><i class="ri-arrow-right-up-line me-1 align-right"></i> {{ \App\Models\Customer::where('created_at', '>=', now()->subDays(7))->count() }}</span></p>
  86 |     |     |     |     </div>
  87 |     |     |     <div class="avatar-sm">
  88 |     |     |     <span class="avatar-title bg-light text-primary rounded-3">
  89 |     |     |     |     <i class="ri-user-5-line font-size-24"></i>
  90 |     |     |     </span>
  91 |     |     |     </div>
  92 |     |     |     </div>
  93 |     |     |     </div><!-- end cardbody -->
  94 |     |     | </div><!-- end card -->
  95 |     |     | </div><!-- end col -->
  96 |
  97 |
  98 |     |     |     <!-- New Customer section -->
  99 |     <div class="col-xl-3 col-md-6">
 100 |     |     <div class="card">
 101 |     |     |     <div class="card-body">
 102 |     |     |     |     <div class="d-flex">
 103 |     |     |     |     <div class="flex-grow-1">
 104 |     |     |     |     |     <p class="text-truncate font-size-14 mb-2">Approved Invoices</p>
 105 |     |     |     |     |     <h4 class="mb-2">
 106 |     |     |     |     |     |     {{ \App\Models\Invoice::where('status', 1)->count() }}
 107 |     |     |     |     </h4>
 108 |     |     |     |     |     <p class="text-muted mb-0"><span class="text-success fw-bold font-size-12 me-2"><i class="ri-arrow-right-up-line me-1 align-right"></i> {{ \App\Models\Invoice::where('status', 1)->count() }}</span></p>
 109 |     |     |     |     </div>
 110 |     |     |     <div class="avatar-sm">
 111 |     |     |     <span class="avatar-title bg-light text-primary rounded-3">
 112 |     |     |     |     <i class="ri-user-5-line font-size-24"></i>
 113 |     |     |     </span>
 114 |     |     |     </div>
 115 |     |     |     </div>
 116 |     |     |     </div><!-- end cardbody -->
 117 |     |     | </div><!-- end card -->
 118 |     |     | </div><!-- end col -->
```

```

resources > views > admin > index.blade.php
109 |         </div>
110 |         <div class="avatar-sm">
111 |             <span class="avatar-title bg-light text-primary rounded-3">
112 |                 |     <i class="ri-user-5-line font-size-24"></i>
113 |             </span>
114 |         </div>
115 |     </div><!-- end cardbody -->
116 | </div><!-- end card -->
117 | </div><!-- end col -->
118 |
119 |
120 |
121 | <!-- New Customer section -->
122 | <div class="col-xl-3 col-md-6">
123 |     <div class="card">
124 |         <div class="card-body">
125 |             <div class="d-flex">
126 |                 <div class="flex-grow-1">
127 |                     <p class="text-truncate font-size-14 mb-2">Pending Invoices</p>
128 |                     <h4 class="mb-2">
129 |                         {{ \App\Models\Invoice::where('status', 0)->count() }}<br/>
130 |                     </h4>
131 |                     <p class="text-muted mb-0"><span class="text-success fw-bold font-size-12 me-2"><i class="ri-arrow-right-up-line me-1 align-middle"></i> {{ \App\Models\Invoice::where('status', 0)->count() }}</span></p>
132 |                 </div>
133 |                 <div class="avatar-sm">
134 |                     <span class="avatar-title bg-light text-primary rounded-3">
135 |                         |     <i class="ri-user-5-line font-size-24"></i>
136 |                     </span>
137 |                 </div>
138 |             </div>
139 |         </div><!-- end cardbody -->
140 |     </div><!-- end card -->
141 | </div><!-- end col -->
142 |
143 |
144 | <!-- Total Customers section -->
145 | <div class="col-xl-3 col-md-6">
146 |     <div class="card">
147 |         <div class="card-body">

```

```

resources > views > admin > index.blade.php
133 |         <div class="avatar-sm">
134 |             <span class="avatar-title bg-light text-primary rounded-3">
135 |                 |     <i class="ri-user-5-line font-size-24"></i>
136 |             </span>
137 |         </div>
138 |     </div><!-- end cardbody -->
139 | </div><!-- end card -->
140 | </div><!-- end col -->
141 |
142 |
143 |
144 | <!-- Total Customers section -->
145 | <div class="col-xl-3 col-md-6">
146 |     <div class="card">
147 |         <div class="card-body">
148 |             <div class="d-flex">
149 |                 <div class="flex-grow-1">
150 |                     <p class="text-truncate font-size-14 mb-2">Total Customers</p>
151 |                     <h4 class="mb-2">
152 |                         {{ \App\Models\Customer::count() }}<br/><!-- Count all customers --&gt;
153 |                     &lt;/h4&gt;
154 |                     &lt;p class="text-muted mb-0"&gt;&lt;span class="text-success fw-bold font-size-12 me-2"&gt;&lt;i class="ri-arrow-right-up-line me-1 align-middle"&gt;&lt;/i&gt; {{ \App\Models\Customer::count() }}&lt;/span&gt;&lt;/p&gt;
155 |                 &lt;/div&gt;
156 |                 &lt;div class="avatar-sm"&gt;
157 |                     &lt;span class="avatar-title bg-light text-primary rounded-3"&gt;
158 |                         |     &lt;i class="ri-users-line font-size-24"&gt;&lt;/i&gt;
159 |                     &lt;/span&gt;
160 |                 &lt;/div&gt;
161 |             &lt;/div&gt;
162 |         &lt;/div&gt;&lt;!-- end cardbody --&gt;
163 |     &lt;/div&gt;&lt;!-- end card --&gt;
164 | &lt;/div&gt;&lt;!-- end col --&gt;
165 | &lt;/div&gt;&lt;!-- end row --&gt;
166 |
167 |     &lt;/div&gt;
168 | &lt;/div&gt;
169 @endsection
</pre>

```

Category:

- Show all Categories.
- Add a new Category.
- Edit, Update and Delete categories from existing category list.

```
resources > views > backend > category > category_add.blade.php
1  <!-- resources/views/backend/category/category_add.blade.php -->
2
3  @extends('admin.admin_master')
4  @section('admin')
5
6  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
7
8  <div class="page-content">
9      <div class="container-fluid">
10         <div class="row">
11             <div class="col-12">
12                 <div class="card">
13                     <div class="card-body">
14
15                         <h4 class="card-title">Add Category Page</h4><br><br>
16
17                         <form method="post" action="{{ route('category.store') }}" id="myForm">
18                             @csrf
19
20
21                             <!-- Category Name -->
22                             <div class="row mb-3">
23                                 <label for="name" class="col-sm-2 col-form-label">Category Name</label>
24                                 <div class="form-group col-sm-10">
25                                     <input name="name" class="form-control" type="text" value="{{ old('name') }}"/>
26                                     @error('name')
27                                         <span class="text-danger">{{ $message }}</span>
28                                     @enderror
29                                 </div>
30
31                             <!-- end Category Name -->
32
33                             <input type="submit" class="btn btn-info waves-effect waves-light" value="Add Category">
34
35                         </form>
36
37                     </div>
38
39             </div> <!-- end col -->
40
41         </div>
42     </div>
43
44     <script type="text/javascript">
45         // JavaScript validation code remains the same
46     </script>
47
48     @endsection
```

resources > views > backend > category > category_all.blade.php

```
3 <div class="page-content">
4     <div class="container-fluid">
5         <!-- start page title -->
6         <div class="row">
7             <div class="col-12">
8                 <div class="page-title-box d-sm-flex align-items-center justify-content-between">
9                     <h4 class="mb-sm-0">Category All</h4>
10
11
12             </div>
13         </div>
14     </div>
15     </div>
16     <!-- end page title -->
17
18 <div class="row">
19     <div class="col-12">
20         <div class="card">
21             <div class="card-body">
22
23             <a href="{{ route('category.add') }}" class="btn btn-dark btn-rounded waves-effect waves-light" style="float:right;">Add Category </a> <br> <br>
24
25             <h4 class="card-title">Category All Data </h4>
26
27
28             <table id="datatable" class="table table-bordered dt-responsive nowrap" style="border-collapse: collapse; border-spacing: 0; width: 100%;">
29                 <thead>
30                     <tr>
31                         <th width="5%">$1</th>
32                         <th>Name</th>
33                         <th width="20%">Action</th>
34
35                 </thead>
36                 <tbody>
37
38                     @foreach($categoris as $key => $item)
39                     <tr>
40                         <td> {{ $key+1}} </td>
41                         <td> {{ $item->name }} </td>
42                         <td>
43                             <a href="{{ route('category.edit',$item->id) }}" class="btn btn-info sm" title="Edit Data" > <i class="fas fa-edit"></i> </a>
44
45                             <a href="{{ route('category.delete',$item->id) }}" class="btn btn-danger sm" title="Delete Data" id="delete" > <i class="fas fa-trash-alt"></i> </a>
46
47                         </td>
48
49                     </tr>
50                     @endforeach
51
52             </tbody>
53         </table>
54     </div>
55 
```

```

resources > views > backend > category > category_edit.blade.php
 7  <div class="row">
 8  <div class="col-12">
 9  |   <div class="card">
10  |     <div class="card-body">
11  |
12  |       <h4 class="card-title">Edit Category Page </h4><br><br>
13  |       <form method="post" action="{{ route('category.update') }}" id="myForm" >
14  |         @csrf
15  |         <input type="hidden" name="id" value="{{ $category->id }}>
16  |         <div class="row mb-3">
17  |           <label for="example-text-input" class="col-sm-2 col-form-label">Category Name </label>
18  |           <div class="form-group col-sm-10">
19  |             <input name="name" value="{{ $category->name }}" class="form-control" type="text" >
20  |           </div>
21  |         </div>
22  |         <!-- end row -->
23  |         <input type="submit" class="btn btn-info waves-effect waves-light" value="Update Category">
24  |       </form>
25  |     </div>
26  |   </div>
27  </div> <!-- end col -->
28 </div>
29 </div>
30 </div>
31 <script type="text/javascript">
32 $(document).ready(function (){
33   $('#myForm').validate({
34     rules: {
35       name: {
36         required : true,
37       },
38     },
39     messages :{
40       name: {
41         required : 'Please Enter Your Name',
42       },
43     },
44     errorElement : 'span',
45     errorPlacement: function (error,element) {
46       error.addClass('invalid-feedback');
47       element.closest('.form-group').append(error);
48     },
49     highlight : function(element, errorClass, validClass){
50       $(element).addClass('is-invalid');
51     },
52     unhighlight : function(element, errorClass, validClass){
53     },
54   })
55 })

```

Customer:

In this section, we can see the list of all customers. It will also show the details of a customer (Name, Mobile, Email, Address, Image). Moreover, we can edit & update the customer information.

```
resources > views > backend > customer > 🗃 customer_add.blade.php
13 |     <h4 class="card-title">Add Customer Page </h4><br><br>
14 |
15 |     <form method="post" action="{{ route('customer.store') }}" id="myForm" enctype="multipart/form-data">
16 |         @csrf
17 |
18 |         <div class="row mb-3">
19 |             <label for="example-text-input" class="col-sm-2 col-form-label">Customer Name </label>
20 |             <div class="form-group col-sm-10">
21 |                 <input name="name" class="form-control" type="text" >
22 |             </div>
23 |         </div>
24 |         <!-- end row -->
25 |
26 |
27 |         <div class="row mb-3">
28 |             <label for="example-text-input" class="col-sm-2 col-form-label">Customer Mobile </label>
29 |             <div class="form-group col-sm-10">
30 |                 <input name="mobile_no" class="form-control" type="text" >
31 |             </div>
32 |         </div>
33 |         <!-- end row -->
34 |
35 |
36 |         <div class="row mb-3">
37 |             <label for="example-text-input" class="col-sm-2 col-form-label">Customer Email </label>
38 |             <div class="form-group col-sm-10">
39 |                 <input name="email" class="form-control" type="email" >
40 |             </div>
41 |         </div>
42 |         <!-- end row -->
43 |
44 |
45 |         <div class="row mb-3">
46 |             <label for="example-text-input" class="col-sm-2 col-form-label">Customer Address </label>
47 |             <div class="form-group col-sm-10">
48 |                 <input name="address" class="form-control" type="text" >
49 |             </div>
50 |         </div>
51 |         <!-- end row -->
52 |
53 |         <div class="row mb-3">
54 |             <label for="example-text-input" class="col-sm-2 col-form-label">Customer Image </label>
55 |             <div class="form-group col-sm-10">
56 |                 <input name="customer_image" class="form-control" type="file" id="image">
57 |             </div>
58 |         </div>
59 |         <!-- end row -->
```

```
resources > views > backend > customer > customer_add.blade.php
74  </div>
75  <script type="text/javascript">
76  $(document).ready(function (){
77      $('#myForm').validate({
78          rules: {
79              name: {
80                  required : true,
81              },
82              mobile_no: {
83                  required : true,
84              },
85              email: {
86                  required : true,
87              },
88              address: {
89                  required : true,
90              },
91              customer_image: {
92                  required : true,
93              },
94          },
95          messages :{
96              name: {
97                  required : 'Please Enter Your Name',
98              },
99              mobile_no: {
100                  required : 'Please Enter Your Mobile Number',
101              },
102              email: {
103                  required : 'Please Enter Your Email',
104              },
105              address: {
106                  required : 'Please Enter Your Address',
107              },
108              customer_image: {
109                  required : 'Please Select One Image',
110              },
111          },
112          errorElement : 'span',
113          errorPlacement: function (error,element) {
114              error.addClass('invalid-feedback');
115              element.closest('.form-group').append(error);
116          },
117          highlight : function(element, errorClass, validClass){
118              $(element).addClass('is-invalid');
119          },
120          unhighlight : function(element, errorClass, validClass){
121              $(element).removeClass('is-invalid');
122          }
123      })
124  })
125 
```

```

resources > views > backend > customer > 📄 customer_all.blade.php
12 | | | | | <h4 class="mb-sm-0">Customer All</h4>
13 |
14 |
15 |
16 | | | | | </div>
17 | | | | | </div>
18 | | | | | <!-- end page title -->
19 |
20 |
21 <div class="row">
22 | <div class="col-12">
23 | | <div class="card">
24 | | | <div class="card-body">
25 |
26 <a href="{{ route('customer.add') }}" class="btn btn-dark btn-rounded waves-effect waves-light" style="float:right;">Add Customer </a> <br> <br>
27 |
28 | | | <h4 class="card-title">Customer All Data </h4>
29 |
30 |
31 | | | <table id="datatable" class="table table-bordered dt-responsive nowrap" style="border-collapse: collapse; border-spacing: 0; width: 100%;"
32 | | | | <thead>
33 | | | | | <tr>
34 | | | | | | <th>Sl</th>
35 | | | | | | <th>Name</th>
36 | | | | | | <th>Customer Image </th>
37 | | | | | | <th>Email</th>
38 | | | | | | <th>Address</th>
39 | | | | | | <th>Action</th>
40 |
41 | | | | </thead>
42 |
43 |
44 | | | <tbody>
45 | | | | @foreach($customers as $key => $item)
46 | | | | <tr>
47 | | | | | <td> {{ $key+1}} </td>
48 | | | | | <td> {{ $item->name }} </td>
49 | | | | | <td>  </td>
50 | | | | | <td> {{ $item->email }} </td>
51 | | | | | <td> {{ $item->address }} </td>
52 | | | | | <td>
53 | | | | | <a href="{{ route('customer.edit',$item->id) }}" class="btn btn-info sm" title="Edit Data" > <i class="fas fa-edit"></i> </a>
54 |
55 | | | | <a href="{{ route('customer.delete',$item->id) }}" class="btn btn-danger sm" title="Delete Data" id="delete" > <i class="fas fa-trash-alt"></i> </a>
56 |
57 | | | | </td>
58 |
59 | | | </tbody>

```

```

resources > views > backend > customer > 📄 customer_credit.blade.php
12 | | | | | <h4 class="mb-sm-0">Credit Customer All</h4>
13 |
14 |
15 |
16 | | | | | </div>
17 | | | | | </div>
18 | | | | | <!-- end page title -->
19 |
20 |
21 <div class="row">
22 | <div class="col-12">
23 | | <div class="card">
24 | | | <div class="card-body">
25 |
26 <a href="{{ route('credit.customer.print.pdf') }}" target="_blank" class="btn btn-dark btn-rounded waves-effect waves-light" style="float:right;"><i class="fa fa-print"> Print Credit Customer </i> <br>
27 |
28 | | | <h4 class="card-title">Credit Customer Data </h4>
29 |
30 |
31 | | | <table id="datatable" class="table table-bordered dt-responsive nowrap" style="border-collapse: collapse; border-spacing: 0; width: 100%;"
32 | | | | <thead>
33 | | | | | <tr>
34 | | | | | | <th>Sl</th>
35 | | | | | | <th>Customer Name</th>
36 | | | | | | <th>Invoice No </th>
37 | | | | | | <th>Date</th>
38 | | | | | | <th>Due Amount</th>
39 | | | | | | <th>Action</th>
40 |
41 | | | | </thead>
42 |
43 |
44 | | | <tbody>
45 | | | | @foreach($allData as $key => $item)
46 | | | | <tr>
47 | | | | | <td> {{ $key+1}} </td>
48 | | | | | <td> {{ $item['customer']['name'] }} </td>
49 | | | | | <td> {{ $item['invoice']['invoice_no'] }} </td>
50 | | | | | <td> {{ date('d-m-Y',strtotime($item['invoice']['date'])) }} </td>
51 | | | | | <td> {{ $item->due_amount }} </td>
52 | | | | | <td>
53 | | | | | <a href="{{ route('customer.edit',$item->id) }}" class="btn btn-info sm" title="Edit Data" > <i class="fas fa-edit"></i> </a>
54 |
55 | | | | <a href="{{ route('customer.delete',$item->id) }}" class="btn btn-danger sm" title="Delete Data" id="delete" > <i class="fas fa-trash-alt"></i> </a>
56 |
57 | | | | </td>
58 |
59 | | | </tbody>

```

```
resources > views > backend > customer > 🗑 customer_edit.blade.php
12 | |
13 |     <h4 class="card-title">Edit Customer Page </h4><br><br>
14 |
15 |
16 |
17 |     <form method="post" action="{{ route('customer.update') }}" id="myForm" enctype="multipart/form-data">
18 |         @csrf
19 |         <input type="hidden" name="id" value="{{ $customer->id}}>
20 |         <div class="row mb-3">
21 |             <label for="example-text-input" class="col-sm-2 col-form-label">Customer Name </label>
22 |             <div class="form-group col-sm-10">
23 |                 <input name="name" value= "{{ $customer->name}}" class="form-control" type="text"    >
24 |             </div>
25 |         </div>
26 |         <!-- end row -->
27 |
28 |
29 |         <div class="row mb-3">
30 |             <label for="example-text-input" class="col-sm-2 col-form-label">Customer Mobile </label>
31 |             <div class="form-group col-sm-10">
32 |                 <input name="mobile_no" value= "{{ $customer-> mobile_no}}" class="form-control" type="text"    >
33 |             </div>
34 |         </div>
35 |         <!-- end row -->
36 |
37 |
38 |         <div class="row mb-3">
39 |             <label for="example-text-input" class="col-sm-2 col-form-label">Customer Email </label>
40 |             <div class="form-group col-sm-10">
41 |                 <input name="email" value= "{{ $customer-> email}}" class="form-control" type="email"   >
42 |             </div>
43 |         </div>
44 |         <!-- end row -->
45 |
46 |
47 |         <div class="row mb-3">
48 |             <label for="example-text-input" class="col-sm-2 col-form-label">Customer Address </label>
49 |             <div class="form-group col-sm-10">
50 |                 <input name="address" value= "{{ $customer->address}}" class="form-control" type="text"  >
51 |             </div>
52 |         </div>
53 |         <!-- end row -->
54 |
55 |         <div class="row mb-3">
56 |             <label for="example-text-input" class="col-sm-2 col-form-label">Customer Image </label>
57 |             <div class="form-group col-sm-10">
58 |                 <input name="customer_image" class="form-control" type="file"  id="image">
59 |             </div>
60 |         </div>
```

resources > views > backend > customer >  customer_edit.blade.php

```
--> 90  <script type="text/javascript">
91  |   $(document).ready(function (){
92  |       $('#myForm').validate({
93  |           rules: {
94  |               name: {
95  |                   required : true,
96  |               },
97  |               mobile_no: {
98  |                   required : true,
99  |               },
100 |               email: {
101 |                   required : true,
102 |               },
103 |               address: {
104 |                   required : true,
105 |               },
106 |
107 |           },
108 |           messages :{
109 |               name: {
110 |                   required : 'Please Enter Your Name',
111 |               },
112 |               mobile_no: {
113 |                   required : 'Please Enter Your Mobile Number',
114 |               },
115 |               email: {
116 |                   required : 'Please Enter Your Email',
117 |               },
118 |               address: {
119 |                   required : 'Please Enter Your Address',
120 |               },
121 |
122 |           },
123 |           errorElement : 'span',
124 |           errorPlacement: function (error,element) {
125 |               error.addClass('invalid-feedback');
126 |               element.closest('.form-group').append(error);
127 |           },
128 |           highlight : function(element, errorClass, validClass){
129 |               $(element).addClass('is-invalid');
130 |           },
131 |           unhighlight : function(element, errorClass, validClass){
132 |               $(element).removeClass('is-invalid');
133 |           },
134 |       });
135 |   });
136 |
137 </script>
```

Invoice:

Implementation of the Daily Invoice Report.

```
resources > views > backend > invoice > daily_invoice_report.blade.php
7
8  <div class="row">
9    <div class="col-12">
10      <div class="card">
11        <div class="card-body">
12          <h4 class="card-title">Daily Invoice Report </h4><br><br>
13
14  <form method="GET" action="{{ route('daily.invoice.pdf') }}" target="_blank" id="myForm">
15    <div class="row">
16
17
18
19
20    <div class="col-md-4">
21      <div class="md-3 form-group">
22        <label for="example-text-input" class="form-label">Start Date</label>
23        | <input class="form-control example-date-input" name="start_date" type="date" id="start_date" placeholder="YY-MM-DD">
24      </div>
25    </div>
26
27
28    <div class="col-md-4">
29      <div class="md-3 form-group">
30        <label for="example-text-input" class="form-label">End Date</label>
31        | <input class="form-control example-date-input" name="end_date" type="date" id="end_date" placeholder="YY-MM-DD">
32      </div>
33    </div>
34
35    <div class="col-md-4">
36      <div class="md-3">
37        <label for="example-text-input" class="form-label" style="margin-top:43px;"> </label>
38        <button type="submit" class="btn btn-info">Search</button>
39      </div>
40    </div>
41
42
43
44  </div> <!-- // end row -->
45
46  </form>
47
48  </div> <!-- End card-body -->
```

- Users can select any starting and ending date and it will generate an invoice within that time.
- In the invoice, we can see the product name, its category, unit, unit price, total price, and the action on it.
- There's also an additional option for a discount, which will reduce the price and then we'll get the Grand price(where the discount is included).
- In addition, users can add invoices for any customers. There will be the serial no., customer name, date, description, and the amount. For the pending invoice list there'll be additional status and action options.
- Users can generate the payment invoice for the customers by their name and the invoice no of the product they bought.

- For approving/pending invoices, users can check the payment status of the customers and then they can set the status either approve or pending based on the payment status.
- Lastly, the print invoice page comes, here we get a print option for the invoice list which includes the Serial no., customer name, invoice no., date, description of the items, amount and action. The payment and the total amount of items will also be shown in the list.

```

resources > views > backend > invoice > invoice_add.blade.php
  82 <!-- ----- -->
  83
  84     <div class="card-body">
  85         <form method="post" action="{{ route('invoice.store') }}>
  86             @csrf
  87             <table class="table-sm table-bordered" width="100%" style="border-color: #dd;">
  88                 <thead>
  89                     <tr>
  90                         <th>Category</th>
  91                         <th>Product Name </th>
  92                         <th width="7%">PSC/KG</th>
  93                         <th width="10%">Unit Price </th>
  94                         <th width="15%">Total Price</th>
  95                         <th width="7%">Action</th>
  96
  97                     </tr>
  98                 </thead>
  99
 100                 <tbody id="addRow" class="addRow">
 101
 102                     </tbody>
 103
 104                     <tbody>
 105
 106                         <tr>
 107                             <td colspan="4">Discount</td>
 108                             <td>
 109                                 <input type="text" name="discount_amount" id="discount_amount" class="form-control estimated_amount" placeholder="Discount Amount" >
 110                             </td>
 111                         </tr>
 112
 113                         <tr>
 114                             <td colspan="4">Grand Total</td>
 115                             <td>
 116                                 <input type="text" name="estimated_amount" value="0" id="estimated_amount" class="form-control estimated_amount" readonly style="background-color: #ddd;" >
 117                             </td>
 118                         <td></td>
 119                         </tr>
 120
 121                     </tbody>
 122                 </table><br>
 123
 124
 125                 <div class="form-row">
 126                     <div class="form-group col-md-12">
 127                         <textarea name="description" class="form-control" id="description" placeholder="Write Description Here"></textarea>
 128                     </div>

```

```

resources > views > backend > invoice >  invoice_all.blade.php
19 | | | | | <!-- end page title -->
20 |
21 <div class="row">
22 | <div class="col-12">
23 | | <div class="card">
24 | | | <div class="card-body">
25 |
26 <a href="{{ route('invoice.add') }}" class="btn btn-dark btn-rounded waves-effect waves-light" style="float:right;"> Add Invoice </a> <br> <br>
27 |
28 | | | <h4 class="card-title">Invoice All Data </h4>
29 |
30 |
31 | <table id="datatable" class="table table-bordered dt-responsive nowrap" style="border-collapse: collapse; border-spacing: 0; width: 100%;">
32 | | <thead>
33 | | | <tr>
34 | | | | <th>$l</th>
35 | | | | <th>Customer Name</th>
36 | | | | <th>Invoice No </th>
37 | | | | <th>Date </th>
38 | | | | <th>Desctipion</th>
39 | | | | <th>Amount</th>
40 |
41 | | </thead>
42 |
43 |
44 | <tbody>
45 |
46 | | | @foreach($allData as $key => $item)
47 | | <tr>
48 | | | <td> {{ $key+1}} </td>
49 | | | <td> {{ $item['payment']['customer']['name'] }} </td>
50 | | | <td> {{ $item->invoice_no }} </td>
51 | | | <td> {{ date('d-m-Y', strtotime($item->date)) }} </td>
52 |
53 |
54 | | | <td> {{ $item->description }} </td>
55 |
56 | | | <td> {{ $item['payment']['total_amount'] }} </td>
57 |
58 | | </tr>
59 | | | @endforeach
60 |
61 | | </tbody>
62 | | </table>
63 |
64 | | | </div>
65 | | | </div>
66 | | </div> <!-- end col -->
67 | </div> <!-- end row -->

```

InvisiCode - Generated

```

resources > views > backend > invoice >  invoice_approve.blade.php
22 | @endphp
23 | <div class="row">
24 |   <div class="col-12">
25 |     <div class="card">
26 |       <div class="card-body">
27 |         <h4>Invoice No: #{{ $invoice->invoice_no }} - {{ date('d-m-Y', strtotime($invoice->date)) }} </h4>
28 |         <a href="{{ route('invoice.pending.list') }}" class="btn btn-dark btn-rounded waves-effect waves-light" style="float:right;"><i class="fa fa-list"> Pending Invoice List </i></a> <br> <br>
29 |
30 |         <table class="table table-dark" width="100%">
31 |           <tbody>
32 |             <tr>
33 |               <td><p> Customer Info </p></td>
34 |               <td><p> Name: <strong> {{ $payment['customer']['name'] }} </strong> </p></td>
35 |               <td><p> Mobile: <strong> {{ $payment['customer']['mobile_no'] }} </strong> </p></td>
36 |               <td><p> Email: <strong> {{ $payment['customer']['email'] }} </strong> </p></td>
37 |             </tr>
38 |
39 |             <tr>
40 |               <td></td>
41 |               <td colspan="3"><p> Description : <strong> {{ $invoice->description }} </strong> </p></td>
42 |             </tr>
43 |           </tbody>
44 |
45 |         </table>
46 |
47 |
48 |         <form method="post" action="{{ route('approval.store', $invoice->id) }}>
49 |           @csrf
50 |           <table border="1" class="table table-dark" width="100%">
51 |             <thead>
52 |               <tr>
53 |                 <th class="text-center">Sl</th>
54 |                 <th class="text-center">Category</th>
55 |                 <th class="text-center">Product Name</th>
56 |                 <th class="text-center" style="background-color: #800000; color: white">Current Stock</th>
57 |                 <th class="text-center">Quantity</th>
58 |                 <th class="text-center">Unit Price</th>
59 |                 <th class="text-center">Total Price</th>
60 |               </tr>
61 |
62 |             </thead>
63 |             <tbody>
64 |               @php
65 |                 $total_sum = '0';
66 |               @endphp
67 |               @foreach($invoice['invoice_details'] as $key => $details)
68 |                 <tr>
69 |

```

```

resources > views > backend > invoice >  invoice_pending_list.blade.php
25 |
26 |         <a href="{{ route('invoice.add') }}" class="btn btn-dark btn-rounded waves-effect waves-light" style="float:right;"><i class="fas fa-plus-circle"> Add Inovice </i></a> <br> <br>
27 |
28 |         <h4 class="card-title">Inovice All Data </h4>
29 |
30 |         <table id="datatable" class="table table-bordered dt-responsive nowrap" style="border-collapse: collapse; border-spacing: 0; width: 100%;>
31 |           <thead>
32 |             <tr>
33 |               <th>Sl</th>
34 |               <th>Customer Name</th>
35 |               <th>Invoice No </th>
36 |               <th>Date </th>
37 |               <th>Desctipion</th>
38 |               <th>Amount</th>
39 |               <th>Status</th>
40 |               <th>Action</th>
41 |
42 |             </thead>
43 |
44 |             <tbody>
45 |               @foreach($allData as $key => $item)
46 |
47 |                 <tr>
48 |                   <td> {{ $key+1}} </td>
49 |                   <td> {{ $item['payment']['customer']['name'] }} </td>
50 |                   <td> #{{ $item->invoice_no }} </td>
51 |                   <td> {{ date("d-m-Y", strtotime($item->date)) }} </td>
52 |
53 |                   <td> {{ $item->description }} </td>
54 |                   <td> {{ $item['payment']['total_amount'] }} </td>
55 |
56 |                   <td> @if($item->status == '0')
57 |                     <span class="btn btn-warning">Pending</span>
58 |                   @elseif($item->status == '1')
59 |                     <span class="btn btn-success">Approved</span>
60 |                   @endif </td>
61 |
62 |                   <td> @if($item->status == '0')
63 |                     <a href="{{ route('invoice.approve', $item->id) }}" class="btn btn-dark sm" title="Approved Data" > <i class="fas fa-check-circle"></i> </a>
64 |                   @endif </td>
65 |
66 |                   <td> @if($item->status == '0')
67 |                     <a href="{{ route('invoice.delete', $item->id) }}" class="btn btn-danger sm" title="Delete Data" id="delete" > <i class="fas fa-trash-alt"></i> </a>
68 |                   @endif </td>
69 |
70 |                 </tr>
71 |               @endforeach

```

```

resources > views > backend > invoice > print_invoice_list.blade.php
-->
21  <div class="row">
22    <div class="col-12">
23      <div class="card">
24        <div class="card-body">
25
26        <a href="{{ route('invoice.add') }}" class="btn btn-dark btn-rounded waves-effect waves-light" style="float:right;"><i class="fas fa-plus-circle"> Add Invoice </i></a> <br> <br>
27
28        <h4 class="card-title">Invoice All Data </h4>
29
30
31        <table id="datatable" class="table table-bordered dt-responsive nowrap" style="border-collapse: collapse; border-spacing: 0; width: 100%;">
32          <thead>
33            <tr>
34              <th>Sl</th>
35              <th>Customer Name</th>
36              <th>Invoice No </th>
37              <th>Date </th>
38              <th>Desctipion</th>
39              <th>Amount</th>
40              | <th>Action</th>
41
42            </thead>
43
44
45            <tbody>
46
47            @foreach($allData as $key => $item)
48
49            <tr>
50              <td> {{ $key+1}} </td>
51              <td> {{ $item['payment']['customer']['name'] }} </td>
52              <td> #{{ $item->invoice_no }} </td>
53              <td> {{ date('d-m-Y',strtotime($item->date)) }} </td>
54
55              <td> {{ $item->description }} </td>
56
57              <td> {{ $item['payment']['total_amount'] }} </td>
58
59              <td>
60                <a href="{{ route('print.invoice',$item->id) }}" class="btn btn-danger sm" title="Print Invoice" > <i class="fa fa-print"></i></a>
61                </td>
62
63            </tr>
64
65            @endforeach

```

Product:

- At first, we implemented here about adding products, which includes the fields of Product name, Supplier name(where suppliers are selected from a drop down box, mainly registered suppliers), Unit name(from drop down box), Category name.
- Then, it shows the product list, where serial no., name, supplier name, unit, category and actions will be shown.
- Edit, Update and Delete products from product list. If there's anything that needs to be changed or deleted, we can do it here.
- The editing page has some required fields(e.g. Product name, supplier, unit, category), which must be given, otherwise it'll be an invalid input.

```
resources > views > backend > product > 📄 product_add.blade.php
13 |     <h4 class="card-title">Add Product Page </h4><br><br>
14 |
15 |
16 |
17 <form method="post" action="{{ route('product.store') }}" id="myForm" >
18 |     @csrf
19 |
20 |     <div class="row mb-3">
21 |         <label for="example-text-input" class="col-sm-2 col-form-label">Product Name </label>
22 |         <div class="form-group col-sm-10">
23 |             <input name="name" class="form-control" type="text" >
24 |         </div>
25 |     </div>
26 |     <!-- end row -->
27 |
28 |
29 |     <div class="row mb-3">
30 |         <label class="col-sm-2 col-form-label">Supplier Name </label>
31 |         <div class="col-sm-10">
32 |             <select name="supplier_id" class="form-select" aria-label="Default select example">
33 |                 <option selected="">Open this select menu</option>
34 |                 @foreach($supplier as $supp)
35 |                     <option value="{{ $supp->id }}>{{ $supp->name }}</option>
36 |                 @endforeach
37 |             </select>
38 |         </div>
39 |     </div>
40 |     <!-- end row -->
41 |
42 |     <div class="row mb-3">
43 |         <label class="col-sm-2 col-form-label">Unit Name </label>
44 |         <div class="col-sm-10">
45 |             <select name="unit_id" class="form-select" aria-label="Default select example">
46 |                 <option selected="">Open this select menu</option>
47 |                 @foreach($unit as $uni)
48 |                     <option value="{{ $uni->id }}>{{ $uni->name }}</option>
49 |                 @endforeach
50 |             </select>
51 |         </div>
52 |     </div>
53 |     <!-- end row -->
54 |
55 |
56 |
57 |     <div class="row mb-3">
58 |         <label class="col-sm-2 col-form-label">Category Name </label>
59 |         <div class="col-sm-10">
60 |             <select name="category_id" class="form-select" aria-label="Default select example">
```

```

resources > views > backend > product >  product_all.blade.php
27
28 |     <h4 class="card-title">Product All Data </h4>
29
30
31     <table id="datatable" class="table table-bordered dt-responsive nowrap" style="border-collapse: collapse; border-spacing: 0; width: 100%;>
32         <thead>
33             <tr>
34                 <th>Sl</th>
35                 <th>Name</th>
36                 <th>Supplier Name</th>
37                 <th>Unit</th>
38                 <th>Category</th>
39                 <th>Action</th>
40
41         </thead>
42
43         <tbody>
44
45             @foreach($product as $key => $item)
46             <tr>
47                 <td> {{ $key+1}} </td>
48                 <td> {{ $item->name }} </td>
49                 <td>
50                     
51                     @if($item->supplier && $item->supplier->trashed())
52                         <span class="text-danger">Supplier Deleted</span>
53                     @else
54                         {{ $item->supplier ? $item->supplier->name : 'N/A' }}
55                     @endif
56                 </td>
57                 <td> {{ $item->unit ? $item->unit->name : 'N/A' }} </td>
58                 <td> {{ $item->category ? $item->category->name : 'N/A' }} </td>
59                 <td>
60                     <a href="{{ route('product.edit',$item->id) }}" class="btn btn-info sm" title="Edit Data" >  </a>
61
62                     <a href="{{ route('product.delete',$item->id) }}" class="btn btn-danger sm" title="Delete Data" id="delete" >  </a>
63                 </td>
64             </tr>
65         @endforeach
66
67
68
69

```

```

resources > views > backend > product >  product_edit.blade.php
13
14 |     <h4 class="card-title">Edit Product Page </h4><br><br>
15
16
17     <form method="post" action="{{ route('product.update') }}" id="myForm" >
18         @csrf
19
20         <input type="hidden" name="id" value="{{ $product->id }}>
21
22         <div class="row mb-3">
23             <label for="example-text-input" class="col-sm-2 col-form-label">Product Name </label>
24             <div class="col-sm-10">
25                 <input name="name" value="{{ $product->name }}" class="form-control" type="text" >
26             </div>
27         <!-- end row -->
28
29
30
31         <div class="row mb-3">
32             <label class="col-sm-2 col-form-label">Supplier Name </label>
33             <div class="col-sm-10">
34                 <select name="supplier_id" class="form-select" aria-label="Default select example">
35                     <option selected="">Open this select menu</option>
36                     @foreach($supplier as $supp)
37                         <option value="{{ $supp->id }}" {{ $supp->id == $product->supplier_id ? 'selected' : '' }} >{{ $supp->name }}</option>
38                     @endforeach
39                 </select>
40             </div>
41         <!-- end row -->
42
43
44         <div class="row mb-3">
45             <label class="col-sm-2 col-form-label">Unit Name </label>
46             <div class="col-sm-10">
47                 <select name="unit_id" class="form-select" aria-label="Default select example">
48                     <option selected="">Open this select menu</option>
49                     @foreach($unit as $uni)
50                         <option value="{{ $uni->id }}" {{ $uni->id == $product->unit_id ? 'selected' : '' }} >{{ $uni->name }}</option>
51                     @endforeach
52                 </select>
53             </div>
54         <!-- end row -->
55
56
57
58
59         <div class="row mb-3">
60             <label class="col-sm-2 col-form-label">Category Name </label>
61
62
63
64
65
66
67
68
69

```

```
resources > views > backend > product > product_edit.blade.php
```

```
87
88     <script type="text/javascript">
89         $(document).ready(function (){
90             $('#myForm').validate({
91                 rules: {
92                     name: {
93                         required : true,
94                     },
95                     supplier_id: {
96                         required : true,
97                     },
98                     unit_id: {
99                         required : true,
100                    },
101                     category_id: {
102                         required : true,
103                     },
104                 },
105                 messages :{
106                     name: {
107                         required : 'Please Enter Your Product Name',
108                     },
109                     supplier_id: {
110                         required : 'Please Select One Supplier',
111                     },
112                     unit_id: {
113                         required : 'Please Select One Unit',
114                     },
115                     category_id: {
116                         required : 'Please Select One Category',
117                     },
118                 },
119                 errorElement : 'span',
120                 errorPlacement: function (error,element) {
121                     error.addClass('invalid-feedback');
122                     element.closest('.form-group').append(error);
123                 },
124                 highlight : function(element, errorClass, validClass){
125                     $(element).addClass('is-invalid');
126                 },
127                 unhighlight : function(element, errorClass, validClass){
128                     $(element).removeClass('is-invalid');
129                 },
130             });
131         });
132
133     </script>
134
```

Purchase:

- Purchasing products
- Daily purchase reports
- Purchases adding, where the required fields are Category, product name, units, unit price, description of the product, total price and amount. Then the quantity of the products is also required.

```

resources > views > backend > purchase > daily_purchase_report.blade.php
14
15 <form method="GET" action="{{ route('daily.purchase.pdf') }}" target="_blank" id="myForm">
16   <div class="row">
17
18
19
20     <div class="col-md-4">
21       <div class="md-3 form-group">
22         <label for="example-text-input" class="form-label">Start Date</label>
23         <input class="form-control example-date-input" name="start_date" type="date" id="start_date" placeholder="YY-MM-DD">
24       </div>
25     </div>
26
27
28     <div class="col-md-4">
29       <div class="md-3 form-group">
30         <label for="example-text-input" class="form-label">End Date</label>
31         <input class="form-control example-date-input" name="end_date" type="date" id="end_date" placeholder="YY-MM-DD">
32       </div>
33     </div>
34
35     <div class="col-md-4">
36       <div class="md-3">
37         <label for="example-text-input" class="form-label" style="margin-top:43px;"> </label>
38         <button type="submit" class="btn btn-info">Search</button>
39       </div>
40     </div>
41
42
43
44   </div> <!-- // end row -->
45
46   </form>
47
48   </div> <!-- End card-body -->
49
50
51
52   </div>
53 </div> <!-- end col -->
54 </div>

```

```
resources > views > backend > purchase > daily_purchase_report.blade.php
xx
62  <script type="text/javascript">
63      $(document).ready(function (){
64          $('#myForm').validate({
65              rules: {
66                  start_date: {
67                      required : true,
68                  },
69                  end_date: {
70                      required : true,
71                  },
72              },
73              messages :{
74                  start_date: {
75                      required : 'Please Select Start Date',
76                  },
77                  end_date: {
78                      required : 'Please Select End Date',
79                  },
80              },
81              errorElement : 'span',
82              errorPlacement: function (error,element) {
83                  error.addClass('invalid-feedback');
84                  element.closest('.form-group').append(error);
85              },
86              highlight : function(element, errorClass, validClass){
87                  $(element).addClass('is-invalid');
88              },
89              unhighlight : function(element, errorClass, validClass){
90                  $(element).removeClass('is-invalid');
91              },
92          });
93      });
94  );
95  );
96
97 </script>
98
99
100
101 @endsection
```

```
resources > views > backend > purchase >  purchase_add.blade.php
83
84     <div class="card-body">
85         <form method="post" action="{{ route('purchase.store') }}>
86             @csrf
87             <table class="table-sm table-bordered" width="100%" style="border-color: #dd;">
88                 <thead>
89                     <tr>
90                         <th>Category</th>
91                         <th>Product Name</th>
92                         <th>PSC/KG</th>
93                         <th>Unit Price</th>
94                         <th>Description</th>
95                         <th>Total Price</th>
96                         <th>Action</th>
97
98                     </tr>
99                 </thead>
100
101                 <tbody id="addRow" class="addRow">
102
103                     </tbody>
104
105                 <tbody>
106                     <tr>
107                         <td colspan="5"></td>
108                         <td>
109                             <input type="text" name="estimated_amount" value="0" id="estimated_amount" class="form-control estimated_amount" readonly style="background-color: #ddd;">
110                         </td>
111                         <td></td>
112                     </tr>
113
114                 </tbody>
115             </table><br>
116             <div class="form-group">
117                 <button type="submit" class="btn btn-info" id="storeButton"> Purchase Store </button>
118
119             </div>
120
121         </form>
```

```

resources > views > backend > purchase > 📄 purchase_pending.blade.php
27 |     |     |     |     <h4 class="card-title">Purchase All Pending Data </h4>
28 |
29 |
30 |
31 |     <table id="datatable" class="table table-bordered dt-responsive nowrap" style="border-collapse: collapse; border-spacing: 0; width: 100%;"
32 |         <thead>
33 |             <tr>
34 |                 <th>Sl</th>
35 |                 <th>Purchase No</th>
36 |                 <th>Date </th>
37 |                 <th>Supplier</th>
38 |                 <th>Category</th>
39 |                 <th>Qty</th>
40 |                 <th>Product Name</th>
41 |                 <th>Status</th>
42 |                 <th>Action</th>
43 |
44 |             </thead>
45 |
46 |
47 |             <tbody>
48 |
49 |                 @foreach($allData as $key => $item)
50 |             <tr>
51 |                 <td> {{ $key+1}} </td>
52 |                 <td> {{ $item->purchase_no }} </td>
53 |                 <td> {{ date('d-m-Y',strtotime($item->date)) }} </td>
54 |                 <td> {{ $item['supplier']['name'] }} </td>
55 |                 <td> {{ $item['category']['name'] }} </td>
56 |                 <td> {{ $item->buying_qty }} </td>
57 |                 <td> {{ $item['product']['name'] }} </td>
58 |
59 |                 <td>
60 |                     @if($item->status == '0')
61 |                         <span class="btn btn-warning">Pending</span>
62 |                     @elseif($item->status == '1')
63 |                         <span class="btn btn-success">Approved</span>
64 |                     @endif
65 |                 </td>
66 |
67 |             <td>
68 |                 @if($item->status == '0')
69 |                     <a href="{{ route('purchase.approve',$item->id) }}" class="btn btn-danger sm" title="Approved" id="ApproveBtn"> <i class="fas fa-check-circle"></i> </a>
70 |                 @endif
71 |             </td>
72 |
73 |             </tr>
74 |         @endforeach
75 |

```

Stock:

- Shows how many products are in the inventory Stock.
- Generate Stock report.
- Print Stock report.
- Supplier and product wise report.

```

resources > views > backend > stock > stock_report.blade.php
20
21     <div class="row">
22         <div class="col-12">
23             <div class="card">
24                 <div class="card-body">
25
26             <a href="{{ route('stock.report.pdf') }}" target="_blank" class="btn btn-dark btn-rounded waves-effect waves-light" style="float:right;"><i class="fa fa-print"> Stock Report Print </i></a>
27
28             <h4 class="card-title">Stock Report </h4>
29
30
31             <table id="datatable" class="table table-bordered dt-responsive nowrap" style="border-collapse: collapse; border-spacing: 0; width: 100%;>
32                 <thead>
33                     <tr>
34                         <th>Sl</th>
35                         <th>Supplier Name </th>
36                         <th>Unit</th>
37                         <th>Category</th>
38                         <th>Product Name</th>
39                         <th>In Qty</th>
40                         <th>Out Qty </th>
41                         <th>Stock </th>
42
43                     </thead>
44
45                     <tbody>
46
47                         @foreach($allData as $key => $item)
48
49 @php
50     $buying_total = App\Models\Purchase::where('category_id',$item->category_id)->where('product_id',$item->id)->where('status', '1')->sum('buying_qty');
51
52     $selling_total = App\Models\InvoiceDetail::where('category_id',$item->category_id)->where('product_id',$item->id)->where('status', '1')->sum('selling_qty');
53 @endphp
54
55             <tr>
56                 <td> {{ $key+1}} </td>
57                 <td> {{ $item['supplier']['name'] }} </td>
58                 <td> {{ $item['unit']['name'] }} </td>
59                 <td> {{ $item['category']['name'] }} </td>
60                 <td> {{ $item->name }} </td>
61                 <td> <span class="btn btn-success"> {{ $buying_total }} </span> </td>
62                 <td> <span class="btn btn-info"> {{ $selling_total }} </span> </td>
63                 <td> <span class="btn btn-danger"> {{ $item->quantity }} </span> </td>
64
65             </tr>
66
67         @endforeach

```

```

resources > views > backend > stock > 📈 supplier_product_wise_report.blade.php
59
60 40  <!-- /// Supplier Wise -->
61 41  <div class="show_supplier" style="display:none">
62 42  |   <form method="GET" action="{{ route('supplier.wise.pdf') }}" id="myForm" target="_blank" >
63 43
64 44  |       <div class="row">
65 45  |           <div class="col-sm-8 form-group">
66 46  |               |   <label>Supplier Name </label>
67 47  |               <select name="supplier_id" class="form-select select2" >
68 48  |                   <option value="">Select Supplier</option>
69 49  |                   @foreach($suppliers as $supp)
70 50  |                       <option value="{{ $supp->id }}">{{ $supp->name }}{{ $cat->name }}

```

Suppliers:

- Show suppliers information.
- Edit/Update Suppliers information.
- Add new suppliers.

```
resources > views > backend > supplier >  supplier_add.blade.php
21 | | | | | <!-- Supplier Name -->
22 | | | | | <div class="row mb-3">
23 | | | | |   <label for="name" class="col-sm-2 col-form-label">Supplier Name</label>
24 | | | | |   <div class="form-group col-sm-10">
25 | | | | |     <input name="name" class="form-control" type="text" value="{{ old('name') }}"/>
26 | | | | |     @error('name')
27 | | | | |       <span class="text-danger">{{ $message }}</span>
28 | | | | |     @enderror
29 | | | | |   </div>
30 | | | | </div>
31 | | | | <!-- end Supplier Name -->
32 |
33 | | | | <!-- Supplier Mobile -->
34 | | | | <div class="row mb-3">
35 | | | |   <label for="mobile_no" class="col-sm-2 col-form-label">Supplier Mobile</label>
36 | | | |   <div class="form-group col-sm-10">
37 | | | |     <input name="mobile_no" class="form-control" type="text" value="{{ old('mobile_no') }}"/>
38 | | | |     @error('mobile_no')
39 | | | |       <span class="text-danger">{{ $message }}</span>
40 | | | |     @enderror
41 | | | |   </div>
42 | | | </div>
43 | | | <!-- end Supplier Mobile -->
44 |
45 | | | <!-- Supplier Email -->
46 | | | <div class="row mb-3">
47 | | |   <label for="email" class="col-sm-2 col-form-label">Supplier Email</label>
48 | | |   <div class="form-group col-sm-10">
49 | | |     <input name="email" class="form-control" type="email" value="{{ old('email') }}"/>
50 | | |     @error('email')
51 | | |       <span class="text-danger">{{ $message }}</span>
52 | | |     @enderror
53 | | |   </div>
54 | | </div>
55 | | <!-- end Supplier Email -->
56 |
57 | | <!-- Supplier Address -->
58 | | <div class="row mb-3">
59 | |   <label for="address" class="col-sm-2 col-form-label">Supplier Address</label>
60 | |   <div class="form-group col-sm-10">
61 | |     <input name="address" class="form-control" type="text" value="{{ old('address') }}"/>
62 | |     @error('address')
63 | |       <span class="text-danger">{{ $message }}</span>
64 | |     @enderror
65 | |   </div>
66 | </div>
67 | <!-- end Supplier Address -->
68 |
```

```

resources > views > backend > supplier > 🏠 supplier_all.blade.php
26   <a href="{{ route('supplier.add') }}" class="btn btn-dark btn-rounded waves-effect waves-light" style="float:right;">Add Supplier </a> <br> <br>
27
28   <h4 class="card-title">Supplier All Data </h4>
29
30
31   <table id="datatable" class="table table-bordered dt-responsive nowrap" style="border-collapse: collapse; border-spacing: 0; width: 100%;">
32     <thead>
33       <tr>
34         <th>Sl</th>
35         <th>Name</th>
36         <th>Mobile Number </th>
37         <th>Email</th>
38         <th>Address</th>
39         <th>Action</th>
40
41     </thead>
42
43
44     <tbody>
45
46       @foreach($suppliers as $key => $item)
47
48         <tr>
49           <td> {{ $key+1}} </td>
50           <td> {{ $item->name }} </td>
51           <td> {{ $item->mobile_no }} </td>
52           <td> {{ $item->email }} </td>
53           <td> {{ $item->address }} </td>
54           <td>
55             <a href="{{ route('supplier.edit',$item->id) }}" class="btn btn-info sm" title="Edit Data" > <i class="fas fa-edit"></i> </a>
56
57             <a href="{{ route('supplier.delete',$item->id) }}" class="btn btn-danger sm" title="Delete Data" id="delete" > <i class="fas fa-trash-alt"></i> </a>
58
59           </td>
60
61         </tr>
62       @endforeach
63
64     </tbody>
65
66   </table>
67
68   <!-- end col -->
69   <!-- end row -->

```

```
resources > views > backend > supplier > 📄 supplier_edit.blade.php
12 | |
13 |     <h4 class="card-title">Add Supplier Page </h4><br><br>
14 |
15 |
16 |
17 |     <form method="post" action="{{ route('supplier.update') }}" id="myForm" >
18 |         @csrf
19 |
20 |         <input type="hidden" name="id" value="{{ $supplier->id }}>
21 |
22 |         <div class="row mb-3">
23 |             <label for="example-text-input" class="col-sm-2 col-form-label">Supplier Name </label>
24 |             <div class="form-group col-sm-10">
25 |                 <input name="name" class="form-control" value="{{ $supplier->name }}" type="text" >
26 |             </div>
27 |         </div>
28 |         <!-- end row -->
29 |
30 |
31 |         <div class="row mb-3">
32 |             <label for="example-text-input" class="col-sm-2 col-form-label">Supplier Mobile </label>
33 |             <div class="form-group col-sm-10">
34 |                 <input name="mobile_no" value="{{ $supplier->mobile_no }}" class="form-control" type="text" >
35 |             </div>
36 |         </div>
37 |         <!-- end row -->
38 |
39 |
40 |         <div class="row mb-3">
41 |             <label for="example-text-input" class="col-sm-2 col-form-label">Supplier Email </label>
42 |             <div class="form-group col-sm-10">
43 |                 <input name="email" value="{{ $supplier->email }}" class="form-control" type="email" >
44 |             </div>
45 |         </div>
46 |         <!-- end row -->
47 |
48 |
49 |         <div class="row mb-3">
50 |             <label for="example-text-input" class="col-sm-2 col-form-label">Supplier Address </label>
51 |             <div class="form-group col-sm-10">
52 |                 <input name="address" value="{{ $supplier->address }}" class="form-control" type="text" >
53 |             </div>
54 |         </div>
55 |         <!-- end row -->
56 | 
```

Unit:

- Show units.
- Add a new unit.
- Edit, Update and Delete products from product list.

```

resources > views > backend > unit >  unit_add.blade.php
 8  <div class="page-content">
 9    <div class="container-fluid">
10      <div class="row">
11        <div class="col-12">
12          <div class="card">
13            <div class="card-body">
14              <h4 class="card-title">Add Unit Page</h4><br><br>
15
16              <form method="post" action="{{ route('unit.store') }}" id="myForm">
17                @csrf
18
19                  <!-- Unit Name -->
20                  <div class="row mb-3">
21                    <label for="name" class="col-sm-2 col-form-label">Unit Name</label>
22                    <div class="form-group col-sm-10">
23                      <input name="name" class="form-control" type="text" value="{{ old('name') }}"/>
24                      @error('name')
25                        <span class="text-danger">{{ $message }}

```

```

resources > views > backend > unit > 📄 unit_all.blade.php
22 |     <div class="col-12">
23 |         <div class="card">
24 |             <div class="card-body">
25 |
26     <a href="{{ route('unit.add') }}" class="btn btn-dark btn-rounded waves-effect waves-light" style="float:right;">Add Unit </a> <br> <br>
27
28     <h4 class="card-title">Unit All Data </h4>
29
30
31     <table id="datatable" class="table table-bordered dt-responsive nowrap" style="border-collapse: collapse; border-spacing: 0; width: 100%;">
32         <thead>
33             <tr>
34                 <th width="5%">Sl</th>
35                 <th>Name</th>
36                 <th width="20%">Action</th>
37
38             </thead>
39
40         <tbody>
41
42             @foreach($units as $key => $item)
43
44             <tr>
45                 <td> {{ $key+1}} </td>
46                 <td> {{ $item->name }} </td>
47                 <td>
48                     <a href="{{ route('unit.edit',$item->id) }}" class="btn btn-info sm" title="Edit Data"> <i class="fas fa-edit"></i> </a>
49
50                     <a href="{{ route('unit.delete',$item->id) }}" class="btn btn-danger sm" title="Delete Data" id="delete"> <i class="fas fa-trash-alt"></i> </a>
51
52                 </td>
53             </tr>
54         @endforeach
55
56     </tbody>
57     </table>
58
59
60     </div>
61     </div>
62     </div> <!-- end col -->
63     </div> <!-- end row -->
64
65     </div> <!-- container-fluid -->
66
67
68 @endsection

```

```
resources > views > backend > unit > 🗑 unit_edit.blade.php
22 | | | | <label for="example-text-input" class="col-sm-2 col-form-label">Unit Name </label>
23 | | | | <div class="form-group col-sm-10">
24 | | | |   <input name="name" value="{{ $unit->name }}" class="form-control" type="text" >
25 | | | | </div>
26 | | | </div>
27 | | | <!-- end row -->
28 |
29
30 <input type="submit" class="btn btn-info waves-effect waves-light" value="Update Unit">
31 | | </form>
32 |
33
34
35 | | </div>
36 | </div>
37 </div> <!-- end col -->
38 </div>
39 </div>
40 </div>
41
42 <script type="text/javascript">
43 | | $(document).ready(function (){
44 | |   $('#myForm').validate({
45 | |     rules: {
46 | |       name: {
47 | |         required : true,
48 | |       },
49 |
50 | |     },
51 | |     messages :{
52 | |       name: {
53 | |         required : 'Please Enter Your Name',
54 | |       },
55 |
56 | |     },
57 | |     errorElement : 'span',
58 | |     errorPlacement: function (error,element) {
59 | |       error.addClass('invalid-feedback');
60 | |       element.closest('.form-group').append(error);
61 | |     },
62 | |     highlight : function(element, errorClass, validClass){
63 | |       $(element).addClass('is-invalid');
64 | |     },
65 | |     unhighlight : function(element, errorClass, validClass){
66 | |       $(element).removeClass('is-invalid');
67 | |     },
68 | |   });
69 | });
70 |});
```

Technology (Framework, Languages)

Framework: Laravel 9

Languages: PHP, Javascript, HTML, CSS, MySql.

Github Repository

Link: https://github.com/farhanfaruk007/CSE470_Project

Individual Contribution

ID	Name	Contribution
20301263	Emam Hasan	<ul style="list-style-type: none"> ● User Authentication <ul style="list-style-type: none"> ○ Change Password ● Profile Management <ul style="list-style-type: none"> ○ View Profile ○ Edit Profile ● Whole User Interface ● Order Overview <ul style="list-style-type: none"> ○ Order History ○ Add Order ○ Order Status ○ Approve Order ○ Search Order ○ Print Order Invoice ○ Total order report ○ Download PDF
20301137	Farhan Faruk	<ul style="list-style-type: none"> ● Track <ul style="list-style-type: none"> ○ Total Supplier Count ○ New Customer Count ○ Total Customer Count ○ Total Units Count ○ Total Category Count ○ Total Product Count ○ Approved Purchases Count ○ Pending Purchases Count ○ Approved Order Count ○ Pending Order Count ● Product Overview <ul style="list-style-type: none"> ○ Show All Products ○ Add New Products

		<ul style="list-style-type: none"> ○ Edit/Delete Products ● Purchase overview <ul style="list-style-type: none"> ○ Purchase History ○ Add Purchase ○ Purchase Status ○ Search Purchase ○ Approved purchases ○ Pending purchases ○ Print Purchase List ○ Total Purchase Report ○ Download PDF
20301270	Nafisa Ahmed Progga	<ul style="list-style-type: none"> ● Unit Overview <ul style="list-style-type: none"> ○ Show all Unit Data. ○ Add Unit. ○ Edit/Delete Unit. ● Category overview <ul style="list-style-type: none"> ○ Show all categories. ○ Add Category. ○ Edit/Delete Category.
19301074	Sharmin Akter Keya	<ul style="list-style-type: none"> ● Supplier Overview <ul style="list-style-type: none"> ○ Show all supplier data. ○ Add Supplier. ○ Edit/Delete Supplier. ● Customer Overview <ul style="list-style-type: none"> ○ Show all customer data. ○ Add Customer. ○ Edit/Delete Customer.