

**Point of sales and Inventory Management System**

**BY**

**UZZAL HOSEN**

ID: 161-15-652

**AND**

**MOST. SHADIA AFRIN**

ID: 183- 15-1010

This Report Presented in Partial Fulfillment of the Requirements for the Degree of  
Bachelor of Science in Computer Science and Engineering

Supervised By

**NAZNIN SULTANA**

Assistant Professor

Department of CSE

Daffodil International University

Co-Supervised By

**MD SABAB ZULFIKER**

Lecturer (Senior Scale)

Department of CSE

Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY**

**DHAKA, BANGLADESH**

**SEPTEMBER 2022**

## APPROVAL

This project titled “**Point of sales and Inventory Management System**”, submitted by Uzzal Hosen, ID:161-15-652 and Most. Shadia Afrin, ID:183-15-1010 to the department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 13<sup>th</sup> September 2022.

### BOARD OF EXAMINERS



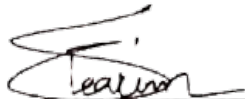
**Dr. S M Aminul Haque**  
**Associate Professor & Associate Head**  
Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Chairman**



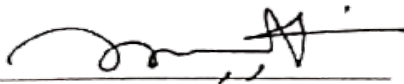
**Fahad Faisal**  
**Assistant Professor**  
Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Internal Examiner**



**Fourcan Karim Mazumder**  
**Assistant Professor**  
Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Internal Examiner**



**Dr. Mohammad Shorif Uddin**  
**Professor**  
Department of Computer Science and Engineering  
Jahangirnagar University

**External Examiner**

## DECLARATION

We hereby declare that, this project has been done by us under the supervision of **NAZNIN SULTANA, Assistant Professor, Department of CSE, Daffodil International University**. We also declared that neither this project nor any part of this project has been submitted elsewhere for award of any degree.

### Supervised by:



---

**Naznin Sultana**  
Assistant Professor  
Department of CSE  
Daffodil International University

### Co-Supervised by:



---

**Md. Sabab Zulfiker**  
Lecturer (Senior Scale)  
Department of CSE  
Daffodil International University

### Submitted by:



---

**Uzzal Hosen**  
ID: 161-15-652  
Department of CSE  
Daffodil International University



---

**Most. Shadia Afrin**  
ID: 183-15-1010  
Department of CSE  
Daffodil International University

## ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project successfully

We really grateful and wish our profound indebtedness to **Naznin Sultana, Assistant Professor**, Department of CSE Daffodil International University, Dhaka. Deep knowledge & keen interest of our supervisor in the field of web application to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project. We would like to express our heartiest gratitude to Pro. Dr. Touhid Bhuiyan, Head of Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

## **ABSTRACT**

This project “Point of sales and Inventory Management System” User of this application will not only get solid information and report on their customers but also will witness one of the most beautiful application in market for customer. Our documentation can be used as a guide for future engineering students who are willing to make a web application. This project very different type of analysis has been made due to the development of the system. There is feasibility study and system requirement information collection which is necessary prior to development. we have used real time charts for customer. Anyone can use its code that are giving in this documentation for customer.

## TABLE OF CONTENTS

<b>CONTENTS</b>	<b>PAGE</b>
Approval	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
<b>CHAPTER 1: INTRODUCTION</b>	<b>1-3</b>
1.1 Introduction	1
1.2 Motivation	1
1.3 Objective	2
1.4 Expected Outcomes	2
1.5 Project Management and Finance	2
1.6 Report Layout	3
<b>CHAPTER 2: BACKGROUND</b>	<b>4</b>
2.1 Preliminaries/Terminologies	4
2.2 Comparative Analysis	4
2.3 Scope of The Problem	4

<b>CHAPTER 3: REQUIREMENT SPECIFICATION</b>	<b>5-8</b>
3.1 Requirement Management	5
3.2 Requirement Collection	5
3.3 Visual Studio Code	5
3.4 PHP	6
3.5 Xampp	6
3.6 Use Case Modeling and Description	7
3.7 Use Diagram	7
3.8 Design Requirements	8
<b>CHAPTER 4: DESIGN SPECIFICATION</b>	<b>9-14</b>
4.1 Front-end Design	9
4.2 Home Page	10
4.3 Interacting Design and User Experience (UX)	10
4.4 Admin Panel	11-14
4.5 Testing Implementation	14
<b>CHAPTER 5: IMPLEMENTATION AND TESTING</b>	<b>15-21</b>
5.1 Introduction	15
5.2 Implementation of database	15-19
5.3 Testing Implementation	20
5.4 Implementation of Font-End Design	21
5.5 Test Results and Reports	21

<b>CHAPTER 6: IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY</b>	<b>22</b>
6.1 Influence of Society	22
6.2 Effects on The Environment	22
5.3 Testing Implementation	22
6.3 Ethical Aspects	22
6.4 Sustainability	22
<b>CHAPTER 7: CONCLUSION AND FUTURE SCOPE</b>	<b>23</b>
7.1 Discussion and Conclusion	23
7.2 Scope for Future Development	24
<b>REFERENCES</b>	<b>25</b>
<b>PLAGIARISM REPORT</b>	<b>26</b>



## LIST OF FIGURES

<b>FIGURES</b>	<b>PAGE</b>
Figure 3.7: Use Diagram	7
Figure 4.3.1: Login	10
Figure 4.3.2: Admin Login	11
Figure 4.3.3: Sales Page	12
Figure 4.3.4: Products	12
Figure 4.3.5 Add Stock	13
Figure 4.3.6 Reports	13
Figure 4.3.7 Transactions	14
Figure 5.2.1: Database connection	15
Figure 5.2.2: Product Pack	16
Figure 5.2.3: User registration	17
Figure 5.2.4: User login	18
Figure 5.2.5: Seller	19
Figure 5.3: Testing Implementation	20

## LIST OF TABLES

<b>TABLE</b>	<b>PAGE</b>
Table 4.5: Register	14

# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction

Point of sales and Inventory Management System. which is a web-based application. From this web application customer will be able to shop easily. The most amazing thing about this site is that every instruction plus description will be in Bangla mean our own language. There is a feature for changing language if anyone prefers with English customer can choose any product. Customer will easily buy a product from our shop. If any register customer can choose any kind of product and pay with card and buy this product. Finally, there can get a product. In this web app are two types of users: normal users, admin. admin can see the whole process such as total product, total employees, total shop number. A normal user's after registration login and see all product and select and pay and finally gets this product. Admin's can control the whole system.

### 1.2 Motivation

Nowadays people who are depend on online shop they are much dependent on websites like online productive, any kind of site that is related to online shop. In the context of Bangladesh now maximum people depended on online. where customer have to e-online shop. Now the main problem is most of the people cannot find the flexible time for shop in this site help us very comfortable of our country. for this people can easily buy product without jam or problem of our country their basic foundation of shop. if any customer want deposit or return product they can. And if we notice that most of the people benefit for this process and they can easily complete shop. Admin or shopkeeper always help that can help those kinds of customer who are facing any kind of problem. So, customer will easily what they need to product with paying money and with card much data.

### **1.3 Objectives**

As the available existing system provides limited functions to the user, thus this project will contain enhanced and more flexible functions to the store. The objectives include While developing this project i kept focus on some objectives that's are Reduces customerstime.

- ❖ To provide good support for the especially job holder customers who haven't time
- ❖ To provide good quality of product.
- ❖ To user-friendly interface is one of the main objectives of this project.
- ❖ To ensure personal information data security.
- ❖ To help customers to new things.
- ❖ To provide interactive productive platform to the users.
- ❖ To find easily buy any kind of product background.
- ❖ To automatically generate weekly report on sales and inventory activities
- ❖ To provide notifications on the goods' expiring date for clearance activity
- ❖ To generate receipt with proper format for customer references
- ❖ To provide point of sales for each day
- ❖ To reduce time and cost to control and manage inventory

### **1.4 Expected Outcome**

I hope that my web site will make the life of our Daffodil International University residents' lot easier. Students can apply for hall seat sitting inside the room and also make hall payment sitting inside the hall without any problem. I thought this web site of mine would be a very nice website.

### **1.5 Project Management and Finance**

Customer can make payment on our shop. If the customer wants to pay online then he must login. If you want to make online payment, some extra money will be deducted because this extra money will be needed to buy our api.

## **1.6 Report Layout**

**Chapter 1:** The project's introduction, inspiration, goal, and expected outcomes have all been covered very first.

**Chapter 2:** This chapter shows background process

**Chapter 3:** This chapter shows Use case modeling, flow chart diagram.

**Chapter 4:** We can know about Front-end design, back-end design, implementation requirement.

**Chapter 5:** Database implementation, Front-end design, test was described.

**Chapter 6:** This chapter covered Sustainability and the Social Impact on Society.

**Chapter 7:** The scope of our project and its prospects for further development to buy our api.

## **CHAPTER 2**

### **BACKGROUND**

#### **2.1 Preliminaries/Terminologies**

I use Laravel [1], which beats other web frameworks due to its powerful features and development tools that speed up the development of web applications. I have tried to make the entire project responsive so that it can be used on any device and is very user-friendly.

#### **2.2 Comparative Analysis**

The web application is designed in such a way that it becomes very easy for a person with little computer knowledge to operate it. This web application is user friendly and does not require any technical person to operate it.

#### **2.3 Scope of The Problem**

Inconsistent Tracking Warehouse Efficiency Inaccurate Data

Changing Demand

Limited Visibility

Manual Documentation

Problem Stock

Supply Chain Complexity

## CHAPTER 3

### REQUIREMENT SPECIFICATION

#### 3.1 Requirement Management

Requirements are the capabilities that a product must have in order to fulfill our project's need to address an issue. To build our project, we needed to outline all of its needs. We acquired a large amount of project-related information and had to figure out all of the project's expectations in order to build this project.

#### 3.2 Requirement Collection

- Visual Studio Code for development.
- JavaScript, php for language.
- Html, CSS and Bootstrap for as styling.
- Xampp for database.

#### 3.3 Visual Studio Code

An editor for source code is available for Windows, Linux, and macOS under the name Visual Studio Code from Microsoft. Among the features are debugging assistance, syntax highlighting, intelligent code completion, snippets, code refactoring, and integrated Git. The theme, keyboard shortcuts, and preferences may all be changed by users, and extensions that add new functionality can also be installed.

### **3.4 Php**

"Hypertext Pre-processor" is the abbreviation for PHP [5]. HTML incorporates the Web programming language PHP. This suggests that HTML for a Web page may contain PHP code. The server that hosts a PHP page reads or "parses" the PHP code when the page is browsed. The HTML code that the browser may read is typically the output of the PHP functions on the page. Because PHP code is transformed into HTML before a page is loaded, users cannot understand it. PHP pages are therefore safe enough to access databases and other private data. The syntax of other languages like C, Java, and Perl has a significant influence on PHP. PHP, on the other hand, has several unique features.

### **3.5 Xampp**

The Apache HTTP Server, MariaDB, and interpreters for PHP and Perl scripts are all included in the free and open-source XAMPP package, which was created by Apache Friends. The majority of commercial web server installations use the same parts as XAMPP, therefore switching from a local test server to a live server is not difficult. A developer may quickly and easily deploy a WAMP or LAMP stack on an operating system with the help of popular add-on programs like WordPress and Joomla thanks to XAMPP's simple implementation. They can also be set up using Bitnami with the same ease.



### 3.6 Use Case Modeling and Description

A use-case [4] model is a representation of how various sorts of users interact with a system to solve a problem. As such, it specifies the users' objectives, the interactions between the users and the system, and the system's needed behavior in order to achieve these goals.

### 3.7 Use Diagram

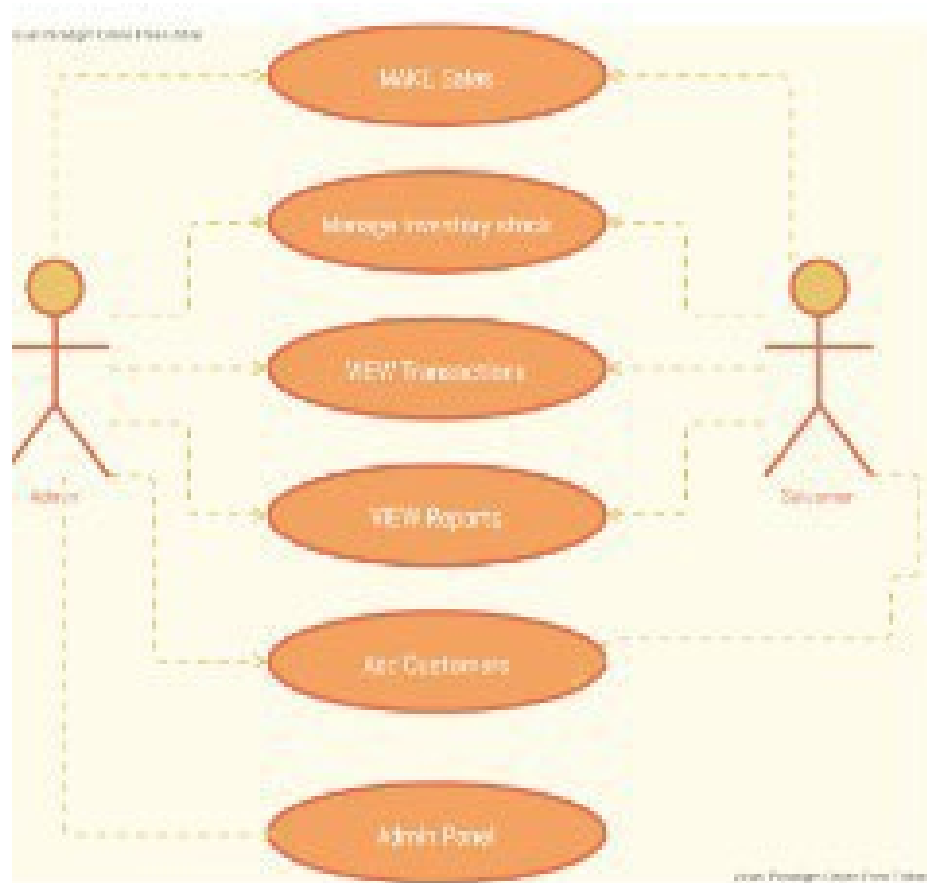


Figure 1.1: Business Process Model of our system

### **3.8 Design Requirement**

In this project, I have used Visual Studio code as our primary code editor. Visual Studio Code is a popular code editor these days. For databases, I used MySQL, it's an open-source relational database management system. For the front-end, I use the bootstrap framework. It helps us to create a responsive design. For the backend of our project, I used the Laravel framework, which is based on PHP and also, I use Livewire, it's a full-stack framework for Laravel that helps us to build dynamic interfaces simply, without leaving the comfort of Laravel. It's a library that makes it simple to build modern, reactive, dynamic interfaces using Laravel Blade as templating language. I have used a variety of Laravel packages that have helped us to create various functionalities in our project. Also, we used JavaScript to perform some tasks.

## **CHAPTER 4**

### **DESIGN SPECIFICATION**

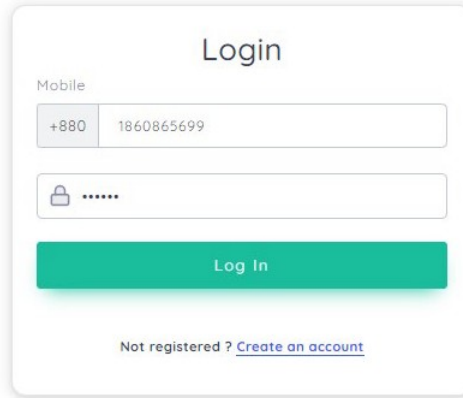
#### **4.1 Front-End Design**

Front-end [2] is very important for such shopping platforms to attract more students. So, I have given a lot of importance to the front-end. I tried to make the whole project responsive so that our project can be used on any device so that we can get more customers. I used Bootstrap framework. It is a very popular front-end framework used to build modern websites and web applications. It is a free front-end framework, intended to make web development faster and easier. It also includes HTML and CSS-based design templates for forms, typography, buttons, navigation, tables, models, image carousels, and many other elements, along with other optional View js plugins.

#### **4.2 Back-End Design**

I used the Laravel [4] framework for the backend programming language in our project. Laravel is a free and open-source very popular PHP web framework. From small projects to very large projects can be done with the Laravel framework. Laravel scores better than other web frameworks due to its powerful features and development tools that facilitate fast web application development. Laravel helps website developers simplify the development process with clean, reusable code that follows the MVC architecture. Since it is a very mature web framework and has a large community of developers, So the solution to any problem can be found very easily that's why we used this framework.

### 4.3 Interacting Design and User Experience (UX)



The image shows a login form titled "Login". It features a "Mobile" label above a text input field containing "+880" and "1860865699". Below this is a password field with a lock icon and six dots. A green "Log In" button is positioned below the password field. At the bottom, there is a link that says "Not registered ? [Create an account](#)".

Figure 4.3.1 Login

An email and password will be provided for the admin. Admin will login with that password and logout again after doing all the necessary work. A specific username and password will be given for the admin.

## 4.4 Admin Panel

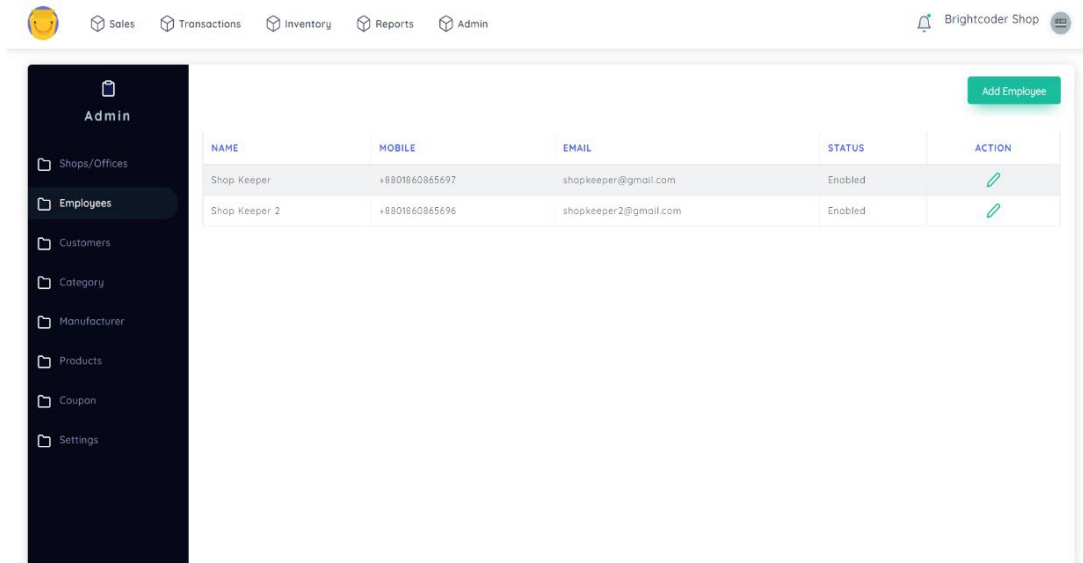


Figure 4.3.2 Admin Panel

This is the admin dashboard. From here admin can see how many floors and how many units are there in the hall and also can see the total number of students. Before entering this dashboard, admin must login. If the admin is not logged in, he will not be able to access the admin panel. How many students are there in the hall and which floor seats are available can be seen in the dashboard. If the admin checks the dashboard regularly then the hall will be able to see how many students will leave the hall in any month.

## Sales Page:

The Sales Page interface includes a navigation menu with 'Sales', 'Transactions', 'Inventory', 'Reports', and 'Admin'. A search bar is located at the top left. The main area displays a grid of product cards, each showing the product name, manufacturer, pack size, and price. A table on the right lists the selected items with columns for Product, Price, Quantity, Sub Total, and Discount. Below the table, there is a coupon code input field and an 'Apply Coupon' button. The summary section shows the Sub Total (590.00), Discount (55.00), Tax (56.40), Paid Amount (591.40), Due (0.000), and Total (591.40). A 'Cash Pay' button is located at the bottom right.

Figure 4.3.3 Sales Page

## Products:

The Products page interface includes a navigation menu with 'Sales', 'Transactions', 'Inventory', 'Reports', and 'Admin'. The main area displays a table of products with the following columns: Name, Manufacturer, Pack Size, Price (without GST), Selling Price (with GST), Qty, and Action. An 'Add Stock' button is located in the top right corner.

NAME	MANUFACTURER	PACK SIZE	PRICE (WITHOUT GST)	SELLING PRICE (WITH GST)	QTY	ACTION
Product 1	Manufacturer 1	2 Box	60	63	48	
Product 1	Manufacturer 1	1 Box	90	94.5	17	
Product 1	Manufacturer 1	3 Box	100	105	27	
Product 2	Manufacturer 2	3 Gram	120	134.4	6	
Product 2	Manufacturer 2	5 Gram	150	168	3	
Product xyz	Manufacturer 2	2 Litre	20	23.6	11	
Prod	Manufacturer 1	1 Box	25	29.5	97	

Figure 4.3.4 Products

## Add Stock:

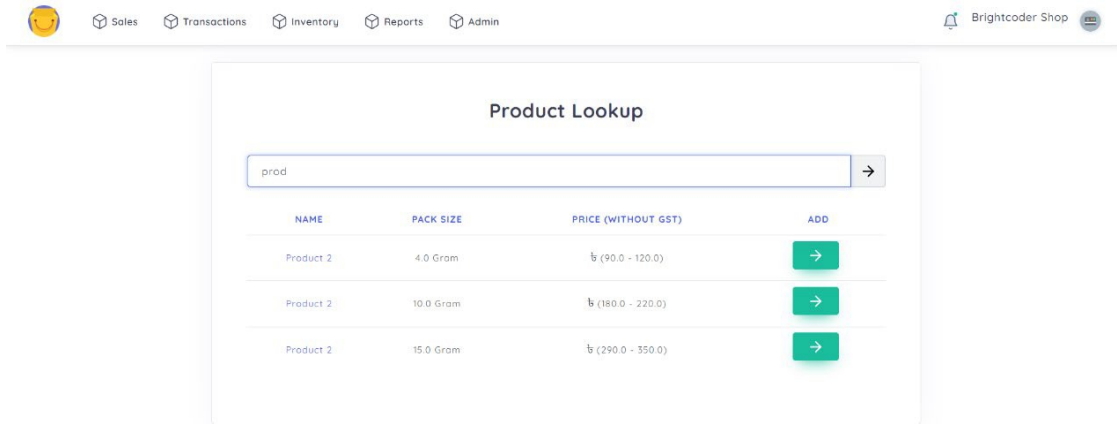


Figure 4.3.5 Add Stock

## Reports:

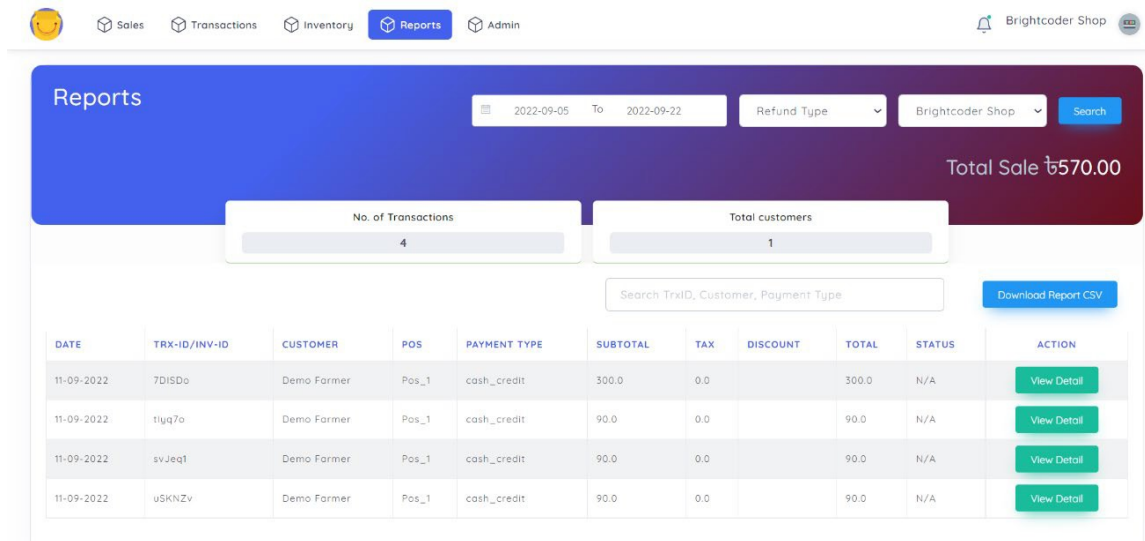


Figure 4.3.6 Reports

## Transactions:

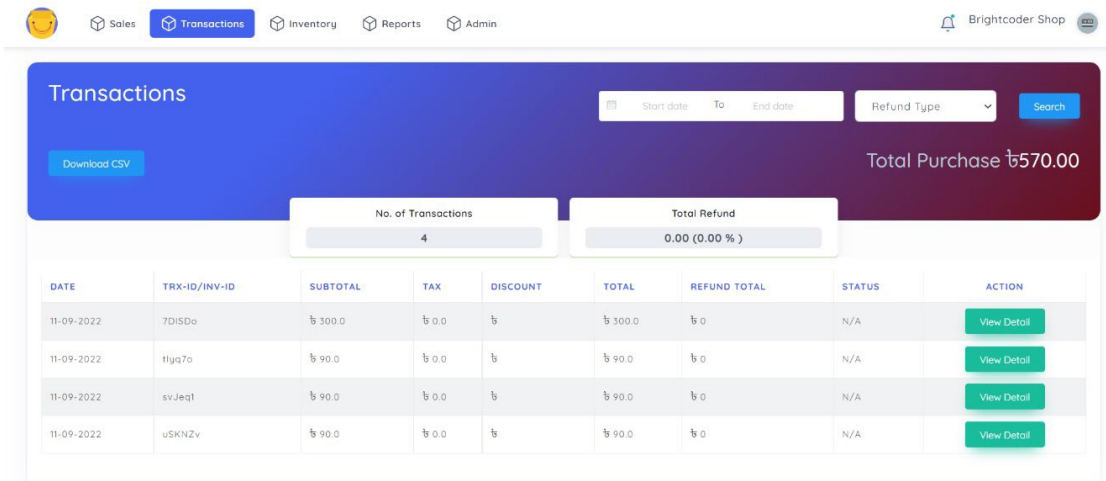


Figure 4.3.7 Transactions

## 4.5 Testing Implementation

Log in Pre-condition/Register

Table 4.5: Register

SL No	Statement	Test Data	Result	Pass/Fail
1	If Phone Number is not correctly inserted	Phone: 01743544365 Password :123456	Please enter correct Phone Number	fail
2	If password is not correctly inserted	Phone: 01743544365 Password :123456	Please enter correct Phone Number	fail
3	If Phone number and password is correctly inserted	Phone: 01743544365 Password :123456	Your registration successfully	pass
4	If Phone number and password is correctly inserted	Phone: 01743544365 Password :123456	Your registration successfully	pass



# CHAPTER 5

## IMPLEMENTATION AND TESTING

### 5.1 Introduction

In this chapter, we will go over the project's implementation procedure. Because this is a web-based application, it includes both a front-end and a back-end component. Both of which will be demonstrated with appropriate screenshots.

### 5.2 Implementation of Database

```
1 APP_ENV=local
2 APP_KEY=base64:L5c0u1L78N3hrMdiBq8LxNtcdS5yblcH8R2ITooU0xg4=
3 APP_DEBUG=true
4 APP_URL=http://localhost
5
6 LOG_CHANNEL=stack
7 LOG_LEVEL=debug
8
9 DB_CONNECTION=mysql
10 DB_HOST=127.0.0.1
11 DB_PORT=3306
12 DB_DATABASE=diu
13 DB_USERNAME=root
14 DB_PASSWORD=
15 DB_CHARSET_INTRANET='latin1'
16 DB_COLLATION_INTRANET='latin1_swedish_ci'
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 QUEUE_CONNECTION=sync
21 SESSION_DRIVER=file
22 SESSION_LIFETIME=120
23
24 MEMCACHED_HOST=127.0.0.1
25
26 REDIS_HOST=127.0.0.1
27 REDIS_PASSWORD=null
28 REDIS_PORT=6379
29
30 #MAIL_MAILER=smtplib
31 #MAIL_HOST=smtplib.gmail.com
32 #MAIL_PORT=587
33 #MAIL_USERNAME=test.bduszel@gmail.com
34 #MAIL_PASSWORD=ldkcnokkzibhyjwvx
35 #MAIL_FROM_ADDRESS=from@example.com
36 #MAIL_FROM_NAME=Agronxlt
37 #MAIL_ENCRYPTION=tls
38
39 AWS_ACCESS_KEY_ID=
40 AWS_SECRET_ACCESS_KEY=
41 AWS_DEFAULT_REGION=us-east-1
42 AWS_BUCKET=
```

Figure 5.2.1: Database connection

## Product Pack:

```
239     return $this->success( message: 'Product List',[
240         'products'=> ShopStockResource::collection($products)
241     ]);
242 }
243 public function lookup($param){
244     $items = ProductPack::with( relations: 'product.tax')->whereHas( relation: 'product',function ($query) use
245         $query->where('name', 'like', '%'. $param . '%')
246         ->orWhere('description', 'like', '%'. $param . '%');
247     }->get()->filter(function($value) {return !isset($value->stock);})->values();
248
249     return $this->success( message: 'Product lists',[
250         'items'=>$items
251     ]);
252 }
253
254 public function addStock(Request $request){
255     $rules = [
256         'id' => 'required',
257         'qty' => 'required',
```

Figure 5.2.2: Product Pack

## User registration:

```
class AuthController extends ApiController
{
    public function register(Request $request)
    {
        $rules = [
            'name' => 'required|string|max:255',
            'business_name' => 'required|string|max:255',
            'mobile' => 'required|string|unique:users,mobile',
            'email' => 'required|string|email|unique:users,email',
            'password' => 'required|string|min:6|confirmed'
        ];
        $validator = Validator::make($request->all(), $rules);
        if ($validator->fails()) {
            return $this->error($validator->errors()->first());
        } else {
            $request->merge(['uid'=>rand(99999,999999)]);
            $credentials = request(['name', 'business_name', 'email', 'mobile', 'uid', 'password']);
            $credentials['password'] = Hash::make($credentials['password']);

            $user = User::create($credentials);
            $userRole = new UserRole();
            $userRole->user_id = $user->id;
            $userRole->role_id = 2;
            $userRole->save();

            $subscription = new Subscription();
            $subscription->user_id = $user->id;
            $subscription->plan_id = 1; // basic
            $subscription->amount = 3000; // basic
            $subscription->expired_at = Carbon::parse($subscription->expired_at)->addDays(15);
            $subscription->save();
            $accessToken = $user->createToken('authToken')->plainTextToken;

            return $this->success( message: 'Registration successful', [
                'access_token' => $accessToken,
                'token_type' => 'Bearer'
            ]);
        }
    }
}
```

Figure 5.2.3: User registration

## User login:

```
public function login(Request $request)
{
    $rules = [
        'mobile' => 'required',
        'password' => 'required|string'
    ];
    $validator = Validator::make($request->all(), $rules);

    if ($validator->fails()) {
        return $this->error($validator->errors()->first());
    }
    elseif (!Auth::attempt($request->only( keys: 'mobile', 'password'))) {
        return $this->error( message: 'Credentials not match', errors: 401);
    }
    if (!\auth()->user()->is_enabled){
        return $this->error( message: 'Account Disabled!');
    }
    $tokenName = 'authToken';
    if (\auth()->user()->roles){
        if (\auth()->user()->is_system_user){
            $tokenName = 'adminAuthToken';
        }
    }
    return $this->success( message: 'Login Success', [
        'access_token' => auth()->user()->createToken($tokenName)->plainTextToken,
        'token_type' => 'Bearer'
    ]);
}
```

Figure 5.2.4: User login

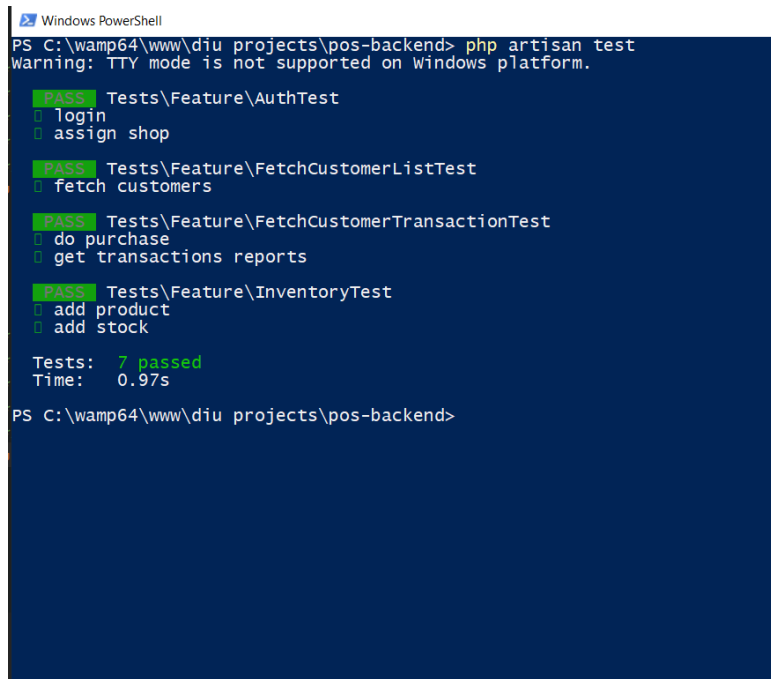
## Seller:

```
1 <template>
2   <!-- BEGIN CONTENT AREA -->
3   <div id="content" class="main-content">
4     <div class="layout-gx-spacing">
5
6     <!-- CONTENT AREA -->
7     <template v-if="showInvoice">
8       <vuejs-view :transactions="transaction" :settings="settings" :customer="customer" :user="user" @onBack="handleBack"></vuejs-view>
9     </template>
10
11     <div v="else" class="row layout-top-spacing">
12       <div class="col-7 layout-spacing">
13         <div class="widget widget-account-invoice-three">
14           <div class="row widget-content widget-content-area product-picker justify-content-center">
15             <div class="input-group mb-3">
16               <input ref="card" v-model="searchQuery" type="text" class="form-control" autocomplete="false" placeholder="Search Product Name, Manufacturer, Pack Size">
17               <!-- <div class="input-group-append @click="searchProducts">-->
18               <!-- <span id="helpCard" class="input-group-text"></span-->
19               </div-->
20             </div>
21
22             <div v-for="product in filteredProducts" :key="product.id" class="col-3 mb-3">
23               <div class="card component-card_4">
24                 <div class="card-body product product-hover" :class="{ 'product-hover': sales.details.indexOf(product) !== -1}" @click="handleProduct(product)">
25                   <div class="user-info">
26                     <div class="media-body cursor-pointer">
27                       <div class="card-user_name text-success mb-2">{{ product.name }}</div>
28                       <div class="card_manufacturer">{{ product.manufacturer }}</div>
29                       <span class="badge outline-badge-primary">{{ product.pack_size }}</span>
30                       <span class="badge badge-warning ml-2">{{ product.qty }}</span>
31                       <span class="badge badge-danger counter ml-4">{{ $currency.sign }}</span> {{ product.selling_price }}</span>
32                     </div>
33                   </div>
34                 </div>
35               </div>
36             </div>
37           </div>
38
39           <div v-if="1" class="row mt-3">
40             <div class="col-12 layout-spacing">
41               <div class="widget widget-account-invoice-three sales">
42                 <div class="widget-heading mb-4">
43                   <div class="wallet-user-info">
44                     <div class="user-name">
45                       <span class="customer-area">
46                     </div>
47                     <div class="add">
48                       <div class="form-inline justify-content-end">
49                         <input ref="card" v-model="depositAmount" type="number" min="0" class="form-control" placeholder="$currency.sign">
50
51                         <button type="submit" class="btn btn-success ml-3" @click="customerDeposit">Deposit Cash/Advance</button>
52                       </div>
53                     </div>
54                   </div>
55                 <div class="wallet-balance">
56                   <div class="text-white">Total Transaction Amount: <span class="w-currency">{{ $currency.sign }}</span>{{ customer.credit_limit }}</div>
57                   <div class="text-white">Utilized Credit: <span class="w-currency">{{ $currency.sign }}</span>{{ utilizedCredit }}</div>
58                 </div>
59               </div>
60             </div>
61           </div>
62         </div>
63       </div>
64     </div>
65   </div>
66 </template>
```

Figure 5.2.5: Seller

### 5.3 Testing Implementation

As this is a web-based application, testing is most critical before distributing it. These tests will assess overall performance and provide guidance on how to make required adjustments to the project. The testing can also inform us how long it takes to utilize the program and how to save time. This program may be used to run a variety of tests. We conducted several experiments in this location, and the results are displayed below.



```
Windows PowerShell
PS C:\wamp64\www\diu\projects\pos-backend> php artisan test
Warning: TTY mode is not supported on Windows platform.

[PASS] Tests\Feature\AuthTest
  [ ] login
  [ ] assign shop

[PASS] Tests\Feature\FetchCustomerListTest
  [ ] fetch customers

[PASS] Tests\Feature\FetchCustomerTransactionTest
  [ ] do purchase
  [ ] get transactions reports

[PASS] Tests\Feature\InventoryTest
  [ ] add product
  [ ] add stock

Tests: 7 passed
Time: 0.97s

PS C:\wamp64\www\diu\projects\pos-backend>
```

Figure 5.3: Testing Implementation

## **5.4 Implementation of Font-End Design**

For this type of shopping platform, the front-end is crucial to attracting more customers. Therefore, the front-end has been given a lot of importance. In an effort to increase my customer base, I have made every effort to make the project responsive, making it usable on any device. A framework called bootstrap was employed. A lot of contemporary websites and web applications are made using this front-end framework. With the aim of accelerating and simplifying web development, it is a free front-end framework. In addition, it comes with additional optional and HTML and CSS-based [5] design templates for forms, typography, buttons, navigation, tables, modals, image carousels, and many other elements.

## **5.5 Test Results and Reports**

We received good findings after doing the above-mentioned tests. Because the entire project was developed with Laravel [6] and there are over 40 files with thousands of lines of code, it's simple to make mistakes. We discovered and corrected our errors using Unit Testing. The integration test was carried out concurrently with the application's development process. We test user interfaces, APIs, and databases via function testing. The results of testing on the website were positive.

## **CHAPTER 6**

### **IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY**

#### **6.1 Influence of Society**

Small business owners can manage their business effortlessly from my website without making any investment. The administrator will receive a very small percentage of the revenue from each order. I believe that my concept will greatly help small business owners as well as consumers, administrators and our society as a whole.

#### **6.2 Effects on The Environment**

Generation IM's global study found that management system is more carbon-efficient than traditional retail outlets. The business's carbon-efficient operations cover sectors including goods transport

#### **6.3 Ethical Aspects**

It is very important to ensure that the person providing the assessment. I ensure that Student can only Application in I Website. Popular development platform Laravel which I used is famous for its security.

#### **6.4 Sustainability**

The model we used seems to give a take able accuracy over 80% for all of our data. The society we live in is developing every second. Even though there could be a model which can outperform the accuracy, the working model and procedures may remain the same, the principle obey project for application are used in various purposes where the server handles most of the request.



## **CHAPTER 7**

### **CONCLUSION AND FUTURE SCOPE**

#### **7.1 Discussion and Conclusion**

Conclude the description about the project, the project developed using PHP with Laravel is based on the specification of user requirements and analysis of existing systems, with flexibility for future improvements. To conclude, Inventory Management System is a simple desktop-based application basically suitable for small organization. It has every basic item which are used for the small organization. Our team is successful in making the application where we can update, insert and delete the item as per the requirement. This application also provides a simple report on daily basis to know the daily sales and purchase details. This application matches for small organization where there small limited iodoforms. Through it has some limitations, our team strongly believes that the implementation of this system will surely benefit the organization.

## **7.2 Scope for Future Development**

Since this project was started with very little knowledge about the Inventory Management System, we came to know about the enhancement capability during the process of building it. Some of the scope we can increase for the betterment and effectiveness or listed below:

interactive user interface design.

Manage Stock Go down wise.

Use of Oracle as its database.

Online payment system can be added.

Making the system flexible in any type.

Sales and purchase return system will be added in order to make return of products.

Lost and breakage.

## REFERENCES

- [1] Learn About Laravel, available at <<<https://www.laravel.com>>>, last accessed on 4-12-2022 at 9:00 PM
- [2] Front end discussion < <https://airfocus.com/glossary/what-is-a-front-end/>> last accessed on 3-7-2022
- [3] Use Case diagram < <https://www.lucidchart.com/pages/uml-use-case-diagram>> last accessed on 11-7-2022 at 10:20 AM.
- [4] Learn About Laravel, available at <<<https://www.laravel.com>>>, last accessed on 4-12-2022 at 9:00 PM
- [5] Jon Duckett, HTML and CSS, CSS Basic, 2nd Edition, The Ebook Press, 2018, pp.16-40
- [6] Learn About Laravel, available at <<<https://en.wikipedia.org/wiki/Laravel>>>, last accessed on 1-12-2022 at 5:00 AM.

# PLAGIARISM REPORT

28%

SIMILARITY INDEX

26%

INTERNET SOURCES

1%

PUBLICATIONS

19%

STUDENT PAPERS

## PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	5%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	5%
3	ir.csuc.edu.gh:8080 Internet Source	3%
4	pdfcoffee.com Internet Source	3%
5	Submitted to University of Teesside Student Paper	1%
6	www.educba.com Internet Source	1%
7	jozilla.net Internet Source	1%
8	comprensivofeltre.it Internet Source	1%
9	Submitted to Coventry University Student Paper	1%