SMALL STORE MANAGEMENT SYSTEM

FOR

INTERNET AND WEB PROGRAMMING

(CSE3002)

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Computer Science Engineering

By

Damayanti Chattopadhyay (18BCI0222)

Reema Agarwal (18BCE2386)

Jenise James (18BCE2375)

Under the guidance of

Dr. LOKESH KUMAR R

Associate Professor Grade 1

Department of Analytics

VIT®

UNIVERSITY

(Estd. u/s 3 of UGC Act 1956)

ACKNOWLEDGEMENT

"It is not possible to prepare a project report without the assistance & encouragement of other people. This one is certainly no exception."

We would like to express our gratitude and appreciation to those who gave us the possibility to complete this report. A special note of thanks to our faculty, Dr. Lokesh Kumar R, whose help, stimulating suggestions and encouragement, helped us coordinate our project and in writing this report.

We extend our gratitude to VELLORE INSTITUTE OF TECHNOLOGY for giving us this opportunity.

We also acknowledge with a deep sense of reverence, gratitude towards our parents and members of our family, who have always supported us morally and economically.

At last but not the least gratitude goes to all our friends who directly or indirectly helped us to complete this project report.

CERTIFICATE

This is to certify that the project work entitled "***Small Store Management System***" that is being submitted by "***Damayanti Chattopadhyay (18BCI0222), Reema Agarwal (18BCE2386), Jenise James (18BCE2375)***" for Internet and Web Programming (CSE3002) is a record of bonafide work done under our supervision. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted for any other course.

Place: Vellore

Date: 30th October, 2020

Signature of Student:

Damayanti Chattopadhyay

Reema Agarwal

Jenise James

ABSTRACT

Technology has changed the way of business. Many businesses have been able to reach out to a bigger audience with the help of e-Commerce platforms. Even at stores, a proper database management system has benefitted the owners in keeping a track of the inventory along with lucidly managing the accounts and finance.

Our focus for this project is to build a Website that will primarily focus on Small Stores, like a local 'Kirana' shop. We know that with so many technology companies trying to establish a loyal customer base, affording an electronic device has become easier than before. Many organizations are selling their products at very affordable rates to the lower income group as well.

We want to help the shopkeepers maintain their store. These local shops have a loyal base of customers from the neighborhood, who also buy things and pay at the end of the month. We want to create a digital system that will integrate the loaning feature with the website and help the shopkeeper keep a track of the customers and the amount they owe.

Talking about inventory, these shops get their stocks replenished regularly and also employ few helpers to manage the store. However, many shopkeepers complain of getting cheated on the quantity of inventory by these helpers. We want to solve this problem by creating a database for the shopkeeper to update the inventory and make sure that nothing gets stolen.

We will make sure that the user interface is very easy with small icons and is available in the major Indian languages, too. So, the shopkeepers who are most comfortable in their native language can also use this system. The icons will guide them in understanding the content of the different buttons.

Big changes are only possible with small steps. When our Government is promoting 'Make in India', as the future of our nation, we want to equip our small businesses with the right tools. The journey of a mile starts with a single step!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

HTML – Hyper Text Markup Language

CSS – Cascading Style Sheets

PHP - PHP: Hypertext Preprocessor

DFD- Data Flow Diagram

1. INTRODUCTION

1.1 SYSTEM OVERVIEW

The name of the website is Saamagri. It is a small store management system. It comprises of an integrated language-change option to facilitate the shopkeeper and make her comfortable in her language. The separate web pages provide information on the two main functionalities of our website: Inventory management and Bookkeeping. The email button allows the user to directly send emails from the website to different vendors or other concerned personnel.

The sign-up and login options allow new staff to be added to the database of users. Once the user has logged in, she will get four options to choose from: View Inventory, Manage Inventory, Access Logbook and Bill Generation. The view inventory option displays the full stock of items available at the store and it can be edited according to the convenience of the shopkeeper, using the Manage Inventory option. The bill generation allows the shopkeeper to even produce pdfs that she can give to her customer. The credit due to the customer gets automatically logged into the logbook, which can be viewed from the Access logbook page.

Finally, our team page showcases the development team behind this project. The explore option showcases the different news articles that had motivated us to take up this project.

1.2 OBJECTIVE

- We aim to create a website for the small stores.
- The website can be translated from English to any other regional language of the shopkeeper's choice.
- The Database System will automatically reduce the quantity of the inventory when that particular item is bought by the customer and the bill is generated.
- The Customer Payment Activity Log book integrated in the website will help the shopkeeper maintain the accounts better.
- We know that not every one of these shopkeepers are educated well enough to operate the website if it's too complicated. Therefore, we will be using icons with images that will depict the functionality of a particular button.
- Our main objective is to empower the small businesses of our country. After all, the journey of a mile, begins with a single step.

1.3 APPLICATION

1. The website has an integrated bill generating system

2. After the bill generation, the customer logbook is automatically updated.

3. The Inventory management feature allows the staff and the admin to view the stock at a glance and even edit or delete the price or the item itself, according to their convenience.

4. When fresh stock arrives at the store, the staff or the admin can log on to the website and update the inventory.

5. The separate log in at the access logbook page makes sure that only authorized people are allowed to view and update the financial records of the store.

6. To make the admin understand the two major functionalities of the website, i.e., logbook and inventory, two separate web pages have been integrated with adequate information.

7. On the Home page, the explore button reveals the current news articles that have motivated us to take up this project.

8. The integrated mail system, allows the admin to directly contact any necessary personnel via mail.

1.4 LIMITATIONS

1. The website is locally hosted right now. So, it cannot be accessed by other users.

2. There are no strong security mechanisms present on the website.

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

Small stores/kirana are present in India because of their closeness to the consumer and consumer trust in them. The major features that are followed by such stores currently include

1. These stores are located in vicinity of consumers at their convenience. They can meet the immediate needs of people. For items of small value or for just one or two items one need not go to a mall or hyper market.

2. The owners of the stores offer advice to consumers and help them in making the right choice.

3. They offer goods on credit and also deliver goods at the residence of the buyer. They can also take orders over the phone and ensure home delivery of low value items to customers.

4. Bill generation is done by writing the items on a paper with the total amount calculated by hand.

5. All customer accounts and inventory details are maintained manually in writing. The majority of the kirana businesses run on traditional concept of handwritten paper billing and book-keeping.

The existing system faces many challenges like

1. Discrepancies in inventory details or customer accounts due to human error
2. Unauthorized access to store information by staff which can even lead to incidents of stealing
3. Errors in bill generation which can be taken advantage by the consumer
4. Poor window display at the store.
5. Poor management of the store.

2.1 PROPOSED SYSTEM

Our proposed system attempts to blend the relationship marketing maintained by these stores with the consumers along with digital management of all the stores aspects. This will be implemented through a website which will have the following features:

1. Inventory management can be done digitally depending on the stock quantity
2. Similarly, customer credit can be updated through digital records
3. Access to the system can only be granted through proper credentials
4. The contents of the website can be viewed at the language of the user's choice

2.2.1 BENEFITS OF PROPOSED SYSTEM

The proposed system helps integrate technology into the existing paradigm of the small stores to help in their efficient management. The benefits incurred from this upgrade system include

1. Unauthorized access can be controlled which can reduce unwanted incidents among working staff
2. Values in inventory details and customer accounts can be free from human error as it will be updated digitally
3. Proper awareness about stock items and their quantities can help improve the window display of these stores and can help reduce discrepancies in deliveries
4. Bill generation will also be error free with all the details of the transaction specified in the bill printed
5. The website concept will be user friendly even for those with minimal experience using technology which is implemented using icons, pictures, and the option to change to the local language used.

3. REQUIREMENT SPECIFICATION

3.1 HARDWARE REQUIREMENT

Version 1909

Processor- Intel® Core™ i7-8550U CPU

Installed RAM- 8.00 GB

3.2 SOFTWARE REQUIREMENT

Edition – Windows 10 Home

System type- 64-bit operating system, x64-based processor

• HTML

Hypertext markup language Also called web pages. A markup language is a set of markup tags The Web pages created with HTML alone are static, meaning the user can't interact with the Web page. All users see the same Web page. Dynamic Web pages, on the other hand, allow the user to interact with the Web page. It enables the user to make and structure segments, passages, headings, divisions, paragraphs and add links to other Webpages for web Pages and applications.

• CSS

CSS is the abbreviation for Cascading Style Sheets. It specifies the styling of HTML elements. We can specify the font, font-size, font-color, background color and so much more. CSS is used to create the layout and look of pages.

• JavaScript

One language widely used to make Web pages dynamic is JavaScript. JavaScript is useful for several purposes, such as mouse-overs (for example, to highlight a navigation button when the user moves the mouse pointer over it) or accepting and validating information that users type into a Web form. JavaScript is a dynamic computer programming language for the Web. Its implementations allow client-side script to interact with the user and make dynamic pages.

• PHP

PHP is a language that is particularly well suited to interact with databases. PHP can accept and validate the information that users type into a Web form and can also move the information into a database. PHP files can contain HTML, CSS, JavaScript, text and PHP code. PHP code is executed on the server, and the result obtained is returned to the browser as plain HTML. PHP performs various system functions like it can create, open, read, write, and close files from a system. It helps in encrypting data.

• **MYSQL**

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB. MySQL is very friendly to PHP, the most appreciated language for web development. MySQL uses a standard form of the well-known SQL data language.

- **XAMPP**

XAMPP is which is a light-weight easy to install bundle that will allow us to do local development on websites in case we don't have a server hosted already. XAMPP consists of the three main things that we need to know when starting web development. They are: Apache Web Server, PHP & My SQL.

4. SYSTEM DESIGN SPECIFICATION

4.1 SYSTEM ARCHITECTURE

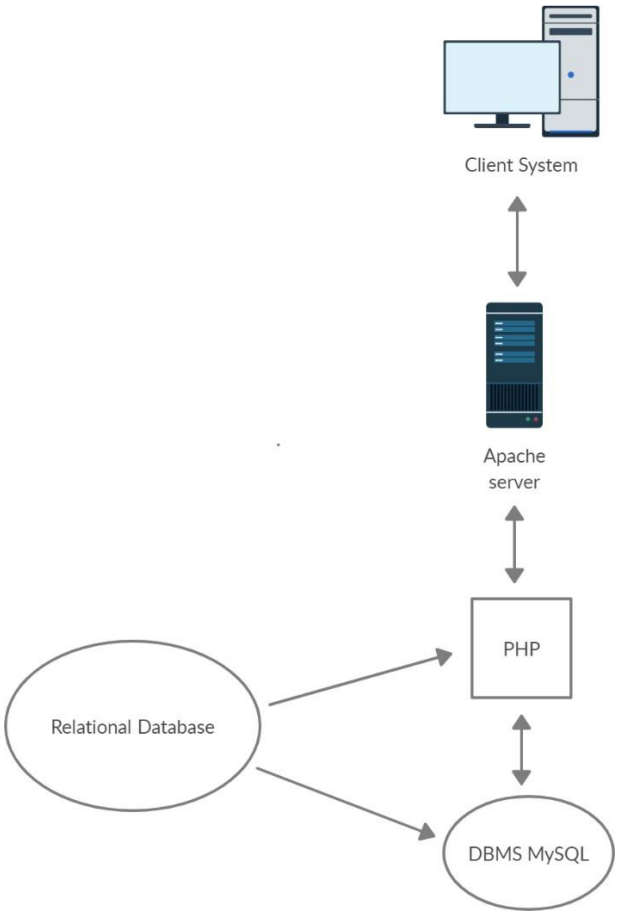


FIG. 1

4.2 DETAILED DESIGN

DFD LEVEL-0

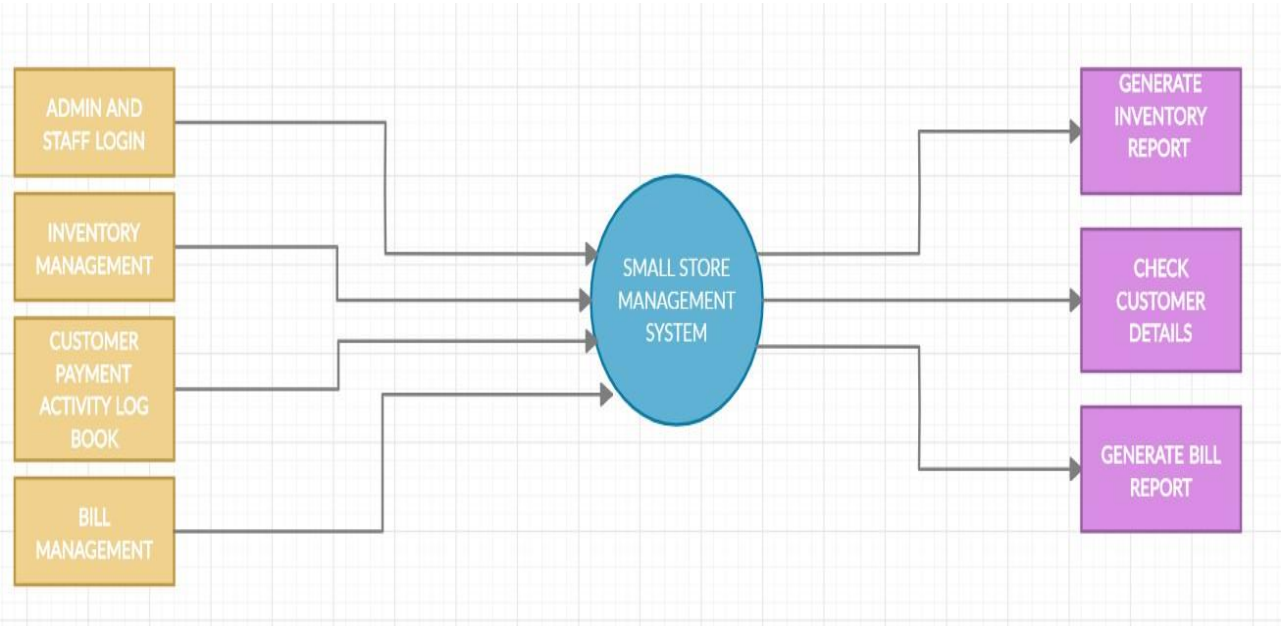


FIG. 2

DFD LEVEL -1

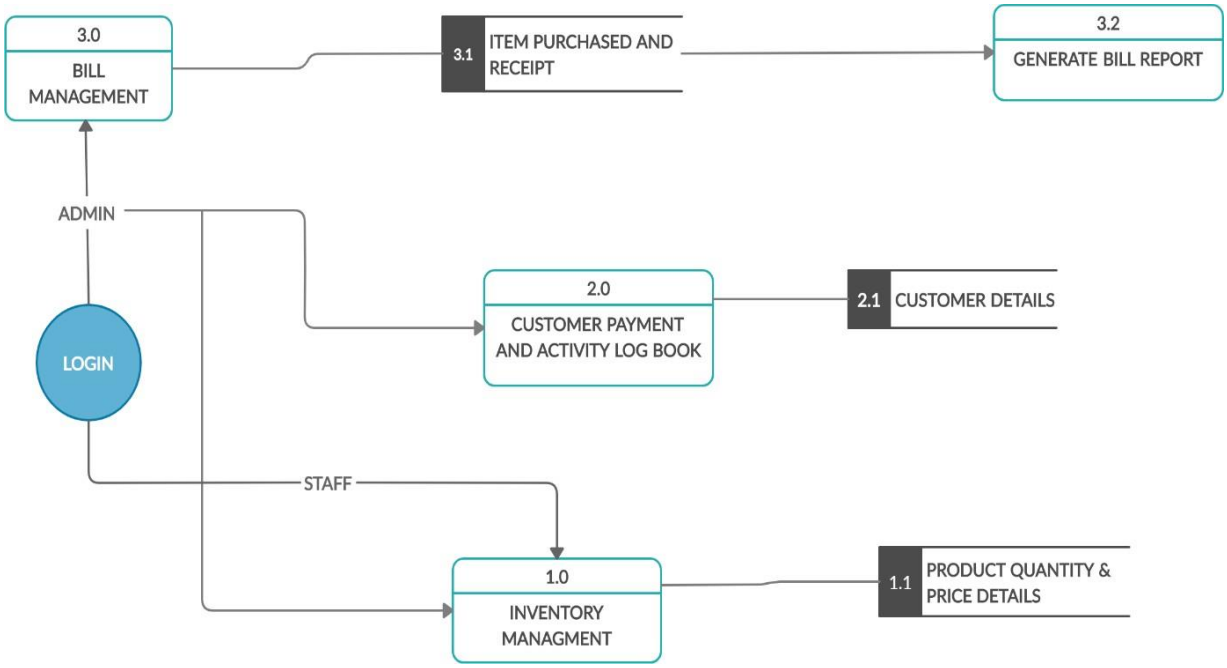


FIG. 3

DFD LEVEL-2

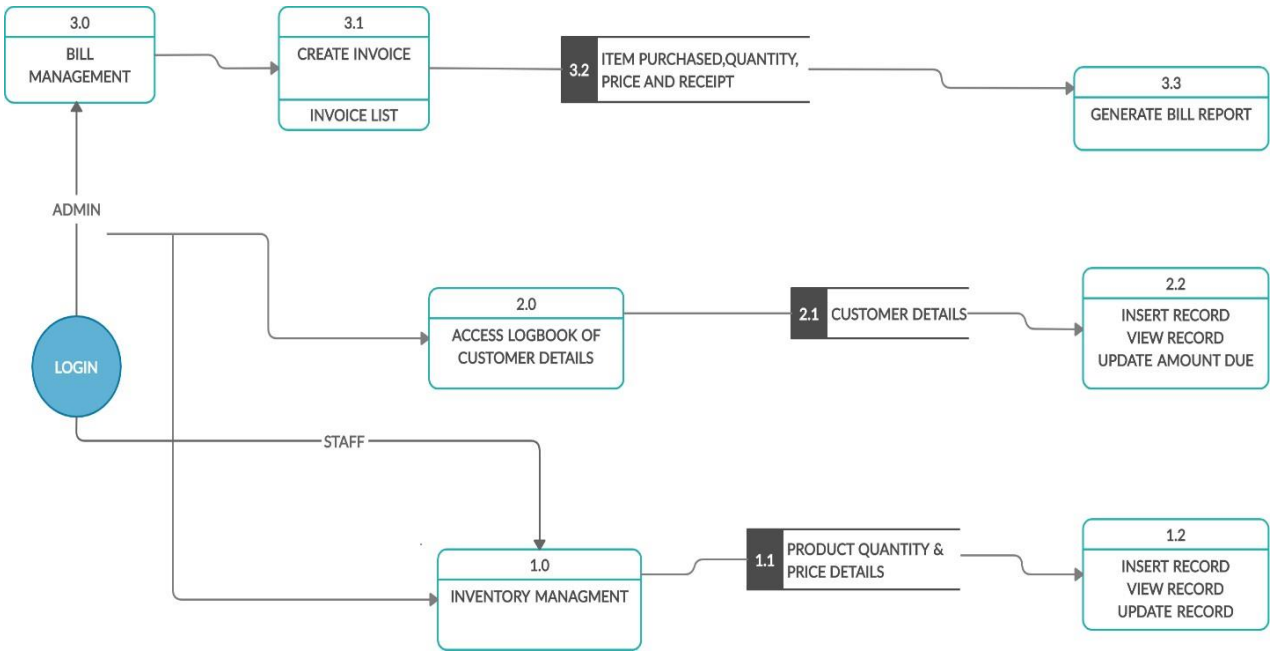


FIG. 4

4.3 DATABASE DESIGN

TABLES IN DATABASE MANAGEMENT

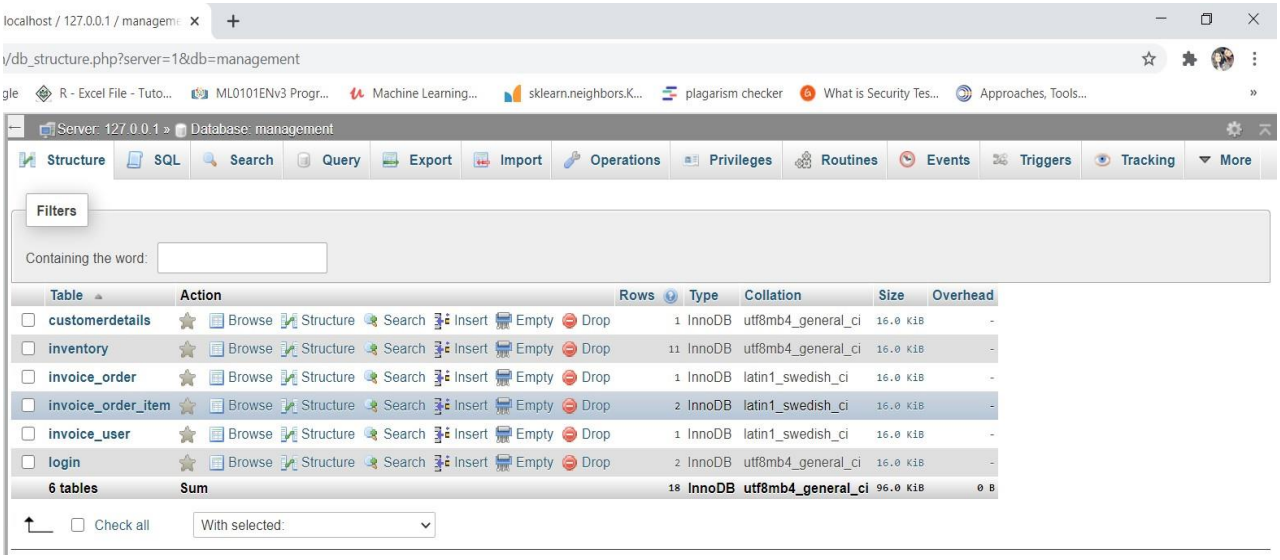


FIG. 5

CUSTOMER DETAILS STRUCTURE

FIG. 6

CUSTOMER DETAILS

FIG. 7

INVENTORY STRUCTURE

FIG. 8

INVENTORY TABLE

+ Options

	id	Products	Quantity(unit/kg)	Price(unit/kg)
Edit Copy Delete	1	sugar	20	50
Edit Copy Delete	2	Noodles	10	70
Edit Copy Delete	3	Biscuits	40	10
Edit Copy Delete	4	Flour	50	27
Edit Copy Delete	5	Oil	40	75
Edit Copy Delete	6	Chocolates	100	15
Edit Copy Delete	7	Flakes	50	100
Edit Copy Delete	8	Ghee	40	450
Edit Copy Delete	9	Coffee	20	75
Edit Copy Delete	10	Shampoo	25	70
Edit Copy Delete	11	maggi	5	10

Check all With selected: Edit Copy Delete Export

Show all | Number of rows: 25 Filter rows: Search this table Sor

FIG. 9

INVOICE_ORDER STRUCTURE

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	order_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	user_id	int(11)			No	None			Change Drop More
3	order_date	timestamp			No	current_timestamp()			Change Drop More
4	order_receiver_name	varchar(250)	latin1_swedish_ci		No	None			Change Drop More
5	order_receiver_address	text	latin1_swedish_ci		No	None			Change Drop More
6	order_total_before_tax	decimal(10,2)			No	None			Change Drop More
7	order_total_tax	decimal(10,2)			No	None			Change Drop More
8	order_tax_per	varchar(250)	latin1_swedish_ci		No	None			Change Drop More
9	order_total_after_tax	double(10,2)			No	None			Change Drop More
10	order_amount_paid	decimal(10,2)			No	None			Change Drop More
11	order_total_amount_due	decimal(10,2)			No	None			Change Drop More
12	note	text	latin1_swedish_ci		No	None			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Fulltext Add to central columns

FIG. 10

INVOICE_ORDER

order_id	user_id	order_date	order_receiver_name	order_receiver_address	order_total_before_tax	order_total_tax	order_tax_per	order_total_after_tax	order_amount_paid	order_total_amount_due	note
12	111	2020-10-29 21:05:44	Reema	sikar	130.00	2.60	2	132.60	72.00	60.60	

elected: Edit Copy Delete Export

FIG. 11

INVOICE_ORDER_ITEM STRUCTURE

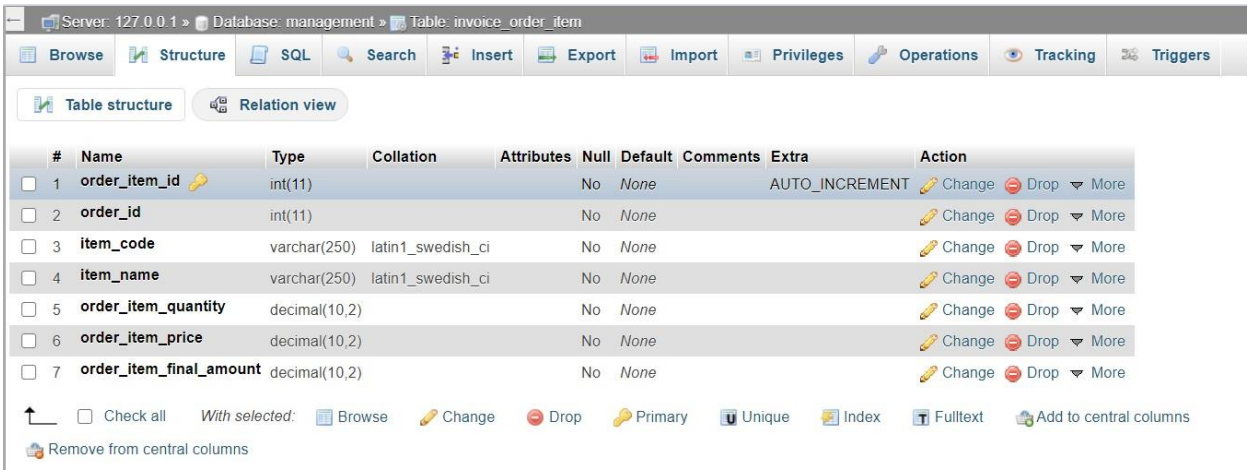


FIG. 12

INVOICE_ORDER_ITEM

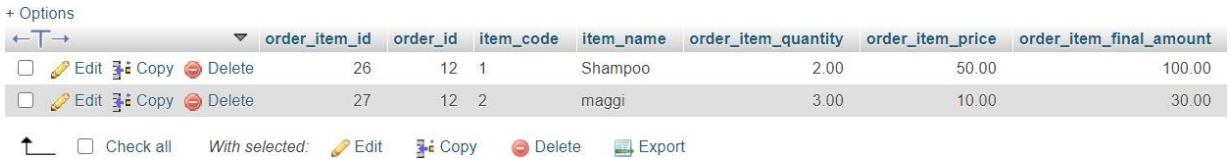


FIG. 13

INVOICE_USER STRUCTURE

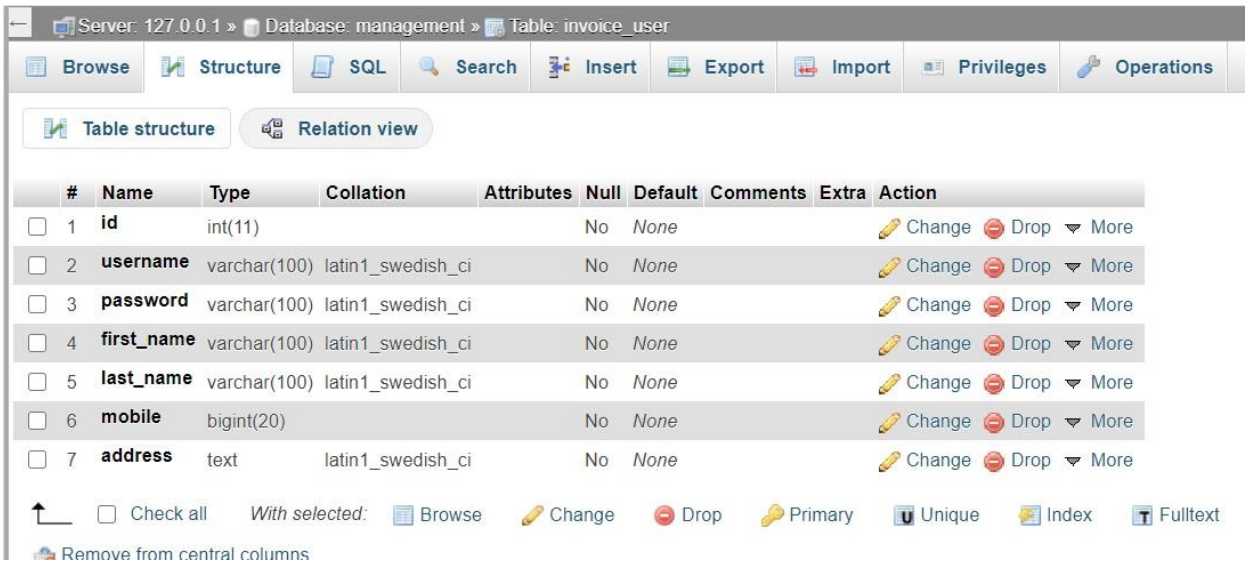


FIG. 14

INVOICE_USER

Show all | Number of rows: 25 Filter rows: Search this table

+ Options

id	username	password	first_name	last_name	mobile	address
111	admin	12345	Jenise	James	1236576546	India

FIG. 15

LOGIN STRUCTURE

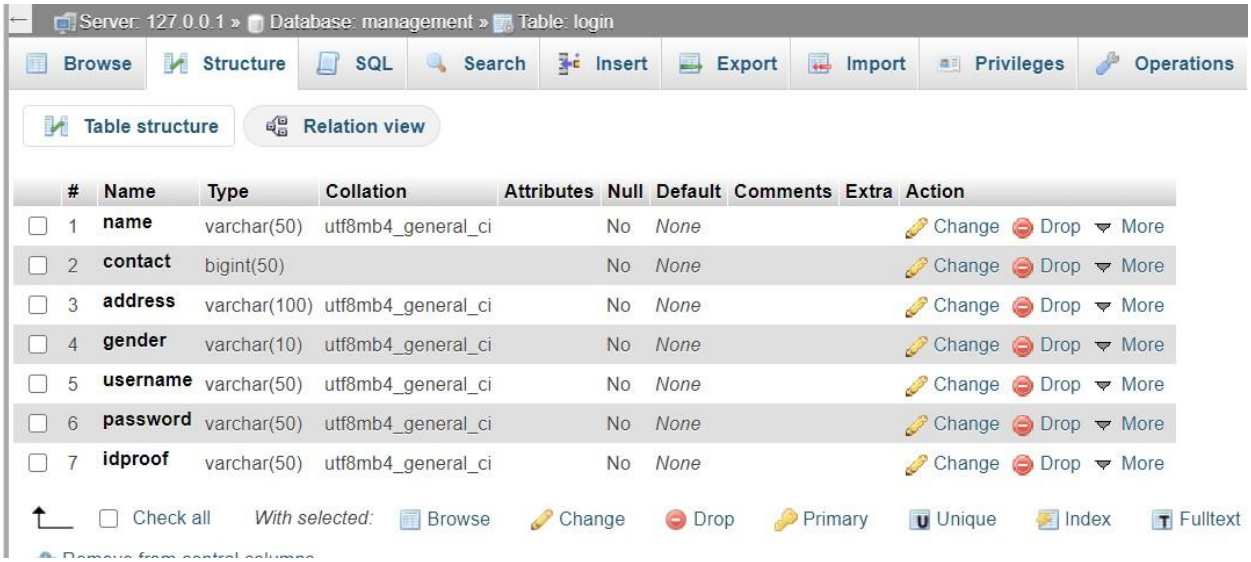


FIG. 16

LOGIN

+ Options

name	contact	address	gender	username	password	idproof
sonali	7726076179	nepal	female	sonaligupta	mE039Gp9Lait1Q6WGw06Yb4cikj0b4A/pUZiPVaDCv4=	pancardd
reema	7426931082	sikar	female	admin	Ghjd6pvG5dow53x7xV8eZeWXDuixJMVXSsonUMG4fa4=	adharcard

FIG. 17

5. SYSTEM IMPLEMENTATION

5.1 MODULE DESCRIPTION

The main modules incorporated in this project include:

- User Interface: Simplicity is the key while creating the user interface. It mainly includes icons and pictures to allow the users to select and choose based on recognition. This will eliminate any technological or literacy disadvantages from the user's side.
- Inventory management: Inventory Management is one of the key features of our website. The stock of the inventory is backed up in a database which can be accessed by both the owner and the other staff of the shop. The table also gives the option to update the price and quantity of stock.
- Customer Activity logbook: Small stores usually have a set of customers who regularly shop at the store and keep tabs for their payment. Using database management, the dues of the customers can be followed up by the shopkeeper in an organized manner. It has been directly integrated with the bill generation system to eliminate the need for manual intervention.
- Change language option: We have implemented an option to change the language from English to a number of other languages as per the convenience of the end user.

6. CONCLUSION AND FUTURE ENHANCEMENTS

The past four months that we, the Team of Saamagri, had invested in the project, were a complete learning experience. The website which we have built is a small effort by a team of future engineers to better the lives of the lesser fortunate people. The Government of India in recent years has continuously brought in new policies and changes to change the outlook of the local stores. The frontend of the website has been built on HTML, CSS and JavaScript. For the implementation of Backend, we have used PHP and MySQL database. The website is locally hosted at the moment, but we want to take our hard work to the mass where it can genuinely benefit people.

In the future, we will work on the following aspects of the website:

- Host the website and make it accessible to other users.
- Work on integrating the language framework throughout the website, and not just on individual web pages.
- Embed more security features in the website
- Work on introducing buying options for the customers
- Integrate a chat box that can solve any queries that the customer might have regarding the website

7. APPENDICES

7.1 APPENDIX 1 – SAMPLE SOURCE CODE

Home.html

```
<!DOCTYPE html>
<head>
<title>Saamagri-Online Grocery Store</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="css/style2.css">
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
</head>
<body>
<div class="hero">
<div class="nav-bar">
<div class="nav-logo">
<img src="logo.png">
</div>
<div class="nav-links" id="nav-links">
<i class="fa fa-close" onclick="closeMenu()"></i>
<ul>
<a href="#"><li>HOME</li></a>
<a href="inventory.html"><li>INVENTORY</li></a>
<a href="log1.html"><li>LOG BOOK</li></a>
</ul>
<a href="login.php"><button type="button" class="btn">LOG IN</button> </a>
</div>
<i class="fa fa-bars" onclick="showMenu()"></i>
</div>
<div class="banner-title">
<h1>Only the <span>Fresh</span></h1>
<a href="explore.html"><button type="button" class="btn">EXPLORE</button></a>
</div>
<div class="vertical-bar">
<div class="search-icon">
<a href="about.html"><i class="fa fa-address-card"></i></a>
<a href="mailto:someone@example.com"><i class="fa fa-envelope-open-o"></i></a>
</div>
<div class="social-icons">
<i class="fa fa-shopping-basket"></i>
<i class="fa fa-handshake-o"></i>
```

```
<i class="fa fa-money"></i>
</div>
</div>
</div>
<div align="right">
<div id="google_translate_element"></div>
<span><script type="text/javascript">
function googleTranslateElementInit() {
new google.translate.TranslateElement({
pageLanguage: 'en',
layout: google.translate.TranslateElement.InlineLayout.SIMPLE
  }, 'google_translate_element');
}
</script><script
src="//translate.google.com/translate_a/element.js?cb=googleTranslateElementInit"></script>
</script></span></div>
<script>
var show=document.getElementById("nav-links");
function showMenu(){
show.style.right="0";
}
function closeMenu(){
show.style.right="-200px";
}
</script>
</body>
</html>
```

Welcome.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Welcome Page</title>
<link rel="stylesheet" type="text/css" href="css/wel.css">
<link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Raleway:400,500,600&Cinzel">
</head>
<body>
<section class="b1">
<div class="header">
<ul>
<li><a href="home.html">Home</a></li>
<li><a href="inventory.html">Inventory</a></li>
```

```
<li><a href="log1.html">Log Book</a></li>
<li><a href="login.php">Log In</a></li>
</ul>
</div>
<div class="in1">
<div class="content">
<h1>Welcome to Saamagri</h1>
<a class= "btn" style="background-color:#CCCCCC; color:#000000; font-size: 25px"
href="inventory.php">View Inventory</a>
<a class= "btn" style="background-color:#CCCCCC; color:#000000; font-size: 25px"
href="dashboard.php">Manage Inventory</a>
<a class= "btn" style="background-color:#CCCCCC; color:#000000; font-size: 25px"
href="logdashboard.php">Access Log Book</a>
<a class= "btn" style="background-color:#CCCCCC; color:#000000; font-size: 25px"
href="index.php">Bill Generation</a>
</div>
</div>
</section>
</body>
</html>
```

Insert.php

```
<?php
require('database.php');
$status = "";
if(isset($_POST['new']) && $_POST['new']==1){
   $product =$_REQUEST['product'];
   $quan = $_REQUEST['quan'];
   $price = $_REQUEST['price'];
   $ins_query="insert into inventory
   (`Products`,`Quantity(unit/kg)`,`Price(unit/kg)`)values
   ('$product','$quan','$price')";
   mysqli_query($conn,$ins_query)
   or die(mysql_error());
   $status = "New Record Inserted Successfully.
   </br></br><a href='view.php'>View Inserted Record</a>";
}
?>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Insert New Record</title>
<link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Raleway:400,500,600&Cinzel">
```

```
</head>
<body>
<div class="form">
    <div class="a">
 <a href="welcome.html">Back to home</a>
| <a href="dashboard.php">Dashboard</a>
| <a href="view.php">View Records</a>
</div>
<div class="content">
<h1>Insert New Record</h1>
<form name="form" method="post" action="">
<input type="hidden" name="new" value="1" />
<input type="text" name="product" placeholder="Enter Product" required /><br><br>
<input type="number" name="quan" placeholder="Enter Quantity" required /><br><br>
<input type="number" name="price" placeholder="Enter Price" required /><br><br>
<button name="submit" id="submit" type="submit" value="Submit">Submit</button>
</form>
<p style="color:#FF0000;"><?php echo $status; ?></p>
</div>
</div>
</body>
</html>
```

View.php

```
<?php
require('database.php');
?>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>View Records</title>
<link rel="stylesheet" href="css/style.css" />
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
<body>
<div class="form">
| <a href="welcome.html">Back to home</a>
| <a href="dashboard.php">Dashboard</a>
| <a href="insert.php">Insert New Record</a>
<div class="header">
<h1>View Records</h1>
</div>
<table width="100%" border="1" style="border-collapse:collapse;">
<thead>
```

```
<tr>
<th><strong>S.No</strong></th>
<th><strong>Products</strong></th>
<th><strong>Quantity(unit/kg)</strong></th>
<th><strong>Price(unit/kg)</strong></th>
<th><strong>Edit</strong></th>
<th><strong>Delete</strong></th>
</tr>
</thead>
<tbody>
<?php
$count=1;
$sel_query="Select * from inventory ORDER BY id ASC;";
$result = mysqli_query($conn,$sel_query);
while($row = mysqli_fetch_assoc($result)) { ?>
<tr><td align="center"><?php echo $count; ?></td>
<td align="center"><?php echo $row["Products"]; ?></td>
<td align="center"><?php echo $row["Quantity(unit/kg)"]; ?></td>
<td align="center"><?php echo $row["Price(unit/kg)"]; ?></td>

<td align="center">
<a href="edit.php?id=<?php echo $row["id"]; ?>">Edit</a>
</td>
<td align="center">
<a href="delete.php?id=<?php echo $row["id"]; ?>">Delete</a>
</td>
</tr>
<?php $count++; } ?>
</tbody>
</table>
</div>
</body>
</html>
```

Edit.php

```
<?php
require('database.php');
$id=$_REQUEST['id'];
$query = "SELECT * from inventory where id='".$id."'";
$result = mysqli_query($conn, $query) or die ( mysqli_error($conn));
$row = mysqli_fetch_assoc($result);
?>
<!DOCTYPE html>
<html>
```

```
<head>
<meta charset="utf-8">
<title>Update Record</title>
</head>
<body>
<div class="form">
<p><a href="dashboard.php">Dashboard</a>
| <a href="insert.php">Insert New Record</a>
|
<h1>Update Record</h1>
<?php
$status = "";
if(isset($_POST['new']) && $_POST['new']==1)
{
$id=$_REQUEST['id'];
$product =$_REQUEST['product'];
$quan = $_REQUEST['quan'];
$price = $_REQUEST['price'];
$update="UPDATE inventory SET `Products`='$product', `Quantity(unit/kg)`='$quan',
`Price(unit/kg)`='$price' where id='$id'";
mysqli_query($conn, $update) or die(mysqli_error($conn));
$status = "Record Updated Successfully. </br></br>
<a href='view.php'>View Updated Record</a>";
echo '<p style="color:#FF0000;">'.$status.'</p>';
}else {
?>
<div>
<form name="form" method="post" action="">
<input type="hidden" name="new" value="1" />
<input name="id" type="hidden" value="<?php echo $row['id'];?>" />
<input type="text" name="product" placeholder="Enter Product"
required value="<?php echo $row['Products'];?>" /><br>
<br>
<input type="number" name="quan" placeholder="Enter Quantity"
required value="<?php echo $row['Quantity(unit/kg)'];?>" /><br>
<br>
<input type="number" name="price" placeholder="Enter Price"
required value="<?php echo $row['Price(unit/kg)'];?>" /><br>
<br>
<button name="submit" id="submit" type="submit" value="Update">Update</button>
</form>
<?php } ?>
</div>
</div>
</body>
</html>
```

Invoice_list.php

```
<?php
session_start();
include('header.php');
include 'Invoice.php';
$invoice = new Invoice();
$invoice->checkLoggedIn();
?>
<title>Invoice list</title>
<script src="js/invoice.js"></script>
<link href="css/style1.css" rel="stylesheet">
<div class="container">
<h1 class="title">Invoice System</h2>
<?php include('menu.php');?>
<table id="data-table" class="table table-condensed table-striped">
<thead>
<tr>
<th>Invoice No.</th>
<th>Create Date</th>
<th>Customer Name</th>
<th>Invoice Total</th>
<th>Print</th>
<th>Edit</th>
<th>Delete</th>
</tr>
</thead>
<?php
$invoiceList = $invoice->getInvoiceList();
foreach($invoiceList as $invoiceDetails){
$invoiceDate = date("d/M/Y, H:i:s", strtotime($invoiceDetails["order_date"]));
echo '
<tr>
<td>'.$invoiceDetails["order_id"].'</td>
<td>'.$invoiceDate.'</td>
<td>'.$invoiceDetails["order_receiver_name"].'</td>
<td>'.$invoiceDetails["order_total_after_tax"].'</td>
<td><a href="print_invoice.php?invoice_id='.$invoiceDetails["order_id"].'" title="Print
Invoice"><span class="glyphicon glyphicon-print"></span></a></td>
<td><a href="edit_invoice.php?update_id='.$invoiceDetails["order_id"].'" title="Edit
Invoice"><span class="glyphicon glyphicon-edit"></span></a></td>
<td><a href="#" id="'.$invoiceDetails["order_id"].'" class="deleteInvoice" title="Delete
Invoice"><span class="glyphicon glyphicon-remove"></span></a></td>
</tr>
}
 ?> </table> </div>
```

Create_invoice.php

```
<?php
session_start();
include('header.php');
include 'Invoice.php';
$invoice = new Invoice();
$invoice->checkLoggedIn();
if(!empty($_POST['companyName']) && $_POST['companyName']) {
        $invoice->saveInvoice($_POST);
        header("Location:invoice_list.php");
}
?>
<title> Create Invoice </title>
<script src="js/invoice.js"></script>
<link href="css/style1.css" rel="stylesheet">
<div class="container content-invoice">
<form action="" id="invoice-form" method="post" class="invoice-form" role="form"
novalidate="">
<div class="load-animate animated fadeInUp">
<div class="row">
<div class="col-xs-8 col-sm-8 col-md-8 col-lg-8">
<h2 class="title">Invoice System</h2>
<?php include('menu.php');?>
</div>
</div>
<input id="currency" type="hidden" value="$">
<div class="row">
<div class="col-xs-12 col-sm-4 col-md-4 col-lg-4">
<h3>From,</h3>
<?php echo $_SESSION['user']; ?><br>
<?php echo $_SESSION['address']; ?><br>
<?php echo $_SESSION['mobile']; ?><br>
<?php echo $_SESSION['username']; ?><br>
</div>
<div class="col-xs-12 col-sm-4 col-md-4 col-lg-4 pull-right">
<h3>To,</h3>
<div class="form-group">
<input type="text" class="form-control" name="companyName" id="companyName"
placeholder="Company Name" autocomplete="off">
</div>
<div class="form-group">
<textarea class="form-control" rows="3" name="address" id="address" placeholder="Your
Address"></textarea>
</div></div>
</div>
```

```
<div class="row">
<div class="col-xs-12 col-sm-12 col-md-12 col-lg-12">
<table class="table table-bordered table-hover" id="invoiceItem">
<tr>
<th width="2%"><input id="checkAll" class="formcontrol" type="checkbox"></th>
<th width="15%">Item No</th>
<th width="38%">Item Name</th>
<th width="15%">Quantity</th>
<th width="15%">Price</th>
<th width="15%">Total</th>
</tr>
<tr>
<td><input class="itemRow" type="checkbox"></td>
<td><input type="text" name="productCode[]" id="productCode_1" class="form-control"
autocomplete="off"></td>
<td><input type="text" name="productName[]" id="productName_1" class="form-control"
autocomplete="off"></td>
<td><input type="number" name="quantity[]" id="quantity_1" class="form-control quantity"
autocomplete="off"></td>
<td><input type="number" name="price[]" id="price_1" class="form-control price"
autocomplete="off"></td>
<td><input type="number" name="total[]" id="total_1" class="form-control total"
autocomplete="off"></td>
</tr>
</table>
</div>
<div class="row">
<div class="col-xs-12 col-sm-3 col-md-3 col-lg-3">
<button class="btn btn-danger delete" id="removeRows" type="button">- Delete</button>
<button class="btn btn-success" id="addRows" type="button">+ Add More</button>
</div>
</div>
<div class="row">
<div class="col-xs-12 col-sm-8 col-md-8 col-lg-8">
<h3>Notes: </h3>
<div class="form-group">
<textarea class="form-control txt" rows="5" name="notes" id="notes" placeholder="Your
Notes"></textarea>
</div>
<br>
<div class="form-group">
<input type="hidden" value="<?php echo $_SESSION['userid']; ?>" class="form-control"
name="userId">
<input data-loading-text="Saving Invoice..." type="submit" name="invoice_btn" value="Save
Invoice" class="btn btn-success submit_btn invoice-save-btm">
```

```
<div class="col-xs-12 col-sm-4 col-md-4 col-lg-4">
<span class="form-inline">
<div class="form-group">
<label>Subtotal: &amp;nbsp;</label>
<div class="input-group">
<div class="input-group-addon currency">$</div>
<input value="" type="number" class="form-control" name="subTotal" id="subTotal"
placeholder="Subtotal">
</div>
<div class="form-group">
<label>Tax Rate: &amp;nbsp;</label>
<div class="input-group">
<input value="" type="number" class="form-control" name="taxRate" id="taxRate"
placeholder="Tax Rate">
<div class="input-group-addon">%</div>
</div></div>
<div class="form-group">
<label>Tax Amount: &amp;nbsp;</label>
<div class="input-group">
<div class="input-group-addon currency">$</div>
<input value="" type="number" class="form-control" name="taxAmount" id="taxAmount"
placeholder="Tax Amount">
</div>
<div class="form-group">
<label>Total: &amp;nbsp;</label>
<div class="input-group">
<div class="input-group-addon currency">$</div>
<input value="" type="number" class="form-control" name="totalAftertax" id="totalAftertax"
placeholder="Total">
<div class="form-group">
<label>Amount Paid: &amp;nbsp;</label>
<div class="input-group">
<div class="input-group-addon currency">$</div>
<input value="" type="number" class="form-control" name="amountPaid" id="amountPaid"
placeholder="Amount Paid">
</div>
<div class="form-group">
<label>Amount Due: &amp;nbsp;</label>
<div class="input-group">
<div class="input-group-addon currency">$</div>
<input value="" type="number" class="form-control" name="amountDue" id="amountDue"
placeholder="Amount Due">
<div class="clearfix"></div>
</form>
```

7.2 APPENDIX 2 – SCREENSHOTS/OUTPUTS

HOME

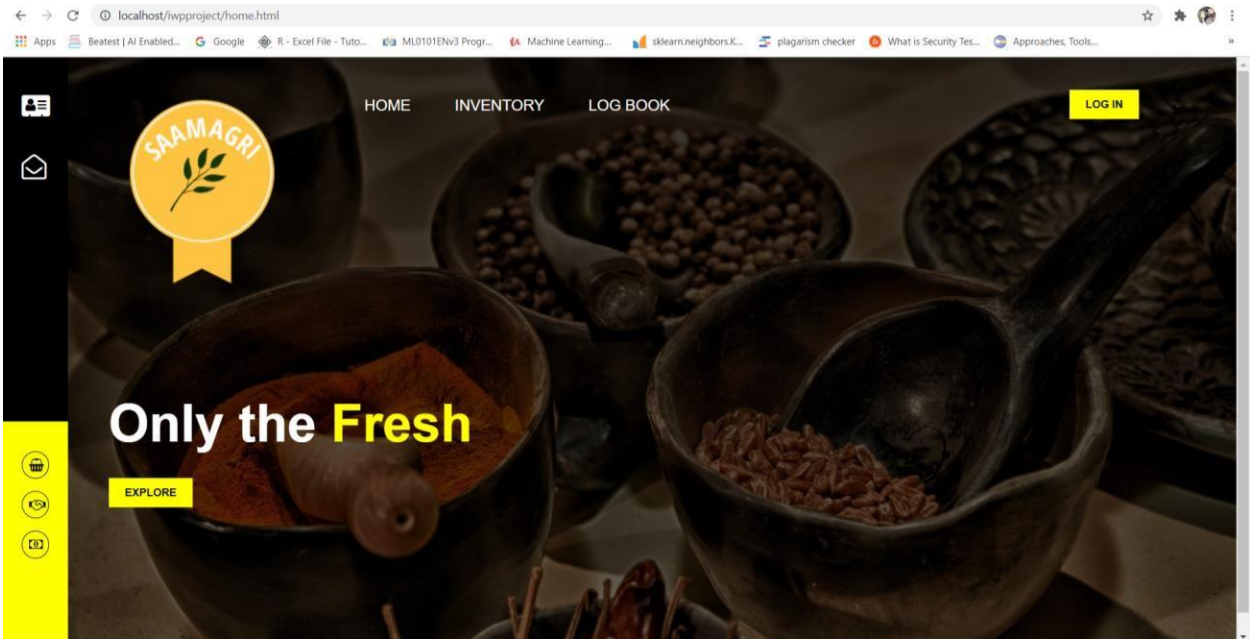


FIG. 18

TEAM

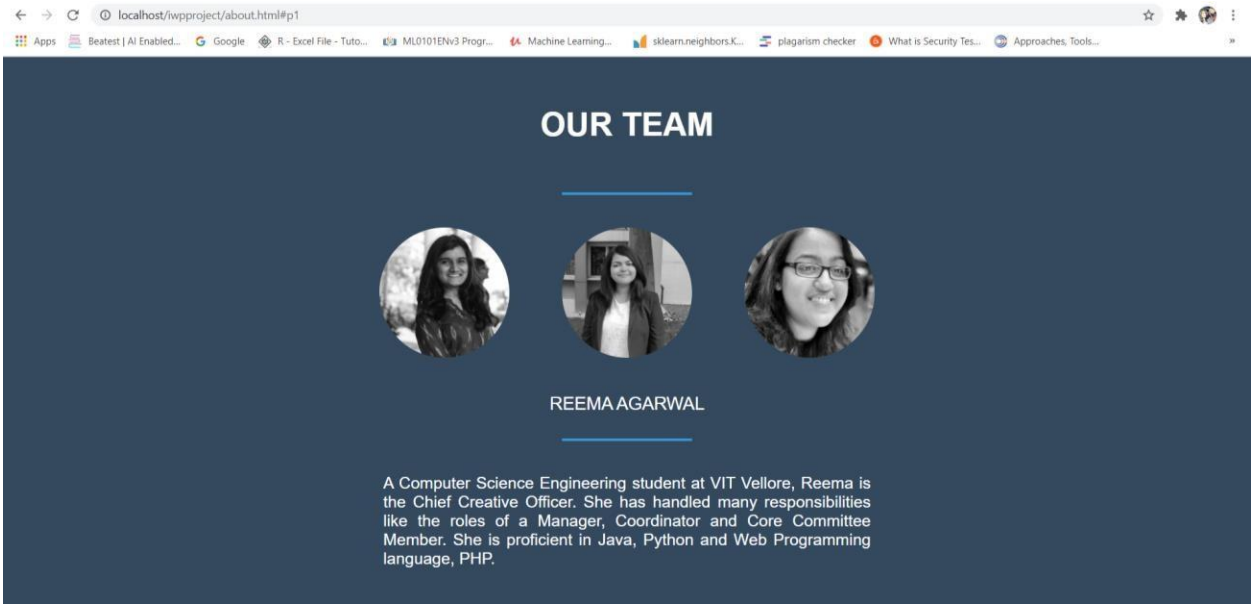


FIG. 19

EXPLORE

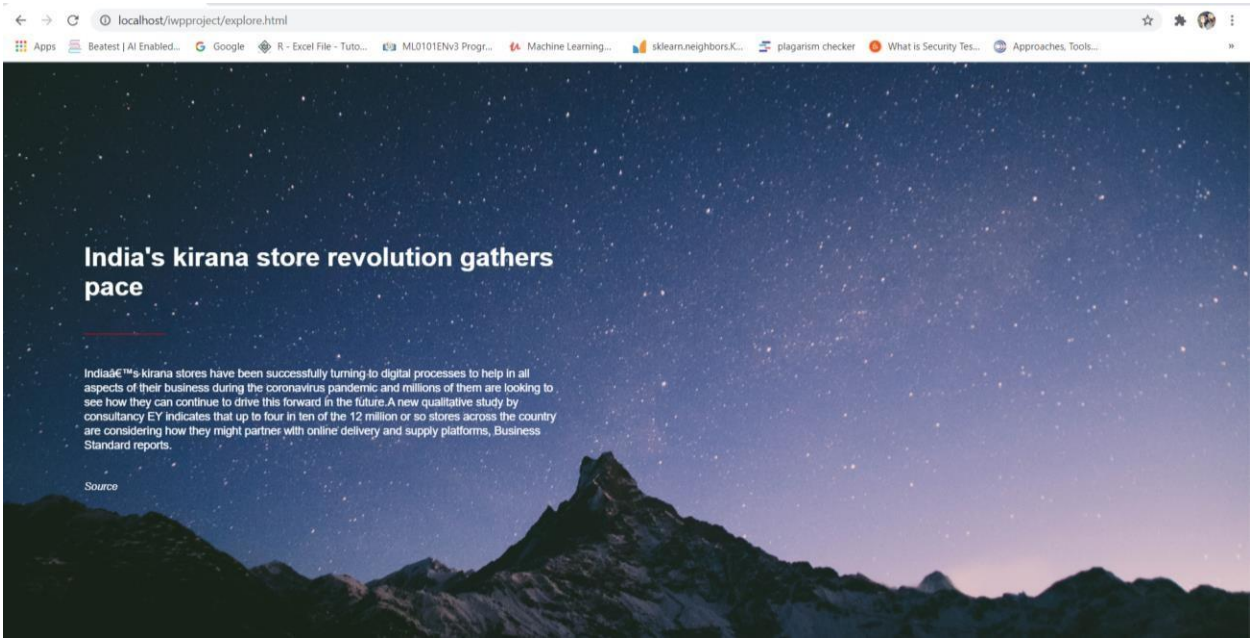


FIG. 20

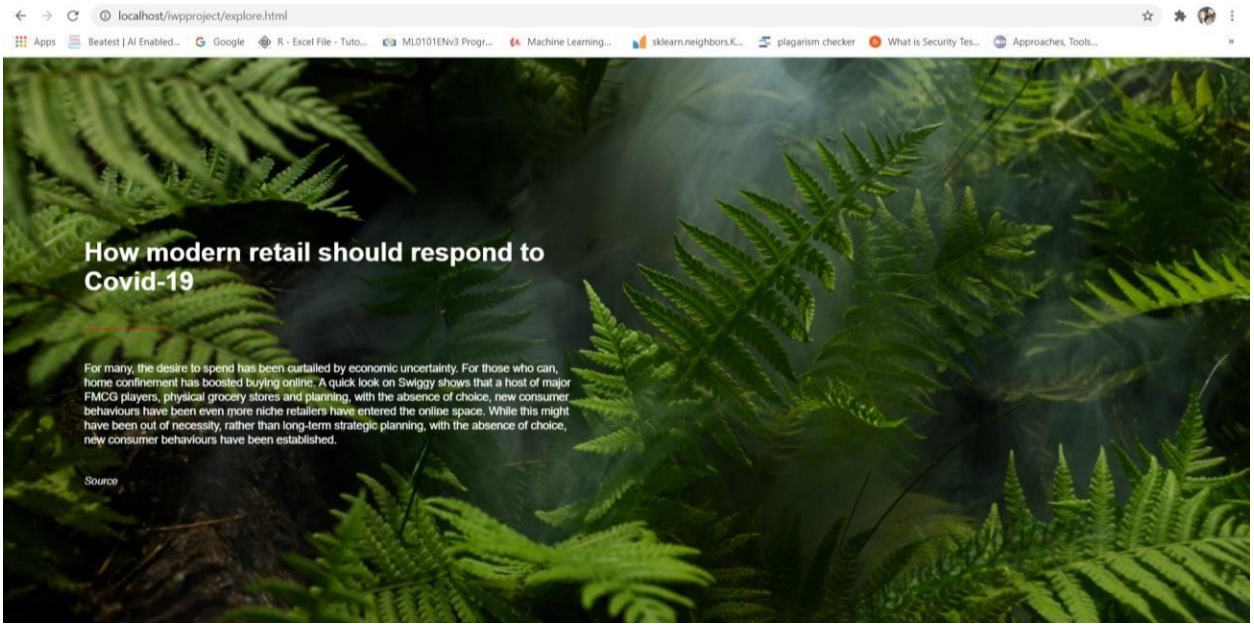


FIG. 21

INVENTORY

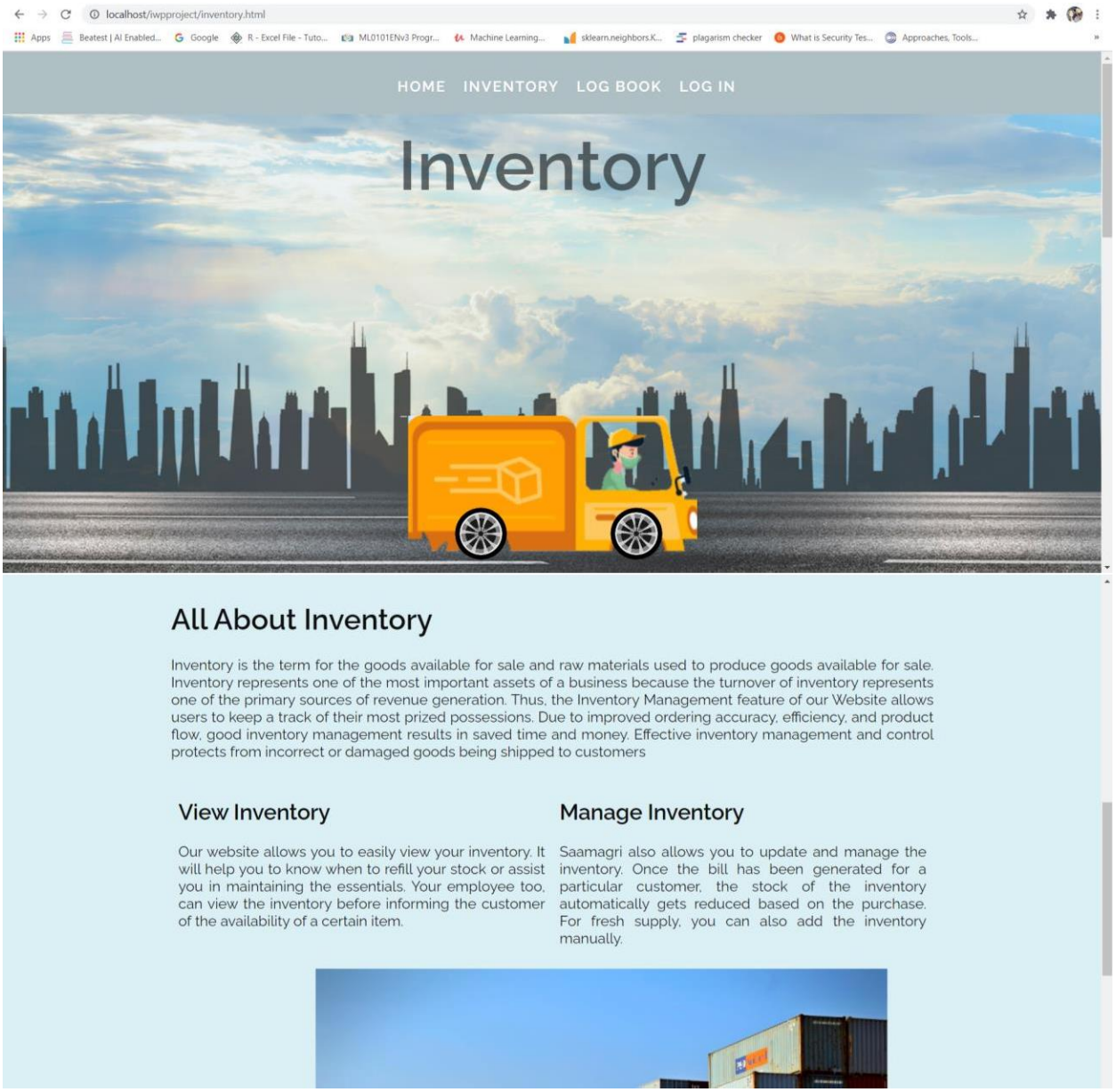


FIG. 22

LOGBOOK

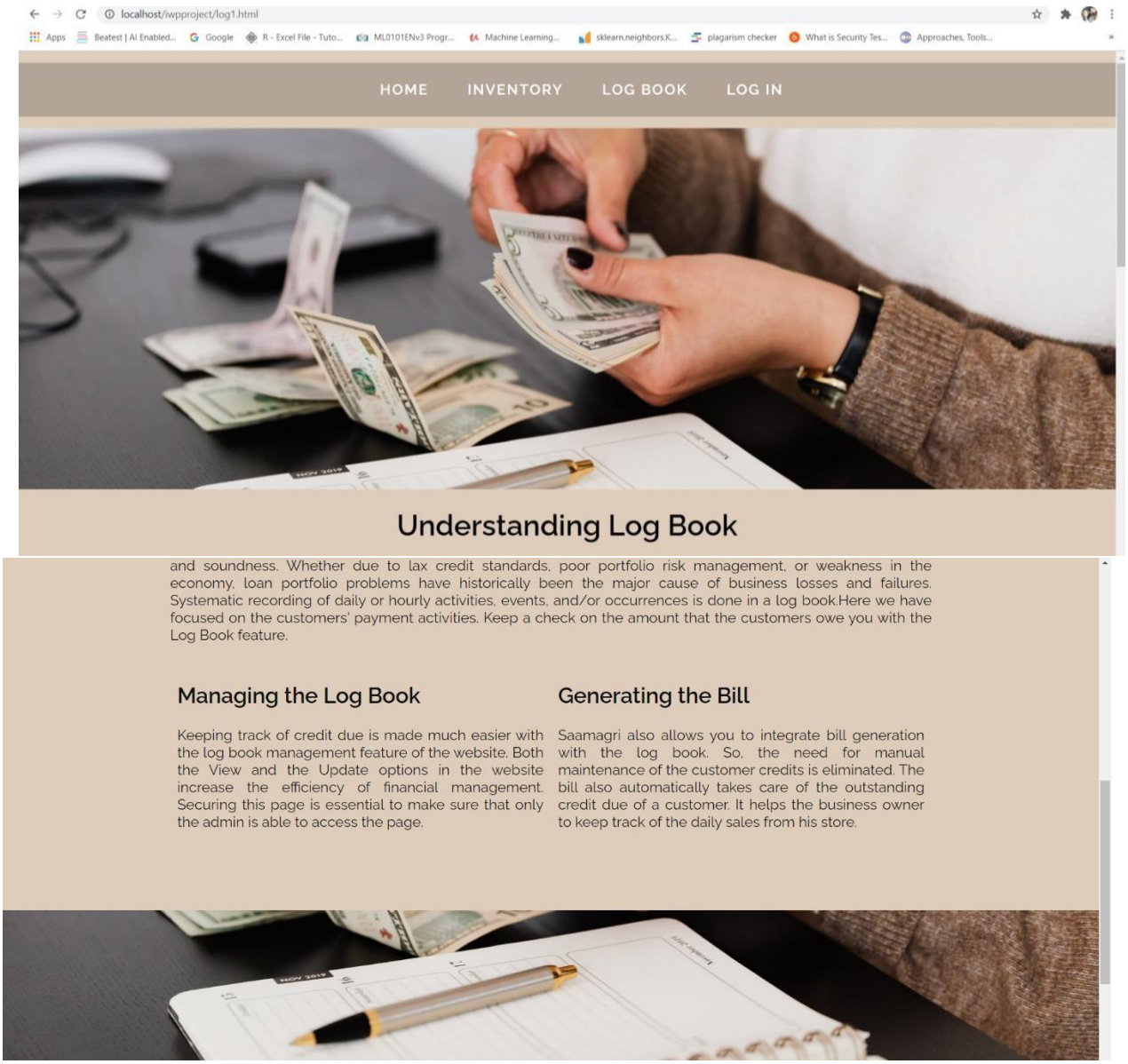


FIG. 23

LOGIN

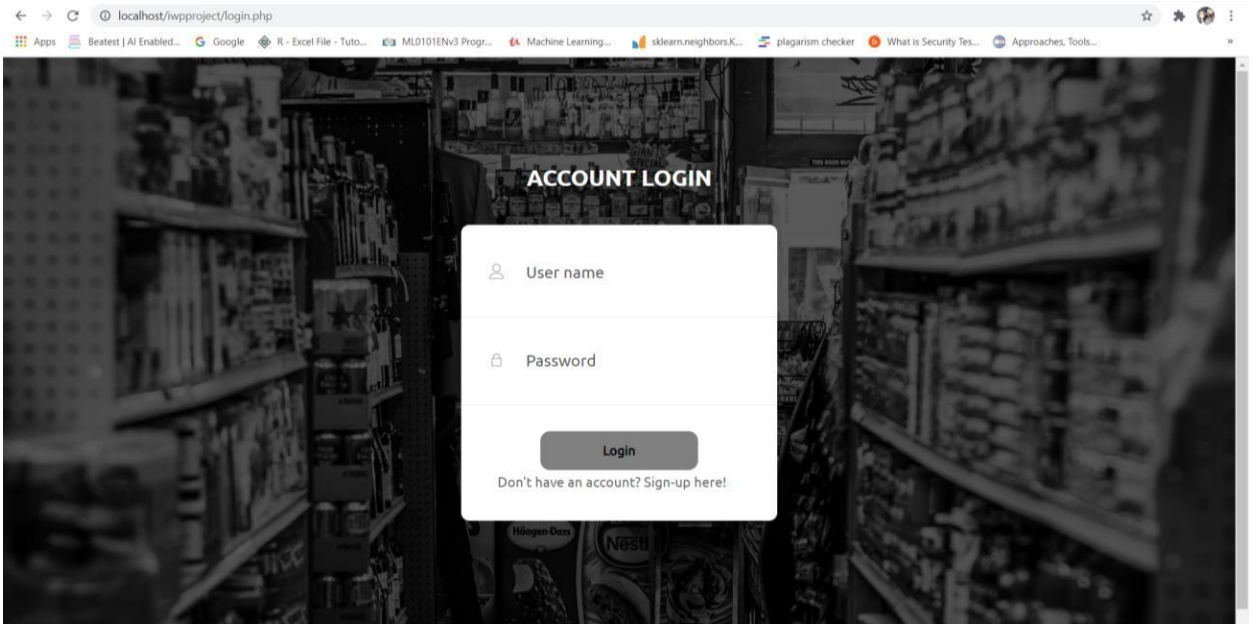


FIG. 24

SIGN UP

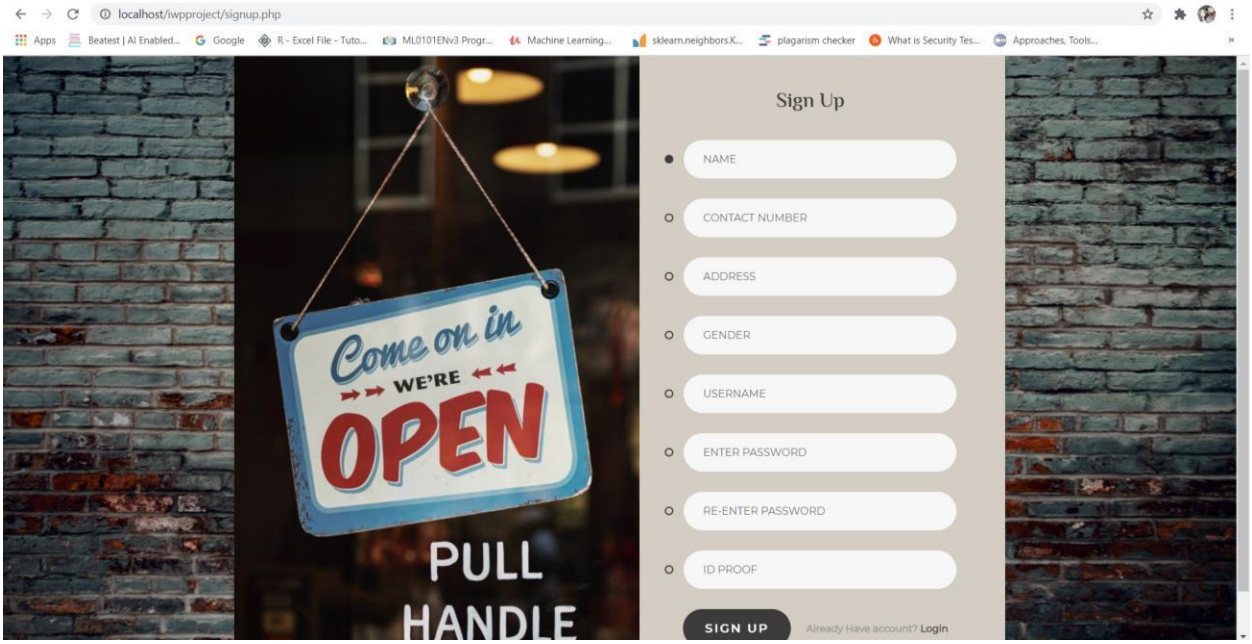


FIG. 25

WELCOME

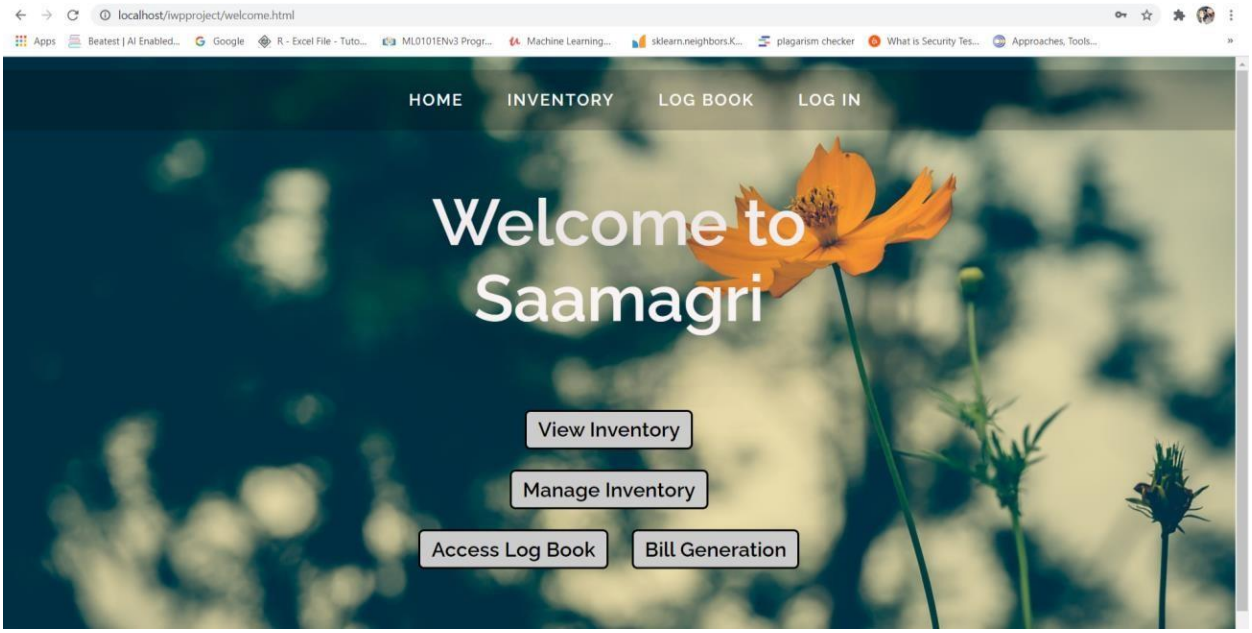
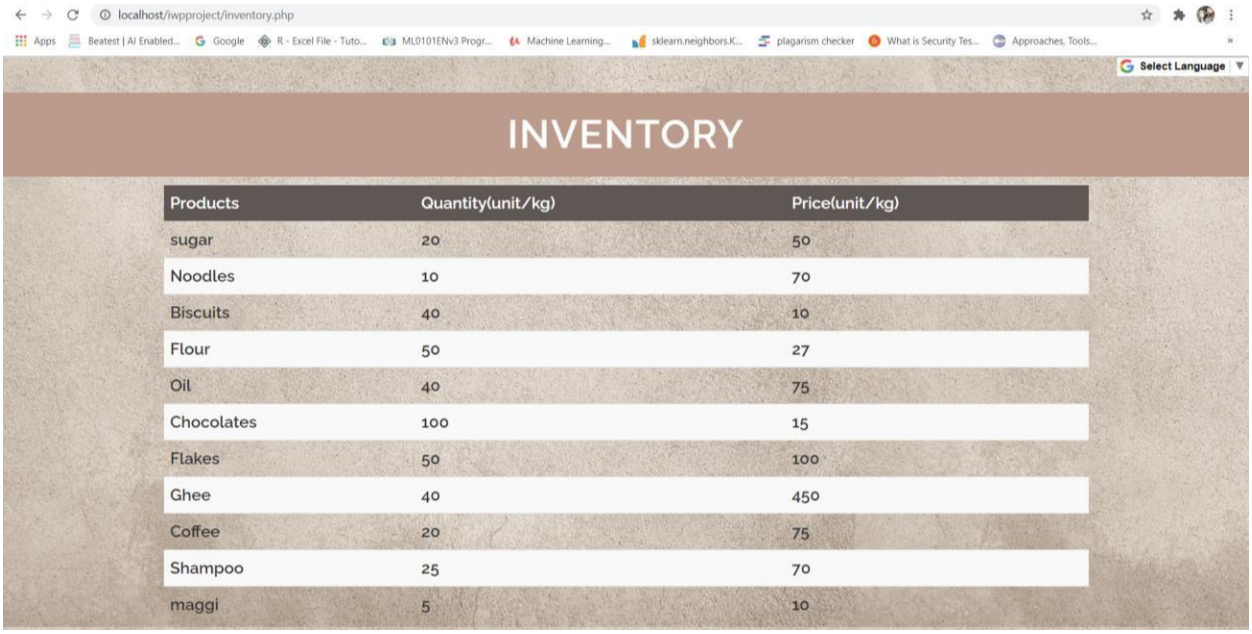


FIG. 26

VIEW INVENTORY



Products	Quantity(unit/kg)	Price(unit/kg)
sugar	20	50
Noodles	10	70
Biscuits	40	10
Flour	50	27
Oil	40	75
Chocolates	100	15
Flakes	50	100
Ghee	40	450
Coffee	20	75
Shampoo	25	70
maggi	5	10

FIG. 27

MANAGE INVENTORY

DASHBOARD PAGE

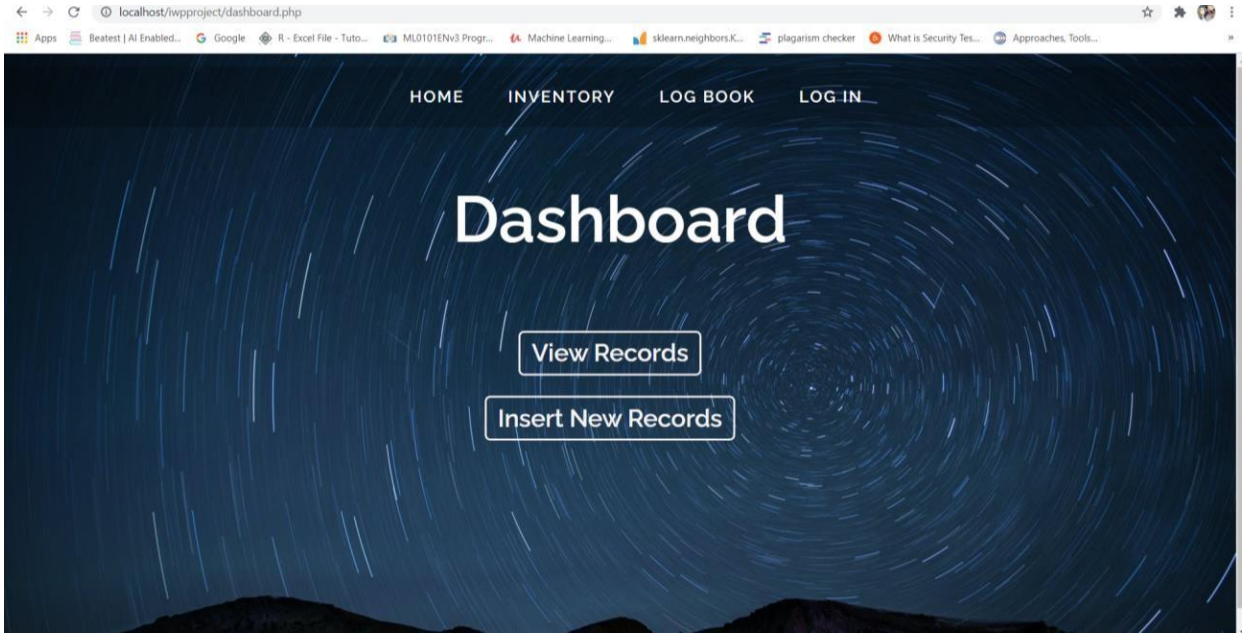
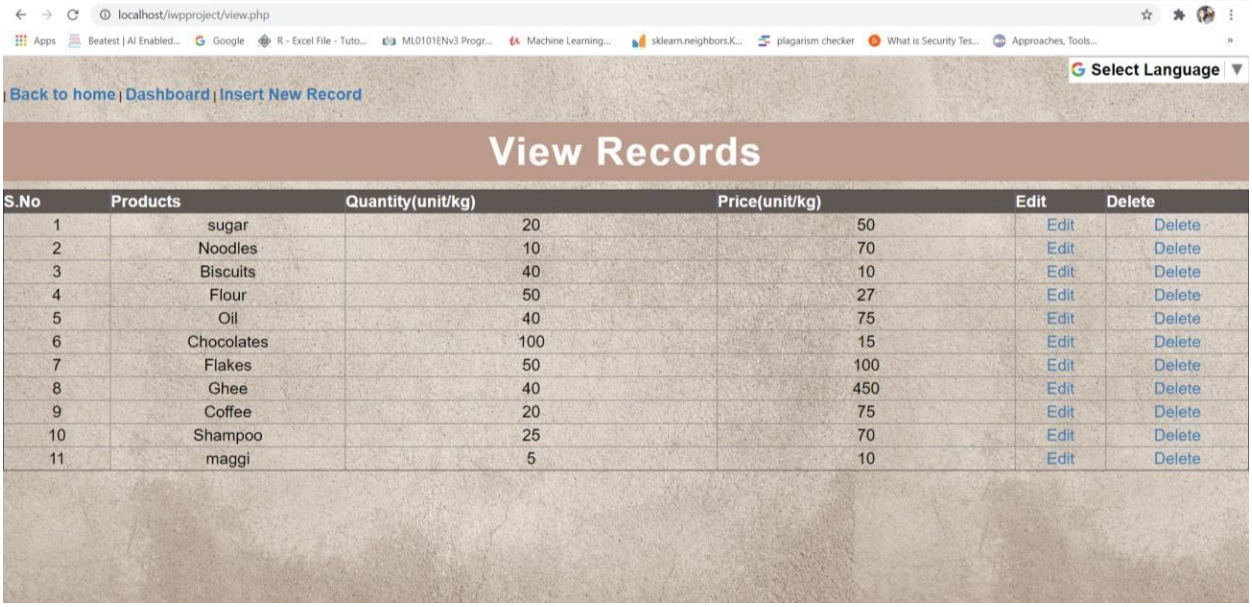


FIG. 28

VIEW RECORD FOR INVENTORY



S.No	Products	Quantity(unit/kg)	Price(unit/kg)	Edit	Delete
1	sugar	20	50	Edit	Delete
2	Noodles	10	70	Edit	Delete
3	Biscuits	40	10	Edit	Delete
4	Flour	50	27	Edit	Delete
5	Oil	40	75	Edit	Delete
6	Chocolates	100	15	Edit	Delete
7	Flakes	50	100	Edit	Delete
8	Ghee	40	450	Edit	Delete
9	Coffee	20	75	Edit	Delete
10	Shampoo	25	70	Edit	Delete
11	maggi	5	10	Edit	Delete

FIG. 29

EDIT RECORD FOR INVENTORY

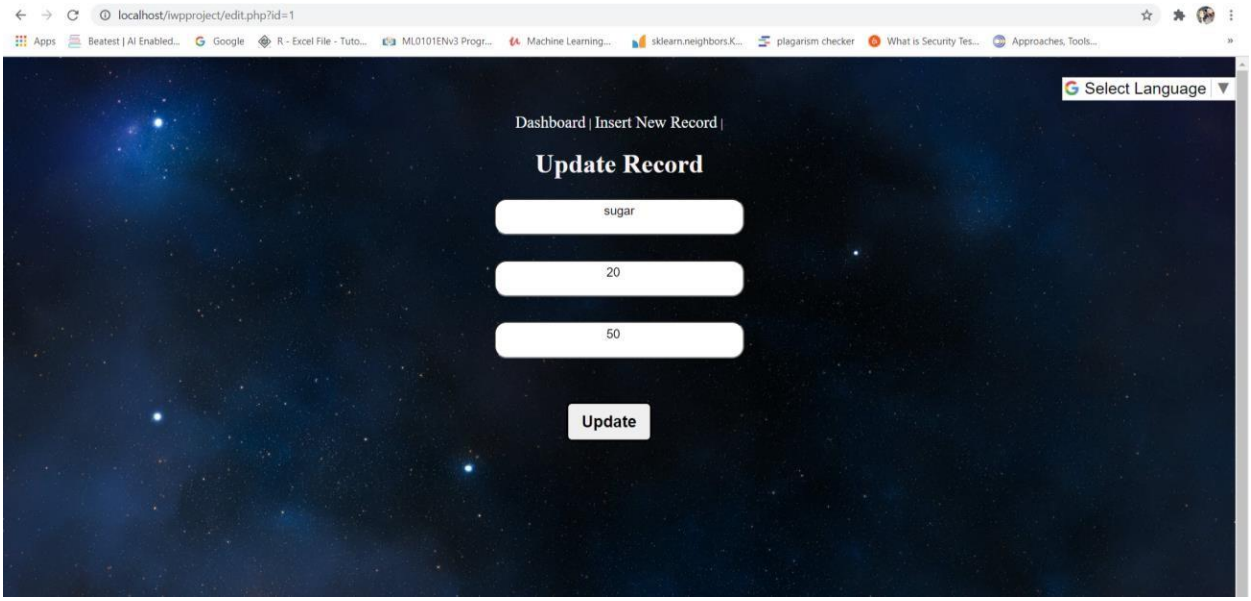


FIG. 30

INSERT NEW RECORD FOR INVENTORY

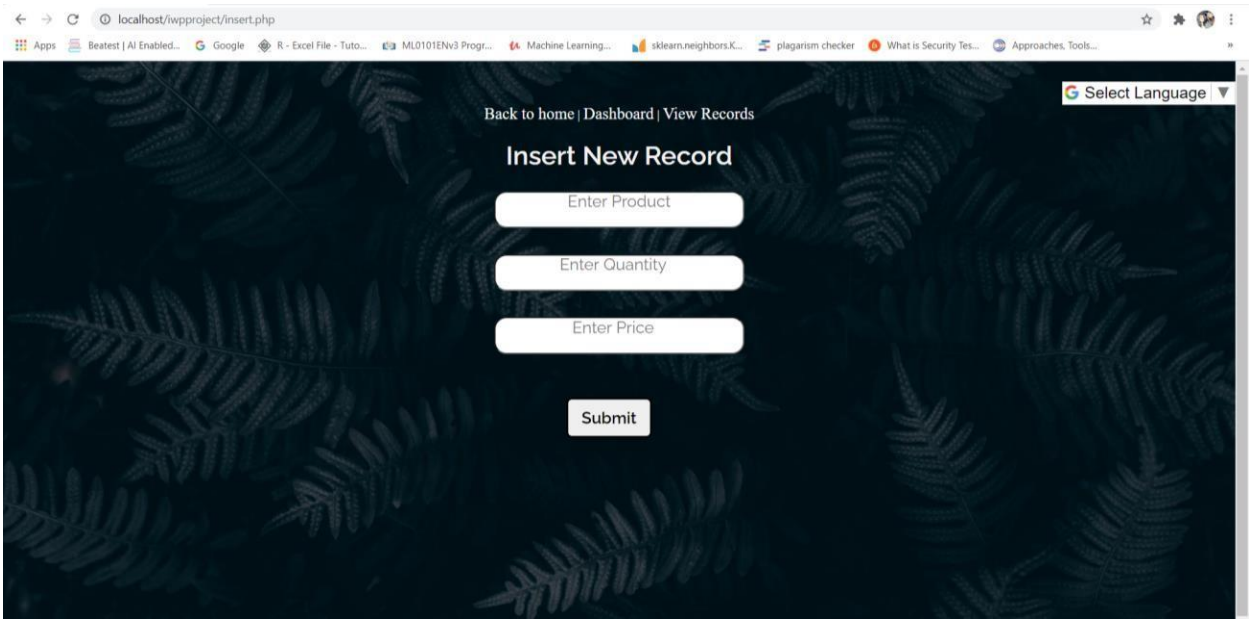


FIG. 31

ACCESS LOGBOOK

LOGBOOK DASHBOARD

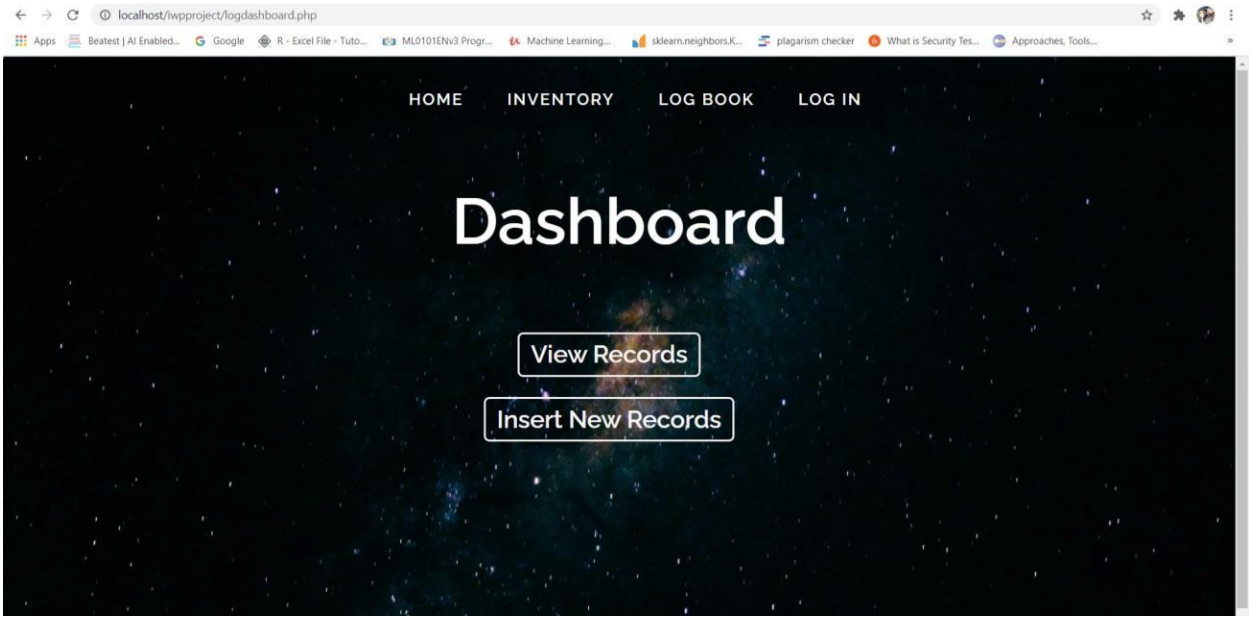


FIG. 32

VIEW RECORDS FOR CUSTOMER DETAILS

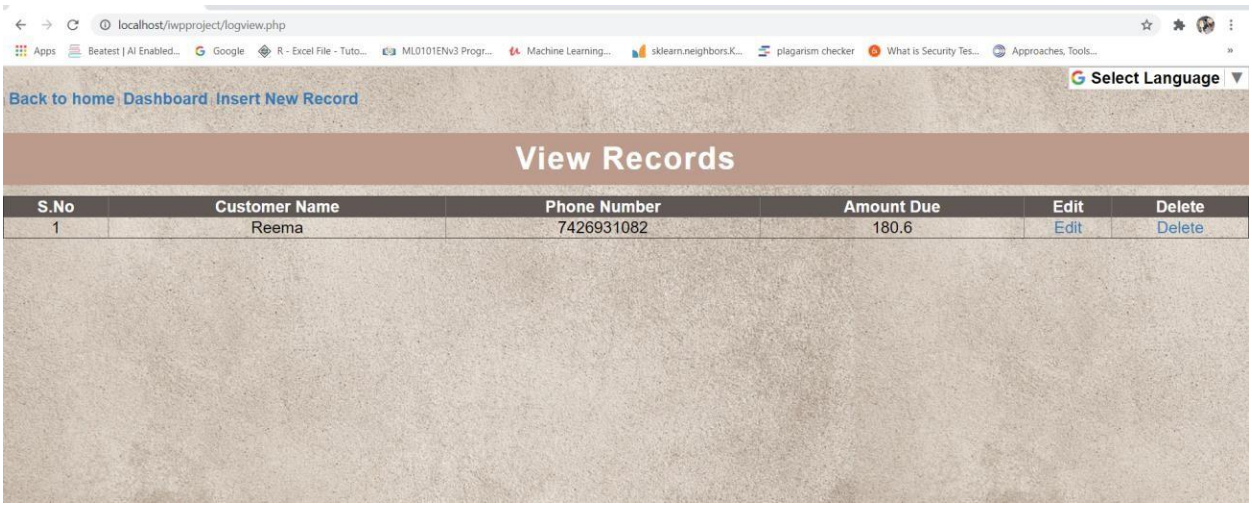


FIG. 33

UPDATE RECORDS FOR CUSTOMER DETAILS

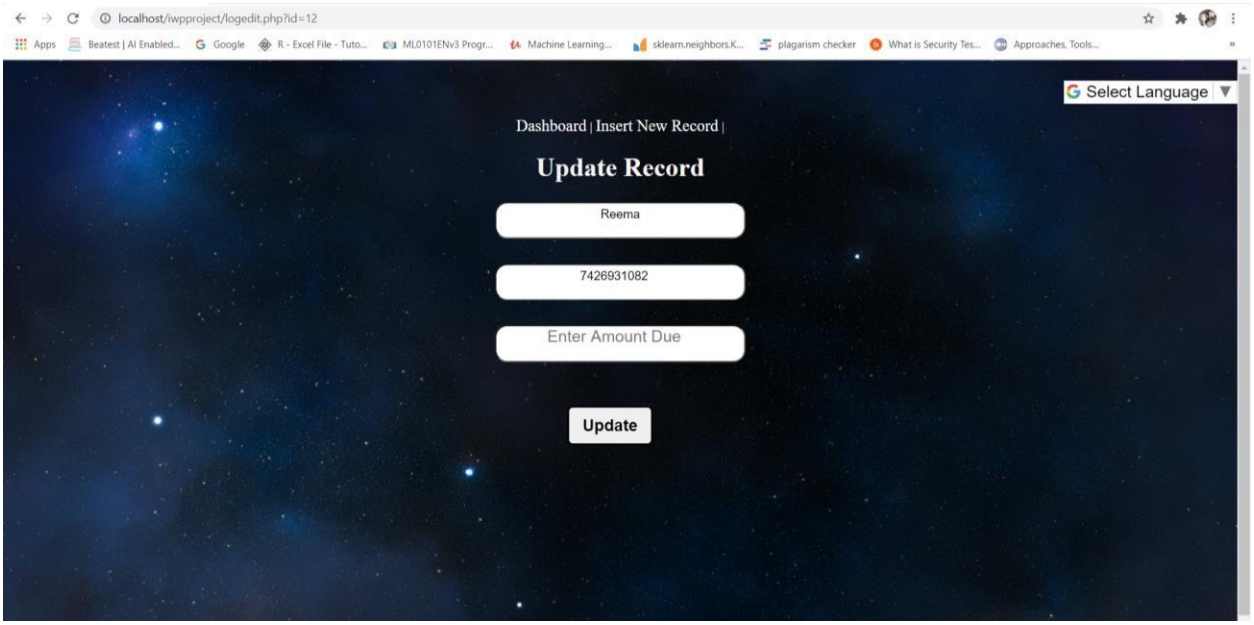


FIG. 34

INSERT NEW RECORDS FOR CUSTOMER DETAILS

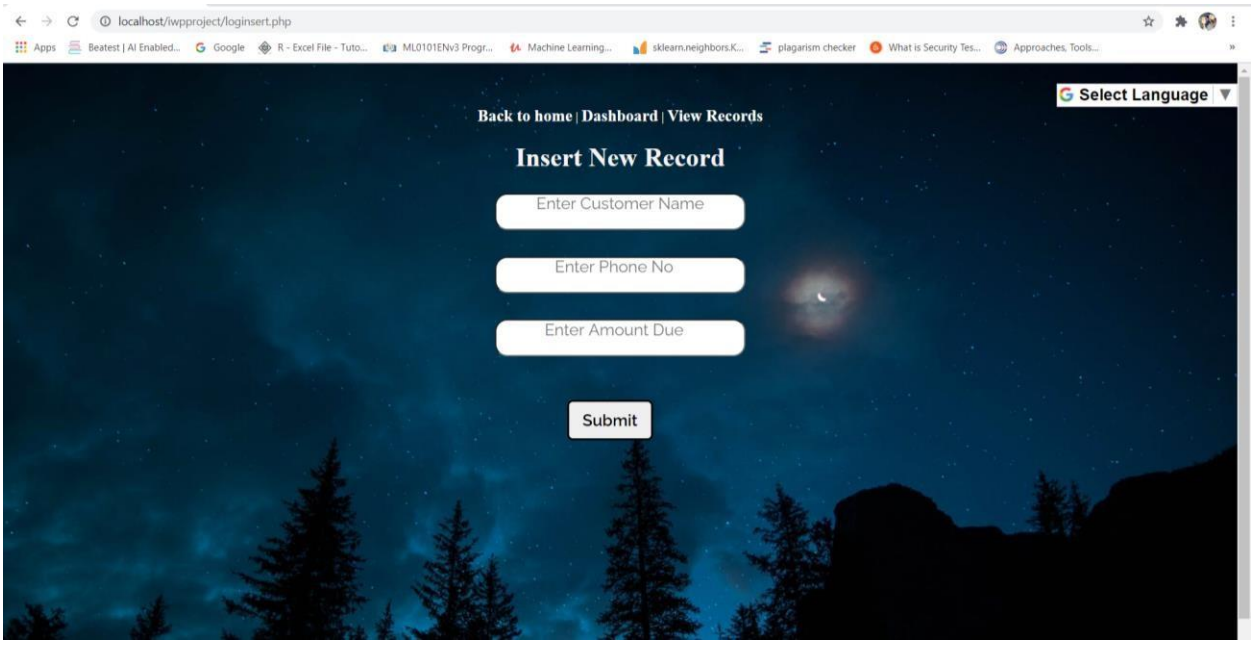


FIG. 35

LOGIN FOR INVOICE

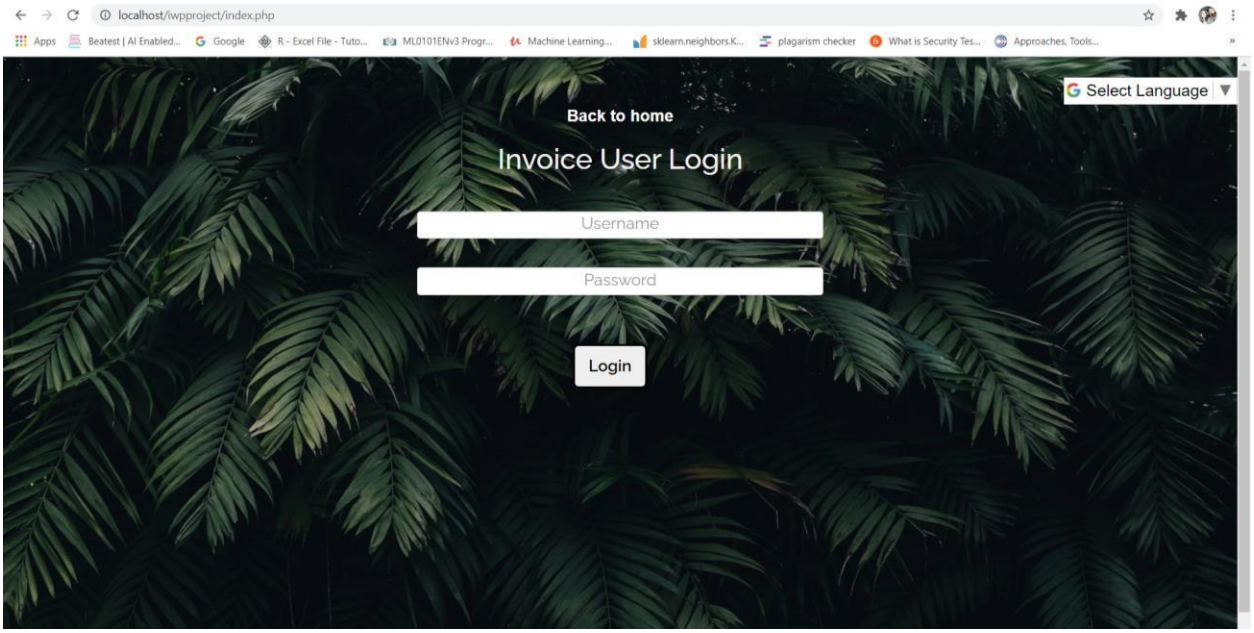


FIG. 36

INVOICE LIST

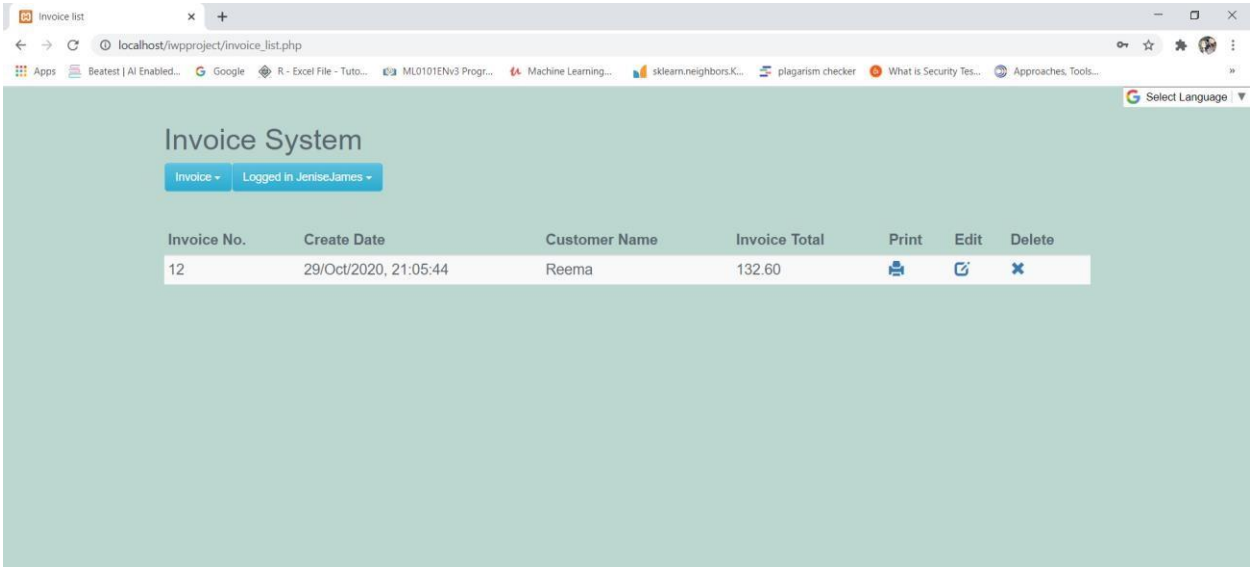


FIG. 37

EDIT INVOICE

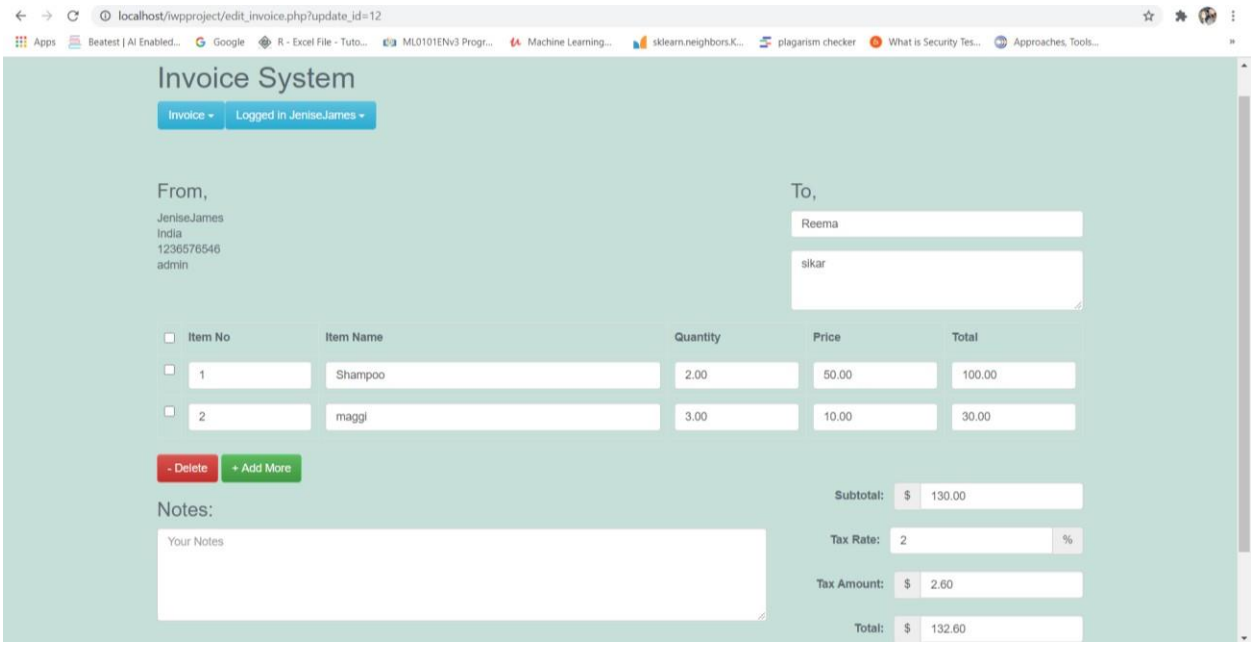


FIG. 38

CREATE INVOICE

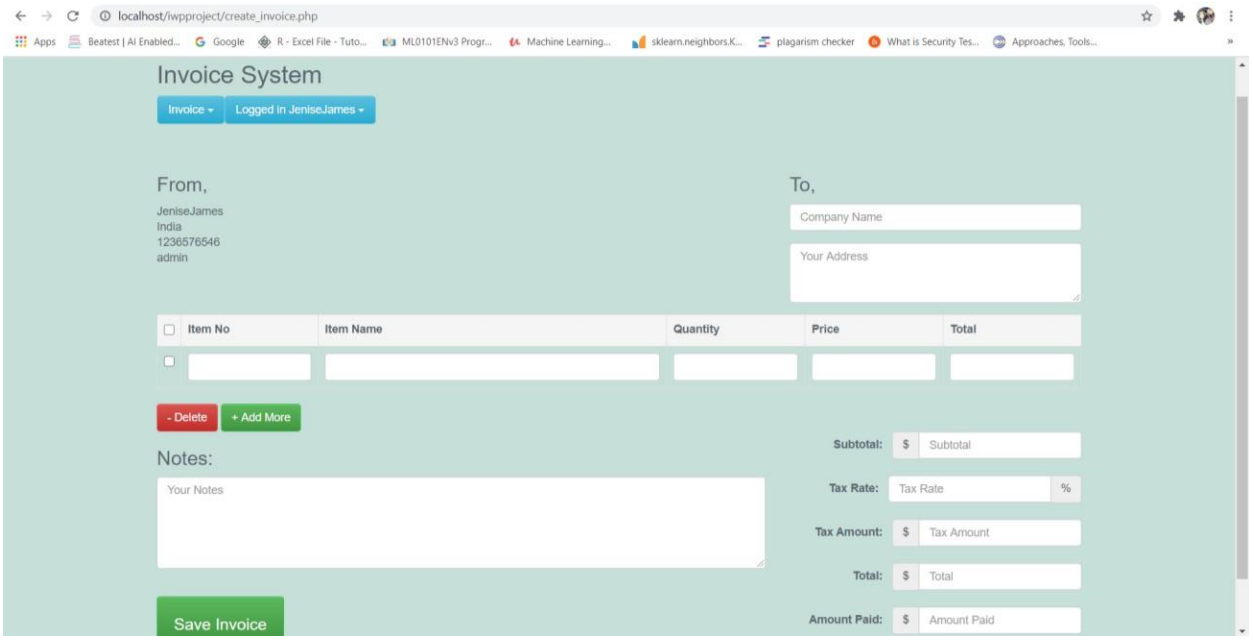


FIG. 39

PRINT INVOICE

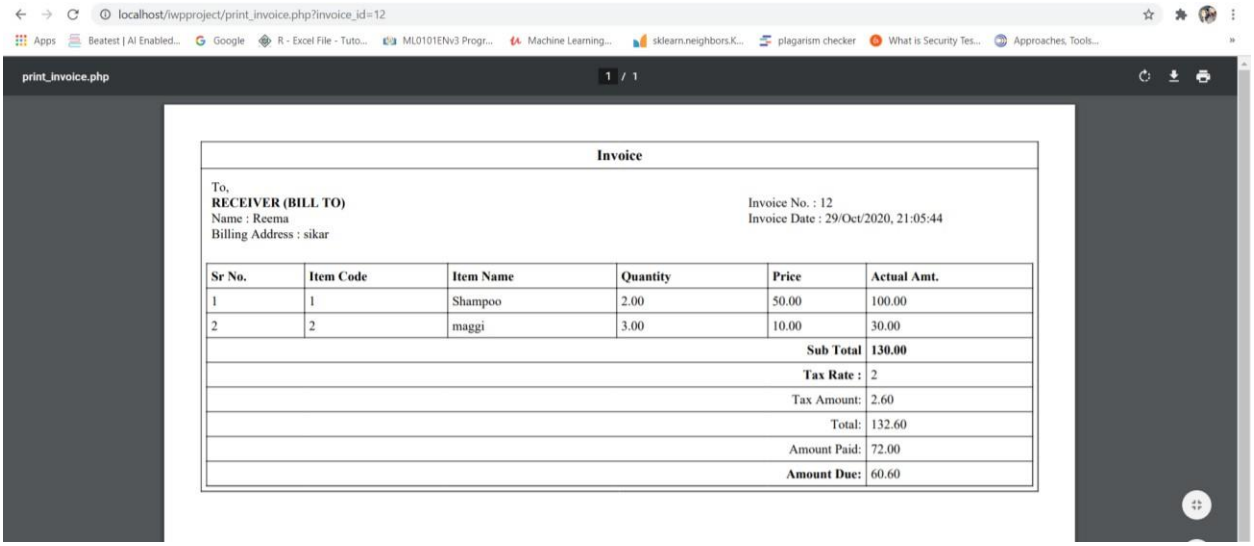


FIG. 40

WELCOME PAGE IN HINDI

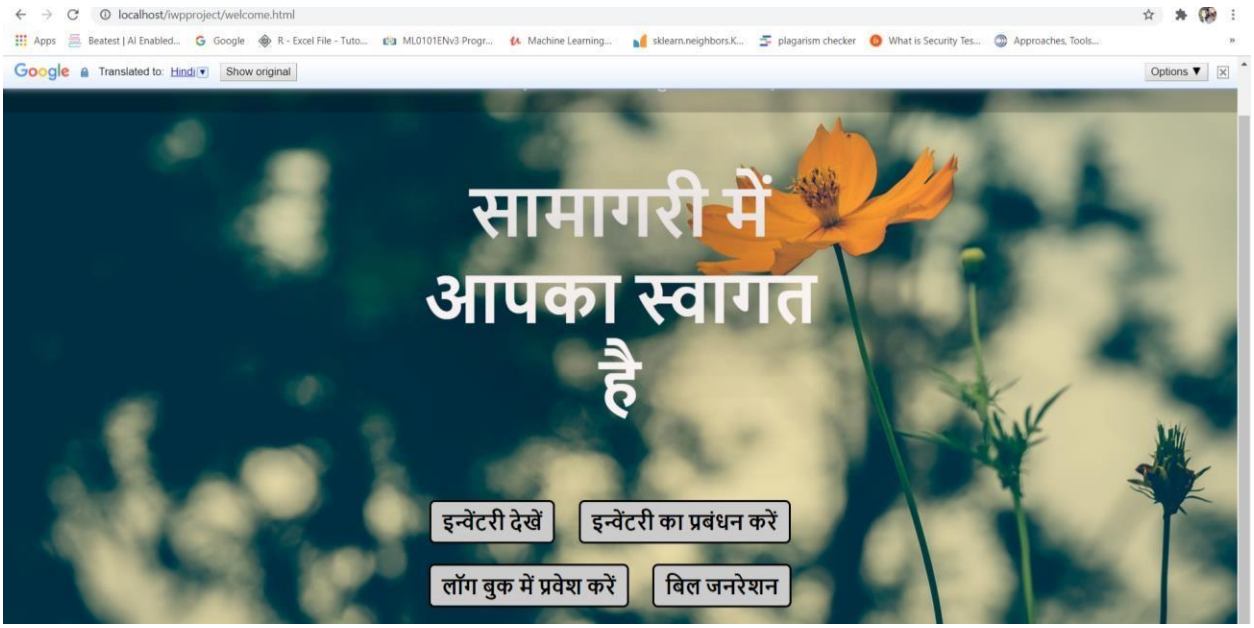


FIG. 41

INVENTORY PAGE IN TAMIL

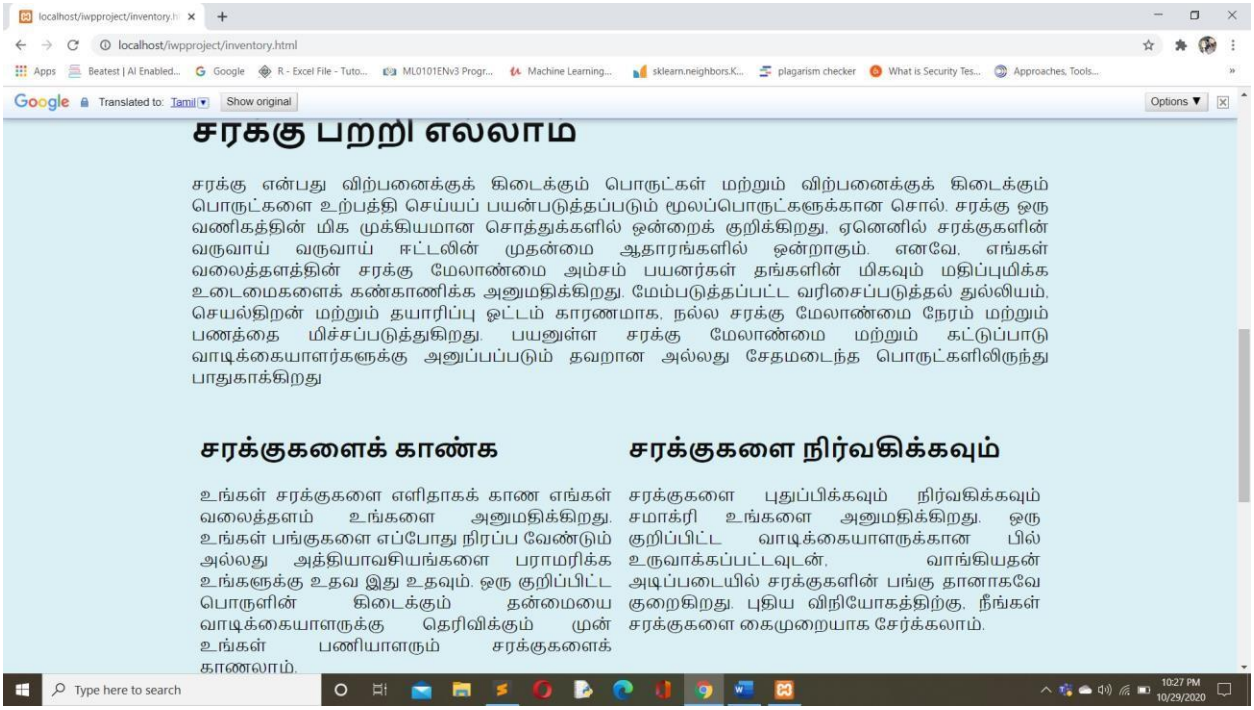
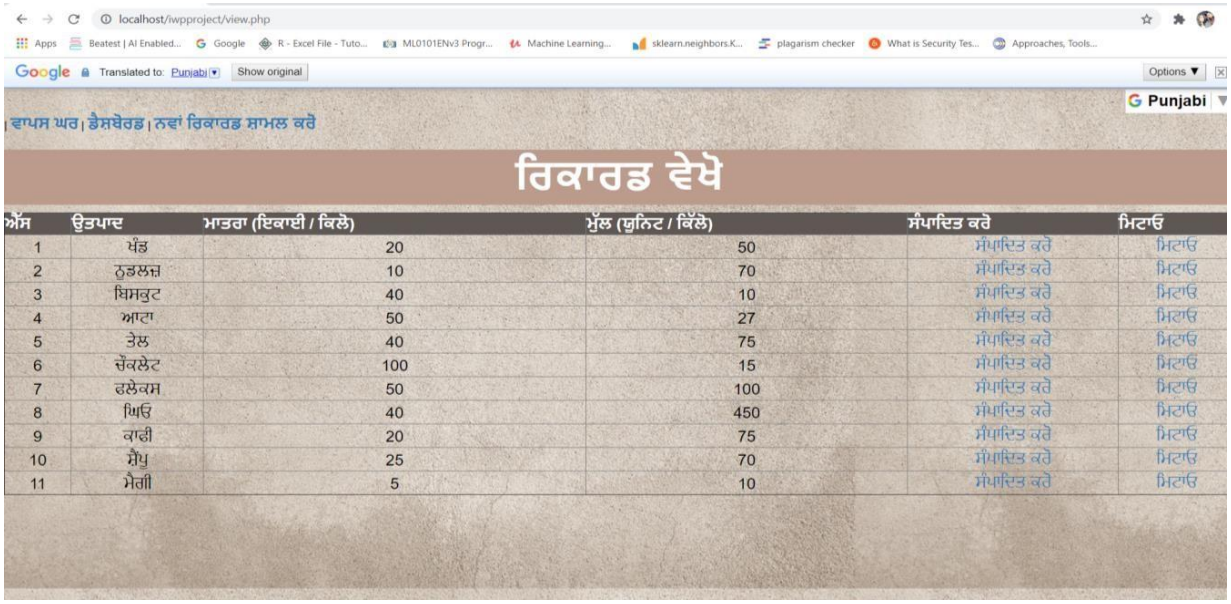


FIG. 42

VIEW INVENTORY IN PUNJABI



ਐੱਸ	ਉਤਪਾਦ	ਮਾਤਰਾ (ਇਕਾਈ / ਕਿਲੋ)	ਮੁੱਲ (ਯੂਨਿਟ / ਕਿੱਲੋ)	ਸੰਪਾਦਿਤ ਕਰੋ	ਮਿਟਾਓ
1	ਖੰਡ	20	50	ਸੰਪਾਦਿਤ ਕਰੋ	ਮਿਟਾਓ
2	ਨੂਡਲਜ਼	10	70	ਸੰਪਾਦਿਤ ਕਰੋ	ਮਿਟਾਓ
3	ਬਿਸਕੁਟ	40	10	ਸੰਪਾਦਿਤ ਕਰੋ	ਮਿਟਾਓ
4	ਆਟਾ	50	27	ਸੰਪਾਦਿਤ ਕਰੋ	ਮਿਟਾਓ
5	ਤੇਲ	40	75	ਸੰਪਾਦਿਤ ਕਰੋ	ਮਿਟਾਓ
6	ਚੌਕਲੇਟ	100	15	ਸੰਪਾਦਿਤ ਕਰੋ	ਮਿਟਾਓ
7	ਫਲੇਕਸ	50	100	ਸੰਪਾਦਿਤ ਕਰੋ	ਮਿਟਾਓ
8	ਘਿਓ	40	450	ਸੰਪਾਦਿਤ ਕਰੋ	ਮਿਟਾਓ
9	ਕਾਫੀ	20	75	ਸੰਪਾਦਿਤ ਕਰੋ	ਮਿਟਾਓ
10	ਸ਼ੈਂਪੂ	25	70	ਸੰਪਾਦਿਤ ਕਰੋ	ਮਿਟਾਓ
11	ਮੈਗੀ	5	10	ਸੰਪਾਦਿਤ ਕਰੋ	ਮਿਟਾਓ

FIG. 43

8. REFERENCES

8.1 LIST OF WEBSITES (URLs)

- https://www.geeksforgeeks.org/web-technology/
- https://www.w3schools.com/php/default.asp
- https://www.javatpoint.com/sql-tutorial