**CHAPTER 1**

**INTRODUCTION**

**1.1 INTRODUCTION TO DBMS**

**Database** is a collection of related data and data is collection of facts and figure that can be procedure information.

Mostly data represents recordable facts. Data aids in producing information, which is based on facts. For example, if we have data about marks obtained by all students, we can then conclude about toppers and average marks.

A Database management system stores data in such a way that it becomes easier to retrieve, manipulate, and produce information.

**Characteristics of Database Management System :**

Traditionally, data was organized in file formats. DBMS was a new concept then, and all the research was done to make it overcome the deficiencies in traditional style of data management. A modern DBMS has the following characteristics,

* Real-word entity
* Relation-based tables
* Isolation of data and application
* Less redundancy
* Consistency
* Query language
* ACID Properties
* Multiple views
* Multi users and concurrent access
* Security

**Advantages and Disadvantages of DBMS:**

**Advantages:**

* Reduction of Redundancy :This is perhaps the most significant advantage of using DBMS. Redundancy creates several problems like, requiring extra storage space, entering same data more than once during data insertion, and deleting data from more than one place during deletion.
* Sharing of Data :In paper-based record keeping, data cannot be shared among many users. But in DBMS, many users can share the same database if they are connected via network.

**Disadvantages:**

* As DBMS needs computer, we have to invest a good amount in acquiring the hardware, software, installation facilities and training of user.
* We need to keep regular backups because a failure can occur any time. Taking backup is a lengthy process and the computer system cannot perform any other jobs at this times.

**1.2 ABOUT SQL**

SQL is a language to operate database; it includes database creation, deletion, fetching rows, modifying rows, etc. SQL is an ANSI (American National Standards Institute) standard language, but there are many different versions of SQL language.

**What is SQL :**

SQL is a Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relation database.

SQL is the standard language for Relational Database System. All the Relation Database

Management System (RDBMS) like My SQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

**Why SQL ?**

SQL is widely popular because it offers the following advantages:

* Allows user to access data in the RDBMS.
* Allows the user to describe the data.
* Allows user to create and drop database and table.
* Allows user to set permission on table, procedures and views.
* Allows user to create view, stored procedure, function in a database.
* Allows user to define data in a database and manipulate that data.
* Allows to embed within other language using SQL modules, libraries and pre- compilers

**Brief history of SQL :**

* 1970 – Dr. Edgar F. “Ted” Codd of IBM is known as the father of RDBMS. He described the relational model for database.
* 1974 – Structured Query Language appeared.
* 1978 – IBM worked to develop Codd’s ideas and released a product named System/R.
* 1986 – IBM developed the first prototype of relational database and standardized by ANSI. The first relational database was released by Relational Software which later came to be known as ORACLE.

**SQL Commands :**

**DDL – Data Definition Language**

|  |  |
| --- | --- |
| **Commands** | **Description** |
| CREATE | Creates a new table. |
| ALTER | Modifies an existing database objects, such as tables. |
| DROP | Deletes the table, a view or other objects of the table. |

**Fig 1.1 :** Data Definition Language

**DML – Data Manipulation Language**

|  |  |
| --- | --- |
| **Commands** | **Description** |
| SELECT | Retrieves certain records from one or more tables. |
| INSERT | Creates a record. |
| UPDATE | Modifies the record. |
| DELETE | Deletes records |

**Fig 1.1 :** Data Manipulation Language

**DCL – Data Control Language**

|  |  |
| --- | --- |
| **Commands** | **Description** |
| GRANT | Gives a privilege to the user. |
| REVOKE | Takes back privileges granted from user. |

**Fig 1.1 :** Data Control Language

**1.3 PROJECT DESCRIPTION**

**Overview**

The project saccharine management system is developed for a sweet shop. Saccharine management system is designed for easy, effective, time saving and paper free management of customers and order records. This report discusses the result of the work done in development of Saccharine Management System on “PHP” front-end platform and “MySQL" as back-end platform.

**Problem Statement**

A successful sweet shop owner needs a platform to sell the sweets. Sales through markets may not get enough profits and the presence of middlemen causes the chances of owners even losing their margin price or the best price for their sweets.

**Solution**

The solution for the above problem defined is that the owner needs to provide the details of the sweet which is for sale to the customers by using the application which is developed. As the application receives the details of sweet it will processed and placed in view of customer.

**Project Scope**

The scope of the project is clear to give a simple and attractive application to simplify the work as well as to reduce the efforts while doing it offline or we can say by doing it with old methods. In this application we are able to save database of sweets present on the site.

To avoid the limitations of the existing manual system we are coming up with a webpage application which helps the customer to get sweets directly to the place where they need. The proposed systems help them in many ways. It helps them do billing very easily. Account maintenance also becomes easier. They can keep track of their sales and account details. The software is provided with all the entries to entry any new product or customer to add or modify and delete.

The proposed system maintains the information in the database. It is used as an internet application. This system provides high security. This system provides easy business

solutions to the users.

**Advantages**

* Offers and discount.
* Plenty of choices.
* Easy to buy.
* Optimum level of service to customers department.
* Data manipulations are more reliable and accurate.
* Portable and flexible.
* Prevent data redundancy.
* Reduce the work load of employees.

**CHAPTER 2**

**REQUIREMENTS SPECIFICATION**

* 1. **SOFTWARE SPECIFICATION**

Operating System : Windows 10

Front End : HTML and CSS

Back End : Maria DB

Tool : XAMPP

Web Server : Apache

Scripting language : PHP

**2.2 HARDWARE SPECIFICATION**

Processor : Intel dual core

Hard Disk : 1 TB HDD

RAM : 8 GB

**CHAPTER 3**

**DESIGN**

**3.1 SCHEMA DIAGRAM**

**SWEET**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| sid | sname | stype | quantity | sprice |

**BRANCH**

|  |  |
| --- | --- |
| bid | blocation |

**CUSTOMER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cid | cname | phone | address | bid |

**ORDERS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| oid | odate | cid | bid | sid | quantity |

**PAYMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| payno | ptype | pamt | oid |

**DELIVERY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| dno | dplace | deldate | payno | cid | status |

**FEEDBACK**

|  |  |  |  |
| --- | --- | --- | --- |
| cid | oid | rating | description |

**Fig 3.1 :** Schema Diagram

**3.2 ER DIAGRAM**

select

M N

sweet

customer

N

1

branch

checks for

feedback

M

gives

N N

M

book the

has

orders

delivery

payment

N

1

1

1

have

1

**Fig 3.2 :** ER Diagram

**3.3 DATAFLOW DIAGRAMS**

Request to login Check details

Admin

Admin table

Response Response from db

Request for add

Sweet table

Response from db

Request for view

Orders table

Response from db

delivery table

Add to delivery

Response from db

**Fig 3.3(a) :** Admin DFD

Request to login Check details

customer table

Customer

Response Response from db

Request for view

sweet table

Response from db

Add to order

orders table

Response from db

Add to payment

payment table

Response from db

**Fig 3.3(a) :** Customer DFD

**3.4 USE CASE DIAGRAMS**

|  |
| --- |
| Login |
| View sweets |
| Manage and add sweets |
| Manage customers |
| View orders |
| Confirm orders |
| View feedback |
| Logout |

**ADMIN**

|  |
| --- |
| Login |
| View sweets |
| Place order |
| Pay for order |
| View bill |
| View order history |
| Give feedback |
| Logout |

**Fig 3.4(a) :** Use Case Diagram of Admin Module

**CUSTOMER**

**Fig 3.4(b) :** Use Case Diagram of Customer Module

**3.5 SEQUENCE DIAGRAMS**

System

Database

Admin

Customer

Customer request to view sweets

View sweets

return

display sweets list

place order request

view request

confirm order

return order confirmation

display order confirmation

**Fig 3.5 :** Sequence Diagram

**CHAPTER 4**

**IMPLEMENTATION**

**4.1 DESCRIPTION OF TABLES**

**SWEET**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FIELD** | **TYPE** | **NULL** | **KEY** | **DEFAULTS** | **EXTRA** |
| sid | int(11) | NO | PRI | NULL | auto\_increment |
| sname | varchar(20) | YES |  | NULL |  |
| stype | varchar(20) | YES |  | NULL |  |
| quantity | int(11) | YES |  | NULL |  |
| sprice | int(11) | YES |  | NULL |  |

**Fig 4.1(a) :** SWEET

**CUSTOMER**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FIELD** | **TYPE** | **NULL** | **KEY** | **DEFAULTS** | **EXTRA** |
| cid | int(11) | NO | PRI | NULL |  |
| cname | varchar(20) | YES |  | NULL |  |
| phone | bigint(20) | YES |  | NULL |  |
| address | varchar(20) | YES |  | NULL |  |
| bid | int(11) | YES | MUL | NULL |  |

**Fig 4.1(b) :** CUSTOMER

**ORDERS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FIELD** | **TYPE** | **NULL** | **KEY** | **DEFAULTS** | **EXTRA** |
| oid | int(11) | NO | PRI | NULL | auto\_increment |
| odate | timestamp | YES |  | current\_timestamp() |  |
| cid | int(11) | YES | MUL | NULL |  |
| bid | int(11) | YES | MUL | NULL |  |
| sid | int(11) | YES | MUL | NULL |  |
| quantity | int(11) | YES |  | NULL |  |

**Fig 4.1(c) :** ORDERS

**BRANCH**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FIELD** | **TYPE** | **NULL** | **KEY** | **DEFAULTS** | **EXTRA** |
| bid | int(11) | NO | PRI | NULL |  |
| blocation | varchar(20) | YES |  | NULL |  |

**Fig 4.1(d) :** BRANCH

**PAYMENT**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FIELD** | **TYPE** | **NULL** | **KEY** | **DEFAULTS** | **EXTRA** |
| payno | int(11) | NO | PRI | NULL | auto\_increment |
| ptype | varchar(20) | YES |  | NULL |  |
| pamt | int(11) | YES |  | NULL |  |
| oid | int(11) | YES | MUL | NULL |  |

**Fig 4.1(e) :** PAYMENT

**DELIVERY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FIELD** | **TYPE** | **NULL** | **KEY** | **DEFAULTS** | **EXTRA** |
| dno | int(11) | NO | PRI | NULL | auto\_increment |
| dplace | varchar(20) | YES |  | NULL |  |
| deldate | date | YES |  | NULL |  |
| payno | int(11) | YES | MUL | NULL |  |
| cid | int(11) | YES | MUL | NULL |  |
| status | varchar(20) | YES |  | NULL |  |

**Fig 4.1(f) :** DELIVERY

**FEEDBACK**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FIELD** | **TYPE** | **NULL** | **KEY** | **DEFAULTS** | **EXTRA** |
| cid | int(11) | NO | PRI | NULL |  |
| oid | int(11) | NO | PRI | NULL |  |
| rating | int(11) | YES |  | NULL |  |
| description | varchar(20) | YES |  | NULL |  |

**Fig 4.1(g) :** FEEDBACK

**4.2 CONSTRAINTS ON TABLES**

**SWEET**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PRIMARY KEY | FOREIGN KEY | NOT NULL | AUTO\_INCREMENT | TIME\_TIMESTAMP |
| sid |  | sid | sid |  |

**Fig 4.2(a) :** SWEET

**CUSTOMER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PRIMARY KEY | FOREIGN KEY | NOT NULL | AUTO\_INCREMENT | TIME\_TIMESTAMP |
| cid | bid | cid |  |  |

**Fig 4.2(b) :** CUSTOMER

**BRANCH**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PRIMARY KEY | FOREIGN KEY | NOT NULL | AUTO\_INCREMENT | TIME\_TIMESTAMP |
| bid |  | bid | bid |  |

**Fig 4.2(d) :** BRANCH

**ORDERS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PRIMARY KEY | FOREIGN KEY | NOT NULL | AUTO\_INCREMENT | TIME\_TIMESTAMP |
| oid | cid,bid,sid | oid | oid | odate |

**Fig 4.2(c) :** ORDERS

**PAYMENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PRIMARY KEY | FOREIGN KEY | NOT NULL | AUTO\_INCREMENT | TIME\_TIMESTAMP |
| payno | oid | payno | payno |  |

**Fig 4.2(e) :** PAYMENT

**DELIVERY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PRIMARY KEY | FOREIGN KEY | NOT NULL | AUTO\_INCREMENT | TIME\_TIMESTAMP |
| dno | payno,cid | dno | dno |  |

**Fig 4.2(f) :** DELIVERY

**FEEDBACK**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PRIMARY KEY | FOREIGN KEY | NOT NULL | AUTO\_INCREMENT | TIME\_TIMESTAMP |
| cid,oid | cid,oid | cid,oid |  |  |

**Fig 4.2(g) :** FEEDBACK

**4.3 BACK END IMPLEMENTATION**

**CUSTOMER LOGIN:**

<?php

$connect=mysqli\_connect("localhost","root","","saccharine");

echo "welcome";

if(isset($\_POST['cid']))

{

$cid=$\_POST['cid'];

$cname=$\_POST['cname'];

$sql="SELECT \* FROM customer where cid='".$cid."' and cname='".$cname."' limit 1";

$records=mysqli\_query($connect,$sql);

if(mysqli\_num\_rows($records)==1){

echo '<script language="javascript">alert("you have successfully logged in");

location="cust\_s.php";

</script>'; }

else{

echo '<script language="javascript">alert("invalid user id or name");

location="cust.html";

</script>'; }

}

?>

**ADMIN LOGIN:**

if(isset($\_POST['admin'])){

$admin=$\_POST['admin'];

$password=$\_POST['password'];

$sql="SELECT \* FROM admin where admin='".$admin."' and password='".$password."' limit 1";

$records=mysqli\_query($connect,$sql);

if(mysqli\_num\_rows($records)==1)

{

echo '<script language="javascript">alert("you have successfully logged in");

location="admin\_s.php";

</script>'; }

**INSERT INTO SWEET TABLE:**

if(!empty($\_POST)) {

$sid=$\_POST['sid'];

$sname=$\_POST['sname'];

$stype=$\_POST['stype'];

$quantity=$\_POST['quantity'];

$sprice=$\_POST['sprice'];

$sql="INSERT INTO sweets(sid,sname,stype,quantity,sprice) VALUES('$sid','$sname','$stype','$quantity','$sprice')";

if(mysqli\_query($con,$sql)) {

echo " inserted successfully";

exit ();

}

**INSERT INTO ORDERS TABLE:**

if(!empty($\_POST)) {

$cid=$\_POST['cid'];

$bid=$\_POST['bid'];

$sid=$\_POST['sid'];

$quantity=$\_POST['quantity'];

$sql="INSERT INTO orders(cid,bid,sid,quantity) VALUES('$cid','$bid','$sid','$quantity')";

if(mysqli\_query($con,$sql)) {

echo '<script language="javascript">alert("continue with payment");</script>';

header("location:pay.php");

}

**INSERTING INTO PAYMENT TABLE:**

if(!empty($\_POST)) {

$ptype=$\_POST['ptype'];

$sql="INSERT INTO payment(ptype) VALUES('$ptype')";

$records = mysqli\_query($con,$sql);

if($records) {

echo '<script language="javascript">alert("order placed sucessfully");

location="extra.php";

</script>';}

**INSERT INTO CUSTOMER TABLE:**

if(!empty($\_POST)) {

$cid=$\_POST['cid'];

$cname=$\_POST['cname'];

$phone=$\_POST['phone'];

$address=$\_POST['address'];

$bid=$\_POST['bid'];

$sql="INSERT INTO customer(cid,cname,phone,address,bid) VALUES('$cid','$cname','$phone','$address','$bid')";

if(mysqli\_query($con,$sql)) {

echo '<script language="javascript">alert("sign up sucessful");

location="cust.html";

</script>'; }

**FEEDBACK TABLE:**

<?php

$result=mysqli\_query($connect,"SELECT f.\*,cname,sname FROM feedback f,customer c,orders o,sweet s where f.cid=c.cid and o.oid=f.oid and o.sid=s.sid");

?>

<table margin="20px" width="600" border="1" cellpadding="1" cellspacing="1">

<tr>

<th>Customer id</th>

<th>Customer name</th>

<th>Order id</th>

<th>Sweet</th>

<th>Rating</th>

<th>Description</th>

<?php

while($row=mysqli\_fetch\_array($result,MYSQLI\_ASSOC))

{

$cid=$row['cid'];

$cname=$row['cname'];

$oid=$row['oid'];

$sname=$row['sname'];

$rating=$row['rating'];

$description=$row['description'];

?>

<tr>

<td><?php echo $row["cid"];?></td>

<td><?php echo $row["cname"];?></td>

<td><?php echo $row["oid"];?></td>

<td><?php echo $row["sname"];?></td>

<td><?php echo $row["rating"];?></td>

<td><?php echo $row["description"];?></td>

</tr>

<?php

**4.4 FRONT END IMPLEMENTATION**

**HOME PAGE:**

<!DOCTYPE html>

<head>

<title>sweet</title>

<link href='style.css' type='text/css' rel='stylesheet'>

</head>

<body>

<div class="bgimage">

<div class="menu">

<div class="lmenu">

<h1> Saccharine Management System </h1>

</div>

<div class="rmenu">

<ul>

<li id="firstlist"> HOME </li>

<li><a href="admin.html"> ADMIN LOGIN </a></li>

<li><a href="cust.html"> CUSTOMER LOGIN </a> </li>

<li><a href="b.php"> BRANCH</a> </li>

<li> <a href="contact.html">CONTACT</a> </li>

</ul>

</div>

</div>

<div class="text">

<h4> Sweet.with.smile </h4>

<h1> ABOUT </h1>

<h3> A sweetshop is a place where it manufactures variety of sweets and sells it. Saccharine management system is designed for easy, effective, time saving management of employee and customer.</h3>

</div></div> </body>

</html>

**ADMIN LOGIN:**

<!DOCTYPE html>

<html>

<head>

<title> Saccharine Management System </title>

<link href='style.css' type='text/css' rel='stylesheet'>

</head>

<body>

<div class="img"></div>

<form action="v2.php" method="post">

<div class="container">

<div class="log">

<span class="login">Login to continue</span> </div>

<div><input type="text" placeholder="Enter Username" name="admin" required> </div> <div>

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

</div><div> <button type="submit">Login</button> </div>

</div></form>

</body>

</html>

**CUSTOMER LOGIN:**

<!DOCTYPE html>

<html>

<head>

<title>Saccharine Management System</title>

<link href='style.css' type='text/css' rel='stylesheet'>

</head>

<body>

<div class="img1">

</div>

<form action="v3.php" method="post">

<div class="container">

<div class="log"> <span class="login">Login to continue</span> </div>

<div><input type="text" placeholder="Enter unique id" name="cid" required>

</div>

<div> <input type="password" placeholder="Enter username" name="cname" required> </div> <div> <button type="submit">Login</button> </div>

<div class="addd" >

<h4> New customer? </h4><a href="signup.html"><h3>Sign up</h3></a></div>

</div></form>

</body>

</html>

**CUSTOMER SIGNUP:**

<!DOCTYPE html>

<html>

<head>

<title> Saccharine Management System </title>

<link href='style.css' type='text/css' rel='stylesheet'>

</head>

<body>

<div class="back">

<div class="img2"></div>

<form action="cus.php" method="post">

<div class="container">

<div class="log">

<span class="login">Sign up</span> </div>

<div class="box">

<div><input type="text" placeholder="Enter unique id" name="cid" required></div>

<div><input type="text" placeholder="Enter Name" name="cname" required></div>

<div><input type="text" placeholder="Enter Phone no" name="phone" required></div>

<div><input type="text" placeholder="Enter Address" name="address" required></div>

<div><input type="text" placeholder="Enter Branch id" name="bid" required></div>

</div><div> <button type="submit">Sign up</button> </div>

<div> <h3>Already have an account?</h3><h2><a href='cust.html'>Login</a></h2>

</div></div></form></div>

</body>

**SWEET TABLE:**

<!DOCTYPE html>

<head>

<title>sweet</title>

<link href='style.css' type='text/css' rel='stylesheet'>

</head>

<body>

<div class="bg">

<div class="amenu"><div class="lmenu">

<h1> Saccharine Management System </h1></div>

<div class="rmenu">

<ul>

<li ><a href="indica.html"> HOME</a> </li>

<li><a href="b.php"> BRANCH</a> </li>

<li> <a href="contact.html">CONTACT</a> </li>

<li><a href="indica.html"> LOGOUT </a></li>

</ul></div></div>

<div class="vmenu"><div class="limenu">

<ul><li id="firstlist"><a href="admin\_s.php">Sweets </a> </li>

<li><a href="delete.php"> Customers</a> </li>

<li><a href="order.php"> Orders</a> </li>

<li><a href="delivery.php"> Delivery </a> </li>

</ul></div></div>

<div class="sweet">

<div class="add"><form action="add.php" method="post">

<div class="addd">

<ul>

<li ><a href="admin\_s.php">Add sweet</a> </li>

<li><a href="viewsweet.php"> View sweet</a> </li> </ul></div>

<div><input type="text" placeholder="Sweet name" name="sname" required> </div>

<div><input type="text" placeholder="Sweet type" name="stype" required> </div>

<div><input type="text" placeholder="Quantity" name="quantity" required></div>

<div> <input type="text" placeholder="price per kg" name="sprice" required> </div>

<div class="but"> <button type="submit">Add</button>

<button type="reset" value="reset">Reset </button></div>

</form>

</div></div>

</div></body></html>

**ORDERS TABLE:**

<!DOCTYPE html>

<head>

<title>sweet</title>

<link href='style.css' type='text/css' rel='stylesheet'>

</head>

<body>

<div class="bg"><div class="amenu"><div class="lmenu">

<h1> Saccharine Management System </h1></div>

<div class="rmenu">

<ul>

<li ><a href="indica.html"> HOME</a> </li>

<li><a href="b.php"> BRANCH</a> </li>

<li> <a href="contact.html">CONTACT</a> </li>

<li><a href="indica.html"> LOGOUT </a></li></ul>

</div></div>

<div class="vmenu"><div class="limenu">

<ul><li id="firstlist"><a href="cust\_s.php">Sweets </a> </li>

<li><a href="addorder.php"> Orders</a> </li>

<li><a href="deliveryv.php"> Delivery </a> </li>

<li><a href="myorder.php"> My orders </a> </li>

<li><a href="feedback.php"> Feedback</a> </li>

</ul></div></div>

<div class="sweet">

<div class="add"><form action="" method="post">

<div class="addd">

<ul><li ><h3 style="color: white; line-height:100px; margin-left:-50px; font-family: comic sans ms;">Enter details to place your order</h3> </li> </ul></div>

<div><input type="text" placeholder="customer id" name="cid" required> </div> <div>

<input type="text" placeholder="branch id" name="bid" required> </div>

<div><input type="text" placeholder="Sweet id" name="sid" required> </div>

<div><input type="text" placeholder="Quantity" name="quantity" required></div>

<div class="but"> <button type="submit" style=" margin-left: 150px; ">Add</button>

<button type="reset" value="reset">Reset </button>

</div>

</form>

</div>

</div></div>

</body>

</html>

**4.5 SQL QUERIES**

**Insert queries:**

INSERT INTO sweet (sname,stype,quantity,sprice) VALUES ('$sname','$stype','$quantity','$sprice');

INSERT INTO customer (cid,cname,phone,address,bid) VALUES ('$cid','$cname','$phone','$address','$bid');

INSERT INTO orders(cid,bid,sid,quantity) VALUES('$cid','$bid','$sid','$quantity');

INSERT INTO payment(ptype) VALUES('$ptype');

INSERT INTO delivery(deldate,payno,status) VALUES('$deldate','$payno','waiting');

INSERT INTO feedback(cid,oid,rating,description) VALUES ('$cid','$oid','$rating','$description');

**Display queries:**

SELECT \* FROM sweet;

SELECT \* FROM branch;

SELECT \* FROM customer;

SELECT \* FROM orders;

SELECT \* FROM payment;

select d.\*,cname,phone from customer c,delivery d where c.cid=d.cid;

**Update queries:**

update delivery set status='$status' where dno='$update\_dno';

**Delete queries:**

DELETE FROM customer where cid='$delete\_cid';

**4.6 TRIGGERS AND STORED PROCEDURE**

**TRIGGERS:**

1. To fetch delivery place of customer for particular order from customer address of customer table.

CREATE TRIGGER `deliver` BEFORE INSERT ON `delivery`

FOR EACH ROW

BEGIN

set NEW.dplace = (select address from customer c,orders o,payment p where

p.oid=o.oid and o.cid=c.cid and payno = NEW.payno);

set NEW.cid = (select o.cid from orders o,payment p where p.oid=o.oid and

payno=NEW.payno);

END

1. To calculate total price by fetching price per quantity from sweet table and store in payment table of respective order.

CREATE TRIGGER `payt` BEFORE INSERT ON `payment`

FOR EACH ROW

BEGIN

set NEW.oid = (select oid from orders o order by oid desc limit 1);

set NEW.pamt = (select (sprice\*o.quantity) as pay from orders o,sweet s where

s.sid=o.sid order by oid desc limit 1);

END

**STORED PROCEDURES:**

To discount payment amount.

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `discount`(IN `pno` INT, IN `per` FLOAT)

NO SQL

UPDATE payment SET pamt= (pamt/per) WHERE payno=pno$$

DELIMITER ;

**4.5 PSEDO CODES**

**ADMIN MODULE:**

**Step 1:** Login to account.

**Step 2:** Add new sweets, View available sweet.

**Step 3:** View customers.

**Step 4:** Check for new orders.

**Step 5:** Initiate delivery for paid orders.

**Step 6:** Update delivery status.

**Step 7:** View feedback.

**CUSTOMER MODULE:**

**Step 1:** Login to account.

**Step 2:** View available sweet.

**Step 3:** Place order for required sweet by providing customer id, sweet id, quantity and

Branch id.

**Step 4:** Select the payment mode and complete order process.

**Step 5:** Apply discounts if available, then finalized bill will be displayed.

**Step 6:** Provide feedback.

**CHAPTER 5**

**TESTING**

**5.1 TEST CASES**

Test case is a set of test inputs. executions and expected results developed for a

particular objective.

An excellent test case satisfies the following criteria:

* Reasonable probability of catching errors
* Does interesting things
* Doesn’t do unnecessary things
* Neither too simple nor too complex
* Allows isolation and identification of errors

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **DESCRIPTION** | **INPUT** | **EXPECTED VALUE** | **ACTUAL VALUE** | **RESULT** |
| **1.** | Verify admin login page | user name and password | Admin login should be successful | Admin login successful | PASS |
| **2.** | Verify admin login page | user name and password | Invalid “username / password” error message should be displayed | Error message displayed | PASS |
| **3.** | Verify admin login page | Any fields left blank | “Please fill out this field” should be displayed | “Please fill out this field” is displayed | PASS |
| **4.** | Verify customer login page | user id and password | Customer login should be successful | Customer login successful | PASS |
| **5.** | Verify customer login page | user id and password | Invalid “username / password” error message should be displayed | Error message displayed | PASS |
| **6.** | Verify customer login page | Any fields left blank | “Please fill out this field” should be displayed | “Please fill out this field” is displayed | PASS |
| **7.** | Verify sweet details | Add sweet details | “Inserted” message should display | “Inserted” message is displayed | PASS |
| **8.** | Verify sweet details | Enter invalid data | “Not inserted” error message | “Not inserted” error message | PASS |
| **9.** | Verify sweet details | When view button is clicked | Sweet details should be displayed | Sweet details is displayed | PASS |
| **10.** | Verify placing order | Enter valid order detail | “order placed” message should display | “order placed” message display | PASS |
| **11.** | Verify placing order | Enter invalid order detail | “order not placed” message should display | “order not placed” message display | PASS |
| **12.** | Verify customer deletion | Delete link clicked | Customer details should be deleted | Customer details is deleted | PASS |
| **13.** | Verify delivery status update | Enter update status | Delivery status should be updated | Delivery status is updated | PASS |
| **14.** | Verify order details | When view link is clicked | Order details should be displayed | Order details is displayed | PASS |
| **15.** | Verify payment details | When view link is clicked | Payment details should be displayed | Payment details is displayed | PASS |
| **16.** | Verify delivery details | When view link is clicked | Delivery details should be displayed | Delivery details is displayed | PASS |
| **17.** | Verify feedback details | When view link is clicked | Feedback details should be displayed | Feedback details is displayed | PASS |
| **18.** | Verify feedback details | Enter invalid data | “Not inserted” error message | “Not inserted” error message | PASS |
| **19.** | Verify feedback details | Enter valid data | “Inserted” message should display | “Inserted” message is displayed | PASS |
| **20.** | Verify branch details | When view button is clicked | Branch details should be displayed | Branch details is displayed | PASS |
| **21.** | Verify contact details | When view button is clicked | Contact details should be displayed | Contact details is displayed | PASS |

**Fig 5.1 :** Test Cases

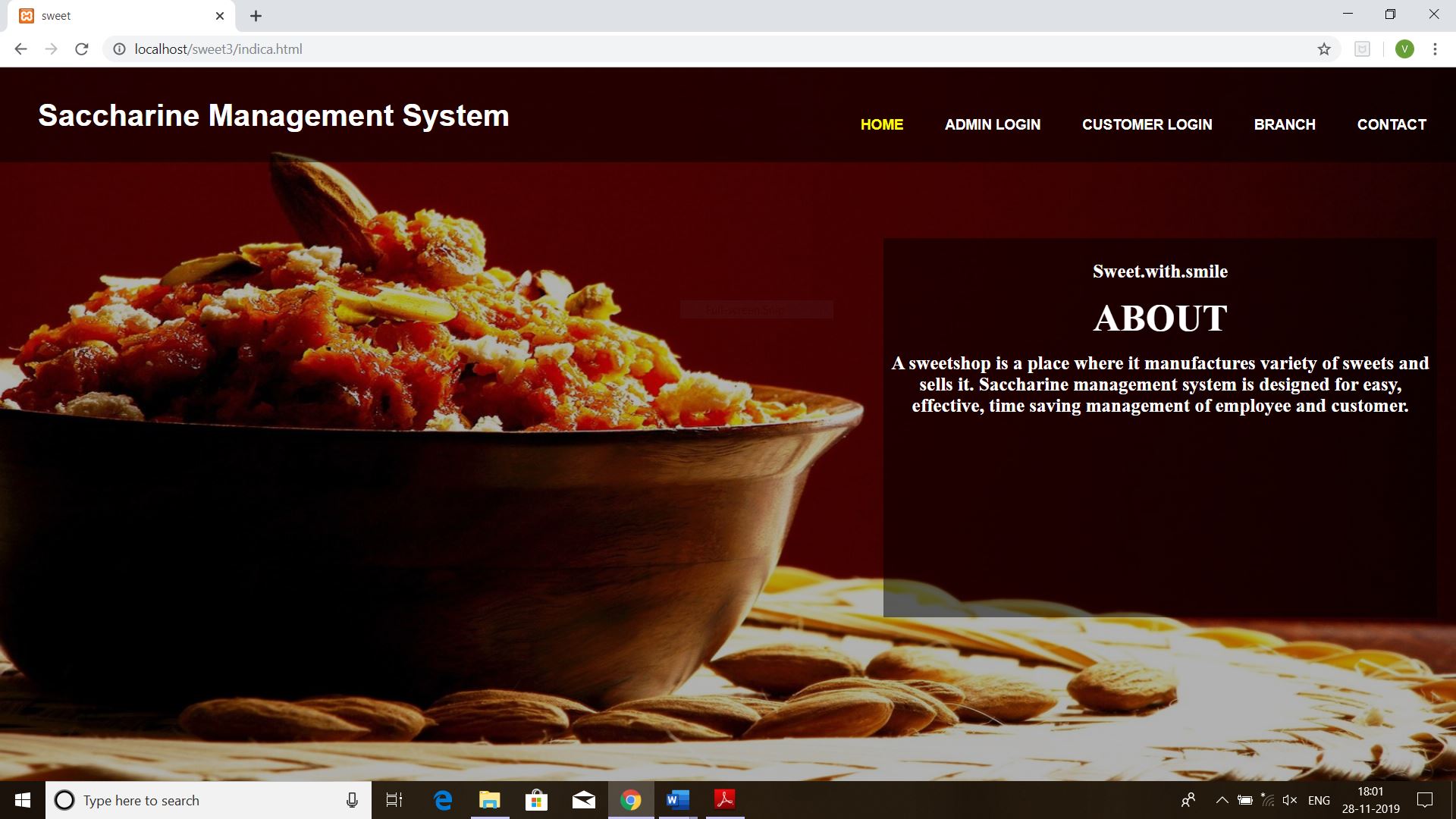
**CONCLUSION AND FUTURE ENHANCEMENT**

The application “Saccharine management system" would be a helpful for owners of sweet shops as well as common people. Those who are using the phone can easily view the items and if they wish to buy they can place an order. The data sent by them will be stored in the corresponding orders. By this application, the customers may be able to buy sweets directly from the owner. Thus profiting the owners better through the online web portal. Using this application , owners may manage the shop loss and prevent the wastage old sweets and can get proper price for their produce. Special features which would popularize the application are Simple Graphical User Interface (GUI) that can be used by everybody. One shop solution to all kinds of issues to the owners. The main objective of this project is to build a website which will help owners to sell their products to different cities and obtain real benefits of online marketing which eliminates the middlemen who snatch away the profit from owners which is satisfied in this project.

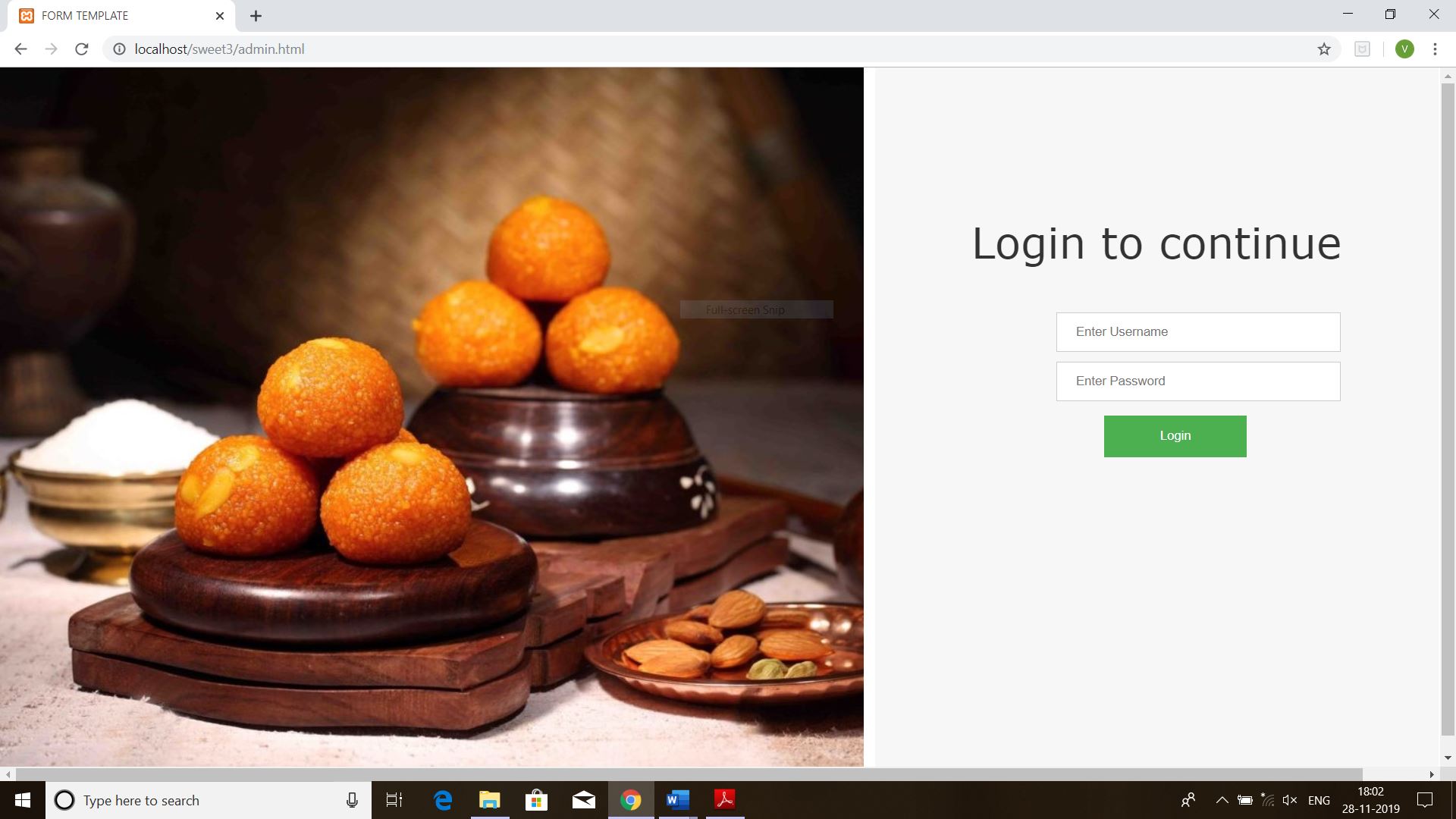
In future enhancement customers will be able to track live delivery updates through GPRS. Customer module will be more customized and more options will be available. View of sweet database will be upgraded and sweet view will be displayed with its images. View of web page will be upgraded and will be more interactive.

**SNAPSHOTS**

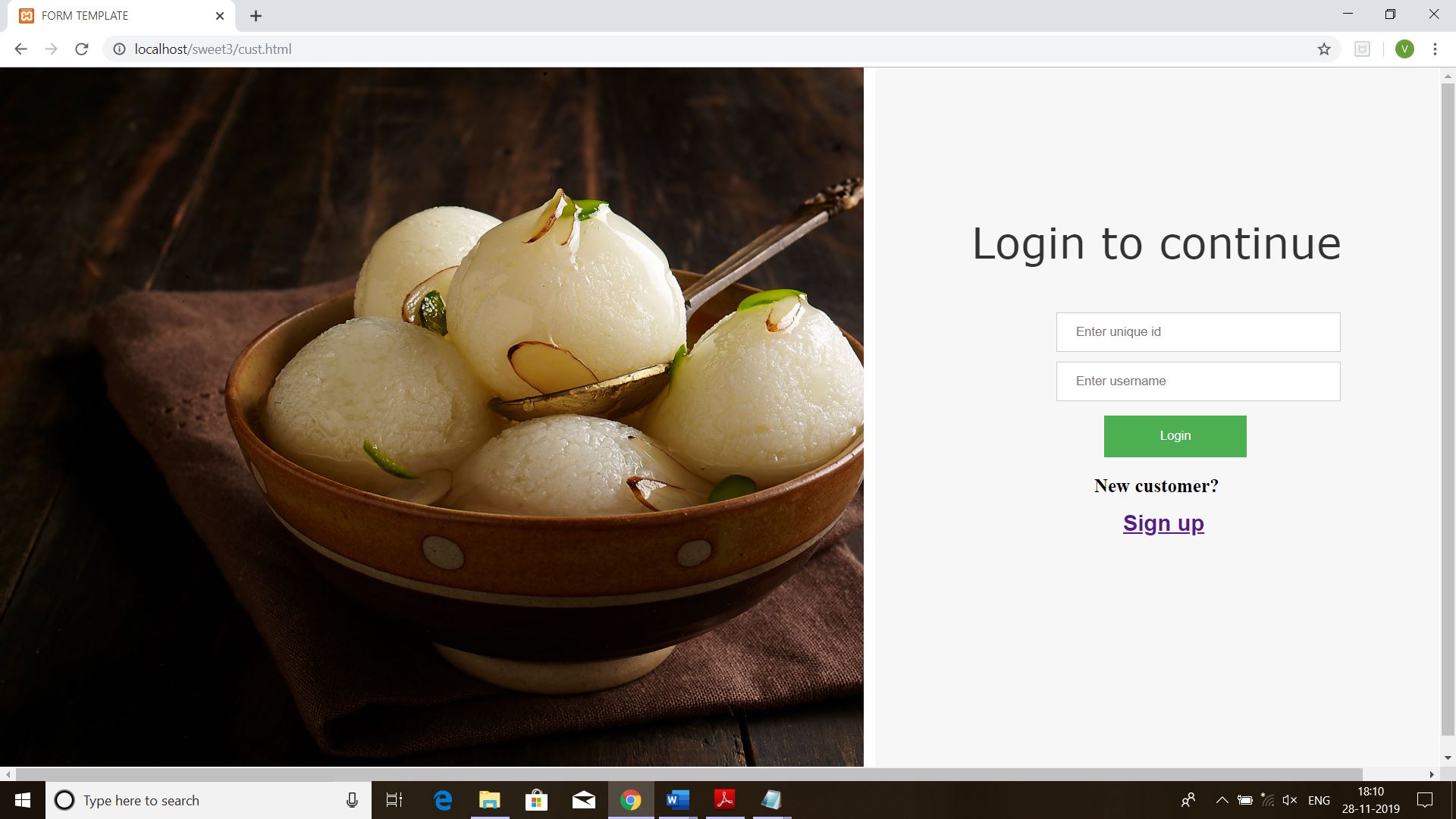
**HOME PAGE**



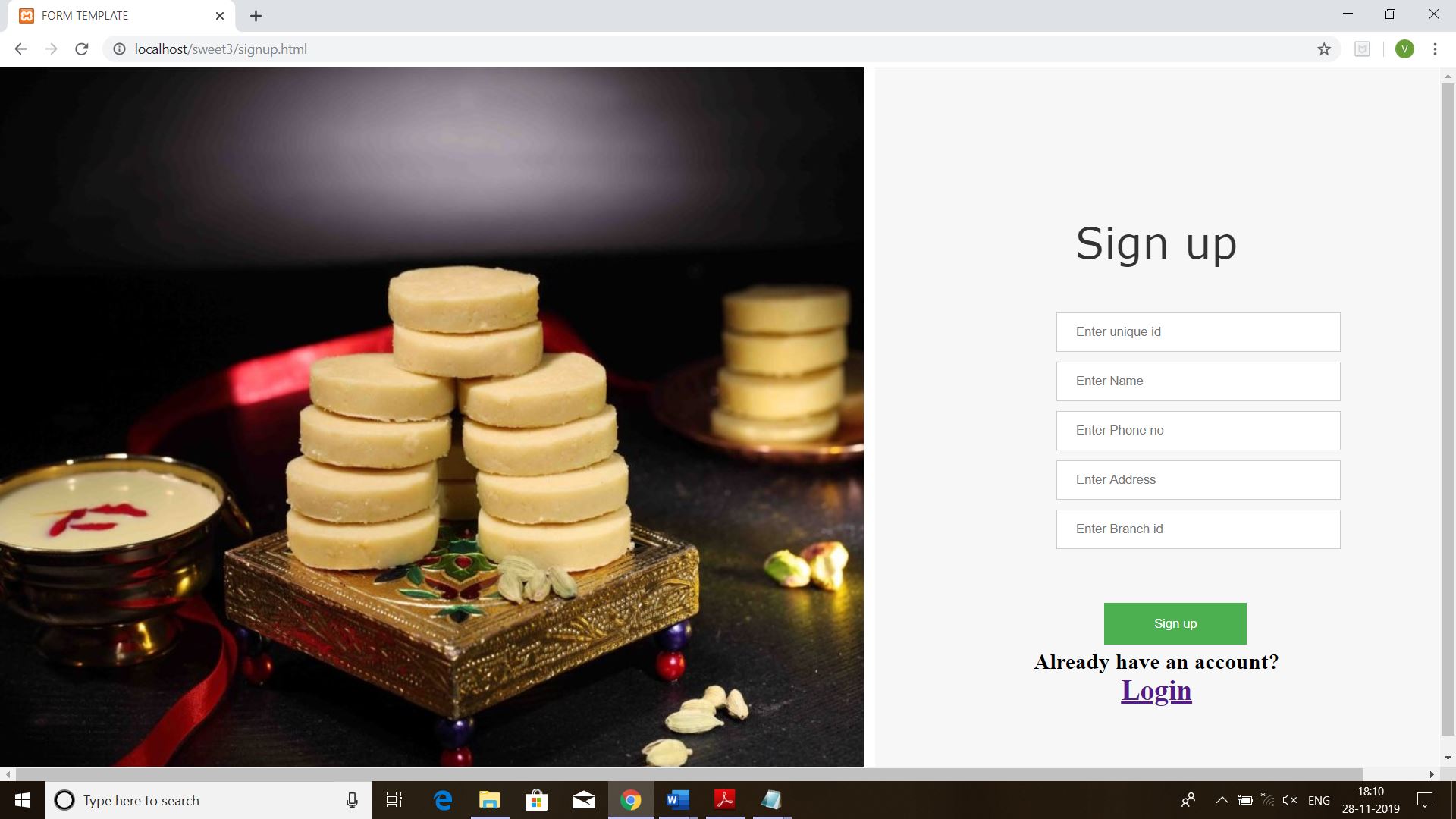
**ADMIN LOGIN**



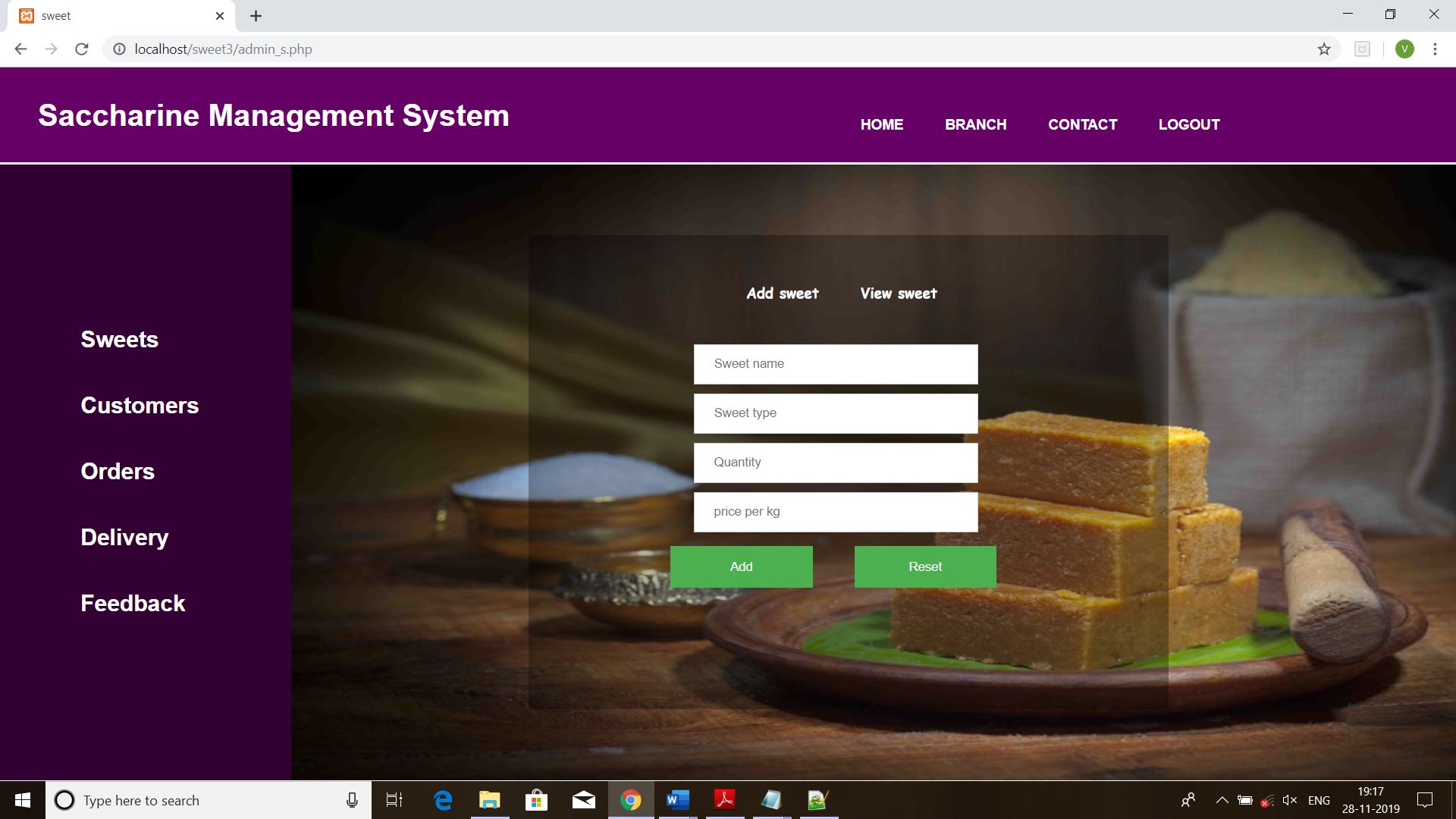
**CUSTOMER LOGIN**



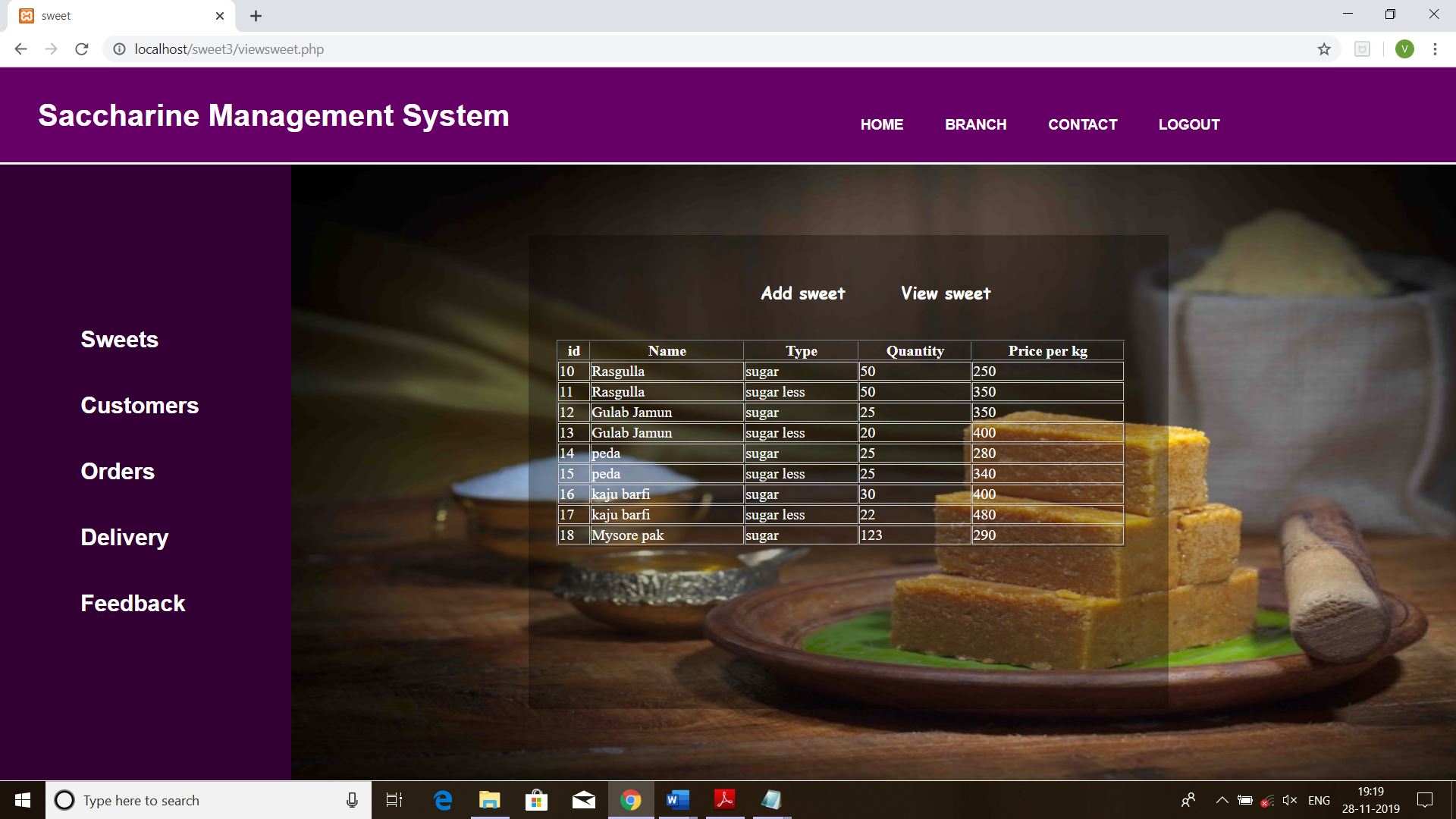
**CUSTOMER SIGNUP**



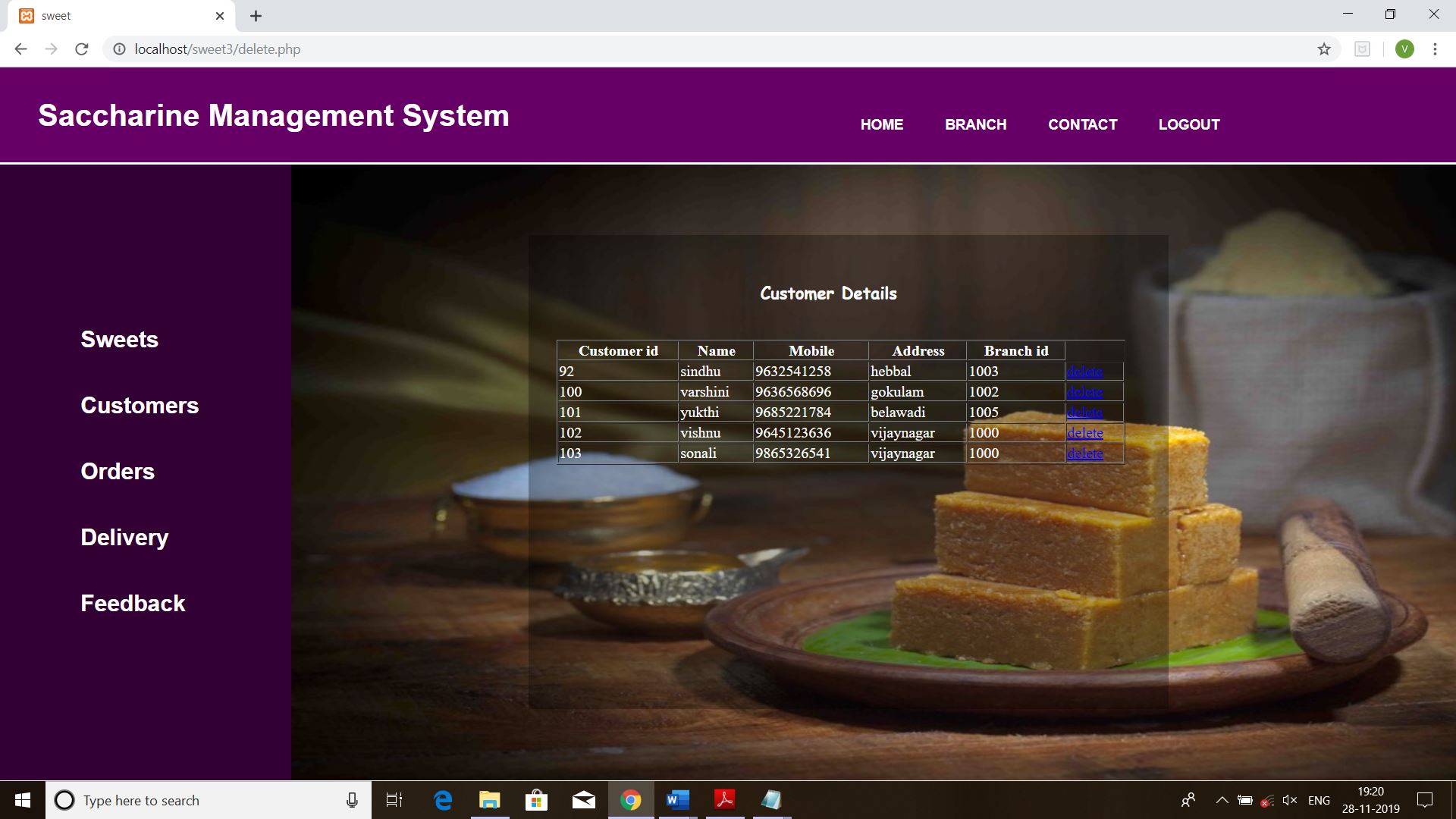
**ADD SWEET**



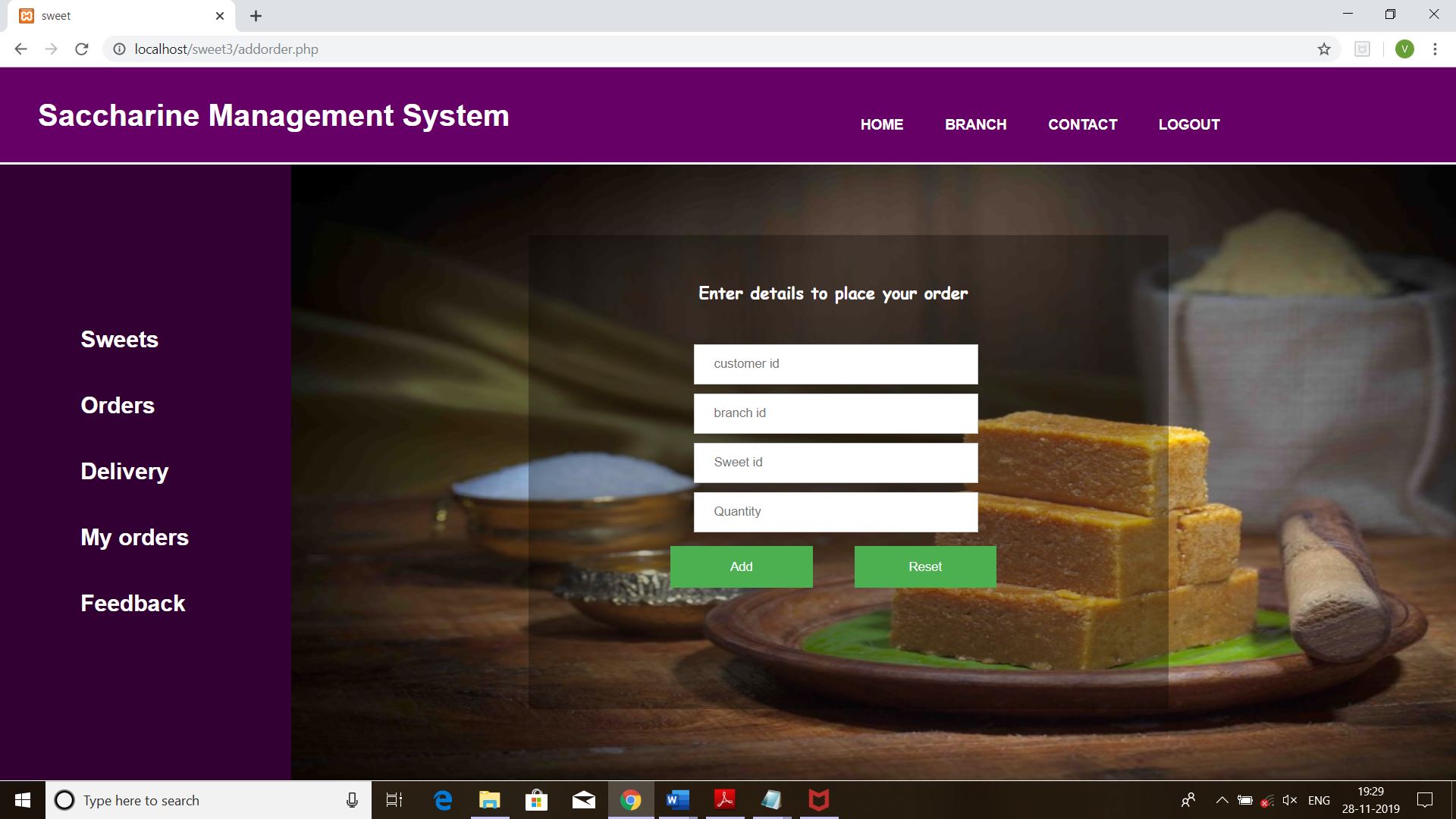
**VIEW SWEET**



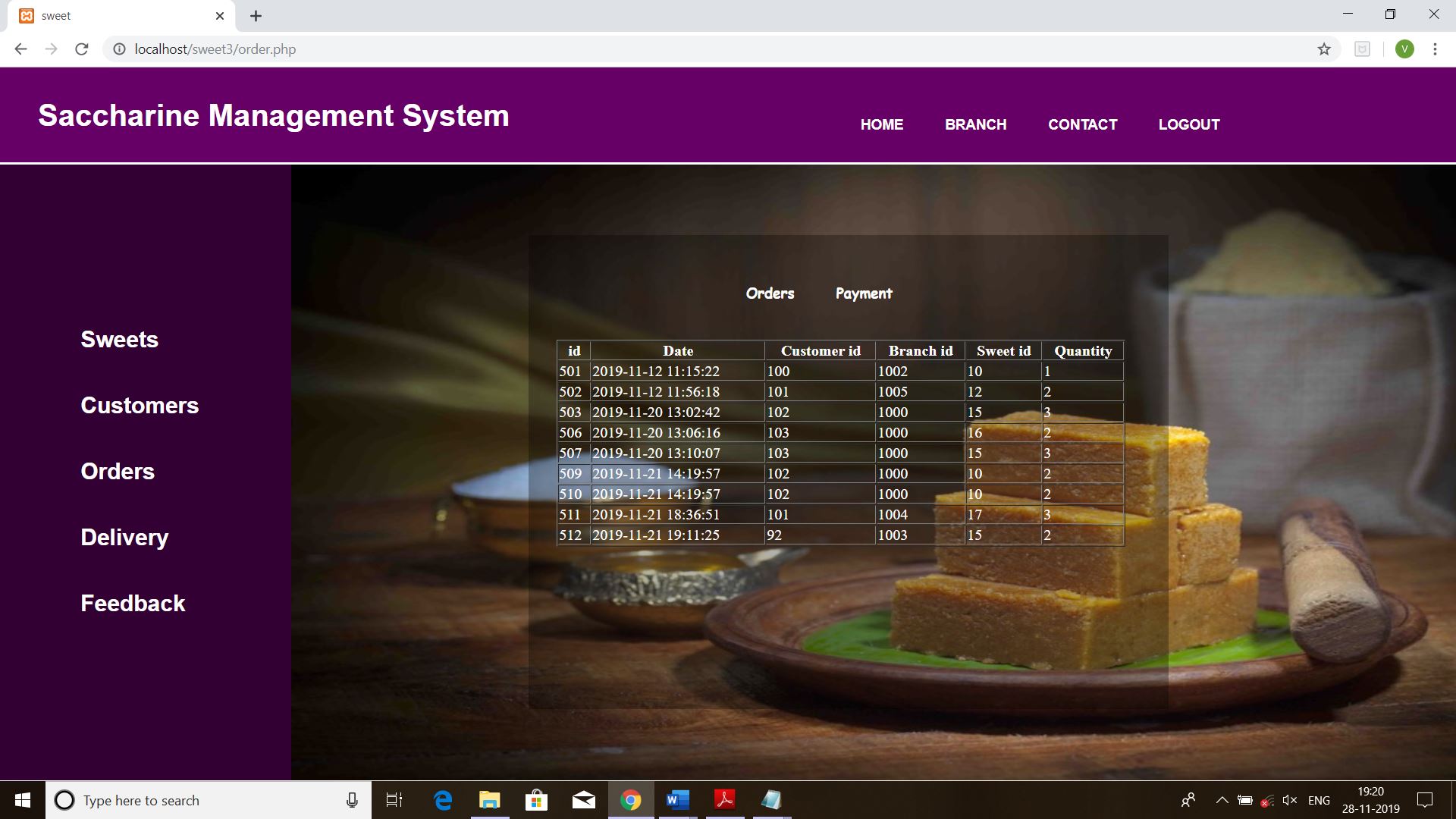
**VIEW CUSTOMERS**



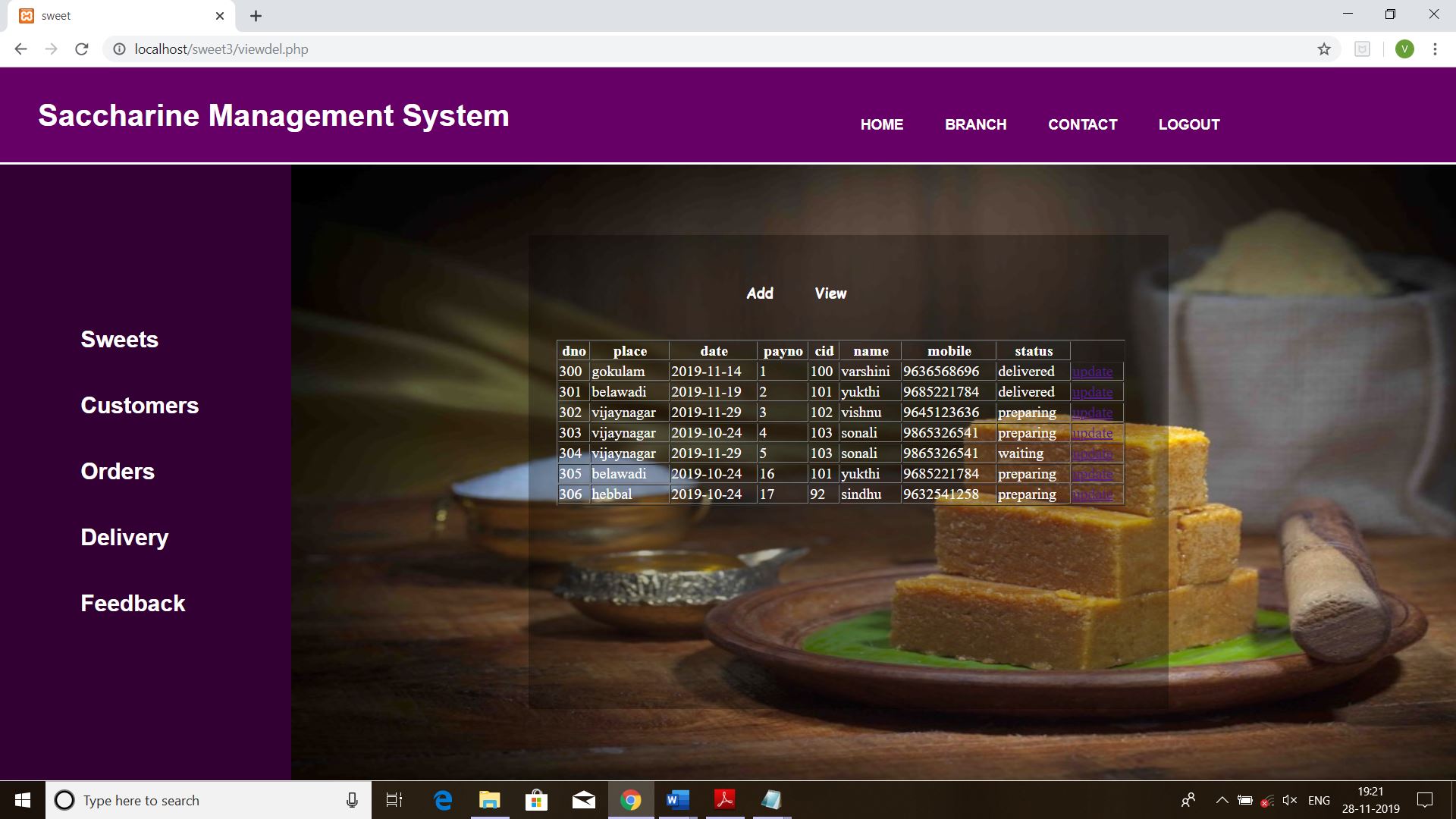
**PLACE ORDER**



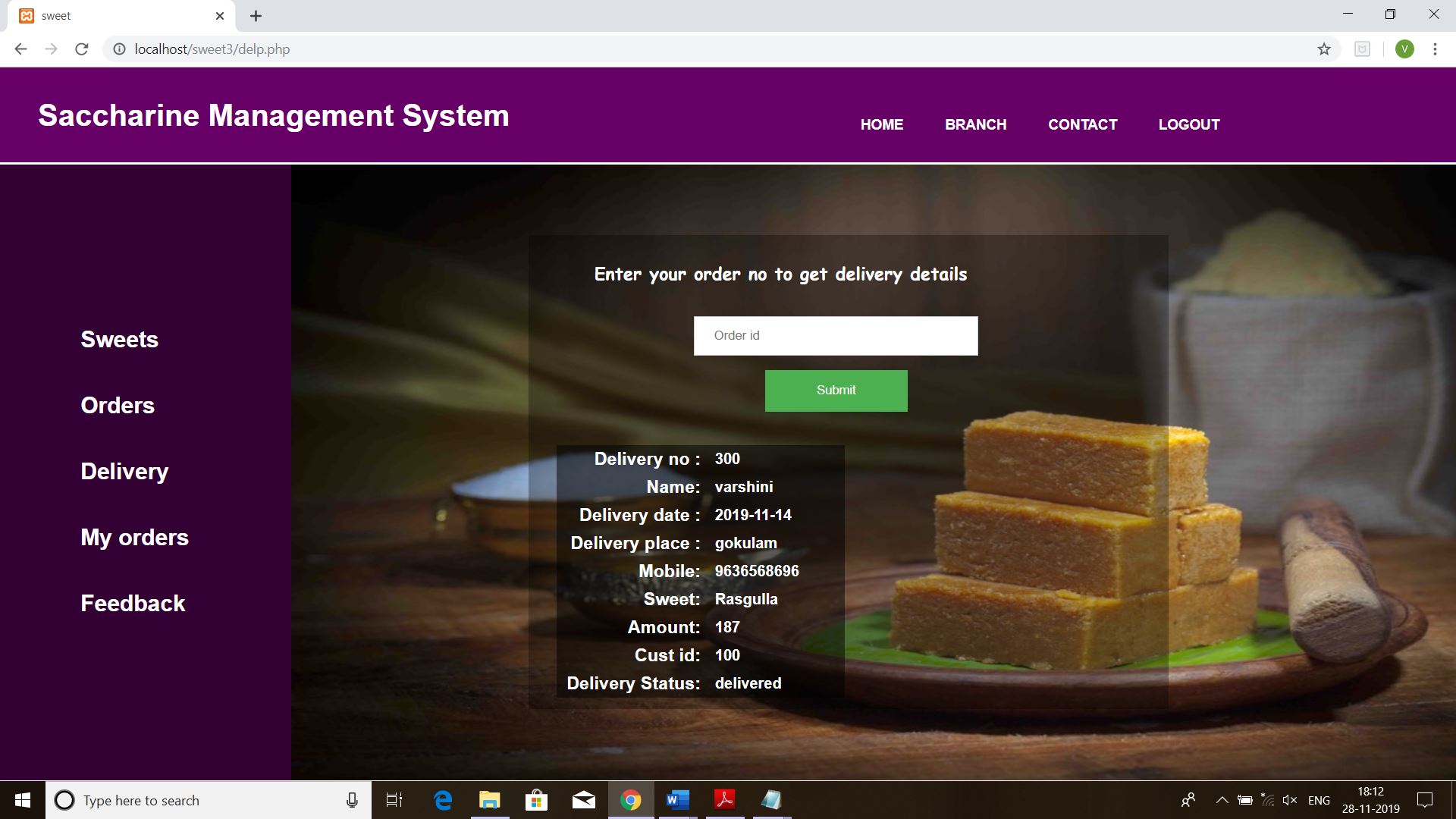
**VIEW ORDERS**



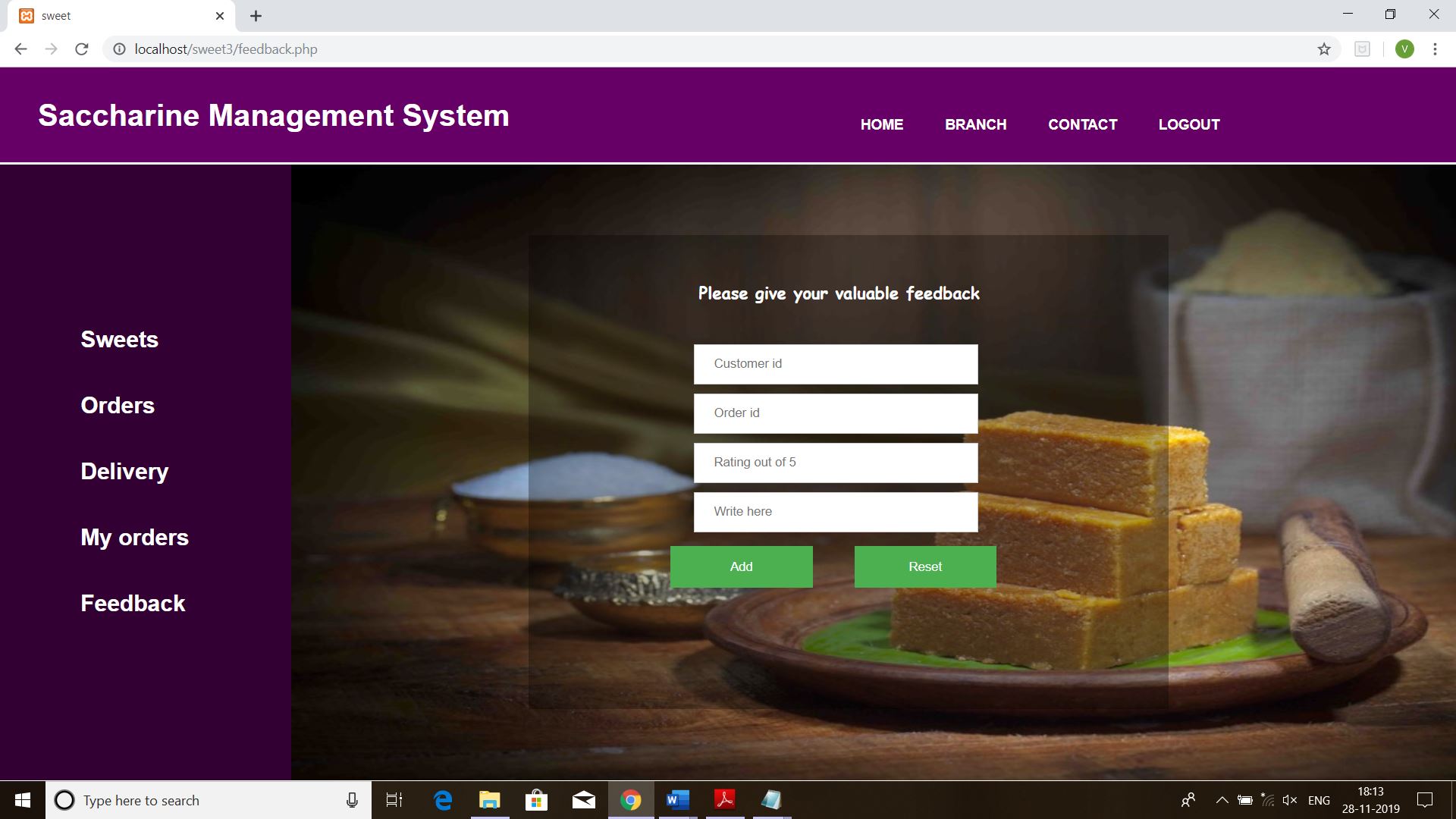
**VIEW DELIVERY**



**VIEW BILLS**



**FEEDBACK**



**REFERENCES**

Websites:

* <https://stackoverflow.com/>
* <https://www.w3schools.com/html/>
* <https://www.w3schools.com/php/>
* <https://github.com/>

YouTube links:

<https://youtu.be/L5RpqspNAuc>

<https://youtu.be/dwVj_g3TpZ4>

<https://youtu.be/31P3JGOKzYE>

<https://youtu.be/uyaV_EWWRmo>

<https://youtu.be/n8lXLWi0oNc>

<https://youtu.be/F0uED-w9E48>