**Conestoga College**

**Reporting Systems and Database Development**

**(1517)**

**Implementing Data Security (PROG8640)**

**Group Project**

**Group-2**

***Submitted By:***

***Basavraj Jaliminche (8800149)***

***Preeja Anilal(8791796)***

***Jino Wiseson Anbu Alaguraj- 8807543***

***kirtvir singh swatch (8800968)***

**AUG 14, 2022**

**Restaurant Management System**

**Overview**

Restaurant Management System is a web application for restaurants to manage their orders for both dining and take-out orders. This application can handle both users and admin. In person dining would need to capture table numbers, seat numbers, reservations and so on. Both would also need to capture order number, Date/time placed, date/time served, cost of food, cost of drinks.

**Objective**

Objective of this section is to just develop a web application which can handle all day-to-day operations of Restaurant for instance booking new orders, items price list, tables booking etc.

This Website has 2 main pages.

* One is for Admin (Only admin can login to this page and do operations regarding to payment and settling orders)
* Second page is for all the users who want to book the tables in advance and includes food items. They can also take out the food as well.
* We will then try to breach the website using SQL injection and discuss the way to cure it.
* Available pages in the application:

Login Page:

Users can login here using their name and phone number.

**Languages used:**

Front end: Html, CSS, JavaScript

Database: MySQL

Backend: Php

**Breaching Technique:**

SQL injection - SQL injection (SQLi) is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database.

**Required Software:**

Wampserver - If you are using Windows Operating System then you can install WMAP server which stands for Windows Apache MySQL and PHP or Perl or python.

**Part I – Web App Development – Customer – End User:**

As we already know that php is a server-side scripting language, so we need server to run php code. There are many servers to run php program. If you are using Windows Operating System then you can install WMAP server which stands for Windows Apache MySQL and PHP or Perl or python. It is a complete package or stack. And if you are using Linux Operating System then you can install Lamp which stands for Linux Apache MySQL and PHP or Perl or python. XAMP is used for MacOS.

Here we are using WAMP server. When you installed WAMP server you will get wamp folder in your C drive. In this folder there is another folder name www. To run php code we must put all php files in this folder. Php files having .php extension. We can use Notepad or Notepad++ to create .php files.

**Login Page**

**Graphical user interface, application

Description automatically generated**

After submitting these data, move to next page which can see Dine in Order, Takeaway Order, Reserve Table, My Order and Logout Tabs.

**Graphical user interface

Description automatically generated**

**Dine-in Order**

For placing the Dine in Order, clicking on the Dine in Order tab, we can find this page. Enter corresponding data to the input field. Then you can see the amount you want to pay and submit it.

**Graphical user interface, table

Description automatically generated**

**Takeaway Order Page**

For place Take Out Order, open corresponding page and enter data into the input field such as food order and drink order.

**Graphical user interface

Description automatically generated**

**Reserve Table Page**

For Reserving your table, click on the tab and enter you wanted table details into the input field and submit the page.

**Graphical user interface, application

Description automatically generated**

**My Order Page**

Finally, we can see the order details with all entered data. It’s available in My order page.

**Graphical user interface, table, website

Description automatically generated**

In here, we can see the payment status, it is in pending status. So, we need to make the payment with help of admin.

**Admin Page:**

The restaurant owner is the admin. He or she can login the page.

Only admin can login with the his credentials.

Graphical user interface, website

Description automatically generated

After login admin can see which are the new orders.He can do by clicking the new order.Admin can mark order as paid.

Graphical user interface

Description automatically generated

Also by clicking on order History ,we can get the all orders completed.

Graphical user interface

Description automatically generated

**Database:**

We have created one database with the 4 tables .Name of the database is ‘restaurentmanagment’.

A screenshot of a computer

Description automatically generated

When we click on Adminlogin – It will show us all the active admins for restaurant manager website.

As we can see as of now we have 2 admins – admin and KirtSwatch. To add new admin user – click in insert and add details

A screenshot of a computer

Description automatically generated

Similarly, we have userlogin details db as well as this will store all the users who entered in our website to book tables or food.

Graphical user interface, application

Description automatically generated

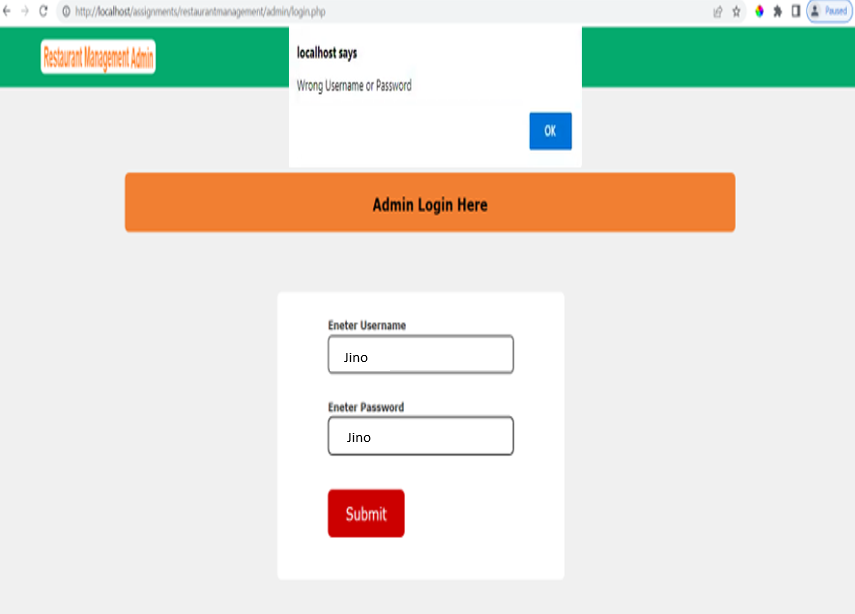
**Part III – Data Breaching Using SQL injection - Admin end:**

SQL injection (SQLi) is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. It generally allows an attacker to view data that they are not normally able to retrieve. This might include data belonging to other users, or any other data that the application itself is able to access. In many cases, an attacker can modify or delete this data, causing persistent changes to the application's content or behaviour.

In our project, we will demonstrate the technique to bypass the authentication of a vulnerable login page [using SQL injection](https://portswigger.net/web-security/sql-injection).

To check for potential SQL injection vulnerabilities, we have entered a single quote in to the "Name" field and submitted the request using the "Login" button. The application provides us with an SQL error message. The error message includes the SQL query used by the login function. We can use this information to construct an injection attack to bypass authentication. The first account in a database is often an administrative user, we can exploit this behaviour to log in as the first user in the database.

This is our admin login page, if I try to enter any other credential it will give wrong username and password error message.



Enter some appropriate syntax to modify the SQL query into the "User Name" input.

In this example we used **' or 1=1 --.**

This causes the application to perform the query:

**SELECT \* FROM users WHERE username = '' OR 1=1-- ' AND password = 'aaaaaa'**

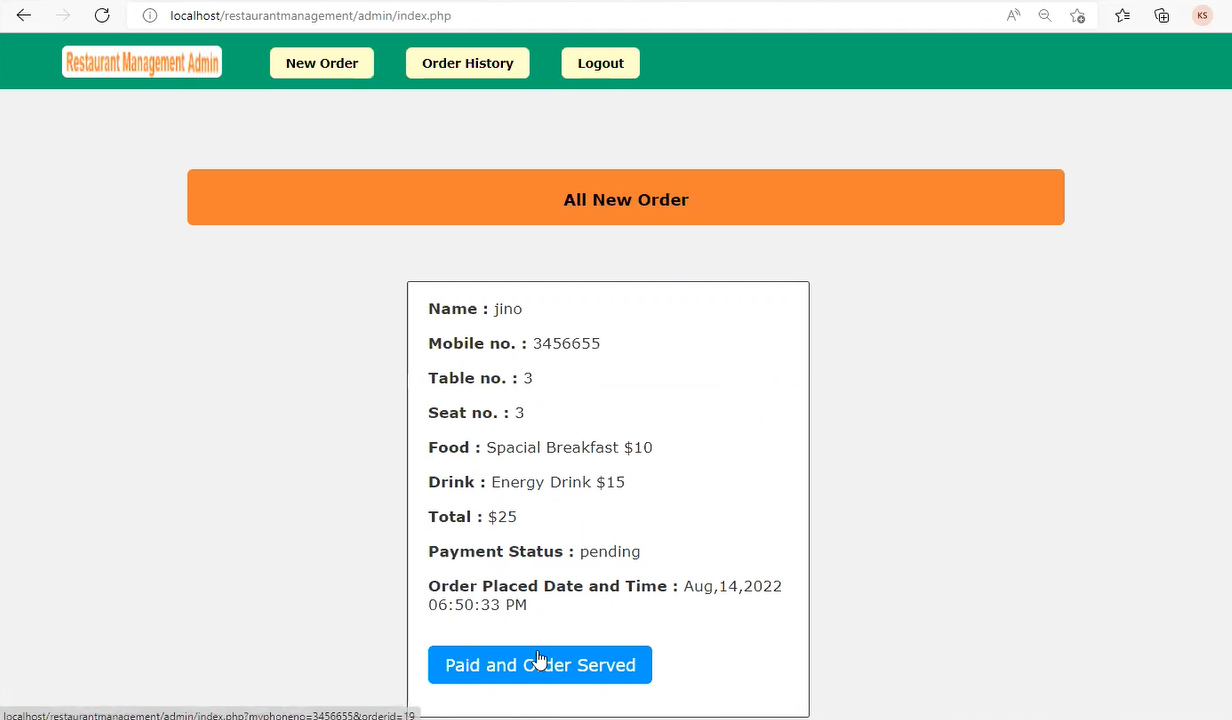
Because the comment sequence (--) causes the remainder of the query to be ignored, this is equivalent to:

SELECT \* FROM users WHERE username = ' ' **OR 1=1**

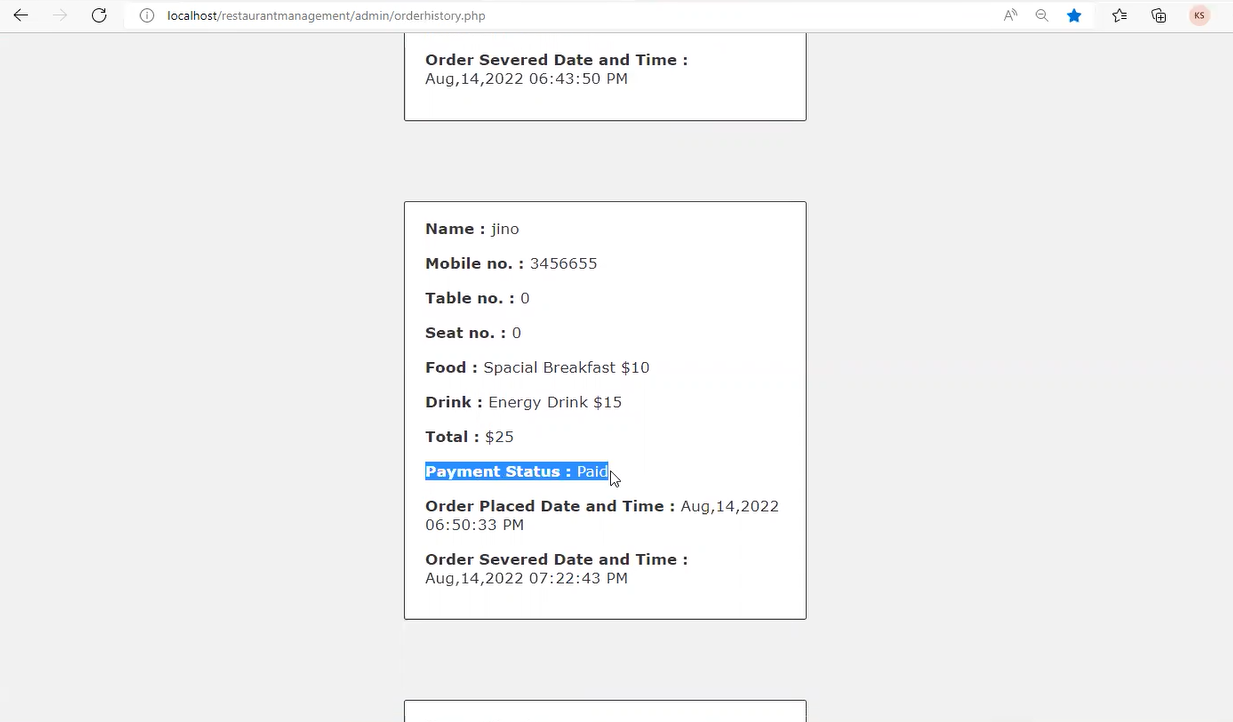
Graphical user interface, website

Description automatically generated

Now I can access the Admin page and I can change any customer payment here for instance I’m going to change the payment status of Jino.



And, here we can see the payment of customer Jino status is paid. Eventually, we changed the customer data with help of SQL Injection.



Backend code Where the password authenticating is being compromised by the week code. If we see the code the username and password is unprotected, they are getting by directly with help of Ajax.

A screenshot of a computer

Description automatically generated with medium confidence

**Part III – Data Securing – Admin end:**

**References**

[**http://www.tizag.com/mysqlTutorial/mysql-php-sql-injection.php**](http://www.tizag.com/mysqlTutorial/mysql-php-sql-injection.php)

[**https://www.w3schools.com/html/**](https://www.w3schools.com/html/)

[**https://www.w3schools.com/css/**](https://www.w3schools.com/css/)

[**https://www.w3schools.com/javascript/**](https://www.w3schools.com/javascript/)

**https://www.spiceworks.com/it-security/application-security/articles/what-is-sql-injection/**