RESTAURANT MANAGEMENT SYSTEM

#### A Project Report

Submitted in partial fulfillment of the Requirements

for the award of the Degree of

### BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

**By**

## YASH GOPI WAGHADHARE

## Seat Number : 4021185

### Under the esteemed guidance of

### MRS. MEDHA SAHASRABUDDHE

**Assistant Teacher at Information Technology**

**Department of Gogate Jogalekar College, Ratnagiri**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

### GOGATE JOGALEKAR COLLEGE, RATNAGIRI

***( Affiliated to University of Mumbai )***

**RATNAGIRI, 415612 MAHARASHTRA**

**2022-23**

# PROJECT PROPOSAL

|  |  |
| --- | --- |
| **P.R.N. Number :** 2020016401098674 | **Roll Number :** TTA-15 |
|  |  |
| 1. **Name of Student :** | Yash Gopi Waghadhare |
| 1. **Title of the Project :** | Online Restaurant Management System |
| 1. **Name of the Guide :** | Mrs. Medha Sahasrabuddhe. |
| **4.Teaching experience :** | 9 years |
| **5. Is this your first submission ?** | Yes | No |
| Signature of the Student | Signature of the Guide |
| Date : | Date : |
| Signature of the Coordinator | |
| Date : | |

**GOGATE JOGALEKAR COLLEGE, RATNAGIRI**

***(Affiliated to University of Mumbai)***

### RATNAGIRI-MAHARASHTRA-415612

### DEPARTMENT OF INFORMATION TECHNOLOGY



|  |  |  |
| --- | --- | --- |
| Internal Guide | External Examiner | HOD / Coordinator |
| Date: |  | College Seal |

**CERTIFICATE**

This is to certify that the project entitled, **“ONLINE RESTAURANT MANAGEMENT SYSTEM”** , is bona-fide work of **“YASH GOPI WAGHADHARE”** bearing **Seat No. :** 4021185 submitted in partial fulfillment of the requirements for the award of the degree of BACHELOR OF SCIENCE in **“INFORMATION TECHNOLOGY”** from **“University of Mumbai”**.

# DECLARATION

I hereby declare that the project entitled, **“RESTAURANT MANAGEMANET SYSTEM”** done at **“Gogate Jogalekar College, Ratnagiri”**, has not been in any case duplicated to submitted to any other university for the award of any degree. To the best of my knowledge, other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **“BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)”** to be submitted as a final semester project as part of our curriculum.

**YASH GOPI WAGHADHARE.**

# ABSTRACT

**Customer satisfaction** is the key to the **success of any business**. **Online restaurant management systems** enhance operational efficiency through the use of an online system. A system can increase customer satisfaction by **leaving a system to each employee**, as well as this system allows **managers and owners** to **easily monitor** the **work** and **progress** of the restaurant.

Currently, some **stand-alone systems** are available, but we want to **improve** them by creating an **online system** which greatly improves **communication between owner and employee**. The goal is to create a system that can work efficiently for our clients and it will result in increasing overall efficiency in managing a restaurant business .

# ACKNOWLEDGEMENT

I would like to express my sincere gratitude to everyone for supporting me throughout my project. First, I wish to express my sincere gratitude to my guide **Mrs. Medha Sahasrabuddhe** madam for her enthusiasm, patience, insightful comments, helpful comments and information, practical advice and unceasing ideas that have helped me tremendously at all times. Her immense knowledge and profound experience has enabled me to complete this project successfully. Without her support and guidance, this project would not have been possible.

I also wish to express my sincere thanks to the **Head of the Department** of Information Technology, **Mrs. Medha Sahasrabuddhe** , **Coordinator Dr. Vivek Bhide** , **Vice Principal Dr. Aparna Kulkarni** and **Principal** of GJC **Dr. Prafulladatta Kulkarni** for their support and the facilities they have made available. I am also grateful to the staff of the department of information technology for their consistent support and assistance.

I also wish to express my sincere thanks to our college for their support and for the facilities they have made available. Thanks all for your encouragement. This project has been a wonderful experience where we have learn and experienced many beneficial things.

**TABLE OF CONTENT :**

|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **Index** | **Page No.** |
| **1.** | **Introduction** | **10** |
|  | 1.1 Background | 10 |
|  | 1.2 Objectives | 10 |
|  | 1.3 Purpose, Scope and Applicability | 10 |
|  | 1.3.1 Purpose | 10 |
|  | 1.3.2 Scope | 10 |
|  | 1.3.3 Applicability | 10 |
| **2.** | **System analysis** | **11** |
|  | 2.1 Existing Systems | 11 |
|  | 2.2 Proposed System | 11 |
|  | 2.3 Requirement analysis | 11 |
|  | 2.3.1 User Requirements | 11 |
|  | 2.3.2 Development Requirements | 11 |
|  | 2.4 Planning And Scheduling – Gantt Chart | 12 |
| **3.** | **System Design** | **14** |
|  | 3.1 Modules | 14 |
|  | 3.2 Database | 15 |
|  | 3.3 Figures | 19 |
|  | 3.4 User Interface Design | 23 |
|  | 3.5 Event Table | 25 |

|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **Index** | **Page No.** |
| **4.** | **Implementation and Testing** | **26** |
|  | 4.1 Code | 26 |
|  | 4.2 Testing approaches | 29 |
|  | 4.3 Test Cases Design | 32 |
| **5.** | **Result and discussion** | **34** |
| **6.** | **Conclusion** | **37** |
| **7.** | **Reference** | **38** |

**LIST OF FIGURES :**

|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **Figures** | **Page No.** |
| 1. | 1. R. Diagram | 19 |
| 2. | Activity Diagram | 19 |
| 3. | Sequence Diagram | 21 |
| 4. | Use-Case Diagram | 22 |
| 5. | Class Diagram | 22 |

**LIST OF TABLES :**

|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **Figures** | **Page No.** |
| 1. | Restaurant\_Category | 15 |
| 2. | Restaurant\_Items | 15 |
| 3. | Restaurant\_order | 16 |
| 4. | Restaurant\_order\_item | 16 |
| 5. | Restaurant\_table | 17 |
| 6. | Restaurant\_tax | 17 |
|  | Restaurant\_user | 18 |

**CHAPTER 1: INTRODUCTION**

“Success in the restaurant business is all about making the right decisions.”

In today's digital age, restaurant owners are constantly seeking innovative ways to streamline their operations and enhance customer experiences. The online restaurant management system is a cutting-edge solution that is revolutionizing the way restaurants are managed.

### BACKGROUND :

The project we are developing because the Online Restaurant Management System helps the restaurant manager to manage the restaurant more effectively and efficiently by computerizing ordering and billing.

* 1. **OBJECTIVES :**

1. To determine the nature of online restaurant.
2. To make work easy and efficient.
3. To keep a record of everything in their hand.
4. To access their own data anytime, anywhere.
5. To monitor their sales.
   1. **PURPOSE, SCOPE AND APPLICABILITY :** 
      1. **PURPOSE :**

The Restaurant Management System helps the restaurant manager to manage the restaurant more effectively and efficiently by computerizing orders and billing.

* + 1. **SCOPE :**

The scope of the restaurant management system includes the management of finances, live table status, billing, management of employees.

* + 1. **APPLICABILITY :**

First and most importantly, it is applicable for the owners of restaurants who are not able to manage their restaurants in detail. It can help restaurant managers to manage the restaurant more effectively and efficiently by computerizing billing and inventory control.

**CHAPTER 2 : SYSTEM ANALYSIS**

"Without requirements or design, programming is the art of adding bugs to an empty text file." - Louis Srygley

**2.1 EXISTING SYSTEM** :

An existing stand-alone restaurant management system is a software application that is designed to run on a single computer without requiring a network or internet connection. It is used to manage various aspects of a restaurant's operations, such as order taking, inventory management, employee scheduling, and billing.

The system is installed on a local computer, and all data is stored locally. The system may have limited functionality and may not be able to connect to online ordering platforms or integrate with other third-party applications. However, it provides a cost-effective solution for small restaurants that do not require extensive features and functionality.

**2.2 PROPOSED SYSTEM :**

Online restaurant management systems provide a comprehensive solution for restaurant owners and managers to manage various aspects of their operations, even when they are not physically present. With the ability to access the system from any internet-connected device, restaurant owners can keep track of their business operations from anywhere in the world.

**2.3 REQUIREMENT ANALYSIS :**

**2.3.1 User Requirements :**

* Internet connected device
* Any browser

**2.3.2 Development Requirements :**

**2.3.2.1 Software Requirements :**

* Visual Studio Code
* XAMPP server

**2.3.2.2 Hardware Requirements :**

* PC / Laptop with 4GB ram + I3 level processor

**2.4 PLANNING AND SCHEDULING – GANTT CHART :**

**2.4.1 : EXPECTED GANTT CHART :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tasks** | **Duration (Date)** | **Start Date** | **Finish Date** |
| General meeting | 3 | 07-07-2022 | 10-07-2022 |
| Requirement Gathering | 7 | 15-07-2022 | 21-07-2022 |
| Documentation | 5 | 01-08-2022 | 05-08-2022 |
| Module | 8 | 09-10-2022 | 16-10-2022 |
| Database | 10 | 11-11-2022 | 20-11-2022 |
| Diagram | 4 | 21-12-2022 | 24-12-2022 |
| Coding | 25 | 25-12-2022 | 18-01-2023 |
| Testing | 5 | 12-02-2023 | 16-02-2023 |
| Final Documentation | 3 | 22-03-2023 | 24-03-2023 |

**2.4.2 : ACTUAL GANTT CHART :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tasks** | **Duration (Date)** | **Start Date** | **Finish Date** |
| General meeting | 3 | 07-08-2022 | 10-08-2022 |
| Requirement Gathering | 7 | 15-08-2022 | 21-08-2022 |
| Documentation | 5 | 05-09-2022 | 09-09-2022 |
| Module | 8 | 09-11-2022 | 16-11-2022 |
| Database | 10 | 11-12-2022 | 20-12-2022 |
| Diagram | 4 | 21-12-2022 | 24-12-2022 |
| Coding | 25 | 25-12-2022 | 18-01-2023 |
| Testing | 5 | 12-02-2023 | 16-02-2023 |
| Final Documentation | 3 | 22-03-2023 | 24-03-2023 |

**CHAPTER 3 : SYSTEM DESIGN**

**“**Good system design is about ensuring it works efficiently, effectively, and adapts to future needs.**”**

#### 3.1 BASIC MODULES :

The system is basically divided into four modules as per the functionality are as follows :

#### MODULE 1 : ADMIN LOGIN

An admin login module is a software component that provides secure access to an administrative dashboard. It typically requires a unique username and password combination for authentication and can be customized to fit specific security requirements.

#### MODULE 2 : ADD PRODUCT CATEGORY

An Add Product category module is a software component that allows administrators to create and manage product categories in an e-commerce platform.

#### MODULE 3 : ADD PRODUCT

An Add Product item module is a software component that allows administrators to add new products to an e-commerce platform. It typically includes features such as product name, product category, pricing, and inventory management.

#### MODULE 4 : ADD TAX

An Add tax module is a software component that allows administrators to configure tax rates for products or services sold through an e-commerce platform. It typically includes features such as tax name, rate and enables the platform to accurately calculate taxes on customer purchases.

#### MODULE 5 : ADD TABLE

An Add table module is a software component that allows them to create and manage table reservations for their customers. It typically includes features such as table number, capacity, date, and time, and enables efficient table management for the restaurant or dining area.

#### MODULE 6 : TAKE ORDER

A take order module is a software component that allows restaurant or hotel staff to take customer orders for food and drinks. It typically includes features such as menu items and order details, and enables efficient order management for the kitchen. This module ensures a smooth and streamlined ordering process for customers and helps staff provide fast and accurate service.

#### MODULE 7 : UPDATE STATUS

An update status enable/disable module is a software component that allows restaurant or food ordering platform administrators to manage the availability of menu items.

**3.2 DATABASE :**

* **RESTAURANT\_CATEGORY :**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Constraint** |
| id | int(11) | primary |
| name | varchar(250) | --- |
| status | enum(‘Enable’,’Disable’) | --- |

* **RESTAURANT\_ITEM:**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Constraint** |
| id | int(11) | primary |
| name | varchar(250) | --- |
| price | decimal(10,2) | --- |
| Category\_id | int(11) | -- |
| status | enum(‘Enable’,’Disable’) | -- |

* **RESTAURANT\_ORDER :**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Constraint** |
| id | int(11) | primary |
| Table\_id | varchar(250) | --- |
| gross\_amount | decimal(12,2) | --- |
| tax\_amount | decimal(12,2) | --- |
| net\_amount | decimal(12,2) | --- |
| created | datetime | --- |
| created\_by | enum('admin', 'waiter', 'cashier') | --- |
| status | enum('In Process','Completed') | --- |

* **RESTAURANT\_ORDER\_ITEM :**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Constraint** |
| id | int(11) | primary |
| order\_id | int(11) | --- |
| category\_id | int(11) | --- |
| Item\_id | int(11) | --- |
| quantity | int(4) | --- |
| created | datetime | --- |
| rate | decimal(12,2) | --- |
| amount | decimal(12,2) | --- |

* **RESTAURANT\_TABLE :**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Constraint** |
| id | int(11) | primary |
| name | varchar(250) | --- |
| capacity | int(3) | --- |
| Item\_id | int(11) | --- |
| quantity | int(4) | --- |
| status | enum(‘Enable’,’Disable’) | --- |

* **RESTAURANT\_TAX :**

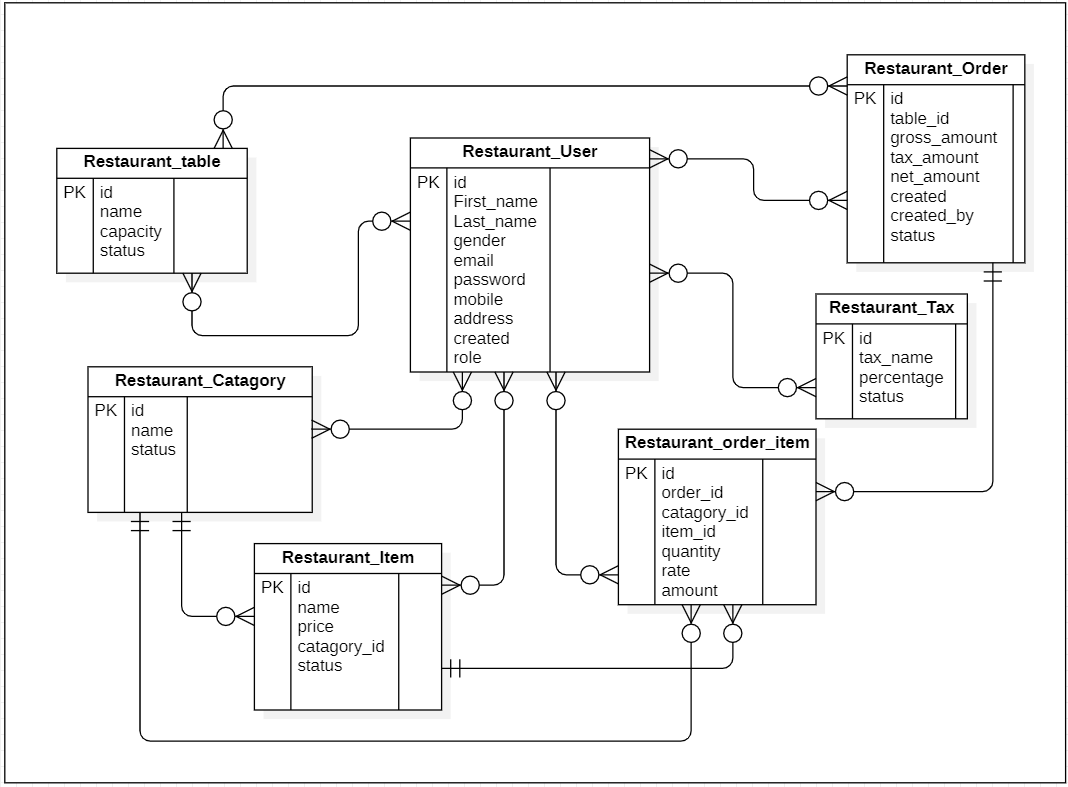
|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Constraint** |
| id | int(11) | primary |
| tax\_name | varchar(250) | --- |
| percentage | decimal(4,2) | --- |
| status | enum(‘Enable’,’Disable’) | --- |

* **RESTAURANT\_USER :**

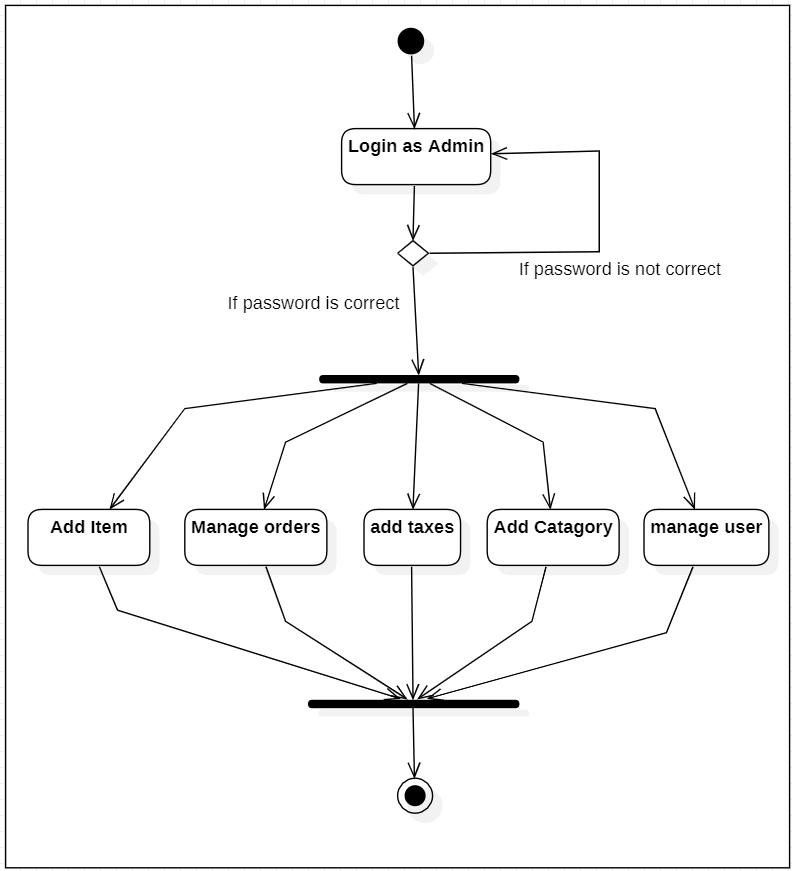
|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Constraint** |
| id | int(11) | primary |
| first\_name | varchar(250) | --- |
| last\_name | varchar(250) | --- |
| gender | enum('male', 'female') | --- |
| email | varchar(250) | --- |
| password | varchar(250) | --- |
| mobile | varchar(250) | --- |
| address | text | --- |
| created | datetime | --- |
| role | enum('admin', 'waiter', 'cashier') | --- |

## 3.3 FIGURES :

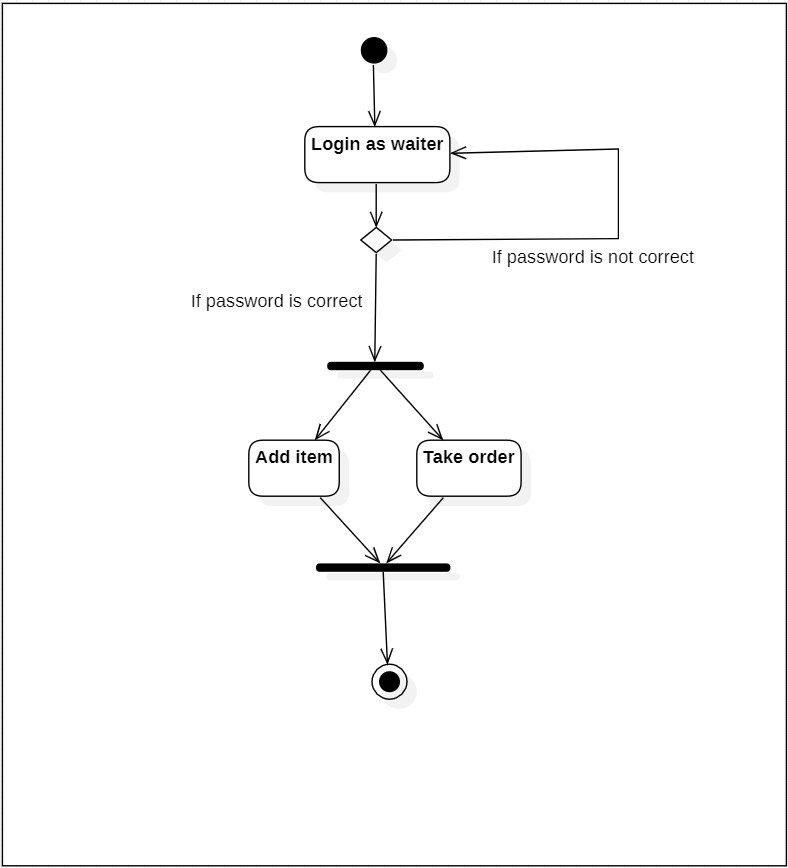
**3.3.1 E.R. DIAGRAM :**



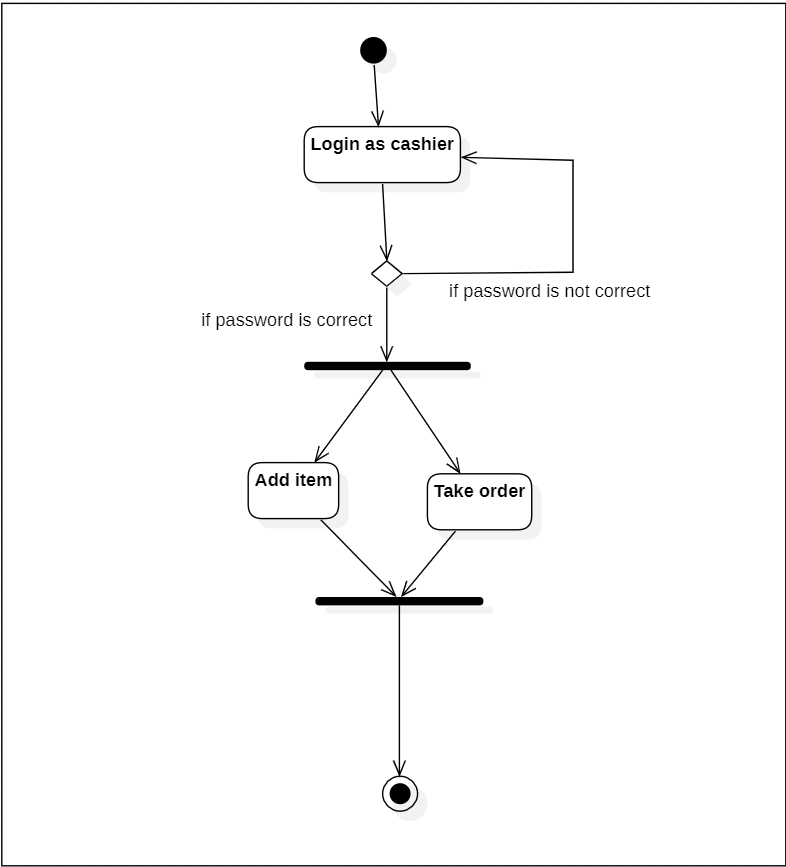
**3.3.2 ACTIVITY DIAGRAM :** [ Admin activity diagram ]



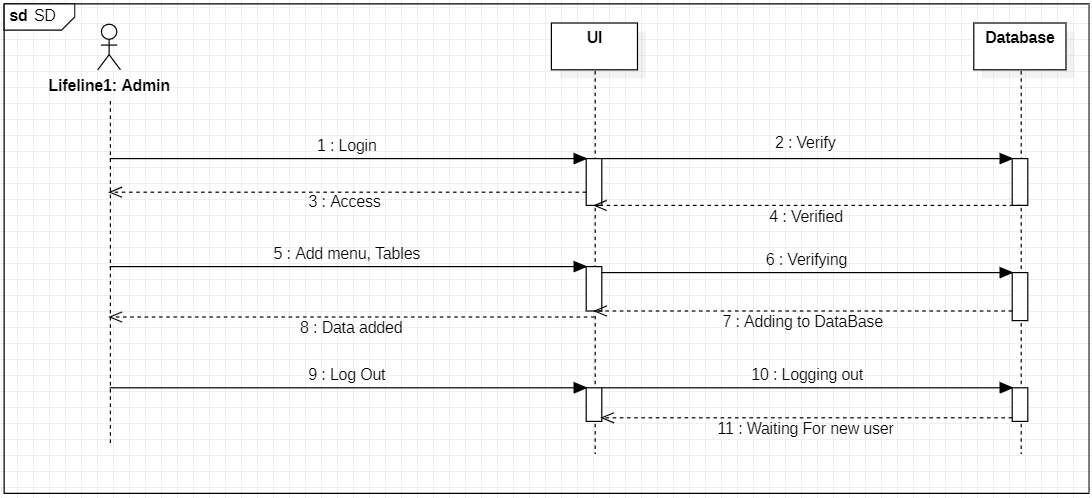
**3.3.2 ACTIVITY DIAGRAM :** [ Waiter activity diagram ]



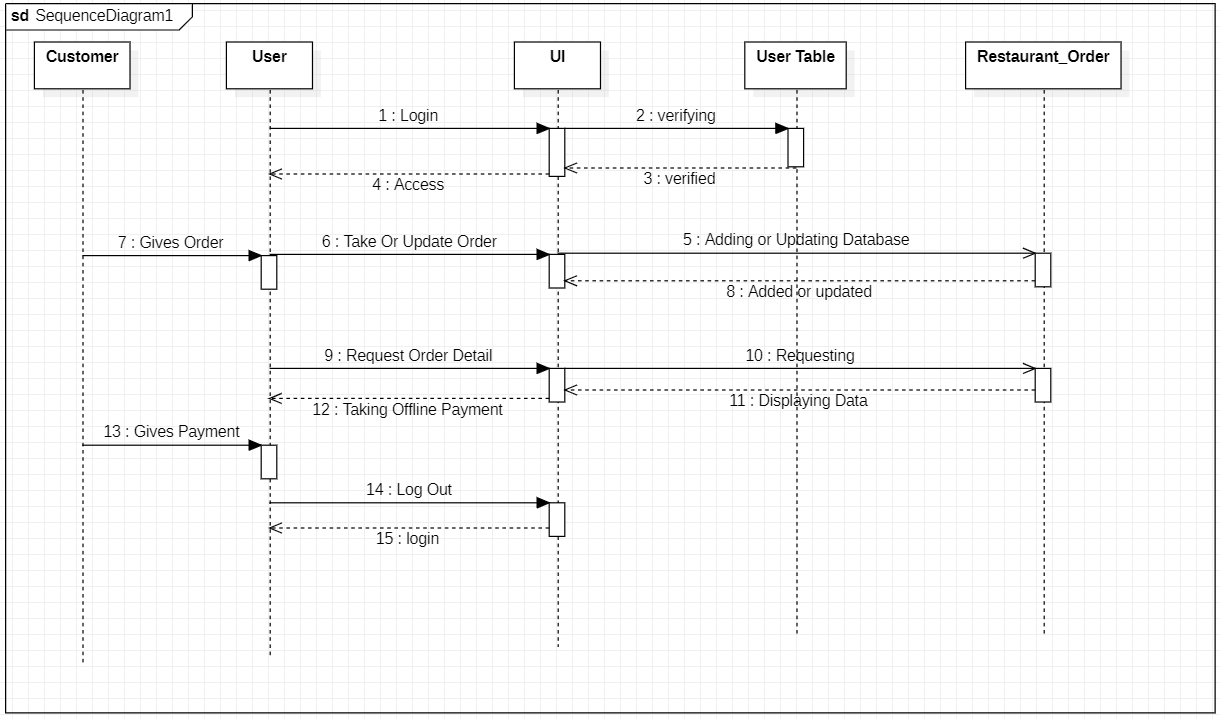
**3.3.2 ACTIVITY DIAGRAM :** [ Cashier activity diagram ]



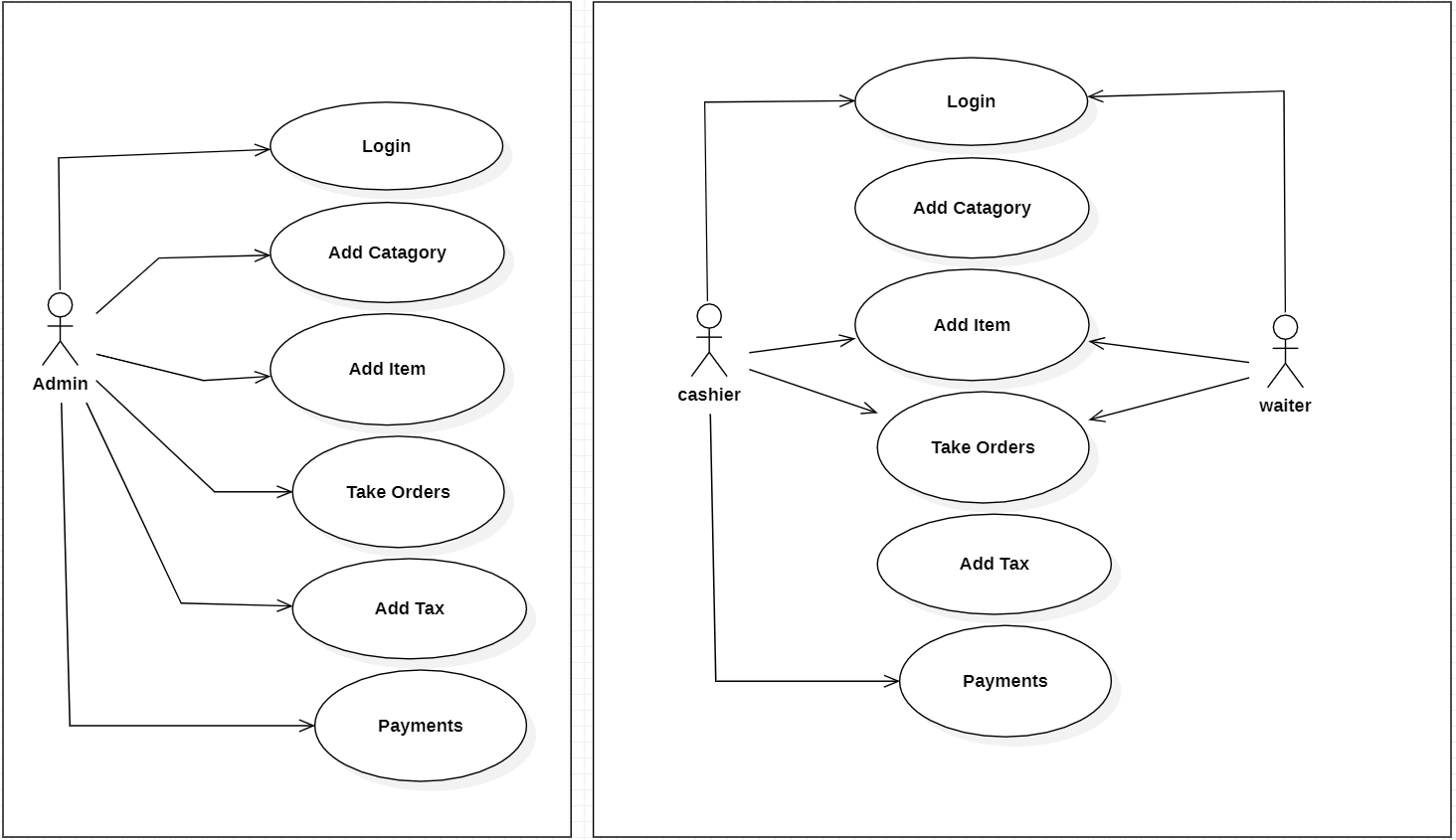
**3.3.3 SEQUENCE DIAGRAM :** [ Admin sequence diagram ]



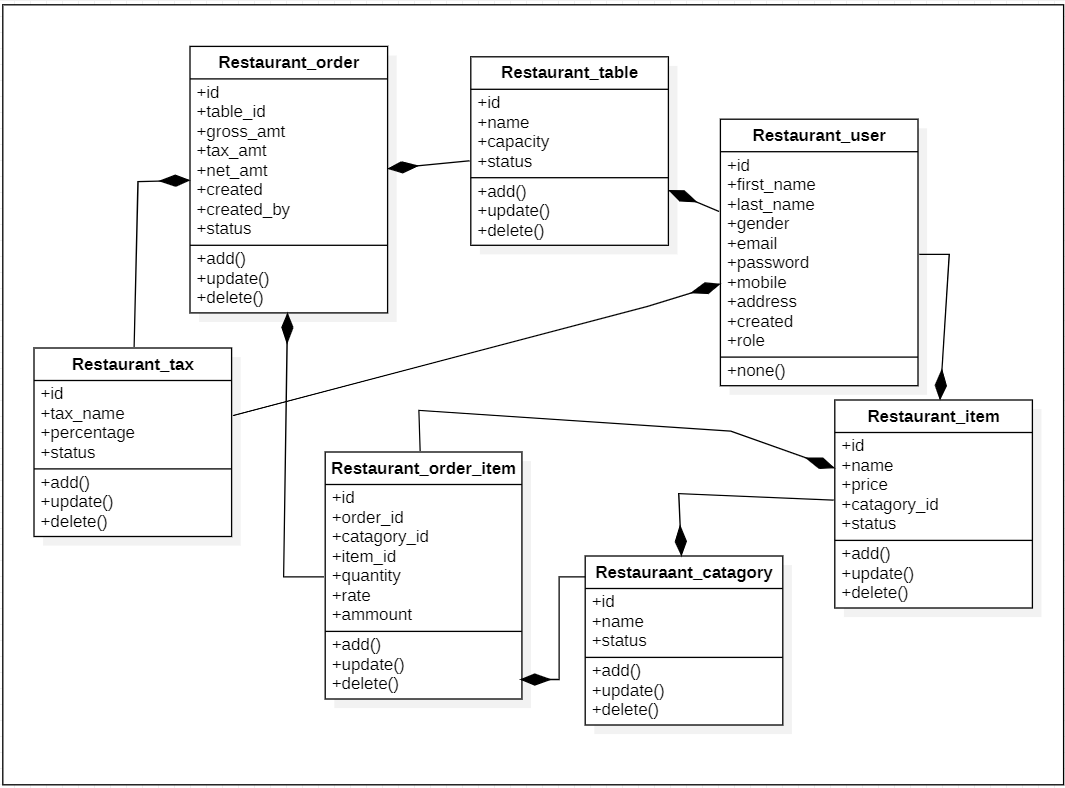
**3.3.3 SEQUENCE DIAGRAM :** [ User sequence diagram ]



**3.3.4 USE CASE DIAGRAM :**

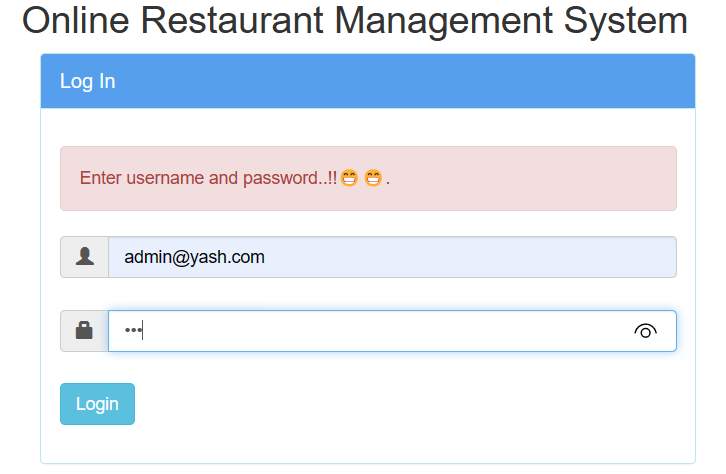


**3.3.5 CLASS DIAGRAM :**

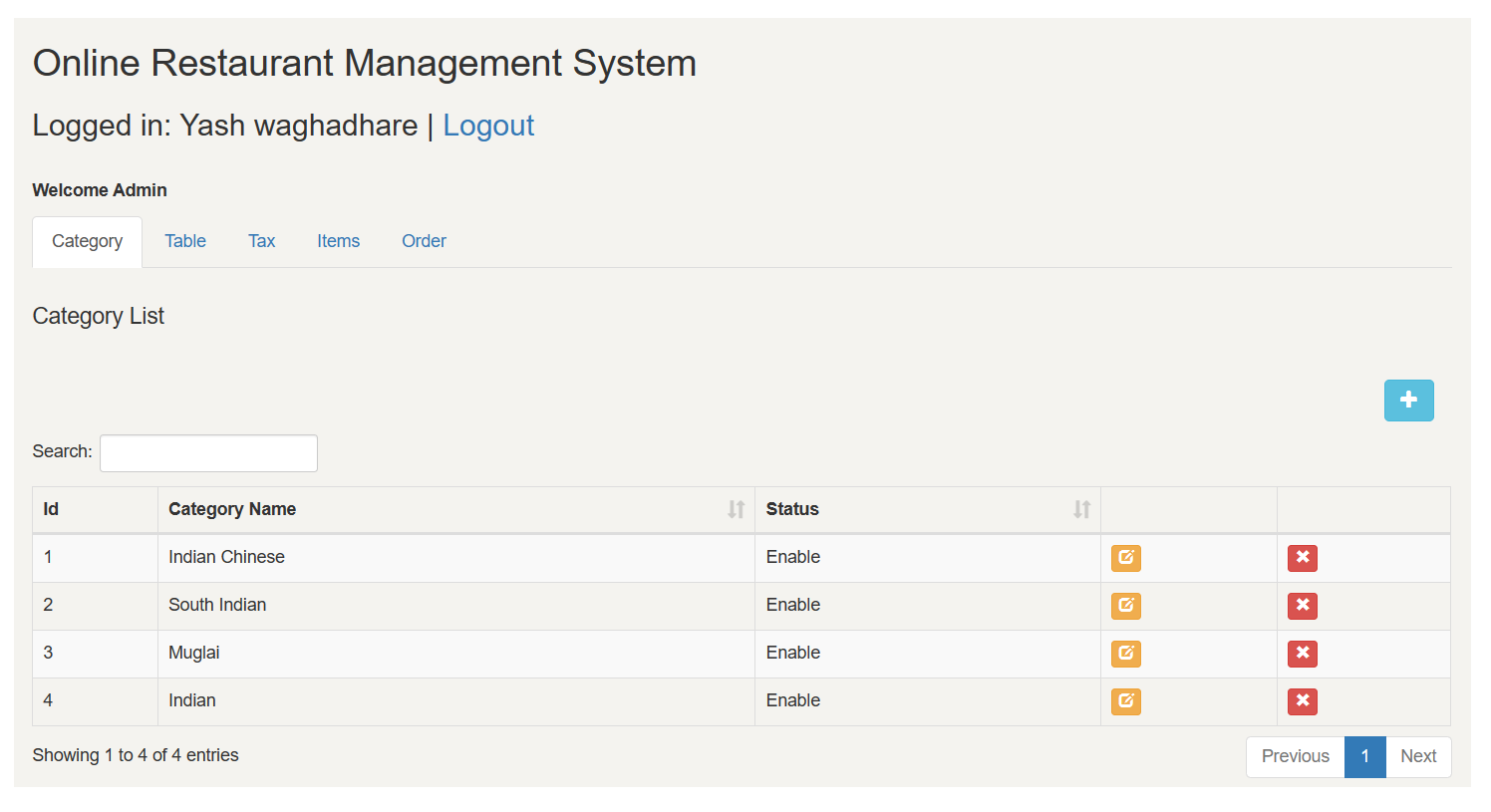


**3.4 USER INTERFACE :**

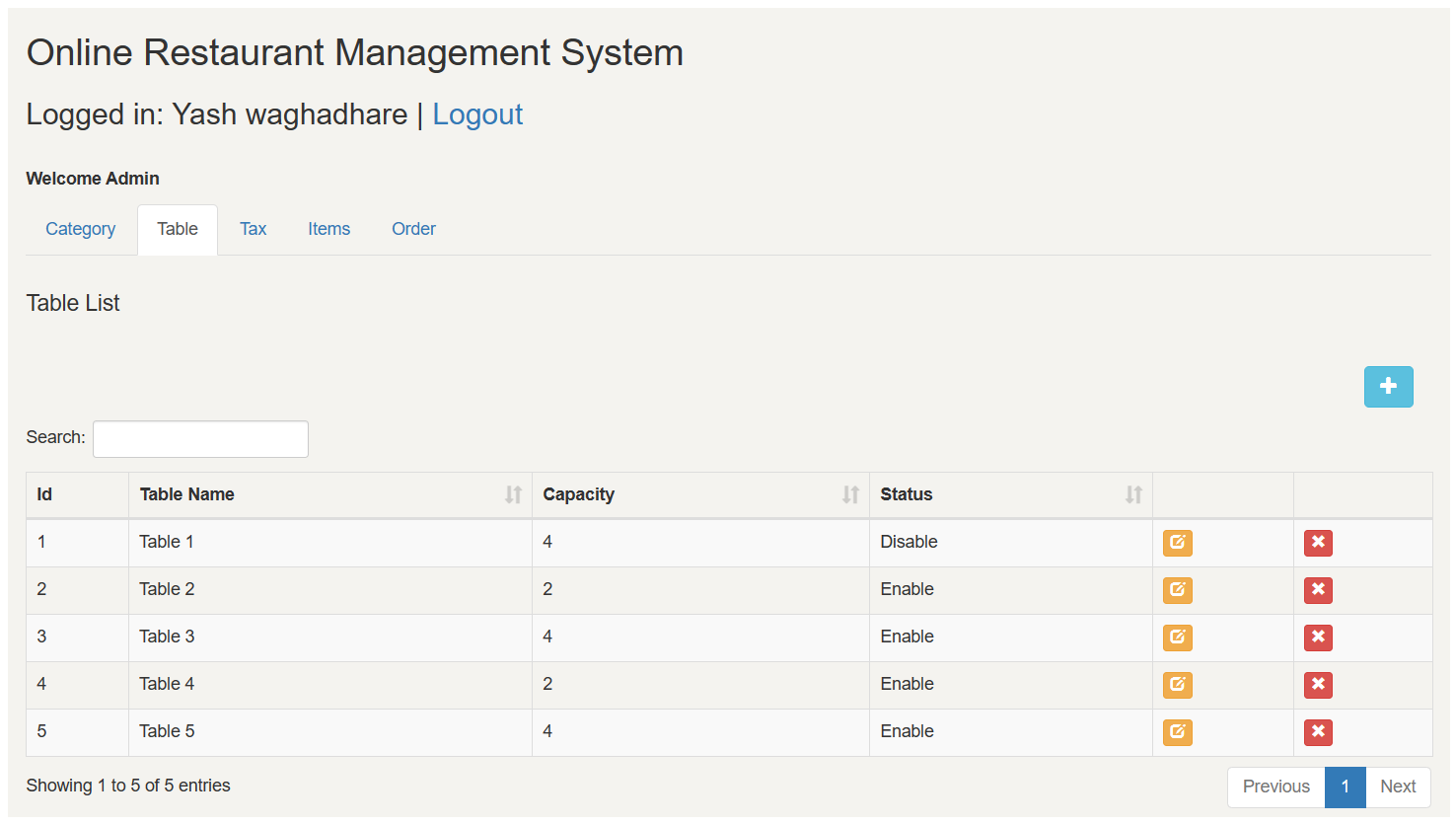
* **LOGIN PAGE :**



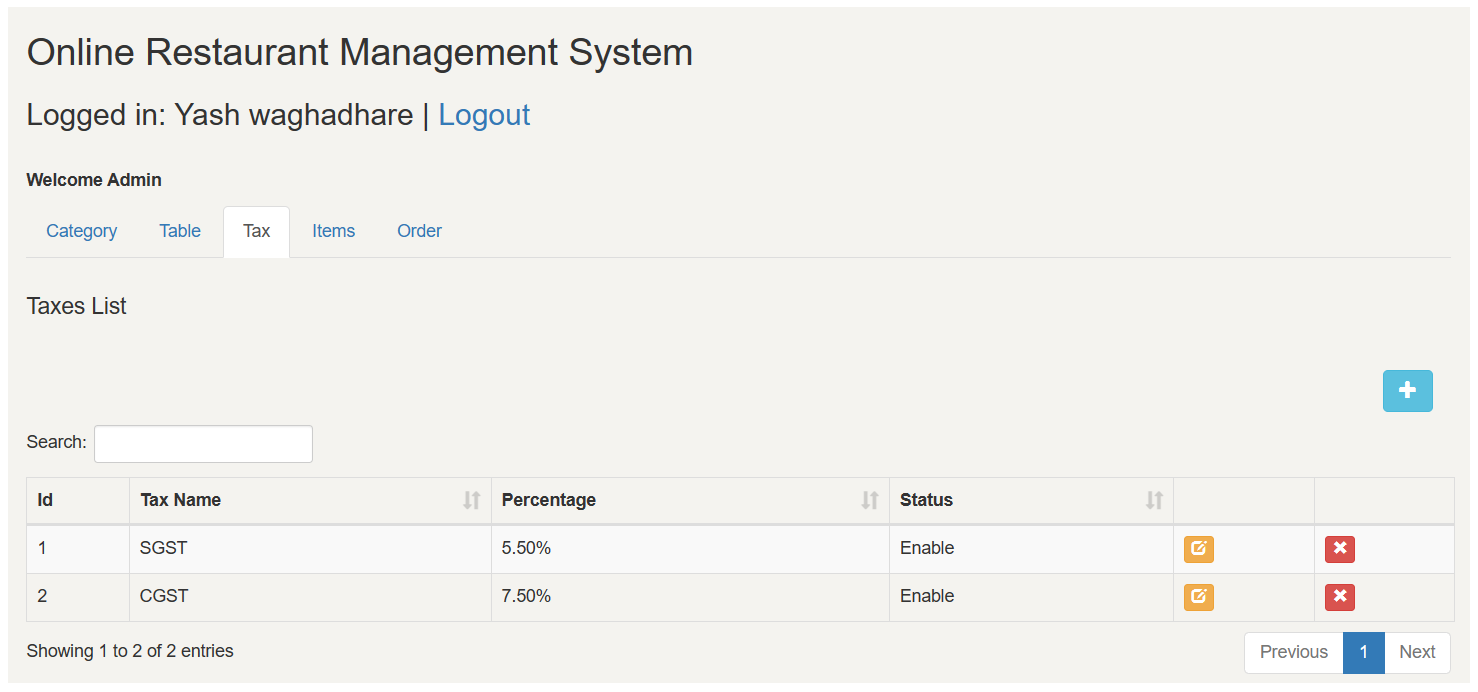
* **CATEGORY MODULE :**



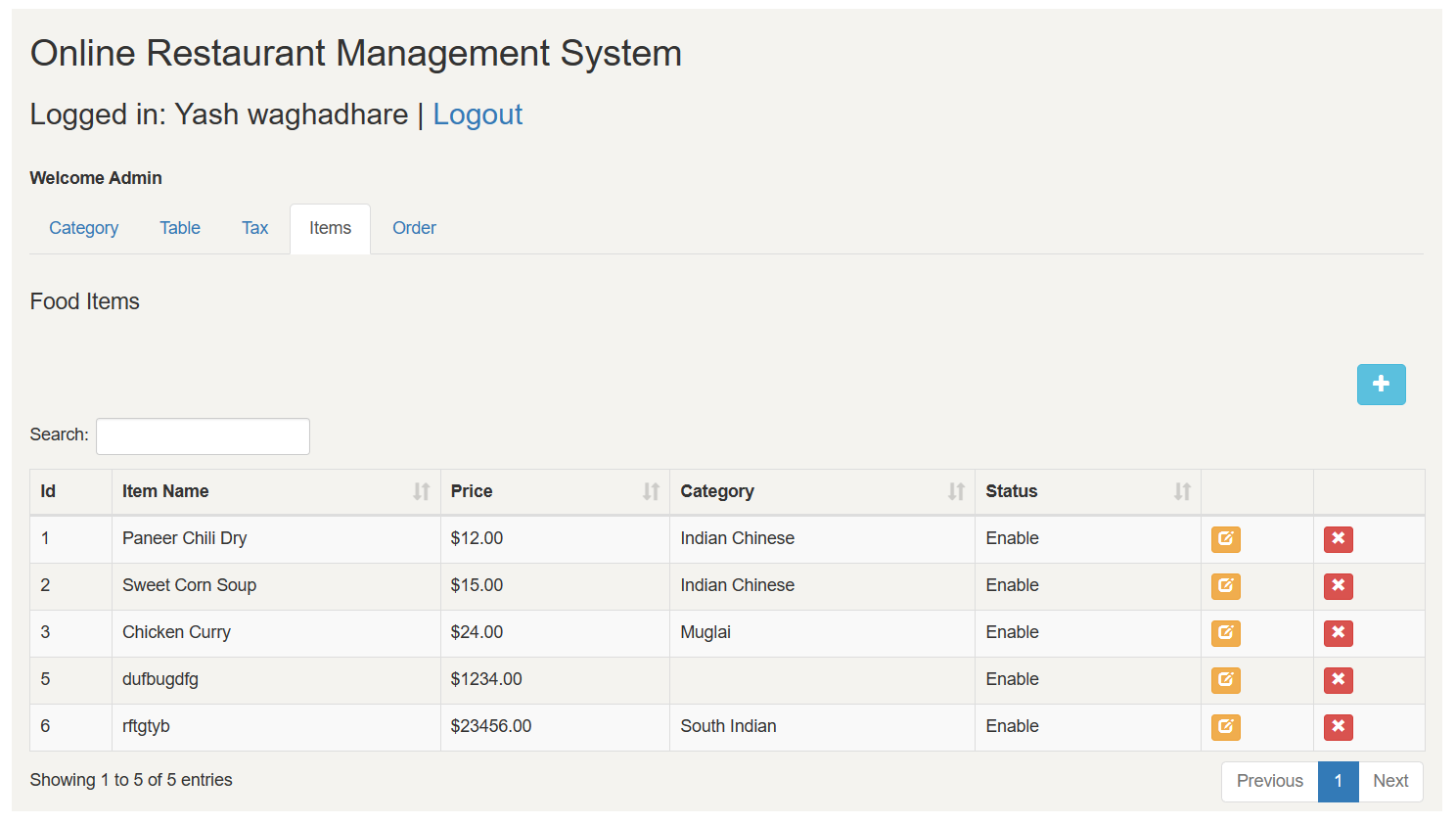
* **TABLE MODULE :**



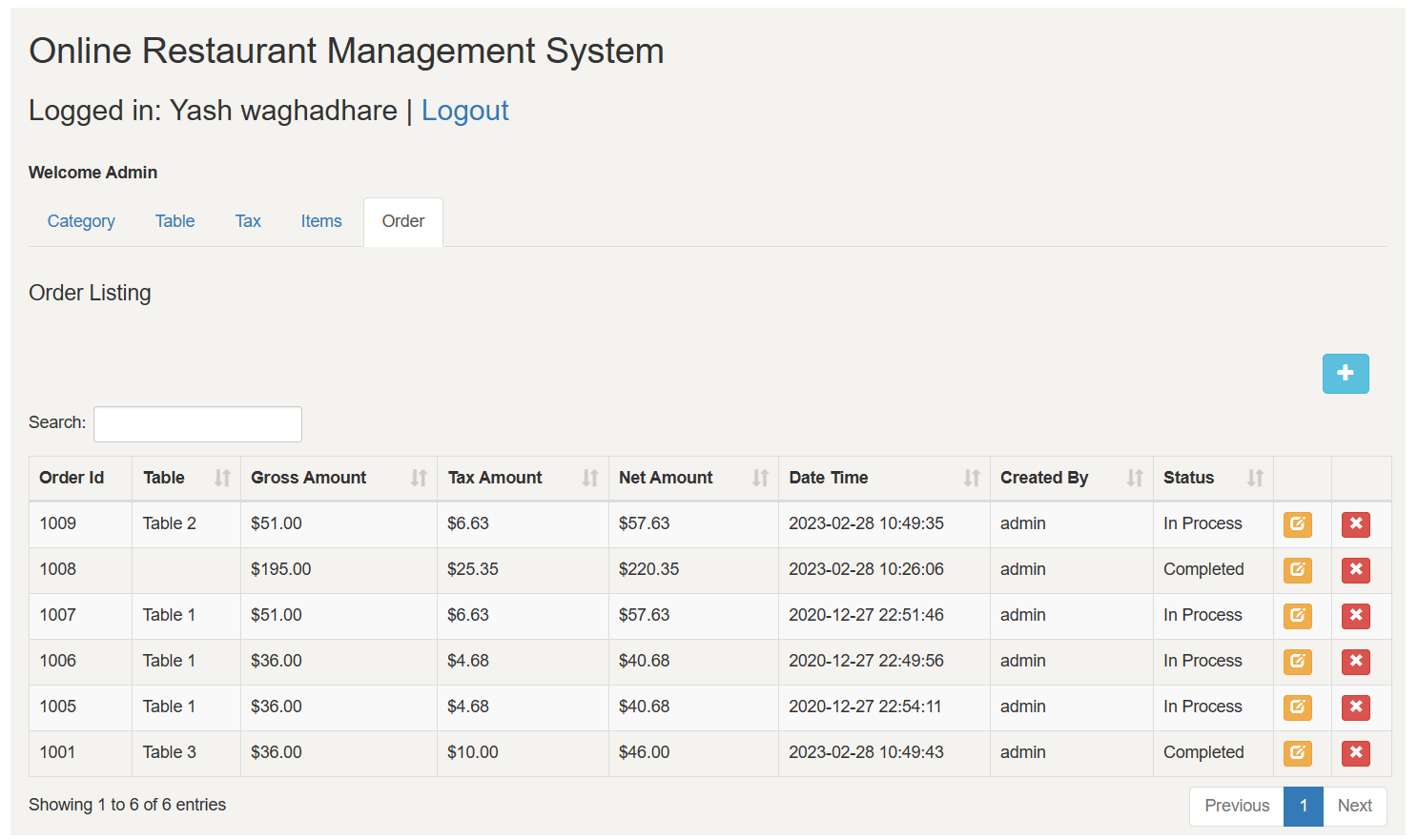
* **TAX MODULE :**



* **ITEM MODULE :**



* **ORDER MODULE :**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Event** | **Trigger** | **Source** | **Destination** | **Activity** | **Response** |
| 1 | Login | Login account | Admin  + User | Server | Login using Username and password | Redirect to next page if the password is correct |
| 2 | Create Category | Add new category | Admin  + User | Server | Add a new category to the database | Get pop-up message |
| 3 | Edit  Category | Edit existing category | Admin  + User | Server | Edit existing category from database | Get pop-up message |
| 4 | Delete Category | Delete existing category | Admin  + User | Server | Remove existing category from database | Get pop-up message |
| 5 | Create  Table | Add new table | Admin | Server | Add a new table number to the database | Get pop-up message |
| 6 | Edit  Table | Edit existing table | Admin | Server | Edit existing table number from database | Get pop-up message |
| 7 | Delete  Table | Delete existing table | Admin | Server | Remove existing table number from database | Get pop-up message |
| 8 | Create  Tax | Add new tax | Admin | Server | Add a new tax to the database | Get pop-up message |
| 9 | Edit  Tax | Edit existing tax | Admin | Server | Edit existing tax from database | Get pop-up message |
| 10 | Delete  Tax | Delete existing tax | Admin | Server | Remove existing tax from database | Get pop-up message |
| 11 | Create  Item | Add new item | Admin | Server | Add a new item to the database | Get pop-up message |
| 12 | Edit  Item | Edit existing item | Admin | Server | Edit existing item from database | Get pop-up message |
| 13 | Delete  Item | Delete existing item | Admin | Server | Remove existing item from database | Get pop-up message |
| 14 | Create  Order | Add new order | Admin  + User | Server | Add a new order to the database | Show detailes |
| 15 | Edit  Order | Edit existing order | Admin  + User | Server | Edit existing order from database | Get pop-up message |
| 16 | Delete  Order | Delete existing order | Admin  + User | Server | Remove existing order from database | Get pop-up message |

**3.5 EVENT TABLE :**

**CHAPTER 4 : IMPLEMENTATION AND TESTING**

“An untested idea is just a thought; tested and implemented, it can transform the world.”

* 1. **CODE :**
* **Database.php**

<?php

session\_start();

class Database{

private $host = 'localhost';

private $user = 'root';

private $password = "";

private $database = "khamang\_restaurants";

public function getConnection(){

$conn = new mysqli($this->host, $this->user, $this->password, $this->database);

if($conn->connect\_error){

die("Error failed to connect to MySQL: " . $conn->connect\_error);

}

else {

return $conn;

}

}

}

?>

* **Index.php**

<?php

include\_once 'config/Database.php';

include\_once 'class/User.php';

$database = new Database();

$db = $database->getConnection();

$user = new User($db);

if ($user->loggedIn()) {

header("Location: category.php");

}

$loginMessage = '';

if (!empty($\_POST["login"]) && !empty($\_POST["email"]) && !empty($\_POST["password"]))

{

$user->email = $\_POST["email"];

$user->password = $\_POST["password"];

$user->loginType = $\_POST["loginType"];

if ($user->login()) {

header("Location: category.php");

} else {

$loginMessage = 'Invalid login! Please try again.';

}

}

else if (empty($\_POST["login"]) || empty($\_POST["email"]) || empty($\_POST["password"]))

{

$loginMessage = 'Enter username and password..!!';

}

include('inc/header.php');

?>

<title>Khamang-Restaurant</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>

<?php include('inc/container.php'); ?>

<div class="content">

<div class="container-fluid">

<h2>Online Restaurant Management System</h2>

<div class="col-md-6">

<div class="panel panel-info">

<div class="panel-heading" style="background:#167ce6ba;color:white;">

<div class="panel-title">Log In</div>

</div>

<div style="padding-top:30px" class="panel-body">

<?php if ($loginMessage != '') { ?>

<div id="login-alert" class="alert alert-danger col-sm-12">

<?php echo $loginMessage; ?>

</div>

<?php } ?>

<form id="loginform" class="form-horizontal" role="form" method="POST" action="">

<div style="margin-bottom: 25px" class="input-group">

<span class="input-group-addon"><i class="glyphicon glyphicon-user"></i></span>

<input type="text" class="form-control" id="email" name="email" value="<?php if (!empty($\_POST["email"])) {

echo $\_POST["email"];

} ?>" placeholder="email" style="background:white;" required>

</div>

<div style="margin-bottom: 25px" class="input-group">

<span class="input-group-addon"><i class="glyphicon glyphicon-lock"></i></span>

<input type="password" class="form-control" id="password" name="password" value="<?php if (!empty($\_POST["password"])) {

echo $\_POST["password"];

} ?>" placeholder="password" required>

</div>

<div style="margin-top:10px" class="form-group">

<div class="col-sm-12 controls">

<input type="submit" name="login" value="Login" class="btn btn-info">

</div>

</div>

</form>

</div>

</div>

</div>

</div>

<?php include('inc/footer.php'); ?>

**4.2 TESTING APPROACH :**

A test approach is a high-level plan or strategy that outlines how testing will be conducted for a particular software application or system. It defines the overall testing objectives, scope, and methodology to be used during the testing process.

The following are some of the key components that are typically included in a test approach:

1. **Test Objectives :**

This outlines the goals of the testing effort, including what is expected to be achieved through the testing process.

1. **Scope :**

This defines the boundaries of the testing effort, including the features, functionalities, and areas of the software that will be tested.

1. **Test Methodology :**

This outlines the testing techniques and methods that will be used to verify the software, such as manual testing, automated testing, or a combination of both.

1. **Test Environment :**

This outlines the hardware and software configurations that will be used during the testing process, as well as any specific tools or systems that will be used.

1. **Test Echedule :**

This outlines the timeline for testing, including when testing will begin, when it will end, and any milestones or deadlines that must be met.

1. **Test Deliverable :**

This outlines the documentation and reports that will be produced as part of the testing process, such as test plans, test cases, test scripts, and defect reports.

By developing a comprehensive test approach, testing teams can ensure that they have a clear understanding of the testing objectives, scope, and methodology, which can help them to more effectively plan, execute, and manage the testing process.

* **INTEGRATION TESTING:**

1. Integration testing is a type of testing that focuses on testing the interactions between different modules or components of a software system.
2. The purpose of integration testing is to identify and resolve any issues that may arise when different components are combined and tested as a group.
3. Integration testing can be performed at different levels, such as module-level, subsystem-level, and system-level.
4. Different techniques can be used to perform integration testing, such as top-down, bottom-up, and hybrid approaches.
5. Integration testing typically involves creating test cases that simulate realistic usage scenarios and validate the interactions between different components.

* **Why I used Integration Technique in my project:**

Integration testing is used to test the interactions between different modules or components of a software system. It helps to identify any errors or issues that may arise when different components are combined and tested as a group. This type of testing is important because software systems are often complex and made up of many interconnected parts. Integration testing ensures that these parts work together as intended, and helps to prevent issues from arising in production.

* **How I used Integration Technique in my project:**

To perform integration testing, we start by testing individual modules or components in isolation (this is called unit testing). Once all the individual components have passed unit testing, we begin to integrate them and test their interactions with one another. There are different integration testing techniques that can be used, such as top-down or bottom-up integration testing. The choice of technique depends on the specific project requirements.

* **GUI TESTING :**

1. GUI testing is a type of testing that focuses on testing the graphical user interface (GUI) of a software application.
2. The purpose of GUI testing is to ensure that the application's user interface is functional, user-friendly, and meets the requirements of the end-users.
3. GUI testing involves testing different aspects of the user interface, such as the layout, design, navigation, controls, and responsiveness.
4. Different techniques can be used to perform GUI testing, such as manual testing, automated testing, and exploratory testing.
5. GUI testing typically involves creating test cases that validate the functionality of different GUI elements, such as buttons, menus, dials, and forms, and verifying that they behave as expected.

* **Why I used GUI Testing in my project:**

GUI testing is used to test the graphical user interface (GUI) of a software application. It ensures that the application's user interface is functional, user-friendly, and meets the requirements of the end-users. This type of testing is important because the GUI is often the most visible part of the software, and it is the part that end-users interact with the most.

* **How I used GUI Testing in my project:**

To perform GUI testing, we create test cases that validate the functionality of different GUI elements, such as buttons, menus, dialogs, and forms, and verify that they behave as expected. We can use manual testing, automated testing, or exploratory testing techniques to perform GUI testing. Additionally, we can use different tools and frameworks to automate GUI testing and reduce the amount of manual effort required.

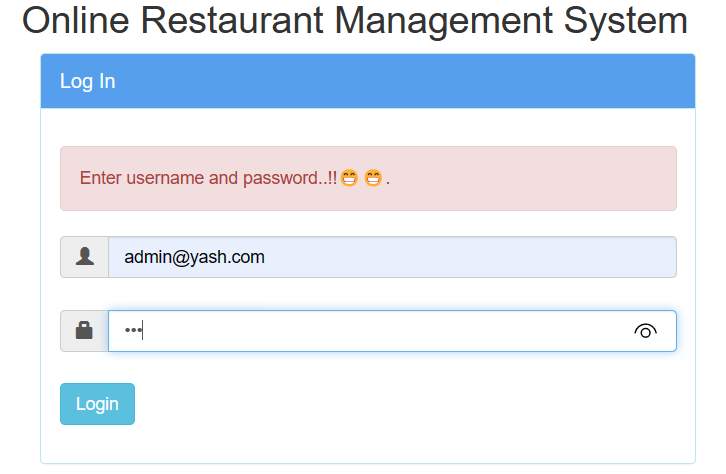
**4.3 TEST CASES :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Test case name** | **Possible Input** | **Expected Result** | **Actual Result** | **Conclusion** |
| 1 | Login | Valid username and password. | Login successful message and redirected to dash bored. | Login successful message and Redirect to dash bored. | Pass |
| Invalid username and password. | Invalid username, password and Login fail message. | Invalid username, password and Login fail message. | Pass |
| 2 | Manage Category | Empty Category Fields | Wrong input message | Wrong input message | Pass |
| Invalid Input | Invalid Input message. | Invalid Input message. | Pass |
| 3 | Manage Table | Empty Table Fields | Wrong input message | Wrong input message | Pass |
| Invalid Input | Invalid Input message. | Invalid Input message. | Pass |
| 4 | Manage Tax | Empty Tax Fields | Wrong input message | Wrong input message | Pass |
| Invalid Input | Invalid Input message. | Invalid Input message. | Pass |
| 5 | Manage Item | Empty Items Fields | Wrong input message | Wrong input message | Pass |
| Invalid Input | Invalid Input message | Invalid Input message. | Pass |
| 6 | Manage Order | Empty Fields | Wrong input message | Wrong input message | Pass |

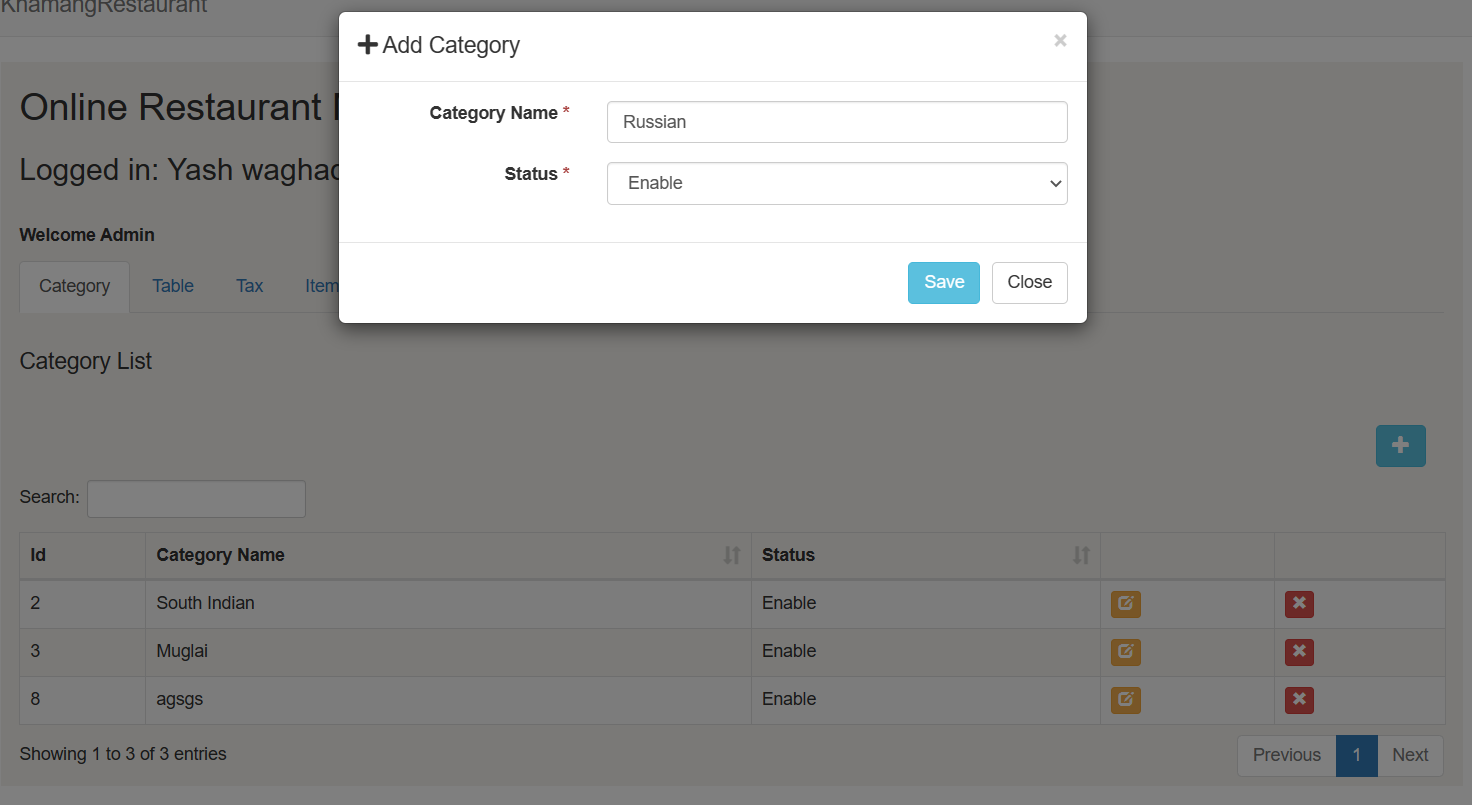
|  |  |  |
| --- | --- | --- |
| **GUI TESTING** | | |
| **GUI COMPONENTS** | **DESCRIPTION** | **CONCLUSION** |
| Navigation | Navigation of the application to ensure that users can easily move from one screen to another and can access all necessary features and functions. | Pass |
| Layout | Layout of the application to ensure that it is consistent across all screens, and that elements such as buttons, text fields, and images are properly aligned and spaced. | Pass |
| Validation | Input validation of the application to ensure that it can handle different types of input, such as text, numbers, and special characters, and that it displays appropriate error messages when input is invalid. | Pass |
| Usability | Usability of the application to ensure that it is easy to use and understand, and that it meets the needs of the intended users. | Pass |
| Performance | Performance of the application to ensure that it is fast, responsive, and can handle a large number of users and transactions. | Pass |
| Security | Security of the application to ensure that it is protected from unauthorized access, data breaches, and other security threats. | Pass |

**CHAPTER 5 : RESULT AND DISCUSSION**

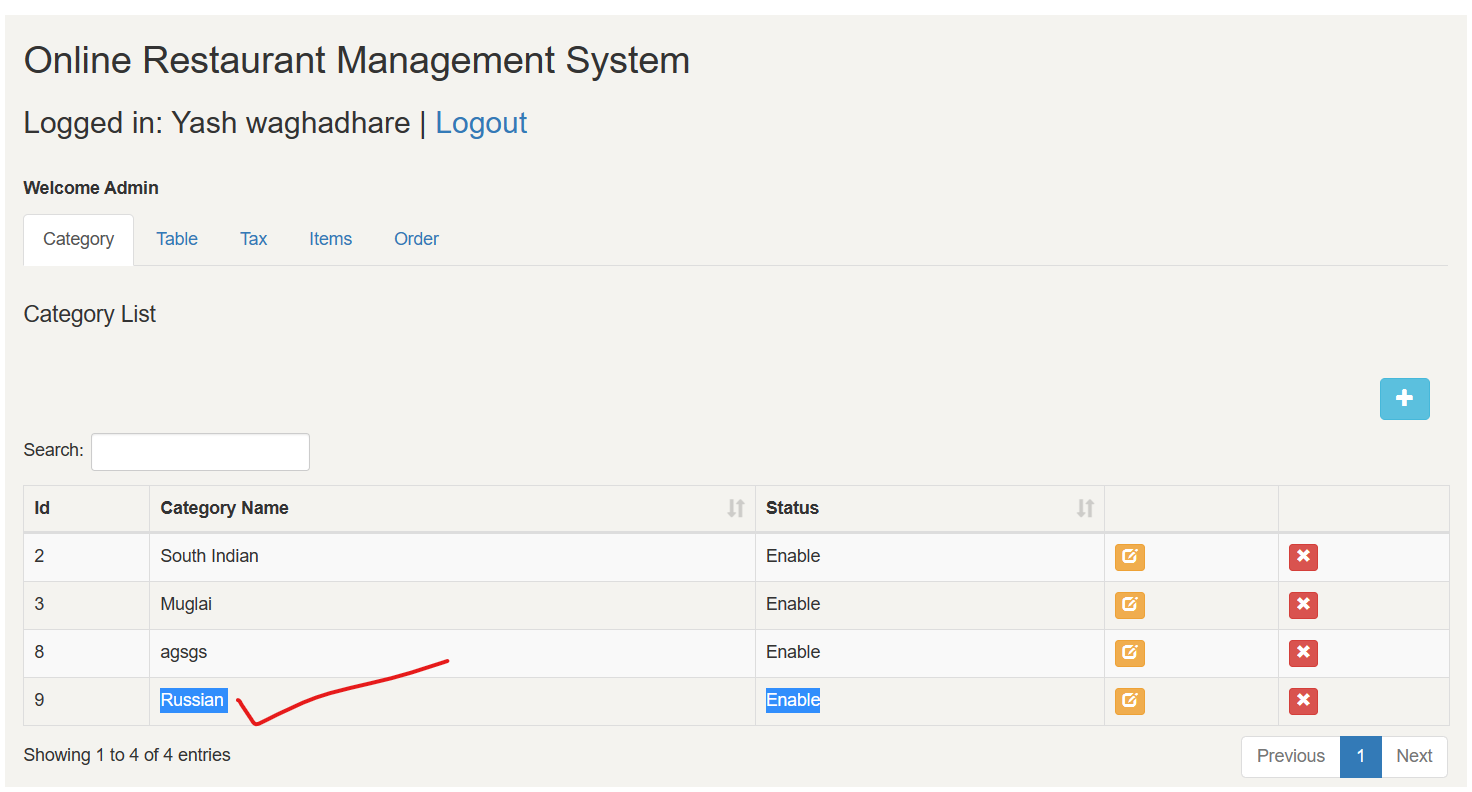
* **LOGIN PAGE :**
* This is the first page that appears when anyone opens the link.
* It asks Username and Password i.e. admin@yash.com and 123.



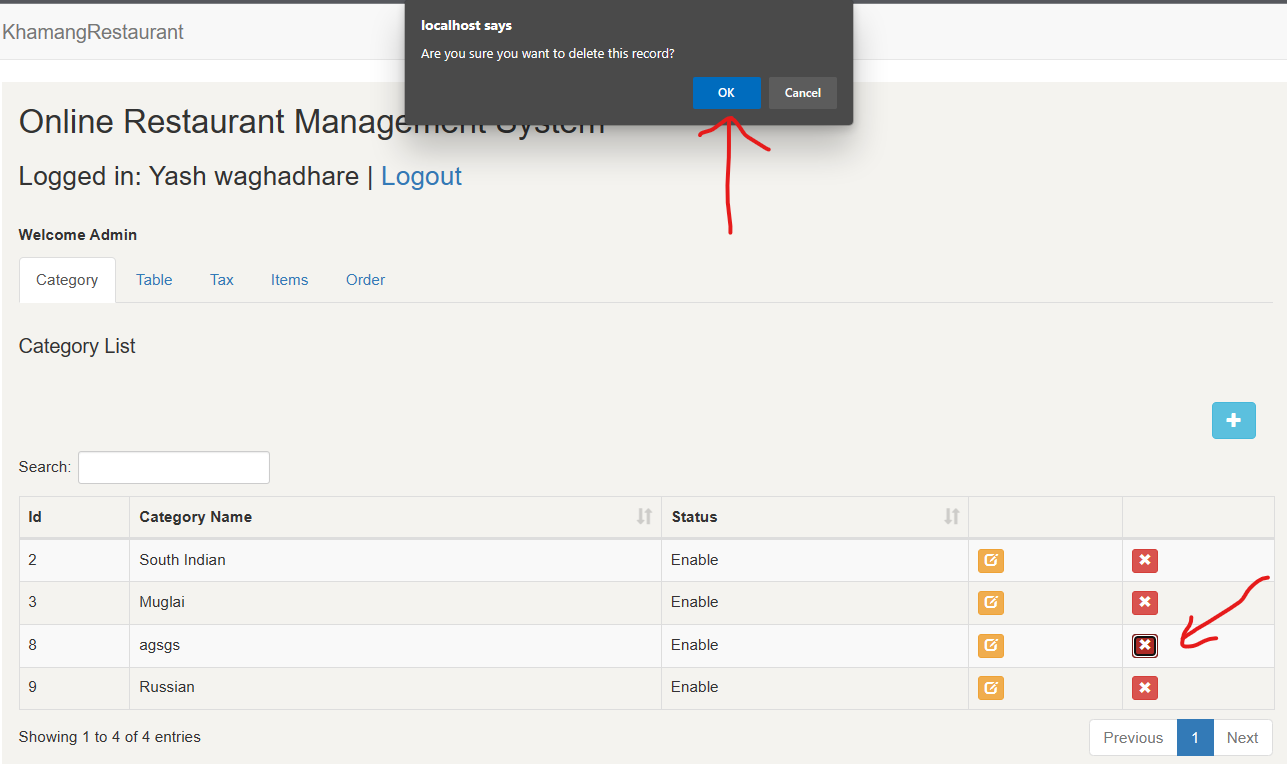
* **MAIN PAGE :**
* Includes all the modules here that allow for users and admin.
* On the admin front page, you can add categories , items , tax, table and you can take order.
* If you want, then you can update or able to delete content that you added.
* Let me take an example of the category module here and then order module.
* **CATEGORY MODULE :**
* Just click on the add button [+] and set the name to category, then select status enable or disable and hit save :



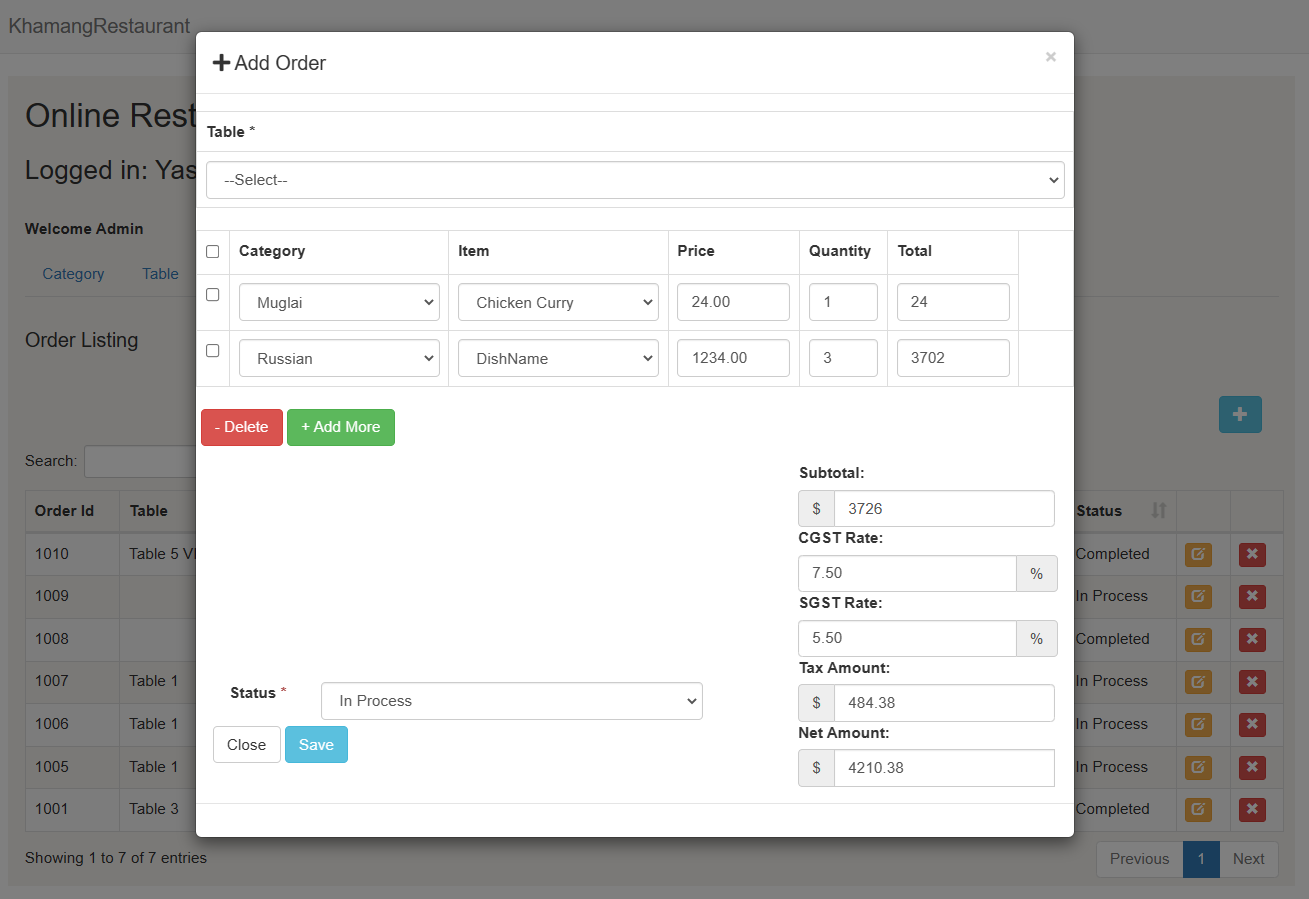
* Then it updates the runtime :



* Then let's delete one category , Just click [X] button and alert-box appears then, click OK and done :



* **ORDER PAGE :**
* "Our order module: VIP treatment for your meal..!!"
* Just click on the [+] icon and then select the table name, then select the category, item and quantity and click on save.
* Here, no need to explain how and where you should click because of the GUI.



**CHAPTER 6 : CONCLUSION**

Customer satisfaction is the backbone of any successful business, and the restaurant industry is no exception. Online restaurant management systems offer a range of benefits that help to improve operational efficiency and customer satisfaction. With these systems, restaurants can easily automate tasks such as order management, inventory tracking, and employee scheduling. This results in faster service times, fewer errors, and a better overall experience for customers.

One of the key benefits of online restaurant management systems is their ability to provide managers and owners with real-time data about their restaurant's performance. This data can be used to make informed decisions about menu items, pricing, etc, which can ultimately lead to increased profitability. Furthermore, by empowering employees with the tools they need to efficiently manage their tasks, these systems can help to create a more productive and motivated workforce, leading to better customer service and increased customer loyalty.

In summary, the use of online restaurant management systems is essential for businesses looking to improve their operational efficiency and customer satisfaction. By investing in these systems, restaurant owners and managers can streamline their operations, provide better service to customers, and, ultimately, achieve long-term success.

**CHAPTER 7 : REFERENCE**

* **WEBSITE REFERRED :**

<https://www.google.com/>

<https://github.com/>

<https://www.geeksforgeeks.org/>

<https://www.youtube.com/>