

NP000611-CT081-3-3-MWM- Individual-Assignment.docx

by SUJAN SHRESTHA .

Submission date: 10-Dec-2023 07:32PM (UTC+0800)

Submission ID: 2253606481

File name: 43820_SUJAN_SHRESTHA_.NP000611-CT081-3-3-MWM-Individual-Assignment_426888_1358825512.docx (10.13M)

Word count: 2406

Character count: 13428



ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT081-3-3-MWM

MOBILE & WEB MULTIMEDIA

NP3F2304IT

HAND OUT DATE: AUGUST 2023

HAND IN DATE: DECEMBER 2023

WEIGHTAGE: 50%

INSTRUCTIONS TO CANDIDATES:

- 1 Submit your assignment at the administrative counter.
- 2 Students are advised to underpin their answers with the use of references (cited using the Harvard Name System of Referencing).
- 3 Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld.
- 4 Cases of plagiarism will be penalized.
- 5 The assignment should be bound in an appropriate style (comb bound or stapled).
- 6 Where the assignment should be submitted in both hardcopy and softcopy, the softcopy of the written assignment and source code (where appropriate) should be on a CD in an envelope / CD cover and attached to the hardcopy.
- 7 You must obtain 50% overall to pass this module.



Acknowledgement

To carry out our Assignment, we had to grab the help and guidelines of some respected personalities and we are tremendously privileged to receive such unwavering amount of support for the accomplishment of our work. The achievement of this assignment gave us immense pleasure, so we would like to express our thankfulness towards **Asia Pacific University (APU) and Lord Buddha Education Foundation (LBEF)** who have provided us the golden opportunity to transform our knowledge and ideas into theoretical concepts and practical.

Furthermore, we would like to extend our profound gratitude to our course Lecturer, Mr. Nutan Sir for providing us with all the constructive guidelines were required for the completion Of our assignment. This research project would not have been possible without his fulfill support, active leadership and encouragement Lastly, we would like to thank my faculty members who have directly or indirectly helped us by conveying crucial information which was beneficial to get the final outcome of our Assignment.

We have gone through several investigations, sites, and books to get exact information for analysis and tried to find the best conclusion.

Your Sincerely,

Sujan Shrestha (NP000611)

BSc.IT, 5TH Semester

Table of Contents

1	Introduction.....	1
1.1	Objectives.....	1
1.2	Scope.....	1
1.3	Target Audiences	1
1.4	Project Schedule.....	2
1.5	Summary of Project.....	2
2	System Requirements	3
2.1	Functional Requirements	3
3	Design.....	4
3.1	Entity Relationship Diagram	4
3.2	Use Case Diagram.....	5
3.3	Database Schema.....	6
3.4	Interface Design	7
4	Implementation.....	29
4.1	Factors related to the Design of the User Interface	29
4.2	Factors related to the development and organization of the system for real-world deployment and its implications.	29
4.3	Steps of Implementation of system	30
5	Conclusion.....	37
6	References.....	38
7	Appendices	39
7.1	User Manual	39

Table of Figures

Figure 1: Project Plan.....	2
Figure 2: Entity Relationship Diagram	4
Figure 3: User Case Diagram.....	5
Figure 4: Database Schema.....	6
Figure 5: Home Page	7
Figure 6 : Home Page Mobile View	7
Figure 7: User Registration Page	8
Figure 8:User Registration Page Mobile View	8
Figure 9: Login page	9
Figure 10: Login page Mobile View	9
Figure 11: Categories Page	10
Figure 12: Categories Page Mobile View	10
Figure 13: Sub Category Page	11
Figure 14: Sub-Category Page Mobile View	11
Figure 15: Food Items Page	12
Figure 16: Food Items Page Mobile View	12
Figure 17: Food Items Detail Page	13
Figure 18: Food Item Detail Page Mobile View	13
Figure 19: Cart Page	14
Figure 20: Cart Page Mobile View	14
Figure 21: Payment View.....	15
Figure 22: Payment Page Mobile View	15
Figure 23: Order History Page.....	16
Figure 24: Feedback Page	16
Figure 25: Feedback Page Mobile View	17
Figure 26: Profile Management Page	18
Figure 27: Category Management (CRUD).....	20
Figure 28: Sub-Categories Management (CRUD).....	22
Figure 29: Food Items Management (CRUD)	24
Figure 30: User Management (RUD)	26
Figure 31: Order Status Management Page	26
Figure 32: Order Details Page.....	27
Figure 33: Order Details Page Mobile View.....	28
Figure 34: Database Tables	30
Figure 35: Backend Development Code Snippets	32
Figure 36: Front-end Development Code Snippets	34
Figure 37: Identity Authentication and Authorization Code snippets.....	35
Figure 38: Stripe Payment Integration Code Snippets.....	36
Figure 39: Stripe Dashboard View.....	37

1 Introduction

With the increasing development in the field of technology, all the related aspects have been positively affected by it. This has been a norm for society as it has positively awarded the people with a gift of online technology which has replaced the old method of having to visit the place for themselves. As in this modern society, people's schedule is tighter than ever before which makes the physical purchasing method of food more time consuming and tedious. This has accelerated the creation of a cutting-edge online meal delivery system, providing a quick and effective replacement for conventional techniques.

A web-based project called Feast Flavor Online Food Delivery System aims to provide users with a smooth online ordering experience. With the admin interface, it enables restaurant employees to oversee and change the menu, guaranteeing that the food products and other information on the website are up to date. Likewise, clients may easily use the system for viewing a wide variety of available choices of delicacies on Feast Flavor with the help of online purchase.

C# programming has been used to create the web project's ASP.NET platform, which interfaces with the SQL Server database. SQL Server 2019 is used to make the database connection, and Visual Studio 2022 is used as the programming environment to create the web application. A solid and comprehensive framework for creating an effective and dynamic web-based solution is provided by this technological stack.

1.1 Objectives

The following are some of the objectives behind the project which are listed below:

- To enable a user-friendly and smooth interface for online food ordering.
- To streamline all the processes involved in online food ordering.
- To enhance user engagement by making the system accessible for all.
- To monitor and optimize the system performance for efficient online food ordering.

1.2 Scope

The main scope of the project is to give customers smooth ordering, monitoring, and menu-browsing experience. It has a strong feedback system, a variety of payment choices, and safe user verification.

1.3 Target Audiences

The audience of this application is categorized into two sections which are:

- i) Food Vendors
- ii) Food Consumers
- iii) General Public Browsers
- iv) Event organizers

1.4 Project Schedule

The project aims to be completed with the implementation of the SDLC Model. This model helps to guide the project and complete the required tasks in the correct order for the successful completion of the project. The diagram below helps to show us the different tasks involved in the project model:

Schedule	Task	Completion Days
Week 1	Project Design	4 days
Week 1	Project setup on MVC platform	2 days
Week 2	Authentication Part of the system	3 days
Week 3	Administrative functionalities	A week
Week 4	General User functionalities	A week
Week 5	System Testing	3 days

Figure 1: Project Plan

1.5 Summary of Project

The project was completed in six weeks, with specific goals being met every week. Building registration and login pages for both admin and customers was the first week's priority. The following week was devoted to creating user interfaces for websites and making sure they operated without a hitch. The third week saw the development of the administration panel and improvements to the responsiveness of the website. In the sixth week of the project, the database was put into place, extensive testing was conducted, and any possible problems were troubleshooted and fixed.



2 System Requirements

The system's stakeholders require certain things to be delivered via the online food ordering system. These requirements clearly defines the purpose and primary objectives and goals of the system. Some of the requirements that need to be fulfilled are listed below:

2.1 Functional Requirements

Member Registration

Missing "," 

Login and Registration

Users should be able to create a profile and login to the system with their email and password. This will also ensure the authentication part of the online food delivery system.

Manage Profile

Users should have a feature to view their profile details and edit certain information about their profile.

Cart Functionality:

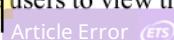
An add to cart system will be added in the Feast Flavor so that users can view and add or remove items to their cart.



Sp. 

Order History:

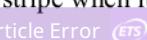
The history of order will act as transaction and purchase history for the users to view their previous orders.



Payment

Support:

The users will be able to pay through certain services like cash on delivery to stripe when it comes to Feast Flavors.

Prep.  Article Error 

Customer Feedback Management:

Users should be able to provide their valuable feedback on delivered orders and admins will be able to view it.

Sp. 

Administrative Function

Manage User (RUD):

Admins should have functionality to view, update and delete user accounts.

S/V 

Manage Products (CRUD):

Admins will be allowed to add, view, update and delete food items.

Sp. 

Manage Categories (CRUD)

Admins will be able to add, view, update and delete certain categories.

Sp. 

Order Management (CRUD)

Order management is an integral part of this system as admins should be able to view and manage the status of orders as Pending, Processing, Shipped or Delivered.

Sp. (ETS)

Missing "", (ETS)

3 Design

This phase of the project includes all the structural designing part of the project where the flow of the data and other processes were design to assist in the development of project (Wiley, 2012).

Certain techniques are used to generate the design of the Feast ~~Flavor~~ Online Food Delivery system

which are provided below.

5 S/V (ETS)

3.1 Entity Relationship Diagram

ERD stands for Entity Relationship Diagram which refers to a structural diagram based on the entities in a database system. This shows the relationship among different entities involved in the system. The ERD helps to assist in the making of databases and establish a good relationship among different models of the system (Cagiltay, 2013).

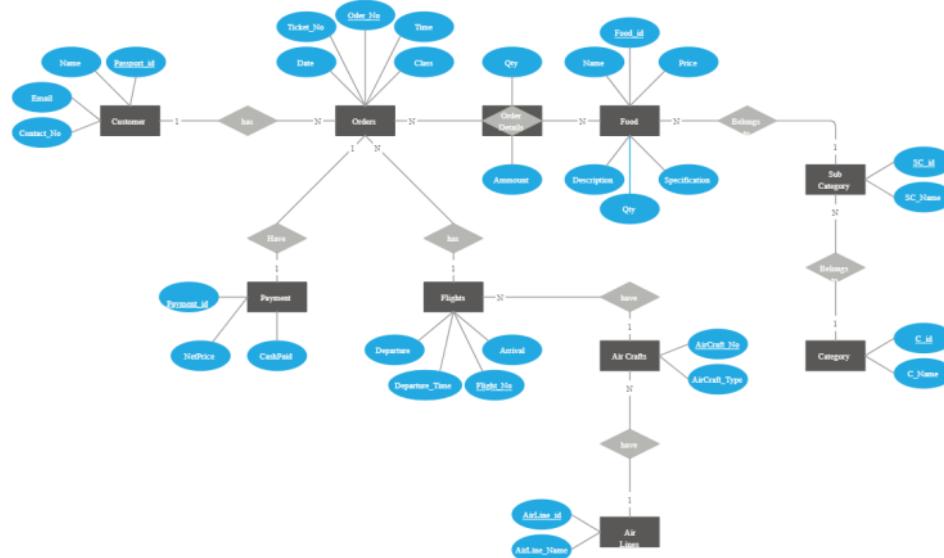


Figure 2: Entity Relationship Diagram

6

3.2 Use Case Diagram

Use cased diagram is a graphical representation of functionalities a certain type of users can perform. Different events of the online food ordering system are represented and connected with the accessible users who can perform them. It shows the relation between different actors and activities available in the system (Parker, 2009).

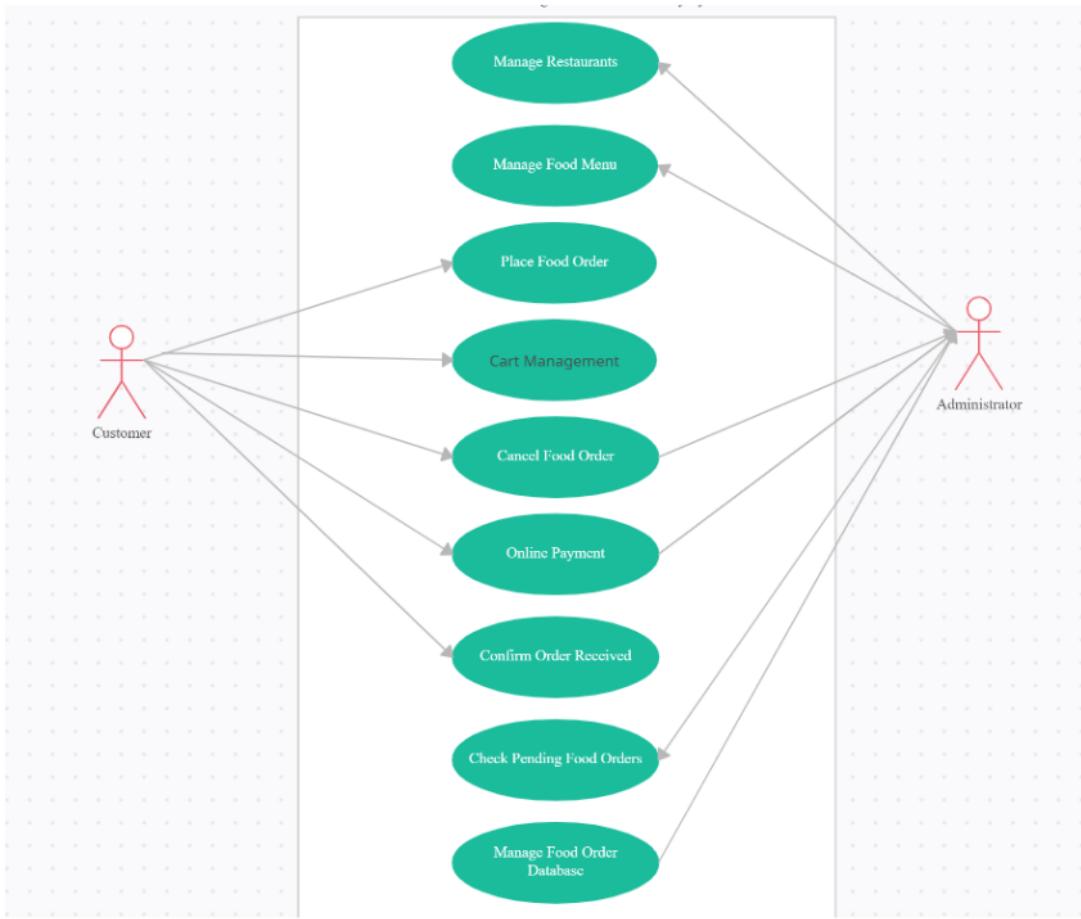


Figure 3: User Case Diagram

3.3 Database Schema

It refers to a blueprint or the ground level structure that defines the organization of data in the database system. The database schema shows the local organization of different tables and fields along with the constraints in the table which maps the relationship between different tables in the database (Cleve, 2015). Here, below a database schema related to Feast Flavor Online Delivery System is shown:

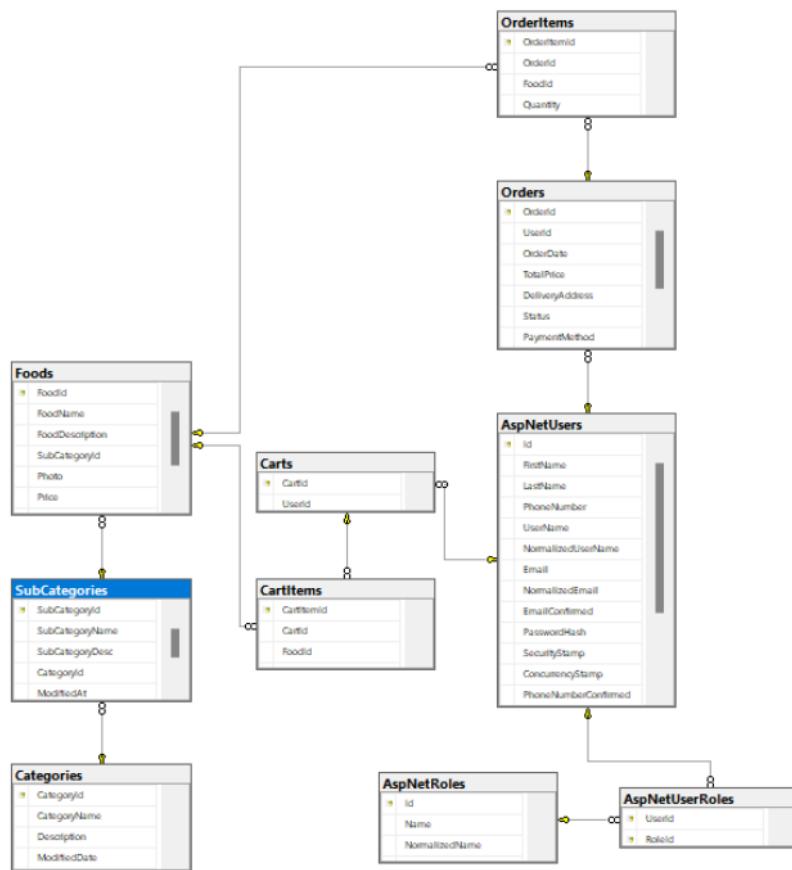


Figure 4: Database Schema

3.4 Interface Design

Home Page

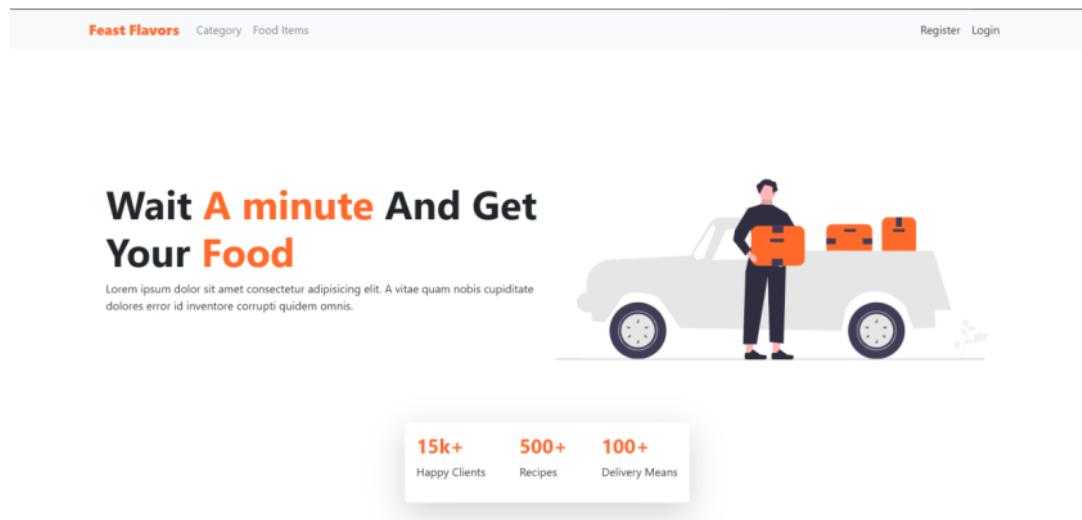


Figure 5: Home Page

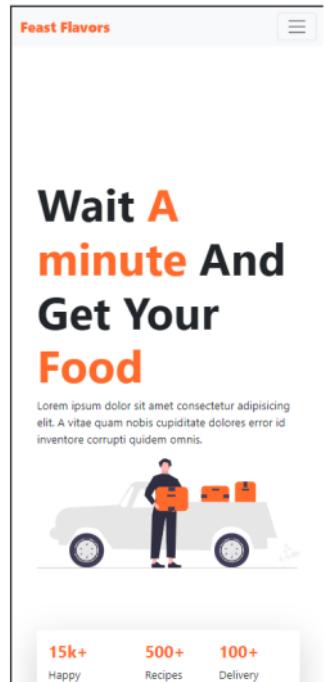


Figure 6 : Home Page Mobile View

A minimal home page where certain instructions related to the online food delivery system is provided. The styling is more focused on modern and minimal styling techniques.

User Registration Page:

The screenshot shows a user registration form titled "Create a new account." It includes fields for First Name, Last Name, Phone Number, Email, Password, and Confirm password. A large orange "Register" button is at the bottom, and a "Login" link is below it. The top navigation bar has links for "Feast Flavors", "Category", and "Food Items", along with "Register" and "Login" buttons.

Figure 7: User Registration Page

The screenshot shows the same user registration form as Figure 7, but it is displayed in a mobile-friendly layout with a vertical orientation. The fields and overall design are identical to the desktop version.

Figure 5:User Registration Page Mobile View

A simple registration page for users to fill up their personal details and register into the system.

Login Page

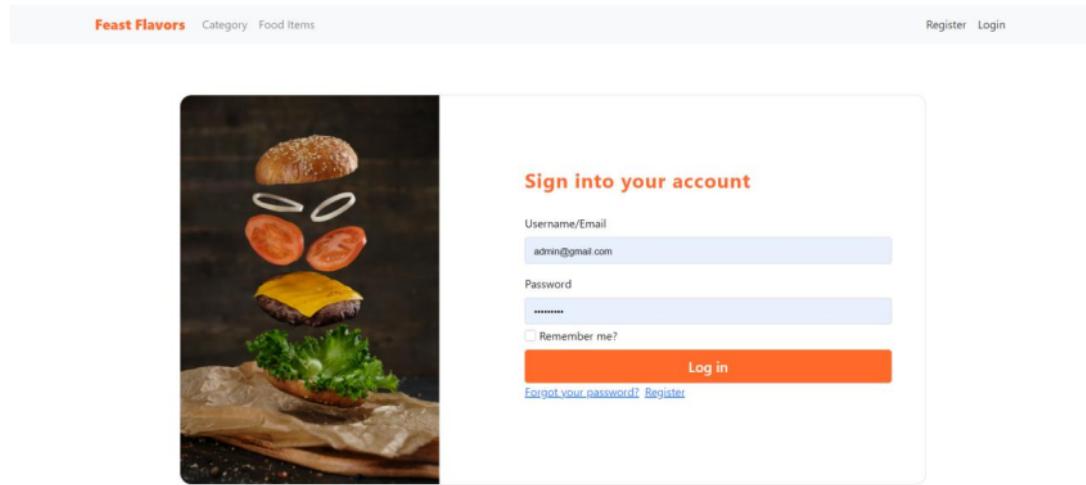


Figure 9: Login page

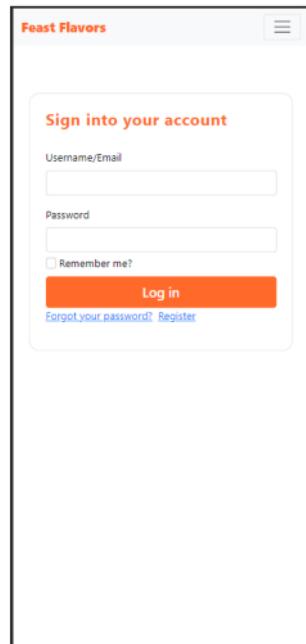


Figure 10: Login page Mobile View

A minimal login page with the use of bootstrap cards and performs along with all the validations in input for Admin and General User to login to the system.

User Categories View

The screenshot shows a user interface for a food delivery or ordering platform. At the top, there is a navigation bar with links for 'Feast Flavors', 'Category', 'Food Items', 'View Cart', and 'Order History'. On the right side of the header, there is a user profile with the email 'Hello surya@gmail.com!' and a 'Logout' button. Below the header, the title 'Categories' is displayed in orange. The main content area contains six cards, each representing a cuisine category:

- Italian Cuisine**: Foods from Italy. Includes a photo of various Italian dishes like pasta and pizza.
- Nepali Cuisine**: Authentic dishes from Nepal. Includes a photo of a traditional Nepali thali.
- English Cuisine**: Old Country based meals. Includes a photo of a roast dinner.
- French**: Bakery Items from France. Includes a photo of various breads and pastries.
- Spanish Cuisine**: A spicy blend of meats. Includes a photo of a seafood paella.
- Chinese**: Authentic dishes from China. Includes a photo of a variety of Chinese dishes.

Each card has a 'Details' button at the bottom.

Figure 6: Categories Page

This screenshot shows the same 'Categories' page but viewed on a mobile device. The layout is more compact. It displays two cards from the original six:

- Italian Cuisine**: Foods from Italy. Includes a photo of various Italian dishes like pasta and pizza.
- Nepali Cuisine**: Authentic dishes from Nepal. Includes a photo of a traditional Nepali thali.

Each card has a 'Details' button at the bottom. The mobile view also includes a navigation menu icon (three horizontal lines) and a 'Feast Flavors' header.

Figure 7: Categories Page Mobile View

A view of categories available in the system where user can browse through different cuisines and view details of the cuisines.

Sub-Category View

The screenshot shows a web-based application interface. At the top, there is a navigation bar with links for 'Feast Flavors' (highlighted in orange), 'Category', 'Food Items', 'View Cart', and 'Order History'. On the right side of the header, it displays the email 'Hello surya@gmail.com!' and a 'Logout' link. Below the header, the main content area has a title 'Sub-Categories' in orange. It features two cards side-by-side. The first card is for 'Pizza', showing a photo of various pizzas, with the name 'Pizza' in orange, a description 'A doughy spread with tons of seasonings', the cuisine 'Italian Cuisine', and a 'Details' button. The second card is for 'Pasta', showing a photo of pasta with broccoli, with the name 'Pasta' in orange, a description 'A traditional dish from Italy', the cuisine 'Italian Cuisine', and a 'Details' button.

Figure 8: Sub Category Page

This screenshot shows the same sub-category page as Figure 8, but it is displayed in a mobile-friendly layout. The content is identical to the desktop version, featuring two cards for 'Pizza' and 'Pasta' with their respective images, names, descriptions, cuisines, and 'Details' buttons. The mobile view includes a vertical menu icon on the left side of the main content area.

Figure 9: Sub-Category Page Mobile View

A detailed view page of categories which consist of the sub categories inside the selected categories. A simple card view is chosen for minimal look and feel in order to make the pages more user friendly.

Food Items View

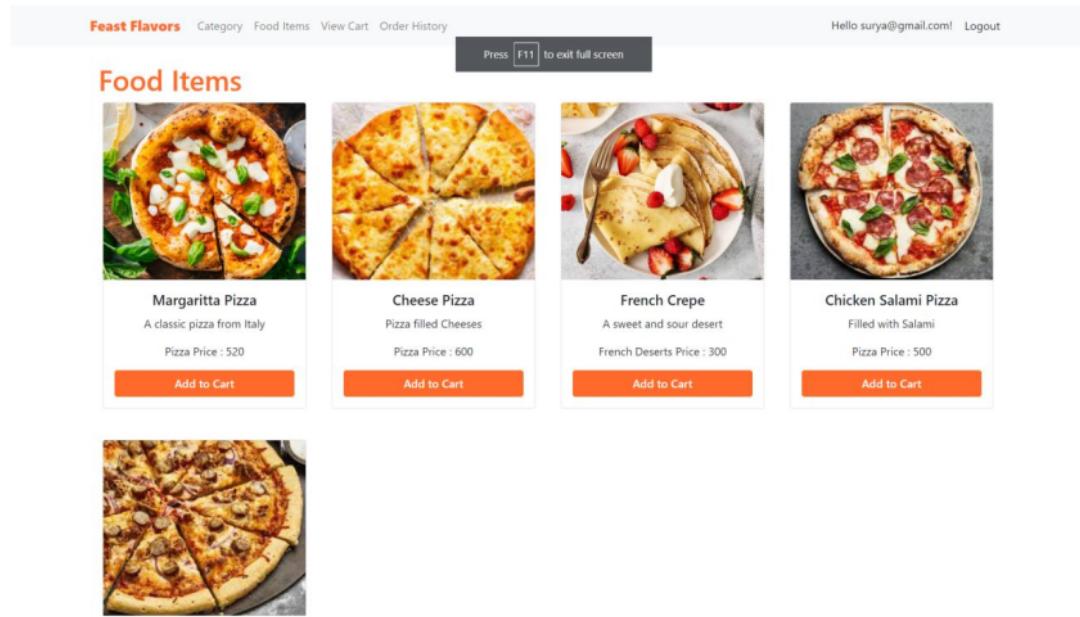


Figure 10: Food Items Page

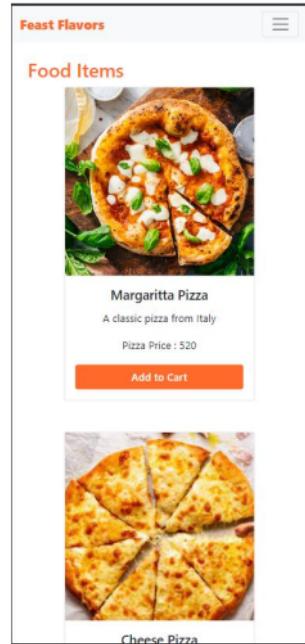


Figure 11: Food Items Page Mobile View

A general view of food items available in the system where users will be able to add certain items to the cart.

4

Food Item Detail Page

The screenshot shows a web page titled "Food Details" for "Margarita Pizza". At the top, there is a navigation bar with links for "Feast Flavors", "Category", "Food Items", "View Cart", and "Order History". On the right, it shows "Hello surya@gmail.com!" and "Logout". Below the navigation, the page title "Food Details" is displayed above the item name "Margarita Pizza". A large image of the pizza is shown on the left. To the right of the image, the "FoodDescription" is listed as "A classic pizza from Italy". Below this, a table provides detailed information about the food item:

FoodName	Margarita Pizza
SubCategory	A doughy spread with tons of seasonings
Price	520
Availability	Yes

Below the table, there is a quantity selector with a value of "2" and an "Add to Cart" button. The URL in the browser's address bar is "Missing "", ETS".

Figure 12: Food Items Detail Page

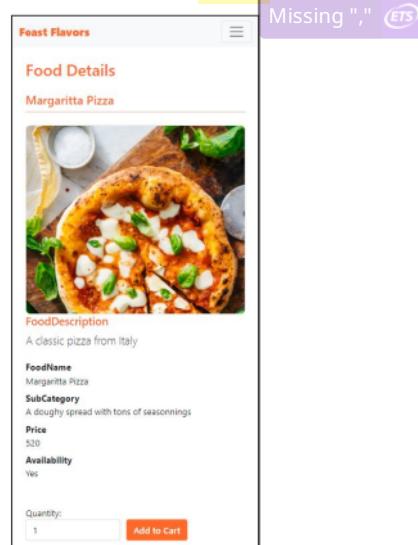


Figure 18: Food Item Detail Page Mobile View

A web page containing detailed information regarding the selected food item where users will be able to select quantity and add the item to their cart.

Cart Page

Food Name	Quantity	Price	Actions
Margarita Pizza	2	520	<button>Remove</button>
French Crepe	1	300	<button>Remove</button>

[Proceed to Payment](#)

Figure 13: Cart Page

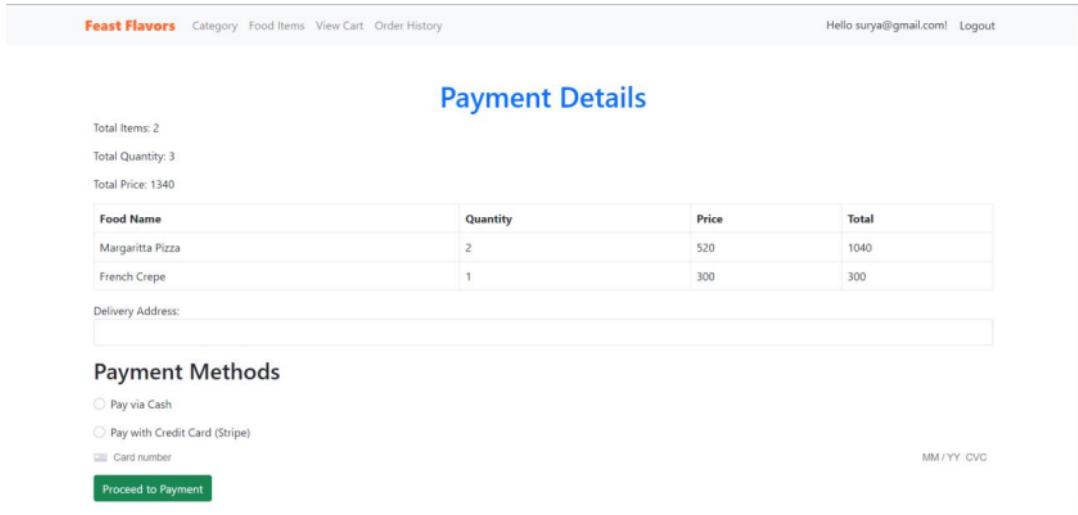
Food Name	Quantity	Price	Actions
Margarita Pizza	1	520	<button>Remove</button>

[Proceed to Payment](#)

Figure 14: Cart Page Mobile View

This is a simple cart page where users will be able to add or remove items in the cart. The functionality of increasing or decreasing the quantity of items is also providing here.

Payment Page



The screenshot shows a payment details page for a meal delivery service. At the top, there's a navigation bar with links for 'Category', 'Food Items', 'View Cart', and 'Order History'. On the right, it shows the user's email 'Hello surya@gmail.com!' and a 'Logout' link. The main content area has a title 'Payment Details'. Below it, there are summary statistics: 'Total Items: 2', 'Total Quantity: 3', and 'Total Price: 1340'. A table lists the food items with their quantities and prices:

Food Name	Quantity	Price	Total
Margarita Pizza	2	\$20	\$40
French Crepe	1	\$30	\$30

Below the table is a field for 'Delivery Address:' with a placeholder text 'Enter delivery address'. Under the title 'Payment Methods', there are three radio buttons: 'Pay via Cash', 'Pay with Credit Card (Stripe)', and 'Card number'. To the right of the card number input is a placeholder 'MM / YY CVC'. A green button labeled 'Proceed to Payment' is at the bottom.

Figure 15: Payment View

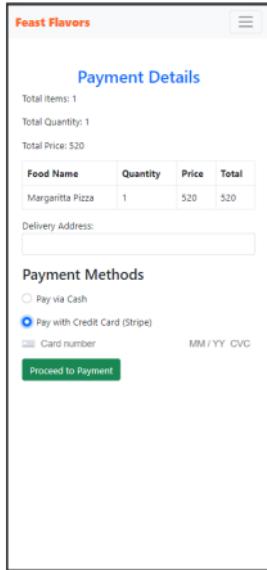


Figure 16: Payment Page Mobile View

This is the payment view page from where user can view the transaction related information where use will be able to add delivery address and add payment method. Here, payment option with Stripe API has been integrated to show the payment method.

3 Order History Page

Order Date	Total Price	Delivery Address	Status	Payment Method	Action	Feedback
2023-12-09 18:46:05	1340	Thali, Kathmandu	Pending	cash	Details	
2023-12-08 21:12:16	1560	Thali, Kathmandu	Delivered	stripe	Details	Provide Feedback

Figure 17: Order History Page

This is a **order history page** that contains details regarding **the orders** of the specific user. Users will be able to check the status of their order and view details of the order.

Feedback Page

Feast Flavors Category Food Items View Cart Order History

Hello surya@gmail.com! Logout

Provide Feedback

Order Date: 2023-12-08 21:12:16
 Total Price: 1560
 Delivery Address: Thali, Kathmandu
 Status: Delivered
 Payment Method: stripe

Feedback:

[Submit Feedback](#)

Figure 18: Feedback Page

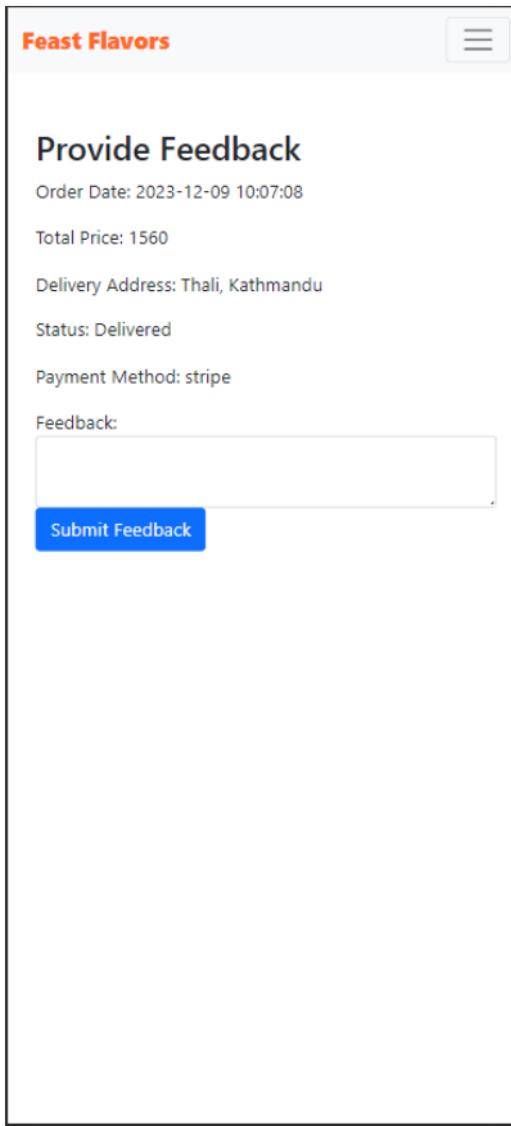


Figure 19: Feedback Page Mobile View

It is a simple page with a input field to provide feedback to the admins on specific orders.
Confused 
User Profile Management Sp. 

The screenshot shows a web application interface for managing account settings. At the top, there is a navigation bar with links for 'Feast Flavors', 'Category', 'Food Items', 'View Cart', and 'Order History'. On the right side of the bar, it says 'Hello sujan@gmail.com!' and has a 'Logout' link. Below the navigation bar, the main title is 'Manage your account' in orange, followed by a sub-section title 'Change your account settings'. To the left, there is a sidebar titled 'Account Navigation' with options: 'Profile' (which is highlighted in blue), 'Email', and 'Password'. The main content area is titled 'Profile' and contains fields for 'Username' (sujan@gmail.com) and 'Phone number' (9868331468). A large blue 'Save' button is at the bottom of this section.

Figure 20: Profile Management Page

Through this page, users can edit or update information regarding their account.

Admin View

Categories Management (CRUD)

The screenshot shows a table listing categories with columns: 'Category Name', 'Category Description', 'ModifiedDate', and 'Actions'. The 'Actions' column contains three buttons: 'Edit' (yellow), 'Details' (cyan), and 'Delete' (red). The table rows are as follows:

Category Name	Category Description	ModifiedDate	Actions
Italian Cuisine	Foods from Italy	12/5/2023 5:15:50 PM	Edit Details Delete
Nepali Cuisine	Authentic dishes from Nepal	12/5/2023 5:11:47 PM	Edit Details Delete
English Cuisine	Old Country based meals	12/5/2023 5:11:56 PM	Edit Details Delete
French	Bakery Items from France	12/5/2023 5:12:05 PM	Edit Details Delete
Spanish Cuisine	A spicy blend of meats	12/5/2023 5:12:24 PM	Edit Details Delete
Chinese	Authentic dishes from China	12/9/2023 8:40:24 AM	Edit Details Delete

Feast Flavors Category Food Items User Management Manage Orders Hello admin@gmail.com! Logout

Edit

Category Name

Description

Image
 No file chosen

Save [Back to List](#)

Feast Flavors Category Food Items User Management Manage Orders Hello admin@gmail.com! Logout

Create Category

Category Name

Description

Image
 No file chosen

Create [Back to List](#)

The screenshot shows the 'Italian Cuisine' category details. At the top, there's a header with 'Feast Flavors' and navigation links: Category, Food Items, User Management, Manage Orders, Hello admin@gmail.com!, and Logout. Below the header, the category name 'Italian Cuisine' is displayed in orange. A large image of various Italian dishes like pizza, pasta, and salads is shown. The 'Category Description' field contains 'Foods from italy'. A 'Subcategories' section includes a 'Create New Subcategory' button. A table lists subcategories: 'Pizza' (Description: 'A doughy spread with tons of seasonings') and 'Pasta' (Description: 'A traditional dish from Italy'). Each row has 'Edit', 'Details', and 'Delete' buttons.

The screenshot shows a confirmation dialog for deleting the 'Italian Cuisine' category. The title is 'Delete Category' with the sub-instruction 'Are you sure you want to delete this?'. Below is the 'Category Details' section, which includes the category name 'Italian Cuisine', description 'Foods from italy', and modified date '12/5/2023 5:15:50 PM'. At the bottom are 'Delete' and 'Back to List' buttons.

Figure 21: Category Management (CRUD)

These views and interface represents the CRUD functionality of Categories management for Admin where there are different pages in order to create, edit, view and delete categories to the system.

Sub-Categories (CRUD)

Feast Flavors
[Category](#)
[Food Items](#)
[User Management](#)
[Manage Orders](#)
Hello admin@gmail.com!
[Logout](#)

Italian Cuisine



Category Description

Foods from Italy

Subcategories

[Create New Subcategory](#)

Subcategory Name	Description	Actions
Pizza	A doughy spread with tons of seasonings	Edit Details Delete
Pasta	A traditional dish from Italy	Edit Details Delete

Feast Flavors
[Category](#)
[Food Items](#)
[User Management](#)
[Manage Orders](#)
Hello admin@gmail.com!
[Logout](#)

Edit SubCategory

SubCategory Name

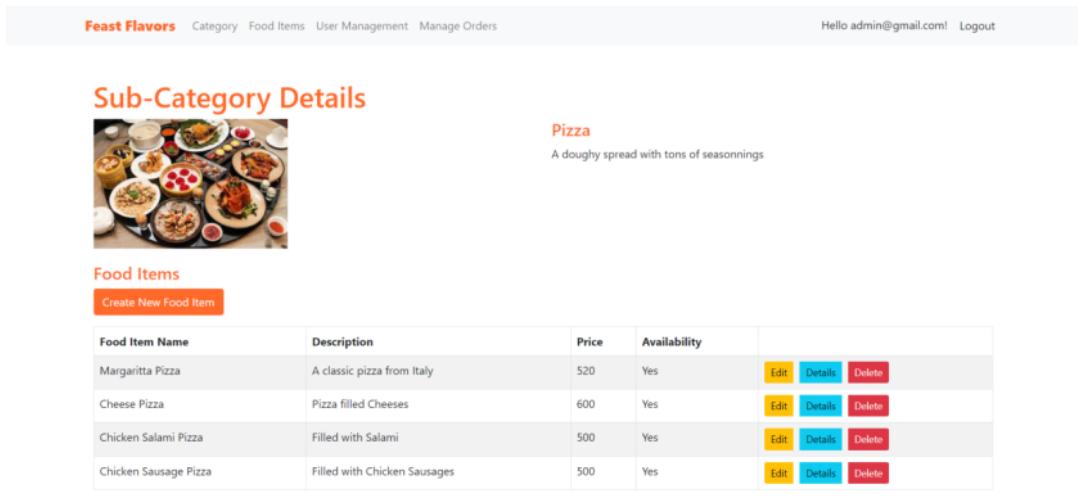
Description

Category

Image

Choose File

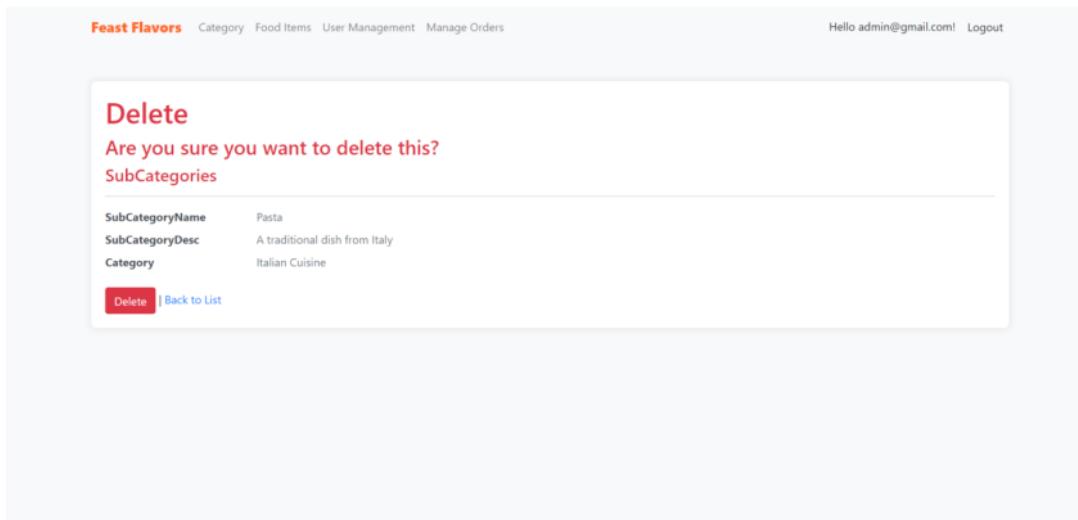
[Save](#) [Back to List](#)



Sub-Category Details

Pizza
A doughy spread with tons of seasonings

Food Item Name	Description	Price	Availability	Action
Margarita Pizza	A classic pizza from Italy	520	Yes	Edit Details Delete
Cheese Pizza	Pizza filled Cheeses	600	Yes	Edit Details Delete
Chicken Salami Pizza	Filled with Salami	500	Yes	Edit Details Delete
Chicken Sausage Pizza	Filled with Chicken Sausages	500	Yes	Edit Details Delete



Delete
Are you sure you want to delete this?

SubCategories

SubCategoryName	Pasta
SubCategoryDesc	A traditional dish from Italy
Category	Italian Cuisine

[Delete](#) | [Back to List](#)

Figure 22: Sub-Categories Management (CRUD)

The above **interfaces** represents **CRUD** functionalities that can be performed by admin on Sub-Categories.



Products/Food Items (CRUD)

Food Items				
Create New				
Food Name	Food Description	Sub Category	Price	Actions
Margarita Pizza	A classic pizza from Italy	Pizza	520	Edit Details Delete
Cheese Pizza	Pizza filled Cheeses	Pizza	600	Edit Details Delete
French Crepe	A sweet and sour desert	French Deserts	300	Edit Details Delete
Chicken Salami Pizza	Filled with Salami	Pizza	500	Edit Details Delete
Chicken Sausage Pizza	Filled with Chicken Sausages	Pizza	500	Edit Details Delete

Create Food				
Food Name	<input type="text"/>			
Description	<input type="text"/>			
SubCategory	<input type="text"/>			
Pizza	<input type="text"/>			
Food Photo	<input type="file"/> Choose File	No file chosen		
Price	<input type="text"/>			
	Create			
	Back to List			

The screenshot displays two pages of a web application for managing food items:

- Edit Food Page:**
 - Header: Feast Flavors, Category, Food Items, User Management, Manage Orders, Hello admin@gmail.com!, Logout.
 - Title: Edit Food
 - Form fields:
 - Food Name: Margarita Pizza
 - Description: A classic pizza from Italy
 - SubCategory: Pizza
 - Food Photo: Choose File [No file chosen]
 - Price: 520
 - Buttons: Save, Back to List
- Delete Page:**
 - Header: Feast Flavors, Category, Food Items, User Management, Manage Orders, Hello admin@gmail.com!, Logout.
 - Title: Delete
 - Text: Are you sure you want to delete this?
 - Section: Food
 - Table showing food details:

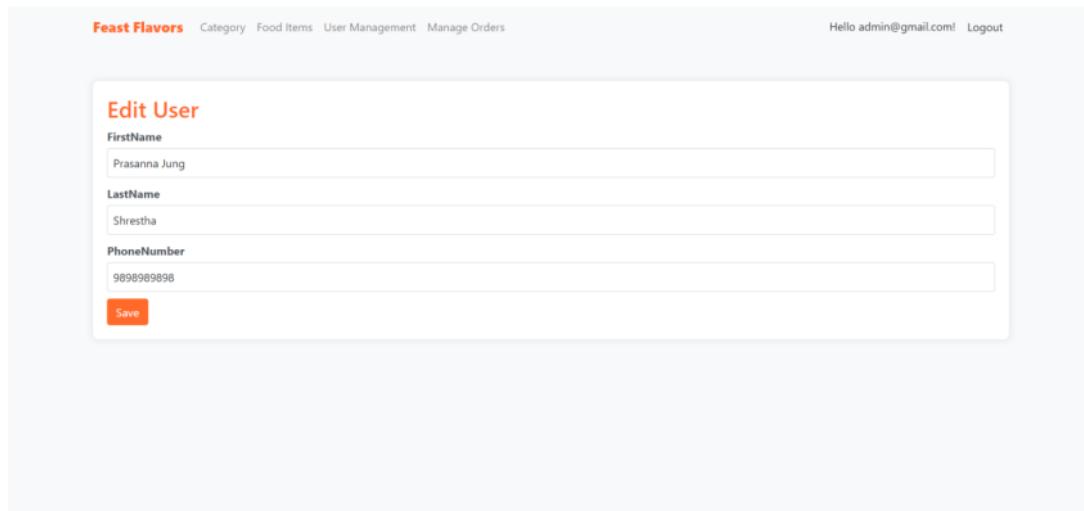
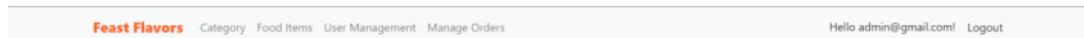
FoodName	Chicken Sausage Pizza
FoodDescription	Filled with Chicken Sausages
SubCategory	A doughy spread with tons of seasonings
Photo	/images/81e77998-6355-4e7d-ae3f-5e1acd870d44_sausage.jpeg
Price	500
Availability	Yes
ModifiedAt	12/5/2023 3:45:00 PM
 - Buttons: Delete, Back to List

Figure 23: Food Items Management (CRUD)

The above interfaces represents **CRUD** functionalities that can be performed by admin on Food Items.

S/V

User Management (RUD)



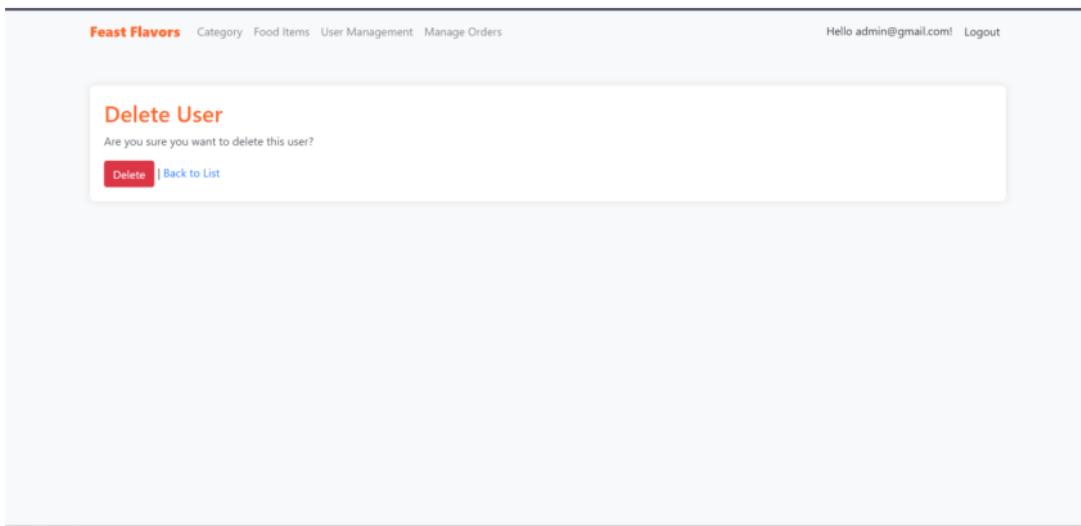


Figure 24: User Management (RUD)

The above ~~interfaces represents RUD~~ functionalities that can be performed by admin on User Management.

S/V (ETS)

Admin Order Management

The screenshot shows a web application interface for 'Feast Flavors'. At the top, there is a navigation bar with links for 'Category', 'Food Items', 'User Management', and 'Manage Orders'. On the right side of the header, it says 'Hello admin@gmail.com!' and has a 'Logout' link. Below the header, a section titled 'Manage Order Status' is shown. A table lists ten orders with columns for Order ID, User, Order Date, Total Price, Status, and Actions. Each row includes a 'View Details' button, a dropdown menu for status, and an 'Update' button. The table rows are numbered 2 through 1028.

Order ID	User	Order Date	Total Price	Status	Actions
2	Sujan Shrestha	2023-12-03 23:03:14	0	Delivered	<button>View Details</button> <button>Delivered</button> <button>Update</button>
4	Sujan Shrestha	2023-12-03 23:53:37	1120	Delivered	<button>View Details</button> <button>Delivered</button> <button>Update</button>
8	Sujan Shrestha	2023-12-05 10:06:39	1000	Delivered	<button>View Details</button> <button>Delivered</button> <button>Update</button>
1012	Sujan Shrestha	2023-12-08 10:23:28	2680	Delivered	<button>View Details</button> <button>Delivered</button> <button>Update</button>
1013	Prasanna Jung Shrestha	2023-12-08 11:24:42	1200	Delivered	<button>View Details</button> <button>Delivered</button> <button>Update</button>
1018	Sujan Shrestha	2023-12-08 16:12:09	1800	Pending Payment	<button>View Details</button> <button>Pending</button> <button>Update</button>
1019	Prasanna Jung Shrestha	2023-12-08 16:16:24	1120	Delivered	<button>View Details</button> <button>Delivered</button> <button>Update</button>
1024	Surya Shrestha	2023-12-08 21:12:16	1560	Delivered	<button>View Details</button> <button>Delivered</button> <button>Update</button>
1027	Sujan Shrestha	2023-12-09 10:07:08	1560	Delivered	<button>View Details</button> <button>Delivered</button> <button>Update</button>
1028	Surya Shrestha	2023-12-09 18:46:05	1340	Pending	<button>View Details</button> <button>Pending</button> <button>Update</button>

Figure 25: Order Status Management Page

The screenshot displays the 'Order Details' page. At the top, there is a header section with the title 'Order Details'. Below this, the 'Order Information' section contains the following data:

Order ID	4
User ID	58b64ecf-84f7-49b0-9209-956ed6f2a35a
Order Date	12/3/2023 11:53:37 PM
Total Price	1120
Status	Delivered
Delivery Address	Thali, Kathmandu
Payment Method	cod
Feedback	Great Service

Below the Order Information is the 'Order Items' section, which lists the items purchased:

Food Name	Quantity	Price
Margaritta Pizza	1	520
Cheese Pizza	1	600

Figure 26: Order Details Page

The screenshot shows a mobile application interface for an order details page. At the top, the title "Feast Flavors" is displayed in orange, followed by a three-line menu icon. Below the title, the section "Order Details" is shown in orange. Under "Order Information", the following details are listed:

- Order ID**: 1027
- User ID**: 58b64ecf-84f7-49b0-9209-956ed6f2a35a
- Order Date**: 12/9/2023 10:07:08 AM
- Total Price**: 1560
- Status**: Delivered
- Delivery Address**: Thali, Kathmandu
- Payment Method**: stripe

Below this, the section "Order Items" is shown in orange. It contains a table with the following data:

Food Name	Quantity	Price
Margaritta Pizza	3	520

At the bottom left, there is a "Back to List" button.

Figure 27: Order Details Page Mobile View

Missing "," (EFS)

4 Implementation

With all the designing process including different techniques such as Entity Relationship Diagram, Use Case Diagram and Database Schema and UI development. We move on towards the actual implementation or development of the project it self.

4.1 Factors related to the Design of the User Interface

User-centric Design:

Design the system on the basis of needs and expectations along with analysis of requirements of the user during the design process.

Responsive Layout

Ensure the adaptability of the user interface remains consistent with changes in the resolution of devices. Implementation of responsive design principles is taken into consideration.

Visual Hierarchy and Clarity:

Provide clear and consistent details to the elements of interface to guide user's attention and maintain logical flow of information to enhance the readability of the system.

Consistent Design:

A specific color scheme, layout and blueprint is maintained over the system considering the importance of typography and iconography of the system to maintain consistent design (Kim, 2020).

4.2 Factors related to the development and organization of the system for real-world deployment and its implications.

Scalability

The system should be implemented with ability to handle growth in terms of users, data and transactions in future so a selection of scalable technologies and architecture is important.

Security Concerns

A robust security measure should be considered to protect and prevent breach of user data and ensure the integrity of the transaction done in the system.

Data Backup and Recovery

In case of system failure or any other disruptions that can lead to data loss, a reliable mechanism for data recovery and backup should be created.

Performance monitoring and Optimization

In order to identify and address the bottlenecks and optimize resource usage of the system, certain monitoring tools can be used to track system metrics and conditions.

Cross browser and Cross platform compatibility

The system should comply to work with different web browsers and platforms for which a responsive design and test should be carried out on multiple platforms.

4.3 Steps of Implementation of system

Environment Setup

The chosen development environment with necessary tools and frameworks is .NET core with IDE Visual Studio of course. And a Microsoft SQL Server is chosen to handle all the database implementations.

Database Design and Implementation

After the environment setup, a database plan was created to design and implement a database required for the system with the help of above-mentioned designing tools (McLeod, 2011).

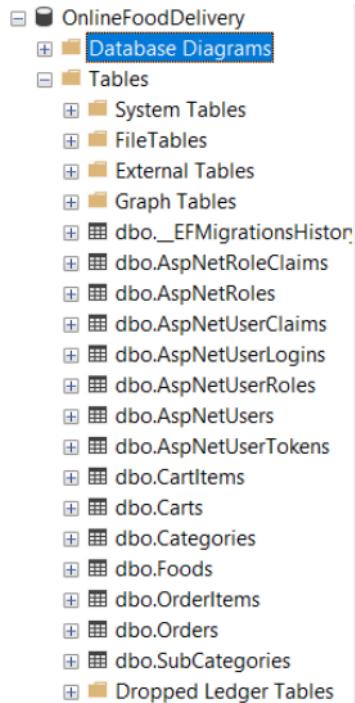


Figure 28: Database Tables

A database will all the necessary table is created for the development of the system.

This phase of the implementation includes development of server-side logic using C# with creation of certain models and mechanisms. Business logic including handling food items, categories, orders and other core functions are implemented in this phase. Here, below some snippets are provided where Model-View-Controller (MVC) architecture is used.

```
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Create([Bind("CategoryId,CategoryName,Description,CategoryImage")] Categories categories, IFormFile imageFile)
{
    if (ModelState.IsValid)
    {
        var webRootPath = _WebHostEnvironment.WebRootPath;
        Console.WriteLine("WebRootPath: " + webRootPath);

        // Handle image upload
        if (imageFile != null)
        {
            categories.CategoryImage = "/images/" + Guid.NewGuid() + "_" + imageFile.FileName;
            var imagePath = Path.Combine(_WebHostEnvironment.WebRootPath, "images", categories.CategoryImage.TrimStart('/'));

            //categories.CategoryImage = "/images/" + Guid.NewGuid() + "_" + imageFile.FileName;
            //var imagePath = Path.Combine(_WebHostEnvironment.WebRootPath, "images", categories.CategoryImage);

            Console.WriteLine("ImagePath: " + imagePath);

            using (var stream = new FileStream(imagePath, FileMode.Create))
            {
                await imageFile.CopyToAsync(stream);
            }
        }

        categories.ModifiedDate = DateTime.Now;
        _context.Add(categories);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    return View(categories);
}

[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Edit(int id, [Bind("FoodId,FoodName,FoodDescription,SubCategoryId,Photo,Price")] Food food, IFormFile photoFile)
{
    if (id != food.FoodId)
    {
        return NotFound();
    }

    if (ModelState.IsValid)
    {
        try
        {
            // Handle image upload
            if (photoFile != null)
            {
                //food.Photo = "/images/" + Guid.NewGuid() + "_" + imageFile.FileName;
                //var imagePath = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot", "images", food.Photo);

                food.Photo = "/images/" + Guid.NewGuid() + "_" + photoFile.FileName;
                var imagePath = Path.Combine(_WebHostEnvironment.WebRootPath, food.Photo.TrimStart('/'));

                using (var stream = new FileStream(imagePath, FileMode.Create))
                {
                    await photoFile.CopyToAsync(stream);
                }
            }
            food.ModifiedAt = DateTime.Now;
            _context.Update(food);
            await _context.SaveChangesAsync();
        }
        catch (DbUpdateConcurrencyException)
        {
            if (!FoodExists(food.FoodId))
            {
                return NotFound();
            }
            else
            {
                throw;
            }
        }
        return RedirectToAction(nameof(Index));
    }
    ViewData["SubCategoryId"] = new SelectList(_context.SubCategories, "SubCategoryId", "SubCategoryName", food.SubCategoryId);
    return View(food);
}
```

Article Error  Missing "" 

```

// Add item to the cart
[HttpPost]
0 references
public async Task<IActionResult> AddToCart(int foodId, int quantity)
{
    var user = await _userManager.GetUserAsync(user);

    // Retrieve or create a cart for the user
    var cart = _context.Carts
        .Include(c => c.CartItems)
        .ThenInclude(ci => ci.Food)
        .FirstOrDefault(c => c.UserId == user.Id);

    if (cart == null)
    {
        cart = new Cart { UserId = user.Id, CartItems = new List<CartItem>() };
        _context.Carts.Add(cart);
    }

    var food = await _context.Foods.FindAsync(foodId);

    if (food != null)
    {
        var cartItem = cart.CartItems?.FirstOrDefault(ci => ci.FoodId == foodId);

        if (cartItem != null)
        {
            // Update quantity if the item is already in the cart
            cartItem.Quantity += quantity;
        }
        else
        {
            // Add a new item to the cart
            cart.CartItems.Add(new CartItem { Food = food, Quantity = quantity });
        }
    }

    await _context.SaveChangesAsync();
}

return RedirectToAction("ViewCart");
}

0 references
public async Task<IActionResult> ProvideFeedback(int orderId)
{
    var order = await _context.Orders.FindAsync(orderId);

    if (order != null && order.Status == "Delivered")
    {
        return View(order);
    }
    else
    {
        // Handle invalid or non-delivered order
        return NotFound();
    }
}

[Authorize]
[HttpPost]
0 references
public async Task<IActionResult> SubmitFeedback(int orderId, string feedback)
{
    var order = await _context.Orders.FindAsync(orderId);

    if (order != null && order.Status == "Delivered")
    {
        order.Feedback = feedback;
        await _context.SaveChangesAsync();

        // Redirect to order details or another appropriate page
        return RedirectToAction("OrderDetails", new { id = orderId });
    }
    else
    {
        // Handle invalid or non-delivered order
        return NotFound();
    }
}

```

Figure 29: Backend Development Code Snippets

Frontend Development

The user interface of this system is developed through the use of HTML, CSS, JavaScript and

Jquery for certain functionalities. There is also the use of Bootstrap framework of CSS which is

Article Error

used to create responsive views for users across different devices. Some of the code snippets are provided below:

```
[model: OnLineFoodOrdering.Models.Entity.Categories]
[ViewData["Title"] = "Delete";
Layout = "~/Views/Shared/_Layout.cshtml";]

<div class="container mt-5">
    <h1>Delete Category</h1>

    <div>
        <h3>Are you sure you want to delete this?</h3>
        <div>
            <h4>Category Details:</h4>
            <hr />
            <dl class="row">
                <dt class="col-sm-2">
                    @Html.DisplayNameFor(model => model.CategoryName)
                </dt>
                <dd class="col-sm-10">
                    @Html.DisplayFor(model => model.CategoryName)
                </dd>
                <dt class="col-sm-2">
                    @Html.DisplayNameFor(model => model.Description)
                </dt>
                <dd class="col-sm-10">
                    @Html.DisplayFor(model => model.Description)
                </dd>
                <dt class="col-sm-2">
                    @Html.DisplayNameFor(model => model.ModifiedDate)
                </dt>
                <dd class="col-sm-10">
                    @Html.DisplayFor(model => model.ModifiedDate)
                </dd>
            </dl>
            <form asp-action="Delete">
                <input type="hidden" asp-for="CategoryId" />
                <div class="form-group">
                    <input type="submit" value="Delete" class="btn btn-danger" /> |
                    <a href="#" asp-action="Index" class="btn btn-secondary">Back to List</a>
                </div>
            </form>
        </div>
    </div>
</div>

<div class="container mt-3">
    <h2 style="color: #0070C0;">Food Items</h2>
    <@model OnLineFoodOrdering.Models.Entity.FoodItems>
    <@{
        var actionCreate = ViewData["Info-primary"] == "Create New" ? "Create" : "Edit";
    }>
    <div class="table-responsive">
        <table border="1" class="table table-striped table-bordered">
            <thead>
                <tr>
                    <th>Food Name</th>
                    <th>Food Description</th>
                    <th>Sub Category</th>
                    <th>Initial Availability</th>
                    <th>Actions</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>@foreach (var item in Model)<br/>
                        @item.FoodName</td>
                    <td>@item.FoodDescription</td>
                    <td>@item.SubCategory.SubCategoryName</td>
                    <td>@item.InitialAvailability</td>
                    <td>
                        @if (item.InitialAvailability != null)
                        {
                            <button type="button" class="btn btn-primary" style="background-color: #0070C0; color: white; border-radius: 50%; padding: 5px 10px; margin-right: 10px;">Edit</button>
                            <button type="button" class="btn btn-warning" style="background-color: #FFB703; color: white; border-radius: 50%; padding: 5px 10px;">Delete</button>
                        }
                    </td>
                </tr>
            </tbody>
        </table>
    </div>
    <div class="row rounded-4 justify-content-end">
        <div class="col-12 col-md-4">
            <@* Search Form Roots in Model *@
            <div class="input-group">
                <input type="text" class="form-control" placeholder="Search food items..." style="border: none; border-radius: 50%; padding: 10px 15px; width: 100%; height: 40px; font-size: 14px; border-bottom: 1px solid #ccc; border-left: none; border-right: none; margin-bottom: 10px;" />
                <span class="input-group-append" style="border: none; border-radius: 50%; background-color: #0070C0; color: white; padding: 5px 10px; font-weight: bold; font-size: 14px; border-left: 1px solid #ccc; border-top: 1px solid #ccc; border-bottom: 1px solid #ccc; border-right: none; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin-left: -10px;">
                    <img alt="Search icon" style="width: 100%; height: 100%; object-fit: cover; border-radius: 50%; border: 1px solid #ccc; border-left: none; border-top: 1px solid #ccc; border-bottom: 1px solid #ccc; border-right: none; margin: auto;"/>
                </span>
            </div>
        </div>
    </div>
</div>
```

```

<div class="container">
  <@model IModelContainer<Model>>
  <div class="row">
    <div class="col-md-12">
      <@model DisplayFor<Model> > model.CategoryImage> </div>
      <div class="col-md-12">
        <div style="max-height: 300px; overflow: hidden;">
          <img alt="Category Image" class="img-fluid w-100 h-100" />
        </div>
      </div>
      <div class="col-md-6">
        <@model DisplayFor<Model> > model.Description> </div>
        <div class="col-md-6">
          <@model DisplayFor<Model> > model.Description> </div>
        </div>
      </div>
    </div>
  </div>
  <div>
    <h4>SubCategories </h4>
    <p> <a href="#" asp-action="Create" asp-route-model="SubCategories" asp-route-categoryId="@Model.CategoryId" class="btn btn-primary">Create New Subcategory </a> </p>
    <table class="table table-striped table-bordered">
      <thead>
        <tr>
          <th>Subcategory Name </th>
          <th>Description </th>
          <th>Actions </th>
        </tr>
      </thead>
      <tbody>
        <@foreach (var subcategory in Model.SubCategories)>
          <tr>
            <td> <@Model.DisplayFor<Model> > subcategory.SubCategoryName> </td>
            <td> <@Model.DisplayFor<Model> > subcategory.SubCategoryDetails> </td>
            <td>
              <div class="btn-group" style="gap: 10px; role="group">
                <a href="#" asp-action="Edit" asp-route-model="SubCategories" asp-route-categoryId="@Model.CategoryId" asp-route-subCategoryId="@subcategory.SubCategoryId" class="btn btn-warning btn-sm">Edit </a>
                <a href="#" asp-action="Details" asp-route-model="SubCategories" asp-route-categoryId="@Model.CategoryId" asp-route-subCategoryId="@subcategory.SubCategoryId" class="btn btn-info btn-sm">Details </a>
                <a href="#" asp-action="Delete" asp-route-model="SubCategories" asp-route-categoryId="@Model.CategoryId" asp-route-subCategoryId="@subcategory.SubCategoryId" class="btn btn-danger btn-sm">Delete </a>
              </div>
            </td>
          </tr>
        <@endforeach>
      </tbody>
    </table>
  </div>
</div>
<div class="container mt-2">
  <h3 style="color: #FF602D;"> Sub-Categories </h3>
  <div class="row gy-4 gx-5" style="gap: 20px; justify-content: space-between; margin-bottom: 20px; border-bottom: 1px solid #ccc; padding-bottom: 10px; position: relative; z-index: 1; >
    <@foreach (var subcategory in Model.SubCategories)>
      <div class="card" style="width: 180px; position: relative; z-index: 1; >
        <div class="card-image" style="background-size: cover; background-position: center; height: 100%; width: 100%; >
          <img alt="Subcategory Image" class="card-img-top" style="width: 100%; object-fit: cover; >
        </div>
        <div class="card-body text-center" style="position: absolute; bottom: 0; left: 0; width: 100%; height: 100%; >
          <h3 class="card-title" style="margin: 0; font-size: 1em; font-weight: bold; color: #FF602D;"> @subcategory.SubCategoryName </h3>
          <p class="card-text" style="margin: 0; font-size: 0.8em; color: #FF602D;"> @subcategory.SubCategoryDetails </p>
          <@if (User.IsInRole("Admin"))>
            <a href="#" asp-action="Edit" asp-route-model="SubCategories" asp-route-categoryId="@Model.CategoryId" asp-route-subCategoryId="@subcategory.SubCategoryId" style="background-color: #FF602D; font-weight: 900; color: white; text-decoration: none; border-radius: 5px; padding: 5px 10px; margin: 5px 0; >
              Article Error <span style="font-size: 0.8em; margin-left: 5px;"> ETS </span>
            </a>
          <@endif>
        </div>
      </div>
    <@endforeach>
  </div>
</div>

```

Figure 30: Front-end Development Code Snippets

User Authentication and Authorization

For the implementation of the Authentication and authorization, the identity extension of the .NET environment is used. The roles among the normal user and admin user is also setup with Identity package. From sign in to the log in every mechanism is implemented using identity package.

Article Error 

```

microFoodOrdering

using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Identity;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using System;
using System.Threading.Tasks;

public class SeedData
{
    public static async void Initialize(IApplicationBuilder app)
    {
        using (var scope = app.ApplicationServices.CreateScope())
        {
            var serviceProvider = scope.ServiceProvider;
            try
            {
                var userManager = serviceProvider.GetRequiredService<UserManager<ApplicationUser>>();
                var roleManager = serviceProvider.GetRequiredService<RoleManager<IdentityRole>>();

                await SeedRoles(roleManager);
                await SeedUsers(userManager);
            }
            catch (Exception ex)
            {
                // Log the error
            }
        }
    }

    private static async Task SeedRoles(RoleManager<IdentityRole> roleManager)
    {
        string[] roleNames = { "Admin", "Customer" };
        foreach (var roleName in roleNames)
        {
            var roleExist = await roleManager.RoleExistsAsync(roleName);
            if (!roleExist)
            {
                // Create the roles and seed them to the database
                await roleManager.CreateAsync(new IdentityRole(roleName));
            }
        }
    }

    private static async Task SeedUsers(UserManager<ApplicationUser> userManager)
    {
        // You can add your logic to create and seed users here if needed
        // For example:
        var adminUser = new ApplicationUser()
        {
            UserName = "admin@example.com",
            Email = "admin@example.com",
            FirstName = "admin",
            LastName = "admin",
            PhoneNumber = "9999999999",
            // Add other properties as needed
        };

        var result = await userManager.CreateAsync(adminUser, "AdminPassword123");
        if (result.Succeeded)
        {
            // Assign the admin user to the "Admin" role
            await userManager.AddToRoleAsync(adminUser, "Admin");
        }
    }
}

Index.cshtml

@using Microsoft.AspNetCore.Identity
@using DelinieFoodOrdering.Models.Entity
@inject SignInManager<ApplicationUser> SignInManager
@inject UserManager<ApplicationUser> UserManager




@if (SignInManager.IsSignedIn(User))
{
    <li class="nav-item">
        <a id="Manage" class="nav-link text-dark" asp-area="Identity" asp-page="/Account/Manage/Index" title="Manage">Hello @UserManager.GetUserName(User)!</a>
    </li>
    <li class="nav-item">
        <form id="logoutForm" class="form-inline" asp-area="Identity" asp-page="/Account/Logout" asp-route-returnUrl="@Url.Action("Index", "Home", new { area = "" })">
            <button id="logout" type="submit" class="nav-link btn btn-link text-dark">Logout</button>
        </form>
    </li>
}
else
{
    <li class="nav-item">
        <a class="nav-link text-dark" id="register" asp-area="Identity" asp-page="/Account/Register">Register</a>
    </li>
    <li class="nav-item">
        <a class="nav-link text-dark" id="login" asp-area="Identity" asp-page="/Account/Login">Login</a>
    </li>
}

```

Figure 31: Identity Authentication and Authorization Code snippets

Payment Integration (Stripe)

An external online payment mechanism is used in the system named Stripe which allows users to pay via their credit card and place the order in the application. For this API integration is done with Stripe secret key and use of Stripe.Net extension. Here are the code snippets of this implementation:

```

{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "DefaultConnection": "Server=LAPTOP-S681NM7J;Database=OnlineFoodDelivery;Trusted_Connection=True;MultipleActiveResultSets=True"
  },
  "Stripe": {
    "PublicKey": "pk_test_51OL0NCJ0Q7G03LnRuPjsC6QYGynyXAHYSjzlegyR6mGxIuGoJnyKSRZ0BLX5K8p3N2oF6TTEvCxqsUMvUH3kzX80QdcG0eKd",
    "SecretKey": "sk_test_51OL0NCJ0Q7G03LnRrSAFL9tOZbUoApj0Xw60rzFxAcvuUQLrVles90XQLK7irkVAUEtovKru07dNLcgToedJcap00rgryGRyM"
  }
}

builder.Services.Configure<StripeSettings>(builder.Configuration.GetSection("Stripe"));

[HttpPost]
public async Task<ActionResult> PlaceOrder(string paymentMethod, string deliveryAddress)
{
  if (ModelState.IsValid)
  {
    var user = await userManager.GetUserAsync(User);
    var cart = context.Carts
      .Include(c => c.CartItems)
      .ThenInclude(ci => ci.Food)
      .FirstOrDefault(c => c.UserId == user.Id);

    if (cart != null && cart.CartItems != null && cart.CartItems.Any())
    {
      var totalPrice = cart.CartItems.Sum(ci => ci.Food?.Price * ci.Quantity ?? 0);

      var order = new Order
      {
        UserId = user.Id,
        OrderDate = DateTime.Now,
        TotalPrice = totalPrice,
        DeliveryAddress = deliveryAddress,
        Status = "Pending",
        PaymentMethod = paymentMethod,
        OrderItems = cart.CartItems.Select(ci => new OrderItem
        {
          FoodId = ci.FoodId,
          Quantity = ci.Quantity,
          Price = ci.Food?.Price ?? 0
        }).ToList(),
      };

      if (paymentMethod.Equals("stripe", StringComparison.OrdinalIgnoreCase))
      {
        // Set up Stripe API Key
        StripeConfiguration.ApiKey = _stripeSettings.SecretKey;

        // Create a PaymentIntent with return_url
        var options = new PaymentIntentCreateOptions
        {
          Amount = (long)totalPrice * 100, // Amount in cents
          Currency = "usd",
          PaymentMethod = "pm_card_visa", // You should obtain a PaymentMethod ID using Stripe.js
          ConfirmationMethod = "manual",
          Confirm = true,
          ReturnUrl = Url.Action("OrderHistory", "Order", null, Request.Scheme), // Redirect to OrderHistory
        };

        var service = new PaymentIntentService();
        var paymentIntent = service.Create(options);

        // Update order with Stripe PaymentIntent ID
        order.StripePaymentIntentId = paymentIntent.Id;

        // Set the order status to "Pending Payment" initially
        order.Status = "Pending";
      }

      _context.Orders.Add(order);

      // Clear the cart after placing the order
      _context.Carts.Remove(cart);

      await _context.SaveChangesAsync();

      // Display a success message using TempData
      TempData["SuccessMessage"] = "Order placed successfully!";
    }

    // Redirect to the order history page
  }
}

```

Figure 32: Stripe Payment Integration Code Snippets

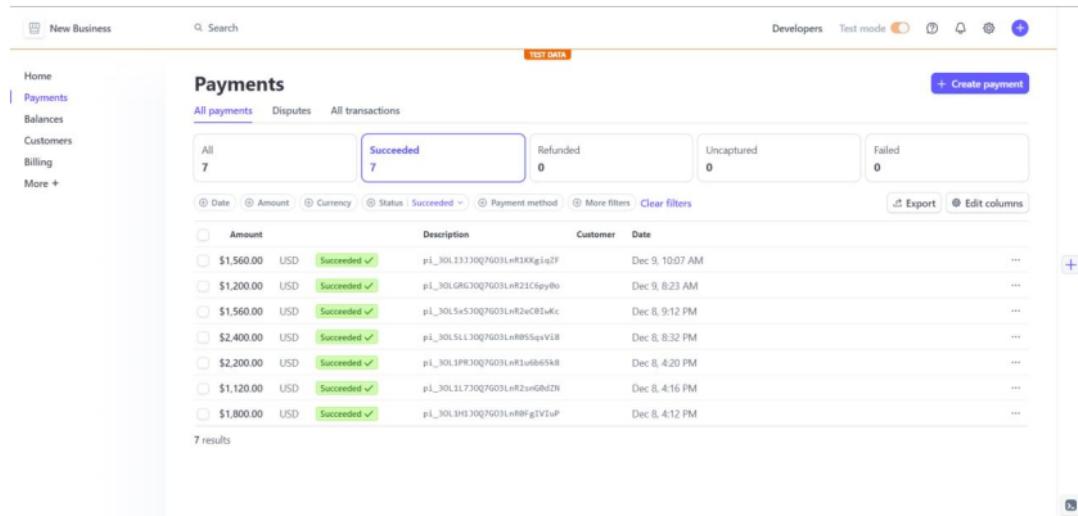


Figure 33: Stripe Dashboard View

5 Conclusion

In conclusion, Feast Flavors, a online food delivery platform successfully implements a user friendly interface along with robust functionalities. The project has achieved and met all the requirements and goals by creation of intuitive interface, secure backend process implementation and considerations of different factors of designing, development, and deployment. This online food delivery system depicts a commitment to provide a seamless and efficient experience on both ends of the administration and user.

6 References

- Cagiltay, N. E. (2013, August 8). Performing and analyzing non-formal inspections of entity relationship diagram. *Journal of Systems and Software*, 2184-2195.
doi:<https://www.sciencedirect.com/science/article/abs/pii/S0164121213001015>
- Cleve, A. (2015). Understanding database schema evolution. *Science of Computer Programming*.
doi:<https://doi.org/10.1016/j.scico.2013.11.025>
- Kim, J. (2020). Designing emotionally evocative homepages. *mpirical study of the quantitative relations between design factors and emotional dimensions.*
doi:<https://doi.org/10.1016/j.ijhcs.2003.06.002>
- McLeod, L. (2011, October 18). Factors that affect software systems development project outcomes. 1–56. doi:<https://doi.org/10.1145/1978802.1978803>
- Parker. (2009). Use case diagrams in support of use case modeling. *Journal of Database Management*, 1-24. Retrieved December 8, 2023, from <https://circle.visual-paradigm.com/docs/uml-and-sysml/use-case-diagram/>
- Wiley, J. (2012). System Design and Practice. Retrieved December 8, 2023, from https://books.google.com.np/books?hl=en&lr=&id=w8k3aXBnDP4C&oi=fnd&pg=PP8&dq=system+design&ots=M5Fi33zQd1&sig=nnZRXSomeLPMj5Un4DW993UdEao&rdir_esc=y#v=onepage&q=system%20design&f=false

7 Appendices

7.1 User Manual

1. Getting Started

Accessing the System

User Registration

Login

2. User Profile Management

Viewing and Editing Profile Information

Changing Password

3. Browsing and Ordering

Exploring the Product Catalog

Adding Items to the Shopping Cart

Viewing and Managing the Shopping Cart

Placing an Order

4. Order Tracking

Checking Pending Orders

Viewing Order History

Tracking Delivery Status

5. Feedback and Ratings

Providing Feedback on Products and Transactions

Rating System

6. Administrator Functions

Managing User Accounts

Adding and Updating Products

Managing Product Categories

Monitoring and Managing Orders

7. Security and Privacy

User Authentication

Data Encryption

ORIGINALITY REPORT



PRIMARY SOURCES

1	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	2%
2	Submitted to Sunway Education Group Student Paper	1%
3	Submitted to Wawasan Open University Student Paper	1%
4	Submitted to University of Plymouth Student Paper	1%
5	www.ukessays.com Internet Source	<1%
6	ellamleiamuslimah.blogspot.com Internet Source	<1%
7	dspace.nm-aist.ac.tz Internet Source	<1%

Exclude quotes On

Exclude matches Off

Exclude bibliography On

NP000611-CT081-3-3-MWM-Individual-Assignment.docx

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5



P/V You have used the passive voice in this sentence. You may want to revise it using the active voice.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



Article Error You may need to remove this article.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



Article Error You may need to use an article before this word.



P/V You have used the passive voice in this sentence. You may want to revise it using the active voice.



Article Error You may need to use an article before this word.

PAGE 6



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

PAGE 7



Missing "," Review the rules for using punctuation marks.



Article Error You may need to use an article before this word. Consider using the article **the**.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



Article Error You may need to remove this article.



Prep. You may be using the wrong preposition.



Article Error You may need to use an article before this word.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



S/V This subject and verb may not agree. Proofread the sentence to make sure the subject agrees with the verb.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

PAGE 8



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



Missing ","



S/V This subject and verb may not agree. Proofread the sentence to make sure the subject agrees with the verb.



Missing "," Review the rules for using punctuation marks.

PAGE 9



Missing "," Review the rules for using punctuation marks.



Article Error You may need to remove this article.

PAGE 10



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15



Article Error You may need to remove this article.

PAGE 16

PAGE 17



Missing "," Review the rules for using punctuation marks.

PAGE 18



Missing "," Review the rules for using punctuation marks.



Missing "," Review the rules for using punctuation marks.

PAGE 19



Article Error You may need to use an article before this word.

PAGE 20



Confused You have used either an imprecise word or an incorrect word.



Proper Nouns You may need to use a capital letter for this proper noun.

PAGE 21



Confused You have used either an imprecise word or an incorrect word.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26



S/V This subject and verb may not agree. Proofread the sentence to make sure the subject agrees with the verb.

PAGE 27

PAGE 28



S/V This subject and verb may not agree. Proofread the sentence to make sure the subject agrees with the verb.

PAGE 29

PAGE 30



S/V This subject and verb may not agree. Proofread the sentence to make sure the subject agrees with the verb.

PAGE 31

PAGE 32



Missing "," Review the rules for using punctuation marks.

PAGE 33



Missing "," Review the rules for using punctuation marks.



Prep. You may be using the wrong preposition.



Compound These two words should be a compound word.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



S/V This subject and verb may not agree. Proofread the sentence to make sure the subject agrees with the verb.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



Article Error You may need to use an article before this word.



Article Error You may need to use an article before this word.



Article Error You may need to use an article before this word. Consider using the article **the**.



Article Error You may need to use an article before this word.



Article Error You may need to use an article before this word.



Missing "," Review the rules for using punctuation marks.



Missing "," Review the rules for using punctuation marks.

PAGE 34



Article Error You may need to use an article before this word.



Article Error You may need to use an article before this word.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

PAGE 35



Article Error You may need to use an article before this word. Consider using the article **the**.



Missing "," Review the rules for using punctuation marks.



Missing "," Review the rules for using punctuation marks.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



Missing "," Review the rules for using punctuation marks.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



Article Error You may need to use an article before this word.



Missing ","



Article Error You may need to use an article before this word.



Article Error You may need to use an article before this word.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



Article Error You may need to remove this article.



Confused You have used either an imprecise word or an incorrect word.



Article Error You may need to use an article before this word. Consider using the article **the**.



Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

PAGE 42

PAGE 43