**Online Food Ordering System**

A Comprehensive Project Report

By

Pasupulati Sai Meghana

**Acknowledgment :-**

This project, **"Online Food Ordering System"**, was developed as part of my internship program with **Infosys Springboard**, an innovative learning platform.

**About Infosys Springboard :-**

Infosys Springboard is a leading-edge initiative by Infosys, providing learners with access to world-class resources, mentorship, and industry-relevant training. The platform emphasizes practical learning through real-world projects, fostering skill development in emerging technologies.

**My Internship Experience :-**

Through this internship, I gained invaluable insights into full-stack development, team collaboration, and modern software engineering practices. The project encapsulates the knowledge and skills I have acquired, including front-end and back-end development using Django, SQLite3, and interactive web technologies.

**Abstract :-**

This report presents the development and implementation of an **Online Food Ordering System**, a web-based application designed to streamline the process of ordering food from various restaurants. The system caters to three primary user roles: **Customer**, **Restaurant**, and **Delivery Personnel**, each equipped with distinct functionalities tailored to their needs.

The project aims to enhance user convenience by offering features such as menu browsing, cart management, order tracking, and role-specific dashboards. Built using **Django**, **SQLite3**, **HTML**, **CSS**, and **JavaScript**, the system emphasizes modularity, scalability, and user-friendly interfaces.

Key highlights of this project include:

* **Customer Experience**: Easy navigation, food item selection, and secure purchasing options.
* **Restaurant Management**: Efficient order handling and menu updates.
* **Delivery Optimization**: Streamlined order delivery status and logistics tracking.

This project demonstrates the integration of modern web development principles and frameworks, highlighting my technical expertise and problem-solving abilities gained through the **Infosys Springboard** internship program.

**Table Of Contents :-**

[Introduction :- 4](#_Toc1861930955)

[Problem Statement :- 5](#_Toc80110384)

[Objectives :- 5](#_Toc1895901193)

[Architecture Overview :- 6](#_Toc917862059)

[Work Flow :- 6](#_Toc7264663)

[Database Design :- 7](#_Toc2037983394)

[Technologies Used 8](#_Toc1141690581)

[Key Modules / Apps 9](#_Toc1962022022)

[Commands used 10](#_Toc429902979)

[Website images :- 11](#_Toc340959219)

[Future scope :- 20](#_Toc1104388591)

[Conclusion 21](#_Toc1448219490)

# **Introduction :-**

In the digital age, online food ordering systems have revolutionized the way people interact with restaurants and delivery services. These platforms enable customers to order their favorite meals with just a few clicks, while also allowing restaurants and delivery personnel to manage their operations more efficiently.

The **Online Food Ordering System** project is a web-based application designed to address the needs of three key user roles: **Customers**, **Restaurants**, and **Delivery Personnel**. This system bridges the gap between diners and food providers by offering seamless interactions, real-time updates, and convenient order management features.

# Problem Statement :-

The traditional process of ordering food often involves inefficiencies, such as long wait times, miscommunication, and a lack of tracking options. This project aims to overcome these challenges by introducing a unified platform where all stakeholders can collaborate effectively.

# Objectives :-

* **For Customers**: Provide an intuitive interface for exploring menus, placing orders, and tracking deliveries.
* **For Restaurants**: Offer tools for managing orders, updating menus, and viewing customer feedback.
* **For Delivery Personnel**: Streamline delivery assignments and provide real-time delivery status updates.

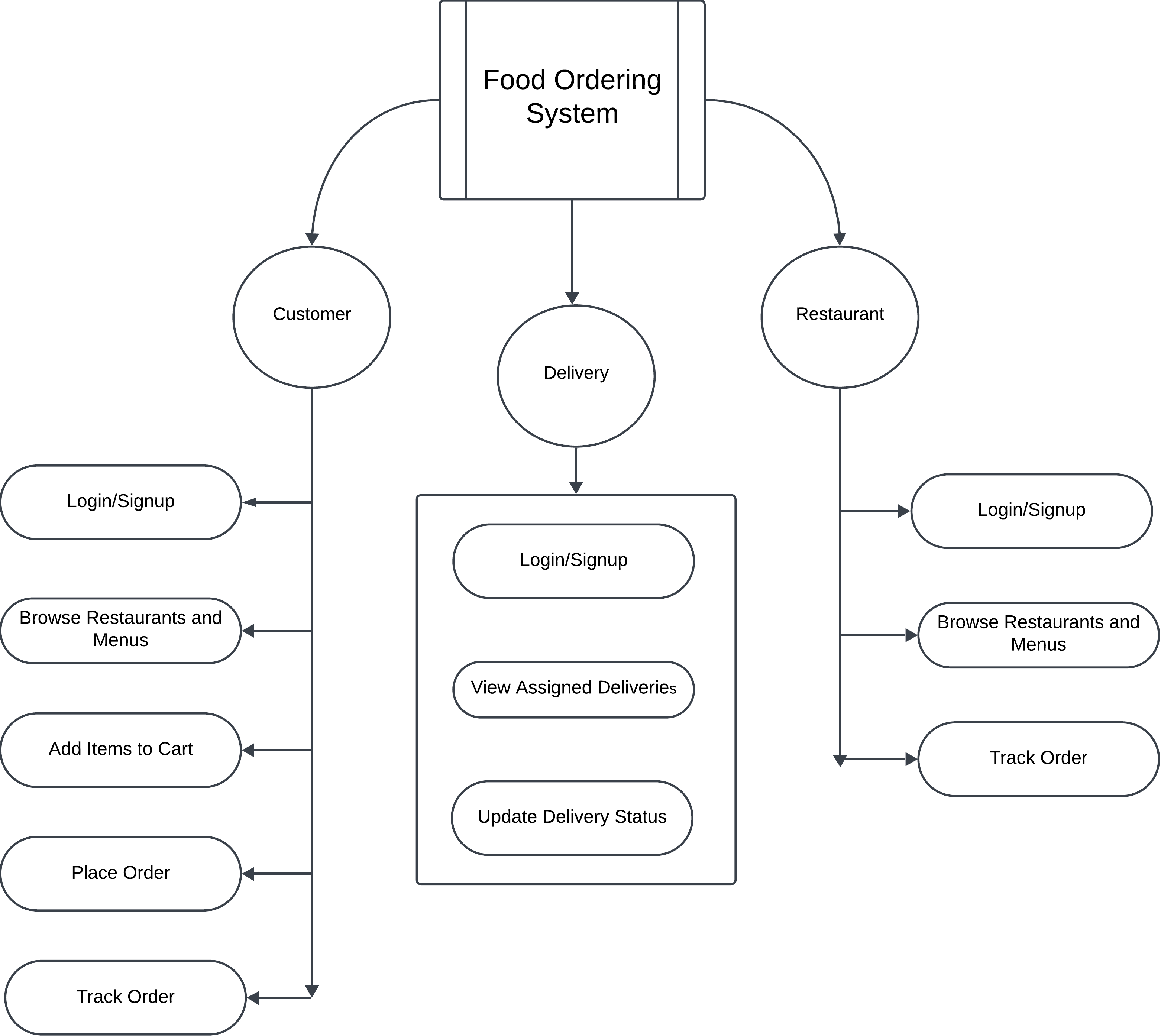
By leveraging modern web technologies and frameworks, this project ensures scalability, security, and a user-friendly experience, making it a robust solution for the ever-growing demand for online food services.

# Architecture Overview :-

The system follows a **three-tier architecture**:

* **Presentation Layer**: The user interface, built using HTML, CSS, and JavaScript, provides an intuitive experience for all users.
* **Application Layer**: The backend logic, implemented using Django, handles data processing, authentication, and user-specific workflows.
* **Data Layer**: SQLite3 is used to store user data, order details, and menu information securely and efficiently.

# Work Flow :-



# Database Design :-

Key tables in the database:

* **Customer Table**: Stores user credentials and profile information.
* **CartItem Table**: Contains the cart items which are added by customer.
* **Orders Table**: Lists food items which are ordered by customer.
* **Delivery Table**: Stores user credentials and profile information.
* **Restaurant Table**: Stores user credentials and profile information.
* **MenuItem Table**: Contains list of food items of that restaurant.

Technologies Used :-

* **HTML** – Used for creating the structure and content of web pages. It defines the layout, headings, paragraphs, and other elements of the website.
* **CSS** – Utilized for styling the web pages, including setting the layout, colors, fonts, and responsiveness to ensure the site looks great across different devices.
* **JavaScript (JS)** – Implemented for interactivity, such as handling form validations, dynamic updates (like real-time order tracking), and enhancing the overall user experience.
* **Django** – The backend framework used to handle user authentication, database management, and server-side logic. Django's powerful ORM (Object-Relational Mapping) simplifies database operations, and its built-in features make the development process faster.
* **Pillow** - **Pillow** is a Python Imaging Library (PIL) fork, which adds image processing capabilities to your Python applications.
* **SQLite3** – A lightweight relational database used for storing all the system's data, including user profiles, orders, restaurant menus, and delivery status updates. Its simplicity and integration with Django make it a great choice for the project.

Key Modules / Apps :-

1. Customer Module :-

* **Features**: Allows customers to sign up, log in, browse restaurants and menus, add items to the cart, place orders, track order status, and manage their profiles.
* **Responsibilities**: This module handles the customer's entire journey on the platform, ensuring a smooth experience from login to order completion.

2. Delivery Module :-

* **Features**: Enables delivery personnel to sign up, log in, view assigned deliveries, update delivery status, and track orders.
* **Responsibilities**: This module focuses on managing the delivery process, ensuring that delivery personnel can efficiently complete their tasks and update the status of deliveries in real-time.

3. Restaurant Module :-

* **Features**: Allows restaurant owners to sign up, log in, manage their menu, view incoming orders, and update order statuses.
* **Responsibilities**: This module facilitates restaurant management by providing tools to update the menu, manage orders from customers, and maintain a smooth operation.

Commands used :-

* **Install and Set Up Virtual Environment**

Create a virtual environment by running the command python -m venv <env\_name>.

Activate the virtual environment:

* + On Windows, navigate to <env\_name>\Scripts\activate.
  + On macOS/Linux, use the command source <venv\_name>/bin/activate.
* **Create a Django Project**

Use the command django-admin startproject <project\_name> to create a new Django project.

* **Create a Django App**

Inside the project directory, create an app with the command python manage.py startapp <app\_name>.

* **Perform Database Migrations**

Generate migration files based on your models by running python manage.py makemigrations. This prepares the SQL needed to modify the database schema.

Apply these migrations to the database with the command python manage.py migrate. This updates the database schema to match your current models.

* **Run the Development Server**

Start the development server using python manage.py runserver. This allows you to view your project in a browser at <http://127.0.0.1:8000/>.

# Website images :-

Starting page (index.html) :-

A screenshot of a login form

Description automatically generated

Customer :-

login :-

A login screen with green and black text

Description automatically generated

Signup page :-

A screenshot of a login form

Description automatically generated

Home page :-

A screenshot of a computer

Description automatically generated

About us page :-

A screenshot of a website

Description automatically generated

Restaurants page :-

A screenshot of a restaurant

Description automatically generated

Menu items for restaurant page :-

A screenshot of a food menu

Description automatically generated

Cart page :-

A screenshot of a computer

Description automatically generated

Profile page :-

A screenshot of a computer

Description automatically generated

Edit profile page :-

A screenshot of a computer

Description automatically generated

Delivery :-

Login page :-

A login screen with green and white text

Description automatically generated

Singup page :-

A login form with green and black text

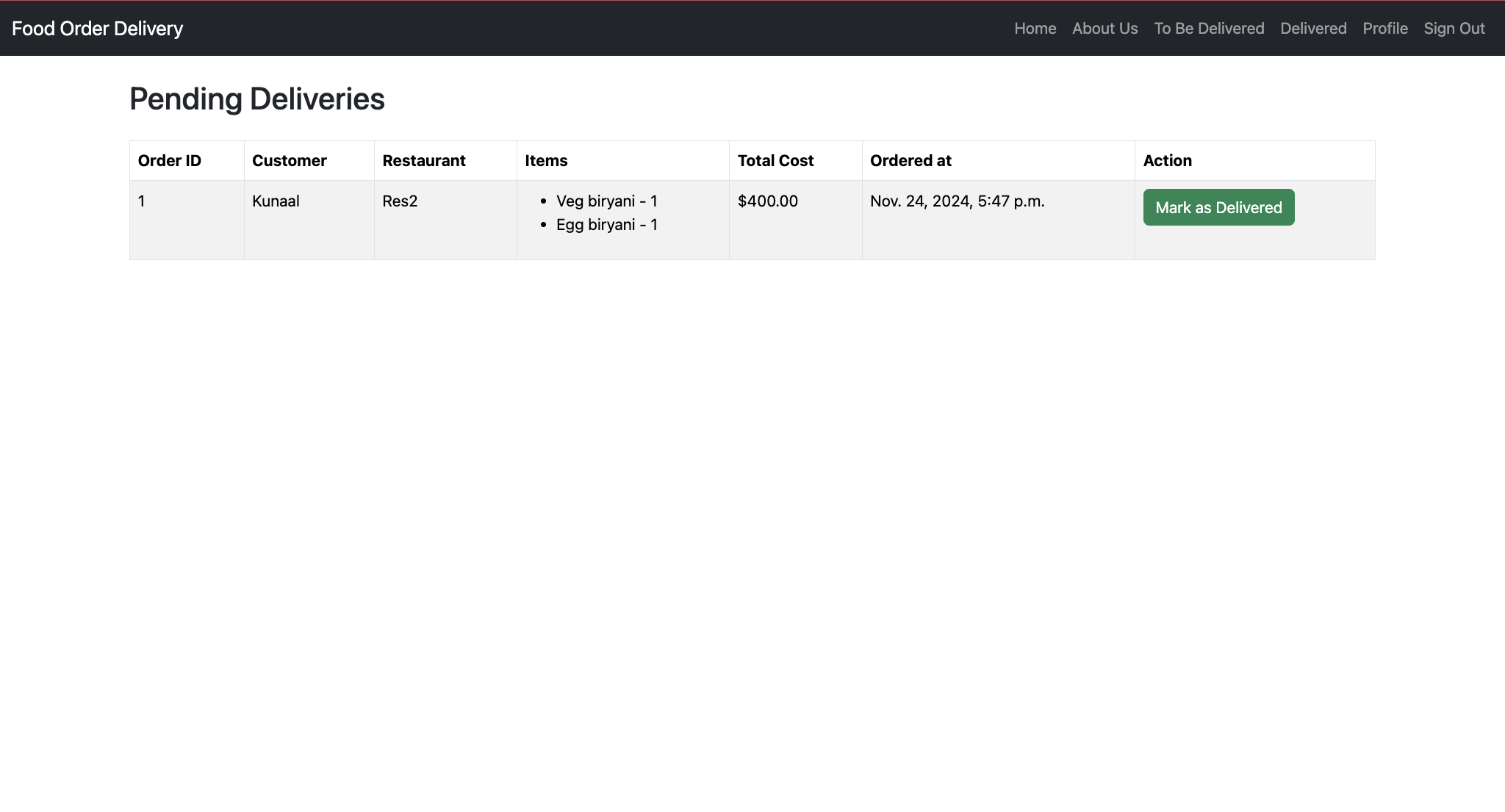
Description automatically generated

Home page :-

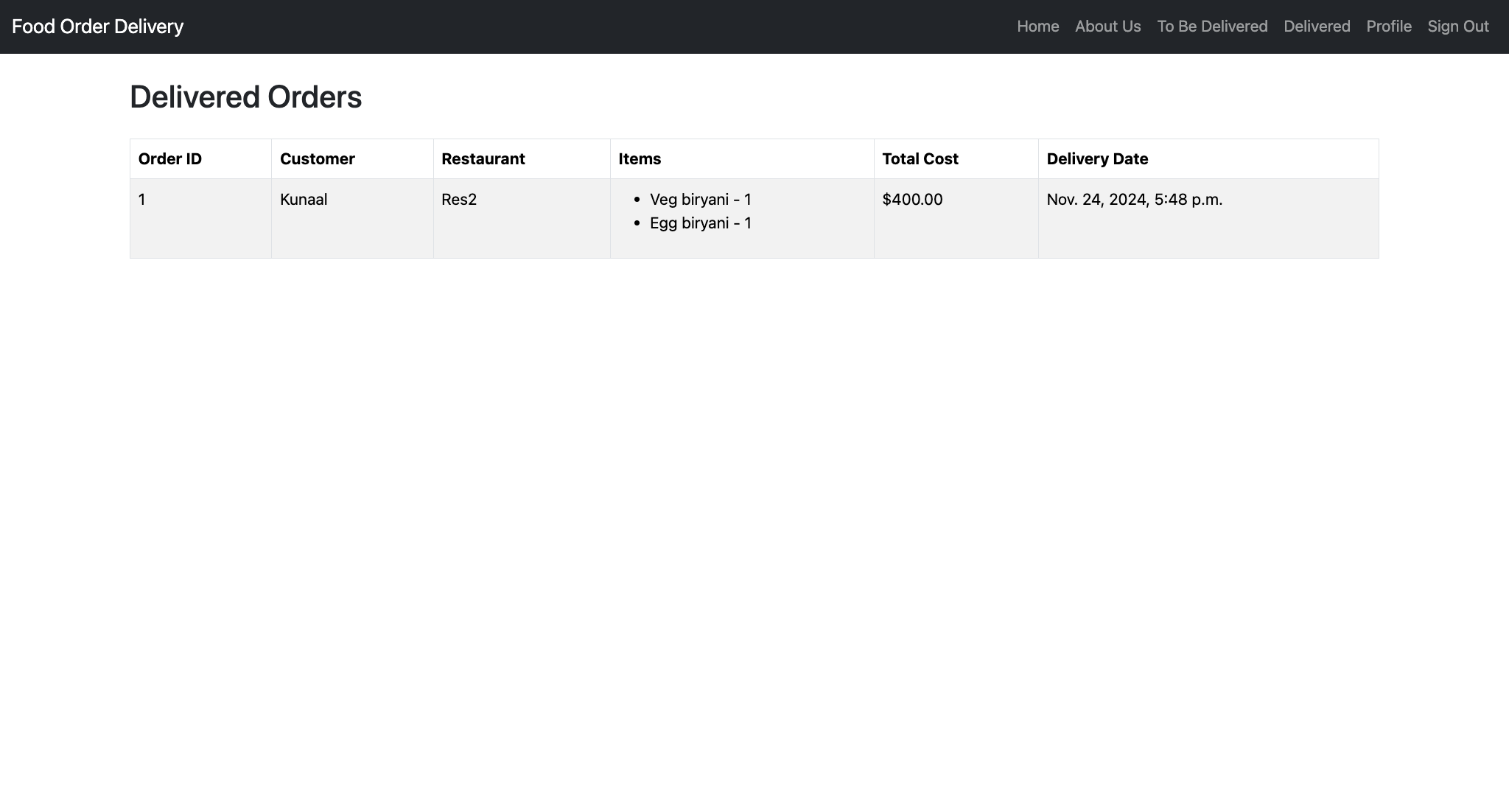
A screenshot of a delivery dashboard

Description automatically generated

To be delivered page :-



Delivered page :-



Restaurant :-

Login page :-

A login screen with green and black text

Description automatically generated

Signup page :-

A screenshot of a sign up form

Description automatically generated

Home page :-

A screenshot of a menu

Description automatically generated

About us page :-

A screenshot of a website

Description automatically generated

Menu page :-

A screenshot of a menu

Description automatically generated

Adding menu item modal :-

A screenshot of a menu

Description automatically generated

Edit menu item page :-

A screenshot of a menu

Description automatically generated

# Future scope :-

* **GPS Tracking for Delivery Personnel**

Enable real-time location tracking for customers to view their order status on a live map.

Optimize delivery routes for faster and more efficient service.

* **Item Rating Feature**

Allow customers to rate menu items, helping restaurants improve quality.

Provide insights for other customers to make better food choices.

* **Customer Feedback System**

Facilitate feedback for restaurants and delivery personnel to improve service quality.

Enable customers to share their experiences, ensuring accountability and transparency.

Conclusion :-

The Food Ordering System project successfully delivers a streamlined platform for customers, restaurants, and delivery personnel. The system provides:

* **Customer Convenience**: Enables easy browsing of restaurants, menu items, cart management, order placement, and real-time tracking.
* **Restaurant Efficiency**: Allows seamless management of menus and orders, helping restaurants maintain high service quality.
* **Delivery Integration**: Facilitates delivery personnel to manage assigned deliveries and update statuses effectively.
* **Role-Based Access**: Implements secure authentication and authorization for each user role, ensuring data integrity and privacy.
* **Interactive UI**: A responsive and user-friendly interface enhances the experience across all device types.