



ANNA UNIVERSITY REGIONAL CAMPUS COIMBATORE - 641046

NAAN MUDHALVAN COURSE

PHASE 3

NAME	MOHANA GURU V
NAAN MUDHALVAN REGISTER NO.	au710021106020
DOMAIN	CLOUD COMPUTING
PROJECT	E-COMMERCE APPLICATION ON IBM CLOUD FOUNDRY

INTRODUCTION:

It is a growing trend especially in urban areas and on college campuses that allows people to order from restaurants featuring interactive menus, by use of their Internet connection. The current food delivery platforms provide different discounts on the same item but in different platform. It is also seen that delivery time of an item is not same in all the applications.

ABSTRACT:

Online **FOOD DELIVERY SYSTEM** is a website designed primarily for use in the food delivery industry. This system will allow hotels and restaurants to increase scope of business by reducing the labor cost involved. This system allows to quickly and easily manage an online menu which customers can browse and use to place orders with just few clicks. Restaurant employees then use these orders through an easy to navigate graphical interface for efficient processing.

KEY COMPONENTS OF ONLINE FOOD DELIVERY SYSTEM:

An online food delivery system is a complex ecosystem that involves various key elements to function efficiently and provide a seamless experience for customers, restaurants, and delivery personnel. Here are the key elements of an online food delivery system:

1. USER INTERFACE:

Website and/or mobile app: Customers interact with the system through a user-friendly website or mobile application to browse menus, place orders, and track deliveries.

2. RESTAURANT LISTING:

Comprehensive restaurant database: The system should have an extensive list of partnered restaurants, each with a dedicated page showcasing their menu, prices, reviews, and other relevant information.

3. MENU MANAGEMENT:

Menu integration: Restaurants should be able to easily upload and update their menus, including prices, descriptions, and images, through a dedicated dashboard.

4. ORDERING SYSTEM:

Cart and checkout: Customers should be able to add items to their cart, customize orders, and securely check out, with options for various payment methods.

5. USER ACCOUNTS:

Search functionality: Users should be able to search for specific dishes, cuisine types, or restaurants. Filters like price range and dietary preferences can enhance the user experience.

6. RATING AND REVIEWS:

Customer feedback:

Customers should have the option to rate and review restaurants and delivery drivers, providing valuable information for others.

PROCEDURE TO RUN THE CODE IN Visual Studio:

Step 1:

Create a new **Folder Name**: “**Online food delivery system**” on your computer.

Step 2:

Launch Visual Studio and **open** the folder in the editor.

Step 3:

Inside the folder, create a **new Python file** named “**food_delivery.py**”.

Step 4:

Open the “**food_delivery.py**” file and written the code needed for this project.

Step 5:

Save the file.

CODE:

```
from flask import Flask, request, jsonify
```

```
app = Flask(__name)
```

```
# Define routes for various functionalities
```

```
@app.route('/api/menu', methods=['GET'])
```

```
def get_menu():
```

```
    # Code to retrieve the menu items from the database
```

```
    menu = [{"name": "Burger", "price": 250}, {"name": "coffe", "price":  
350}]
```

```
    return jsonify(menu)
```

```
@app.route('/api/order', methods=['POST'])
```

```
def place_order():
```

```
    # Code to process orders, validate payment, and store in the database
```

```
    data = request.json
```

```
    # Process the order and return a response
```

```
    return jsonify({"message": "Order placed successfully"})
```

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```

Code Explanation**1. Import Libraries:**

- The code begins by importing the necessary Python libraries:

- **json**: Used for working with JSON data, which is commonly used in IoT applications for data interchange.

2. *Defining Routes:*

The code defines two routes for different functionalities:

GET /api/menu: This route is used to retrieve the menu items from the database and return them as JSON. It responds to HTTP GET requests.

POST /api/order: This route is used to place an order. It expects order data in JSON format in the request body and processes the order. It responds to HTTP POST requests

3. `if __name__ == '__main__':`
`app.run(debug=True)`

This conditional block ensures that the Flask app is only run if the script is executed directly (not when imported as a module). It starts the Flask development server with debugging enabled, allowing you to see detailed error messages in the console while you're developing your application. To run the application, use the command `python food_delivery.py` where `food_delivery.py` is the name of your script.

OUTPUT:

```
#####  
    Welcome To Hacker Caffee  
#####  
Please Chose your Order:  
Coffe - Price: 350Tk  
Burgers - Price: 250Tk  
Meatloaf - Price: 150Tk  
Lasagna - Price: 220Tk  
#####  
Enter Item Name:coffe  
Order More or Type: done  
Enter Item Name:burgers  
Order More or Type: done  
Enter Item Name:done  
Order More or Type: done  
#####  
You order the follwing Items:  
1 - Coffe - 350  
2 - Burgers - 250  
Your Total Price: 600  
#####  
Thanks! Order Again.
```

CONCLUSION:

With online food ordering system, restaurant an mess menu online can be set up and customers can easily place order. This is achieved through an easy to use graphical interface menu options. The users can add any number of items to the cart from any of the available food categories. We have created website in focus of future food ordering system, this website will helpful to many people.