PROJECT TITLE

HOMEMADE PICKLES AND SNACKS – E-COMMERCE WEB APPLICATION

Technology Stack: Flask, HTML, CSS, JS, AWS DynamoDB, EC2, IAM, SNS

Submitted by

Kanikaram Dedeepya Sree

Roll No: 228X1A1214

TABLE OF CONTENTS

1. Introduction

Homemade Pickles and Snacks is an e-commerce web application that brings the essence of traditional homemade pickles and snacks into the digital marketplace. With increasing demand for homemade, hygienic, and preservative-free food items, this platform offers customers a delightful way to browse, select, and order from a variety of authentic Indian products.

The platform has been built with user-friendliness and responsiveness in mind. It includes features such as login, signup, categorized product listings, shopping cart, order placement, and order tracking. Users can also provide feedback on their shopping experience.

The backend logic is handled by Flask, a Python-based micro web framework. The frontend is constructed using HTML, CSS, and JavaScript. To scale and manage data securely, cloud services like AWS DynamoDB, EC2, SNS, and IAM are integrated. DynamoDB is used for storing data, EC2 for hosting, SNS for order notifications, and IAM for permission management.

The project demonstrates real-time shopping flow, product handling, and basic cloud integration techniques. It provides a realistic simulation of an online food store, which can be further scaled into a full-fledged e-commerce platform.

2. Objective

The main objectives of this project are as follows:

1. To develop a user-friendly and visually appealing interface using HTML, CSS, and JavaScript.
2. To implement secure login and signup features using Flask.
3. To provide users with an organized view of products categorized into veg pickles, non-veg pickles, and snacks.
4. To allow users to add and remove items from a virtual shopping cart.
5. To facilitate a smooth order placement experience with delivery tracking.
6. To accept and store feedback for quality improvement.
7. To store user and order data securely using AWS DynamoDB.
8. To deploy the application on AWS EC2 for reliable hosting.
9. To send order confirmation messages through AWS SNS.
10. To ensure role-based secure access via AWS IAM.
11. To demonstrate real-world integration of cloud services into a web application.
12. To simulate the workflow of an e-commerce shopping experience.
13. To practice full-stack development with Flask and AWS.
14. To create a scalable structure for future enhancements.
15. To offer an innovative solution for local food businesses to sell online.

3. Features

The Homemade Pickles and Snacks project includes several front-end and back-end features that make the application interactive, secure, and user-friendly.

Frontend Features:
- Clean and intuitive user interface using HTML and CSS.
- Categorized menu for Veg Pickles, Non-Veg Pickles, and Snacks.
- Add to Cart and Remove from Cart functionality.
- Smooth order form with validation and order tracking bar.
- Feedback submission form with validations.
- Responsive layout for mobile and desktop users.

Backend Features:
- Flask-based routing and backend processing.
- User login and signup using hashed passwords.
- Cart session management with item persistence.
- Data handling with AWS DynamoDB for orders and feedback.
- SMS notifications using AWS SNS.
- Hosting and deployment using AWS EC2.
- Role-based access with AWS IAM for resource protection.

4. Technologies Used

The technologies used in the Homemade Pickles and Snacks project include:

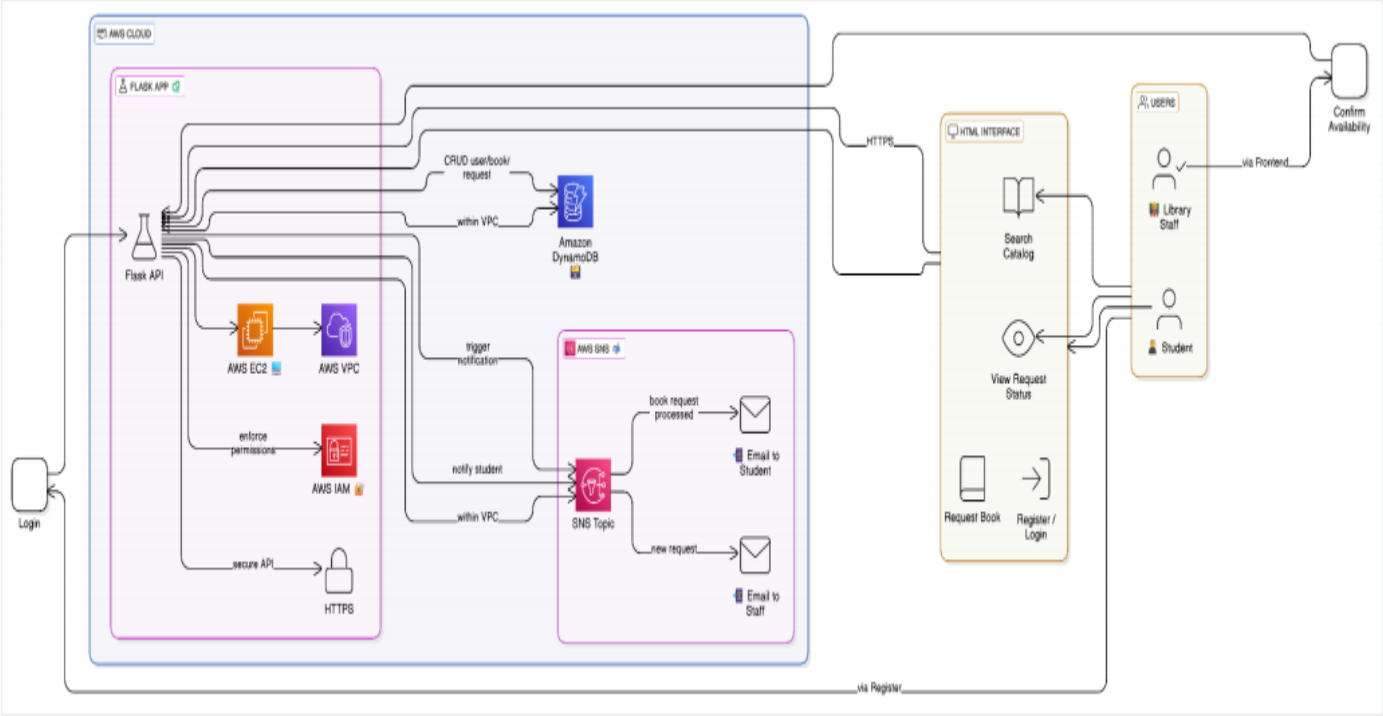
- HTML5: Structuring the front-end interface.
- CSS3: Styling components for visual appeal.
- JavaScript: Dynamic interactions like cart and button actions.
- Python: Backend logic and request handling.
- Flask: Web framework used for routing and sessions.
- Jinja2: Templating engine for rendering dynamic HTML.
- AWS DynamoDB: NoSQL database used to store users, orders, and feedback.
- AWS EC2: Virtual cloud server to host the Flask application.
- AWS SNS: Notification service to send SMS alerts to users.
- AWS IAM: Identity and Access Management to control secure access to AWS services.

5. System Architecture

The system follows a modular architecture using client-server design principles:

1. Client Layer: The user interacts with the web interface (HTML/CSS/JS).
2. Application Layer: Flask manages routing, user sessions, and logic.
3. Database Layer: AWS DynamoDB stores data like user info, orders, and feedback.
4. Notification Layer: AWS SNS sends real-time order messages.
5. Deployment Layer: Hosted on AWS EC2 with IAM providing secure access controls.

[Architecture Diagram Placeholder: Users → Flask App → DynamoDB/SNS/EC2]

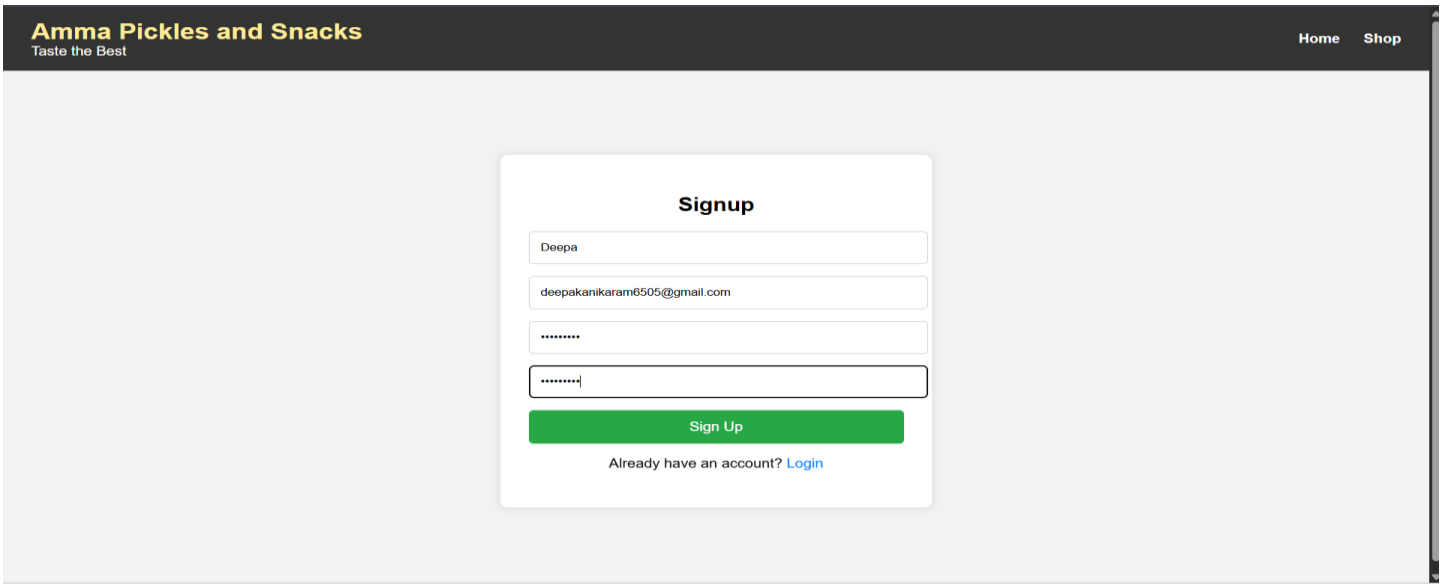


6. Modules Overview

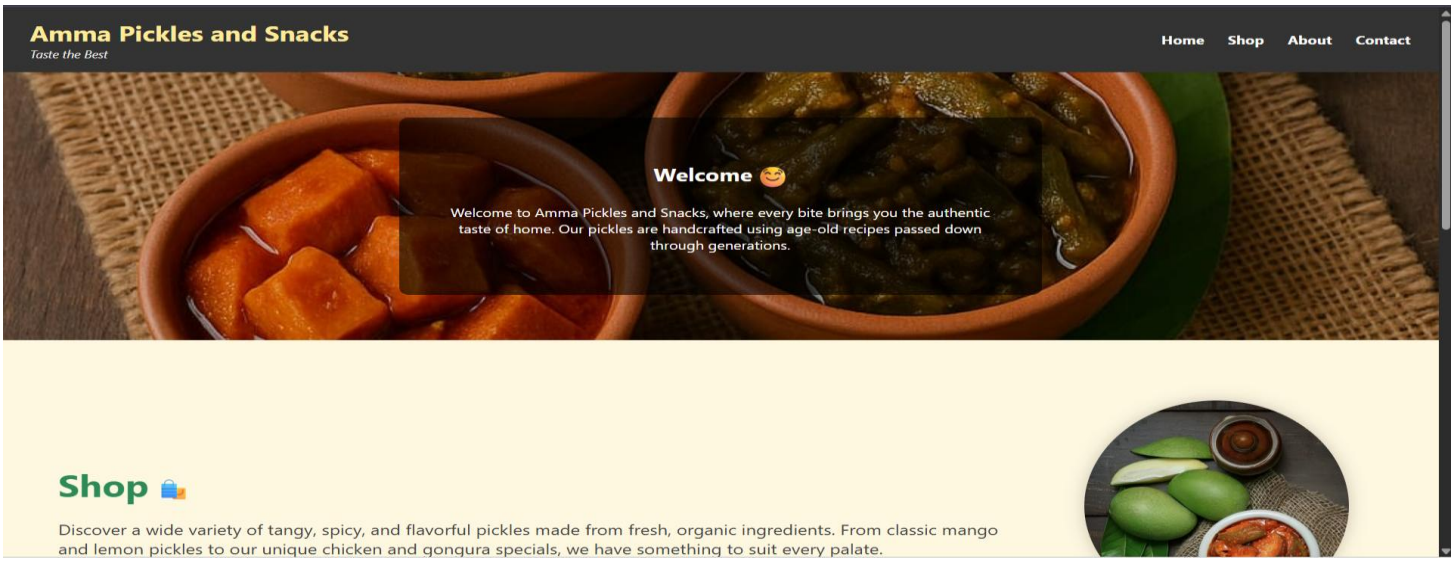
This section describes the major modules and pages in the Homemade Pickles and Snacks web application:

- index.html: Welcome page with brand title and navigation to login/signup.

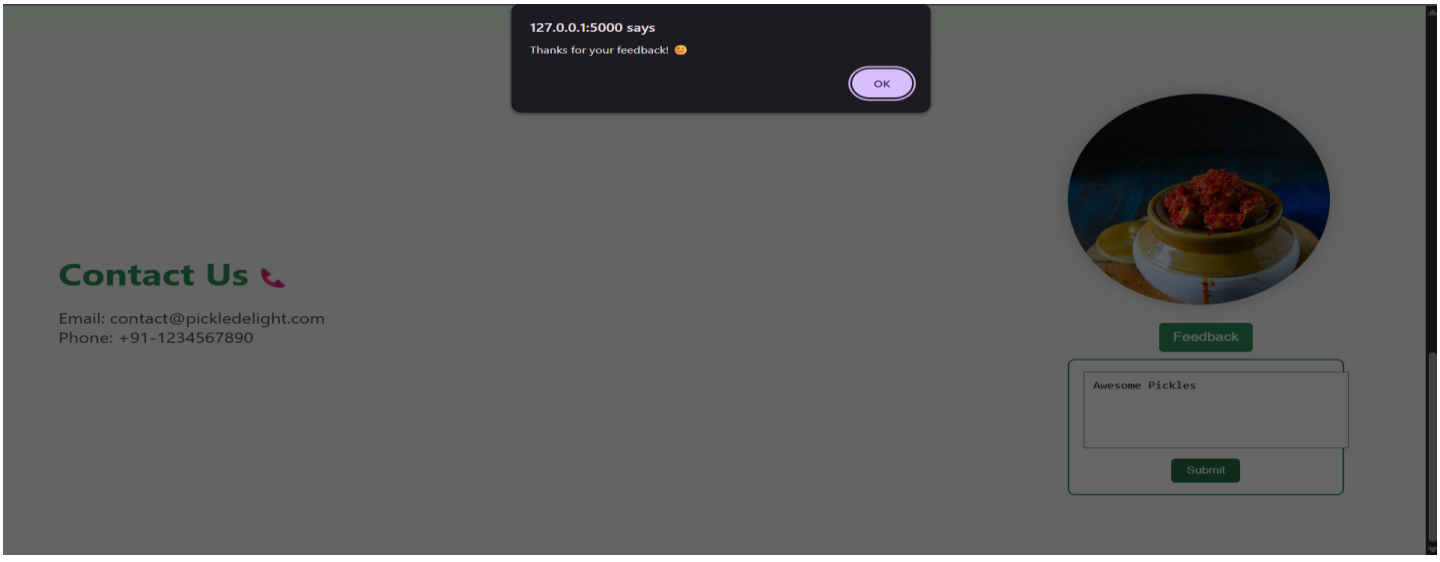
- login.html / signup.html: User authentication forms with validation.



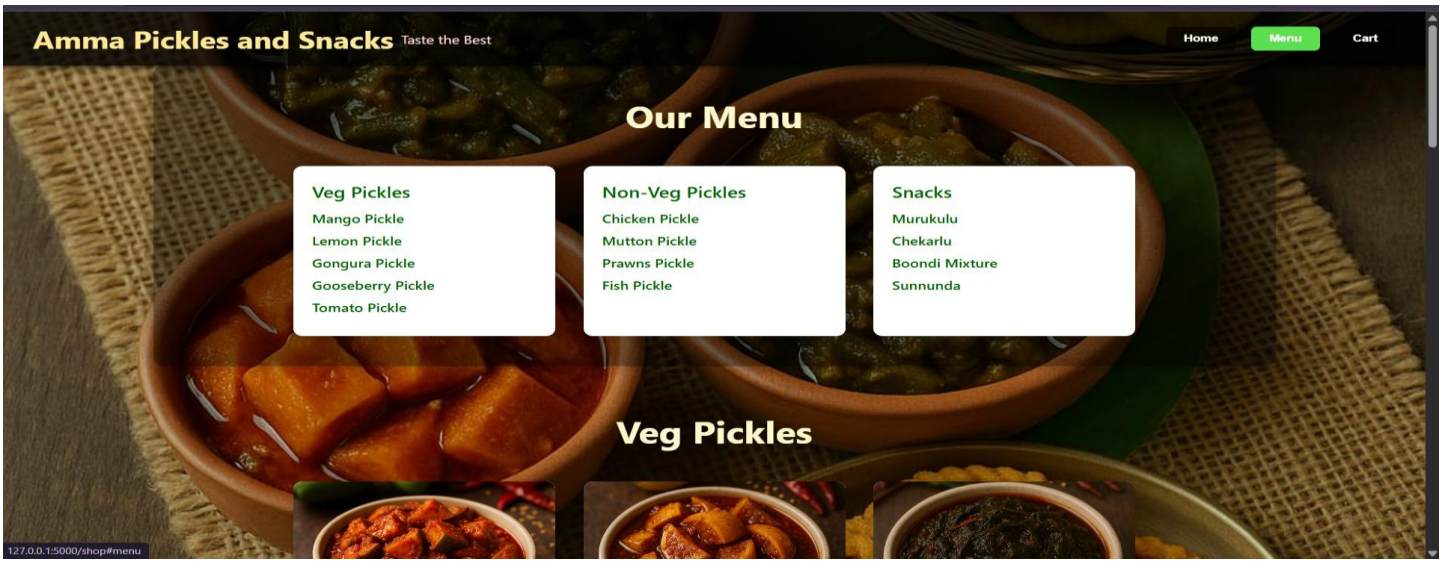
- home.html: User dashboard with options to navigate to Shop, About, Contact, and Feedback sections.



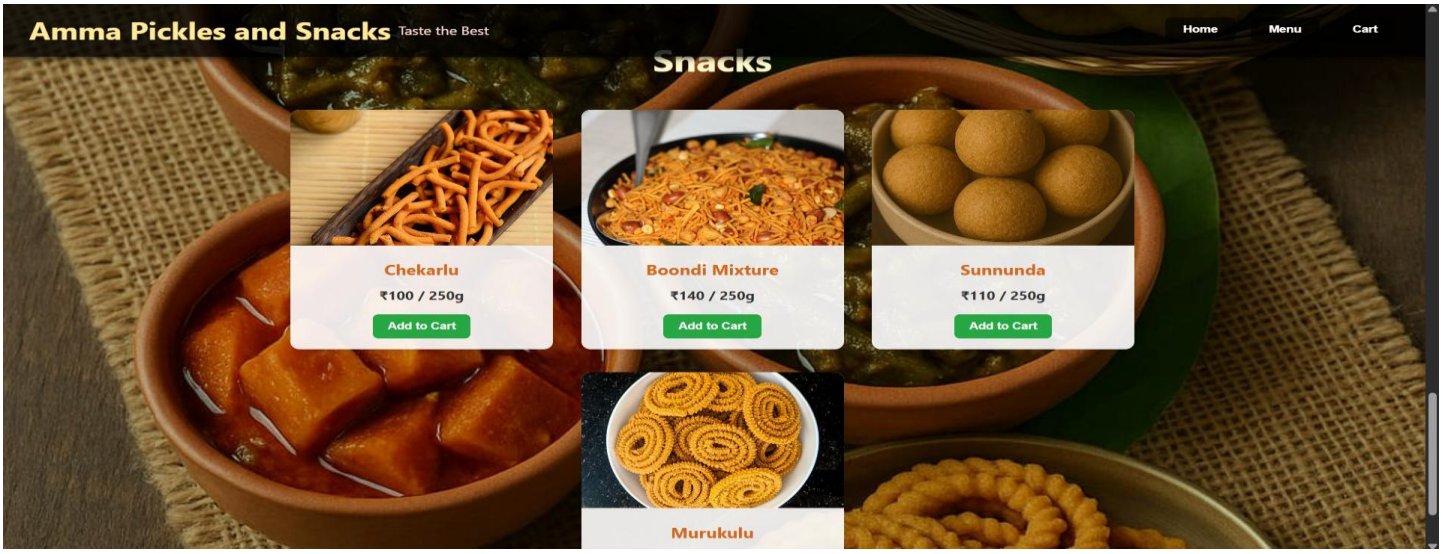
-feedback page



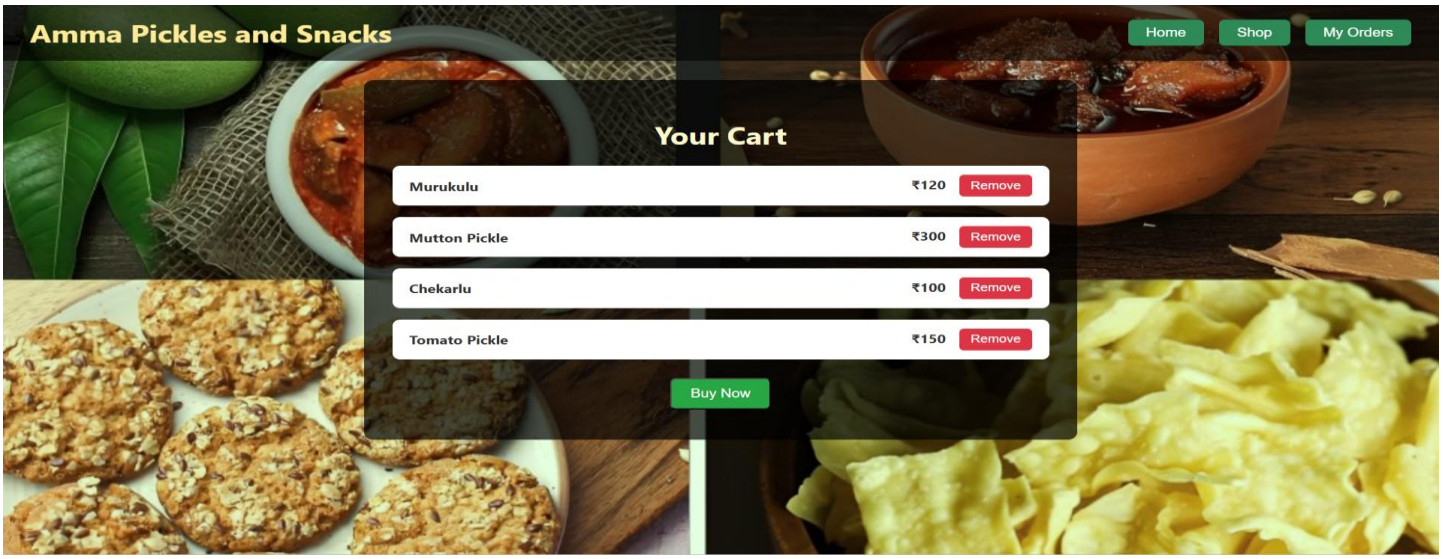
- shop.html: Lists categorized products under Veg Pickles, Non-Veg Pickles, and Snacks with Add to Cart buttons.



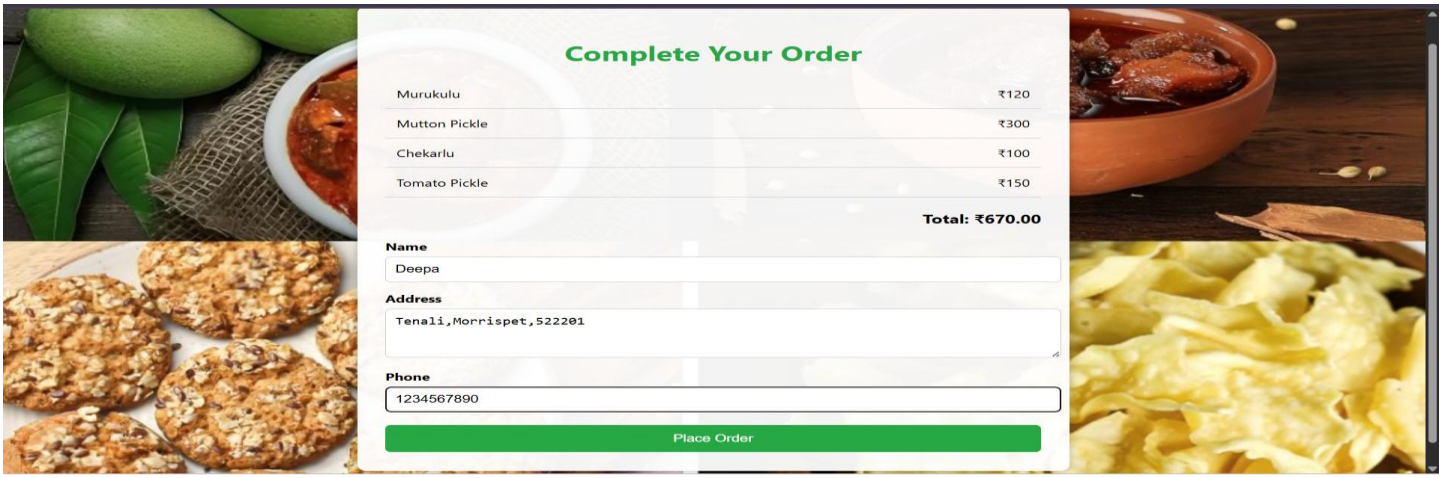
-menu page



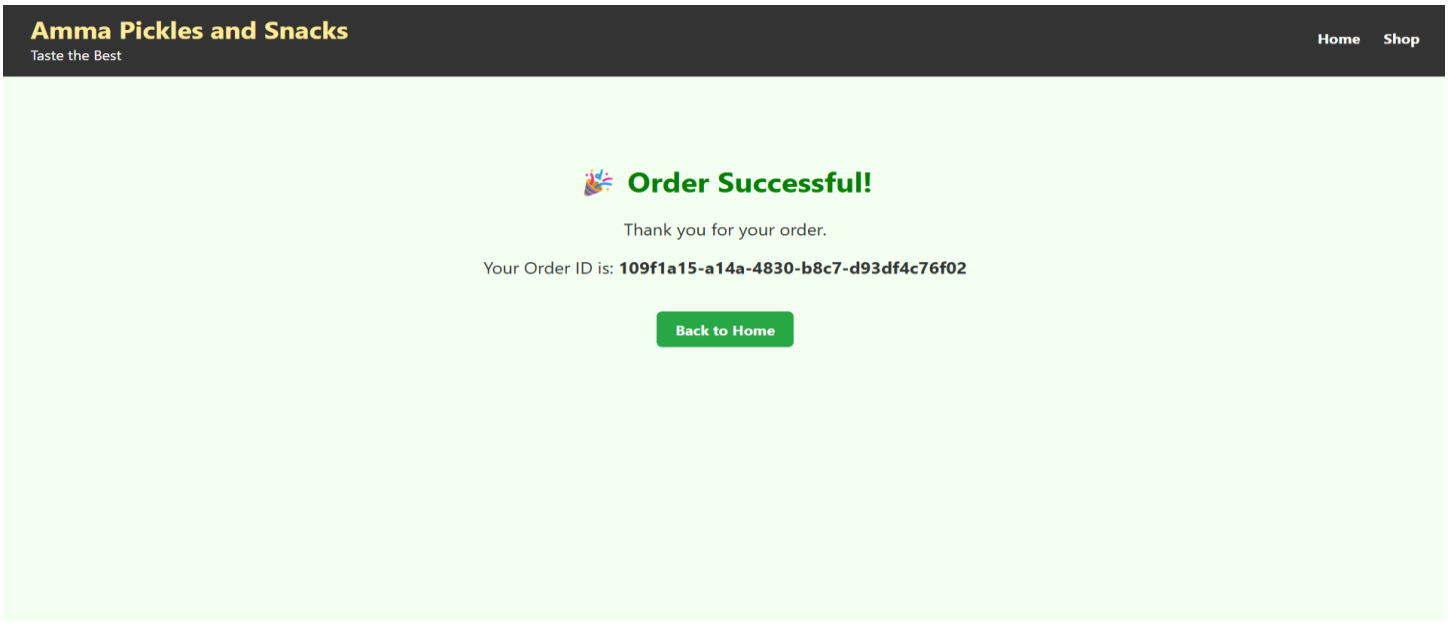
- cart.html: Displays all added items in cart with options to remove or proceed to Buy Now.



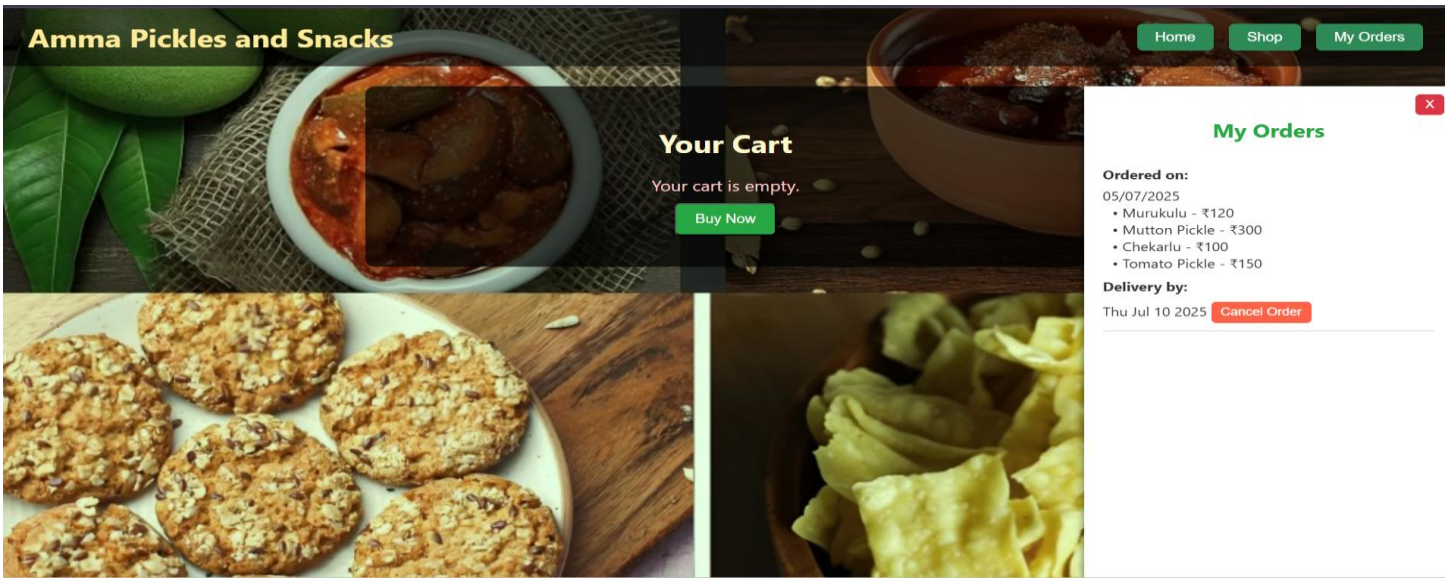
- buynow.html: Checkout page where users enter delivery details and see the order tracker.



- success.html: Displays order confirmation with tracking message.



-order tracking



7. AWS Integration

The Homemade Pickles and Snacks application integrates several AWS services for scalability and cloud readiness:

- AWS DynamoDB: Used to store all user data, product orders, and feedback entries in a NoSQL format. The integration is handled using boto3.
- AWS SNS: Configured to send SMS alerts upon successful order placements. Uses simple publish method from boto3 SNS client.
- AWS EC2: The web application is deployed on a virtual EC2 instance with port access configured for HTTP.
- AWS IAM: Roles and permissions are defined using IAM policies to provide secure access to DynamoDB and SNS APIs, avoiding hardcoded credentials.

This integration ensures real-world deployment scenarios are achieved and makes the app production-ready.

```
use_dynamo = False
sns = None
users_table = None
orders_table = None
feedback_table = None
local_users = {}
local_orders = []

try:
    session_boto = boto3.Session()
    dynamodb = session_boto.resource('dynamodb', region_name='ap-south-1')
    sns = session_boto.client('sns', region_name='ap-south-1')

    # Test connections
    dynamodb.meta.client.list_tables()
    users_table = dynamodb.Table('Users')
    orders_table = dynamodb.Table('Orders')
    feedback_table = dynamodb.Table('Feedback')

    use_dynamo = True
except NoCredentialsError:
    print("No AWS credentials found, falling back to local storage.")
except ClientError as e:
    print(f"AWS Client error: {str(e)}")
```

```
    if use_dynamo:
        try:
            orders_table.put_item(Item=order)
            message = f"Hi {name}, your order {order_id} is confirmed. Total ₹{total}. Thank you!
            sns.publish(
                TopicArn='arn:aws:sns:ap-south-1:123456789012:YourTopicName',
                Message=message,
                Subject='Order Confirmation'
            )
        except ClientError as e:
            print("SNS or DynamoDB error:", e)
    else:
        local_orders.append(order)

    return render_template('success.html', order_id=order_id)

return render_template('buynow.html')
```

8. Conclusion

The Homemade Pickles and Snacks project is a well-structured, full-stack web application designed to deliver homemade food products online in a user-friendly way. From creating a clean user interface to implementing backend logic and cloud integration, the project reflects practical understanding of end-to-end development.

The system successfully simulates a complete online shopping workflow, including product browsing, cart management, order tracking, and feedback collection. The integration with AWS services such as DynamoDB, SNS, EC2, and IAM showcases a strong foundation for building scalable and secure cloud-native applications.

This project serves as an excellent example of combining traditional food business ideas with modern web technologies. It can be further enhanced by adding payment gateways, admin dashboards, and user order history. Overall, it is a complete demonstration of frontend, backend, and cloud integration in a single unified platform.