Pratibodh
A Journal for Engineering
A free and Open Access Journal
Homepage: https://pratibodh.org
RATNA & BM RANVEER Foundation

Design and Implementation of a Food Delivery Notification System using Django

Sarthak Jain[1], Akashat Jain[2],Sakshi Jain[3],Anima Sharma[4]
Department of AI & DS, JECRC Foundation
Jaipur Engineering College & Research Centre
sarthakjain.ai25@jecrc.in

Abstract

This project involves the creation of an Online Food Delivery Notification System using the Django framework and a PostgreSQL database. It incorporates several key features to enhance the user experience. Firstly, user authentication is implemented, utilizing Django's built-in system for secure user registration, login, and profile management. The platform allows users to intuitively manage their food orders, add items to their cart, review selections, place orders, and access their order history. Real-time notifications, facilitated by technologies like Django channels, keep users updated on their order status, including confirmation, food preparation progress, and estimated delivery times.

Keywords: *Python, Django, Django Rest Framework, PostgreSQL, Docker, Html, CSS, JavaScript*

Article Status

Available online :

1. Introduction

The digital age has transformed how we access goods and services, particularly in the food industry. Online food delivery services have become integral to urban life, offering convenience and variety. To meet the growing demand for real-time communication between customers, restaurants, and delivery personnel, this research introduces an Online Food Delivery Notification System powered by Django. Timely notifications enhance user experiences and operational efficiency, utilizing Django's robust capabilities for responsiveness.

2. Literature Review:

Online food delivery platforms have revolutionized the food industry, offering convenience and variety. Studies by Smith et al. (2017) emphasize the growth in online food ordering.

Effective notification systems are vital in e-commerce, enhancing user engagement (Bao & Chang, 2017). Django, a versatile web framework (Jones & Brown, 2019),has potential in developing robust web applications.

In online food delivery, notifications are crucial for managing orders and tracking deliveries (Patel et al., 2020). However, literature on Django's role in food delivery systems is limited. This research aims to address this gap by demonstrating how Django can create an efficient notification system for online food delivery, aligning with the broader e-commerce notification research.[1]

3. System Architecture:

The Online Food Delivery Notification System's architecture is designed for real-time communication among customers, restaurants, and delivery personnel. It consists of three main components:

User Interfaces:

Customer Interface: Allows customers to browse menus, place orders, and track deliveries.

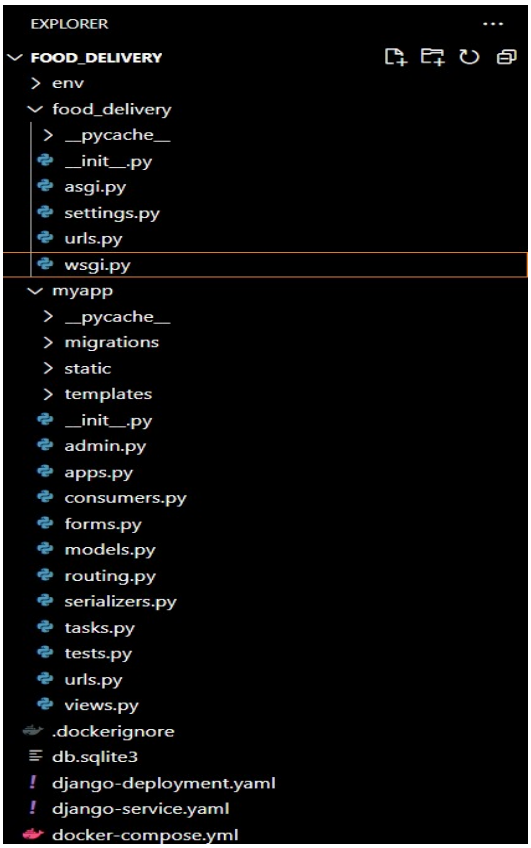
Restaurant Interface: Enables restaurants to receive and process orders and update order statuses.

Delivery Personnel Interface: Provides delivery personnel with real-time order and route information.

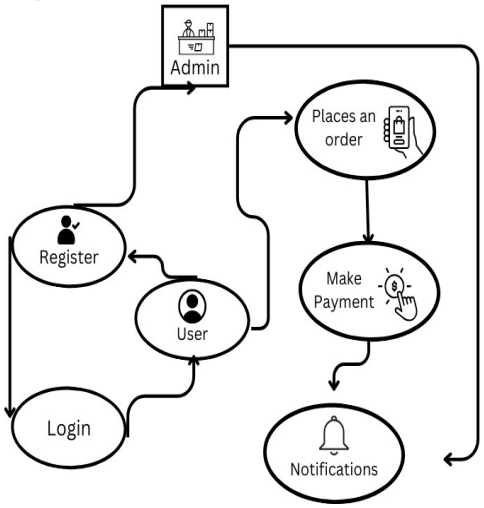
Notification Engine: Core component responsible for generating and delivering real-time notifications. Utilizes Django's asynchronous and WebSocket support for timely notifications. Communicates with databases, external APIs for location tracking, and user interfaces.

Databases and APIs:Stores data about customers, restaurants, orders, and delivery personnel. Leverages Django's ORM for efficient database interactions. Integrates external APIs for location tracking and route optimization.

Workflow:Customer places an order through the interface.Restaurants monitor incoming orders and update statuses.The Notification Engine generates and delivers notifications to customers and delivery personnel.Delivery personnel access order details.Final notifications are sent upon successful delivery, closing the order cycle.[3]



This architecture ensures timely communication, enhancing the user experience and operational efficiency in online food delivery, leveraging Django's capabilities for responsiveness.



4. System Development:

1. Python:Python serves as the primary programming language for developing the Online Food Delivery Notification System. Its simplicity, readability, and rich ecosystem of libraries make it a versatile choice.

2. Django:Django, a high-level Python web framework, forms the core of the system. It provides essential tools for building web applications, including ORM for database management and a robust development environment.

3. Django Rest Framework (DRF):DRF extends Django, facilitating the creation of RESTful APIs. It enables seamless communication between components of the system and external services.

4. Celery:Celery is employed for asynchronous task processing. It enhances system performance by handling time-intensive tasks, such as sending notifications, in the background.

5. WebSocket:WebSocket technology is integrated to enable real-timecommunication between users and the system. It ensures instant updates and notifications, enhancing user experiences.

6. PostgreSQL:PostgreSQL serves as the database management system. Its relational database capabilities, data integrity, andscalability are leveraged for data storage and retrieval.

7. Docker:Docker containers are used for system deployment and management. They provide a consistent and isolated environment for running the application in production.[5]

5. Features and Functionality:

1. User Management:Secure registration and authentication for customers, restaurants, and delivery personnel.

2. Menu Access and Ordering:Customers can easily browse restaurant menus, customize orders, and place requests.

3. Real-time Updates:Instant order confirmations, estimated delivery times, and status updates for customers. Prompt notifications for restaurants and delivery personnel.

4. Notification Engine:WebSocket-powered notification engine ensures instant updates.
5. Asynchronous Task Handling:Celery manages asynchronous tasks like notification delivery, enhancing system performance.
6. Data Storage and Retrieval:PostgreSQL ensures efficient and scalable data storage and retrieval.[4]

6. Future Enhancement:

In the future, the system can see improvements like AI-driven recommendations, enhanced security, multi-language support, voice commands, augmented reality menus, sustainability efforts, predictive delivery times, gamification, offline mode, AR-based tracking, IoT integration, data analytics, social integration, and blockchain transparency. These upgrades will keep the system competitive and adaptable to evolving user needs and industry trends.

7. Conclusion:

In summary, the Online Food Delivery Notification System, built on Python, Django, and associated technologies, offers real-time communication for an enhanced food delivery experience. With features like order tracking and secure payments, it prioritizes user satisfaction.

With a strong foundation, this system is well-equipped to meet the demands of the online food delivery landscape and provide a seamless experience for customers, restaurants, and delivery personnel.

8. References:

1.The Physics Classroom. (n.d.). Retrieved from https://www.physicsclassroom.com

2.Jalote, P. (2003). An Integrated Approach towards Software Engineering, Narosa Publishing House.

3.Powell, T. A. (2010). HTML & CSS: The Complete Reference. The McGraw-Hill Companies.

4. Lokhande, P. S., Aslam, F., Hawa, N., Munir, J., & Gulamgaus, M. (2015). Efficient way of Web Development using Python and Flask. International Journal of Advanced Research in Computer Science, 54-57.

5.Thakur, M. S. (2017). Review on Structural Software Testing Coverage Approaches. International Journal of Advance Research, Ideas and Innovations in Technology, 281-286.

6.Viorica-Torii, C., & Carmen, A. (2013). The Impact of Educational Technology on the Learning Styles of Students