Django Ecommerce Project: Pizza Shop

Ujwal Sahu

TY - B.Voc AIDS

Batch 2

Abstract: This project delivers a full-featured online pizza ordering platform where users can register, browse the menu, customize and add pizzas to a shopping cart, and complete secure payments via Razorpay. An integrated AI chatbot assists customers by answering menu-related questions in real time, while a dedicated admin panel lets business owners manage pizzas, view orders, and update delivery status. From user authentication through order tracking, this application provides a seamless, end-to-end e-commerce experience suitable for real-world pizza delivery operations.

Key Features: AI Chatbot, Payment Gateway, User Authentication, Shopping Cart

Project structure:

```
pizza_project/
├── pizza_shop/        # Main Django app
├── pizza_project/     # Project settings
├── templates/         # HTML templates
├── static/         # CSS, JS, images
├── media/          # User uploaded files
├── requirements.txt    # Dependencies
└── manage.py        # Django management script
```
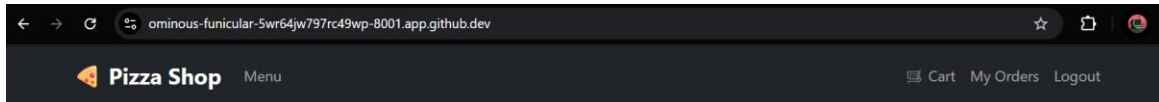
Main page: Pizza Menu Management

- Displays all available pizzas with images, descriptions, sizes, and prices.
- Users can Add the pizzas in the cart by clicking the "Add to Cart" button

Shopping Cart System:

- Add to Cart: Users can add pizzas with selected quantities.
- Update Cart: Users can modify quantities or remove items before checkout.

Checkout page:

- Collects delivery address, contacts details, and initiates payment.

Order Management System:

- After placing the order, the users see an Order display page.
- Where further user can navigate to orders or main page

- Users can see all their Order History

- Admin can see all the order details placed by the users

User Management System:

- Handles user registration, login, profile updates, and order history.

Pizza Management System:

- Allows admins to create, edit, and delete pizza listings with images and availability.

AI Chatbot Integration:

- Provides real-time, menu-specific assistance using a Google Gemini RAG-powered chatbot.
- Google Gemini API is used.
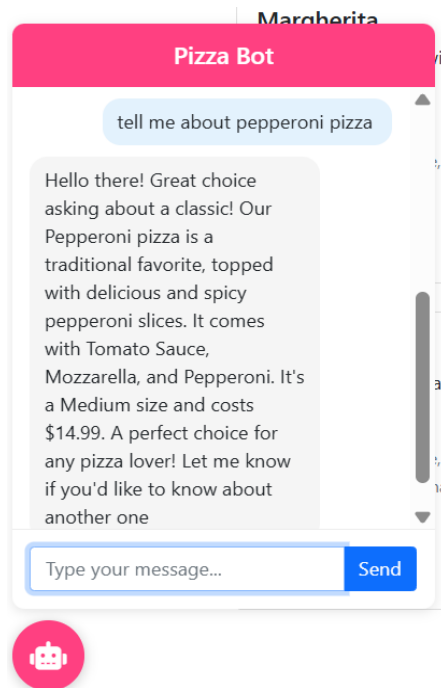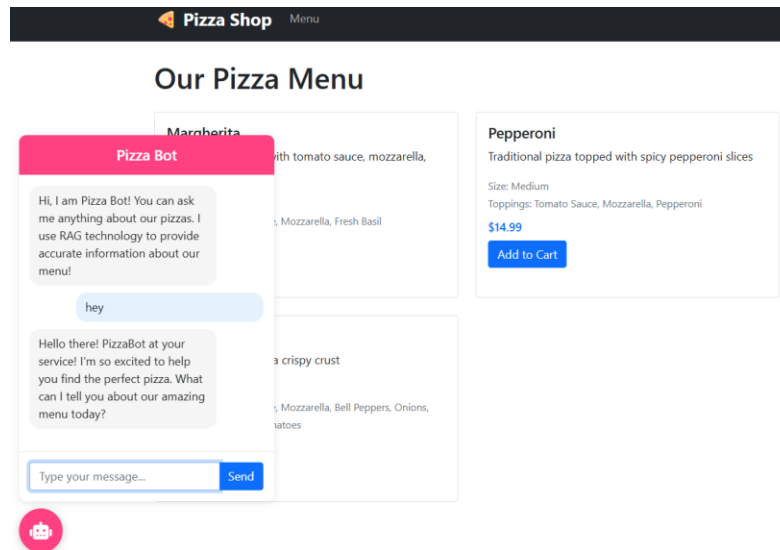
Razor Pay Integration:

- Processes and verifies secure online payments before confirming orders.
- User clicks on the "Pay Now" button

- Users get a Razor-pay payment method page:

Razor Pay Dashboard:

- Admin Dashboard: after the order is confirmed the admin dashboard is updated.

Conclusion:

The Pizza Shop E-Commerce application delivers a cohesive online ordering experience by combining user registration, dynamic pizza menu browsing, shopping cart operations, and secure Razorpay payments within a Django framework. An AI-powered chatbot provides real-time menu assistance, while the custom admin panel lets business owners manage pizzas and monitor orders effortlessly. Built on Django's MVT architecture with environment-based configuration and robust security measures, the project ensures both a seamless customer journey and a maintainable codebase. With room for future enhancements like user reviews or notifications, this solution stands ready for real-world deployment.