Full Stack Python With AI Integration and AWS Deployment

Mini Project 1

Module: Django Basics
Project Title: Shopping Cart Functionality

Project Overview
Build the core cart system of an e-commerce site where users can browse products, add them to a cart, view contents, and remove items.

Objective
Students will learn:

- Creating product and cart models
- Adding and removing cart items
- Using Django sessions for cart persistence
- Displaying cart totals

Scope

- Product listing page with "Add to Cart" button
- Cart page showing products, quantities, and total price
- Remove item functionality
- Session-based cart persistence

Technical Requirements

- Language: Python 3.x
- Framework: Django 4.x+
- Database: SQLite / PostgreSQL
- Tools: VS Code, Django Admin, Browser DevTools

Mini Project 2

Module: Django Authentication
Project Title: User Registration & Login

Project Overview
Integrate Django's authentication system so users can register, log in, log out, and manage their profile.

Objective
Students will learn:

- Django's built-in User model
- Creating registration and login forms
- Session-based user management
- Restricting cart and checkout pages to logged-in users

Scope

- Registration page
- Login/Logout functionality
- Profile management page
- Authentication checks for cart and checkout

Technical Requirements

- Language: Python 3.x
- Framework: Django 4.x+
- Database: SQLite / PostgreSQL
- Tools: VS Code, Django Admin, Browser DevTools

Mini Project 3

Module: Django Advanced Features
Project Title: Checkout & Order Management

Project Overview
Allow users to checkout, save orders in the database, and view purchase history.

Objective
Students will learn:

- Designing order models
- Checkout forms for shipping/billing
- Storing order items linked to users
- Displaying past orders in user profiles

Scope

- Checkout page
- Store orders and ordered items
- Order confirmation page
- Order history page

Technical Requirements

- Language: Python 3.x
- Framework: Django 4.x+
- Database: SQLite / PostgreSQL
- Tools: VS Code, Django Admin, Browser DevTools

Mini Project 4

Module: API Development
Project Title: E-Commerce REST API

Project Overview
Expose the e-commerce backend through REST APIs so it can be consumed by a frontend.

Objective
Students will learn:

- Django REST Framework basics
- Creating endpoints for products, cart, orders, and authentication
- Using JWT authentication
- Configuring CORS for cross-origin requests

Scope

- API for product listing, search, and details
- API for cart management
- API for checkout and orders
- User authentication via JWT

Technical Requirements

- Language: Python 3.x
- Framework: Django 4.x+, Django REST Framework
- Database: SQLite / PostgreSQL
- Tools: VS Code, Postman, Browser DevTools

Mini Project 5

Module: Frontend Development
Project Title: Responsive E-Commerce Frontend

Project Overview
Build a modern frontend (React.js or Django templates) to display products, manage cart, checkout, and interact with the backend API.

Objective
Students will learn:

- Component-based design (React) or Django templates
- Fetching data from backend APIs
- Implementing search and filter
- Handling user interactions like “Add to Cart”

Scope

- Product listing page with search/filter
- Product detail page
- Cart and checkout pages
- Responsive design for mobile/desktop

Technical Requirements

- Language: HTML, CSS, JavaScript / React.js
- Tools: VS Code, Browser DevTools, Axios/Fetch API

Mini Project 6

Module: Full Django E-Commerce Integration
Project Title: Complete E-Commerce Platform

Project Overview
Integrate all the mini projects into a complete Django-based e-commerce website with product management, cart functionality, authentication, and order checkout.

Objective
To combine all developed modules into a functional platform. Students will practice:

- Django full-stack development
- User authentication and authorization
- Connecting frontend and backend
- Handling cart, checkout, and order storage

Scope of the Project

- User registration, login, and profile
- Product listing, creation, and image uploads
- Shopping cart with add/remove items
- Checkout and order confirmation

Technical Requirements

- Language: Python 3.x, HTML, CSS, JavaScript
- Framework: Django 4.x+
- Database: SQLite / PostgreSQL
- Tools: VS Code, Django Admin, Browser DevTools

Mini Project 7

Module: AI Chatbot Integration
Project Title: AI-Powered E-Commerce Support Chatbot

Project Overview
Integrate an AI chatbot into the e-commerce platform to answer customer queries, recommend products, assist with order tracking, and guide users through the shopping process — 24/7. The chatbot will be trained or connected to relevant data such as product catalogs, FAQs, and store policies.

Objective
To implement a conversational AI assistant that can:

- Answer general questions about the platform
- Provide details about specific products from the database
- Assist users with adding products to the cart and checkout process
- Track and update customers about their orders
- Suggest products based on customer preferences and browsing history

Students will learn:

- Using AI APIs (e.g., OpenAI GPT API)
- Integrating chatbot UI in React
- Connecting chatbot to backend data
- Implementing personalized product recommendations
- Handling real-time messaging and order queries

Scope of the Project

- Chatbot interface embedded in React frontend (floating widget)
- Backend API endpoint to handle chatbot queries
- Integration with AI API for language understanding and responses
- Contextual responses by querying for product and order details
- Fallback to FAQ database for unanswered queries
- Recommendation engine integration for upselling and cross-selling

Technical Requirements

- **Language:** JavaScript
- **Stack:** React.js, Django
- **AI API:** OpenAI GPT (or similar LLM API)
  .

Tools:

- React Chatbot UI library (e.g., react-chatbot-kit, Botpress, custom chat widget)
- Postman for testing
- API keys for AI provider
- **Concepts Used:** API integration, NLP basics, recommendation systems, real-time communication

Mini Project 8

Module: Version Control with Git & GitHub

Project Title: Ecommerce fullstack application with AI integartion

Project Overview
Set up Git for the project and push code to GitHub. Students will learn the complete workflow for managing source code in teams, tracking changes, and collaborating.

Objective
To introduce students to Git and GitHub for version control, ensuring:

- Code history tracking
- Collaboration through branches and pull requests
- Handling merge conflicts
- Maintaining a professional project repository

Scope of the Project
Initialize a local Git repository

- Create .gitignore to exclude environment files and dependencies
- Commit and push to GitHub repository
- Create branches for features (e.g., feature/product listing)
- Merge changes via pull requests
- Use GitHub Issues for task tracking

Technical Requirements

- Tools: Git, GitHub, VS Code
- Concepts Used: Git basics, branching, merging, pull requests, issue tracking

Mini Project 9

Module: Cloud Deployment
Project Title: Deploying E-Commerce Platform to the Cloud

Project Overview
Deploy the complete Python + AI Chatbot **e-commerce application** to a cloud platform, making it accessible to customers worldwide. Students will configure environment variables, connect to a live database, and ensure production readiness for smooth shopping, order tracking, and chatbot interactions.

Objective
To host the application on a cloud platform so users can:

- Access the React-based e-commerce storefront via a public URL
- Use the backend API live for product search, cart, and checkout operations
- Interact with the AI chatbot in real-time for product queries and order tracking

Students will practice:

- Building the frontend for production deployment
- Deploying backend with cloud hosting services
- Connecting to live database for live product and order data
- Managing secrets and environment variables securely in the cloud

Scope of the Project

- Deploy React frontend (Vercel, Netlify, or similar)
- Deploy Django backend (Render, Railway, AWS EC2, or similar)
- Connect backend to for real-time product/order updates
- Configure API keys for AI chatbot securely
- Test live application with Postman and browser

Technical Requirements

Language: Python (Django)
Tools:

- GitHub (for repository hosting)
- Vercel/Netlify (frontend hosting)
- AWS / Render / Railway (backend hosting)
- Mysql / Mongodb (database)
- AI API provider (OpenAI or similar)
- **Concepts Used:** CI/CD basics, environment configuration, secure API key management, cloud hosting