**Aim:**

To design and develop a dynamic AI-powered chatbot for eCommerce platforms that enhances user experience by providing real-time product recommendations, customer support, and seamless transaction assistance.

**Description:**

In the modern digital marketplace, customer engagement and satisfaction are critical factors that drive business success. With the growing complexity of customer needs and the demand for instant responses, businesses are turning to artificial intelligence to streamline communication and improve user experience. One of the most powerful tools emerging in this domain is the **Ecommerce Dynamic Chatbot**—a smart virtual assistant designed to transform how customers interact with online stores.

This project involves developing a dynamic AI-powered chatbot tailored specifically for eCommerce platforms. Unlike traditional rule-based chatbots, this solution leverages **natural language processing (NLP)** and **machine learning** to understand user intent, engage in context-aware conversations, and provide intelligent responses. The chatbot is capable of handling a wide range of functions, from guiding users through product searches to managing orders and providing after-sales support.

The chatbot integrates seamlessly with an eCommerce website or app, appearing as an interactive widget or embedded chat interface. It allows customers to ask questions like “What are the best budget smartphones?” or “Where is my order?”, and responds with accurate, real-time information. It can also proactively engage users by suggesting products based on their browsing history, location, and behavior, thus enhancing the overall shopping experience.

The project is built using a modern **tech stack** comprising a React-based frontend for clean and intuitive UI, a Node.js and Express backend for handling requests and business logic, and MongoDB or PostgreSQL for persistent data storage. The intelligence behind the chatbot comes from AI models such as those provided by the **DeepSeek API** or **OpenAI GPT**, combined with libraries like **LangChain** for contextual memory and dynamic prompt handling. Authentication and user tracking can be handled using **Firebase Auth** or **JWT** to ensure secure interactions.

In addition to serving as a digital shopping assistant, the chatbot also collects and analyzes customer interaction data, helping businesses gain valuable insights into user preferences and behavior. This data can be used to refine marketing strategies, personalize recommendations, and optimize inventory management.

Overall, the Ecommerce Dynamic Chatbot not only boosts operational efficiency but also drives customer loyalty, reduces response time, and helps businesses scale their support without increasing human resource costs.

**Technology Stack:**

**Frontend:**

HTML,JAVASCRIPT

**Backend:**

Python,open router API

**Impact:**

* **Enhanced Customer Experience:** Offers 24/7 real-time assistance, improving satisfaction and retention.
* **Increased Sales:** Personalized recommendations help boost conversion rates.
* **Reduced Operational Cost:** Automates common queries, reducing the need for large support teams.
* **Data Insights:** Collects valuable customer interaction data to optimize marketing and inventory strategies.

**PROGRAM CODE:**

**from flask import Flask, request, jsonify**

**from openai import OpenAI**

**from flask\_cors import CORS  # To enable frontend-backend communication**

**# Initialize Flask app and CORS**

**app = Flask(\_\_name\_\_)**

**CORS(app)**

**# Initialize OpenAI Client with DeepSeek via OpenRouter**

**client = OpenAI(**

**base\_url="https://openrouter.ai/api/v1",**

**api\_key="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",  # Replace with your key**

**)**

**# Function to query DeepSeek model**

**def get\_response\_from\_deepseek(user\_message):**

**completion = client.chat.completions.create(**

**extra\_headers={**

**"HTTP-Referer": "http://localhost:5500",  # Optional, for OpenRouter stats**

**"X-Title": "Smart E-Commerce Assistant"**

**},**

**model="deepseek/deepseek-chat",**

**messages=[**

**{"role": "user", "content": user\_message}**

**]**

**)**

**return completion.choices[0].message.content**

**# Chatbot route**

**@app.route("/chat", methods=["POST"])**

**def chat():**

**data = request.get\_json()**

**user\_message = data.get("message", "")**

**if not user\_message:**

**return jsonify({"reply": "Please enter a message!"})**

**reply = get\_response\_from\_deepseek(user\_message)**

**return jsonify({"reply": reply})**

**# Run server**

**if \_\_name\_\_ == "\_\_main\_\_":**

**app.run(debug=True, port=8000)**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<title>Smart E-Commerce Chatbot</title>**

**<link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">**

**<style>**

**body {**

**font-family: 'Roboto', sans-serif;**

**background: linear-gradient(to right, #e3f2fd, #ffffff);**

**display: flex;**

**justify-content: center;**

**align-items: center;**

**height: 100vh;**

**margin: 0;**

**}**

**.chat-container {**

**width: 400px;**

**background: #ffffff;**

**border-radius: 15px;**

**box-shadow: 0 4px 20px rgba(0, 0, 0, 0.2);**

**display: flex;**

**flex-direction: column;**

**overflow: hidden;**

**}**

**.chat-header {**

**background-color: #007bff;**

**color: white;**

**padding: 15px;**

**text-align: center;**

**font-size: 1.3rem;**

**}**

**.chat-box {**

**padding: 15px;**

**height: 400px;**

**overflow-y: auto;**

**background-color: #f1f3f5;**

**}**

**.chat-message {**

**margin: 10px 0;**

**}**

**.user-message {**

**text-align: right;**

**color: #007bff;**

**}**

**.bot-message {**

**text-align: left;**

**color: #333;**

**}**

**.chat-input {**

**display: flex;**

**border-top: 1px solid #ccc;**

**}**

**.chat-input input {**

**flex: 1;**

**padding: 15px;**

**border: none;**

**font-size: 1rem;**

**outline: none;**

**}**

**.chat-input button {**

**background-color: #007bff;**

**color: white;**

**border: none;**

**padding: 15px 20px;**

**font-size: 1rem;**

**cursor: pointer;**

**}**

**</style>**

**</head>**

**<body>**

**<div class="chat-container">**

**<div class="chat-header">Smart E-Commerce Assistant</div>**

**<div class="chat-box" id="chat-box">**

**<div class="chat-message bot-message">Hi! I'm your shopping assistant. Ask me anything.</div>**

**</div>**

**<div class="chat-input">**

**<input type="text" id="user-input" placeholder="Type your message..." />**

**<button onclick="sendMessage()">Send</button>**

**</div>**

**</div>**

**<script>**

**async function sendMessage() {**

**const input = document.getElementById("user-input");**

**const message = input.value.trim();**

**const chatBox = document.getElementById("chat-box");**

**if (!message) return;**

**// Show user message**

**chatBox.innerHTML += `<div class="chat-message user-message">${message}</div>`;**

**input.value = "";**

**chatBox.scrollTop = chatBox.scrollHeight;**

**// Send message to backend**

**const response = await fetch("http://localhost:8000/chat", {**

**method: "POST",**

**headers: { "Content-Type": "application/json" },**

**body: JSON.stringify({ message: message })**

**});**

**const data = await response.json();**

**// Show bot reply**

**chatBox.innerHTML += `<div class="chat-message bot-message">${data.reply}</div>`;**

**chatBox.scrollTop = chatBox.scrollHeight;**

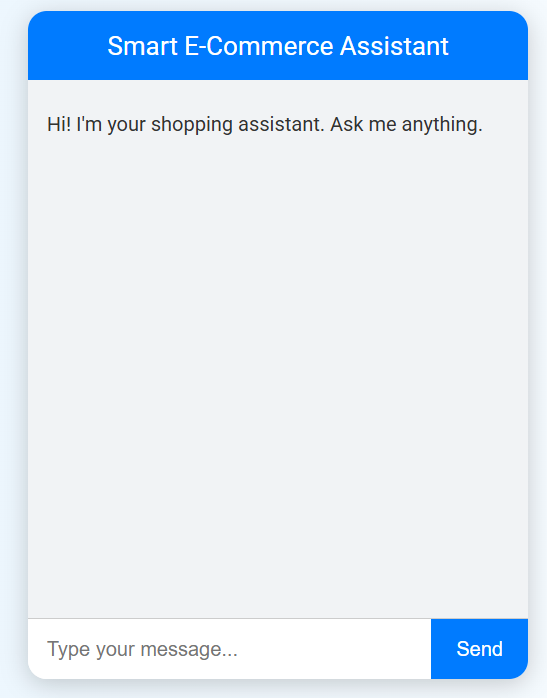
**}**

**</script>**

**</body>**

**</html>**

**SCREENSHOTS:**



A screenshot of a social media post

AI-generated content may be incorrect.

A screenshot of a social media post

AI-generated content may be incorrect.

**RESULT:**

Thus, we Developed EcoBot, an AI-powered chatbot offering personalized waste management guidance for sellers, buyers, upcyclers, and volunteers.

Integrated AI image recognition and NLP to support waste classification, eco-friendly practices, and sustainable logistics was implemented successfully.