Gyaanify - an azure ai chatbot

Developed during a 30-day Artificial Intelligence Training Program, co-organized by Rockwell Automation and the Training & Placement Cell, KIET Group of Institutions, Muradnagar, for the Batch of 2022-2026.

INTRODUCTION TO THE AZURE AI CHATBOT - A HELPFUL ASSISTANT POWERED BY AZURE OPENAI

The Azure AI Chatbot is an intelligent virtual assistant designed to interact with users in natural language. Powered by Azure OpenAI, it leverages advanced language models to provide fast, accurate, and human-like responses. Whether answering questions, assisting with tasks, or guiding users through processes, the chatbot ensures a seamless conversational experience. Built using Flask for the backend and a responsive web frontend, it serves as a robust example of AI integration in modern applications.

Problem & Solution

PROBLEM

In many web applications, users often struggle to get quick, relevant, and human-like assistance. Static FAQ sections and rigid support systems fail to provide conversational, contextual, and intelligent interaction — especially in academic, customer support, and information-driven environments.

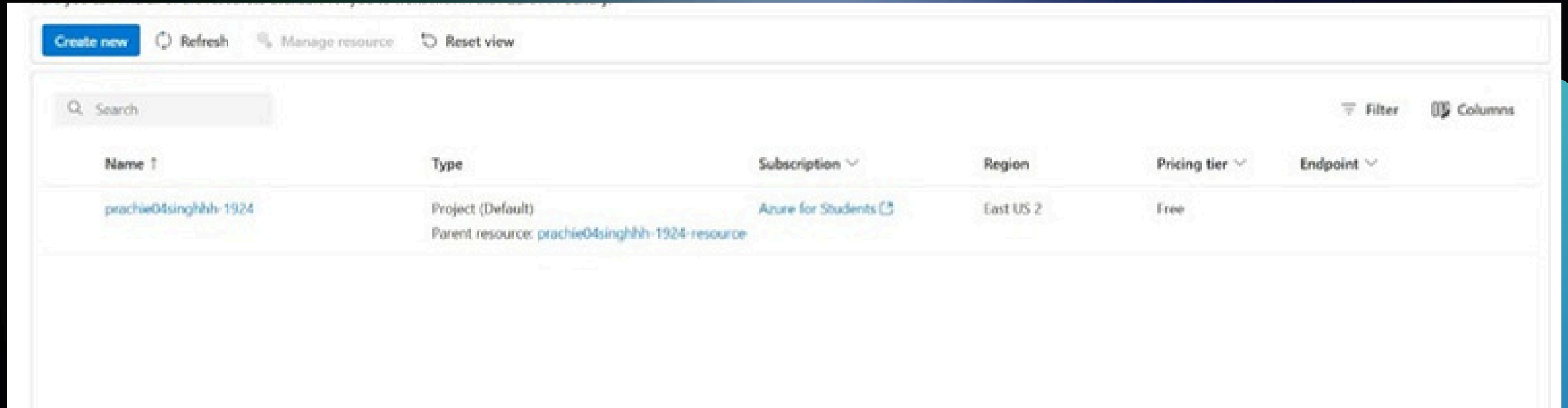
SOLUTION

The Azure AI Chatbot offers an intelligent, real-time conversational assistant powered by Azure OpenAI's GPT models. It enables users to ask questions naturally and receive relevant, context-aware answers through a clean, responsive web interface. This chatbot reduces dependency on manual support, enhances user engagement, and serves as a scalable solution for automated interactions.

Project Workflow - Setting Up the Environment

Creating a Resource Group:

- Go to the Azure portal and select "Create a Resource" under Azure Multi-service Account.
- Choose "Resource Group" and provide a unique name.
- Select the appropriate region and click "Review + create".



About the Chatbot

01

Regular Meetings

Plan meetings to discuss progress, resolve issues, and align team efforts.

02

Project Management Tools

Use platforms like Slack or Microsoft Teams for quick updates and collaboration.

03

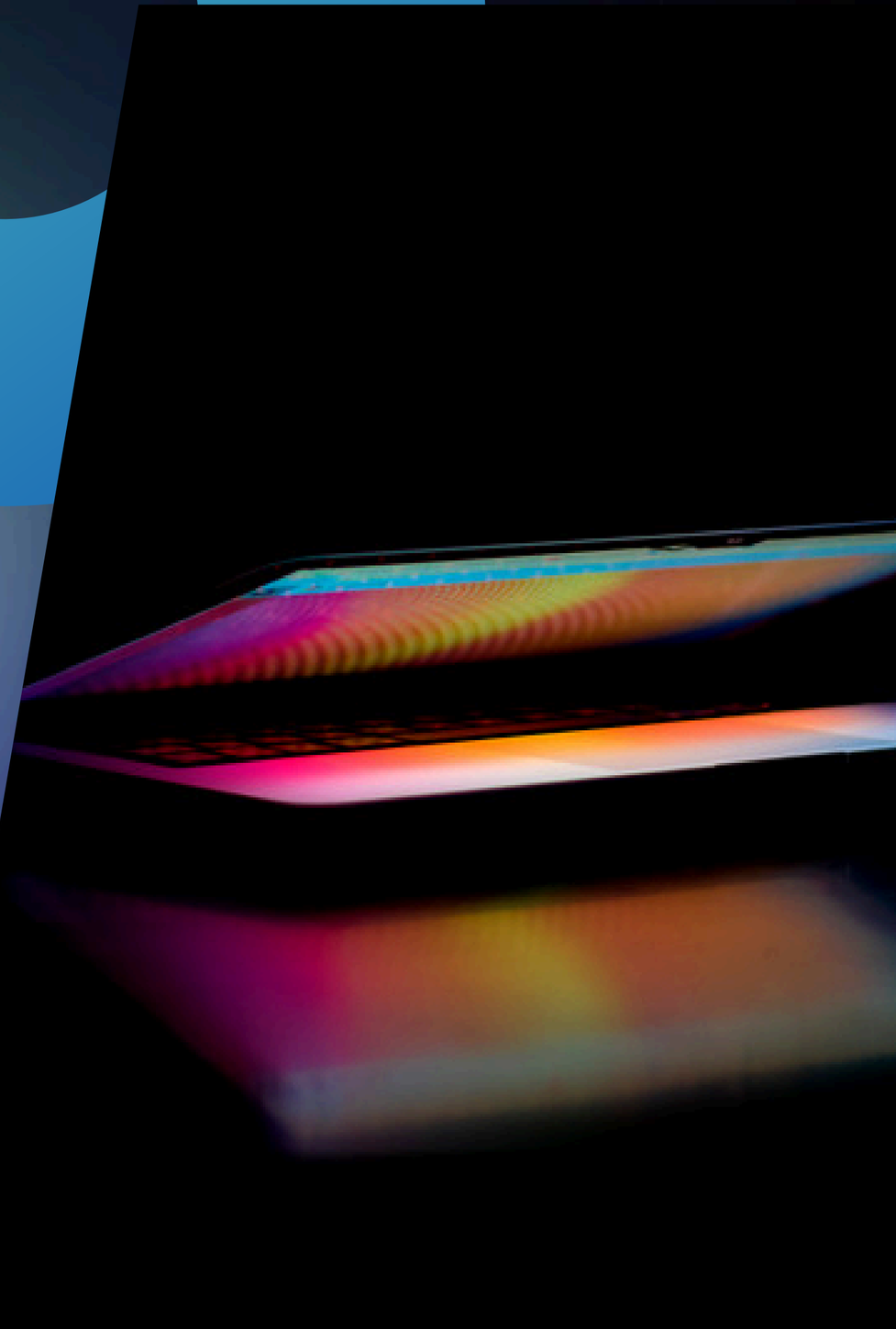
Instant Messaging

Use platforms like Slack or Microsoft Teams for quick updates and collaboration.

04

Email

Maintain a formal channel for documentation and shared information.



✨ Key Features of the Chatbot

💬 Contextual Chat	⚡ Fast & Responsive	🌐 Web-Based Interface	🔒 Secure Integration
Understands and remembers recent conversation flow for relevant and coherent replies.	Delivers real-time responses powered by Azure OpenAI with minimal latency.	Accessible via browser, with a clean and interactive user experience.	Uses .env for secure credential management and Flask-based API calls.

Technology Stack

Backend (Server-Side)

- Flask – Lightweight Python web framework for API development
- Flask-CORS – Enables secure communication between frontend and backend
- OpenAI (Azure SDK) – Integrates GPT-based models via Azure OpenAI API
- dotenv – Loads API keys and environment variables securely from .env
- logging – Handles request logs and error tracking for debugging

Frontend (Client-Side)

- HTML5 & CSS3 – Builds the chatbot's structure and aesthetic UI
- Vanilla JavaScript – Manages user input, message rendering, and interactions
- Fetch API – Handles POST requests to communicate with the Flask backend

AI/LLM Integration

- Azure OpenAI Service – Hosts GPT-4o or GPT-3.5 models via secure API
- Deployment Name – Uses named model deployment (e.g., gpt-4o) from Azure portal

Architecture Diagram

- **User Interaction**

The user enters a message into the web-based chatbot interface built with HTML, CSS, and JavaScript.

- **Frontend Sends Request**

The message is sent to the Flask backend using the Fetch API through a POST request to the /chat endpoint.

- **Backend Handles Processing**

Flask receives the request, appends it to the conversation history, and prepares it for the AI model.

- **Azure OpenAI Integration**

The backend securely sends the conversation to the deployed GPT model (e.g., GPT-4o) using the Azure OpenAI SDK.

- **Response from AI Model**

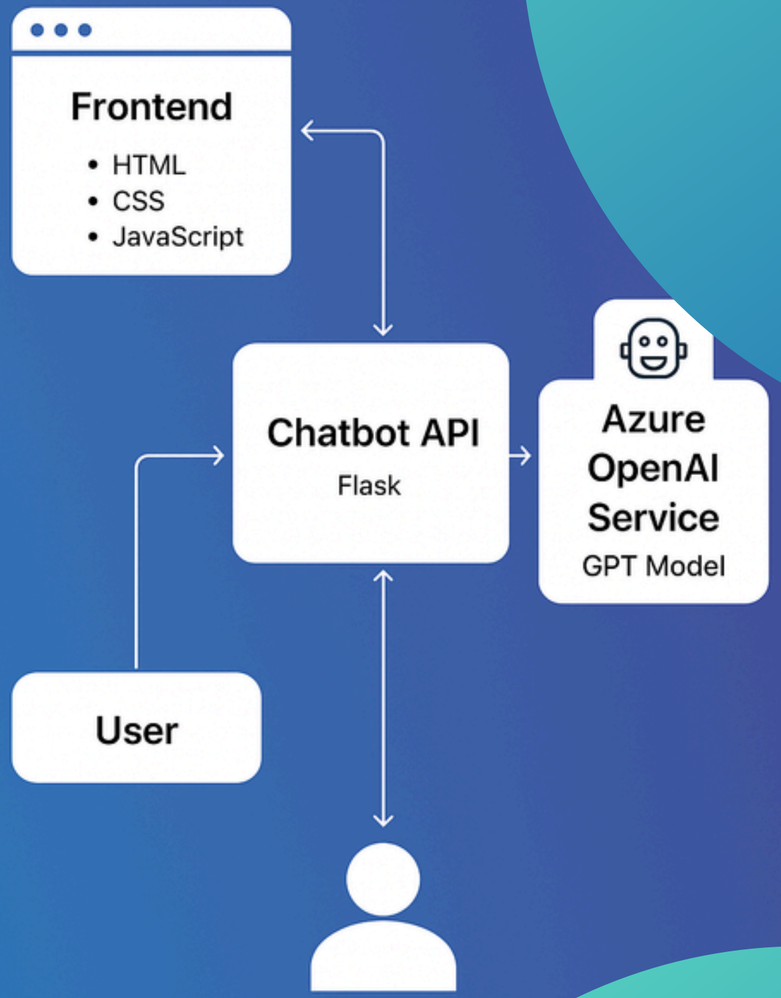
Azure processes the input and generates a relevant, context-aware response.

- **Backend Returns Response**

The Flask backend receives the AI response and sends it back as JSON.

- **Frontend Displays Reply**

The JavaScript frontend updates the chat interface with the bot's reply in real-time.



Code Summary: Python (Flask) & JavaScript (Frontend)

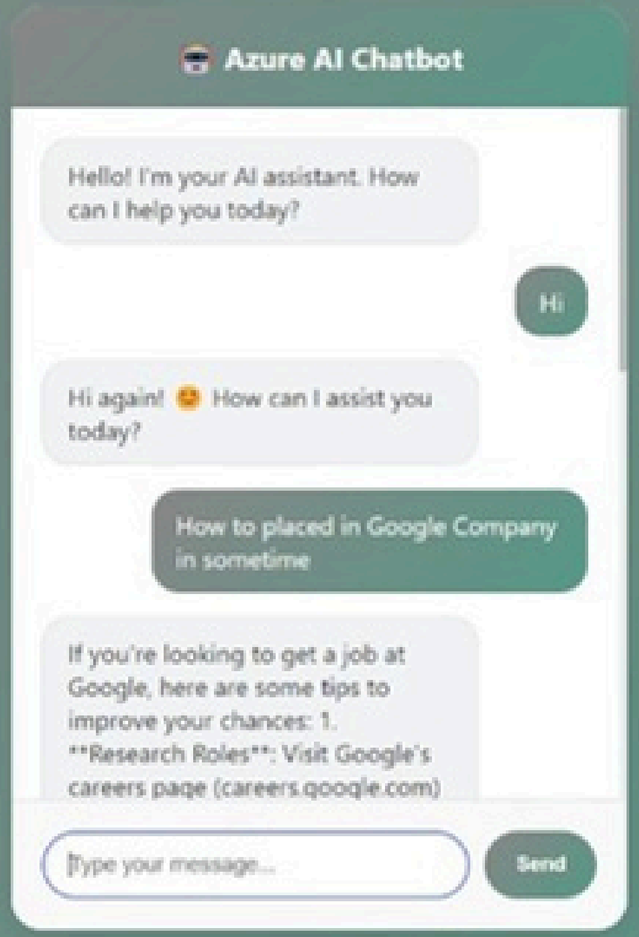
Python (Flask Backend)

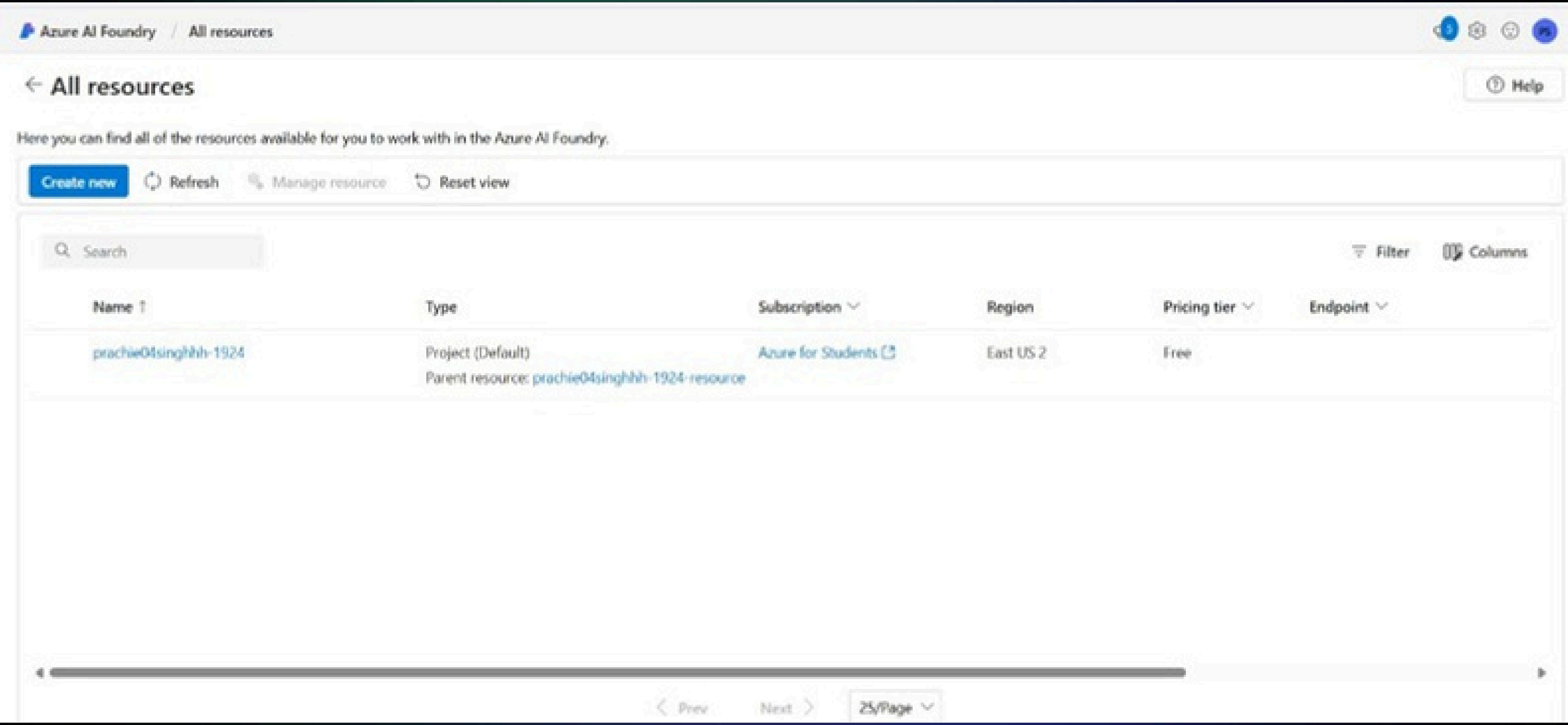
- Handles API Requests
- Receives user input via /chat endpoint using POST requests.
- Maintains Conversation Context
- Stores message history (last 10 messages) per user/IP for continuity.
- Connects to Azure OpenAI
- Sends conversation history to deployed GPT model (e.g., GPT-4o) via SDK.
- Processes & Returns Responses
- Extracts AI-generated reply and sends it back as JSON.
- Other Endpoints
- /clear clears chat history; /health checks API key configuration.

JavaScript (Frontend)

- Captures User Input
- Listens for “Send” button click or Enter key to get the user message.
- Uses Fetch API
- Sends the message to the Flask backend via POST /chat.
- Updates Chat UI Dynamically
- Adds user and bot messages to the chat container in real time.
- Typing Indicator
- Shows animated typing dots while waiting for the bot’s response.
- Error Handling
- Displays friendly messages for server errors or connection issues.

RESULT


Azure AI Chatbot
Hello! I'm your AI assistant. How can I help you today?
Hi
Hi again! How can I assist you today?
How to placed in Google Company in sometime
If you're looking to get a job at Google, here are some tips to improve your chances: 1. **Research Roles**: Visit Google's careers page (careers.google.com)
Type your message...
Send


Azure AI Foundry / All resources
All resources
Help
Here you can find all of the resources available for you to work with in the Azure AI Foundry.
Create new
Refresh
Manage resource
Reset view
Search
Filter
Columns
Name
Type
Subscription
Region
Pricing tier
Endpoint
prachie04singhhh-1924
Project (Default)
Parent resource: prachie04singhhh-1924-resource
Azure for Students
East US 2
Free
Prev
Next
25/Page

Statistics & Future Scope

📈 Statistics / Industry Insights

- 80% of businesses plan to integrate chatbots by 2027 to improve customer engagement and reduce workload. (Gartner)
- AI chatbot market is projected to grow from $5.4B in 2023 to $15.5B by 2028.
- Chatbots can handle 70% of customer queries without human intervention, improving efficiency.
- In education, AI assistants are being adopted by over 40% of universities for student support systems.

🚀 Future Scope

- Multilingual Support

Integrate translation APIs for regional and global accessibility.

- Voice Input/Output Integration

Add speech-to-text and text-to-speech for a voice-enabled experience.

- Database Connectivity

Connect to real datasets to provide personalized, data-driven responses.

- Domain-Specific Knowledge

Train the chatbot for academic help, healthcare FAQs, or HR onboarding.

- Mobile App Version

Convert the chatbot into a cross-platform mobile app for wider reach.

Analytics Dashboard

- Track user queries, response time, satisfaction rating, and usage trends.

🗂️ About the Dataset

The Azure AI Chatbot does not use a traditional static dataset. Instead, it leverages a pre-trained Large Language Model (LLM) via Azure OpenAI Service, which is already trained on vast amounts of publicly available internet data up to its cutoff.

📌 Key Points:

- **Model-Based Knowledge**

The chatbot uses Azure-hosted models like GPT-4o or GPT-3.5, which have been pre-trained on diverse datasets including books, websites, forums, and articles.

- **No Local Dataset Required**

Since the intelligence comes from the cloud-based LLM, no additional dataset needs to be uploaded or managed locally.

- **Contextual Input**

Instead of relying on a fixed dataset, the bot responds based on real-time user input and maintains short-term conversation history (last 10 messages) for continuity.

- **Security Note**

No user data is permanently stored; conversations exist in memory temporarily and can be cleared via the reset endpoint.

We extend our heartfelt gratitude to Rockwell Automation and the Training & Placement Cell, KIET Group of Institutions, Muradnagar for organizing the 30-day AI Training Program.
Special thanks to our mentor, Mr. Rohit Ranjan, for his constant guidance and support throughout the project.
This chatbot project is a result of collaborative learning, innovation, and teamwork by the 2022-2026 batch.

Thank You

👥 **Team Members**

- Nandan Mishra
- Jyoti Kumari
- Juhi Kumari
- Prachi Singh