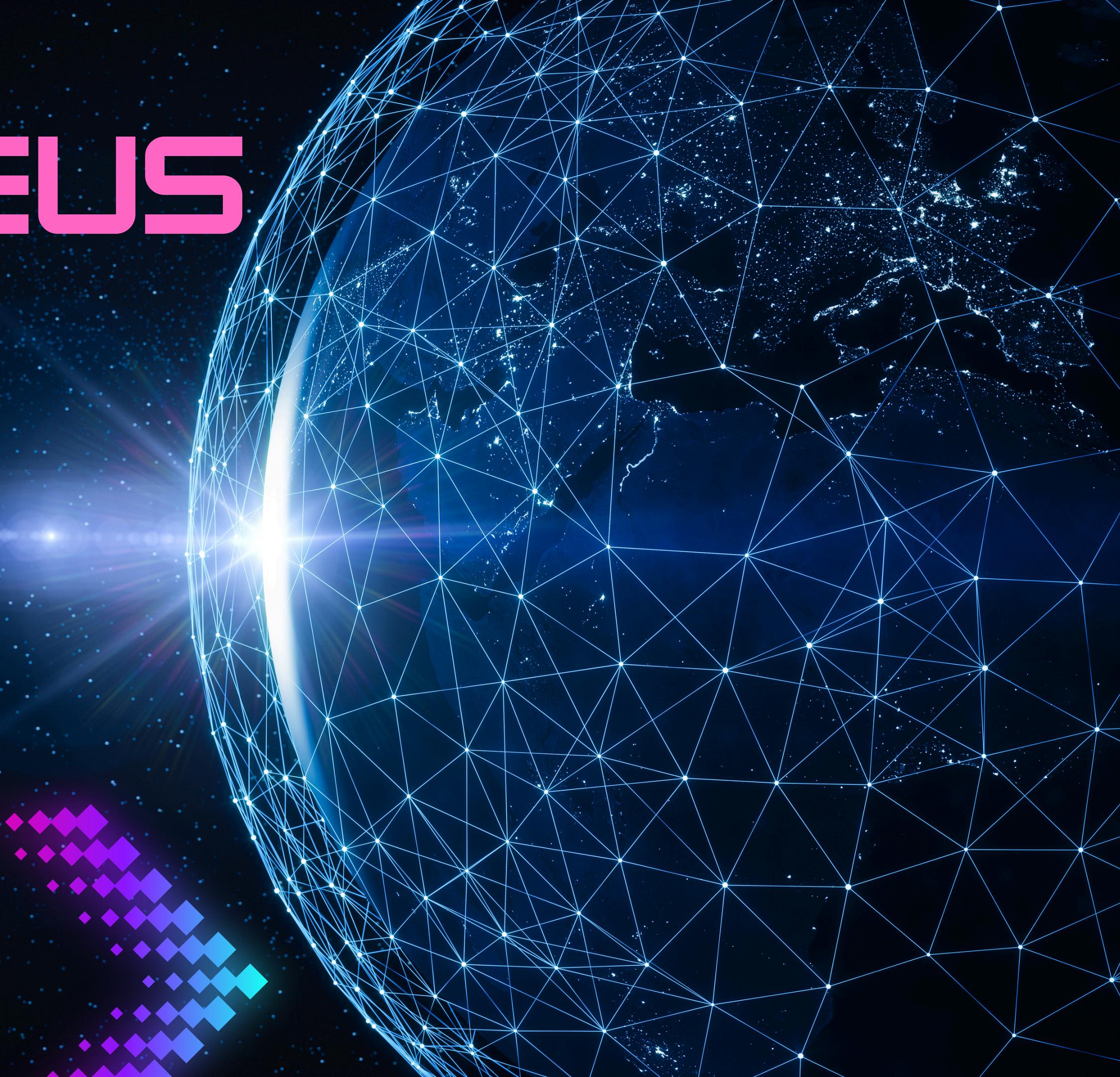


# CLODOXEUS

## Description and Requirements



# INTRODUCTION

## Company Overview

CloudXeus is a fast-growing edtech company that specializes in delivering high-quality training programs for Microsoft Azure and cloud certifications. With a global audience of learners, CloudXeus has trained over 500,000 students in areas such as cloud computing, AI, data engineering, and DevOps.

The company's mission is simple:  
**“Empower professionals with the skills and confidence to pass cloud certification exams and accelerate their careers.”**



# BUSINESS CHALLENGES

## Building a Knowledgebase

- As the company continues to grow, CloudXeus faces increasing demand on its support team. Every day, students ask questions such as
  - How do I access my course?”
  - “Do you offer practice exams for AZ-104?”
  - “What is the difference between AZ-900 and AZ-305?”
- Most of these are repetitive FAQ-style questions, which takes up valuable support time that could be better spent helping students with complex issues.

## Building a review system

- CloudXeus receives feedback every day—from student course reviews, support tickets etc.
- Today, this feedback is stuck in raw text, making it difficult to analyze trends or take data-driven action.
- The sentiments need to be consolidated, and be searchable in nature.

# BUSINESS CHALLENGES

## Using LLM's

- CloudXeus is exploring how different generations of GPT models (GPT-4 and GPT-5) perform on real-world student queries.
- While a single model can provide answers, organizations increasingly need to compare model performance side by side to:
- Understand differences in accuracy, creativity, and cost.
- Evaluate token consumption for budgeting purposes.
- Identify which model is better suited for specific business use cases.

## Course Catalog Search

- As CloudXeus expands its online training platform, the number of courses and supplementary materials (PDFs, images, videos) keeps growing.
- Currently, the site only provides basic keyword search over course names. This makes it hard for learners to discover content—especially when it lives in supporting files like PDFs, videos, or images.

# PROPOSED SOLUTIONS

## Building a Knowledgebase

- CloudXeus wants to implement a chatbot powered by Azure AI Language to serve as the first line of support. This chatbot should:
  - Provide instant answers to frequently asked questions.
  - Be easy to update as new courses and policies are introduced.
  - Act as a foundation for future expansion into more advanced conversational AI solutions.



# PROPOSED SOLUTIONS

## Building a Knowledgebase

- To test the waters, CloudXeus has decided to build a pilot project:
  - Knowledge Base → Create a set of 20+ FAQs about the platform and courses.
  - Azure Language Studio → Host and publish the knowledge base.
  - Python Chatbot → Build a simple interface to interact with the knowledge base.



# LEARNING OBJECTIVES

1

## Understand Azure AI Language Question Answering

- Learn what a knowledge base is and how it powers a QnA bot.
- Create and manage a QnA project in Language Studio.

2

## Build a Knowledge Base

- Collect and organize FAQs into a structured format.
- Import questions and answers into Azure Language Studio using a file

3

## Build a chatbot

- Build a Python program that can interact with the knowledge base.
  - Create a bot-like interface for users.
- Build the interface in React.js that will call the Python program.

4

## Deploy the solution

- Deploy the python module using Azure Functions .
- Host the chatbot interface on Azure as a static web site. We will use Azure Storage accounts.

# PROPOSED SOLUTIONS

## Building a review system

- CloudXeus will build a sentiment enrichment pipeline using Azure AI Search + Azure AI Services, where every sentence of student feedback is automatically analyzed, categorized, and projected into Azure Table Storage for analytics. A Pilot project will be developed that will entail the following:
  - Ingest Feedback → Upload student reviews (CSV/TXT/JSON) into Azure Blob Storage.
  - Azure AI Search Indexer → Pulls documents and cracks them into text.
  - Add an enrichment pipeline to understand the sentiment of each text.
  - Project the sentiments onto Azure Table storage.



# LEARNING OBJECTIVES

1

## Design an Enrichment Pipeline

- Understand how to use the built-in skills to build the enrichment pipeline.
- Understand how skill outputs flow through the enrichment tree.

2

## Understand Azure Search

- Understand each element when it comes to Azure Search.
- Use JSON all the way - Build the index, indexer and Skillset.

3

## Projections

- Understand again via JSON configuration how to define projections.

# PROPOSED SOLUTIONS

## Using LLM's

- CloudXeus will build a side-by-side model comparison chat interface powered by Azure AI Foundry. This system will:
  1. Deploy Multiple GPT Models
    - Create two deployments in Azure AI Foundry: one for GPT-5 and one for GPT-4.
  2. Develop a Python Backend Service
- Implement a Python module that:
  - Sends user/system prompts to both models.
  - Returns responses with token usage details.
  - Supports conversational follow-ups in ongoing chats.
  - Integrates Azure Content Safety checks for every request.
  - Package this service in a Docker container and deploy it to an Azure Virtual Machine.



# PROPOSED SOLUTIONS

## Using LLM's

### 3. Create a Chat-Like Frontend

- Develop a React.js web interface with two side-by-side panels:
  - One panel shows GPT-4 responses.
  - The other panel shows GPT-5 responses.
- Deploy the frontend as a Static Web App on Azure Storage.

### 4. Provide Insights on Model Behavior

- Students will directly compare model outputs.
- Track token usage to understand cost tradeoffs.
- Learn how Content Safety filters integrate into real solutions.



# LEARNING OBJECTIVES

1

## Understand Multi-Model Deployments in Azure AI Foundry

- Learn how to create and manage multiple GPT model deployments.
- Explore versioning and deployment naming conventions.

2

## Build a Python AI Backend

- Use the Azure OpenAI SDK to send prompts to multiple models.
- Parse responses, capture metadata

3

## Package and Deploy with Docker

- Containerize the Python service.
- Deploy and run the container on an Azure Virtual Machine.

4

## Develop a Chat Interface

- Create a simple UI with dual panels for side-by-side comparison.
- Connect the frontend to the Python backend via REST APIs.

# PROPOSED SOLUTIONS

## Course Catalog Search

CloudXeus wants to implement an intelligent search system powered by Azure AI Search. This solution should:

- Allow students to search across course metadata (names, descriptions, categories, prices).
- Extract and index content from PDFs and images for deeper discoverability.
- Extract information from an Azure Cosmos DB database.



# LEARNING OBJECTIVES

1

## Setup the data store

- Setup Azure Cosmos DB with data.
- Setup Azure Storage Accounts with supplementary data.

2

## Configure Azure AI Search

- Create data sources, indexes, and indexers.
- Apply semantic search and cognitive skills for enrichment.

3

## Integrate Azure AI Video Indexer

- Extract transcripts from video content.
- Use the transcripts in the search.

4

## Multiple indexes

- Understand how to use multiple indexes.
- Here we need to index through Azure storage and Azure Cosmos DB.

# THANK YOU



So let's get started.