EduTutor AI: Personalized

Learning with Generative AI and LMS Integration

Introduction

* - Overview of EduTutor AI
* - Importance of personalized learning
* - Role of generative AI and LMS integration

System Architecture

* - Overview of the system's components
* - Interaction of different modules
* - Include system diagram (if available)

Modular AI-Powered Architecture

* - Explanation of modular design
* - Flexibility and scalability
* - Benefits in AI-powered applications

Project Flow

* - Introduction to milestone-based flow
* - Agile and iterative development approach

Milestone 1 - Requirements

Specification

* - Gathering system and functional requirements
* - Stakeholder involvement
* - Tools used (documentation, diagrams)

Milestone 2 - Initialization of

Environment Variables

* - Setting up the development environment
* - Environment variables for security and configuration
* - Tools used (Docker, .env files, etc.)
* from ibm\_watsonx\_ai import WatsonxAI
* ai = WatsonxAI(api\_key='YOUR\_KEY')
* response = ai.generate(prompt="Explain Newton's Laws")

Milestone 3 - AI Integration with

IBM Watsonx

* - Using IBM Watsonx for AI capabilities
* - How integration enhances learning
* - APIs and services used
* OPENAI\_API\_KEY=your\_key
* PINECONE\_API\_KEY=your\_key
* GOOGLE\_CREDENTIALS=path\_to\_credentials.json

Milestone 4 - Google Classroom

Sync

* - Synchronization with Google Classroom
* - Benefits for educators and students
* - API usage and challenges
* from googleapiclient.discovery import build
* service = build('classroom', 'v1', credentials=creds)
* courses = service.courses().list().execute()

Milestone 5 - Pinecone Vector DB

Integration

* - Role of Pinecone as a vector database
* - Managing embeddings and search
* - Use case in personalized recommendations
* import pinecone
* pinecone.init(api\_key="your\_key", environment="us-west1-gcp")
* index = pinecone.Index("student-progress")

Milestone 6 - Streamlit Frontend UI

* - Building a user-friendly frontend
* - Streamlit for rapid UI development
* - Example UI elements and features
* import streamlit as st
* st.title("EduTutor AI")
* st.text\_input("Ask your question:")

Conclusion

* - Summary of project architecture and progress
* - Real-world applications of EduTutor AI
* - Future improvements and scalability

References

* - IBM Watsonx Documentation
* - Google Classroom API
* - Pinecone Vector DB Docs
* - Streamlit Official Docs
* - Internz Platform Resources
* - Relevant research articles