# ****Christian Mpabuka****

# ****ITAI 2376 Deep Learning in Artificial Intelligence****

# ****Professor: Anna Devarakonda****

# AI Agent Creation Capstone Project Proposal

# 1. Problem Statement

Many students struggle to learn programming because they lack personalized guidance. Traditional learning platforms often offer static content that doesn't adjust to individual needs. This project aims to create an AI-powered interactive learning companion that adapts to each student's progress, provides customized exercises, and offers immediate feedback to improve learning results.

**2. Project Option**

Selected Option: Option 4 - Interactive Learning Companion

# 3. Agent Design

The agent will follow a modular architecture:

* Input Processing: Natural language understanding to interpret student queries.
* Memory System: Learner profiles stored in a vector database (e.g., FAISS).
* Reasoning Component: Chain-of-Thought reasoning for hint generation and Planning-then-Execution forlearning path adjustments.
* Output Generation: Personalized exercises, feedback, and explanations using LLMs.

# 4. Tool Selection

Tool 1: Python Code Execution Environment

* Used to evaluate student code and give feedback.

Tool 2: Vector Database (e.g., FAISS or ChromaDB)

* Used to store and retrieve learner history and relevant content.

# 5. Development Plan

|  |  |
| --- | --- |
| **Date** | **Milestone** |
| July 22 | Research, Proposal |
| July 24 | Basic Architecture, tool integration |
| July 27 | Prototype with feedback loop |
| August 2 | Testing and refinement |
| August 5 | Final Testing, report, and video |

# 6. Evaluation Strategy

* Quantitative: Accuracy of code evaluation, hint effectiveness, and learning gains (pre/post quizzes).
* Qualitative: User satisfaction surveys and feedback.
* Adaptability: How well the agent adjusts to difficulty based on performance.

# 7. Resource Requirements

* Platform: Google Colab
* Libraries: Transformers, FAISS, OpenAI API, Gradio
* Hardware: GPU access (optional for LLM inference)

# 8. Risk Assessment

|  |  |
| --- | --- |
| **Risk** | **Mitigation Strategy** |
| Tool integration failures | Use mock tools and fallback strategies |
| LLM limitations | Break tasks into smaller steps |
| Time constraints | Use checkpoints and Google Drive backups |
| Colab session timeouts | Use checkpoints and Google Drive backups |

### **Conclusion**

This project introduces a transformative approach to programming education by creating an AI-powered interactive learning companion tailored to each student's needs. Using advanced tools such as Python code execution environments, vector databases, and large language models, the system will provide personalized guidance, adaptive exercises, and instant feedback. The modular agent design ensures scalability and flexibility, while the development plan and risk mitigation strategies offer a clear path for successful implementation. Through both quantitative and qualitative evaluation methods, the project aims to demonstrate measurable improvements in learning outcomes and user satisfaction. Ultimately, this initiative seeks to empower students with a more engaging, responsive, and effective way to learn programming skills.