**Aaron Essien - aaroness@buffalo.edu**

**Darius Farngalo - dariusfa@buffalo.edu**

**Milton Morris - morris56@buffalo.edu**

**Oluwaseyi Ogunleye - ooogunle@buffalo.edu**

**Team member names and UB Email**

**Motivation/Introduction** - 2-4 paragraphs describing the reason for your project. This should allow someone not familiar with your problem to understand it enough to see why AI is a suitable approach to helping with it, and should be an introduction to the problem itself.

Has your professor ever given you 100 slides consisting of what seems to be random words? Have you ever struggled with understanding what information to put in your notes from your slides, or staying engaged when transferring your notes from the slides to a separate file? Why waste your time manually inputting all the information on the slides, when you can upload the documents and convert them into interactive studying materials?

Our project seeks to conquer the issue of students having overwhelming slides, (not all of which are useful) and the mundane task of transferring your slides to make it into a more readable and understandable format. Not only will this convert the slides into notes, but our project will also offer interactive components such as creating tests for users to attempt which will adapt to the learning style and pace of the individual user.

**Proposed Approach -** 5-6 paragraphs specifying how you're going to attempt to solve your problem. This is where you'll explain how you'll build your AI solution, what you'll need to create, what you'll be borrowing, and what you'll need to figure out as you go.

Our project involves building an AI tutor that inputs lecture slides, summarizes the main points, and generates quizzes and questions based on the material. The proposed approach can be divided into three main components: slide processing and summarization, question generation, and interactive quiz generation.

*Slide Processing and Summarization*

The first step is processing lecture slides and extracting meaningful content. We plan to use OCR tools, such as Tesseract, to extract text from slide images, especially for PDF-based or image-heavy slides. Once we have the text, we will clean the data by removing irrelevant words or phrases such as headers, footers, and repeated information and will then implement the language models to identify and extract the main points.

*Question Generation*

For the question generation part, we will utilize pre-trained models, such as T5 or GPT-3, on question-answer datasets. The key task here is to convert the summaries into meaningful questions. We plan to generate a variety of questions, such as multiple-choice, true/false, and short-answer types. By leveraging a combination of rule-based methods (for simple questions like factual recall) and transformer-based models (for more complex, conceptual questions), we aim to produce a dynamic range of quiz questions tailored to the material.

*Quiz Generation and User Interaction*

Once the questions are generated, we will create a quiz generation interface that allows users to select the type and difficulty level of the quiz. The quiz system will dynamically generate tests based on the content, and users can interact with the tutor by asking clarifying questions. For this, we will incorporate a conversational agent, potentially leveraging GPT-4, to allow users to query the tutor about specific concepts from the lecture. This would simulate a real tutoring experience where students can dive deeper into topics they find difficult.

*Model Deployment and User Interface*

For the frontend, we will use React to create a user-friendly interface where students can upload lecture slides, view summaries, and interact with the quizzes. We will also ensure that our solution is modular, so it can be extended or modified easily to handle more types of input or different learning environments.

*Challenges and Adaptations*

Throughout the process, we expect to face several challenges, such as improving the accuracy of the summaries and ensuring that the generated questions are of high quality. We will experiment with different models and fine-tuning techniques to ensure that the output is coherent and relevant. Additionally, integrating the question generation with the interactive quiz system will require careful handling of user inputs and responses.

**Proposed Timeline -** This should be a list of the amount of time for each component of your proposed approach. Be realistic! It'll take you more time than you think for each component.

Week 1 (Current week):

* Task: Set up the development environment and version control, gather initial sample data, and begin implementing the text extraction component such as Tesseract OCR or other suitable tool.
* Goal: Ensure text extraction from slides is working, and the environment is set up for further work.

Week 2:

* Task: Implement text processing, and start testing summarization models
* Goal: Produce basic summaries from the lecture slides.

Week 3:

* Task: Refine the summarization model and test its output on a variety of slides. Explore and decide on the model for question generation.
* Goal: Finalize the summarization process.

Week 4:

* Task: Begin implementing the question generation component, fine-tuning T5 or GPT-3 to generate multiple-choice, true/false, and short-answer questions based on the summaries.
* Goal: Generate basic questions from summarized content.

Week 5:

* Task: Integrate the question generation component with the summarization output. Ensure that questions are contextually relevant and accurate.
* Goal: Have a functioning summarization and question generation pipeline.

Week 6:

* Task: Develop a basic frontend interface where users can upload slides and view summaries. Begin integrating the question generation system with the frontend.
* Goal: Build a basic user interface for uploading slides and displaying summaries/questions.

Week 7:

* Task: Finalize quiz generation based on the user-selected difficulty levels and question types. Begin adding the conversational component to allow users to ask questions about the slides.
* Goal: Complete quiz generation and start working on the chatbot interaction.

Week 8:

* Task: Finalize and test the conversational agent (e.g., GPT-4), ensuring it can respond to user queries based on the lecture summaries.
* Goal: Build a functional conversational agent.

Week 9:

* Task: Conduct full-system integration. Ensure all components (text extraction, summarization, question generation, quiz system, conversational agent) are working together.
* Goal: Complete integration, fix bugs, and optimize performance.

Week 10:

* Task: Final testing, user feedback, and refinement. Prepare final documentation and presentation.
* Goal: Polish the system, finalize the project report, and prepare for the presentation.

**References -** If extending some kind of research paper, link the paper. If you're using certain technologies in your solution, list them. If building off of a practical example, like from a video or existing codebase, link them.

* BART (Bidirectional and Auto-Regressive Transformers) for Summarization: https://arxiv.org/abs/1910.13461
* T5 (Text-to-Text Transfer Transformer) for Question Generation: https://arxiv.org/abs/1910.10683
* Tesseract OCR for Text Extraction from Images: https://github.com/tesseract-ocr/tesseract
* Hugging Face Transformers Library for implementing BERT, GPT, and T5 models: https://huggingface.co/transformers/
* GPT-3 for Conversational AI: https://arxiv.org/abs/2005.14165