Report (for Task 2: Quiz Bot)

1. Introduction (Section 1)

- **Objective**: The goal of this project is to develop a Quiz Bot that leverages a Large Language Model (LLM) to generate and assess quiz questions based on network security course materials such as the lecture slides. The bot is designed to operate locally to ensure data privacy, as no data is transmitted over the internet or to third-party services. Instead, the bot runs entirely on a local system using GPT4All, an open-source LLM, which allows us to meet the project requirement of maintaining data privacy. By using GPT4All, we avoid the need for external API calls, which would otherwise compromise the security and privacy of the quiz content.
- **Assigned to**: Nagachandrika Gandra

2. System Framework (Section 2)

- The Quiz Bot is designed to operate locally and generate quiz questions based on network security materials, ensuring data privacy and secure operation within a local environment. The system integrates several key components to process lecture slides, generating random and topic-specific questions, and provide feedback to users.

1. **System Components**:

   The Quiz Bot is designed to operate locally, ensuring that all data and processes remain within the local environment to maintain security and privacy. The key components of the Quiz Bot include several technologies working together. First, GPT4All will be used as the Local Large Language Model (LLM), which handles the quiz questions based on the network security material. This material includes lecture slides. To process and retrieve relevant text, we use Hugging Face's Sentence Transformers to convert the textual content into vector embeddings. Faiss serves as the vector database for storing and retrieving these embeddings efficiently. In addition, SQLite is employed to store the quiz questions and answers in a structured format for easy querying. The user interface, built with Django, allows users to interact with the bot, request quiz questions, and receive feedback. This entire system operates locally, ensuring that no data is transmitted over the internet and that privacy is maintained at all times. Each of these components works together to form a secure environment for the user.

2. **Quiz Generation and Feedback**:

   The quiz generation process involves two types of questions: randomly generated questions and topic-specific questions. The bot uses GPT4All to create both types of questions using the content from the network security lecture slides and textbooks. The Hugging Face Sentence Transformers take the text from these documents, after PyPDF2 converts the pdfs to text, convert it into vector embeddings, and store the embeddings in the vector database which would be the Faiss for easy retrieval. When a user requests a quiz, the system searches the vector database to find the most relevant sections of the documents. Based on the retrieved information, GPT4All formulates the questions. The questions can be either multiple-choice, true/false, or open-ended questions. Once the user answers a question, the bot provides real-time feedback, indicating whether the answer is correct or incorrect. For multiple-choice and true/false questions, the feedback is checked against stored answers in SQLite, while for open-ended questions, the LLM analyzes the response and provides feedback. The bot also assigns a score based on the user's answers.

3. **Data Privacy and Security**:

   One of the primary objectives of this project is to ensure that all data privacy requirements are strictly adhered to. To achieve this, the Quiz Bot is designed to operate entirely within a local environment. This means that no external API calls are made, and no data is sent to external servers or cloud providers. By avoiding cloud services, we eliminate the risks associated with sending sensitive network security materials over the internet. Instead, we rely on GPT4All, which is an open-source LLM that runs directly on the user's machine without the need for internet connectivity. By using GPT4All, all operations, including quiz generation and feedback, occur locally, ensuring that sensitive materials such as network security slides remain secure and private. In addition, the use of a local server environment ensures that the application can operate safely without risking unauthorized access or data breaches. This setup allows us to already mitigate potential privacy risks while maintaining a secure local environment for all operations.

4. **Technologies and Tools**:

   To build this Quiz Bot, we are leveraging a variety of tools and technologies. Python is the core programming language used to implement the logic of the bot, while Django serves as the framework for building the user interface and managing all other components. GPT4All is used as the Large Language Model (LLM) to generate quiz questions locally, without needing internet access or external API calls. Hugging Face's Sentence Transformers are used to process the lecture slides by converting them into vector embeddings. These embeddings are then stored and managed in Faiss, a highly efficient vector database optimized for large-scale similarity searches. SQLite is used to store quiz questions and answers, ensuring that structured data is easily accessible for querying when needed. Additionally, we use PyPDF2 to convert the lecture slides from PDF format into plain text, which is then processed by the Sentence Transformers. These tools collectively form a cohesive system that allows the Quiz Bot to function efficiently while maintaining strict privacy standards.

- **Assigned to**: Anthony Humphreys

3. Data Sources and Training (Section 3)

- **Description**: The Quiz Bot relies on one key data source to generate questions and provide accurate feedback. The primary source for training the bot include lecture slides and other relevant course materials. These materials are converted into machine-readable text using PyPDF2, which extracts the text from PDF documents. Once the text has been extracted, it is transformed into vector embeddings using Hugging Face's Sentence Transformers. This process allows the system to represent the meaning and context of the documents numerically, making it easier to retrieve relevant content when generating quiz questions. The Local Large Language Model (GPT4All) is then trained on these embeddings to generate quiz questions based on the network security content.
- **Assigned to**: Kole Trumble

4. Quiz Generation Process (Section 4)

- **Explanation**: The quiz generation process for the Quiz Bot is designed to create two types of questions: random questions and topic-specific questions. GPT4All plays a critical role in generating these questions by using the network security documents provided as training material. For random question generation, the bot selects topics at random from the vector embeddings stored in the Faiss database. It then formulates questions based on those topics, ensuring that the user is tested on a wide range of material. For topic-specific questions, the user can request a quiz focused on a particular aspect of network security, The bot retrieves relevant information from the Faiss database and generates questions specific to that topic.

- **Assigned to**: Wei Qiu

5. Feedback Mechanism (Section 5)

- **Explanation**: The feedback mechanism of the Quiz Bot is designed to evaluate user answers and provide immediate, accurate responses. For multiple-choice and true/false questions, the bot compares the user's answer with the correct answer stored in the SQLite database. When a user selects an answer, the bot checks the response against the stored data and immediately informs the user whether their answer is correct or incorrect. In the case of multiple-choice questions, the bot can also display the correct answer if the user's response is wrong, providing an opportunity for learning.
- **Assigned to**: Member 5

6. Privacy Considerations (Section 6)

- **Explanation**: We will keep the privacy measures in place for the Quiz Bot, focusing on how data is kept locally to avoid compromising privacy, by keeping the network locally on a local host and avoiding putting the server on a cloud server host. We will avoid using the OpenAi library, since requests from the internet would be made if we used it, so we will be using GPT4ALL since it runs a large language model (LLM) privately on desktops and laptops with no api calls. Since the application operates entirely on a local network, the potential privacy risks are inherently low and already mitigated by the local setup.
- **Assigned to**: Member 6

7. Challenges and Solutions (Section 7)

- **Explanation**: Some of the challenges we encountered was identifying the correct tools to meet the project's requirements. Since using OpenAI would violate the requirement to keep the project entirely local, we opted for GPT4ALL, which runs locally without relying on external API calls. Another challenge was converting the lecture slides and PDFs into a format that GPT4ALL could process. To address this, we decided to use the PyPDF2 library to extract text from PDFs and Hugging Face's Sentence Transformers as the embedding model to convert this text into vector representations. Once we had the vectorized data, we needed a vector database to efficiently store and retrieve these embeddings for GPT4ALL. We chose Faiss as our vector database, which solved the problem of efficiently managing and querying the vector documents for GPT4ALL. This allows GPT4ALL to generate quiz questions and provide feedback based on the retrieved data.
- **Assigned to**: Member 7

8. Conclusion and Future Work (Section 8)

- **Summary**: In this first phase of development, we successfully designed the structure of the Quiz Bot to operate locally, ensuring privacy by using GPT4All and other open-source tools like Hugging Face's Sentence Transformers and Faiss. This foundation sets the stage for generating quiz questions and providing immediate feedback to users. Moving forward, our focus will be on implementing and testing the bot's full functionality, refining the quiz generation process, and expanding the question types. Future improvements may also include optimizing the user experience and exploring more advanced feedback mechanisms.
- **Assigned to**: Member 8

Group Member	**Contributions**
Anthony Humphreys	● Researched and identified the necessary tools for the project. Shared notes with the team to outline the steps for delivering the system and notes defining the project's infrastructure. ● Wrote the system framework section of the project report
Nagachandrika Gandra	● Wrote the introduction section in the project report
Kole Trumble	● Wrote the data sources and training section of the project report
Wei Qiu	● Wrote the quiz generation process section of the project report
Diyva Sai Sre Konatham	● Wrote the feedback mechanism section
Bharath Reddy	● Wrote the privacy considerations section
Adithya Chowdary	● Wrote the challenges and solutions section
Phaninder Reddy Masapeta	● Wrote the conclusion and future works sections