AI-Powered Quiz Generator: Project Report

ELshafey Nadin SIMEAO Guilhem JOSEPH Remy
GOUNOU Rajath BUI HUY Huang

February 16, 2025

Abstract

This report documents the development and implementation of an AI-powered quiz generator application. The application leverages the DeepSeek-67B model to generate quizzes with randomized question types, including multiple-choice, fill-in-the-blank, and drag-and-drop questions. The project is implemented using Python, Streamlit, and the CAMEL framework. This report provides an overview of the methodology, implementation details, and results of the project.

1 Introduction

The AI-powered quiz generator is designed to create interactive quizzes for young students. The application uses advanced AI models to generate questions dynamically, ensuring a variety of question types and difficulty levels. The project aims to provide an engaging and educational tool for students while showcasing the capabilities of AI in educational technology.

This project is intended to be deployed in the market as a fully functional website, allowing students to benefit from its interactive and adaptive learning features. By collaborating with established learning platforms, the application will integrate seamlessly into existing educational ecosystems, providing a scalable and accessible solution for students worldwide. The goal is to enhance learning experiences through AI-driven personalization and real-time feedback, fostering collaboration and engagement in educational environments.

2 Methodology

The project follows a structured approach to generate quizzes:

- **Model Selection:** The DeepSeek-67B model is used for generating quiz questions. This model is selected for its ability to handle complex natural language tasks.
- **Question Generation:** The Teacher Agent, implemented using the CAMEL framework, generates questions with different formats (e.g., multiple-choice, fill-in-the-blank).

- **User Interaction:** The Streamlit framework is used to create an interactive web interface for users to take the quiz and receive feedback.
- **Answer Validation:** The application checks user answers against the correct answers provided by the AI model and provides immediate feedback.

3 Implementation

The application is implemented in Python using the following key components:

- **Streamlit:** For creating the web interface.
- **CAMEL Framework:** For defining and managing AI agents.
- **DeepSeek-67B Model:** For generating quiz questions and answers.

Below is a snippet of the core implementation:

```python
import os
import streamlit as st
import nest_asyncio
import textwrap
from camel.agents import ChatAgent
from camel.messages import BaseMessage
from camel.types import ModelPlatformType, ModelType
from camel.models import ModelFactory

# Allow nested asyncio loops for Streamlit compatibility
nest_asyncio.apply()

# Streamlit Page Configuration
st.set_page_config(page_title="Multi-Type Quiz Generator", layout="wide")
st.title("AI-Powered Quiz Generator")
st.markdown("This app generates quizzes with **randomized question types** for young students.")

# Sidebar: API Key Setup
st.sidebar.header("      API Key Setup")
aiml_api_key = "76ad069025ee4bda834f0505b3de3e39"  # Example key, replace it with yours
if aiml_api_key:
    os.environ["AIML_API_KEY"] = aiml_api_key
else:
    st.sidebar.warning("      Please enter your AIML API key to proceed.")
    st.stop()

st.sidebar.success("    API Key Set.")
```

Listing 1: Core Implementation of Quiz Generation

4 Results

The application successfully generates quizzes with randomized question types. Users can interact with the quiz, submit answers, and receive immediate feedback. The following features are implemented:

- Dynamic quiz generation with multiple question types.
- Interactive user interface for answering questions.
- Real-time answer validation and feedback.

5 Conclusion

The AI-powered quiz generator demonstrates the potential of AI in educational applications. By leveraging advanced language models and interactive frameworks, the project provides a scalable and engaging tool for students. Future work could include expanding the range of question types, integrating with learning management systems, and improving the AI model's accuracy.

References

- CAMEL Framework: `https://github.com/1sarthakbhardwaj/CAMEL-Deepseek-AIMLAPI`
- Streamlit Documentation: `https://docs.streamlit.io/`
- DeepSeek Model: `https://www.deepseek.com/`