Aiva-The AI Classroom Assistant

Swetha Anil, Sneha Ann Abraham and Devika S Nair
Saintgits Group of Institutions, Kottayam, Kerala

Abstract: In the era of digital transformation, education systems are increasingly integrating Artificial Intelligence (AI) to enhance teaching and learning experiences. This project, titled "Aiva-The AI Classroom Assistant", aims to bridge learning gaps and streamline classroom interactions between students and teachers through a smart, interactive platform. The system is designed to facilitate personalized learning by providing students with features like doubt-solving, AI-generated quizzes, instant feedback, and performance analysis reports. Teachers can upload content, monitor individual student progress, and access insightful reports to better understand student needs.Utilizing Intel's OpenVINO toolkit, the project optimizes inference for fast and efficient processing

1.INTRODUCTION

Aiva-The AI Classroom Assistant is an innovative tool designed to revolutionize the way students learn.By leveraging artificial intelligence, this assistant provides personalized support to students, helping them overcome challenges.The AI-powered system offers real-time doubt-solving, enabling students to clarify their concepts instantly.With its advanced algorithms, the assistant generates customized quizzes and assessments to evaluate student understanding.The system's performance analysis feature provides detailed insights into student strengths and weaknesses.By identifying areas of improvement, the assistant helps students focus their efforts and achieve better academic outcomes.The AI-Powered Classroom Learning Assistant is designed to be user-friendly and accessible.It can be integrated into existing learning management systems, making it easy to implement.

The assistant's natural language processing capabilities enable it to understand and respond to student queries effectively.By providing instant feedback and guidance, the assistant helps students stay motivated and engaged.The AI-Powered Classroom Learning Assistant has the potential to transform the education sector.By harnessing the power of AI, it offers a more efficient and effective way to learn.The assistant's adaptive learning capabilities ensure that students receive tailored support.This personalized approach to learning helps students achieve their full potential.The AI-Powered Classroom Learning Assistant is an exciting development in education technology.It represents a significant step forward in the use of AI in learning.By combining AI with education, we can create a more supportive and inclusive learning environment.The possibilities for this technology are vast and exciting.

2. LIBRARIES USED

In the project for various tasks, following tools are installed.

Python
Git
GitHub Account
VS Code
OpenVINO

3. METHODOLOGY

Objective

To develop a smart classroom assistant that helps students solve their doubts, generate quizzes from study material, students to take and review quizzes, and both teacher and students to view performance reports, using AI and NLP techniques.

1. **Project Architecture Overview**

The system follows a modular client-server architecture with:

Frontend: Streamlit (Student + Teacher dashboards)

Backend: Flask (API server)

Database: SQLAlchemy ORM with SQLite

AI Models: Hugging Face Transformers for quiz generation and feedback

2. Teacher Workflow

1. Login Authentication

The teacher logs into the system via their credentials and is provided with a dashboard to manage educational content and monitor students.

2. Content Upload

Teachers upload subject-wise content (text/PDF), which is stored and used to generate quizzes using AI models.

3. Student Report Access

Teachers can select a class and subject to view the list of students. For each student, they can:

View quiz report summary

Access detailed quiz feedback

Monitor overall progress

3. Student Workflow -

1. Login Authentication

The student logs into the application using their credentials and is routed to their personalized dashboard.

2. Doubt Solving

Students can type in subject-specific doubts. These are stored and later answered by the teacher. The student can also view previously answered doubts.

4. Quiz Taking

Based on the uploaded content, students can take quizzes generated using a question generation model (T5). Each quiz consists of 15 multiple-choice questions.

4. Feedback Generation

After submission, the quiz is analyzed. The system compares submitted vs. correct answers and identifies incorrect responses. Based on these, feedback is generated, highlighting:

Wrong questions and correct answers

Key weak topics to focus on
Overall Performance score

5. Report Generation

4. Reports Section

4.1 For Students

View reports by subject:
Total quizzes taken
Average score
Progress chart
Previously asked doubts

4.2 For Teachers
Enter class & subject to:
See all student names with average scores
Click on a name to see that student's quiz report and doubts asked by the students .

5. AI Models & Techniques

Task	Model/Technique
Answer Extraction	HuggingFace QA pipeline (DistilBERT / BERT)
Quiz Generation	valhalla/t5-base-e2e-qg from HuggingFace
Distractor Options	Generated from content or contextually similar distractors
Weak Topic Extraction	sentence-transformers (semantic) / keybert (keyword-based)

6. Database Models

Student: ID, name, class, institution,email, password
Teacher: ID, name, institution,email, password
StudyMaterial: Teacher ID,Subject, class, content text
QuizResult: Student ID, subject, score percent,submitted & correct answers
Doubt: Student ID, subject, question, answer

7. Frontend Features (Streamlit)

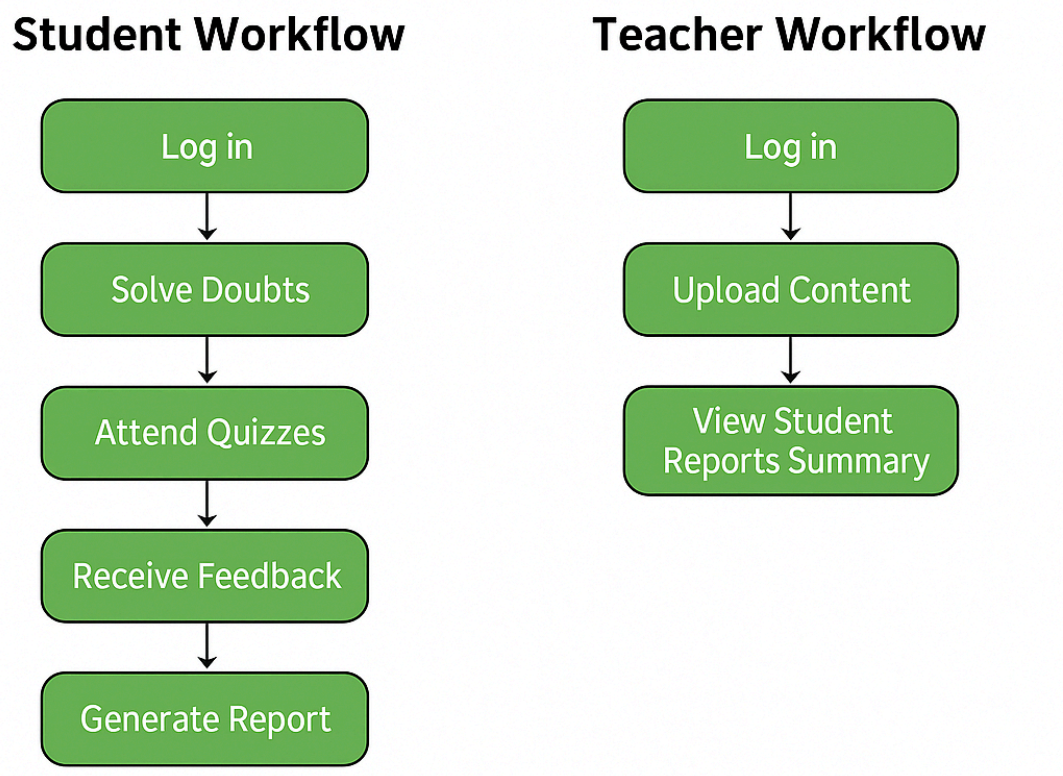
Interactive dashboards for teachers and students
Navigation handled via st.sidebar.radio
State managed using st.session_state (e.g., to remember selected students, reports, login sessions)

8. Performance Monitoring

Line charts for visualizing student progress
Easy tracking of average performance over time and per subject

9. Outcome

Teachers save time via automated quiz generation
Students receive personalized feedback
Teachers monitor student progress and target weak areas efficiently



4. IMPLEMENTATION

System Architecture Overview:

1. Frontend (Streamlit)

Built using Streamlit for an interactive web interface.
User roles: Student and Teacher.
Navigation via sidebar with dynamic page rendering.

2. Backend (Flask API)

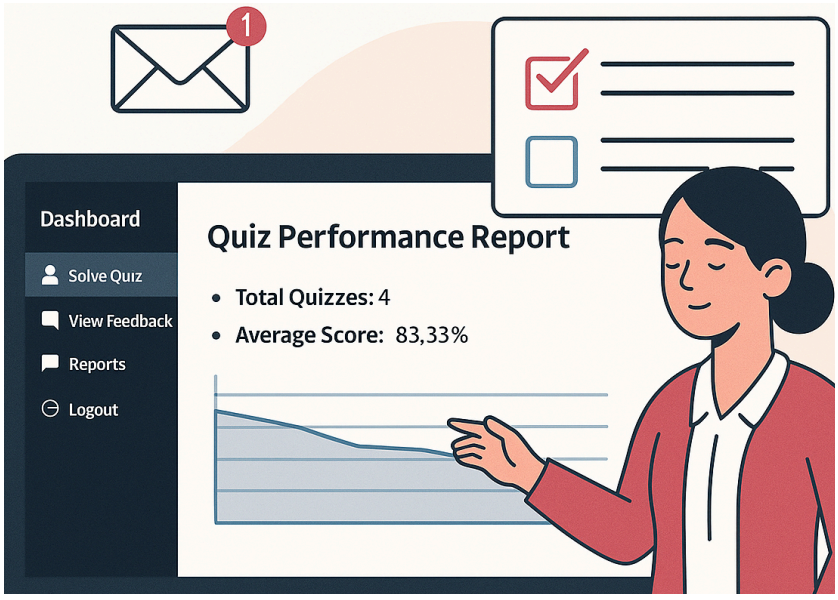
RESTful API built using Flask.
Communicates with frontend using HTTP requests.
Handles quiz submission, feedback, report generation, doubt retrieval, etc.

3. Database (SQLite)

Stores student info, quiz results, doubts, content uploads.

Models used:

Student
QuizResult
Doubt
Content



5. RESULTS AND DISCUSSIONS

Overall Outcomes:

1. User Role Separation and Navigation
Separate interfaces and dashboard flows for students and teachers.
Navigation system using a sidebar that ensured modular interaction.

2. Doubt-Solving Feature
Students could submit text-based doubts with or without attachments.
Doubts and their responses were stored and retrievable in student and teacher dashboards.

3. Seamless Student-Tailored Quiz System
Automatically generated quizzes based on uploaded academic content (text/PDF).
Each quiz included 15 well-structured multiple-choice questions with plausible options.
Real-time scoring and storage of performance metrics in the database.

4. Effective Performance Feedback
Students received detailed feedback after each quiz:
Overall score percentage.
List of incorrect questions with correct answers.
Identified weak topics (based on wrong answers).
Teachers could also access the same feedback.

5. Insightful Performance Reporting
Dynamic report section with:
Total quizzes taken.
Average score.
Trend chart showing performance over time.
Subject-wise analysis.
Teachers could view individual student reports by selecting class and subject.

6. KEY OBSERVATIONS:

1.Quiz Quality:
Using a fine-tuned question generation model (T5-based) yielded meaningful, topic-relevant questions.
Distractors (wrong options) were generated to ensure realistic confusion, improving assessment quality.

2.Feedback Analysis:
Weak topic extraction using NLP models (KeyBERT & Sentence Transformers) provided insights, though some limitations existed in topic precision for vague answers.

3.Teacher Utility:
The teacher report view allowed instructors to drill down to individual student performance by class and subject.
Enhanced teaching support by surfacing quiz patterns and doubt areas.

4.Student Engagement:
Students received actionable feedback, reinforcing personalized learning.
Simplified user interface encouraged usage without technical confusion.

5. Semantic Weak Topic Identification
Switched from basic keyword extraction (KeyBERT) to semantic similarity models (sentence-transformers) to analyze incorrect answers more accurately.
Resulted in more meaningful and topic-aligned weak point suggestions.

6. Security and Access Control
User roles (student, teacher) were strictly separated in UI logic and API access patterns.
Student-specific data like quiz results and doubts were accessible only to that student or the assigned teacher.

7. CHALLENGES

Some Shortcomings and their description:
Limited Feedback Accuracy Feedback on weak topics may not always reflect actual conceptual gaps.
No Real-time Interaction Lacks live chat or instant help for student doubts during quiz attempts.
Basic Text Input Only Currently supports only text-based doubts; no image/audio input implemented.
No Adaptive Learning The system doesn't personalize quiz difficulty based on past student behavior.
Subject Input Sensitivity Student and teacher inputs (e.g., subject names) need to be precise.
Backend Dependency Heavy reliance on model processing might cause performance delays.
Limited Visual Feedback Reports and feedback are mostly text-based, with minimal graphical summaries.
Keyword-based Feedback Limitations KeyBERT or basic NLP models can extract misleading or irrelevant topics.

8. DEVELOPMENT OF PROJECT:

Sneha Ann Abraham- Frontend Developer
Build the UI (student input form, quiz display, report viewer).
Design student interaction pages
Connect with backend APIs
Devika S Nair - Backend Developer
Set up Python server
Handle Al model integration via OpenVINO
Run inference using OpenVINO
Handle request-response logic
Swetha Anil- AI/ML & Data Analyst
Choose, convert, and integrate pretrained models into OpenVINO
Design quiz logic and report generation
Fine-tune / convert models
Generate sample questions
Analyze quiz data and prepare insights

8. CONCLUSION:

The AI-Powered Classroom Learning Assistant project successfully demonstrates how artificial intelligence can enhance personalized education through automated content analysis, intelligent quiz generation, real-time performance evaluation, and student-teacher interaction support. This end-to-end solution integrates multiple AI capabilities within a simple yet powerful platform designed for both students and teachers.

At its core, the system empowers students to self-assess their understanding through AI-generated quizzes tailored to classroom content. The quiz results are not only scored but also analyzed to provide personalized feedback, highlighting specific weak areas, incorrect responses, and suggested topics for improvement. This helps students take ownership of their learning process and make targeted efforts to improve.

For teachers, the system offers a centralized dashboard to upload classroom materials, track student performance, and resolve doubts efficiently. Through the teacher report section, educators can visualize each student's progress across subjects, identify patterns of conceptual misunderstanding, and provide timely interventions. The integration of a doubt-solving mechanism enhances engagement and supports continuous learning outside the classroom.

9. ACKNOWLEDGMENTS

We would like to express our heartfelt gratitude and appreciation to Intel© Corporation for providing an opportunity to this project.First and foremost, we would like to extend our sincere thanks to our team mentor Starlet Ben Alex for her invaluable guidance and constant support throughout the project.We are deeply indebted to our college Saintgits College of Engineering and Technology for providing us with the necessary resources,and sessions on machine learning. We extend our gratitude to all the researchers, scholars, and experts in the field of machine learning and natural language processing and artificial intelligence, whose seminal work has paved the way for our project. We acknowledge the mentors, institutional heads, and industrial mentors for their invaluable guidance and support in completing this industrial training under Intel© -Unnati Programme whose expertise and encouragement have been instrumental in shaping our work.

10. REFERENCES

1. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems (NeurIPS).
https://arxiv.org/abs/1706.03762

2. Raffel, C., et al. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (T5). Journal of Machine Learning Research, 21(140), 1–67.
https://arxiv.org/abs/1910.10683

3. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.
https://arxiv.org/abs/1910.01108

4. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.
https://arxiv.org/abs/1908.10084

5. Grootendorst, M. (2020). KeyBERT: Minimal and easy keyword extraction with BERT.
GitHub Repository: https://github.com/MaartenGr/KeyBERT

6. Hugging Face Transformers. (2020). Transformers: State-of-the-art Natural Language Processing for Pytorch and TensorFlow 2.0.
https://huggingface.co/transformers

7. OpenVINO™ Toolkit. Intel AI. (n.d.). OpenVINO: High-performance deployment of AI inference.
https://docs.openvino.ai/

8. Streamlit Inc. (2020). Streamlit: Turn Python scripts into interactive apps.
https://streamlit.io/

9. Flask Documentation. Pallets Projects. (n.d.). Flask Web Development Framework.
https://flask.palletsprojects.com/

10. Intel® Unnati AI Lab Resources. (2024). AI Tools and Teaching Resources for Education Sector.
https://unnati.intel.com