Intel Unnati Classroom Assistant : EduAI

Project Developers

Developer 1

Name: Arshabrata Bhaumik

College: Manipal Institute of Technology, Manipal

Email: arshabrata@gmail.com

Phone: +91-9163129477

Developer 2

Name: Gitaansh H Bhuradia

College: Manipal Institute of Technology, Manipal

Email: gitaanshbhuradia008@gmail.com

Phone: +91-9677704900

Mentor

Name: Priya B Kamath

College: Manipal Institute of Technology, Manipal

Email: priya.kamath@manipal.edu

Phone: +91-9844551318

Abstract

EduAI is an AI-powered educational platform designed to enhance classroom learning and teaching experiences through intelligent tutoring, resource management, and role-based access control. The system leverages OpenVINO-optimized machine learning models to provide personalized learning assistance for students while offering comprehensive content management tools for teachers and administrators.

The platform addresses critical educational challenges through several key innovations: JSON-based resource processing enables seamless integration of PDF documents with intelligent text extraction and chunking, allowing students to receive contextually-aware responses based on their uploaded materials. This approach eliminates the traditional barrier between static educational content and dynamic AI assistance, ensuring that AI responses are grounded in actual course materials rather than generic knowledge.

Real-time conversational AI with voice-to-text capabilities transforms traditional Q&A interactions into natural dialogue, accommodating diverse learning styles and accessibility needs. The intelligent suggestions system provides proactive learning recommendations and study guidance, helping students identify knowledge gaps and optimize their learning paths based on interaction patterns and subject matter complexity.

Role-based content management differentiates between student and teacher experiences, where teachers can upload and organize subject-specific resources while students receive tailored explanations appropriate to their academic level. This dual-role architecture ensures pedagogically sound interactions while maintaining content authority and academic integrity.

Advanced performance optimization techniques achieve 50% memory reduction and 40-60% faster response times through intelligent batch processing, advanced caching mechanisms, and connection pooling. The ultra-optimized server architecture with memory management prevents system overload during concurrent usage, making it suitable for institutional deployment where multiple classrooms may access the system simultaneously.

The platform implements a microservices architecture with React.js frontend, Node.js backend, Python Flask AI inference servers, and MongoDB database, supporting concurrent user access with robust JWT authentication and session management. The system demonstrates significant scalability improvements through JSON-based PDF processing for contextual AI responses, intelligent suggestion system for proactive learning guidance, role-specific interfaces optimizing teacher workflow and student engagement, ultra-optimized inference pipeline reducing computational overhead, and scalable architecture supporting institutional deployment requirements, making it viable for deployment in educational institutions requiring responsive, AI-assisted learning solutions that can handle the dynamic demands of modern classroom environments.

Software Stack

Frontend Technologies

React 18 - Modern UI framework with component-based architecture
Vite - Fast build tool and development server
Bootstrap 5 - Responsive CSS framework for mobile-first design
Axios - HTTP client for API communication
React Context API - State management for authentication and themes

Backend Technologies

Node.js + Express.js - Main API server for authentication, user management, and file operations
Python Flask - AI inference server with ultra-optimized performance
MongoDB - Document database for user data, chat history, and resource metadata
JWT (JSON Web Tokens) - Secure authentication and authorization
Multer - File upload middleware for PDF processing

AI and Optimization Stack

OpenVINO Toolkit - Intel's optimization framework for AI model inference
DeepSeek-R1-Distill-Qwen-1.5B-int4-ov - Quantized language model for efficient inference
Vosk - Speech recognition for voice-to-text functionality
pdf-parse - PDF text extraction library
Intelligent Caching System - Multi-layer caching with TTL and LRU eviction
Development and Deployment Tools
ESLint & Prettier - Code quality and formatting
Nodemon - Development server auto-reload
Winston - Advanced logging with rotation
CORS - Cross-origin resource sharing configuration
psutil - System monitoring and performance tracking

Performance Optimization Features

Connection Pooling - Optimized HTTP connections with retry strategies
Batch Processing - Request batching for efficient model inference
Memory Management - Advanced garbage collection and cleanup strategies
Compression - Response and cache data compression
Redis-like Caching - In-memory caching with compression and statistics

Features

User Authentication and Role-Based Access

When users first access the platform, they are taken to the login page. Here, they can either log in with their existing email and password or create a new account. To ensure data privacy and secure access, JWT (JSON Web Token) authentication is used across the system.

Login Functionality

To log in, users must provide their registered email address, password, and select their role — either student or teacher. The system enforces role-specific login, meaning a user cannot log in as a student if they registered as a teacher, and vice versa. This maintains clear boundaries between user experiences and permissions.

Signup Process

New users can sign up by entering their name, email, phone number, and choosing a password. During this process, they must also select their intended role. This role determines the interface and features available to them once logged in. All provided information is securely stored and managed through the backend.

Student User Interface

Once a student logs in, they are greeted with a clean and intuitive chat-based interface designed to support academic interaction.

Header Section

In the top-right corner, the student can see their profile picture and access a dropdown menu. From here, they can update their name, phone number, password, and profile image, or choose to log out.

Sidebar Navigation

The left sidebar allows students to start a new chat. Clicking this opens a prompt where they can set the topic of the chat. Once added, the chat is saved in their chat history for future reference.

Subject Resource Access

At the top of the screen, a “Subjects” button is available. Clicking this shows a list of all available subjects along with the learning materials uploaded by teachers. These can include PDFs and notes that students can download and study independently.

Chat Interaction Tools

The main chat area includes several useful features to enhance the learning experience:

- A simple text box for entering academic questions
- A dropdown to select the subject context for the conversation
- A button to upload files like PDFs or images, which the AI uses to improve response accuracy
- A microphone icon for voice-based input
- A send button to submit questions to the backend
- An option to delete chats when they are no longer needed

Students can also choose whether their queries are based on uploaded resources or general subject knowledge.

Suggestions Page

If students are looking to explore beyond the provided materials, they can navigate to the Suggestions page. Here, they can:

- Add a new topic under a selected subject
- Submit the topic to receive related articles and YouTube videos
- See a list of previously searched topics grouped by subject

This feature helps students independently research topics and return to them later as needed.

Teacher User Interface

Teachers log in by selecting the teacher role. Once inside, they are directed to a dedicated dashboard where they can manage subjects, upload resources, and track student engagement.

Dashboard Overview

The dashboard features a graph that shows which subjects have the most student doubts. Hovering over the graph reveals the percentage and total number of doubts for each subject. This gives teachers insight into which areas might need more attention or support.

Subject Management

On the Subjects page, teachers can:

- View all existing subjects
- Add new subjects to the curriculum
- Edit the subject name and description later if needed
- Delete subjects that are outdated or unnecessary

Each subject is displayed as a card, making it easy to manage and access.

Resource Management

The Resources page allows teachers to upload and organize study materials. These resources are used to fine-tune the chatbot's responses and help ensure that student queries are answered in a way that aligns with the teacher's intended syllabus.

Teachers can:

- Upload PDFs or documents for any subject
- Add and edit headings and labels to group resources clearly
- Remove resources that are outdated or incorrect

All uploaded content is editable to allow for continuous improvement.

Chat Assistant Access

Teachers also have access to the same chatbot interface as students. This allows them to:

- Try out the chat system themselves
- Experience the student perspective
- Use the tool for their own exploration and learning

Profile Management

Teachers can keep their profile information up to date using the profile dropdown. They can change their name, password, phone number, and profile picture.

Suggestions Page

Just like students, teachers can access the Suggestions page to explore helpful resources. This allows them to:

- Add new research topics under a subject
- View articles and videos related to the topic
- Save suggestions grouped by subject for later use

This is a great way for teachers to discover new material that could benefit their students.

Future Scope

1. Enhanced AI Capabilities

Multi-modal Learning: Integration of image and video processing for comprehensive educational content analysis
Adaptive Learning Algorithms: Implementation of personalized learning paths based on student performance analytics
Advanced NLP Features: Integration of more sophisticated language models for better context understanding and response generation
Real-time Collaboration: AI-powered group study sessions and collaborative problem-solving

2. Scalability and Performance Improvements

Kubernetes Deployment: Container orchestration for auto-scaling and high availability
Distributed Computing: Implementation of distributed AI inference across multiple nodes
Edge Computing Integration: Offline AI capabilities for areas with limited internet connectivity
GPU Acceleration: Enhanced support for CUDA and other GPU acceleration frameworks

3. Advanced Analytics and Insights

Learning Analytics Dashboard: Comprehensive analytics for tracking student progress and engagement
Predictive Modeling: AI-driven insights for identifying students at risk and recommending interventions
Performance Benchmarking: Advanced metrics and KPIs for educational effectiveness measurement
Real-time Feedback Systems: Instant feedback mechanisms for both students and teachers

4. Extended Platform Features

Mobile Application: Native iOS and Android applications for enhanced accessibility
Augmented Reality (AR) Integration: AR-based interactive learning experiences
Blockchain Integration: Secure credentialing and certificate verification systems
Multi-language Support: Internationalization for global educational institutions

5. Integration and Interoperability

LMS Integration: Seamless integration with existing Learning Management Systems (Canvas, Moodle, Blackboard)
Third-party API Integrations: Enhanced connectivity with educational tools and platforms
Single Sign-On (SSO): Enterprise-grade authentication integration
Data Export/Import: Comprehensive data portability and migration tools

6. Advanced Security and Compliance

FERPA Compliance: Full compliance with educational privacy regulations
Advanced Encryption: End-to-end encryption for sensitive educational data
Audit Logging: Comprehensive audit trails for educational institutions
Threat Detection: AI-powered security monitoring and threat detection

Breakdown of contribution of the developers:

Arshabrata Bhaumik:

1. **Frontend Development**
- Built the React.js frontend using modular components.
- Developed interfaces for chat, suggestion page, video/file upload, and feedback display.

2. **Backend (Node.js + Express)**
- Implemented JWT-based authentication and secure routing.
- Created REST API endpoints for chat handling, resource uploads, and user interactions.
- Integrated Multer for handling file uploads.

3. **Database Integration**
- Configured MongoDB using Mongoose.
- Defined schemas for users, chats, and resource metadata.
- Implemented efficient chat pagination and update logic.

4. **Suggestion Page**
- Designed and implemented the suggestions interface.
- Connected backend endpoints to fetch and display related content.

5. **Making the demo**
- Made the demo along with the voiceover

6. **Report and Documentation**
- Contributed to technical documentation for frontend and backend flow, along with features of the project.

Gitaansh H Bhuradia:

1. **Flask Backend Development**
- Built the Python backend to communicate with the language model.
- Handled incoming prompts, file content, and metadata.

2. **Model Integration and Prompting**
- Integrated LLM to generate contextual responses.
- Structured prompts with subject-specific context and resource injection.

3. **PDF Parsing and Preprocessing**
- Implemented PDF to text/JSON conversion.
- Extracted relevant sections based on academic subjects.

4. **Subject and Resource Logic**
- Added LMS logic for subject/resource upload and management across frontend & backend.
- Generated dynamic prompts by linking questions with relevant uploaded content.

5. **Server Optimization**
- Improved model latency through preprocessing and efficient memory use.
- Handled large payloads and ensured robust error handling.

6. **Frontend Collaboration**
- Worked to integrate Flask responses into the frontend.
- Reviewed interface behavior for model response rendering.

7. **Report and Documentation**
- Contibuted to technical documentation for model, prompt structure, and backend flow.

Link for project demo

https://drive.google.com/file/d/11sRZqKlo8bDoo9KVdPb2S3LApgh75vHY/view?usp=sharing

Link for the project source code

GreatCoder69/intel_classroom_assistant

References

Technical Documentation

- MongoDB Documentation - Database operations and schema design
- React Documentation - Frontend framework for user interfaces
- Express.js Guide - Backend API framework
- JWT Authentication - Token-based authentication implementation
- Multer Documentation - File upload handling

AI/ML Libraries

- Transformers Documentation - Model loading and inference
- OpenVINO Toolkit - Model optimization
- PyMuPDF Documentation - PDF processing

RESEARCH PAPERS

Educational AI Systems

1. "Artificial Intelligence in Education: A Review" (2019)
   Authors: Chen, Chen, Lin
   Journal: IEEE Access
   Relevant to AI tutoring system implementation

Document Processing and Knowledge Extraction

2. "Investigating students' cognitive processes in generative AI-assisted digital multimodal composing and traditional writing" (2022)
   Authors: Zhang, Liu, Wang
   Journal: Computers & Education
   Relates to PDF processing pipeline

ADDITIONAL RESOURCES

Performance Optimization
- Flask Performance Best Practices - Optimization strategies
- React Performance Optimization - Frontend optimization techniques

Database Design
- MongoDB Schema Design Patterns - Database architecture patterns