Eduke – A Smart Academic Management System

Introduction

Eduke is an advanced academic management system designed to simplify the interaction between students, parents, teachers, and administrators. It enables efficient management of attendance, marks, and performance evaluations while integrating an AI-powered chatbot to assist users with personalized tips, academic queries, and performance improvement strategies.

The system leverages historical academic data to predict student performance and provides actionable insights to improve grades and overall academic success. By providing tailored access and features for each user role, Eduke ensures a seamless and collaborative experience for all stakeholders.

Features

For Students

- View attendance, marks, and evaluations.
- Receive study tips and academic advice from the chatbot.
- Interact with teachers through the integrated chat feature.

For Parents

- Access their child's attendance, marks, and evaluations.
- Provide performance evaluation data for factors like study time, sleep time, and stress management.
- Submit an overall "Parent Rating" for their child.
- Communicate with teachers via chat.

For Teachers

- Manage attendance and mark records.
- Evaluate students on metrics like focus, participation, and homework completion.
- Provide "Teacher Ratings" as part of overall student evaluation.
- Interact with students and parents via chat.

For Admins

- Manage user accounts for students, parents, and teachers.
- Assign classes and subjects to teachers and students.
- Monitor system usage and generate reports.

AI-Powered Chatbot

- Provides tailored responses based on user roles.
- Offers academic advice, tips for grade improvement, and answers to academic queries.

- Fetches real-time data from the system's database to ensure accuracy.

Database Design

Eduke's database is structured with **10 relational tables** to store and manage data efficiently:

1. **Users Table**: Stores common details for all user roles.
2. **Students Table**: Manages student-specific data, including roll numbers and class associations.
3. **Parents Table**: Links parents to their child and stores login credentials.
4. **Teachers Table**: Maintains teacher profiles and their class assignments.
5. **Classes Table**: Defines classes and links them to teachers.
6. **Subjects Table**: Stores subject details and unique codes.
7. **Attendance Table**: Records attendance for students by class and subject.
8. **Marks Table**: Maintains students' percentage scores for subjects.
9. **Student Evaluation Table**: Tracks detailed performance evaluations with inputs from teachers and parents.
10. **Chat Table**: Supports real-time messaging between students, parents, and teachers.

Objectives

1. To develop an AI-powered academic management system that simplifies and automates essential processes like attendance tracking, marks management, and student evaluations.
2. To enable personalized insights into student performance using data-driven approaches and machine learning models.
3. To foster better communication and collaboration between students, parents, and teachers through integrated features like chat and AI chatbot.
4. To create a centralized system that enhances decision-making for educators and parents.

Process

1. **Requirement Analysis**: Identify the specific needs of students, parents, teachers, and admins.
2. **System Design**: Develop database schema, define user interfaces, and plan system workflows.
3. **Database Setup**: Design relational tables to store user data, attendance, marks, evaluations, and chats.
4. **Feature Development**:
    - Build role-based access for users.
    - Implement chat functionality for communication.
    - Design and train an AI chatbot for academic advice and personalized interactions.

    - Develop a performance prediction model using machine learning.
5. **Testing**: Perform thorough testing to ensure accuracy, reliability, and user satisfaction.
6. **Deployment**: Host the system on a cloud-based platform for accessibility.
7. **User Training and Feedback**: Educate users about system features and gather feedback for continuous improvement.

Resources

Hardware Requirements:

- **Processor**: Dual-Core, 2.0 GHz or higher (e.g., Intel Core i3 or equivalent).
- **RAM**: Minimum 4GB.
- **Storage**: 500GB or more.
- **Operating System**: Windows 7/8/10/11 (64-bit), Linux, or macOS.
- **Web Browser**: Google Chrome, Mozilla Firefox, Brave or Microsoft Edge (latest versions).
- **Display**: 1024x768 resolution or higher.
- **Network**: Stable internet connection with a minimum of 5 Mbps.

Software Requirements:

- **Backend**: Python with Django Framework.
- **Frontend**: HTML, CSS, JavaScript, Bootstrap.
- **Database**: MySQL.
- **AI Tools**: TensorFlow or Scikit-learn for machine learning, OpenAI's GPT for chatbot.
- **Version Control**: Git/GitHub.

Hardware:

- A reliable server machine with sufficient processing power and memory.
- Devices like laptops or desktops for user access.

Software:

- Python 3.x
- Django Framework
- MySQL Database Server
- TensorFlow/Scikit-learn for AI modules
- OpenAI API for chatbot functionality
- Web browser for user interface interaction.

Limitations

1. The system's performance prediction accuracy depends on the quality and quantity of data.
2. Limited functionality of the chatbot for non-academic queries.
3. Initial implementation requires significant training and familiarization for users.
4. Resource-intensive AI models may increase system latency or hosting costs.

Conclusion

Eduke is a comprehensive solution to modernize academic management by automating routine tasks, predicting student performance, and enhancing collaboration between stakeholders. By integrating AI technologies and user-friendly features, Eduke provides actionable insights and fosters an environment conducive to academic success. While the system has certain limitations, its scalable architecture and innovative approach make it a valuable tool for educational institutions.

References

1. Python Software Foundation. Python Language Reference. Available at https://www.python.org
2. Django Documentation. Available at https://docs.djangoproject.com
3. MySQL Documentation. Available at https://dev.mysql.com/doc/
4. OpenAI API for Chatbot Integration. Available at https://platform.openai.com