STUDENT ASSISTANT

Mini-Project Report

Submitted in partial fulfillment of the requirement of the degree of

Master's in Computer Application

in

K.R. Mangalam University

by

Nitesh (2401560025)

Under the supervision of

Supervisor Name : Mr. Aijaz

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDEX

ABSTRACT

In the evolving landscape of digital education, the demand for efficient tools to aid in curriculum development and student practice is ever-growing. The "Student Assistant" is an innovative application designed to automate the generation of practical programming assignments. This tool leverages AI to create tailored assignments based on specific topics or uploaded notes, providing educators with a streamlined way to produce varied learning materials and offering students targeted practice.

Built with Python for its versatility and rich ecosystem, and utilizing the Gemini API for AI-driven content generation, the application features a user-friendly graphical interface developed with CustomTkinter. It also incorporates PDF handling capabilities with FPDF and PyPDF2. "Student Assistant" delivers a seamless, responsive experience.

The platform focuses on providing an intuitive environment where educators can easily generate assignments with adjustable difficulty levels, and students can receive targeted practice material, enhancing the overall learning experience. By automating the assignment creation process, "Student Assistant" aims to save educators time, support effective learning, and empower students to achieve academic success in a technology-driven educational setting.

INTRODUCTION

"Student Assistant" is an innovative application designed to support programming education by automating the generation of practical assignments. The application provides a user-centric platform focused on the efficient creation of targeted learning exercises. It addresses the challenges faced by educators in creating varied and relevant assignments and the need for students to access focused practice materials.

The platform utilizes the Gemini API to generate assignments based on user-provided topics or uploaded documents, offering adjustable difficulty levels to cater to different learning stages. By incorporating a user-friendly GUI built with CustomTkinter and PDF handling capabilities, "Student Assistant" ensures an accessible and efficient workflow for both educators and students.

This tool not only streamlines the assignment creation process but also promotes effective learning by providing students with relevant and challenging exercises. By transforming traditional methods of assignment development into an automated, technology-enabled process, "Student Assistant" enhances the educational experience and supports academic success.

MOTIVATION

"The motivation behind creating "Student Assistant" stems from the need to address the time-consuming nature of developing practical programming assignments and the importance of providing students with targeted practice. Educators often spend significant time crafting assignments that align with specific learning objectives and difficulty levels. Students, on the other hand, require consistent and relevant practice to reinforce their understanding of programming concepts.

"Student Assistant" was conceptualized to provide a streamlined solution that automates the assignment generation process, saving educators time and delivering valuable learning resources to students. By leveraging the power of AI, the application can quickly generate varied and challenging assignments based on specific topics or uploaded notes.

Using Python for development, along with the Gemini API for AI processing, CustomTkinter for the GUI, and FPDF and PyPDF2 for PDF handling, "Student Assistant" offers a robust and efficient system. This technology stack enables the application to provide a user-friendly experience while delivering high-quality, automatically generated assignments.

Ultimately, "Student Assistant" aims to empower educators to focus on teaching and students to concentrate on learning by simplifying and enhancing the assignment process."

LITERATURE REVIEW

A comprehensive literature review of the "Student Assistant" platform explores existing research, technologies, and digital tools that address challenges and innovations in educational content creation, AI-driven learning tools, and user interface design.

1. **AI in Education:**
    - Research has shown the potential of AI to personalize learning and automate tasks in education.
    - AI-driven tools can assist in generating educational content, providing customized feedback, and supporting student learning.
    - "Student Assistant" aligns with this trend by utilizing the Gemini API to automate the creation of programming assignments, reducing the workload on educators.
2. **GUI Design for Educational Applications:**
    - User-friendly interfaces are crucial for the adoption and effectiveness of educational tools.
    - Intuitive design and clear navigation can enhance the user experience and promote efficient use of the application.
    - "Student Assistant" employs CustomTkinter to create a visually appealing and easy-to-use interface, ensuring accessibility for educators and students.
3. **PDF Handling in Educational Tools:**
    - PDF is a widely used format for distributing educational materials.
    - Tools that can process and generate PDF files offer flexibility and convenience for users.
    - "Student Assistant" incorporates PyPDF2 to extract text from uploaded PDFs and FPDF to generate downloadable assignment files, enhancing its utility in educational settings.
4. **Gaps in Existing Assignment Tools:**
    - Many existing tools lack the ability to automatically generate varied assignments based on specific topics or notes.
    - The integration of AI to create tailored assignments with adjustable difficulty levels is a unique feature of "Student Assistant".
    - This platform fills the gap by providing an efficient and intelligent solution for assignment generation, addressing the needs of both educators and students.

PROBLEM STATEMENT

"In the current educational environment, educators face significant challenges in developing engaging and effective practical programming assignments. The manual creation of assignments is time-consuming, often requiring educators to spend considerable effort in designing questions that cover relevant concepts and cater to different skill levels.

Students, on the other hand, need access to a variety of practice materials to reinforce their learning and develop problem-solving skills. However, finding suitable assignments that align with specific topics and difficulty levels can be challenging.

The lack of an efficient tool to automate the assignment generation process leads to several problems:

- Educators experience increased workload and reduced time for other teaching responsibilities.
- Students may struggle to find appropriate practice materials, hindering their learning progress.
- The quality and variety of assignments may be limited due to time constraints.

Therefore, there is a need for a tool like "Student Assistant" that can automate the generation of programming assignments, providing educators with a time-saving solution and students with access to targeted practice materials."

SOLUTION

"Student Assistant" addresses the challenges of assignment creation by providing an AI-powered tool that automates the generation of practical programming exercises. The application offers a user-friendly interface and leverages the Gemini API to create tailored assignments based on specific topics or uploaded notes.

The solution offers the following key features:

- **Automated Assignment Generation:** The Gemini API is used to generate programming assignments based on user input, saving educators time and effort.
- **Topic and Note Input:** Educators can provide specific topics or upload notes, and the application will generate relevant assignments.
- **Adjustable Difficulty Levels:** The application generates assignments with varying difficulty levels (e.g., Easy, Intermediate) to cater to different learning stages.
- **PDF Output:** The generated assignments can be downloaded as PDF files for easy distribution and access.
- **User-Friendly GUI:** The application features a clear and intuitive graphical user interface built with CustomTkinter, ensuring ease of use for educators.

By automating the assignment creation process, "Student Assistant" streamlines the workflow for educators and provides students with valuable practice materials, ultimately enhancing the learning experience."

OBJECTIVES

"The primary objectives of the "Student Assistant" project are:

- **Automated Assignment Generation:** To develop an application that can automatically generate practical programming assignments based on user-provided topics or uploaded notes.
- **Efficient Workflow for Educators:** To provide educators with a tool that simplifies and speeds up the assignment creation process, saving them valuable time.
- **Targeted Practice for Students:** To offer students access to a variety of relevant and challenging programming exercises that reinforce their learning.
- **Adjustable Difficulty Levels:** To enable the generation of assignments with varying difficulty levels to cater to different learning needs and skill levels.
- **User-Friendly Interface:** To design and implement an intuitive graphical user interface that is easy to navigate and use for both educators and students.
- **PDF Output for Easy Distribution:** To provide the functionality to download generated assignments as PDF files, facilitating easy sharing and access.

These objectives focus on creating a tool that enhances the efficiency of assignment creation, improves the quality of learning materials, and supports effective programming education."

Tools/TechnologiesUsed

The development of "Student Assistant" involved the use of several key technologies, each chosen for its capabilities and suitability for the project's requirements.

1. **Programming Language: Python**
    - Python was selected as the primary programming language due to its versatility, readability, and extensive libraries.
    - Its rich ecosystem and frameworks simplified the development process and enabled efficient implementation of the application's logic.
2. **AI Model: Gemini API**
    - The Gemini API was used to provide the AI-powered assignment generation functionality.
    - Gemini's capabilities in natural language processing and code generation allowed for the creation of relevant and varied programming exercises.
3. **GUI Development: CustomTkinter**
    - CustomTkinter was chosen for creating the graphical user interface.
    - It provides a modern look and feel, along with customizable widgets, enhancing the user experience.
4. **PDF Handling: FPDF and PyPDF2**
    - FPDF was used to generate PDF files of the generated assignments, enabling easy distribution and access.
    - PyPDF2 was used to extract text from uploaded PDF files, allowing users to input notes directly from existing documents.
5. **Threading:**
    - The threading module was used to handle background tasks, such as AI processing, without freezing the GUI, ensuring a smooth user experience.

FUNCTIONALITY

Assignment Generatio

- **Topic/Note Input:** Educators can input a specific topic or upload a PDF containing notes to guide the assignment generation.
- **AI-Powered Generation:** The Gemini API is used to automatically generate programming assignment questions based on the provided input.
- **Difficulty Levels:** The generated assignments include questions with varying difficulty levels (e.g., Easy, Intermediate) to cater to different student skill levels.

User Interface

- **Topic Entry:** A user-friendly text entry field allows educators to input the assignment topic or paste notes.
- **Assignment Display:** A textbox displays the generated assignment questions in a clear and readable format.
- **Theme Switching:** Users can switch between light and dark themes for comfortable use in different environments.

File Handling

- **PDF Upload:** Educators can upload PDF files containing lecture notes, which the application will use to generate assignments.
- **PDF Download:** The generated assignments can be downloaded as PDF files for easy sharing and printing.

Additional Features

- **Loading Animation:** A visual loading animation indicates when the application is generating the assignment.
- **Error Handling:** The application includes error handling to manage issues such as invalid input or API errors.

CONCLUSION

The Student Assistant application effectively addresses the challenges of generating programming assignments by providing an efficient and user-friendly tool for educators. By automating the assignment creation process and offering features such as topic/note input, adjustable difficulty levels, and PDF output, the application saves educators time and provides students with valuable practice materials. Developed using Python, the Gemini API, CustomTkinter, FPDF, and PyPDF2, Student Assistant ensures a robust and practical solution. It streamlines the assignment workflow for educators and supports effective learning for students, contributing to a more productive and engaging educational experience.

REFERENCES

- Python Documentation: https://docs.python.org/3/
- Gemini API Documentation:
- CustomTkinter Documentation:
- FPDF Documentation: http://www.fpdf.org/
- PyPDF2 Documentation: