**Artificial Intelligence Corrector**

**Final Project Report**

**DIPLOMA OF PROGRAMMING AND WEB DEVELOPMENT TECHNOLOGY**

**SUBMITTED BY**

Mohammed Saleh Al-Mutairi

**SUPERVISED BY**

Eng. Ayman Ali

Date: 3/12/2024

# TABLE OF CONTENTS

[TABLE OF CONTENTS 2](#_Toc184224741)

[Abstract 3](#_Toc184224742)

[Chapter 1. INtroduction 4](#_Toc184224743)

[1.1 Aim & Objectives 4](#_Toc184224744)

[1.2 Scope and limitations 4](#_Toc184224746)

[1.2.1 Scope 4](#_Toc184224747)

[1.2.2 Limitations 4](#_Toc184224748)

[Chapter 2. Methodology 5](#_Toc184224749)

[Chapter 3. Technologies used 6](#_Toc184224750)

[Chapter 4. CONCLUSION AND FUTURE work 7](#_Toc184224751)

[REFERENCES 8](#_Toc184224752)

# Abstract

Artificial Intelligence Corrector leverages cutting-edge technology to revolutionize the grading process in education. This bleeding-edge tool empowers teachers to evaluate thousands of students’ answers in no time, saving valuable time and enabling them to focus on teaching rather than the laborious task of grading exams.

# INtroduction

## Aim & Objectives

## In recent years, the integration of artificial intelligence (AI) in education has transformed traditional practices, offering innovative solutions to longstanding challenges. One of the most time-consuming tasks for educators is grading student exams, which often involves repetitive and meticulous effort. This process not only takes valuable time away from teaching but can also be prone to human error and inconsistency.

The **Artificial Intelligence Corrector** is designed to address these challenges by leveraging cutting-edge AI technologies to automate and enhance the grading process. This tool empowers educators to evaluate thousands of student answers in a fraction of the time it would take manually, ensuring accuracy, consistency, and efficiency. By alleviating the burden of grading, teachers can redirect their focus toward delivering quality instruction and engaging with students more effectively.

This introduction explores the background and significance of the project, outlining its objectives and the scope of its application. It highlights how this innovative tool aligns with the growing demand for technological advancements in education and the potential it holds to transform traditional teaching methods.

## Scope and limitations

### Scope

The Artificial Intelligence Corrector is designed specifically for digital exams, automating the grading process for digital submissions. It evaluates code-based answers accurately and consistently, supporting scalability for large batches of exams.

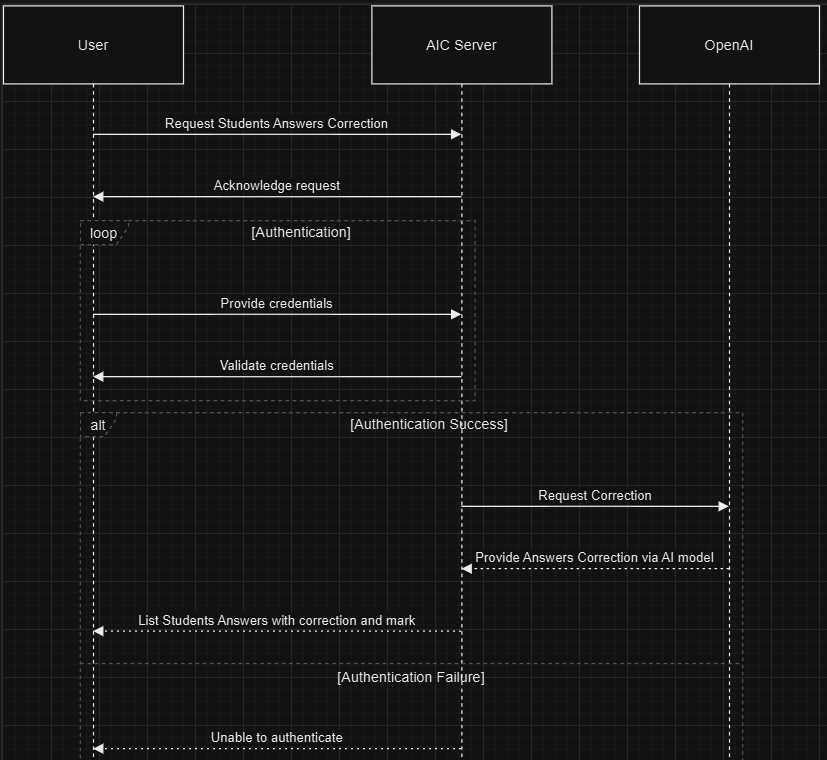
### Limitations

The tool does not support handwritten or physical exam papers. Its functionality is limited to digital-related assessments and requires clear, well-formatted digital inputs.

# Methodology

The Artificial Intelligence Corrector integrates cutting-edge technologies, specifically OpenAI’s advanced AI models and Next.js, to provide an efficient and scalable solution for grading programming exams. OpenAI’s models are used to accurately analyze and evaluate code submissions, ensuring high precision in grading by assessing correctness, logic, and efficiency. The use of OpenAI allows the system to handle diverse programming problems with varying solutions, making it adaptable to different exam formats.

Next.js plays a key role in ensuring high performance and responsiveness, offering a seamless user experience for educators.



# Technologies used

The Artificial Intelligence Corrector leverages a combination of advanced technologies to deliver a scalable and efficient solution for grading programming exams:

1. **OpenAI**:  
   OpenAI’s powerful AI models are used to analyze and evaluate programming code submissions, ensuring accurate grading based on correctness, logic, and efficiency. These models allow the system to handle various programming languages and problem types with precision.
2. **Vercel AI SDK**:  
   The Vercel AI SDK is used to integrate OpenAI’s capabilities seamlessly into the platform, providing efficient AI-powered processing for real-time evaluation of code submissions. It enhances the system’s scalability and responsiveness.
3. **Next.js**:  
   Next.js serves as the framework for the frontend, ensuring high performance and optimized user experience. It enables fast server-side rendering and efficient handling of large-scale data, making the tool highly responsive and scalable.
4. **React Server Components**:  
   React Server Components are utilized to build a dynamic, interactive user interface. These components help enhance the overall performance by allowing server-side rendering of UI elements, improving load times and responsiveness.

This combination of OpenAI, Vercel AI SDK, Next.js, and React Server Components ensures that the Artificial Intelligence Corrector delivers a fast, scalable, and efficient solution for automating the grading process for programming exams.

# CONCLUSION AND FUTURE work

The **Artificial Intelligence Corrector** successfully utilizes advanced technologies like OpenAI, Vercel AI SDK, Next.js, and React Server Components to provide an efficient, scalable solution for automating the grading of programming exams. By automating the grading process, the tool saves valuable time for educators, enabling them to focus more on teaching. It offers accurate, consistent, and unbiased evaluations, making it a reliable tool for programming exam grading.

**Future Work**

1. **Support for Paper-Based Exams**:  
   One of the key future developments is the ability to support paper-based exams. The tool could be extended to process scanned or photographed paper exams, integrating Optical Character Recognition (OCR) to convert written answers into digital format for evaluation.
2. **Extension Beyond Programming Exams**:  
   Future iterations of the tool could expand its functionality to handle other types of exams, including multiple-choice, short-answer, and essay-based assessments. This would make it a more versatile tool for various subjects and exam formats.
3. **Automated Feedback and Plagiarism Detection**:  
   Adding automated feedback to help students improve their submissions, along with plagiarism detection capabilities, would further enhance the system’s effectiveness in maintaining academic integrity.
4. **Integration with Learning Management Systems**:  
   The tool could be integrated with platforms like Moodle or Blackboard, enabling seamless grading and management of assignments within existing educational ecosystems.

In conclusion, the Artificial Intelligence Corrector represents a significant advancement in automating grading for programming exams. With further development, it has the potential to revolutionize grading across various exam formats, making it a powerful tool for the future of education.

# REFERENCES

1. Eng. Ayman Ali

2. Eng. Ahmed Al Khalifa

3. OpenAI <https://platform.openai.com/docs/api-reference/models>

4. Vercel AI SDK <https://sdk.vercel.ai/>

5. Next.js framework <https://nextjs.org/>

6. React Library <https://react.dev/>