Code Grader - AI Enhanced Grading System

HIMANI PARIKH – 1322085

KASHYAP SHUKLA - 1317566

CSCI 870 - MASTER'S PROJECT

PROJECT SUPERVISOR: DR. WENJIA LI

Team Introduction

Himani Parikh

System Design, Backend & Frontend Developer

Kashyap Shukla

DB Design, Backend & Frontend Developer

Problem Overview

Rising Student Numbers in Computer Science Curricula

Increased Grading Workload for Faculty Members

Challenges in Providing Timely Feedback on Assignments

Conventional Grading Approaches Leading to Delays

Hindered Student Learning and Academic Development

Existing Approaches

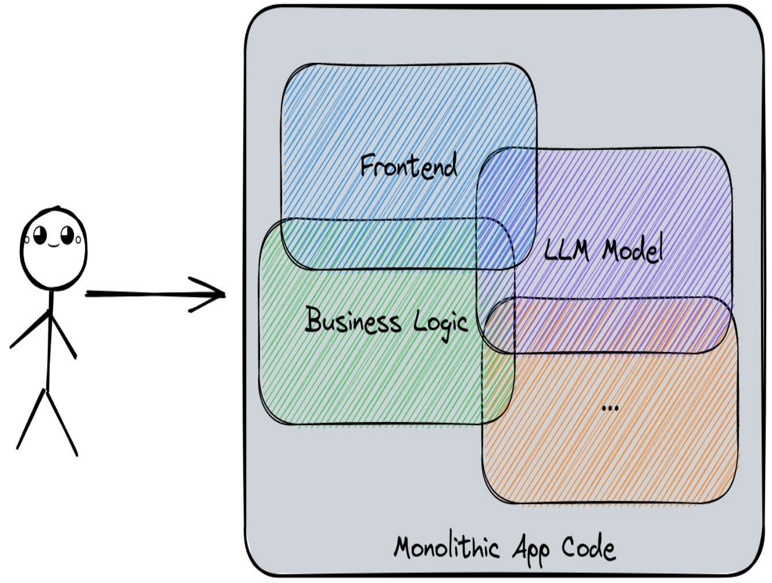
- Manual Grading by Professor or TA
- Unit testing
- Sketching Synthesis and Error Statistical Modeling(ESM)
- Peer-To-Peer Feedback
- Random Inputs Test Cases

Introduction

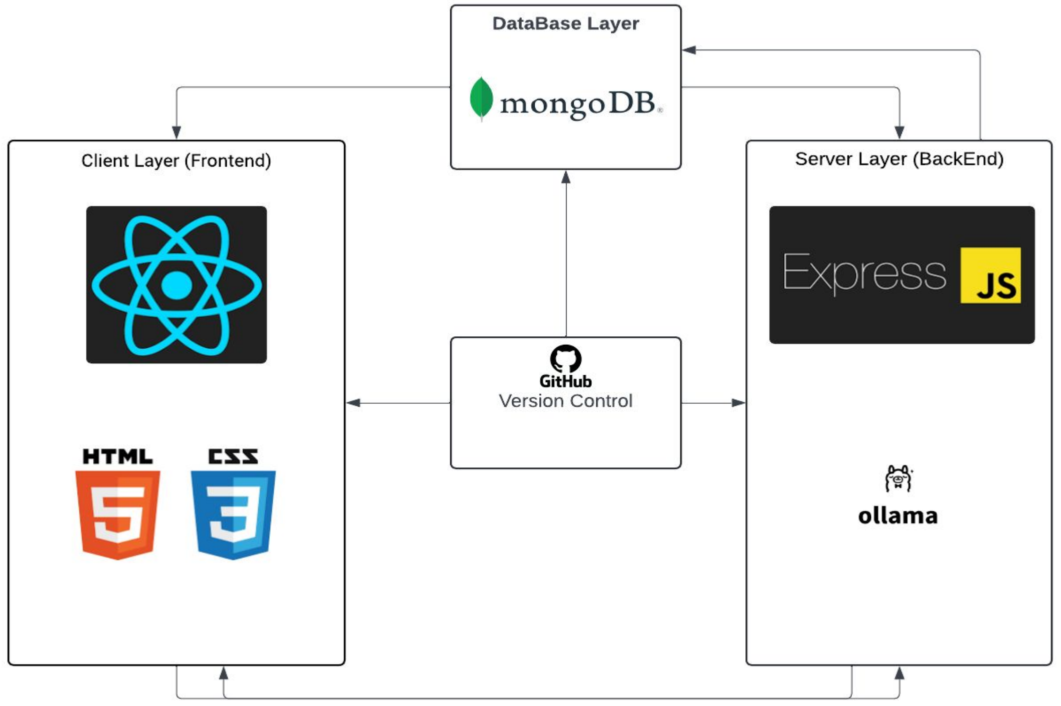
Developed a user-friendly web application interface for educators to seamlessly upload assignments, review auto-graded results, and provide final evaluations.

Integrated Meta's CodeLlama model for advanced text understanding and generation

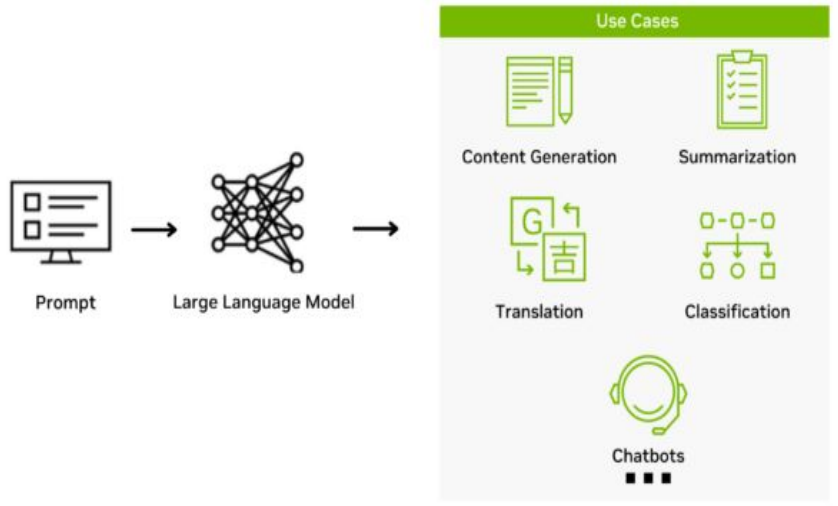
Build an automated feedback generation module with an accuracy of >80% to provide constructive critique and suggestions using the LLM's language abilities



Tools and Technologies



3 Key Advantage of LLM



Domain-Specific Expertise :

- By training the LLM on your organization’s specific data and terminology, it develops a nuanced understanding of your domain , generating highly relevant outputs.

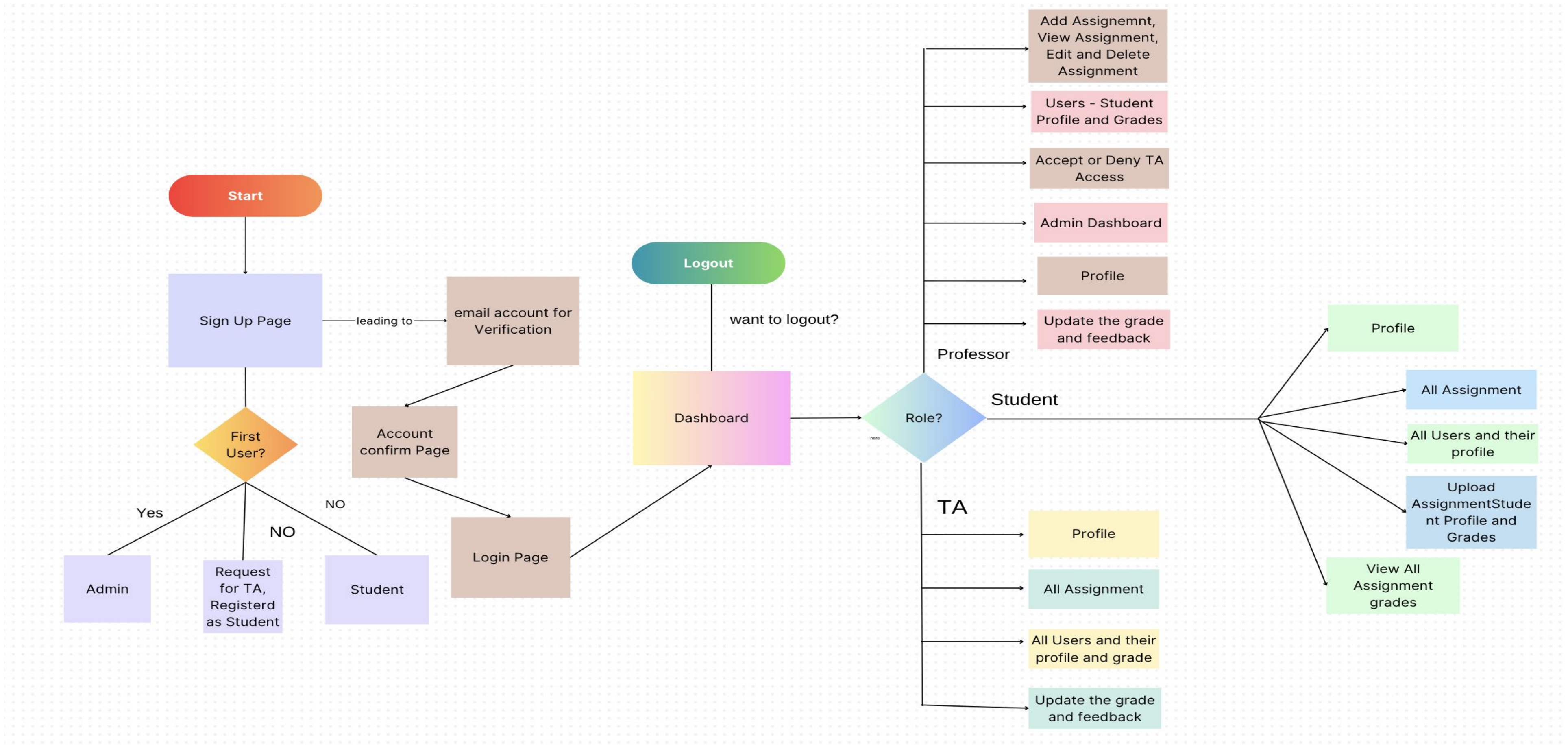
Enhanced Data Security :

- Your organization data never leaves your network, mitigating security risks of data transfer and cloud storage

Customization and Control :

- You have complete control over the training process and the model itself according to your unique needs.

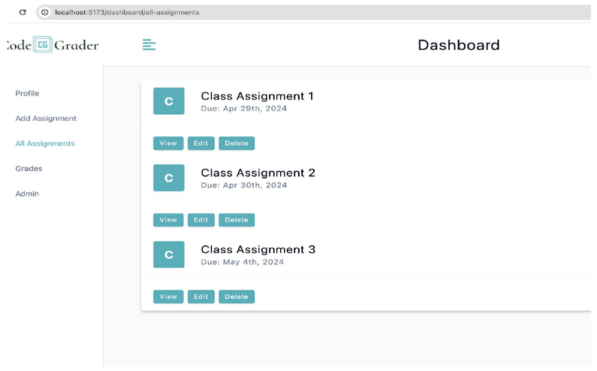
Application Workflow


Start
Sign Up Page
leading to
email account for Verification
Account confirm Page
Login Page
First User?
Yes
NO
NO
Admin
Request for TA, Registerd as Student
Student
Logout
want to logout?
Dashboard
Role?
Professor
Add Assignemnt, View Assignment, Edit and Delete Assignment
Users - Student Profile and Grades
Accept or Deny TA Access
Admin Dashboard
Profile
Update the grade and feedback
Student
Profile
All Assignment
All Users and their profile
Upload AssignmentStudent Profile and Grades
View All Assignment grades
TA
Profile
All Assignment
All Users and their profile and grade
Update the grade and feedback

Features

- Assignment Management (CRUD) for Professors.
- Submission Upload Interface.
- Automated Feedback and Grading System.
- Administrative Dashboard.
- Manual Grade and Feedback Moderation by TAs and Professors.
- TA Permission Management by Professors.
- Assignment Status Tracking and Filtering – Admin.

Application


localhost:5173/dashboard/all-assignments

Code CG Grader

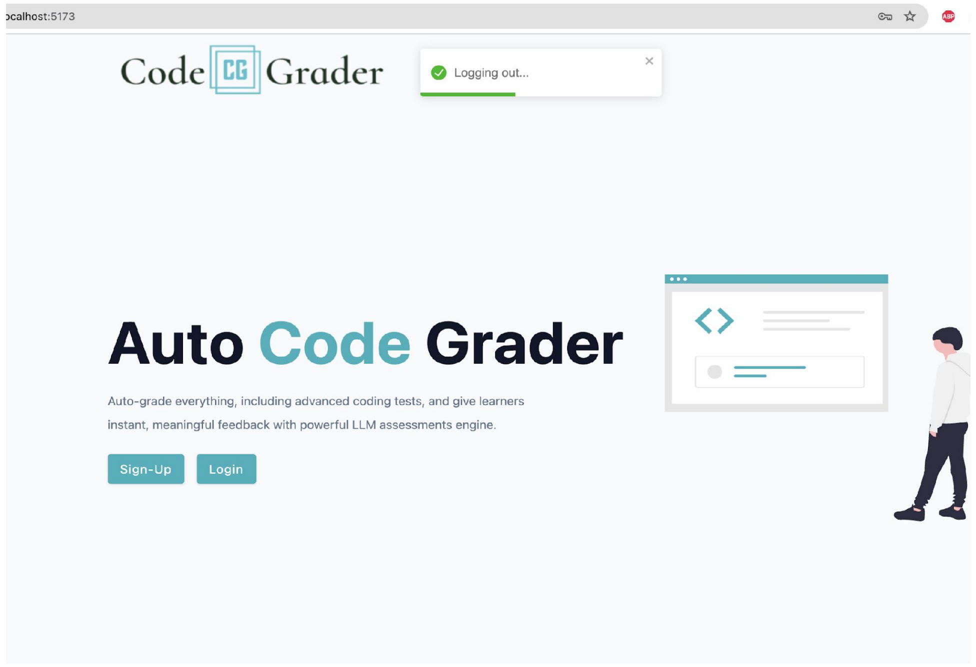
Dashboard

- Profile
- Add Assignment
- All Assignments
- Grades
- Admin

Class Assignment 1
Due: Apr 29th, 2024
View Edit Delete

Class Assignment 2
Due: Apr 30th, 2024
View Edit Delete

Class Assignment 3
Due: May 4th, 2024
View Edit Delete


localhost:5173

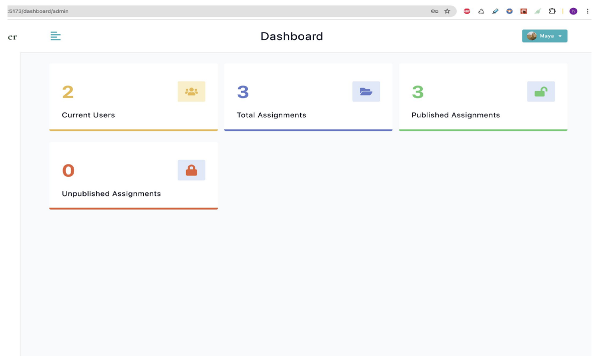
Code CG Grader

Logging out...

Auto Code Grader

Auto-grade everything, including advanced coding tests, and give learners instant, meaningful feedback with powerful LLM assessments engine.

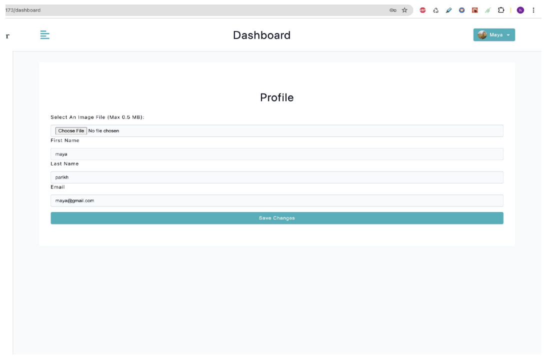
Sign-Up Login


:5173/dashboard/admin

Dashboard

Maya

2	Current Users
3	Total Assignments
3	Published Assignments
0	Unpublished Assignments


173/dashboard

Dashboard

Maya

Profile

Select An Image File (Max 0.5 MB):
Choose File No file chosen

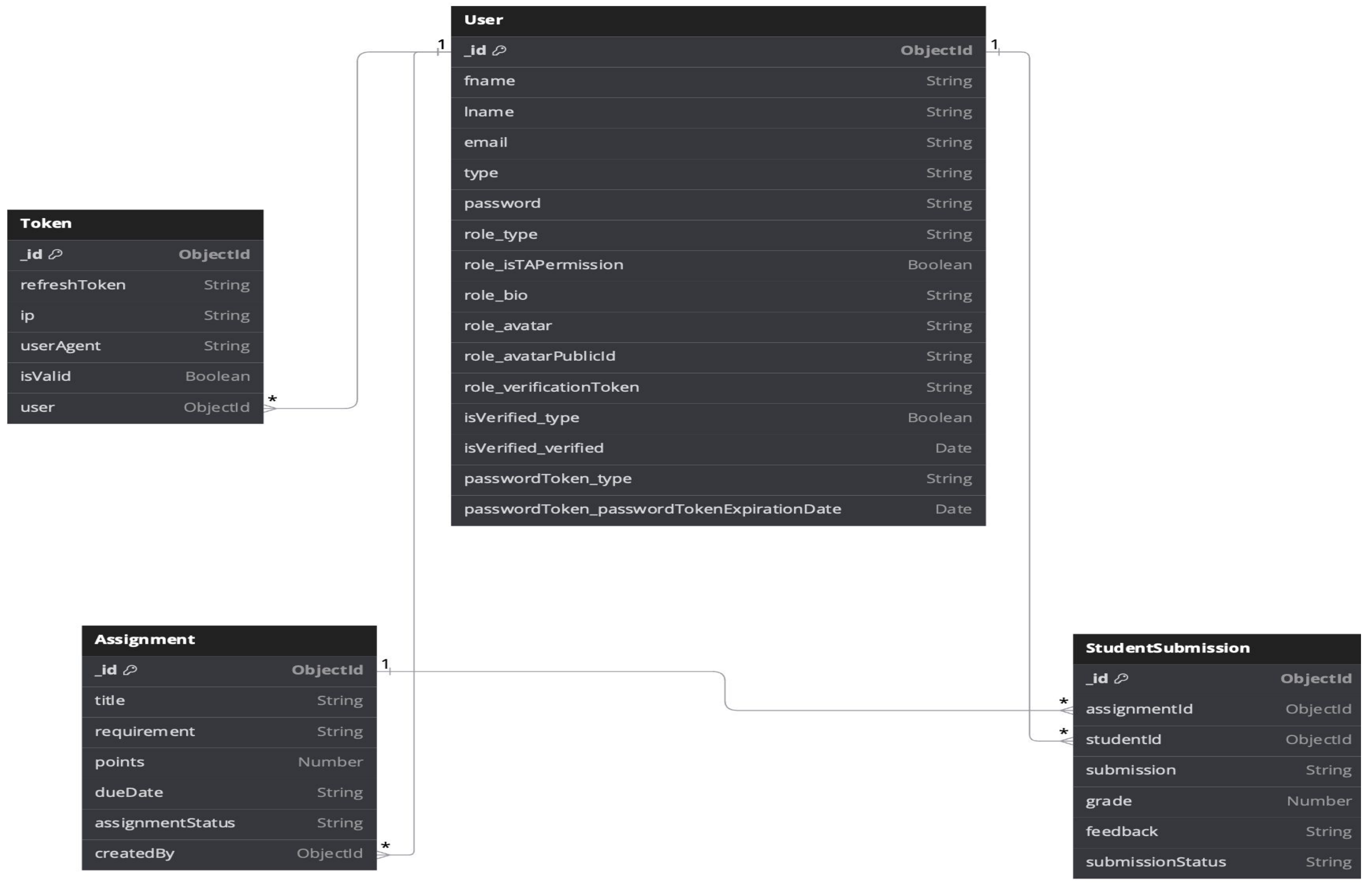
First Name: maya

Last Name: parikh

Email: maya@gmail.com

Save Changes

Database Diagram



Token	
_id	ObjectId
refreshToken	String
ip	String
userAgent	String
isValid	Boolean
user	ObjectId

User	
_id	ObjectId
fname	String
lname	String
email	String
type	String
password	String
role_type	String
role_isTAPermission	Boolean
role_bio	String
role_avatar	String
role_avatarPublicId	String
role_verificationToken	String
isVerified_type	Boolean
isVerified_verified	Date
passwordToken_type	String
passwordToken_passwordTokenExpirationDate	Date

Assignment	
_id	ObjectId
title	String
requirement	String
points	Number
dueDate	String
assignmentStatus	String
createdBy	ObjectId

StudentSubmission	
_id	ObjectId
assignmentId	ObjectId
studentId	ObjectId
submission	String
grade	Number
feedback	String
submissionStatus	String

Key Challenges

- Seamless integration of custom machine learning models into Node.js and Express.js backend
- Unsuccessful attempts with various methods:
  - Leveraging Hugging Face Interface API - https://huggingface.co/Himaniparikh/code-grader
  - Converting transformer model to TensorFlow model for TensorFlow.js integration
  - Utilizing tokenizer.json and tokenizer model files with @xenova/transformers
  - Building API using llm-api Node module
  - Converting model to PyTorch model using ONNX runtime platform

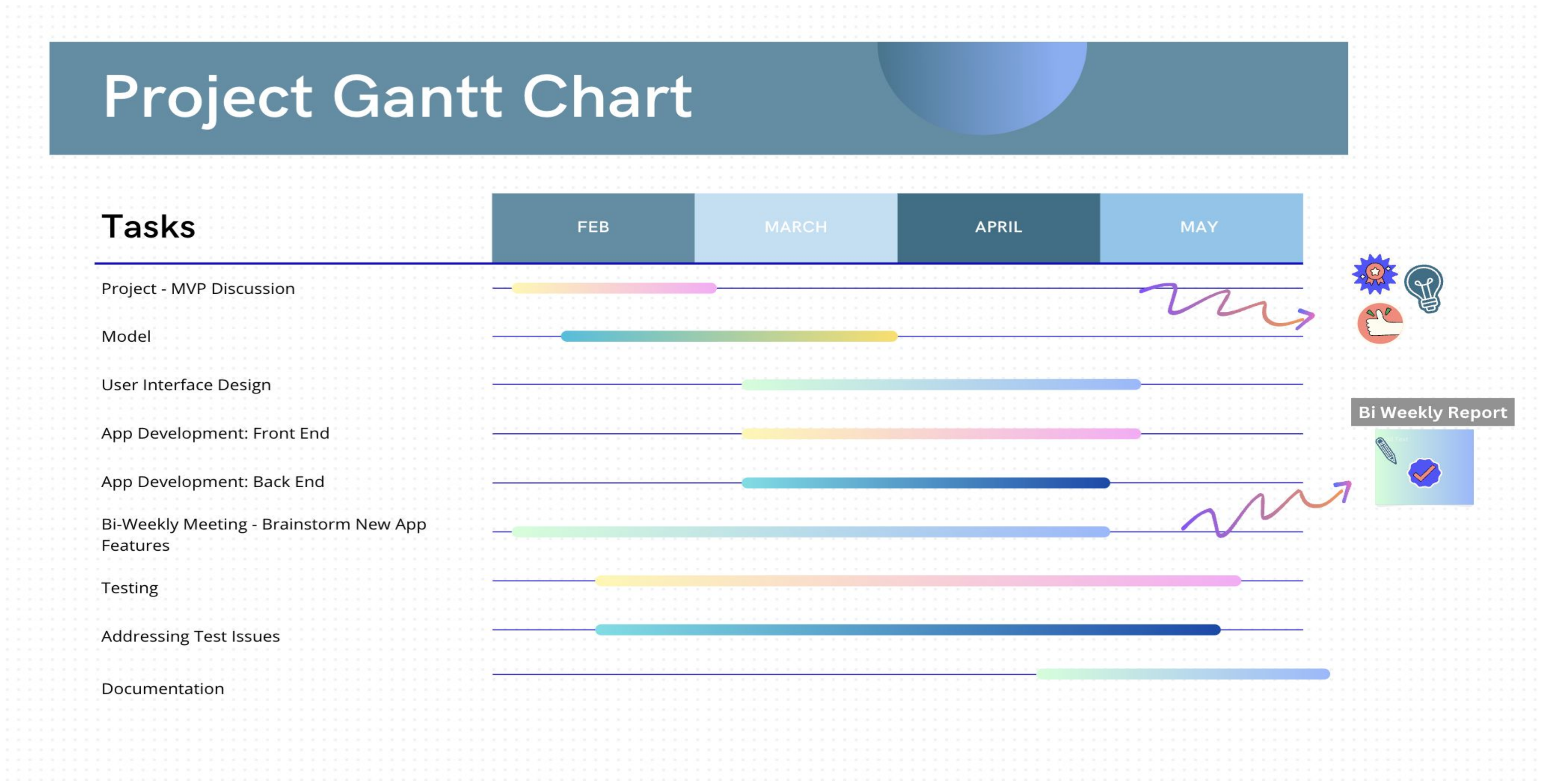
Solution

- Adopted OllaMa, a lightweight and faster tool for running custom language models locally, to overcome the integration challenges faced with previous methods and tools.

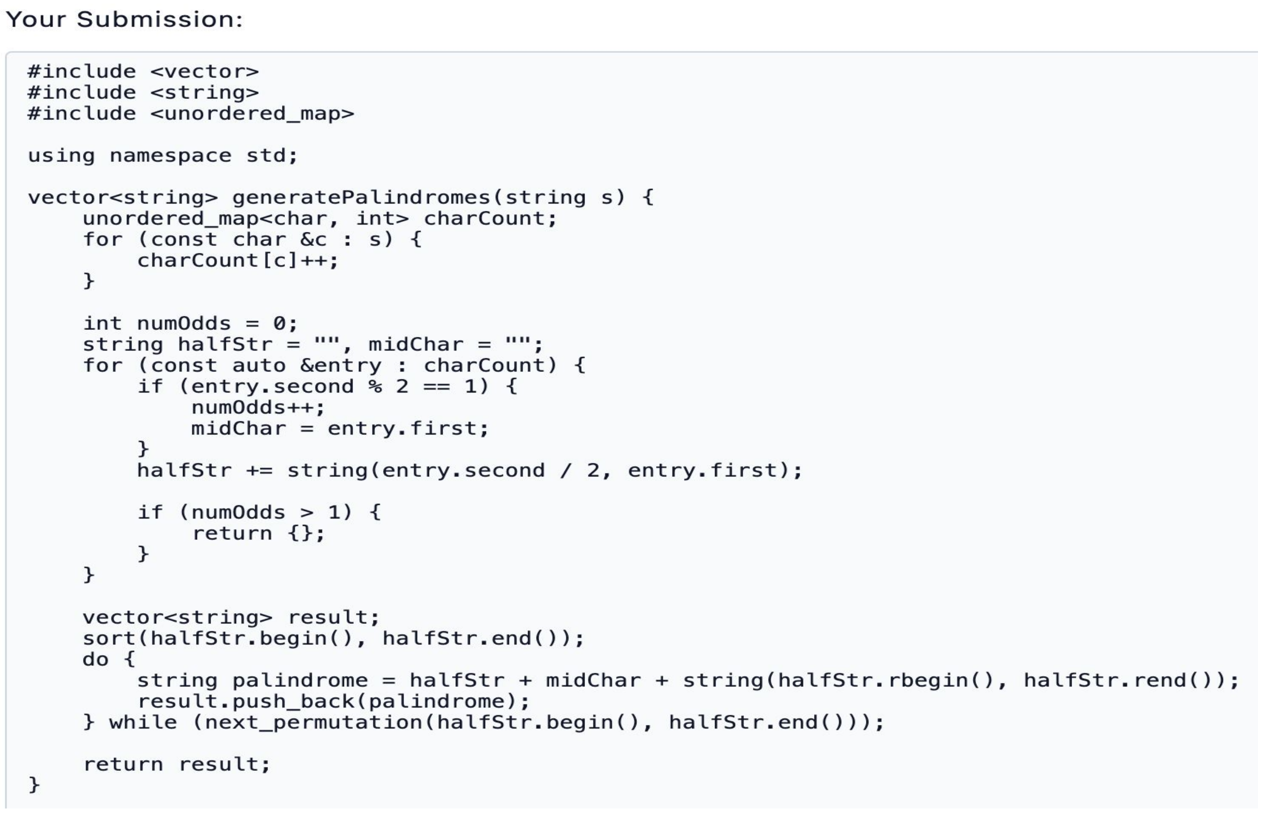
Key Challenges continue..

- Encountered proxy error while connecting frontend and backend
  - **Solution**
  - Configured Vite.js server proxy to forward API requests to the backend server, resolving the proxy error issue when connecting frontend and backend.
- Effective prompt engineering to customize and optimize the language model's responses for the specific use case of automated grading

Timeline


Project Gantt Chart
Tasks
FEB
MARCH
APRIL
MAY
Project - MVP Discussion
Model
User Interface Design
App Development: Front End
App Development: Back End
Bi-Weekly Meeting - Brainstorm New App Features
Testing
Addressing Test Issues
Documentation
Bi Weekly Report

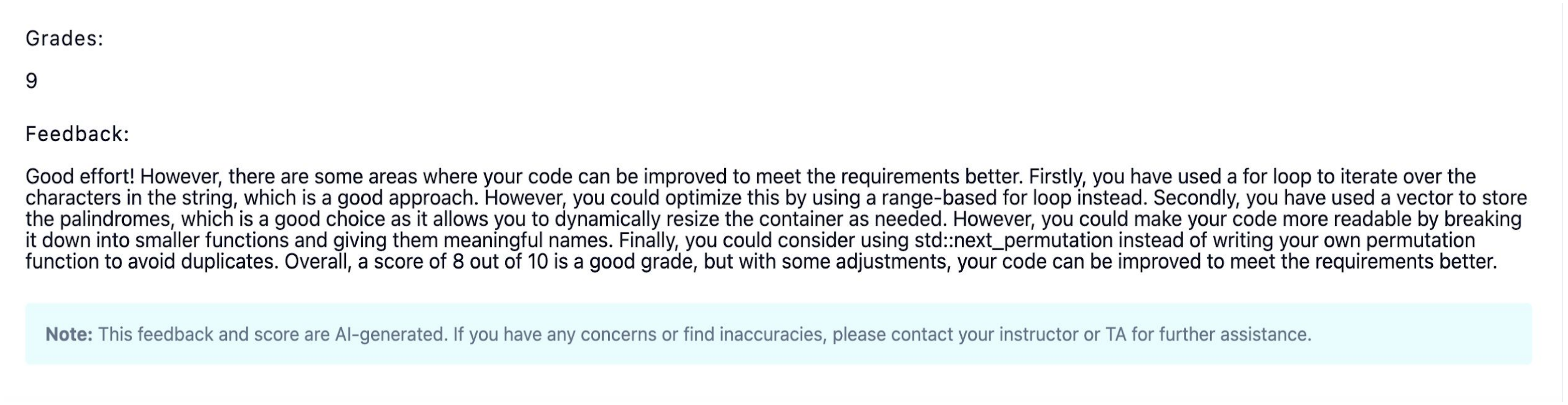
Result



Your Submission:

```
#include <vector>
#include <string>
#include <unordered_map>

using namespace std;

vector<string> generatePalindromes(string s) {
    unordered_map<char, int> charCount;
    for (const char &c : s) {
        charCount[c]++;
    }

    int numOdds = 0;
    string halfStr = "", midChar = "";
    for (const auto &entry : charCount) {
        if (entry.second % 2 == 1) {
            numOdds++;
            midChar = entry.first;
        }
        halfStr += string(entry.second / 2, entry.first);

        if (numOdds > 1) {
            return {};
        }
    }

    vector<string> result;
    sort(halfStr.begin(), halfStr.end());
    do {
        string palindrome = halfStr + midChar + string(halfStr.rbegin(), halfStr.rend());
        result.push_back(palindrome);
    } while (next_permutation(halfStr.begin(), halfStr.end()));

    return result;
}
```

Result



Grades:

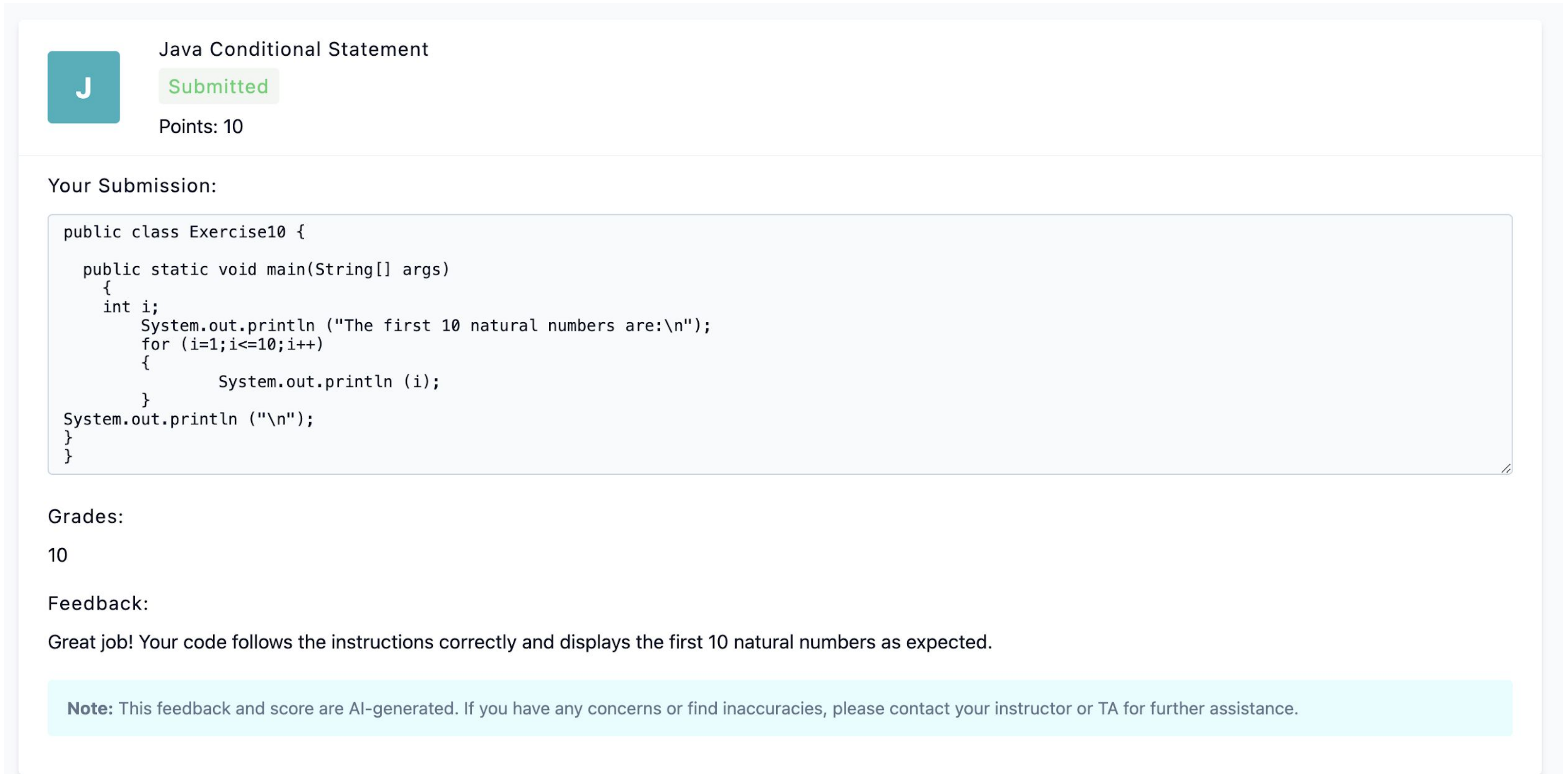
9

Feedback:

Good effort! However, there are some areas where your code can be improved to meet the requirements better. Firstly, you have used a for loop to iterate over the characters in the string, which is a good approach. However, you could optimize this by using a range-based for loop instead. Secondly, you have used a vector to store the palindromes, which is a good choice as it allows you to dynamically resize the container as needed. However, you could make your code more readable by breaking it down into smaller functions and giving them meaningful names. Finally, you could consider using std::next_permutation instead of writing your own permutation function to avoid duplicates. Overall, a score of 8 out of 10 is a good grade, but with some adjustments, your code can be improved to meet the requirements better.

Note: This feedback and score are AI-generated. If you have any concerns or find inaccuracies, please contact your instructor or TA for further assistance.

Result



J

Java Conditional Statement

Submitted

Points: 10

Your Submission:

```
public class Exercise10 {

  public static void main(String[] args)
    {
    int i;
        System.out.println ("The first 10 natural numbers are:\n");
        for (i=1;i<=10;i++)
        {
                System.out.println (i);
        }
System.out.println ("\n");
}
}
```

Grades:

10

Feedback:

Great job! Your code follows the instructions correctly and displays the first 10 natural numbers as expected.

Note: This feedback and score are AI-generated. If you have any concerns or find inaccuracies, please contact your instructor or TA for further assistance.

Future Considerations

- Dynamic Prompt Engineering based on Grading Rubrics from Instructors.
- File Submission Interface for Student Assignments.
- OAuth Integration for Secure Authentication.
- Plagiarism Detection and Similarity Checking.
- Conversational User Interface for Student-Instructor Interaction.
- Configurable Assignment Attempt Limits by Professors.
- User Experience (UX) and User Interface (UI) Enhancements.

Conclusion

- Our AI-powered grading system revolutionizes educational assessment, delivering fast, fair, and insightful evaluation through advanced AI technologies, streamlining workflows and driving innovation in assessment practices.

References

- https://www.researchgate.net/publication/332137224_Automated_Grading_Systems_for_Programming_Assignments_A_Literature_Review
- https://doi.org/10.1145/3501385.3543957
- https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10343006&isnumber=10342889
- https://www . amazon . science/publications/better- context-makes-better-code-language-modelsa-case-study-on-function-call-argument-completion
- https://ollama.com/blog

Project Link - GitHub

https://github.com/Himani324/MERN-with-AI-codegrader/tree/main

https://github.com/Himani324/MERN-with-AI-codegrader/wiki

Thank you!

QUESTIONS?