Software Engineering Final Project Report

Problem Identification:

The aim of our final project is to design and implement software that calculates the final grade for a course based on various components, such as homework, quizzes, exams, and a final project. This software aims to streamline the grade calculation process, providing a consistent and accurate representation of a student's performance.

The grading breakdown includes Homework (15%), Quiz (5%), Mid-Term Exam (25%), Final Exam (30%), and Final Project (25%). The project involved creating a system that calculates the final grade based on these components.

Problem Solved by the Product:

Manual and Time-Consuming Grade Calculation:

- The software streamlines the grading process, eliminating the labour-intensive manual calculations.
- It ensures a more efficient workflow, reducing the time required for grade computation.

Error-Prone Manual Evaluation:

- By automating the grading process, the software minimizes the risk of human errors in grade evaluation.
- It enhances the accuracy and reliability of final grades.

Lack of Centralized Data Management:

- The software introduces a centralized system for storing and processing grade-related data.
- It facilitates organized data management, making critical information easily accessible.

Scenarios Of Student Data:

Student - 1

Input:

Homework: 18, 17, 20, 19, 16, 18, 20, 17

Quiz: 8, 9, 10, 7, 8

Mid-Term Exam: 85

Final Exam: 92

Final Project: 88

Output: Given by Software

Final Percentage: 88.64375

Final Grade: B+

Student - 2

Input:

Homework: 20,20,20,20,20,20,20,20

Quiz: 10,10,10,10,10

Mid-Term Exam: 100

Final Exam: 100

Final Project: 100

Output: Given by Software

Final Percentage: 100.0

Final Grade: A

Manual Evaluation to determine Percentage and Grade to the Student:

Student - 1

Input:

Homework: 18, 17, 20, 19, 16, 18, 20, 17

Quiz: 8, 9, 10, 7, 8

Mid-Term Exam: 85

Final Exam: 92

Final Project: 88

Output:

Mean = (Sum of All numbers) / Total number of items

All HomeWorks (mean) = (18+17+20+19+16+18+20+17) / 8 = 17.875

All Quiz marks (mean) = (8+9+10+7+8) / 5 = 8.4

Mid – Term Marks = 85

Final Exam marks: 92

Final Project marks: 88

Total Percentage : 88

Final Grade: B+

Student - 2

Input:

Homework: 20,20,20,20,20,20,20,20

Quiz: 10,10,10,10,10

Mid-Term Exam: 100

Final Exam: 100

Output:

Mean = (Sum of All numbers) / Total number of items

All HomeWorks (mean) = (20+20+20+20+20+20+20+20) / 8 = 20

All Quiz marks (mean) = 10+10+10+10+10) / 5 = 10

Mid – Term Marks = 100

Final Exam marks: 100

Final Project marks: 100

Total Percentage : 100

Final Grade: A

Features of this project:

a. Functionality:

- The program allows the user to input scores for homework, quizzes, midterm exam, final exam, and final project.
- It calculates the final grade using the provided weights and displays the result.
- The program allows the user to input grades for homework, quizzes, midterm exam, final exam, and final project.
- It calculates the average grades for homework and quizzes and single grades for exams and the final project.
- The final letter grade is determined based on the calculated total grade percentage.

b. Usability:

- The code is designed to be user-friendly by prompting the user for input in a clear and understandable manner.
- It provides a straightforward interface for the user to interact with and obtain the final grade.
- Clear prompts for entering grades are included in the user-friendly code design.

- It handles errors by notifying the user when an entered grade exceeds the maximum allowed and prompts for a valid grade.

c. Reliability:

- The code is reliable as it performs the calculations based on the specified weights and provided student scores.
- It follows a structured approach to ensure accurate and consistent results.
- It uses recursive methods for input validation, ensuring that the user provides valid grades.

d. Performance:

- The code is efficient in terms of performance, with a simple and straightforward algorithm for grade calculation.
- It handles user input effectively and processes the data in a timely manner.
- The code is efficient and performs calculations with minimal complexity.
- It provides an easy-to-use interface by accepting input from the user in a straightforward way.

e. Extensibility:

- The code can be extended by adding more features or modifying existing ones without significant changes.
- It follows modular programming principles, making it easier to adapt for future enhancements.
- The code is easily extendable.
- You can add more conditions for letter grades or modify the weights without significant changes

Project Management:

a. Timeline:

- November 23rd: Commenced project with requirement formulation.
- November 24th: Initiated coding phase.
- November 28th: Conducted preliminary testing with minor errors identified.
- December 1st to December 3rd: Addressed and resolved working errors.
- December 4th: Code deemed ready.
- December 5th to December 8th: Finalized comprehensive project documentation.

b. Milestones:

- Requirement Definition: Meticulously gathered and documented project requirements.
- Coding Onset: Launched the coding phase in accordance with outlined requirements.

- Testing & Debugging: Rigorously tested the code, identifying and addressing minor errors.
- Code Refinement: Focused on refining the code and resolving identified errors.
- Finalization: Concluded the coding phase, deeming the code final.
- Documentation Completion: Accomplished comprehensive project documentation.

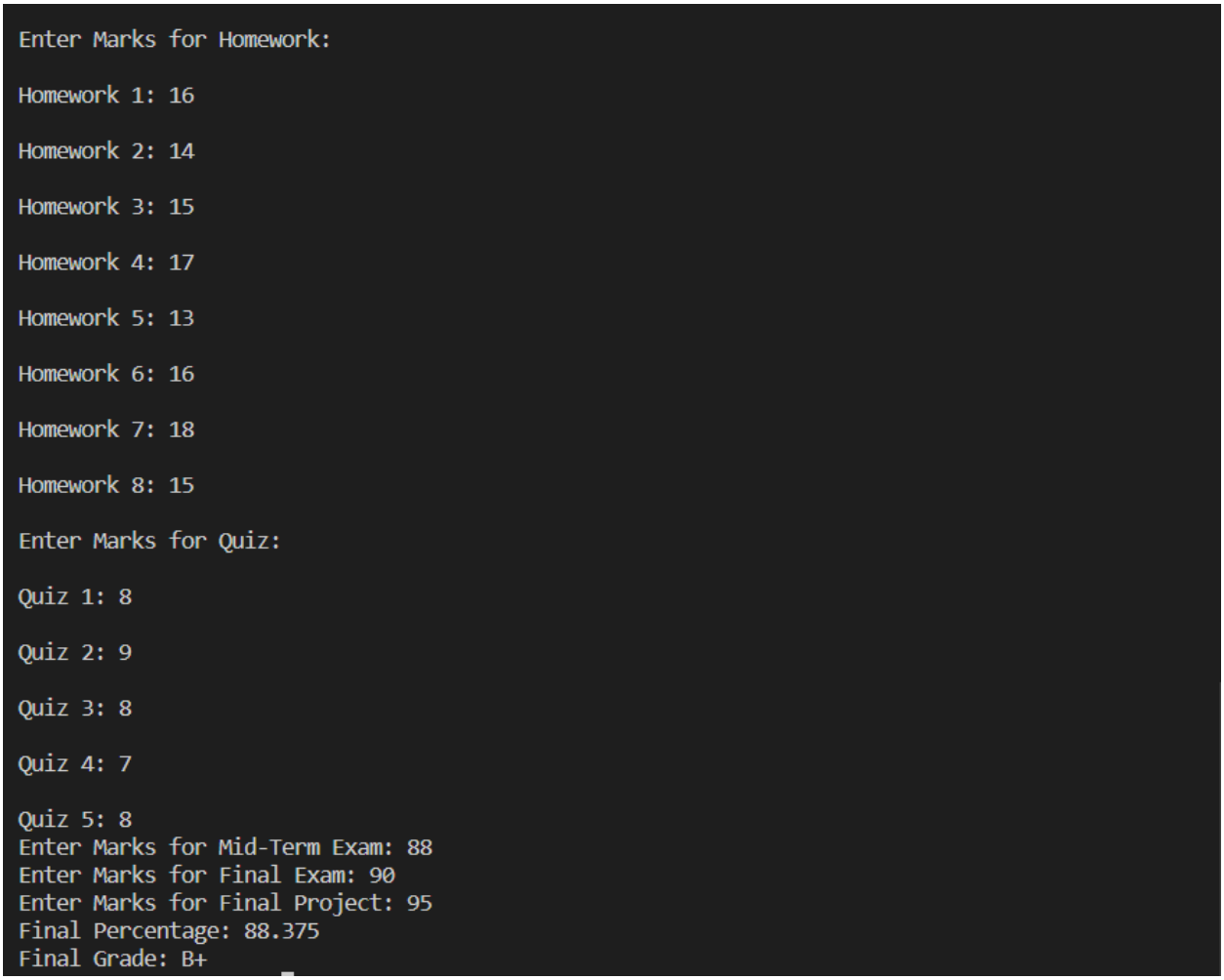
c. Tools Used:
   - Integrated Development Environment (IDE): Leveraged [mention IDE name] for efficient coding.
   - Version Control: Utilized Git for version control, maintaining the repository on GitHub.
   - Documentation: Employed Microsoft Word for creating thorough project documentation.

d. Development Process:
   - Planning: Methodically defined project requirements, outlining subsequent development phases.
   - Coding: Executed coding based on meticulously outlined requirements.
   - Testing: Systematically tested the code, promptly identifying and addressing minor errors.
   - Error Resolution: Skilfully resolved identified errors, ensuring code robustness.
   - Finalization: Concluded the coding phase with a finalized, refined code.
   - Documentation: Concluded comprehensive project documentation, ensuring clarity and completeness.

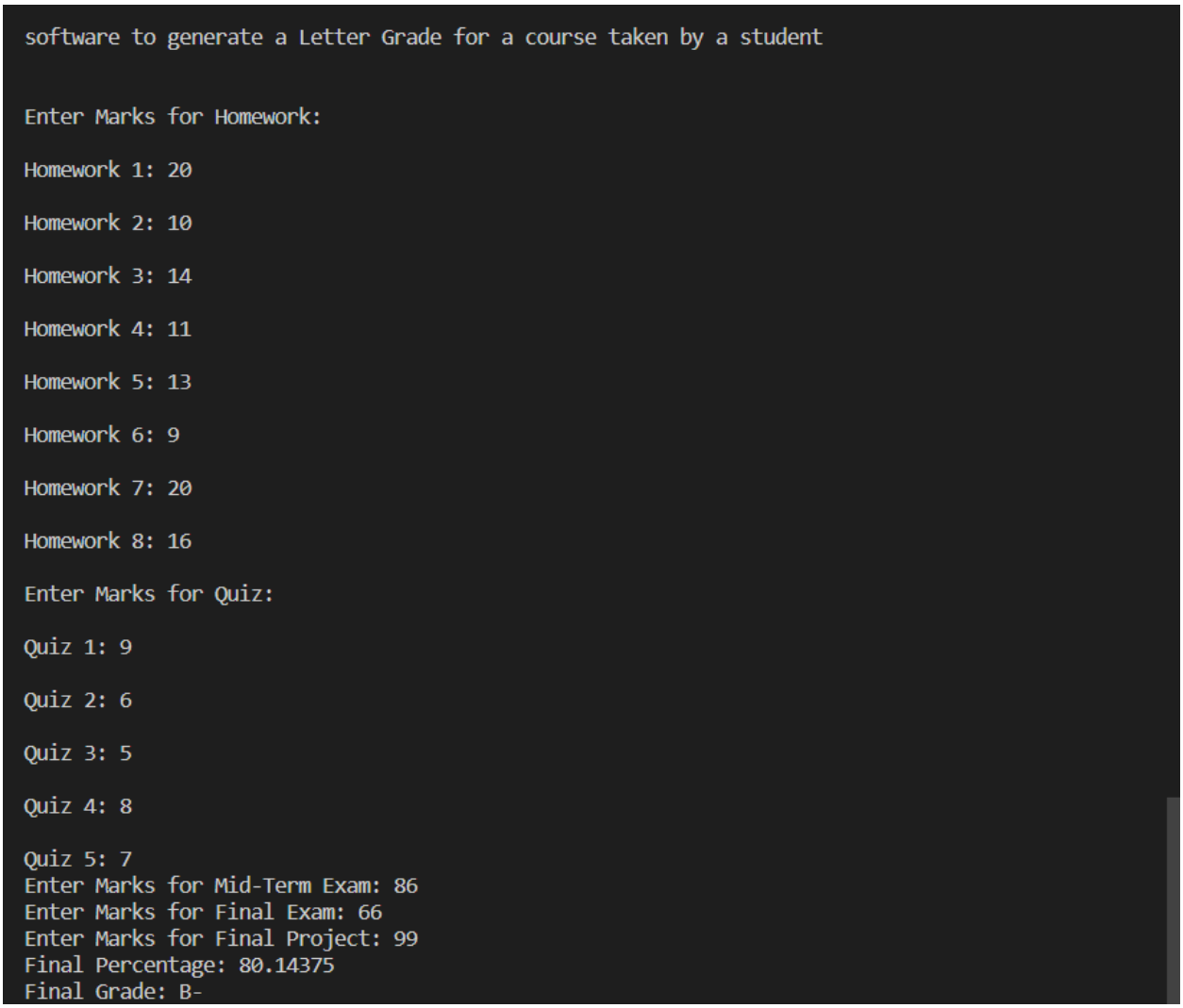
e. Difficulties Encountered:
   - Testing Challenges: Encountered challenges during the testing phase, adeptly resolved through collaborative efforts.
   - Error Handling: Addressed issues related to effective error handling, enhancing code reliability.
   - Complex Logic: Tackled challenges associated with intricate logic and calculations.
   - Documentation Clarity: Ensured documentation clarity through thorough review and revision.
   - Time Management: Effectively managed time to meet project deadlines efficiently.

Screenshots of Test Run of the Software:

a. Screenshot – 1 : Testing the Grade Accuracy for a user



b. Screenshot – 2 : Testing the Grade Accuracy for a user



c. Screenshot – 3 : Testing with “string” input and “Wrong Input ”

```
The Homework tests are for 20 marks and Quiz tests are for 10 marks. Please make sure input value doesn't exceed the maximum marks


Enter Marks for Homework:


Homework 1: zfgnzfkj
Please enter valid input

Homework 1: 548
Error , Please give a valid input.

Homework 1: 11

Homework 2: 18

Homework 3: 18

Homework 4: 14

Homework 5: 19

Homework 6: 20

Homework 7: 16

Homework 8: 19

Enter Marks for Quiz:


Quiz 1: 5

Quiz 2: 9

Quiz 3: 8

Quiz 4: 7

Quiz 5: 6
Enter Marks for Mid-Term Exam: 69
Enter Marks for Final Exam: 97
Enter Marks for Final Project: 100
Final Percentage: 87.50625
Final Percentage: B+
```

```
Grade Equivalent to Percentages

92 >= A <= 100
90 >= A- <= 91
87 >= B+ <= 89
82 >= B <= 86
80 >= B- <= 81
77 >= C+ <= 79
72 >= C <= 76
70 >= C- <= 71
67 >= D+ <= 69
62 >= D <= 66
60 >= D- <= 61
F <= 60
```

Future Enhancements:

- While the Grade Software project meets its current objectives, future enhancements could include additional features, support for more grading components, or integration with other educational systems. The modular design facilitates these potential improvements.

Conclusion for Grade Software Project:

- In conclusion, the Grade Software project successfully addressed the need for an automated and efficient system to calculate final grades for students based on various components such as homework, quizzes, exams, and a final project. The project aimed to streamline the grading process, eliminate manual errors, and provide a consistent and fair representation of a student's performance.

Team Information and Report made by :

Name: Vasanth

B-No: B00116672

Name: Ramesh

B-No: B00116634