TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Minor Project Report

On

C-Programming for Student Grade Management System

Submitted by:

Krishna Kandel	(THA081BEI014)
Nishanta Poudel	(THA081BEI025)
Pranish Pokhrel	(THA081BEI029)
Prateek Chaulagain	(THA081BEI030)

Submitted to:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

March, 2025

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Minor Project Report
On
C-Programming for Student Grade Management System

Submitted by:

Krishna Kandel	(THA081BEI014)
Nishanta Poudel	(THA081BEI025)
Pranish Pokhrel	(THA081BEI029)
Prateek Chaulagain	(THA081BEI030)

Submitted To:

Department of Electronics and Computer Engineering
Thapathali Campus
Kathmandu, Nepal

In partial fulfillment for the award of the Bachelor’s Degree in Electronics, Communication and Information Engineering.

Under the Supervision of

Prajwol Pakka

March, 2025

DECLARATION

We hereby declare that the report of the project entitled **“C-Programming for Student Grade Management System”** which is being submitted to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, in the partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in **Electronics, Communication and Information Engineering**, is a bonafide report of the work carried out by us. The materials contained in this report have not been submitted to any University or Institution for the award of any degree and we are the only author of this complete work and no sources other than the listed here have been used in this work.

Krishna Kandel	(THA081BEI014)	________________________
Nishanta Poudel	(THA081BEI025)	________________________
Pranish Pokhrel	(THA081BEI029)	________________________
Prateek Chaulagain	(THA081BEI030)	________________________

Date: March, 2025

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, a minor project work entitled "**C-Programming for Student Grade Management System**" submitted by **Krishna Kadel, Nishanta Poudel, Pranish Pokhrel** and **Prateek Chaulagain** in partial fulfillment for the award of Bachelor's Degree in Electronics, Communication and Information Engineering. The Project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled and ready to undertake any related work to their field of study and hence we recommend the award of partial fulfillment of Bachelor's degree of Electronics and Communication Engineering.

Prajwol Pakka
Project Supervisor,
Department of Electronics and Computer Engineering, Thapathali Campus

Umakanta Ghimire
Head of the Department,
Department of Electronics and Computer Engineering, Thapathali Campus

March, 2025

COPYRIGHT

ACKNOWLEDGEMENT

We would like to express our sincere gratitude towards Institute of Engineering, Thapathali Campus for providing us such a great opportunity to let us do this minor project. It has been a pleasure working and collaborating with fellows who share similar enthusiasm and dedication to accomplish our goals.

Krishna Kandel	(THA081BEI014)
Nishanta Poudel	(THA081BEI025)
Pranish Pokhrel	(THA081BEI029)
Prateek Chaulagain	(THA081BEI030)

ABSTRACT

This project is based on the C programming language. It involves the use of structure, pointer, loop to build a program that takes data of students from the user, processes it and organizes it in a systematic and efficient manner. This program takes the Name and Marks of the students. It processes Name to sort and assign Roll number, processes Marks to calculate GPA and place it in the order of Rank.

Keywords: GPA,SGMS

Table of Contents

List of Figures

Abbreviations

CSV	Comma Separated Values
GPA	Grade Point Average
GCC	GNU Compiler Collection
I/O	Input/Output
IDE	Integrated Development Environment
NG	Non-Graded
OS	Operating System
PHP	Hypertext Preprocessor
SGMS	Student Grade Management System
SQL	Structured Query Language
UI	User Interface

1. INTRODUCTION

1.1 Background

The main institution from which everyone goes through is definitely academic institutions. The movement of pursuing advancement in education affiliation leads to educational institutions. The term educational institutions includes each and every kind of institution which provides education either formal or informal. The students being attended there have to appear in different kinds of examinations and assessments. The management of grade results seems too slow in nature and seems overwhelming when calculating each and every mark to corresponding GPA. This an attempt to create a Student Grade Management System using C-programming language. It will allow the user to input important student data regarding exams and store it in a systematic order.

In educational institutes, managing student data is crucial for ensuring seamless academic administration. The Student Grade Management System (SGMS) is designed to effectively store, manage, evaluate, and categorize students' results, along with tracking other important information. SGMS makes it easy to quickly find and update information, and it helps everyone stay on the same page. Whether it's tracking how a student is doing or pulling up a report, the system makes it all simpler and less stressful. It's like having a reliable partner that takes care of the admin work, so everyone—students, teachers, and staff—can do their best.

1.2 Motivation

The idea behind creating the SGMS is to promote paperless functioning in administration offices for calculating grade results more precisely, conveniently plus fastly. To create a system that helps to minimize the redundancy of same functioning for long iterations. The time consuming- act of grade calculating manually should be discarded and as a replacement SGMS should be the best choice for this.

1.3 Problem Definition

The **Student Grade Management System** is designed to automate the process of recording, managing, and retrieving student grades efficiently. The manual system of maintaining student records is prone to errors, time-consuming, and difficult to update. This project aims to develop a **C-program** that allows users to **add, rank and display student records**, calculate grades based on predefined criteria, and store data using file handling for future access. By providing a structured and user-friendly approach, the system enhances accuracy, reduces manual workload, and ensures efficient academic record management.

1.4 Project Objectives

The main objectives of our project are listed below:

- To develop a system that efficiently records, processes, and manages student grades, reducing manual effort and errors.
- To provide accurate grade calculation with a grade sheet.

1.5 Project Scope and Applications

SGMS is an efficient and simple tool designed in the C programming language to help school, college, campus and any kind of academic related institution which relates with finalizing results only based on marks entry. The criteria to lay under the scopes of SGMS is to have an examinee and the marks criteria of subject(s) on which s/he has appeared.

1.6 Report Organization

This report is divided into 9 sections. Each chapter discusses different issues related to the system SGMS. A basic introduction about the functionalities and requirements of SGMS has been described in chapter 1. The project statement, objectives, and applications of the project have also been described in this chapter. Chapter 2 covers the important background information and history about the different techniques used regarding the grade management of students using different prototypes. Chapter 3 gives information about architecture and data flow in the system during its execution.Methodology and working principles of the grade management processes are described in Chapter 4. The block diagram of the system as well as the different processes used for separation are also described here. In chapter 6, the implementation of the project is defined. In chapter 7, the outputs are illustrated. The future enhancements which can be made in this project are described in chapter 8. Finally, the conclusion drawn from this project, by the team members, is covered in chapter 9.

2. LITERATURE REVIEW

This literature review explores existing studies, projects and technologies. Numerous studies highlight the importance of automated grading systems in enhancing academic administration. Most existing systems rely on web-based solutions using technologies like PHP, Python, and SQL databases. However, these systems may have performance overhead due to web server dependencies.

2.1 Student Management System by *ishan-cse*:

This system was innovated by Md. Harun Aur Rashid Khan Ishan, is a console-based application created using C programming language. This system was compiled in Code::Blocks IDE using GCC compiler. [1]

The system is designed as a basic structure for every choice the whole interface got changed. Firstly, the user is given choices for either teacher or student information. On the requirements, the interface proceeds. Either for the teacher or student's section. The interface asks if an individual is registered or new on the server. It provides the feature of updates like deleting individuals from the server database. The server stores the salary details about teachers and tuition fees for students based on a predefined database.

The system uses the verification for the existing students and teacher specimen by their assigned id and password at the time of registration process as a new user while accessing the database. This procedure of verification seems difficult as the administrators would need to remember id and password for every teacher and students. The logic behind every function seems scattered across different parts of the program. Instead of using a clean, modular way, it seems written separately instead of being generalized into reusable functions.

The system lacks vulnerability and portability which are the basic prerequisites for a system development. Using system-specific commands make the program dependent on a particular environment rather than being flexible across. The decision of using structures students and teachers under a single structure is lame as they design makes the system inflexible, wasting memory when it's not fully used and causing limitations when more data needs to be stored.

The strong point of using user credentials for logging into the database seems weak due to unuse of strong security. Using a logout feature would have solved this issue which will be the backbone for the system. Using structured prompts for data input of teachers, students, and their attendances seems good to initiate. The process of data modification is kind of similar to what we are using for our system SGMS.

The system should feature the use of dynamic memory instead of static arrays, ensuring efficiency and scalability. Security should be taken seriously by implementing password encryption and decryption models instead of storing raw values. Lastly, replacing system-dependent functions with standard C functions would improve portability, ensuring the program can run across different operating systems without modification.

3. REQUIREMENT ANALYSIS

The program is quite lightweight and can run on virtually any PC that has the bare minimum of hardware and software requirements. The code is also optimized for most efficiency which further contributes to ease in hardware and software aspects.

3.1 Hardware and Software Requirements

- Processor: Intel Pentium or higher
- Input Devices: Keyboard and Mouse
- Programming Language: C
- Compiler: GCC (MinGW for Windows), Turbo C++, or Dev-C++
- Operating System: Windows, Linux, or macOS
- IDE (Optional): Code::Blocks, Visual Studio Code, or Dev-C++

3.2 Feasibility Study

3.2.1 Economic Feasibility

This addresses whether the project is cost-effective and worth investing in. This is a pretty simple project developed in C, so the development cost is minimal. If the students or in-house develop it, the cost is negligible. If outsourced, the cost depends on the developer's rate.

3.2.2 Operational Feasibility

This considers whether the system meets user requirements and is also easy to use. Simple interface for teachers/admins to input and update results, Speedy and accurate retrieval of student records. secured access, Ease of Use, Error Handling, Input validation, data integrity, and Maintenance are the key factors.

4. SYSTEM ARCHITECTURE AND METHODOLOGY:

4.1 Overview of the architecture:

This project is mainly based on file handling in C. It takes the data of students, sorts the data alphabetically, calculates GPA and rank of the students. The information is then stored in a text file.

4.2 Modules

Modular approach has been adopted for this program. The program consists of various modules as discussed below:

4.2.1 Input Module:

This module receives the necessary input required from the user. It takes the name of students and their marks in the individual subjects.It stores the input data in a CSV file for further processing.

4.2.2 Sorting Module:

This module uses a sorting algorithm to sort the names of all students and generate the roll number based on the ascending order of alphabets.

4.2.3 Processing Module:

This module calculates the GPA of the students and filters the failed students from the list. Those who are unable to get at least 40 percent of full marks are identified as failed by this program. Amongst the passed students, the best scoring ones are now sorted in numerical order and the assigned numeral is saved as the rank of that student.

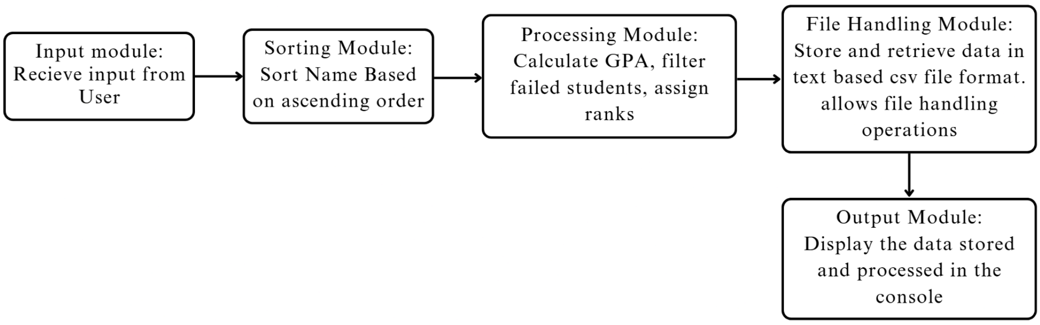
4.2.4 File handling Module:

This module interacts with the text file. We will be using a user readable csv format for this program. This module will be storing and retrieving the GPA and ranking of the students.

4.2.5 Output Module:

This module will display the result of all the students in the alphabetically sorted manner. It will display the roll number, name, GPA and rank of the students.

4.2.6 Data Flow:



Input module: Recieve input from User
Sorting Module: Sort Name Based on ascending order
Processing Module: Calculate GPA, filter failed students, assign ranks
File Handling Module: Store and retrieve data in text based csv file format. allows file handling operations
Output Module: Display the data stored and processed in the console

Figure 1: flowchart diagram for modular data flow in the program

4.3 METHODOLOGY:

4.3.1 Problem Definition:

At first we must have a clear idea of what we are trying to tackle by this project. We have declared the low efficiency and accuracy in record keeping the students' exam data in educational institutions as our problem statement. Our program aims to help the academic officials store the data digitally in an efficient way. This is our sole intention that we expect happens after the completion of this project.

4.3.2 System Design

To begin thinking about how we are going to achieve it, we brainstormed about the proper way to execute our plan. We came up with the system design that is explained as follows:

4.3.2.1 Input

In the first step the program asks for the name of the student to the user. The user inputs the student's name.. Then the program tells the user to input marks of the student with their roll number (generated from the sorting module discussed below). After the marks of all students are retrieved by the program, the input session ends.

4.3.2.2 Processing

Now, all the names, received from the input module, are stored in a CSV file. The csv file supplies these values to the next module which is the sorting module. This module is responsible for sorting the students name and assigning the roll number beginning from 1 to the last number of students. After the user inputs the marks of the students, the marks are stored in the file too. The students are now ranked by the ranking module based on the total marks obtained. The GPA of passed students is also calculated. Only the students with at least 40% marks in each subject are ranked. Others are labelled 0 GPA by the system. Then the processing session ends.

4.3.2.3 Output

Now all the necessary processing has been done by our program. All the retrieved data is processed and stored. This data is now ready to be displayed to the user. The path to the CSV file with all the processed data is shown at the end of the program with a success message.

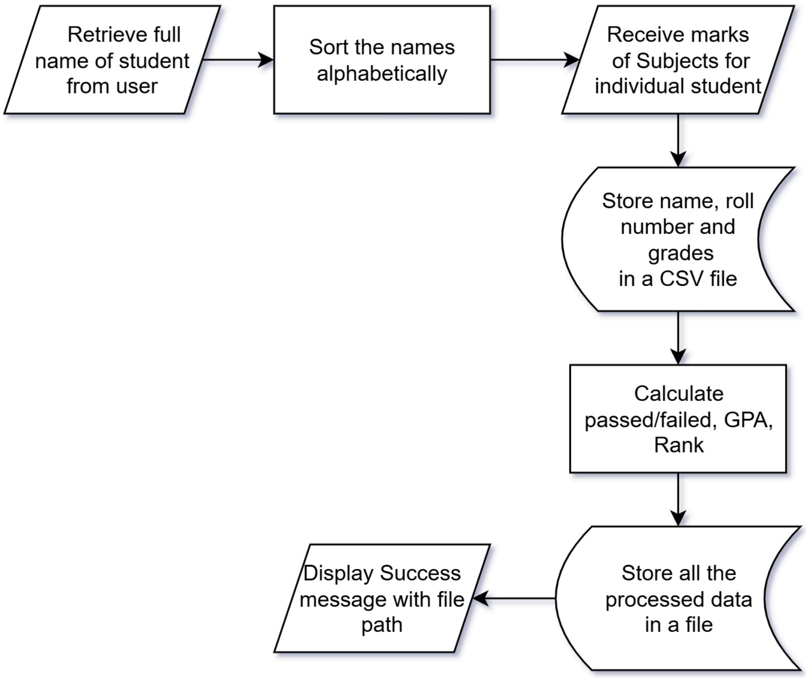

Retrieve full name of student from user
Sort the names alphabetically
Receive marks of Subjects for individual student
Store name, roll number and grades in a CSV file
Calculate passed/failed, GPA, Rank
Store all the processed data in a file
Display Success message with file path

Figure 2: flow chart of the program during runtime

4.3.3 Testing and Debugging

We will run this program for different volumes of students and observe the errors that occur (if any). We will use printf() function on different steps to check if the flow of the program is as intended. Other topics may also be discussed as the advancement of writing code begins. Proper comments are to be strictly added in the program for better readability and understandability.

5. IMPLEMENTATION DETAILS

At first, the SGMS welcomes the user with a greeting message enabling the user to go through the menu list on which the program data flow depends. The user can explore one of multiple options available in the menu interface. The user gets two options first viz. student section and grade management section, under which a menu list appears. Going through the student section, the user has three paths from which s/he can be redirected to any one of add, edit and view student details. There are four options to choose from which are described as follows.

5.1 Add Student

From the menu when the number 1 is selected from the keyboard, it asks the user to input the student's name. Then, it asks the user to input marks of 6 subjects. Then the system returns back to the main menu. To add more students, one may reselect the first option again by pressing 1 in the keyboard.

5.2 Display Students

From the menu when the number 2 is selected from the keyboard, the total students who are currently registered are shown in the console menu. Then after it shows the details of all the students, it shows the menu once again from which one can re-navigate.

5.3 Rank Students

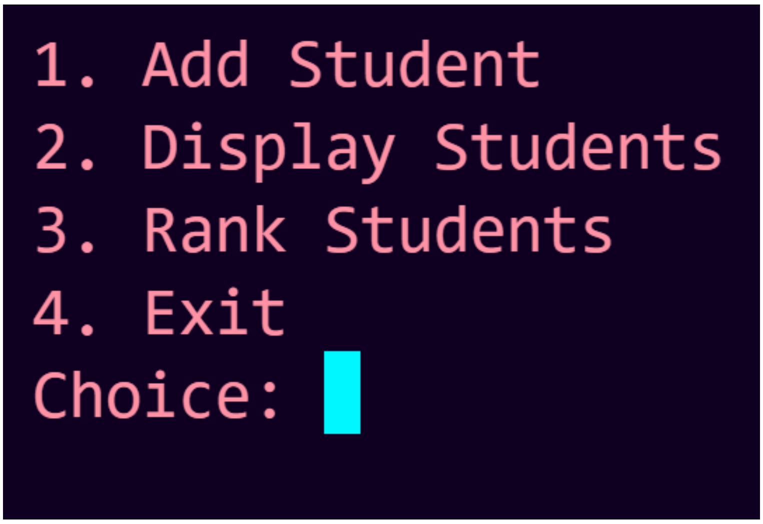
From the menu when the number 3 is selected from the keyboard, the details of students after ranking them based on their GPA is shown in the console menu.Then after it shows the details of all the students sorted on the order of their rank, it shows the menu once again from which one can re-navigate.

5.4 Exit

From the menu when the number 4 is selected from the keyboard, the system closes itself down. It is to be noted that one may only press this after completion of tasks for that session. Otherwise the system needs to be started once again.

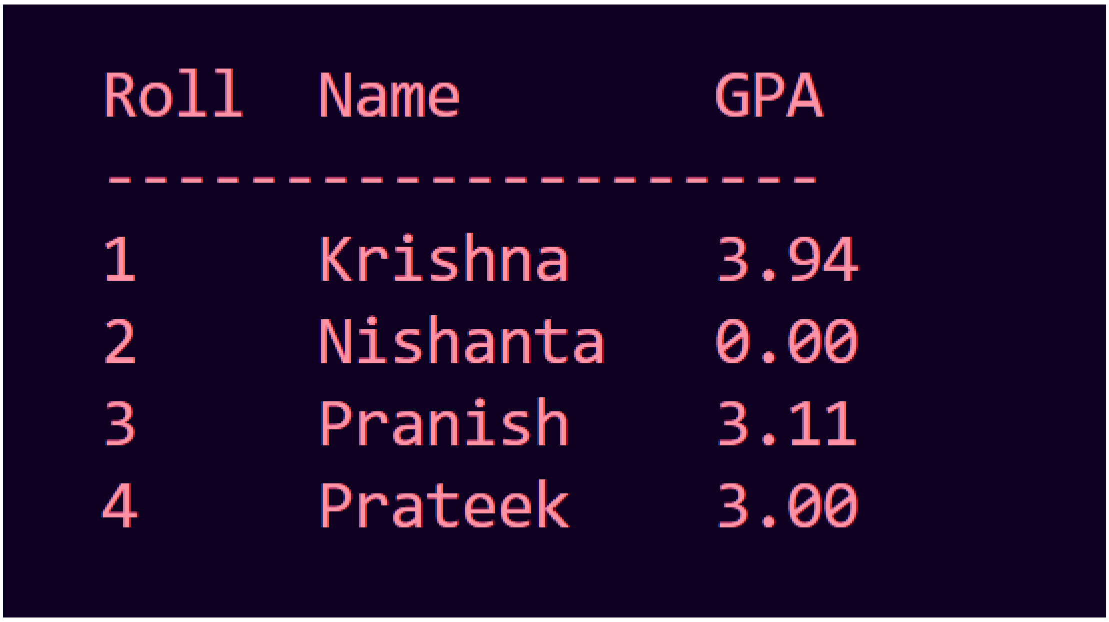
6. RESULTS AND ANALYSIS

The system ran as intended during our test run. We supplied it with the data of four students and ranked the supplied data. The result from the console menu are shown below:



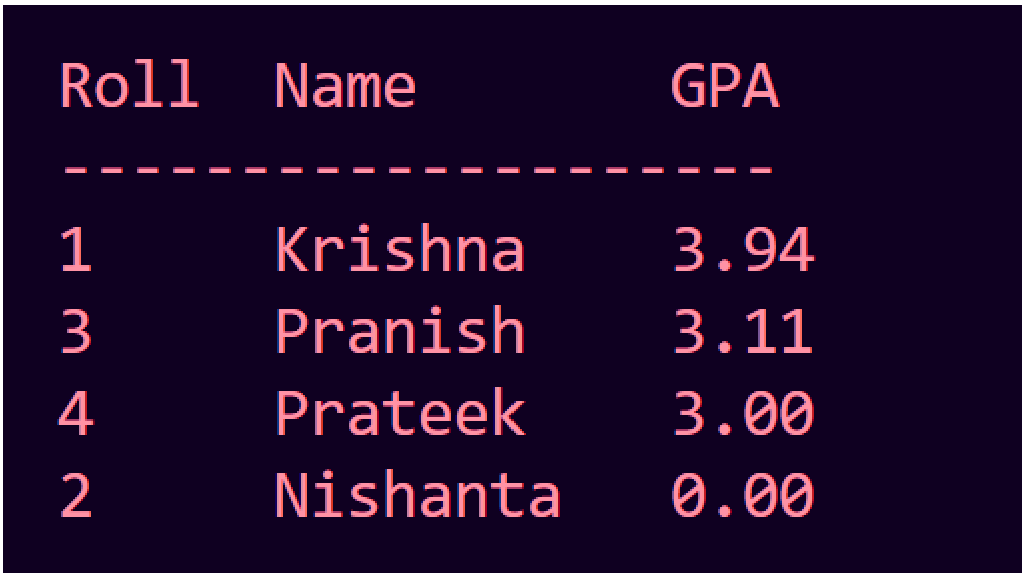
```
1. Add Student
2. Display Students
3. Rank Students
4. Exit
Choice:
```

Figure 3: main menu interface after running the program



```
Roll  Name       GPA
--------------------
1     Krishna    3.94
2     Nishanta   0.00
3     Pranish    3.11
4     Prateek    3.00
```

Figure 4 : students ordered on alphabetical basis



```
Roll   Name         GPA
--------------------
1      Krishna      3.94
3      Pranish      3.11
4      Prateek      3.00
2      Nishanta     0.00
```

Figure 5 : students ordered on rank basis

The reason the rank is displayed zero for student Nishanta is because a failing mark of less than 24 was given in one of the subjects which enabled it to just return GPA as 0. which is actually NG, this a plus point to be considered in the enhancement section.

7. FUTURE ENHANCEMENT

No matter what the project is, there are always rooms for future enhancements. A better UI can be adopted for a proper user friendly experience. An application can be made with this program with adequate privileges for students and teachers.

8. CONCLUSION

Overall, the project was a fun collaboration with fellows from backgrounds with similar interests. We pushed ourselves beyond our current knowledge and broadened the spectrum of what we know in this field. The process of overcoming the hurdles with working as a team really helped to ignite the spark of what is possible.

9. APPENDICES

The gantt chart for this project is as follows:

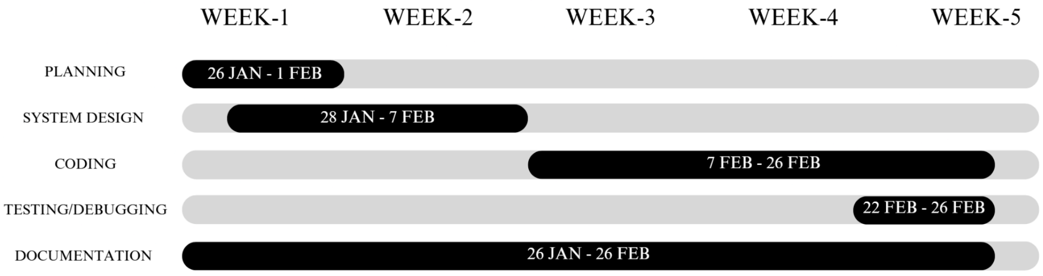


Figure 3: gantt chart for expected time estimation of the project

References

[1] https://github.com/ishan-cse/Student-Management-System-in-C-programming

[2] https://en.wikipedia.org/wiki/Academic_grading_in_Nepal

[3] https://ieeexplore.ieee.org/document/10430341

[4] https://www.geeksforgeeks.org/student-information-management-system/

[5] https://www.scaler.com/topics/student-record-management-system-c-project/