Project Report (CSP100)

on

Student Data Management System

Department of Computer Science and Engineering

Submitted To :

DR. JASPINDER KAUR

Assistant Professor

CSE Department

Submitted By :

Abhiraj Banerjee (221210005)

Aditya Nayak (221210009)

Akshat (221210014)

Anish Kumar (221210019)

Ankit Pal (221210021)

NATIONAL INSTITUTE OF TECHNOLOGY DELHI

2023

TABLE OF CONTENT

DECLARATION

Academic honesty and integrity are fundamental principles that uphold the credibility and reliability of scholarly work. As responsible individuals, we acknowledge and affirm that this written submission reflects our own thoughts and ideas, expressed in our own words. It serves as a testament to our intellectual capabilities, critical thinking, and understanding of the subject matter.

While formulating this submission, we have diligently ensured that any external ideas, concepts, or words derived from other sources are appropriately cited and referenced. We firmly believe in giving credit where it is due, acknowledging the contributions of others to the body of knowledge. By accurately citing and referencing these original sources, we not only demonstrate academic integrity but also foster a spirit of academic collaboration and respect.

We understand the importance of avoiding misrepresentation, fabrication, or falsification of any idea, data, fact, or source within our submission. It is our responsibility to present accurate and reliable information, ensuring that our work contributes to the pursuit of knowledge in an ethical and honest manner. Any violation of these principles undermines the integrity of our academic pursuits and compromises the trust that is placed in us as scholars.

In adhering to the principles of academic honesty, we have been meticulous in our research, diligently reviewing and examining a wide range of credible sources. We have sought to validate the accuracy and authenticity of the information we have utilized, ensuring that it aligns with recognized standards of scholarship. We have critically analyzed the material, employing our own judgement and interpretation to construct a well-informed and coherent written submission.

Furthermore, we recognize that failing to cite or obtain proper permission from sources, when necessary, can have serious consequences. It is essential to respect the intellectual property rights of others and abide by legal and ethical guidelines. By acknowledging the contributions of others through appropriate citation and seeking permission when required, we demonstrate our commitment to fostering an environment of fairness, transparency, and mutual respect.

We acknowledge that any violation of academic integrity principles can result in disciplinary action by our educational institution. Such violations can also have legal implications, as they may infringe upon the rights of original authors or sources. Therefore, we are committed to upholding the highest standards of academic honesty and integrity, striving to produce work that is a genuine reflection of our own ideas while crediting the contributions of others.

In conclusion, this expanded declaration affirms our commitment to academic honesty and integrity. It highlights our acknowledgment of the importance of citing and referencing sources, avoiding misrepresentation, and adhering to ethical standards.

ACKNOWLEDGEMENT

We extend our heartfelt gratitude to the Honorable Director, Prof. Ajay K. Sharma, for his unwavering support and dedication in providing us with outstanding facilities and a conducive environment throughout our project work. His visionary leadership and commitment to fostering excellence have played a vital role in shaping our learning experience. We are truly grateful for his guidance and the opportunities he has bestowed upon us.

We would like to express our deepest appreciation to our Supervisor, Dr. Jaspinder Kaur, for her invaluable guidance, expertise, and unwavering commitment to our project. Her profound knowledge and constant support have been instrumental in shaping our research and helping us overcome challenges.

Dr. Jaspinder Kaur's meticulous feedback and insightful suggestions have not only enriched our work but also expanded our understanding of the subject matter. We are truly fortunate to have had the privilege of working under her guidance.

Furthermore, we would like to extend our sincere thanks to our parents and well-wishers whose unwavering support has been our pillar of strength throughout this phase of our academic journey.

Their love, encouragement, and sacrifices have been a constant source of motivation for us. We are deeply indebted to them for their belief in our abilities and their unwavering support, which has enabled us to persevere and achieve our goals.

We would also like to acknowledge the contributions of our peers and classmates who have been an invaluable part of our journey.

Their collaborative spirit, constructive discussions, and shared experiences have enriched our learning process and broadened our perspectives. We are grateful for the camaraderie and the stimulating intellectual environment they have fostered.

Moreover, we would like to express our appreciation to the faculty members and staff of our institution who have played a significant role in shaping our academic development.

In conclusion, our sincere appreciation goes to the Honorable Director, Prof. Ajay K. Sharma, our Supervisor, Dr. Jaspinder Kaur, our parents and well-wishers, our peers and classmates, the faculty members and staff of our institution, as well as the various organizations, institutions, and individuals who have supported us.

Their collective efforts and belief in our abilities have been invaluable, and we are truly grateful for their contributions to our project work.

ABSTRACT

This abstract provides an overview of the research conducted during the development of our student data management system. To ensure a solid foundation for our system, we conducted comprehensive research on popular data management systems and studied existing implementations. By analyzing various approaches to data storage, file handling techniques, and data organization methods, we were able to gain valuable insights into industry best practices and identify the essential features and mechanics to include in our system.

Researching popular data management systems allowed us to understand the challenges and requirements commonly encountered in similar projects. We examined different systems used in educational institutions and other industries to gather insights into their functionalities, user interfaces, and overall effectiveness. This exploration provided us with a broader perspective on the possibilities and limitations of data management systems, enabling us to make informed decisions during the development process.

Additionally, we studied existing implementations of data management systems to learn from their successes and challenges. By reviewing case studies and real-world examples, we gained practical knowledge about the best practices, common pitfalls, and potential improvements in data management system design and implementation. This research allowed us to leverage existing knowledge and build upon proven methodologies, avoiding potential pitfalls and ensuring the effectiveness of our system.

Throughout the research phase, we focused on understanding different data storage options, such as databases, file systems, and data structures. We also explored various file handling techniques, including reading and writing data to files, parsing data from different formats, and managing file permissions. Furthermore, we investigated data organization methods, such as indexing, sorting, and categorizing, to ensure efficient data retrieval and manipulation.

By conducting thorough research, we were able to gather a wealth of knowledge that guided the development of our student data management system. The insights gained from studying popular data management systems and existing implementations provided us with a strong foundation to design and implement a robust and effective solution. This research phase enabled us to leverage industry best practices, avoid potential pitfalls, and create a system that meets the needs of our users while adhering to high standards of data management.

INTRODUCTION

The management of student data is of utmost importance in today's educational institutions. Efficient data management systems play a crucial role in ensuring smooth administrative operations, providing accurate information, and facilitating effective decision-making. In this Project Report, we present a comprehensive Student Data Management System implemented in the C programming language.

The system aims to provide an efficient, user-friendly, and digital solution for managing the data of college students. By leveraging fundamental programming concepts such as loops, conditionals, functions, and file handling, we enhance the efficiency and accuracy of student data management.

Educational institutions traditionally relied on manual paperwork and physical files to maintain student records. However, this approach proved to be time-consuming, prone to errors, and vulnerable to misplacement or damage. With the rapid advancement of technology, it has become imperative to adopt digital solutions that streamline data management processes, offer enhanced accessibility, and ensure data security.

The Student Data Management System implemented in the C programming language offers an effective solution to overcome the challenges faced by traditional data management methods. By harnessing the power of programming, we can automate data entry, storage, retrieval, and organization, reducing manual effort and minimizing the possibility of errors. This digital solution revolutionizes the way student data is collected, stored, and managed, offering significant benefits to educational institutions.

In the realm of modern education, efficient management of student data is critical for the smooth functioning of educational institutions. Traditional methods of maintaining student records using manual paperwork and physical files have proven to be time-consuming, error-prone, and cumbersome. To address these challenges and improve data management processes, we have developed a comprehensive Student Data Management System implemented in the C programming language. By harnessing the power of programming, this system offers an effective solution that automates data entry, storage, retrieval, and organization, significantly reducing manual effort and minimizing the possibility of errors. This digital solution revolutionizes the way student data is collected, stored, and managed, offering substantial benefits to educational institutions.

The Student Data Management System is designed to streamline the management of student information, providing an efficient and user-friendly interface for administrators and authorized personnel to input, access, and update student data. By centralizing student information in a digital format, the system ensures that critical data is easily accessible, organized, and readily available when needed.

One of the primary advantages of the Student Data Management System is the automation of data entry processes. Instead of relying on manual data entry, which is time-consuming and prone to errors, the system allows administrators to input student data electronically. This automation

eliminates the need for redundant data entry tasks, significantly reducing the chances of errors and inconsistencies in the data. It also saves valuable time and resources, enabling administrators to focus on other important aspects of their roles.

The system's data storage and retrieval capabilities are also enhanced through the use of programming techniques. By leveraging the power of programming, the system can efficiently store and retrieve large volumes of student data. It employs data structures and algorithms that optimize data storage and retrieval, ensuring fast and accurate access to student records. This quick and efficient access to student information empowers administrators to make informed decisions, respond promptly to queries, and effectively manage various administrative processes.

Furthermore, the Student Data Management System facilitates data organization based on various criteria such as attendance, marks, library dues, and more. Administrators can generate reports and sort data based on these parameters, allowing for easy categorization and analysis of student information. This categorization feature enables administrators to identify trends, monitor student performance, and efficiently manage student-related processes. By providing a structured view of student data, the system enhances decision-making processes, leading to improved academic support and overall administrative efficiency.

The implementation of the Student Data Management System in the C programming language allows for greater flexibility and customization. C is a powerful and widely-used programming language that provides extensive control over program execution and data manipulation. It offers a rich set of features such as loops, conditionals, functions, and file handling, which enable the development of robust and efficient software applications. By utilizing these programming constructs, we ensure that the Student Data Management System operates smoothly, handles complex data operations, and maintains a user-friendly interface.

Moreover, the system incorporates file handling mechanisms to ensure data persistence. Student data is stored in external files, which allows for data continuity even after the program terminates. This feature ensures that student records are securely stored and can be accessed and updated across multiple sessions. It also facilitates data backup and restore functionalities, providing safeguards against potential data loss or system failures. With data integrity and reliability, the Student Data Management System becomes a dependable tool for educational institutions to manage student information effectively.

In conclusion, the Student Data Management System implemented in the C programming language offers an efficient and innovative solution for educational institutions to overcome the challenges posed by traditional data management methods. By harnessing the power of programming, the system automates data entry, storage, retrieval, and organization, leading to reduced manual effort and enhanced data accuracy. It revolutionizes the way student data is collected, stored, and managed, providing substantial benefits such as improved administrative efficiency, quick data retrieval, and customized.

data categorization. The use of the C programming language ensures a robust and flexible system that can adapt to the specific needs of educational institutions. By adopting this digital solution, educational institutions can streamline their data management processes and focus more on delivering quality education and support to students.

The primary objective of the Student Data Management System is to provide a user-friendly interface that simplifies the collection and storage of student information. The system enables administrators and authorized personnel to input data for new students, including personal details,

academic records, attendance information, library dues, and more. By centralizing student data in a digital format, the system ensures that critical information is easily accessible and readily available whenever required.

One of the key features of the system is its ability to categorize student data based on various parameters. Administrators can generate reports and sort data based on attendance records, marks, library dues, or any other customizable criteria. This categorization functionality facilitates efficient management of student-related processes, enables the identification of trends, and allows for monitoring and evaluation of student performance. By providing quick access to relevant information, the system empowers administrators to make informed decisions, enhancing overall administrative efficiency.

The implementation of the Student Data Management System in the C programming language relies on essential programming concepts such as loops, conditionals, functions, and file handling. These programming constructs provide the necessary tools to automate repetitive tasks, perform calculations, validate inputs, and interact with external data files. By leveraging these powerful programming tools, we ensure that the system operates smoothly, efficiently, and accurately, handling complex data operations while maintaining a user-friendly interface.

Furthermore, the system incorporates file handling mechanisms to enable data persistence. Student data is stored in external files, ensuring that information is retained even after the program terminates. This feature allows for seamless data continuity, enabling administrators to access and update student records across multiple sessions, thereby facilitating long-term data management.

The Student Data Management System is designed with simplicity and ease of use in mind. The user interface is intuitive, providing clear instructions for data entry, retrieval, and report generation. The system incorporates robust error handling mechanisms to validate user inputs and prevent data corruption. Additionally, it offers data backup and restore functionalities, safeguarding against potential data loss or system failures, thus ensuring data integrity and reliability.

In conclusion, the Student Data Management System implemented in the C programming language provides an efficient, user-friendly, and digital solution for managing student data in educational institutions. By digitizing data management processes, the system enhances accessibility, accuracy, and administrative efficiency. Leveraging fundamental programming concepts, the system automates tasks, categorizes data, and ensures data persistence. This Project Report serves as a comprehensive overview of the system's functionality, implementation, and benefits. Through this digital solution, we aim to streamline student data management, contribute to the effective functioning of educational institutions, and empower administrators with the necessary tools to make informed decisions.

MOTIVATION

The motivation behind developing a student data management system using the code we learned in our first year of college study stems from the desire to leverage our programming skills to create a practical and efficient solution for managing student data. As students ourselves, we have experienced the challenges and complexities associated with traditional data management methods. We observed the need for a streamlined and organized system that could effectively handle the vast amount of student information generated in educational institutions.

By applying the code concepts we learned in our first year, we aim to demonstrate the practicality and real-world applicability of our programming knowledge. We wanted to prove that even with a foundational understanding of programming, we can develop a useful and functional system that solves a real problem.

Efficiently processing data from text files was one of the primary goals in our development process. We recognized that student data, including personal details, academic records, attendance information, and library dues, could be stored in text files, offering a flexible and accessible data storage solution. By utilizing our programming skills, we could extract relevant information from these text files, perform calculations, and organize the data in a structured manner.

Creating an organized student management system was another driving factor in our motivation. We witnessed the struggles faced by administrators in manually managing student records, which often led to errors, inefficiencies, and a lack of timely access to crucial information. By developing a system that automates data processing, storage, retrieval, and organization, we aimed to provide a solution that enhances the efficiency, accuracy, and accessibility of student data management.

In conclusion, the motivation behind creating a student data management system using the code we learned in our first year of college study stems from the desire to address the challenges faced in traditional data management methods and showcase the practical applications of our programming knowledge. By efficiently processing data from text files and organizing student information, we aim to provide an effective and user-friendly solution that enhances the efficiency, accuracy, and accessibility of student data management. This project serves as an opportunity to apply our programming skills, deepen our understanding of programming concepts, and gain valuable experience in software development.

OBJECTIVE

The goal of this project is to develop a fully functional student data management system that accurately captures and manages student information. The system will provide a user-friendly interface for data entry, allow users to track student progress, and enable efficient data retrieval and analysis.

The primary objective is to implement the necessary data and mechanics to ensure comprehensive student data management. This includes capturing essential details such as student names, identification numbers, contact information, academic records, attendance records, and library dues. By accurately representing this data, the system will serve as a centralized repository of student information, making it easily accessible and organized.

To enhance usability, the system will offer a user-friendly interface for entering student data. It will provide clear prompts and input fields, guiding users through the data entry process. By implementing intuitive design principles, we aim to make the system accessible to users with varying levels of technical expertise.

Tracking student progress is another key aspect of the project. The system will enable users to monitor and analyze student performance over time. This includes tracking attendance records and identifying students with short attendance, which can help in addressing potential issues and providing timely interventions. By implementing these features, the system will support efficient monitoring and management of student progress.

In addition to data entry and progress tracking, the system will also facilitate data retrieval and display. Users will be able to search for specific students and view their corresponding data. This feature will enable quick access to student information, making administrative tasks more efficient and streamlined. The system will also provide functionalities to generate reports and analyze data, empowering users to make informed decisions based on the available information.

Overall, the project aims to create a fully functional student data management system that meets the objectives of accurate data representation, progress tracking, user-friendly data entry, and efficient data retrieval.

TOOLS & KNOWLEDGE ASPECTS

The development of our student data management system involved a structured and systematic approach to ensure its successful implementation. The methodology encompassed various phases, including planning, research, implementation, and testing. By following this methodology, we aimed to create a robust and efficient system that fulfills the identified objectives and requirements.

The project commenced with the planning phase, where we defined the scope, objectives, and requirements of the data management system. This phase was crucial in establishing a clear understanding of what we aimed to achieve with our system. We identified the key functionalities, such as data entry, storage, retrieval, and organization, and determined the desired outcomes of the system. Additionally, we outlined the specific requirements, considering factors such as data security, user-friendliness, and scalability.

To ensure that our system was built on a solid foundation, we conducted thorough research on popular data management systems and studied existing implementations. This research provided valuable insights into industry best practices and helped us identify the essential features and mechanics to include in our system. We analyzed different approaches to data storage, file handling techniques, and data organization methods. This research phase allowed us to leverage existing knowledge and build upon proven methodologies to create an effective solution.

The implementation phase involved the actual coding of the system using the C programming language. Functions played a crucial role in modularizing our code, enabling us to break down complex tasks into smaller, manageable units. Additionally, we utilized 2-D strings to handle and manipulate student data, ensuring accurate representation and efficient data processing.

In conclusion, the methodology employed in the development of our student data management system involved careful planning, thorough research, implementation using the C programming language, and rigorous testing. By following this methodology, we created a robust and efficient system that addresses the identified objectives and requirements. The use of file handling, functions, and 2-D strings in our implementation showcases the practical application of fundamental programming concepts. Through systematic execution and attention to detail, we ensured the success of our project and the creation of an effective student data management system.

WORKING PRINCIPLE

In order to meet the requirements of the student data management system, we proceeded with designing a robust and efficient architecture. The architecture focuses on utilizing appropriate data structures and algorithms to represent the database, track new data, and perform necessary operations. The design encompasses the use of file handling concepts and functions to read and write data from student management files.

One of the key considerations in the design was the selection of appropriate data structures to store and manage student data. We identified the need for data structures such as arrays, linked lists, or structures to represent student information. These data structures allow us to store and organize data in a structured manner, facilitating efficient retrieval and manipulation.

The architecture utilizes functions such as `file_read_name`, `file_read_cgpa`, `file_read_branch`, and `file_read_semester` to read various data from the student management files. Each of these functions is designed to read specific data fields from the file. The logic implemented in these functions involves iterating through the text file using loops and identifying the relevant lines where the required data is present. By extracting the data after specific characters or delimiters, we can access the values of the respective fields.

In summary, the design and architecture of the student data management system prioritize the use of appropriate data structures, file handling concepts, and modular functions. The architecture ensures efficient data retrieval, manipulation, and storage, while the use of standardized data formats facilitates seamless data processing. By adhering to sound design principles and considering the scalability and extensibility of the system, we aim to create a robust and adaptable solution for managing student data efficiently.

FLOW OF CONTROL

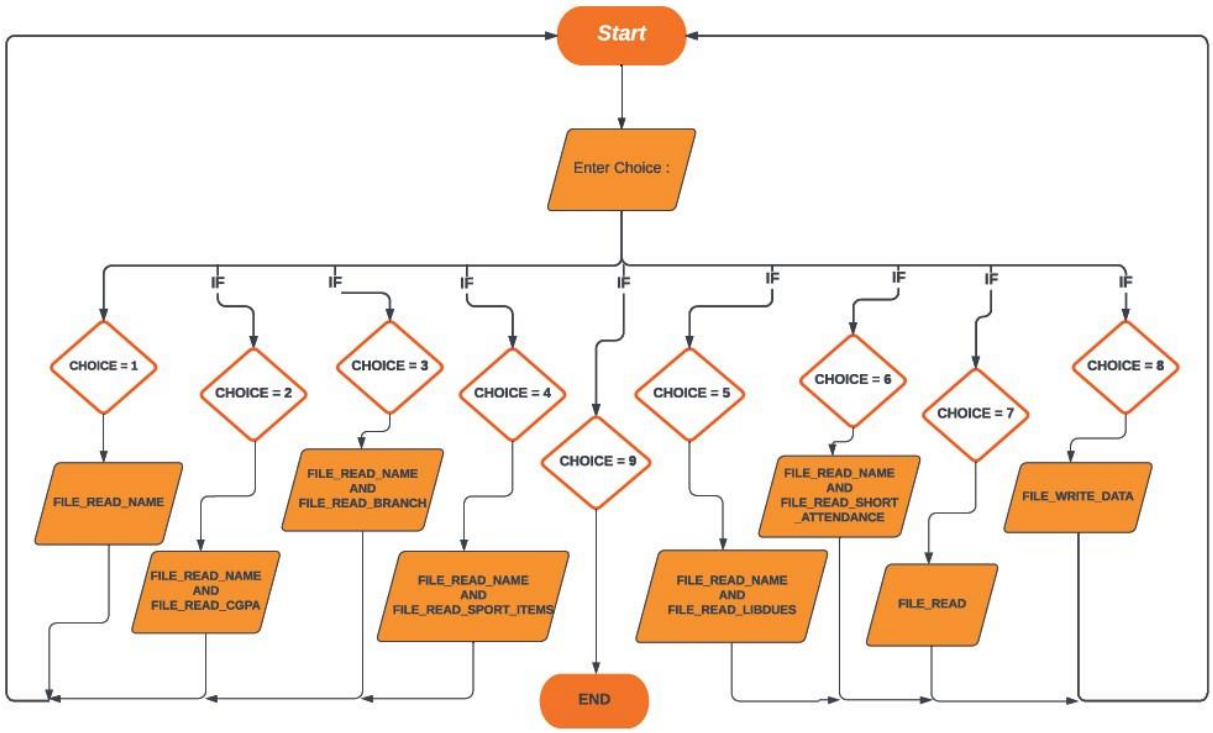

Fig 1. Flowchart

METHODOLOGY

The implementation of the Student Data Management System involved utilizing the features and functionalities provided by the C programming language. We divided the development tasks among team members to ensure efficient and focused implementation. Let's explore the key aspects of the code implementation.

1. File Handling:
One crucial aspect of the implementation was file handling. We utilized C's file handling capabilities to read and write student data to external text files. By opening files in read or write mode, we could access the stored data and modify it as required. We implemented functions to read data from files, update data, and write new data to the files. This file handling mechanism ensured data persistence and allowed for easy retrieval and manipulation of student information.

2. Input Handling:
Another important aspect was handling user input. We designed a user-friendly interface that prompted users to enter student data using appropriate prompts and input fields. We utilized C's input/output functions to read user input and validate it to ensure accurate data entry. By implementing error handling mechanisms, we could validate user inputs, preventing data corruption and ensuring the integrity of the system.

3. Data Structures and Functions:
To organize and manipulate student data effectively, we leveraged C's data structures and functions. We defined structures to represent student information, such as name, ID, contact details, academic records, and attendance. These structures allowed us to store and access multiple data elements as a single entity. We implemented functions to perform various operations on student data, such as adding new students, updating existing records, and retrieving information based on specific criteria.

4. Sorting and Searching:
Efficient data retrieval and analysis were achieved by implementing sorting and searching algorithms. We utilized sorting algorithms, such as bubble sort or quicksort, to sort student records based on specific attributes such as names or IDs. This facilitated easy searching and organization of data, enhancing the system's efficiency. We also implemented searching algorithms, like linear search or binary search, to locate specific student records based on user queries.

In conclusion, the implementation of the Student Data Management System involved utilizing C programming language's features and functionalities. File handling, input handling, data structures, sorting and searching algorithms, and user interface design were the key aspects of the code implementation. By dividing tasks among team members and following best practices, we aimed to create a well-structured and efficient codebase. This code implementation served as the foundation for a fully functional Student Data Management System, enabling effective management and organization of student information.

FUNCTIONS USED

- The program includes several function prototypes for reading and writing student data from/to files.
- In the main function, a menu is displayed with several options for performing different operations on student records.
- The user is prompted to enter a choice from the menu, and based on the choice, the corresponding operation is performed.
- The program reads the existing student records from text files and stores them in memory.
- Depending on the user's choice, different information is extracted from the student records and displayed on the screen.
- If the user chooses option 7, a new student record is created by taking input from the user and writing it to a new text file.
- The program continues to display the menu until the user chooses to exit.

Functions :

```
// list of all the functions
void file_read(int roll);
int file_null(int roll);
void file_read_name(int roll, char *result);
void file_read_semester(int roll, char *result);
void file_read_branch(int roll, char *result);
void file_read_cgpa(int roll, char *result);
void file_read_libdues(int roll, char *result);
void file_read_presentdays(int roll, char *result);
void file_read_short_attendance(int roll, char *result);
void file_read_sportitems(int roll, char *result);
void file_write_data(int new_roll);
```

Fig 2 : List of function

MENU :

```
*--------------------*--------------------*----------------------
--------*
MENU :

1.Print the names of all the students

2.Print the names and CGPA of all the students

3.Print the names the students of a particular branch

4.Print the names and Sports Items issued by all the students

5.Print the names and Library dues of the students

6.Print the names of all the students with Short-Attendance

7.Print the data of a particular student

8.Enter data of a new student

9. Exit
*--------------------*--------------------*----------------------
--------*
Enter the choice :
```

Fig 3 : Menu

TEXT FILE :

```
NAME    :  ANKIT
SEMESTER :  2
BRANCH  :  ELECTRONICS
ROLL    :  2
CGPA    :  6.8
LIBRARY DUES : RS.25
PRESENT DAYS : 250
% ATTENDANCE : 83.3%
TOTAL WORKING DAYS : 300
SHORT ATTENDANCE(Y/N) : N
SPORT ITEM ISSUED : BADMINTON
```

Fig 4 :Text File

When we run our program, it displays a menu as shown in figure 3.

We choose a option and our program gives the desired result.

Fig 2 shows various functions included in our code.

```
void file_read(int roll);
```

1 . This function take roll nuber as argument and prints all of the data of the student with the corresponding roll number. The function scan data from file line by line and prints it line by line.

```
int file_null(int roll);
```

2 . This function checks if the file with the name of roll number of student is available or not.

This function is made to count the number of student's data entered.

```
void file_read_name(int roll, char *result);
```

3 . This function reads the name of the student from the file.

```
void file_read_semester(int roll, char *result);
```

4 . This function reads the semester of the student from the file.

```
void file_read_branch(int roll, char *result);
```

5 . This function reads the semester of the student from the file.

```
void file_read_cgpa(int roll, char *result);
```

6 . This function reads the cgpa of the student from the file.

```
void file_read_libdues(int roll, char *result);
```

7 . This function reads the library dues of the student from the file.

```
void file_read_presentdays(int roll, char *result);
```

8 . Thia function read the present days of a student.

```
void file_read_short_attendance(int roll, char *result)
```

9 . This function reads the short attendance of the students.

```
void file_read_sportitems(int roll, char *result);
```

10 . This function reads the sport items issued by the students.

```
void file_write_data(int new_roll);
```

11 . This function writes data of new students.

When we choose option 1, the name of students are printed as shown in fig.5

```
Enter the choice : 1

The names of the Students are :
ADI

ANKIT

ABHIRAJ

AKHILESH

ANIMESH

ABHINAV

ROHAN

ABHAY

JOHN

SAJAL

*******************************************
DO YOU WANT TO CONTINUE (Y/N)  :
```

Fig 5:Choice 1

When we choose option 2, the name and cgpa of students are printed as shown in fig.6

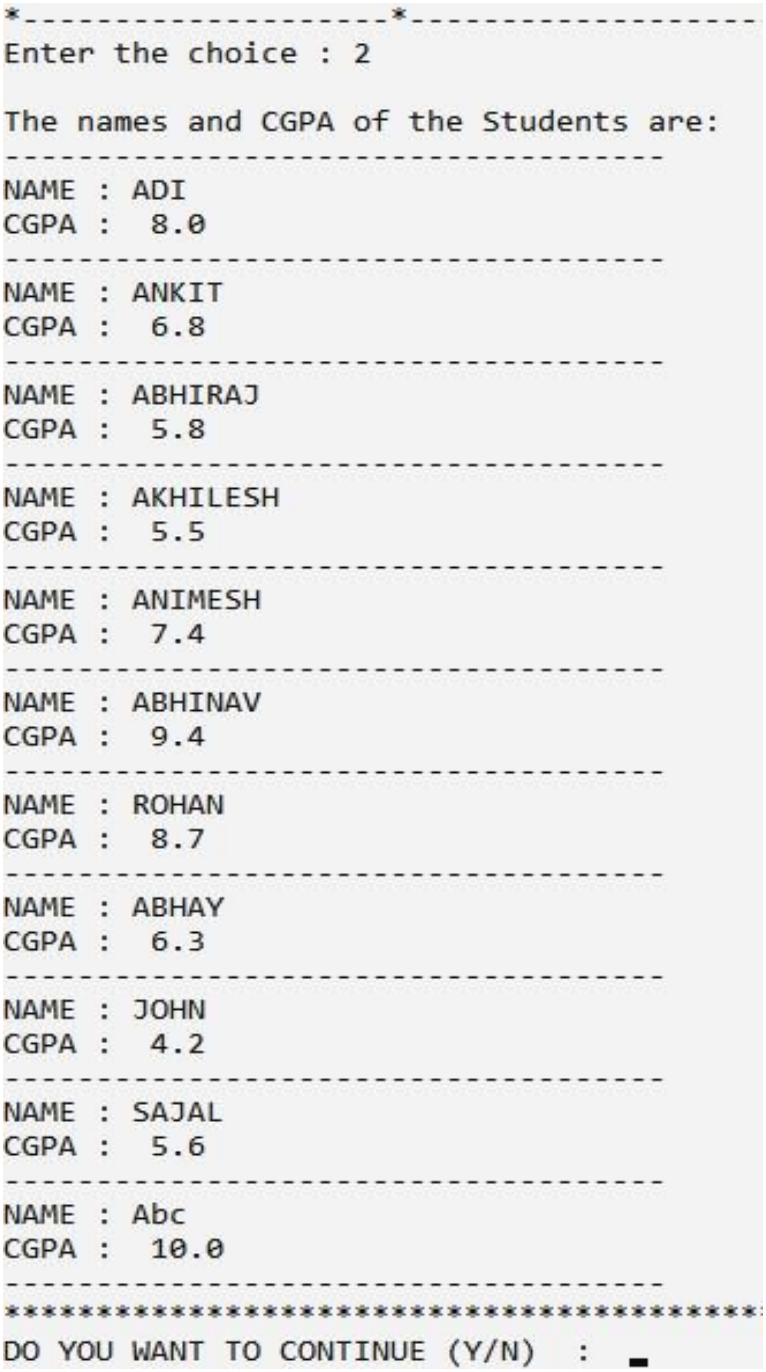

Fig 6: Choice 2

When we choose option 3, the name and branch of students of a particular branch are printed as shown in fig.7

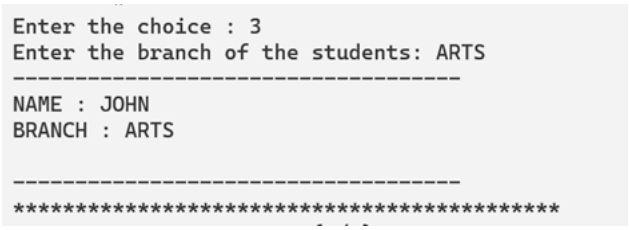

Fig 7: Choice 3

Here we have chosen arts.

When we choose option 4, the name and sports item issued by students are printed as shown in fig.8

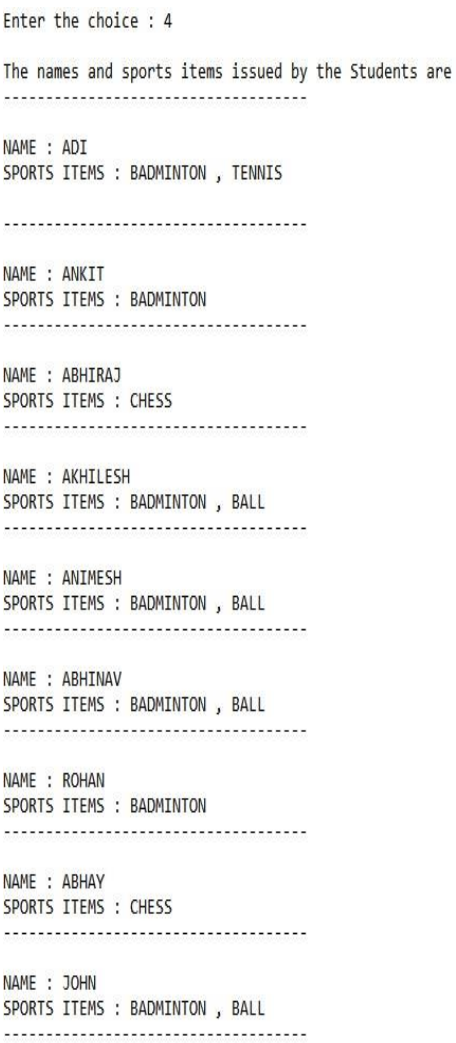

Fig 8: Choice 4

When we choose option 5, the name and library dues of students with library dues are printed as shown in fig.9

```
Enter the choice : 5

The names and library dues of the Students are
-------------------------------------
NAME : ADI
LIBRARY DUES : 6.00
-------------------------------------
NAME : ANKIT
LIBRARY DUES : 25.00
-------------------------------------
NAME : ABHIRAJ
LIBRARY DUES : 15.00
-------------------------------------
NAME : ANIMESH
LIBRARY DUES : 22.00
-------------------------------------
NAME : ROHAN
LIBRARY DUES : 12.00
-------------------------------------
NAME : JOHN
LIBRARY DUES : 25.00
-------------------------------------
NAME : SAJAL
LIBRARY DUES : 13.00
-------------------------------------
*****************************************
DO YOU WANT TO CONTINUE (Y/N)  :
```

Fig 9: Choice 5

When we choose option 6, the name of students with short attendance are printed as shown in fig.10

```
Enter the choice : 6

The names of the students with short attendance are:
-------------------------------------
NAME : ADI
SHORT ATTENDANCE : Y

-------------------------------------
NAME : ABHIRAJ
SHORT ATTENDANCE : Y

-------------------------------------
NAME : ABHAY
SHORT ATTENDANCE : Y

-------------------------------------
NAME : SAJAL
SHORT ATTENDANCE : Y

-------------------------------------
*****************************************
DO YOU WANT TO CONTINUE (Y/N)  :
```

Fig 10: Choice 6

When we choose option 7, we can see the data of a specific student after putting his roll no. as shown in fig.11

```
Enter the choice : 7
Enter the roll number of the student : 6
-------------------------------------
NAME      :   ABHINAV
SEMESTER  :   4
BRANCH    :   MINING
ROLL      :   6
CGPA      :   9.4
LIBRARY DUES : RS.10
PRESENT DAYS : 250
% ATTENDANCE : 83.3%
TOTAL WORKING DAYS : 300
SHORT ATTENDANCE(Y/N) : N
SPORT ITEM ISSUED : BADMINTON , BALL
-------------------------------------
*****************************************
DO YOU WANT TO CONTINUE (Y/N)  :
```

Fig 11: Choice 7

When we choose option 8, we can add the data of a student after putting his roll no. as shown in fig.12.

```
Enter the choice : 8
Enter the details of student -
NAME : Anshul
SEMESTER : 4
BRANCH : ARTS
CGPA : 5.6
LIBRARY DUES : 7.00
PRESENT DAYS : 250
SPORT ITEMS : BADMINTON
Data written to the file successfully.
*****************************************
```

Fig 12: Choice 8

The text file will be saved with roll number as name. and the format in which it stores the data is shown in fig.4

LEARNINGS

The development of the Student Data Management System provided us with valuable insights and skills in the field of development and programming. Throughout the project, we encountered various challenges and gained hands-on experience that helped us enhance our problem-solving abilities and improve our overall programming skills.

One significant aspect of our learning was designing and implementing data with well-defined rules and mechanics. We had to carefully analyze the requirements and determine the appropriate data structures and variables to represent student information accurately. This involved understanding the relationships between different data elements and implementing them in a way that facilitated efficient data management and retrieval. By focusing on data organization and structuring, we learned how to optimize data representation and improve the system's overall performance.

Another important area of learning was validating user input and implementing new input options. We had to ensure that the system could handle a wide range of input scenarios and validate the data entered by users to maintain data integrity. This involved implementing input validation checks, such as checking for the correct data format, handling invalid inputs, and providing meaningful error messages to guide users. By addressing these challenges, we improved our skills in handling user input effectively and ensuring the robustness of the system.

Furthermore, the project allowed us to enhance our skills in code organization, documentation, and user manual creation. We learned the importance of writing clean, modular, and well-documented code. We followed best practices in code organization, such as using meaningful variable and function names, breaking down complex tasks into smaller, manageable functions, and avoiding code duplication. Additionally, we focused on documenting our code thoroughly, including comments that explained the purpose and functionality of different code sections. This practice not only improved code readability but also made it easier for other developers to understand and maintain the codebase in the future.

Overall, the development of the Student Data Management System served as a valuable learning experience for us. It allowed us to apply the programming knowledge we gained in our college studies to a real-world project. Through this project, we improved our problem-solving abilities, gained insights into data management, enhanced our code organization and documentation skills, and learned how to create a user-friendly system. These skills and experiences will undoubtedly benefit us in future projects and contribute to our growth as developers.

FUTURE SCOPE

1. Implementing Additional Features:

To enhance the functionality of the system, additional features can be incorporated. For example, the ability to add and manage various data categories, such as extracurricular activities, discipline records, or project details, would provide a more comprehensive student profile. Furthermore, implementing sorting options based on different criteria, such as alphabetical order or academic performance, would facilitate easier data organization and retrieval.

2. Customizable Options:

Adding customizable options would allow users to tailor the system according to their specific needs. For instance, providing the flexibility to set the required number of classes to prevent short attendance can be a valuable feature. Users could configure the system to generate alerts or notifications when a student's attendance falls below the specified threshold, enabling timely intervention.

3. Multi-User Mode:

Expanding the system to include a multi-user mode would enable collaborative data entry and access. Implementing this mode could involve creating user accounts and granting different levels of access and privileges to administrators, teachers, and students. Additionally, incorporating an online component would allow users to access and update the data remotely, providing convenience and accessibility from anywhere with an internet connection.

5. Testing and User Feedback:

Continued testing and gathering user feedback are essential for identifying any potential issues or areas for improvement. Conducting thorough testing, including both functional and usability testing, can help uncover any bugs or errors that need to be addressed. Collecting user feedback, either through surveys or user testing sessions, can provide valuable insights into the system's strengths and weaknesses, allowing for targeted improvements and refinements.

By implementing these future enhancements, the Data Management System can evolve into a more robust and versatile tool for efficient student data management. Regular updates and improvements based on user needs and technological advancements will ensure that the system remains relevant and effective in meeting the evolving requirements of educational institutions.

FINAL REVIEW AND SUBMISSION

Before submitting the Data Management project, we performed a comprehensive final review to ensure the system's quality, functionality, and documentation. This review encompassed multiple aspects, including testing, user manual review, and code evaluation.

Firstly, we conducted thorough testing of all functionalities to verify their accuracy and reliability. We followed a systematic approach, executing test cases to cover various scenarios and inputs. By analyzing the system's behavior and comparing it against expected results, we identified and addressed any bugs or errors. Through rigorous testing, we aimed to ensure that the Data Management System operated smoothly and produced accurate outputs.

Simultaneously, we reviewed the user manual to guarantee its accuracy and clarity. We carefully examined each section, including instructions for installation, data entry, and report generation. Our goal was to ensure that the user manual provided comprehensive guidance to users, regardless of their technical expertise. We revised and refined the instructions, incorporated any changes or new features implemented during development, and addressed any ambiguities or gaps in the documentation.

Once the final review was completed, we prepared the Data Management project for submission. We ensured that all relevant documentation, including the user manual and source code, was compiled and organized appropriately. We double-checked that the project fulfilled the initial requirements and objectives, and that it met the desired quality standards.

By conducting a thorough final review, we aimed to deliver a high-quality Data Management System that met the specified requirements and provided a reliable solution for managing student data. The review process helped us identify and rectify any issues, resulting in an improved system ready for submission.

CONCLUSION

In conclusion, we have successfully developed a Student Data Management System that fulfills the specified objectives and requirements. The system encompasses a range of features that allow for efficient data management and user-friendly interaction.

One of the key strengths of our Student Data Management System is its ability to display multiple types of information as per the user's requirements. Whether it is attendance records, academic performance, library dues, or any other relevant data, the system provides a comprehensive view of the student's information. This feature enables administrators, teachers, and other authorized users to access and analyze the data they need quickly and conveniently.

The system also incorporates user-friendly controls for entering new data. Through an intuitive interface, users can easily input and update student information, ensuring that the system remains up to date and accurate. Additionally, the system incorporates validation mechanisms to ensure the integrity and accuracy of the entered data, minimizing the possibility of errors and inconsistencies.

Another notable feature of our Student Data Management System is its ability to generate appealing and organized data reports. Users can easily generate reports with just a few clicks, allowing for efficient data analysis and decision-making processes. The system presents the data in a structured and visually appealing format, enhancing the overall user experience and facilitating the understanding of complex information.

Overall, our Student Data Management System provides an efficient and effective solution for managing student data in educational institutions. By automating data entry, storage, retrieval, and organization, the system minimizes manual effort, reduces errors, and enhances administrative efficiency. The user-friendly interface and comprehensive data display options make it accessible and convenient for users of varying technical expertise.

Through the development of this Student Data Management System, we have gained valuable insights into the process of designing and implementing software solutions. We have honed our skills in programming, data management, and user interface design.

In conclusion, our Student Data Management System stands as a reliable and efficient tool for educational institutions to streamline their student data management processes. We believe that our system will contribute significantly to the effective functioning of educational institutions, enabling better data management and facilitating informed decision-making.

COLLABORATION

The development of the Data Management System was a collaborative endeavor that brought together the diverse skills and expertise of our team members. Recognizing the importance of leveraging individual strengths, we allocated responsibilities based on each team member's proficiency and interest in specific areas.

Some team members concentrated on developing the core file handling logic and ensuring seamless data input and retrieval. They worked diligently to implement efficient algorithms for reading and writing data to external files, ensuring the integrity and security of the stored information. Their expertise in file handling and data manipulation played a pivotal role in the successful implementation of the system's key functionalities.

Other team members focused on input validation mechanisms, ensuring that the system could handle various types of user input and promptly detect and respond to any errors or inconsistencies. They meticulously designed and implemented validation rules and error handling routines to guarantee the accuracy and reliability of the entered data. Their attention to detail and commitment to ensuring data integrity significantly contributed to the system's overall robustness.

Regular team meetings and communication channels were established to facilitate effective collaboration throughout the development process. These sessions served as platforms for discussing progress, sharing ideas, and addressing any challenges encountered along the way. By maintaining open lines of communication, we fostered a collaborative and supportive environment where team members could exchange knowledge, seek assistance when needed, and collectively find solutions to complex problems.

Collaboration also extended beyond technical aspects to encompass project management and documentation. Team members collaborated on creating and updating project timelines, ensuring that tasks were completed within the allocated timeframes. Additionally, they collaborated on producing comprehensive documentation, including the user manual and system documentation, to provide clear instructions and insights for users and future developers.

In conclusion, the development of the Data Management System was a testament to the power of collaboration. By harnessing the diverse skills and expertise of our team members, we were able to create a robust and efficient system that met the objectives and requirements of the project. Through regular communication, shared responsibility, and a collaborative mindset, we successfully navigated the challenges encountered during the development process and emerged with a cohesive and high-performing solution.

CONTRIBUTION

CODE-Aditya Nayak , Akshat , Anish Kumar

REPORT-Aditya Nayak , Akshat , Ankit Pal

POSTER-Ankit Pal

VIDEO – Abhiraj Banerjee,Anish Kumar

PRESENTATION - Akshat

REFERENCES

https://www.geeksforgeeks.org/basics-file-handling-c/ for learning file handling.

https://www.javatpoint.com/file-handling-in-c for gaining basic knowledge about functions in file handling.

Book-Let Us C by Yashavant Kanetkar