Project Report

Training Allotment System

Sumit E22CSEU1284

Priyanshu saini E22CSEU1280

Aditya E22CSEU1270

A report submitted in part fulfilment of the degree of

B.Tech in Computer Science

Dr. Sangeeta Kumari

Department of Computer Science
Royal Holloway, University of London

November 10, 2023

Declaration

This report has been prepared on the basis of my own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count:

Student Name: Sumit

Date of Submission: 16/11/23

Signature: Sumit

Abstract -: A training schedule management system is a software application that helps to plan, organize, and monitor the training activities of employees or students. It can also provide feedback, evaluation, and reporting on the training outcomes and performance. A training schedule management system can have the following features:

- Admin will allow to create and modify training schedules according to the availability of trainers, trainees, and resources.
- It can automate the allocation of trainers, trainees, and resources to avoid conflicts and optimize the utilization of resources.
- It can generate reports and analytics on the training effectiveness, efficiency, and satisfaction.
- Student will able to register themself.They also check their schedule.
- Faculty also able to check their respective schedule.

Training is an essential activity for any organization that aims to improve the skills, knowledge, and performance of its students. However, managing the training process can be challenging and time-consuming, especially when there are multiple training programs, trainers, trainees, and resources involved. A training schedule management system is a software solution that simplifies and streamlines the training process by providing tools for planning, organizing, monitoring, and evaluating the training activities. This paper presents the design and implementation of a training schedule management system that can handle various types of training programs, such as online, offline, synchronous, asynchronous, individual, or group.The system can help the users to create and modify training schedules, allocate trainers, trainees, and resources, track the training progress and completion, generate reports and analytics, and facilitate communication and collaboration. The system can improve the training quality, efficiency, and satisfaction for both the trainers and the trainees

Overview-: A training schedule management system serves as an essential tool for educational institutions to streamline the organization and administration of training schedules. This software solution is designed to meet the varied needs of both administrators and faculty members, providing a user-friendly interface to efficiently create, update and manage training schedules.

Table of Contents

Chapter 1: **Introduction**

Training is an essential activity for any organization that aims to improve the skills, knowledge, and performance of its employees or students. However, managing the training process can be challenging and time-consuming, especially when there are multiple training programs, trainers, trainees, and resources involved. Therefore, there is a need for a software solution that can simplify and streamline the training process by providing tools for planning, organizing, monitoring, and evaluating the training activities. This paper presents the design and implementation of a training schedule management system, which is a software application that can handle various types of training programs, such as online, offline, synchronous, asynchronous, individual, or group. The system can help the users to create and modify training schedules, allocate trainers, trainees, and resources, track the training progress and completion, generate reports and analytics, and facilitate communication and collaboration. The system can improve the training quality, efficiency, and satisfaction for both the trainers and the trainees.

a. Login access and Features:

The following are the login supported in the application.

1. **Teacher:** The Teacher's username and password gets added to the system by admin. Following are the features: -

    - View your information
    - Request for change the schedule

2. **Student:** The Student gets enrolled by the admin and username, password gets added to the system by admin. Following are the features of a student: -

- View your information
- View Faculty
- View Schedule

3. **Admin:** The administrator gets a fixed username and password predefined in the code. Following are features:-

- Selection, Insertion, Deletion of Student Data
- Display, Insertion, Deletion of Faculty Data
- Selection, Insertion, Deletion of Faculty Data

Chapter 2: **Problem Definition & Objectives**

a) Problem

Educational institutions often face challenges in effectively coordinating and managing training schedules. These include issues such as manual schedule creation, difficulties in updating schedules and the need for a centralized system for allocating trainers.

Complex Enrollment Processes : The current enrollment procedures in educational institutions are often cumbersome and time-consuming, leading to difficulties for both students and instructors.

Limited time : Often times due to time, training modules get cancelled or compressed. Due to this the participants are not able to grasp all the essential knowledge and experience.

Administrative Overhead : Both students and instructors deal with unnecessary administrative tasks, taking time and effort away from the core learning and teaching activities.

Large or geographically dispersed training audience : When the participants are spread across far distances or are busy in their work routines, organizing training session becomes tedious. To make arrangements for everyone and then bringing every one together requires great effort. Training needs to be customized on the basis of the target audience.

Lack of training budget : Sometimes companies feel that training sessions are costly affairs due to which less budget is allocated for training related activities.

b) Objectives

Provide a user-friendly platform for administrators to create and update training schedules.
Allow faculty members to view and potentially cancel their assigned schedules.

Implement a secure authentication mechanism for both administrators and faculty members.
Streamline the allocation of trainers to specific techniques.

Improved productivity: When employees are well-trained, they are more likely to be productive.

Reduced errors: Well-trained employees are less likely to make mistakes, even under pressure.

Enhanced student well-being: Training can help students to cope with stress, develop healthy habits, and prevent burnout.

Time and Cost Savings: Minimize the time and financial investments required for administrative tasks, benefiting both students and instructors and eliminating the need for expensive traditional methods of course presentation.

Improved Learning Experience: Ultimately, the project aims to enhance the overall online learning experience, creating a user-friendly and efficient platform for students and instructors to engage with education in the digital age.

Chapter 3: **Proposed Work/Methodology**

a - Problem

1.Educational institutions often face challenges in effectively coordinating and managing training schedules. These include issues such as manual schedule creation, difficulties in updating schedules and the need for a centralized system for allocating trainers.

2.**Limited Course Accessibility:** Students often struggle to discover and enroll in courses that match their interests and career goals, limiting their educational options.

3.**Administrative Overhead:** Both students and instructors deal with unnecessary administrative tasks, taking time and effort away from the core learning and teaching activities.

4.**Inadequate Cost Savings:** Traditional methods of course management can be expensive for instructors, considering costs related to in-person seminars, travel, and materials.

5. Lack of clear and realistic goals, deadlines, and plans for the training activity.

6.Difficulty in measuring and evaluating the impact and outcomes of the training activities. This can hinder the ability to assess the return on investment, identify the gaps WEand strengths, and make informed decisions for future training.

b - Objectives

1.Provide a user-friendly platform for administrators to create and update training schedules.

2.Allow faculty members to view and potentially cancel their assigned schedules.

3.Implement a secure authentication mechanism for both administrators and faculty members.

4.Streamline the allocation of trainers to specific techniques

5.To improve the efficiency and effectiveness of training delivery by automating tasks such as scheduling, registration, communication, and reporting.
6.To optimize the use of resources such as trainers, venues, equipment, and budget by reducing conflicts, cancellations, and wastage.

7.To enhance the quality and consistency of training by ensuring that the learning objectives, content, and methods are aligned with the organizational standards and best practices.

8.To increase the engagement and satisfaction of learners and trainers by providing them with flexible, personalized, and interactive learning experiences.

9.To measure and evaluate the impact and outcomes of training by collecting and analyzing data on learner performance, feedback, and retention.

Chapter 4: **Data Structure Used**

Queue: The queue data structure played a pivotal role in managing the assignment submission process. A queue is a linear data structure that follows the First-In-First-Out (FIFO) principle. This means that the first item added to the queue is the first one to be processed. In the context of Removal of schedule, as faculty request for remove their schedule, they were placed in a queue, ensuring a fair and organized system for admin to ensure which request to be accepted.

The queue ensured that assignments were reviewed and graded in the order they were received, eliminating any potential bias or confusion. This feature enhanced transparency and fairness in the educational process. Instructors could efficiently access and manage the a, knowing that they were following a systematic order, which is especially important in scenarios where multiple students submit their work simultaneously.

Overall, the use of a queue data structure streamlined the assignment submission process and enhanced the educational experience for both students and instructors, ensuring that assignments were handled in a fair and orderly manner.

Stack: The stack data structure was utilized to enhance the user experience, primarily in the area of navigation and page history. A stack is a linear data structure that follows the Last-In-First-Out (LIFO) principle, meaning that the last item added is the first one to be removed.

The stack was employed to maintain a history of recently visited pages, enabling users to navigate within the system with ease. For instance, when a user clicked on a course or a specific content page, it was added to the stack. This allowed users to move back through their navigation history efficiently by using features like the back button.

The stack's LIFO principle was particularly beneficial for keeping track of user interactions, ensuring that the most recent actions were readily accessible. This enhanced the overall user experience by simplifying navigation and helping users quickly return to previous screens or content, promoting a more intuitive and user-friendly interface.

By incorporating a stack data structure, the TMS project improved user navigation and facilitated a seamless online learning experience, making it easier for students and instructors to access the content and features they needed.

Linked List: A linked list is a data structure that can be used in the context of a training schedule management system to manage and arrange the numerous training sessions, events, and related information. Every node in a linked list has data and a reference, or link, to the node after it in the list's sequence. The schedule may be managed and updated dynamically with efficiency and flexibility thanks to this framework.

Important Elements of a Linked List for Training Schedule Administration:

Node

A node in the linked list acts as a container for data pertaining to a particular training session. Usually, this information consists of specifics like the date, time, place, designated trainer, and participant list.

Link/Reference:

Each node in the linked list contains a reference to the next node in the sequence. This linkage facilitates the traversal of the list from one node to the next, allowing for efficient navigation through the schedule

Benefits of Linked List Utilisation:

1. Dynamic Structure:

Schedule modifications are easily accommodated via linked lists. Because it doesn't require allocating or deallocating memory for the entire structure, sessions can be added or withdrawn to accommodate a dynamic and changing training schedule.

2. Effective Insertion and Removal:

The linkages between nodes must be updated in order to add or remove a training session, although this can be done more quickly than with other data structures like arrays. This is especially helpful for handling frequent scheduling adjustments.

3. Flexible Memory Allocation:

Memory is allotted to nodes in a linked list only when necessary, enabling effective use of available memory. In contrast, memory needs to be pre-allocated for fixed-size arrays.

4. Ease of Traversal:

Traversing the linked list enables easy display of the entire training schedule or searching for specific sessions. This is essential for users and administrators to quickly access relevant information.

Binary Tree-:

Decision trees: Decision trees can be used to represent the structure of a training program. Each node in the tree represents a decision point, and the branches represent the possible paths that the program can take. This can be used to create a personalized training plan for each student.

Schedule management: Binary trees can be used to represent a training schedule. Each node in the tree represents a training session, and the branches represent the order in which the sessions should be taken. This can be used to ensure that all training sessions are completed in the correct order.

Performance tracking: Binary trees can be used to track the performance of students in a training program. Each node in the tree represents a student, and the branches represent their performance on different training modules. This can be used to identify students who are struggling and provide them with additional support.

Chapter 5: **Language and Tools**

1. C++ LANGUAGE FEATURES:

Object-Oriented Programming (OOP): Utilize OOP principles to model entities like Training Sessions, Trainers, and Participants as classes. Encapsulation, inheritance, and polymorphism can help organize and structure your code.

Standard Template Library (STL): Leverage STL containers like std::vector or std::list for managing collections of objects. This can be useful for storing and manipulating training schedules

File I/O Operations: Use C++ file I/O operations to read from and write to files for storing persistent data, such as the training schedule information.

2. HEADER FILES

Basically this project contains some essentials header files that are very useful while dealing with this project. Below are the details of each and every header files which is used in this project.

PRE DEFINED HEADER FILES

- **#include <iostream>**

iostream stands for standard input-output stream. #include iostream declares objects that control reading from and writing to the standard streams. In other words, the iostream library is an object-oriented library that provides input and output functionality using streams.

- **#include <string.h>**

The library string.h (also referred as cstring) has several common functions for dealing with strings stored in arrays of characters. The string.h header file to be included before using any string function.

- **#include <stdlib.h>**

stdlib.h is a header file and also the Standard Library of C programming language that declares various utility functions for type conversions, memory allocation, algorithms, and other similar use cases.

- **#include <fstream>**

fstream is a data type that represents the file stream generally, and has the capabilities of both ofstream and ifstream which means it can create files, write information to files, and read information from files.

USER DEFINED HARDER FILES

- **#include “adminLogin.h”**

This is basically a user defined header file which includes all the functionalities of admin in the main file so, that the program can work efficiently. This header file will give access to all the features which an admin has.

- **#include “facultyLogin.h”**

This is basically a user defined header file which includes all the functionalities of a faculty in the main file so, that the program can work efficiently. This header file will give access to all the features which a faculty has.

2. USING NAMESPACE STD

A namespace is designed to overcome this difficulty and is used as additional information to differentiate similar functions, variables etc. with the same name available in different libraries.

A std is an abbreviation of standard. std is the standard namespace. cout, cin and a lot of other things are defined in it. you can also call these functions using std::cout, std::cin etc.

The keyword using technically means, use this whenever you can. This refers, in this case, to the std namespace. So whenever the computer comes across cout, cin, end or anything of that matter, it will read it as std:cout, std::cin or std::endl.

Chapter 6: **Source code**

Source code-:

Chapter 7:Results

1. CLASSES

- **class admin**

This is a admin class which has some functionalities and modules with respect to tasks performed by an admin which means,

The Administrative should be able to do the following operations once he has logged in with his unique id and password.

1. Create a new training schedule
2. Update or make changes to the existing schedules
3. Allocates a schedule to a specific trainer
4. Generate a report of the trainings happening on monthly basis for a specific stream
5. Quit Application

- **class faculty**

This is a faculty class which has some functionalities and modules with respect to tasks performed by a faculty which means,

The faculty should be able to do the following operations once he has logged in with his unique user id and password:

1. Find out the schedule assigned to him
2. Cancel a training schedule allocated to him
3. Quit Application

4. MAIN()

The main function i.e main() serves as the entry point of the program execution, here the program has access to command line arguments.

In main it usually controls the program execution by directing the calls to other functions in the program.

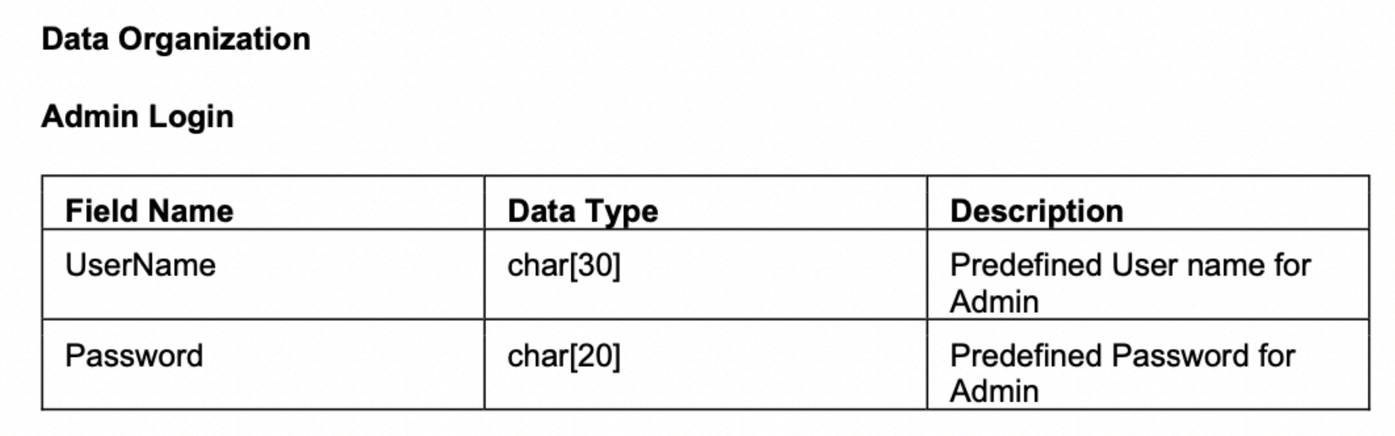
Here in this project we are calling the Admin/Faculty login functions and it’s functionality via calling Login function directly and using switch case to use user functionalities after login.

5. ADMIN FUNCTIONS

- **adminLogin()**

This function is used to authenticate the admin and further give him access to all his operations.

Here, the username and password of an admin is already predefined as per the requirement. So, if an admin enters the correct Id and password then only he can able to login to the system and get access to his operations.



Data Organization

Admin Login

Field Name	Data Type	Description
UserName	char[30]	Predefined User name for Admin
Password	char[20]	Predefined Password for Admin

Note : If an admin enters wrong id and password more than 3 times then the system will get automatically lock and further an admin would not be able to login to the system again.

- **adminModules()**

This function basically gives access to an admin to perform a particular tasks as per his choice.

This function mainly contains five major operations that can be performed by an admin. Below are the details of each operations.

1. Create New Schedule

2. Update Existing Schedule

3. Allot A Trainer To Specific Stream

4. Generate Report On Monthly Basis

5. Quit From The Application

In this function an admin choose a option from above given option and further he will be able to navigate through the new windows with respect to the option chosen by him.

The last operation i,e, quit from the application, basically gives an option to an admin whether he want to quit from the entire application or he just want to navigate to his dashboard.

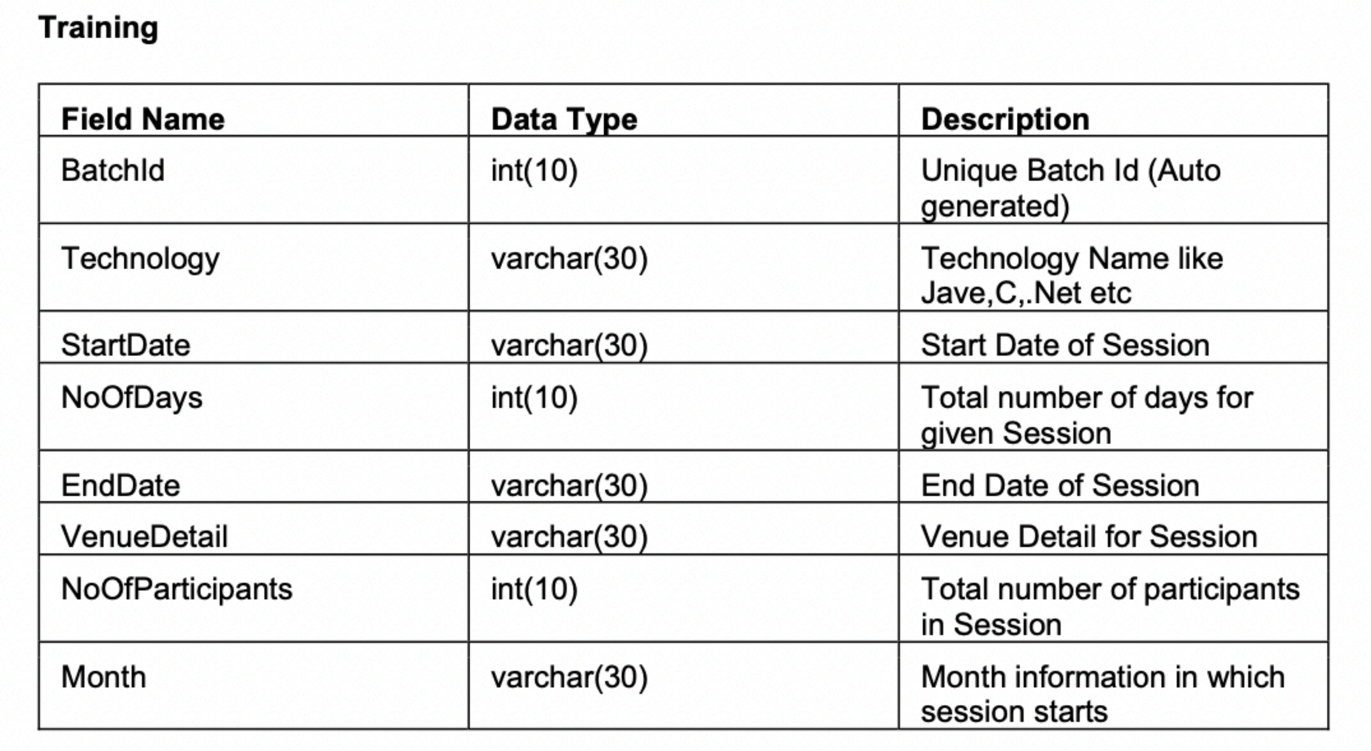
If admin wants to go to on his dashboard then he can simply navigate through it while performing particular operation else he can directly logout from his window.

- **createSchedule()**

As part of this operation, the admin should have the ability to enter all the details of a new training schedule.

Schedule details include:

1. Technology
2. Batch id
3. Start and end date of the batch
4. Faculty details
5. Venue details
6. Number of participants in the batch.



Training

Field Name	Data Type	Description
BatchId	int(10)	Unique Batch Id (Auto generated)
Technology	varchar(30)	Technology Name like Jave,C,.Net etc
StartDate	varchar(30)	Start Date of Session
NoOfDays	int(10)	Total number of days for given Session
EndDate	varchar(30)	End Date of Session
VenueDetail	varchar(30)	Venue Detail for Session
NoOfParticipants	int(10)	Total number of participants in Session
Month	varchar(30)	Month information in which session starts

This function give access to an admin for creating a new schedule which consists some mandatory details which needs to be filled during creation of new schedule.

After entering all the mandatory fields by an admin, the information will get automatically save in system by using a csv(comma separated value) file which an admin can access after creating new schedule.

Lastly admin will get two option whether he want to continue with creating new schedule or he want to go back to his dashboard.

- **updateSchedule()**

As part of this operation, the admin user should have the ability to modify details associated with a training schedule. This will be done by entering the Batch ID that is associated with the schedule which is needed to be updated/ modify.

Other than batch id all the other sections can be modified and if there is no need of updating any particular field then, it can be entered the same again.

Multiple schedule can be modified in a single time operation. If batch id matches then it will work else not.

- **allotTrainer()**

As part of this operation Admin should be able to allocate the specific training schedule to the particular trainer. It will be done by entering the name of trainer and the technology name.

- **generateReport()**

As part of this operation, the admin user should have the ability to generate report of trainings happening on the monthly basis for the specific stream.

This will be done by entering the name of month for which reports needs to be generated along with technology name. If matches display the training report to the admin.

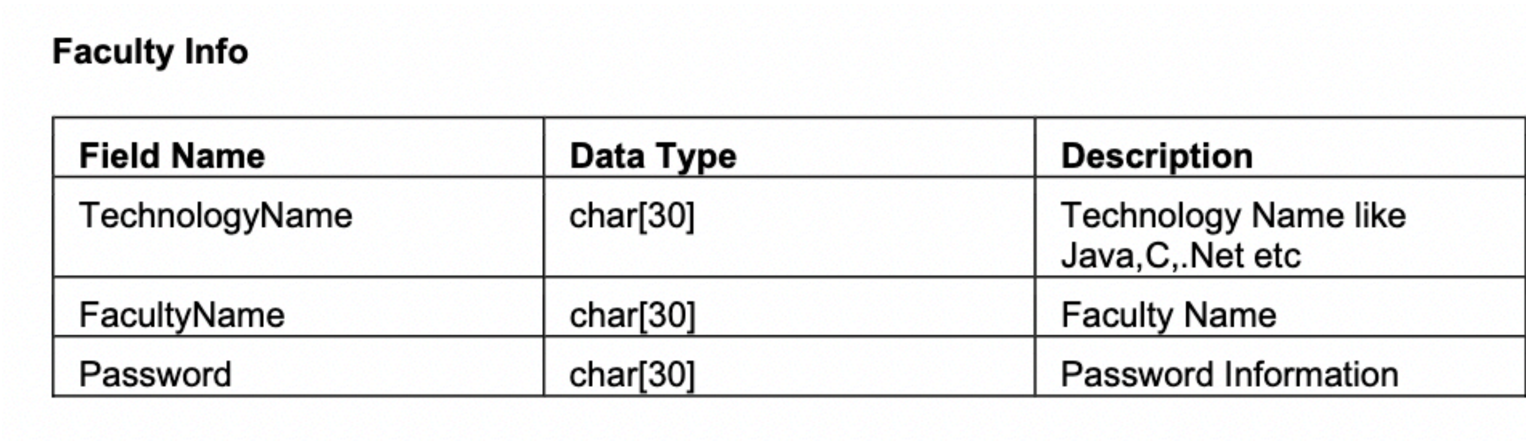
6. FACULTY FUNCTION

- **facultyLogin()**

This function is used to authenticate the faculty and further give him access to all his operations.

Here, the username and password of a faculty are stored in facultyLogin.csv file. So, if faculty enters the correct Id and password and technology name then only he can able to login to the system and get access to his operations.

Note : If a Faculty enters wrong id and password more than 3 times then, the system will get automatically lock and further a faculty would not be able to login to the system again and have to restart the system.



Faculty Info

Field Name	**Data Type**	**Description**
TechnologyName	char[30]	Technology Name like Java,C,.Net etc
FacultyName	char[30]	Faculty Name
Password	char[30]	Password Information

- **facultyModules()**

This function basically gives access to a faculty to perform a particular tasks as per his choice.

This function mainly contains 2 operations that can be performed by an admin. Below are the details of each operations.

1. View schedule : He can view schedule assigned to him.

2. Request to cancel the schedule :- He can request to cancel the assigned schedule to him by admin.

- **StudentModules()**

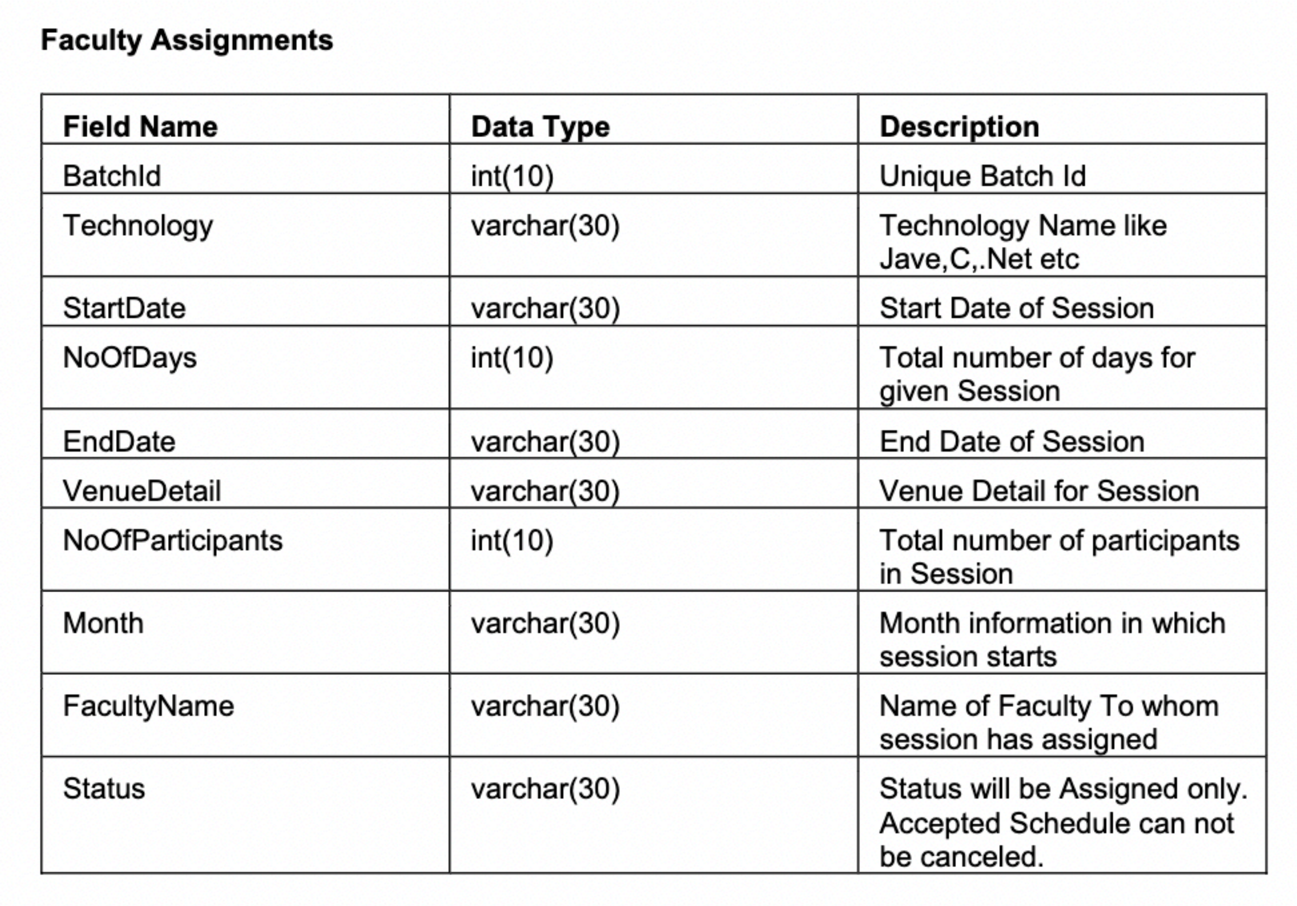
This function basically gives access to a student to view a particular schedule as per his choice.

This function mainly contains 1 operations that can be performed by an admin. Below are the details of each operations.

1. View schedule : He can view schedule assigned to him.

- **viewSchedule()**

Through this function the faculty is able to view the schedule assigned to him by the admin. And here he can accept the assigned schedule and then the status will be yes. After accepting the assigned schedule he can't request to cancel the schedule.



Faculty Assignments

Field Name	Data Type	Description
BatchId	int(10)	Unique Batch Id
Technology	varchar(30)	Technology Name like Jave,C,.Net etc
StartDate	varchar(30)	Start Date of Session
NoOfDays	int(10)	Total number of days for given Session
EndDate	varchar(30)	End Date of Session
VenueDetail	varchar(30)	Venue Detail for Session
NoOfParticipants	int(10)	Total number of participants in Session
Month	varchar(30)	Month information in which session starts
FacultyName	varchar(30)	Name of Faculty To whom session has assigned
Status	varchar(30)	Status will be Assigned only. Accepted Schedule can not be canceled.

- **cancelSchedule()**

Through this function the faculty can request to cancel his schedule.

***Note** : Admin/Faculty can quit the program at the end or in between if they don’t want to continue with the session.

7. CSV FILES

Here the program is accessing three CSV files to store or fetch the data for different functionalities of Admin or Faculty.

These are following CSV files :

1. **schedule.csv** : Storing/Update the value of data entered for creating the schedule by Admin.

2. **trainerallotment.csv** : Storing/Update the whole details of schedule along with trainer name assigned to the the specific technology.

3. **facultyLogin.csv** : Storing the data needed for faculty login into the system.

4. **studentLogin.csv** : Storing the data needed for student login into the system.

Chapter 8: **Conclusion**

A training schedule management system serves as an essential tool for educational institutions to streamline the organization and administration of training schedules. This software solution is designed to meet the varied needs of both administrators and faculty members, providing a user-friendly interface to efficiently create, update and manage training schedules.Training schedule management system emphasizes the system's potential to enhance efficiency, organization, and effectiveness in managing training programs. It highlights the importance of streamlining processes, reducing administrative burden, and providing a user-friendly interface for both administrators and participants. Additionally, the conclusion may touch upon the system's ability to generate insightful reports, track progress, and adapt to evolving training needs. Overall, the conclusion underscores the positive impact of the training schedule management system on optimizing resource allocation, improving communication, and ultimately fostering a more productive and structured training environment.

There are several advantages to using a training schedule management system, including:

1. Reducing manual work and human errors by automating tasks such as course creation, scheduling, registration, communication, and reporting.

2. Increasing production and efficiency by making the best use of available resources, like venues and trainers.
delivering training in a variety of ways, such as in-person, online, or blended learning, to improve its efficacy and quality.

3. Offering insights and data on training performance and outcome to support corporate growth and innovation

Application Development:

• Objective: Transition from a terminal-based system to a more user-friendly and
accessible application.

Benefits:

• Enhance user experience through a visually intuitive interface.
• Facilitate easy access to features for administrators, teachers, and students.
• Enable broader device compatibility, including mobile devices, for on-the-go access.

Database Connection and Development:

Objective: Migrate from the current file-based storage (txt and csv) to a robust
relational database management system.

Benefits:
• Enhance data integrity, security, and scalability.
• Enable real-time data retrieval and updates for improved system responsiveness.
• Facilitate complex queries and reporting functionalities for administrators.

Enhanced User Authentication:

Objective: Implement more secure and versatile authentication mechanisms.

Benefits:

• Strengthen system security against unauthorized access.
• Explore options such as multi-factor authentication for enhanced user protection.

Scalability and Performance Optimization:

Objective: Design the system architecture to accommodate future growth and
increasing user loads.

Benefits:
• Ensure consistent performance as the user base expands.
• Optimize system efficiency to handle a larger volume of concurrent users.

Chapter 9: Bibliography

C++ by John Carey , Shreyans Doshi

https://www.geeksforgeeks.org/header-files-in-c-cpp-and-its-uses/

https://www.geeksforgeeks.org/what-is-linked-list/?ref=roadmap

https://youtube.com/watch?v=lZRaICFZgks

https://www.geeksforgeeks.org/data-structures/linked-list/

https://www.coursera.org/programs/b-tech-computer-science-engineering-2nd-year-49goh/learn/data-structures?specialization=data-structures-algorithms

https://www.codingninjas.com/courses/c-plus-plus-data-structures-and-algorithms?utm_source=google&utm_medium=%5Bsearch%5D&utm_campaign=17201965807_144941638924_p_c%2B%2B%20course__633848361742_c____9301378&gad_source=1&gclid=Cj0KCQiAmNeqBhD4ARIsADsYfTdsxbV4e_DSwZR0ZOpKP4JiINMGh04cbiLjI9LzuJKJQlWHwGN59lYaAq_rEALw_wcB