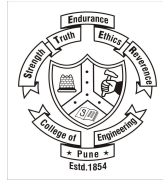


COLLEGE OF ENGINEERING,PUNE



(Electrical Engineering Department)

A Project Report ON

DEADSTOCK REGISTER (MACHINES LABORATORY)

For the Course

Data Structures and Computer Programming Lab

Name: Pragati Narote

MIS: 111905029

Table of Content

Sr. No	Title	Page No.
1.	Introduction	3
2.	Features of the Application	4
3.	Program Workflow	5
4.	User Login	6
5.	Add to Data	11
6.	User Input Validations	13
7.	Modify Data	15
8.	Display Data	17
9.	Search through Data	24
10.	Logout	26
11.	Storing the Data	27
12.	Future Scope	28
13.	Conclusion	28
14	Acknowledgement	29

1. Introduction

General Idea of Project:

In this project, we have built an application that will help in efficient and systematic organisation of Machine Laboratory Data. This project will provide a better way to manage the details of equipments present in the laboratory. The classical way of managing the data is by creating a Deadstock register in Microsoft Excel. In this application, we will provide user with features like adding data to the deadstock, modifying the data, displaying data and statistics, searching through the data; along with auto generation of unique id for each row in the deadstock and also validating the data enter by the user, and storing the data in a CSV (comma-separated values) file. The application will also provide different types of user login; thus, providing some important features only for admin login.

Language Used:

The entire application will be built in C++ programming language. C++ is a general-purpose, cross platform programming language that can be used in building high-performance applications. Till date a large number of games, desktop applications, operating systems, etc are developed in C++.

Moreover, C++ provides a broad variety of libraries(header files) that can be used to perform some general task in a simple manner and efficiently. Thus, C++ is a language of choose for many programming since ages.

Major concepts used:

- A single row of data is stored and managed using the *struct* datatype declaration.
- The entire data i.e., number of rows are stored and managed used *array* data structure.
- To make the program easy to understand and to reduced repeating of code, we have used *functions* that perform small tasks.
- Later, the control flow of entire application is managed by using *switch* statements.
- The data is stored in a csv file. To implement this, we use *fstream* header file.
- Other general control flow statements and loops are also used to perform certain tasks.
- The major datatype used is *string*. String is a collection of characters. Strings are in general object to string class.
- Moreover, *global variables* are used to retain and maintain the size of database, and also store the username and passwords for login.

In simple words, this application is built using simple C++ programming concepts.

2. Features of the application

The different features provided by this application are as follows:

1. User Login
 - a. Admin Login
 - b. Lab Assistant Login
 - c. Student Login
2. Add to Data
3. Modify the Data
4. Display the Data
 - a. Display entire data
 - b. Display only equipments that are presently available in laboratory
 - c. Display only equipments that are presently not-available in laboratory
 - d. Display the data sorted according to date of purchase
 - e. Display the equipments purchased over a period of time and also display the expenditure statistics
 - f. Display the expenditure statistics for the entire data available
5. Search through the Data
 - a. Search and display the equipment by the ID entered by the user
 - b. Search through the data and display all the equipments that have keyword entered by the user in the description
6. Logout
7. Storing Data in CSV file

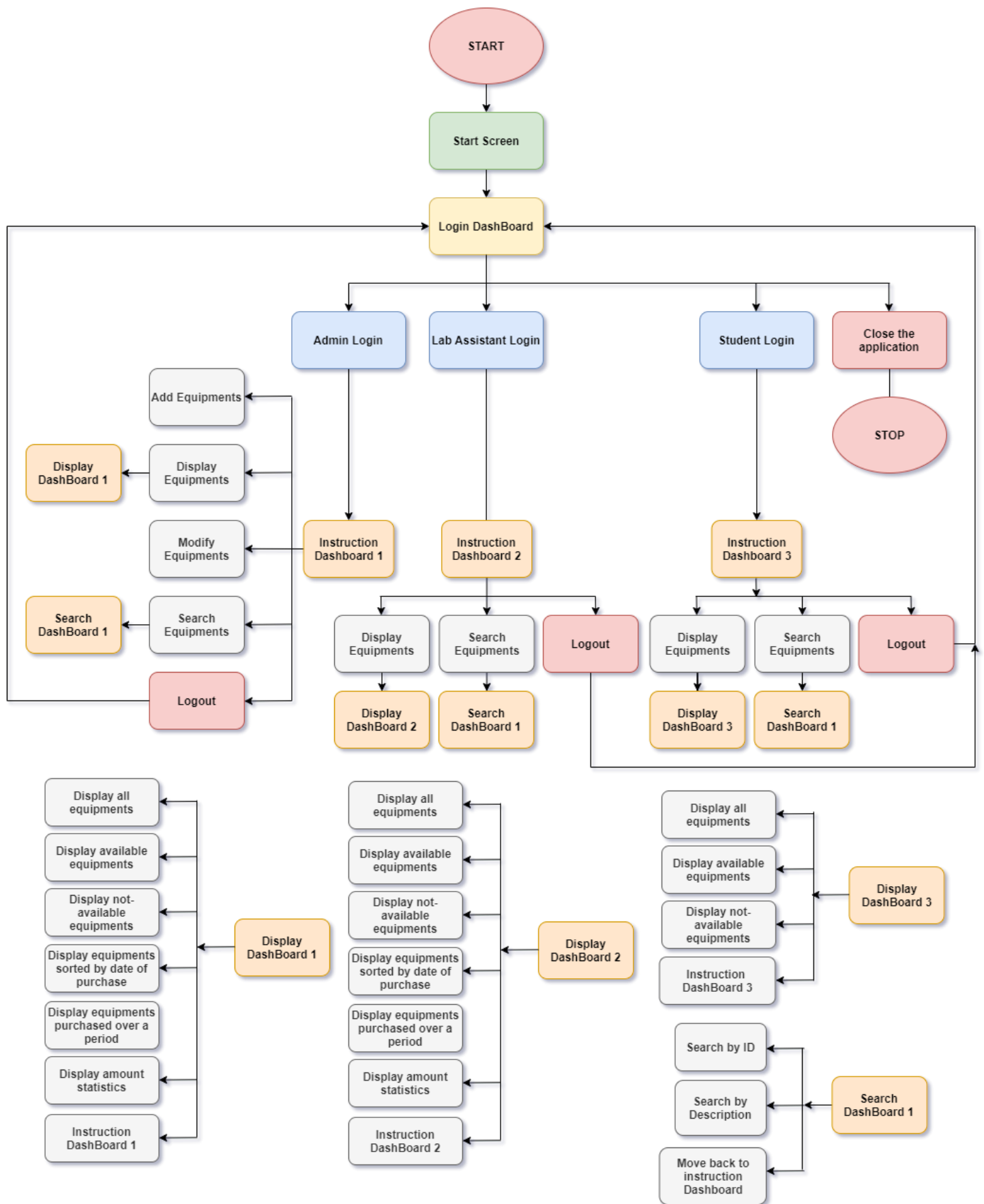
Fields provided for every equipment

1. ID (unique id- this is autogenerated)
2. Description
3. Date of purchase
4. Quantity
5. Amount
6. Grant
7. Purchaser
8. Purchase details
9. Status
 - a. Available
 - b. Not Available
10. Sign of lab coordinator
11. Remarks

Structures used in this Project

1. struct Equipment{};
Member variables
 - string _id;
 - string description;
 - string dop;
 - int quantity;
 - float amount;
 - string grant;
 - string purchaser;
 - string pur_details;
 - bool status;
 - string sign;
 - string remark;
 - int date;
2. struct All_Equipments{};
Member variables:
 - Equipment eqps[1000]

3. Program Workflow



4. User Login

Before discussing the user login features, let's see the flash screen of the application.

Start Screen (Flash Screen)

Below is the flash screen of the application. It simply consists of Project name, Lab name, College name and group member's names. It is result of simple combination of output statements.

```
*****
DEADSTOCK  REGISTER
MACHINE    LAB
*****
COLLEGE OF ENGINEERING, PUNE
*****

Project by DSCP-Group-4
=====
Pragati      111905029
Anand        111910018
Bhakti       111905064
Sanket       111905075
Sakshi       142005003
Sachin       111905039
Roshankumar  111905067
Jayashree    142005008
Jagpreet     111905066
```

User Login

The user login consists of three different types of logins.

1. *Admin login* – The username and password for the admin login will be provide by us. The admin has access to special features like add to data and modify data.
2. *Lab Assistant login* – The username and password for lab assistant login will also be provide by us. The lab assistant has access to all the features except add to data and modify data.
3. *Student login* – The student login is a no username or password login. It is just a simple feature which would help the student to see whether the equipments of their interest is presently available in the lab or not.

W E L C O M E T O M A C H I N E S L A B

Types of Logins available:

A: Login as Admin.
L: Login as Lab Assistant.
S: Login as Student.
E: Close the Application

CHOOSE THE OPERATION THAT YOU WANT TO PERFORM :

(Login Dashboard)

The user can choose between A, L, S, E, a, l, s, e according to their requirement.

Student Login:

If the user chooses S or s; then the user is taken to an instruction dashboard as shown below.

L O G G E D I N A S S T U D E N T

D A S H B O A R D

KEYS FOR THE OPERATIONS:

D: Display Dashboard.
S: Search Dashboard.
E: Logout.

CHOOSE THE OPERATION THAT YOU WANT TO PERFORM :

(If pressed S or s :- Student is directly logged in and instruction dashboard for student login is displayed)

Admin and Lab Assistant Login

Salient Features of Admin and Lab Assistant Login

- The user is asked to enter username and password in admin and lab assistant login
- The username and password are provided by us
- The user gets 3 tries to enter the username and password correctly; if he/she fails 3 times then the program displays an error and moves to the Login Dashboard again; if the user enters the details correctly then the user is logged in.

W E L C O M E T O M A C H I N E S L A B

A D M I N L O G I N

Please enter the user name and password

User Name : admin2

Password : ****_

(Admin login screen)

W E L C O M E T O M A C H I N E S L A B

A D M I N L O G I N

OOPS - Incorrect username or password!!! Try again!!!
Please enter the user name and password

User Name : admin3

Password : ***

(Admin login screen is the 3 tries are not over, but for pervious try username or password were incorrect)

W E L C O M E T O M A C H I N E S L A B

A D M I N L O G I N

OOPS - Incorrect username or password!!! Try again!!!
Please enter the user name and password

User Name : sfsg

Password : ****

**** OOPs!!! You failed logging in 3 times!!! Terminating program!!! ****

(Admin login screen after 3 failed attempts to login in)

W E L C O M E T O M A C H I N E S L A B

A D M I N L O G I N

Please enter the user name and password

User Name : admin

Password : *****

Logging in as admin....

(Admin login screen if login details are correct)

The screens and features for Lab Assistant login are same as that of Admin login.

Functions written to implement logins:

1. char login_instructions()

This function displays the Login Dashboard and takes an input of character(char) from the user. It returns a value of char datatype.

2. void login()

For Admin Login:

This function takes the username and password as input from the user. If the entered username and password are verified with the globally declared ones, then the user is logged in as admin.

3. void login_assistant()

For Lab Assistant Login:

This function takes the username and password as input from the user. If the entered username and password are verified with the globally declared ones, then the user is logged in as lab assistant.

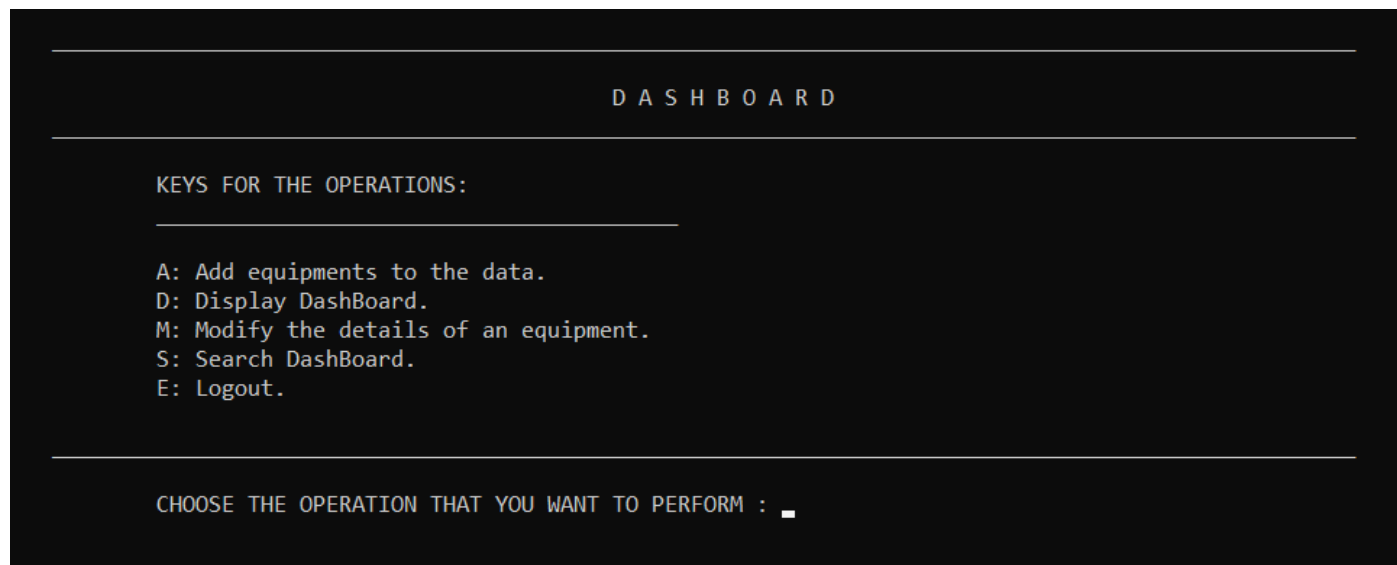
REQUIREMENTS FOR DIFFERENT LOGINS		
Login Type \ Requirements	Username	Password
1. Admin Login	✓	✓
2. Lab Assistant Login	✓	✓
3. Student Login	✗	✗

FEATURES IN DIFFERENT LOGINS

Feature \ Types of Logins	Admin Login	Lab Assistant Login	Student Login
1. Add Data	✓	✗	✗
2. Modify Data	✓	✗	✗
3. Display Data			
a. Whole Data	✓	✓	✓
b. Available Equipments	✓	✓	✓
c. Not Available Equipments	✓	✓	✓
d. Sorted Equipments (by date of purchase)	✓	✓	✗
e. Equipments purchased over a period	✓	✓	✗
f. Amount Statistics	✓	✓	✗
4. Search through Data			
a. Search by ID	✓	✓	✓
b. Search by Description	✓	✓	✓
5. Logout	✓	✓	✓

5. Add to data

Imp: This feature is only available for users logged in through ADMIN login.



(This is the Admin Main Instructions Dashboard)

If A or a is entered then the admin is taken to the add equipments screen.

Salient Features of Add Data Feature:

1. This feature is available only in Admin login.
2. Initially, the user is asked to enter the number(n) of equipments he/she wants to add to the database.
3. Then the user is asked to enter the equipment details for that number(n) of equipments.
4. When the user has entered the details for the n equipments and presses Enter; then the database.

Field types provided for entering the data:

1. *ID* – This is autogenerated. User won't be able to enter it.
2. *Description, Date of Purchase, Quantity, Status* – These fields are required fields. User must enter these details.
3. *Amount, Grant, Purchaser, Purchase details, Sign of Co-ordinator, Remarks* – These fields are optional. User can leave these fields empty.

ADD DATA

Enter the number of equipments you want to add: 2

```

                Id :      ML001
      Description* :      Three phase inductive loading bank
Date of purchase(DD/MM/YYYY)* : 26/06/2017
      Quantity*   :      1
      Amount      :      1200
      Grant       :
      Purchaser   :
Purchase details  :      Aashivi electricals
Status (1: Available; 0: Not Available)* : 1
      Sign of lab coordinator :
      Remarks     :

```

```

                Id :      ML002
      Description* :      Slip ring induction motor
Date of purchase(DD/MM/YYYY)* : 30/10/2003
      Quantity*   :      4
      Amount      :      7500
      Grant       :
      Purchaser   :
Purchase details  :      Super scientific industries
Status (1: Available; 0: Not Available)* : 0
      Sign of lab coordinator :
      Remarks     :

```

Adding the data...■

(Add data screen after entering details of 2 equipments)

Functions written to implement add data feature

1. void read_details(Equipment &E, All_Equipments &All):

This function prompts the user to enter the data for every field and stores the inputted values in the struct E. This function also calls all the functions that created to validate the input given by user, and validates the inputs before added it to struct E.

2. void add2array(All_Equipments &All):

This function calls the read_details function, on desired struct E. Here, struct All is an array of struct E.

3. void add_n_equipments(All_Equipments &All):

This function takes an integer as input 'n' and then calls the add2array function 'n' times. Thus adding 'n' equipments to the All[] array.

Thus, these 3 functions together work at adding equipments to the database.

6. User Input Validations

What is input validation?

Data Validation/ Input Validation is a process of properly testing the inputs received by the application from the user. Input Validation is a very vital step in any applications workflow.

Why validate the user inputs?

Inputs must be validated before doing any kind of operations or processing on it. This is extremely vital because if a wrong user input is not handled properly then it might result in crashing the entire program/applications. Some basis validations include – checking the data type, data size, data range, data content, etc of the user inputs.

Following are the important and necessary validations implemented in our application:

1. Autogenerate ID

The ID of every equipment is autogenerated. This autogenerated ID is unique for every equipment.

Function: `string generate_id();`

This function using a global variable – size; and generate a string of the format ML***, where *** is a three digit number and *** = size. The variable size stores the number of equipments already present in the database. This function returns a string – ML***

2. Validation of Required field

While taking inputs for a form from the user, some fields must be required. The data entered without these fields is meaningless or useless sometimes. Fields like description, date of purchase, quantity and status of equipment are required field in our application.

Function: `string validate_required(string s);`

This functions takes a string s as a parameter and check if this string consists of at least 1 character other than space. If yes then it returns s; else it will recursively prompt the user that the data is required, until the user enters a proper string, and then returns this string. This function returns a string.

3. Validation for Date

In any real world application, it is important to maintain a standard date format in order to avoid confusion and malfunctioning. In this application we used DD/MM/YYYY date format.

Function: `string validate_date(string date)`

This function takes a string date as a parameter and checks if it's format it same as DD/MM/YYYY, where MM is an integer greater than 1 and less than 13; and DD is also an integer. The value of DD depends on the value of MM and YYYY is simply any integer value.

e.g.: 29/02/2001 – invalidate date

29/02/2000 – validate date

12/16/2021 – invalidate date

12-06-2021 – invalidate date (we only use / in dates)

12 Aug 2019 – invalidate date

4. Validation for Quantity

The quantity of any object is an integer value; and more precisely it is a non-negative integer value.

Function: `int validate_non_neg(string q);`

This function takes a string `q` as a parameter and checks if this string only consists of digits i.e., it should not have a single alphabet, or special symbols in it. If the string `q` consists of only digits then it converts `q` into an integer `num` and returns `num`; else it will recursively ask user to enter a positive integer.

5. Validation for Amount

The amount of money is always a non-negative float value with utmost 2 decimal points.

Function: `float validate_amount(string q);`

This function takes a string `q` as a parameter and checks if this string consists of only digits and a single period. If yes then it converts `q` into float `num` and then checks if it has at maximum 2 digits after the period(decimal point); if yes then it returns `num`(float value); otherwise, it will recursively ask the user to give valid input.

6. Validation for Status

The status of an equipment can be either available or not-available. Thus, this can be handled using Boolean values i.e., if 1: Available and if 0: Not Available.

Function: `void validate_status(string num, Equipment &E);`

This function takes a string `num` and an Equipment `E` as parameters. It checks if `num` is either equal to "1" or "0", if yes then it sets status of Equipment `E` accordingly; else it will repeatedly ask the user to enter valid input. This function does not return anything.

```

                                     A D D   D A T A
-----
Enter the number of equipments you want to add: 1
-----

      Id :      ML004
      Description* :
      Description is required :
      Description is required :      Load bank 30A 400V
      Date of purchase(DD/MM/YYYY)* :      29 Feb 2019
Invalid date format. Plz use DD/MM/YYYY :      29/02/2019
Invalid Date format. Plz use DD/MM/YYYY :      29.02.2019
Invalid Date format. Plz use DD/MM/YYYY :      29/13/2029
Invalid Date format. Plz use DD/MM/YYYY :      29/02/2020
      Quantity* :
      Quantity is required :      -9
Invalid input. Enter a positive integer :      three
Invalid input. Enter a positive integer :      3
      Amount :      -7500
Invalid input. Enter a positive float value :      7500.999
Invalid input. Enter a positive float value :      7500.99
      Grant :
      Purchaser :
      Purchase details :      MTMS engineers Pvt Ltd
Status (1: Available; 0: Not Available)* :      available
Invalid input please try again :      1
      Sign of lab coordinator :
      Remarks :

-----
Adding the data....._
```

(In the above image, we have demonstrated all the validations mentioned above. Note: 3 Equipments were already present in the database, so ID generated is ML004. Also, some empty field that got excepted are optional fields)

7. Modify Data

Imp: This feature is only available for users logged in through ADMIN login.

Humans are prone to mistakes and changes are constants of life. Thus, modifying previously entered data is a must feature of any application that is based on the principle of storing and managing data. We have implemented modify data feature in our application. This feature is only available in admin login.

Salient feature of modify data feature:

1. It is available only in admin login.
2. The equipment to be modified can be accessed by its ID only.
3. ID cannot be modified, since it is auto-generated field.
4. When the user enters the ID of equipment to be modified, the previously entered details of that equipment are displayed along with a form to take new modified inputs.
5. All the fields in this form are optional, if the user leaves a field blank then the previously entered information of the field is retained; else it is modified.
6. All the validations that we discussed above are also handled here, except the required data validation (since all fields are optional here).
7. If the Equipment with specified ID is not present in the database, then an error message is displayed.

```
MODIFY DATA

Enter the id of equipment to be modified : ml100

OOPS!!! Equipment with id=ml100 is not present in the data._
```

(Error message when trying to modify an equipment that is not present in the database)

MODIFY DATA

Enter the id of equipment to be modified : ML003

EXISTING INFORMATION :

Id :	ML003
Description :	DC shunt motor with loading arrangement
Date of purchase(DD/MM/YYYY) :	29/02/2020
Quantity :	3
Amount :	7500.99/-
Grant :	
Purchaser :	
Purchase details :	Energio
Status :	1
Sign of lab coordinator :	
Remarks :	

ENTER THE MODIFICATIONS :

Note: If you don't want to modify a particular field then press ENTER

Id :	ML003
Description :	
Date of purchase(DD/MM/YYYY) :	
Quantity :	4
Amount :	Ten Thousand
Invalid input. Enter a positive float value :	10000.50
Grant :	
Purchaser :	
Purchase Details :	
Status (1: Available; 0: Not Available) :	Not available
Invalid input please try again :	0
Sign of lab coordinator :	
Remarks :	

Modifying the data..._

(Demonstration of data modification. Note: No field is required here. All other validations are also handled here)

Functions written to implement add data feature

1. void modify(All_Equipments &All):

This function prompts the user to enter the ID for Equipment whose details he/she wants to modify. If the equipment with that id is present in the database; then the current details of that equipments are displayed by calling a function (void display_single(All_equipments &All, int i)); after the current details are display, the user is prompted to enter the modified field. If the user leaves any field blank then, the current information of that field is retained; else the field is modified with the new info. If the equipment with the id entered is not present in the database, then error message is display. This function does not return anything

8. Display Data

In this application, we provide different options to display(i.e., view data). So, we have a separate Dashboard of the Display Data operations.

Display Dashboard:

```

      _____
      |                                     |
      |             D I S P L A Y   D A S H B O A R D             |
      |_____|
      |
      | KEYS FOR DISPLAY OPERATIONS: |
      |_____|
      |
      | W: Whole Data                |
      | A: Available Equipments      |
      | N: Non-Available Equipments  |
      | D: Sorted Equipments by Date of Purchase |
      | P: Equipments purchased over a Period |
      | S: Amount Statistics          |
      | E: Move to the main operations chart. |
      |_____|
      |
      | CHOOSE THE OPERATION THAT YOU WANT TO PERFORM : _ |
      |_____|

```

(This is the display dashboard for admin login and lab-assistant login)

```

      _____
      |                                     |
      |             D I S P L A Y   D A S H B O A R D             |
      |_____|
      |
      | KEYS FOR DISPLAY OPERATIONS: |
      |_____|
      |
      | W: Whole Data                |
      | A: Available Equipments      |
      | N: Non-Available Equipments  |
      | E: Move to the main operations chart. |
      |_____|
      |
      | CHOOSE THE OPERATION THAT YOU WANT TO PERFORM : |
      |_____|

```

(This is the display dashboard for student login)

DISPLAY FEATURES IN DIFFERENT LOGINS

Types of Logins Feature	Admin Login	Lab Assistant Login	Student Login
1. Whole Data	✓	✓	✓
2. Available Equipments	✓	✓	✓
3. Not-Available Equipments	✓	✓	✓
4. Sorted Equipments by date of purchase	✓	✓	✗
5. Equipments Purchases over a period	✓	✓	✗
6. Amount Statistics	✓	✓	✗

FIELDS DISPLAYED IN DIFFERENT LOGINS

Types of Logins Fields	Admin Login	Lab Assistant Login	Student Login
1. ID	✓	✓	✓
2. Description	✓	✓	✓
3. Date of purchase	✓	✓	✗
4. Quantity	✓	✓	✓
5. Amount	✓	✓	✗
6. Grant	✓	✓	✗
7. Purchaser	✓	✓	✗
8. Purchase Details	✓	✓	✗
9. Status	✓	✓	✓
10. Sign of Lab Co-ordinator	✓	✓	✗
11. Remarks	✓	✓	✗

Different display features are as follows:

1. Whole Data
(Available in all logins)

D I S P L A Y I N G T H E E N T I R E D A T A		
Id :	ML001	
Description :	Three phase inductive loading bank	
Date of purchase(DD/MM/YYYY) :	26/06/2017	
Quantity :	1	
Amount :	1200/-	
Grant :		
Purchaser :		
Purchase details :	Aashivi electricals	
Status :	1	
Sign of lab coordinator :		
Remarks :		
Id :	ML002	
Description :	lip ring induction motor	
Date of purchase(DD/MM/YYYY) :	30/10/2003	
Quantity :	4	
Amount :	7500/-	
Grant :		
Purchaser :		
Purchase details :	Super scientific industries	
Status :	0	
Sign of lab coordinator :		
Remarks :		
Id :	ML003	
Description :	DC shunt motor with loading arrangement	
Date of purchase(DD/MM/YYYY) :	29/02/2020	
Quantity :	4	
Amount :	10000/-	
Grant :		
Purchaser :		
Purchase details :	Enerzio	
Status :	0	
Sign of lab coordinator :		
Remarks :		
Id :	ML004	
Description :	Load bank 30A 400V	
Date of purchase(DD/MM/YYYY) :	29/02/2020	
Quantity :	3	
Amount :	7500/-	
Grant :		
Purchaser :		
Purchase details :	MTMS engineers Pvt Ltd	
Status :	1	
Sign of lab coordinator :		
Remarks :		
Press ENTER to go back to operations chart.		

(Displays the entire data present in the database – admin / lab-assistant login)

2. Available Equipments (Available in all logins)

D I S P L A Y I N G T H E A V A I L A B L E E Q U I P M E N T S		
Id :	ML001	
Description :	Three phase inductive loading bank	
Date of purchase(DD/MM/YYYY) :	26/06/2017	
Quantity :	1	
Amount :	1200/-	
Grant :		
Purchaser :		
Purchase details :	Aashivi electricals	
Status :	1	
Sign of lab coordinator :		
Remarks :		
Id :	ML004	
Description :	Load bank 30A 400V	
Date of purchase(DD/MM/YYYY) :	29/02/2020	
Quantity :	3	
Amount :	7500/-	
Grant :		
Purchaser :		
Purchase details :	MTMS engineers Pvt Ltd	
Status :	1	
Sign of lab coordinator :		
Remarks :		
Press ENTER to go back to operations chart.		

(Displays all the equipments that are presently available in the laboratory – admin / lab-assistant login)

3. Not-Available Equipments (Available in all logins)

D I S P L A Y I N G T H E N O T - A V A I L A B L E E Q U I P M E N T S		
	Id :	ML002
	Description :	lip ring induction motor
Date of purchase(DD/MM/YYYY)	:	30/10/2003
Quantity	:	4
Amount	:	7500/-
Grant	:	
Purchaser	:	
Purchase details	:	Super scientific industries
Status	:	0
Sign of lab coordinator	:	
Remarks	:	
	Id :	ML003
	Description :	DC shunt motor with loading arrangement
Date of purchase(DD/MM/YYYY)	:	29/02/2020
Quantity	:	4
Amount	:	10000/-
Grant	:	
Purchaser	:	
Purchase details	:	Enerzio
Status	:	0
Sign of lab coordinator	:	
Remarks	:	
Press ENTER to go back to operations chart._		

(Displays all the equipments that are presently not available in the laboratory– admin / lab-assistant login)

4. Sorted Equipments by Date of Purchase (Not available in Student login)

D I S P L A Y I N G T H E S O R T E D D A T A		
<hr/>		
Id :	ML002	
Description :	lip ring induction motor	
Date of purchase(DD/MM/YYYY) :	30/10/2003	
Quantity :	4	
Amount :	7500/-	
Grant :		
Purchaser :		
Purchase details :	Super scientific industries	
Status :	0	
Sign of lab coordinator :		
Remarks :		
<hr/>		
Id :	ML001	
Description :	Three phase inductive loading bank	
Date of purchase(DD/MM/YYYY) :	26/06/2017	
Quantity :	1	
Amount :	1200/-	
Grant :		
Purchaser :		
Purchase details :	Aashivi electricals	
Status :	1	
Sign of lab coordinator :		
Remarks :		
<hr/>		
Id :	ML003	
Description :	DC shunt motor with loading arrangement	
Date of purchase(DD/MM/YYYY) :	29/02/2020	
Quantity :	4	
Amount :	10000/-	
Grant :		
Purchaser :		
Purchase details :	Enerzio	
Status :	0	
Sign of lab coordinator :		
Remarks :		
<hr/>		
Id :	ML004	
Description :	Load bank 30A 400V	
Date of purchase(DD/MM/YYYY) :	29/02/2020	
Quantity :	3	
Amount :	7500/-	
Grant :		
Purchaser :		
Purchase details :	MTMS engineers Pvt Ltd	
Status :	1	
Sign of lab coordinator :		
Remarks :		
<hr/>		
Press ENTER to go back to operations chart.▀		

(Equipments are sorted according to their date of purchase and then displayed– admin / lab-assistant login)

5. Equipments purchased over a Period (Not available in Student login)

```

Enter the start date :      01/01/2000
Enter the stop date  :      01/01/2020

DISPLAYING EQUIPMENTS PURCHASED FROM 01/01/2000 TO 01/01/2020

Id : ML002
Description : lip ring induction motor
Date of purchase(DD/MM/YYYY) : 30/10/2003
Quantity : 4
Amount : 7500/-
Grant :
Purchaser :
Purchase details : Super scientific industries
Status : 0
Sign of lab coordinator :
Remarks :

Id : ML001
Description : Three phase inductive loading bank
Date of purchase(DD/MM/YYYY) : 26/06/2017
Quantity : 1
Amount : 1200/-
Grant :
Purchaser :
Purchase details : Aashivi electricals
Status : 1
Sign of lab coordinator :
Remarks :

*****

Total Equipments bought :      2
Total Amount spent :      8700/-

*****

Press ENTER to go back to operations chart.

```

(Equipments bought over specified period are displayed; along with expenditure statistics—admin/lab-assistant login)

6. Amount Statistics (Not available in Student login)

```

DISPLAYING THE AMOUNT STATISTICS

Amount spent on buying available equipments :      8700/-
Amount spent on buying unavailable equipments :      17500/-

TOTAL :      26200/-

Press ENTER to go back to operations chart.

```

(Overall expenditure statistics are displayed— admin / lab-assistant login)

Fields displayed in Student login:

1. Whole Data

Fields that are of use for students like – Id, Description, Quantity, Status are only displayed. Same is true while use other display features through student login.

D I S P L A Y I N G T H E E N T I R E D A T A	
Id :	ML001
Description :	Three phase inductive loading bank
Quantity :	1
Status :	1
Id :	ML002
Description :	lip ring induction motor
Quantity :	4
Status :	0
Id :	ML003
Description :	DC shunt motor with loading arrangement
Quantity :	4
Status :	0
Id :	ML004
Description :	Load bank 30A 400V
Quantity :	3
Status :	1
Press ENTER to go back to operations chart._	

(Displays the entire data present in the database – student login)

Functions implemented for Display feature:

1. **void display_single()** - Displays ith equipment from the database
2. **void display_all()** - Calls display_single multiple times and displays all the equipments present.
3. **void display_avail()** - Parses through the entire data and check the status of each equipment. If status is 1; then it calls display_single on that equipments; thus, displaying only available equipments.
4. **void display_n_avail()** - Parses through the entire data and check the status of each equipment. If status is 0; then it calls display_single on that equipments; thus, displaying only unavailable equipments.
5. **void display_sorted()** - Sorts the entire set of equipments present in the data, with date of purchase as key; and then displays them.
6. **void display_period()** - Sorts the entire set of equipments present in the data, with date of purchase as key; and then displays the equipments purchased from date d1 to date d2. D1 and d2 are taken as input from the user. After displaying the equipments, it also displays a small amount statistics.
7. **void display_statistics()** - Displays the overall expenditure statistics.

9. Search through Data

In this application, we provide different options to search through the data. So, we have a separate Dashboard of the Search through Data operations.

Search Dashboard:

```
SEARCH DASHBOARD

KEYS FOR SEARCH OPERATIONS:

I: Search by ID.
D: Search by description.
E: Move to the main operations chart.

CHOOSE THE OPERATION THAT YOU WANT TO PERFORM : i
```

(Search Dashboard in all types of logins)

Different types of search options are as follows:

1. Search by ID

The application parses through the entire available data and displays the equipment with the ID specified by the user. If no such equipment is found, then an error message is displayed.

```
SEARCH THROUGH DATA BY ID

Enter the ID of the equipment to be searched : ML002

      Id : ML002
Description : lip ring induction motor
Date of purchase(DD/MM/YYYY) : 30/10/2003
      Quantity : 4
      Amount : 7500/-
      Grant :
Purchaser :
Purchase details : Super scientific industries
      Status : 0
Sign of lab coordinator :
      Remarks :

Press ENTER to close the display._
```

(Displays the equipment if found in the database; else it will display an error message)

2. Search by description

The application parses through the entire data available and displays the equipments which have the keyword entered by the users; if no such equipment is found in the database then an error message is displayed.

```
SEARCH THROUGH DATA BY DESCRIPTION

Enter some keywords to find the equipments : induct

Id : ML001
Description : Three phase inductive loading bank
Date of purchase(DD/MM/YYYY) : 26/06/2017
Quantity : 1
Amount : 1200/-
Grant :
Purchaser :
Purchase details : Aashivi electricals
Status : 1
Sign of lab coordinator :
Remarks :

Id : ML002
Description : lip ring induction motor
Date of purchase(DD/MM/YYYY) : 30/10/2003
Quantity : 4
Amount : 7500/-
Grant :
Purchaser :
Purchase details : Super scientific industries
Status : 0
Sign of lab coordinator :
Remarks :

Press ENTER to close the display._
```

(Displays all the equipments that has the keyword in its description)

```
SEARCH THROUGH DATA BY DESCRIPTION

Enter some keywords to find the equipments : xxx

OOPS!!! No matching results found!!!

Press ENTER to close the display._
```

(No equipment in the database has 'xxx' in its description, so error message is displayed)

Functions implemented for Search feature:

1. **int search_by_id()** - Parses through the data, checks if equipment with id entered by user is present in the database or not; if present then displays it; else it prints an error message.
2. **void search_by_description()** - Parses through the data, checks if any equipment has the keyword entered by user in its description; if present then displays all such equipments; else it prints an error message.

10. Logout

The application has three types of logins. Thus, there is a need of the logout feature too. We have implemented the logout feature in the application, when the user logs out, then the Login Dashboard is displayed again.

```

D A S H B O A R D

KEYS FOR THE OPERATIONS:

A: Add equipments to the data.
D: Display DashBoard.
M: Modify the details of an equipment.
S: Search DashBoard.
E: Logout.

CHOOSE THE OPERATION THAT YOU WANT TO PERFORM : e

Logging out.....
```

(On choosing e or E, and pressing enter, then user is logged out and taken back to the Login Dashboard)

Close the application:

```

W E L C O M E   T O   M A C H I N E S   L A B

Types of Logins available:

A: Login as Admin.
L: Login as Lab Assistant.
S: Login as Student.
E: Close the Application

CHOOSE THE OPERATION THAT YOU WANT TO PERFORM : e
```

(On the Login Dashboard, we have 'Close the Application' option. This option will terminate the program)

11. Storing the Data

Every time the user(admin) adds or modifies the data, the data is added to a Comma-Separated Values file (.csv file) named as ml_data.csv. This file is updated every time the add and modify is used.

Moreover, every time we run the application, it will first search for this file(ml_data.csv); if the file does not exist then it will create a blank csv file with name ml_data.csv. If the file is already present, then it will first read the entire file and store all the equipments present in the Equipments array and also set the global variable size accordingly.

	A	B	C	D	E	F	G	H	I	J	K
1	ID	DESCRIPTION	DATE OF PURCHASE	QUANTITY	AMOUNT	GRANT	PURCHASER	PURCHASE DETAILS	STATUS	SIGN	REMARK
2	ML001	Three phase inductive loading bank	26-06-2017	1	1200			Aashivi electricals	1		
3	ML002	lip ring induction motor	30-10-2003	4	7500			Super scientific industries	0		
4	ML003	DC shunt motor with loading arrangement	29-02-2020	4	10000.5			Energzio	0		
5	ML004	Load bank 30A 400V	29-02-2020	3	7500.99			MTMS engineers Pvt Ltd	1		
6											
7											

(All the data that we added and modified so far is presented in the csv file)

Functions implemented for storing data:

1. **void read_data()** - Opens the file ml_data.csv, read the file line by line; splits each line over comma(,) and stores the extracted values in Equipment E. Later, stores this E in the All_Equipments All[] array. For each line it reads, it will increment the global variable size by 1.
2. **void put_data()** - Opens the file ml_data.csv, and writes the data present in All equipments All[] array into the file.

Note: Every time we run the application, the program first checks if the file ml_data.csv already exists or not; if not, then it creates an empty csv file ml_data.csv.

12. Future Scope

This application can be used in schools and colleges for systematic and efficient management of equipments in a laboratory.

This application was particularly created for Machines Laboratory. The application can be extended further by integrating it with the applications for all the laboratory of an institution, even the library management system can be integrated; and a single centralized management system can be implemented and brought in use.

Moreover, an improved Graphical User Interface (GUI) can be implemented in this application.

13. Conclusion

We have successfully implemented a laboratory management system (Machines Laboratory) using C++ as the programming language. The application has features like login, add data, modify data, display the data, searching through the data and sorting the data in a csv file. Moreover, user input validations were implemented successfully, such that there is no way that the program could crash.