

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Santhibastawad Road, Machhe  
Belagavi - 590018, Karnataka, India



## A MINI PROJECT REPORT ON

### **“Meet Management System”**

Submitted in the partial fulfilment of the requirements for the award of the degree of

## **BACHELOR OF ENGINEERING IN INFORMATION SCIENCE AND ENGINEERING**

For the Academic Year 2020-2021

Submitted by

**Abhay PJ**

**Manoj GH**

**Khadija Fatima**

**1JS18IS001**

**1JS18IS048**

**1JS18IS041**

Under the Guidance of

**Mrs. Punitha M Assistant Professor, Dept. of**

**ISE, JSSATE**



**2020-2021**

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

**JSS ACADEMY OF TECHNICAL EDUCATION**

**JSS Campus, Dr. Vishnuvardhan Road, Bengaluru-560060**

JSS MAHAVIDYAPEETHA, MYSURU  
**JSS ACADEMY OF TECHNICAL EDUCATION**  
JSS Campus, Dr.Vishnuvardhan Road, Bengaluru-560060

**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**



**CERTIFICATE**

This is to certify that the File Structures Laboratory Mini Project entitled  
**“Meet Management System”** is a work carried out by **Manoj GH [1JS18IS048]**,  
**Abhay PJ [1JS18IS001]**, **Khadija Fatima [1JS18IS041]** in partial fulfilment for the  
award of degree of Bachelor of Engineering in Information Science and Engineering of  
Visvesvaraya Technological University Belagavi during the year 2020-2021.

---

**Signature of the Guide**

**Mrs. Punitha M**  
**Assistant Professor**  
Dept. of ISE  
JSSATE, Bengaluru

Name of the Examiner:

1. ....

---

**Signature of the HOD**

**Dr. Rekha P.M**  
Associate Prof. & Head  
Dept. of ISE  
JSSATE, Bengaluru

Signature with date

.....

## **ABSTRACT**

Meet management system is a project which aims in developing a computerized system to maintain all the daily work of Students. The main objective of the Meet Management system project is discipline of the planning, organizing and managing the meet links. Our project aims at making the task of students easy. Meet Management is Enterprise Software Application which is used to store , update , delete and add events. We can search events to get the corresponding link .As we receive a lot of meeting links for online lecture and official meeting, it's tough to handle these links so this app is used to retrieve the link by mentioning Topic. Meet Management is entering the records of new event and retrieving the meet link available in the data base. We can search the links and edit the event. This project is developed using Python and concept indexing. The tools used to implement the above project were Tkinter , subprocess , Openpyxl used to operate on file excel. The file structure concept used is indexing. Overall, this project of ours is being developed to help the students in the best way possible and also reduce the human effort.

## **ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible. So, with gratitude, we acknowledge all those whose guidance and encouragement crowned my effort with success.

First and foremost, we would like to thank his **Holiness Jagadguru Sri Shivarathri Deshikendra Mahaswamiji** and **Dr. Mrityunjaya V Latte**, Principal, JSSATE, Bangalore for providing an opportunity to carry out the Project Work as a part of our curriculum in the partial fulfilment of the degree course.

We express our sincere gratitude for our beloved Head of the department, **Dr. Rekha P.M.**, for her co-operation and encouragement at all the moments of our approach.

It is our pleasant duty to place on record our deepest sense of gratitude to our respected guides **Mrs. Punith M Assistant Professor**, for the constant encouragement, valuable help and assistance in every possible way.

We would like to thank all ISE Department Teachers, non-teaching staff for providing us with their valuable guidance and for being there at all stages of our work.

**Manoj GH[1JS18IS048]**  
**Abhay PJ[1JS18IS001]**  
**Khadija F[1JS18IS041]**

# Table of Content

Chapter Title	Page no
<b>Chapter 1. Introduction</b>	
1.1 Meet Management System	1
1.2 Objectives of the Project	2
<b>Chapter 2. Literature Survey</b>	
2.1 History of File Structures	3
2.2 Types of Files	3
2.3 Types of Files Operations	4
<b>Chapter 3. Introduction to Indexing</b>	
3.1 Indexing	5
3.2 Types of Indexing	6
3.3 Purpose of Indexing	7
3.4 Advantages of Indexing	7
3.5 Disadvantages of Indexing	7
<b>Chapter 4. System Requirement</b>	
4.1 Software Requirement	8

4.2 Hardware Requirement	8
--------------------------	---

## **Chapter 5. System Design**

5.2 Student Login	9
-------------------	---

## **Chapter 6. System Implementation**

6.1 Source code	10
6.1.1 Welcome page	11
6.1.2 Options menu	13
6.1.3 Student login	14
6.1.4 Indexing	18

## **Chapter 7. Testing**

6.1 Unit Testing	19
6.2 Integration Testing	19
6.3 System Testing	19

## **Chapter 8. Results**

7.1 Snapshot forms of Project	21-24
-------------------------------	-------

## **Chapter 9. Conclusion and future Work**

9.1 Conclusion	25
9.2 Future Enhancements	25

## **Chapter 10. Bibliography**

10.1 Book References	26
10.2 Web Link	26

## **Chapter 1:**

# **INTRODUCTION**

Meet Management System is based on a concept of Enterprise Software Application which is used to store , update , delete and add events. We can search events to get the corresponding link .As we receive alot of meeting links for online lecture and offical meeting, it's tough to handle these links so this app used to retrieve the link by mentioning Topic.Before stepping into the main system, a user has to pass through a login system to get access, there are two types of login i.e.Student and Librarian Login. To get access to the Librarian's account, a password should be provided for the Librarian only.

### **Meet Management System:**

Features of the Meet Management System, after logging in as a student, they can only view timetable and search for the links by the subject name. All the system is controlled by the student as he/she can view, search, modify, add and links and timetable..

### **Objectives of the Project:**

The aims and objectives of our project are as follows:

- Having a miss interpretation of links
- A secure login system.
- A separate file to find links as per the days.
- A search column to search for a link .
- A student can view timetable and add ,view,delete links.

## **Chapter 2:**

# **LITERATURE SURVEY**

### **History of File Structures:**

Early work with files presumed that files were on tape, since most files were. Access was sequential and the cost of access grew in direct proportion to the size of the file. The indexes made it possible to keep a list of keys and pointers in a smaller file that could be searched more quickly. With the key and pointer, the user had direct access to the large, primary file.

Unfortunately, simple indexes had some of the same sequential flavor as the data files, and as the indexes grew, they too became difficult to manage, especially for dynamic files in which the set of keys changes. Then in the early 1960's, the idea of applying tree structures emerged. Unfortunately, trees can grow very unevenly as records are added and deleted.

In 1963 researchers developed the tree, an elegant, self-adjusting binary tree structure, called AVL tree, for data in memory. The problem was that even with a balanced binary tree, dozens of accesses were required to find a record in even moderate sized files.

It took nearly ten more years of design work before a solution emerged in form of the B-tree. AVL trees grows from top down as records are added, B-trees grow from the bottom up.

B-trees provided excellent access performance, but there was a cost: no longer could a file be accessed sequentially with efficiency. Fortunately, this problem was solved almost immediately by adding a linked list structure at the bottom level of the B-tree. The combination of a B-tree and a sequential linked list is called a B+ tree.

Being able to retrieve information with just three or four accesses is pretty good. But the goal of being able to get what we want with a single request was satisfied by hashing. From early on, hashed indexes were used to provide fast access to file. After the developed of B-trees, researchers turned to work on systems for extendible, dynamic hashing that could retrieve information with one or, at most, two disk accesses no matter how big the file became.



## **TYPES OF FILES:**

As we know that computers are used for storing the information for a permanent time or the files are used for storing the Datta of the users for a long time period. And the files can contain any type of information which means that they can store text, any images or pictures or data in any format. So that there must be some mechanism which are used for storing the information, Accessing the information and also performing some operations on the files.

When we store a file in the system, then we have to specify the name and the type of file. The name of file will be any valid name and type means the application with which the file is linked.

When we say that every file has some type, it means that every file belongs to a special type of application software. When we provide a name to a file, we also specify the extension of the file because the system will retrieve the contents of that file into application software. For example, if there is a file which contains some paintings then this will open in software.

1. **Ordinary file or Simple file:** Ordinary file may belong to any type of application. Forexample, notepad, paint, C program, music[mpeg] etc. All the files that are created by a user are ordinary files. Ordinary files are used for storing information about the user programs. With the help of ordinary files, we can store the information which contain text, database, images or any other type of information.
2. **Directory File:** The files those are stored into a particular directory or folder. These are known as Directory files because they belong to a particular directory.
3. **Special Files:** The special files are those which are not created by the user or the files which are necessary to run a system. These are the files that are created by the system. All the files of an operating system are referred to as special files. There are many types of special files and they are, system files or windows files, input/output files. System files are stored using .sys extension.
4. **FIFO Files:** The first in first Out Files are used by the system for executing the processes in a particular order. Which means that the files which comes first, will be executed first and the system maintains a order known as Sequence order. When a user requests for a service from the system, then the requests of the users are arranged into some files and all the requests will be performed by the system, using some sequential order known as FIFO order.

### **Type of File Operations:**

Files are not created only for reading, we can also perform some operations on files, they are

1. Read Operation: This is meant to read the content which is stored in the files.
2. Write Operation: This is meant to read the content which is stored in the files.
3. Rename or Change the name of file.
4. Copy the file from one location to another.
5. Sorting or Arranging the contents of file.
6. Move or copy the file from one place to another.
7. Delete a file.

Execute the file (this displays the output of the file).

## **Chapter 3:**

# **INTRODUCTION TO INDEXING**

## **Indexing:**

Indexing is a data structure technique which allows you to quickly retrieve records from a database file. An Index is a small table having only two columns. The first column comprises a copy of the primary or candidate key of a table. Its second column contains a set of pointers for holding the address of the disk block where that specific key value stored.

A good hashing algorithm would exhibit a property called the avalanche effect, where the resulting hash output would change significantly or entirely even when a single bit or byte of data within a file is changed. A hash function that does not do this is considered to have poor randomization, which would be easy to break by hackers.

A hash is usually a hexadecimal string of several characters. Hashing is also a unidirectional process so you can never work backwards to get back the original data.

A good hash algorithm should be complex enough such that it does not produce the same hash value from two different inputs. If it does, this is known as a hash collision. A hash algorithm can only be considered good and acceptable if it can offer a very low chance of collision.

## **Types of Indexing:**

**Primary Index** – Primary index is defined on an ordered data file. The data file is ordered on a key field. The key field is generally the primary key of the relation.

**Secondary Index** – Secondary index may be generated from a field which is a candidate key and has a unique value in every record, or a non-key with duplicate values.

**Clustering Index** – Clustering index is defined on an ordered data file. The data file is ordered on a non-key field.

### **Ordered Indexing is of two types –**

Dense Index

Sparse Index

#### **Dense Index**

In dense index, there is an index record for every search key value in the database. This makes searching faster but requires more space to store index records itself. Index records contain search key value and a pointer to the actual record on the disk.

#### **Sparse Index**

In sparse index, index records are not created for every search key. An index record here contains a search key and an actual pointer to the data on the disk. To search a record, we first proceed by index record and reach at the actual location of the data. If the data we are looking for is not where we directly reach by following the index, then the system starts sequential search until the desired data is found.

#### **Multilevel Index**

Index records comprise search-key values and data pointers. Multilevel index is stored on the disk along with the actual database files. As the size of the database grows, so does the size of the indices. There is an immense need to keep the index records in the main memory so as to speed up the search operations. If single-level index is used, then a large size index cannot be kept in memory which leads to multiple disk

## **Purpose of Indexing:**

Here, are the situations in where you need to apply the Indexing method:

- Indexing is to access your data faster; the reads and updates in the database.
- Indexes allow you to lookup a small subset of the fields called as keys, which are stored in a sorted order and find the actual data using the key. These keys in the index point to the actual data records. It is analogous to a book with an index, where one can look for a word or phrase (in the index), find the page number and get to the content, quickly. You don't have to browse thru the book to get to the topic you need to access.
- The indexes also help in sorting operations, if the sort is on an indexed field. Also, a single index can be of use in multiple queries.
- Indexes occupy storage space and while in operation, the memory. When data is created or updated the indexes need updates, sometimes; that is extra writes. So, using indexes needs careful study.
- Databases offer various types of indexes for various needs; e.g., index on multiple fields, unique index, index for text search, etc.

## **Advantages of Indexing:**

- Their use in queries usually results in much better performance.
- They make it possible to quickly retrieve (fetch) data.
- They can be used for sorting. A post-fetch-sort operation can be eliminated.
- Unique indexes guarantee uniquely identifiable records in the database.

## **Disadvantages of Indexing:**

- Difficult to adjust, modify or update
- Due to the nature of the format, any error or adjustment to the command would require rewriting all files after the error.
- It cannot access random file.

## **Chapter 4:**

### **System Requirements**

#### **Software Requirements:**

- Operating system: Windows 7/8/8.1/10 or Linux or macOS
- IDE: Terminal
- Compiler: Python 3

#### **Hardware Requirements:**

- Processor: any intel or AMD processor with at least 2 cores
- RAM: minimum 2GB RAM
- Storage: approximately 500MB storage

## **Chapter 5:**

# **SYSTEM DESIGN**

The library management system has: Student Login

### **5.1 Student Login:**

The student has limited access in terms of power. A student can only

- View the time table present in the database.
- Search for the zoom link present in the database.
- Add, update, delete zoom links of respective subjects.

## **Chapter 6:**

# **SYSTEM IMPLEMENTATION**

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively.

The system can be implemented only after thorough testing is done and if it is found to work according to the specification. It involves careful planning, investigation of existing systems and its constraints on implementations, design of methods to achieve the change. Two major tasks of preparing the implementation are education and training of the users and testing of the system.

The implementation phase consists of several activities. The required hardware and software acquisition are carried out. The system may require the need to develop a software. For this purpose, programs are written and tested.

## **Source code:**

The source code of Meet Management System project is given below:

## **Welcome page:**

At the initiation of the software, the users are greeted with a welcome page. The user can explore the system by pressing enter.

## **Options menu:**

After the welcome screen, an options menu is displayed. The options are:

1. User ID
2. Password
3. Exit



## **Student login:**

The librarian can:

- **Add new event:**

```
from openpyxl import *
from tkinter import *

wb = load_workbook(r'/Users/abhaypayadi/Desktop/FSPProject/dataset/monday.xlsx')
sheet = wb.active

def excel():
    sheet.column_dimensions['A'].width = 30
    sheet.column_dimensions['B'].width = 10
    sheet.column_dimensions['C'].width = 10
    sheet.column_dimensions['D'].width = 20
    sheet.column_dimensions['E'].width = 20

def focus1(event):
    subjecte.focus_set()

def focus2(event):
    teachere.focus_set()

def focus3(event):
    stimee.focus_set()

def focus4(event):
    etimee.focus_set()

def focus5(event):
    linke.focus_set()
```

```
def focus6(event):

    periode.focus_set()
def insert():
    if (subjecte.get() == "" and
        teachere.get() == "" and
        stimee.get() == "" and
        etimee.get() == "" and
        linke.get() == "" and
        periode.get() == "")
        print("empty input")
    else:
        current_row = int(periode.get())+int(1)
        sheet.cell(row=current_row , column=1).value = subjecte.get()
        sheet.cell(row=current_row , column=2).value = teachere.get()
        sheet.cell(row=current_row , column=3).value = stimee.get()
        sheet.cell(row=current_row , column=4).value = etimee.get()
        sheet.cell(row=current_row , column=5).value = linke.get()

        title=Label(Frame_name,text="Event add
        Successfully",font=("Impact",10,"bold"),fg="#d77337",bg="white").place(x=0,y=0)

        wb.save(r'/Users/abhaypayadi/Desktop/FSProject/dataset/monday.xlsx')
if __name__ == "__main__":
    root = Tk()
    root.configure(background='white')
    root.title("ClassRoom PinBoard")
    root.geometry("1199x600+100+50")
    root.resizable(False,False)
    Frame_name=Frame(root,bg="white")
    Frame_name.place(x=150,y=500,height=30,width=210)

    excel()
```

```
subject = Label(root, text="Subject", bg="light green")
teacher = Label(root, text="Teacher", bg="light green")
stime = Label(root, text="Start Time", bg="light green")
etime = Label(root, text="End Time", bg="light green")
link = Label(root, text="Link", bg="light green")
```

```
period = Label(root, text="period", bg="light green")
subject.grid(row=1, column=0)
teacher.grid(row=2, column=0)
stime.grid(row=3, column=0)
etime.grid(row=4, column=0)
link.grid(row=5, column=0)
period.grid(row=6, column=0)
```

```
subjecte = Entry(root)
teachere = Entry(root)
stimee = Entry(root)
etimee = Entry(root)
linke = Entry(root)
periode = Entry(root)
subjecte.bind("<Return>", focus1)
teachere.bind("<Return>", focus2)
stimee.bind("<Return>", focus3)
etimee.bind("<Return>", focus4)
linke.bind("<Return>", focus5)
periode.bind("<Return>", focus6)
subjecte.grid(row=1, column=1, ipadx="100")
teachere.grid(row=2, column=1, ipadx="100")
stimee.grid(row=3, column=1, ipadx="100")
etimee.grid(row=4, column=1, ipadx="100")
linke.grid(row=5, column=1, ipadx="100")
periode.grid(row=6, column=1, ipadx="100")
```

```
excel()

submit = Button(root, text="Submit", fg="Black",
                 bg="Red", command=insert)

submit.grid(row=8, column=1)

root.mainloop()
```

- **Display the Time Table:**

```
from tkinter import *
from tkinter import filedialog, messagebox, ttk
import pandas as pd
root = Tk()
root.geometry("1199x600+100+50")
root.title("Monday Timetable")
root.resizable(0, 0)
frame=Frame(root,bg="white")
frame.pack()
tv = ttk.Treeview(frame)
df=pd.read_excel("/Users/abhaypayadi/Desktop/FSProject/dataset/monday.xlsx")
tv["column"]=list(df.columns)
tv["show"]="headings"
for column in tv["column"]:
    tv.heading(column,text=column)
df_rows=df.to_numpy().tolist()
for row in df_rows:
    tv.insert("", "end", values=row)
tv.pack()
```

- **Code to Search for a respective links:**

```
from openpyxl import *
wb = load_workbook(r'/Users/abhaypayadi/Desktop/Data Set/monday.xlsx')
flag=0
name=input("Enter a name")
sheet = wb.active
column_a=sheet['A']
column_e=sheet['E']
for cell in column_a:
    if(cell.value==name):
        rowno=cell.row
        flag=1
if(flag==0):
    print("no subject found ")
else:
    for cell in column_e:
        if(cell.row==rowno):
            print(cell.value)
            wb.save(r'/Users/abhaypayadi/Desktop/Data Set/monday.xlsx')
```

- **Code to delete zoom link**

```
from tkinter import *
from openpyxl import *
root=Tk()
root.title("ClassRoom Pinboard")
root.geometry("1199x600+100+50")
root.resizable(False,False)
Frame1=Frame(root,bg="white")
Frame1.place(x=150,y=150,height=340,width=500)
def describe():
    extra=Label(Frame1,text="Select the day",font=("Goudy old style",15,"bold"),fg="gray",bg="white").place(x=90,y=250)
global extra
lbl_user=Label(Frame1,text="Subject to be deleted",font=("Goudy old style",15,"bold"),fg="gray",bg="white").place(x=90,y=140)
txt_user=Entry(Frame1,font=("times new roman",15),bg="lightgray")
txt_user.place(x=90,y=170,width=350,height=35)
remove_button=Button(Frame1,text="Remove Subject",command=describe,cursor="hand2",bg="white",fg="#d77337",bd=0,font=("times new roman",12)).place(x=90,y=225)
my_menu=Menu(root)
root.config(menu=my_menu)
def mondayfun():
    wb = load_workbook(r'/Users/abhaypayadi/Desktop/FSProject/dataset/monday.xlsx')
    flag=0
    sheet = wb.active
    column_a=sheet['A']
    name=txt_user.get()
```

```
for cell in column_a:
    if(cell.value==name):
        rowno=cell.row
        print(rowno)
        sheet.delete_rows(rowno)
        flag=1
if(flag==0):
    extra=Label(Frame1,text="Invalid Subject name",font=("Goudy old
style",15,"bold"),fg="gray",bg="white").place(x=90,y=250)
    wb.save(r'/Users/abhaypayadi/Desktop/FSProject/dataset/monday.xlsx')
def tuesdayfun():
    wb = load_workbook(r'/Users/abhaypayadi/Desktop/FSProject/dataset/tuesday.xlsx')
    flag=0
    sheet = wb.active
    column_a=sheet['A']
    name=txt_user.get()
    for cell in column_a:
        if(cell.value==name):
            rowno=cell.row
            print(rowno)
            sheet.delete_rows(rowno)
            flag=1
    if(flag==0):
        extra=Label(Frame1,text="Invalid Subject name",font=("Goudy old
style",15,"bold"),fg="gray",bg="white").place(x=90,y=250)
        wb.save(r'/Users/abhaypayadi/Desktop/FSProject/dataset/tuesday.xlsx')
def wednesdayfun():
    wb = load_workbook(r'/Users/abhaypayadi/Desktop/FSProject/dataset/wednesday.xlsx')
    flag=0
    sheet = wb.active
    column_a=sheet['A']
    name=txt_user.get()
    for cell in column_a:
        if(cell.value==name):
            rowno=cell.row
            print(rowno)
            sheet.delete_rows(rowno)
            flag=1
    if(flag==0):
        extra=Label(Frame1,text="Invalid Subject name",font=("Goudy old
style",15,"bold"),fg="gray",bg="white").place(x=90,y=250)
        wb.save(r'/Users/abhaypayadi/Desktop/FSProject/dataset/wednesday.xlsx')
def thursdayfun():
    wb = load_workbook(r'/Users/abhaypayadi/Desktop/FSProject/dataset/thursday.xlsx')
    flag=0
    sheet = wb.active
    column_a=sheet['A']
    name=txt_user.get()
    for cell in column_a:
        if(cell.value==name):
            rowno=cell.row
            print(rowno)
            sheet.delete_rows(rowno)
            flag=1
```

```
if(flag==0):
    extra=Label(Frame1,text="Invalid Subject name",font=("Goudy old
style",15,"bold"),fg="gray",bg="white").place(x=90,y=250)
    wb.save(r'/Users/abhaypayadi/Desktop/FSProject/dataset/thursday.xlsx')
def fridayfun():
    wb = load_workbook(r'/Users/abhaypayadi/Desktop/FSProject/dataset/friday.xlsx')
    flag=0
    sheet = wb.active
    column_a=sheet['A']
    name=txt_user.get()
    for cell in column_a:
        if(cell.value==name):
            rowno=cell.row
            print(rowno)
            sheet.delete_rows(rowno)
            flag=1
    if(flag==0):
        extra=Label(Frame1,text="Invalid Subject name",font=("Goudy old
style",15,"bold"),fg="gray",bg="white").place(x=90,y=250)
        wb.save(r'/Users/abhaypayadi/Desktop/FSProject/dataset/friday.xlsx')
def saturdayfun():
    wb =load_workbook(r'/Users/abhaypayadi/Desktop/FSProject/dataset/saturday.xlsx')
    flag=0
    sheet = wb.active
    column_a=sheet['A']
    name=txt_user.get()
    for cell in column_a:
        if(cell.value==name):
            rowno=cell.row
            print(rowno)
            sheet.delete_rows(rowno)
            flag=1
    if(flag==0):
        extra=Label(Frame1,text="Invalid Subject name",font=("Goudy old
style",15,"bold"),fg="gray",bg="white").place(x=90,y=250)
        wb.save(r'/Users/abhaypayadi/Desktop/FSProject/dataset/saturday.xlsx')
file_menu=Menu(my_menu)
my_menu.add_cascade(label="Days",menu=file_menu)
file_menu.add_command(label="Monday",command=mondayfun)
file_menu.add_command(label="Tuesday",command=tuesdayfun)
file_menu.add_command(label="Wednesday",command=wednesdayfun)
file_menu.add_command(label="Thursday",command=thursdayfun)
file_menu.add_command(label="Friday",command=fridayfun)
file_menu.add_command(label="Saturday",command=saturdayfun)
root.mainloop()
```

```
        show.getline(l.bid,10,'');
        show.getline(l.bname,25,'');
        show.getline(l.authname,25,'');
        show.getline(l.pubname,25,'\n');
        printf("\n%s\t\t%s\t\t%s\t\t%s",l.bid,l.bname,l.authname,l.pubname);
        i++;
    }
    show.close();
}
```

### **Indexing:**

```
for cell in column_a:
    if(cell.value==name):
        rowno=cell.row
        print(rowno)
        sheet.delete_rows(rowno)
        flag=1
    if(flag==0):
        extra=Label(Frame1,text="Invalid Subject name",font=("Goudy old
style",15,"bold"),fg="gray",bg="white").place(x=90,y=250)
        wb.save(r'/Users/abhaypayadi/Desktop/FSProject/dataset/saturday.xlsx')

for cell in column_a:
    if(cell.value==name):
        rowno=cell.row
        flag=1
if(flag==0):
    print("no subject found ")
else:
    for cell in column_e:
        if(cell.row==rowno):
            print(cell.value)

for column in tv["column"]:
    tv.heading(column,text=column)
df_rows=df.to_numpy().tolist()
for row in df_rows:
    tv.insert("", "end", values=row)
```



## **Chapter 7:**

# **TESTING**

Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is defect free.

### **Unit Testing**

It focuses on smallest unit of software design. In this we test an individual unit or group of inter related units. It is often done by programmer by using sample input and observing its corresponding outputs.

Example:

- a. In a program we are checking if loop, method or if a function is working correctly.
- b. Incorrect or misunderstood arithmetic procedures.
- c. Incorrect initialization.

In this project we have implemented and tested login for different users like student, librarian as each unit.

### **Integration Testing**

When a software test case covers more than one unit, it is considered an integration test. When developing a software test case, the lines between units can quickly evolve into integration tests. The dependency itself will not need to be tested and the integration to it will be mocked or faked.

It is of two types: (i) Top down (ii) Bottom up

Example:

- a. Black Box testing –It is used for validation.

Here we ignore internal working mechanism and focus on what the output would be.

- b. White Box testing- It is used for verification.

Here we focus on internal mechanism i.e. how the output is obtained.

### **System Testing**

In this the software is tested such that it works fine for different operating systems. It is covered under the black box testing technique. In this we focus on required inputs and outputs without focusing on internal working.

This includes functional as well as non-functional testing.

<b>Functionality</b>	<b>Action</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Test Result</b>
Accepting user input for login page for different modules	Logs the user by verifying his existence in the record	User is logged into the Home Page of respective module	User logs in without any problem if the user exists in the record	Pass
Retrieval of all records from the files	Displays all records from the files	Should displays all records from the files	Displays all records from the files	Pass
Error handling in login page	Should reload back to the same page	Should not login into any page and must stay in same page	Does not login into any page and must stay in same page	Pass

Table 7.1 Black box testing table

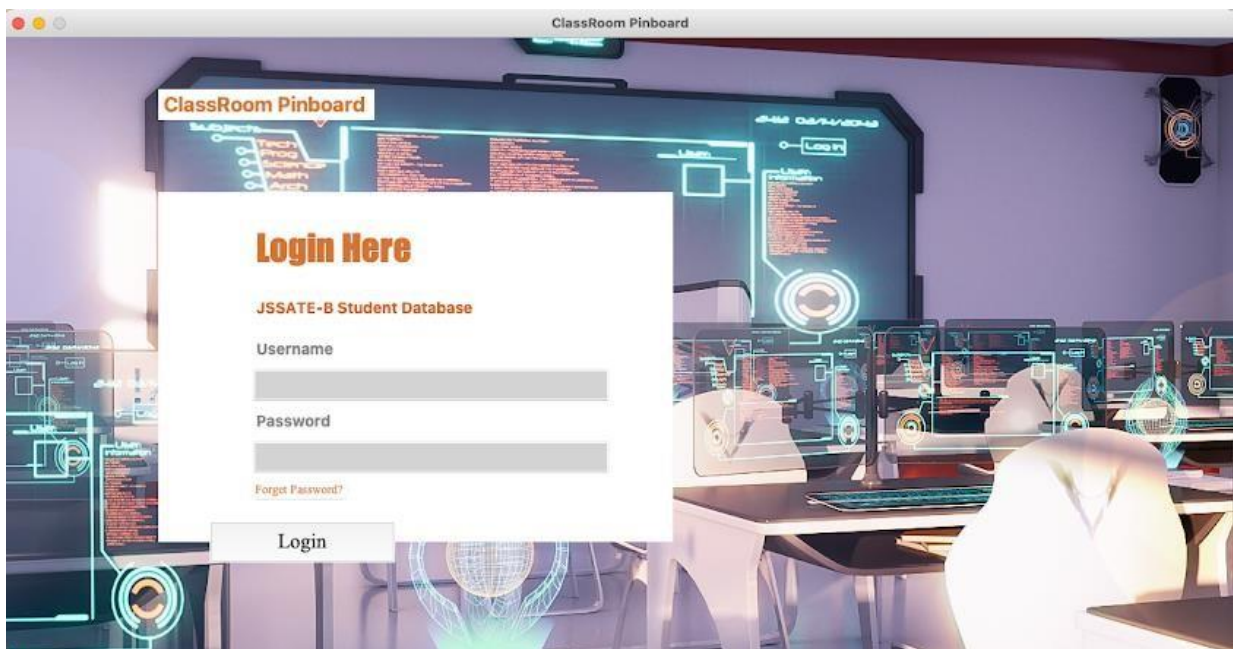
The system has been tested completely for the planned modules and their effective working.

## **Chapter 8:**

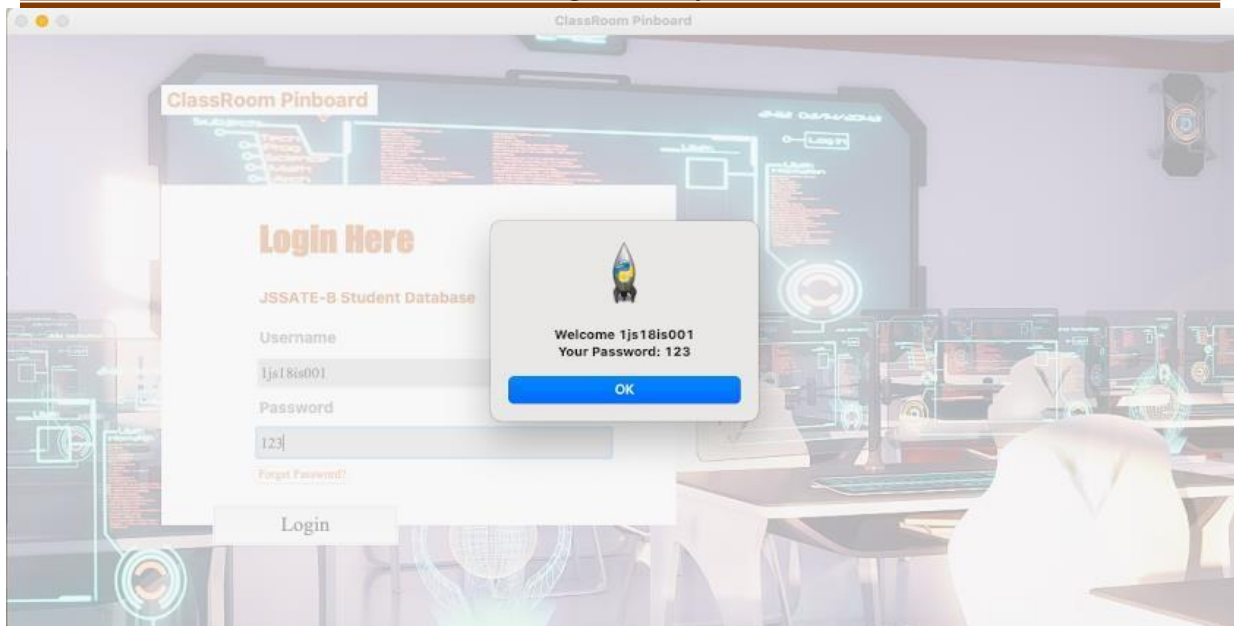
### **RESULTS**

The result contains few of the snap shots of project pages.

#### **8.1 Snap shots:**

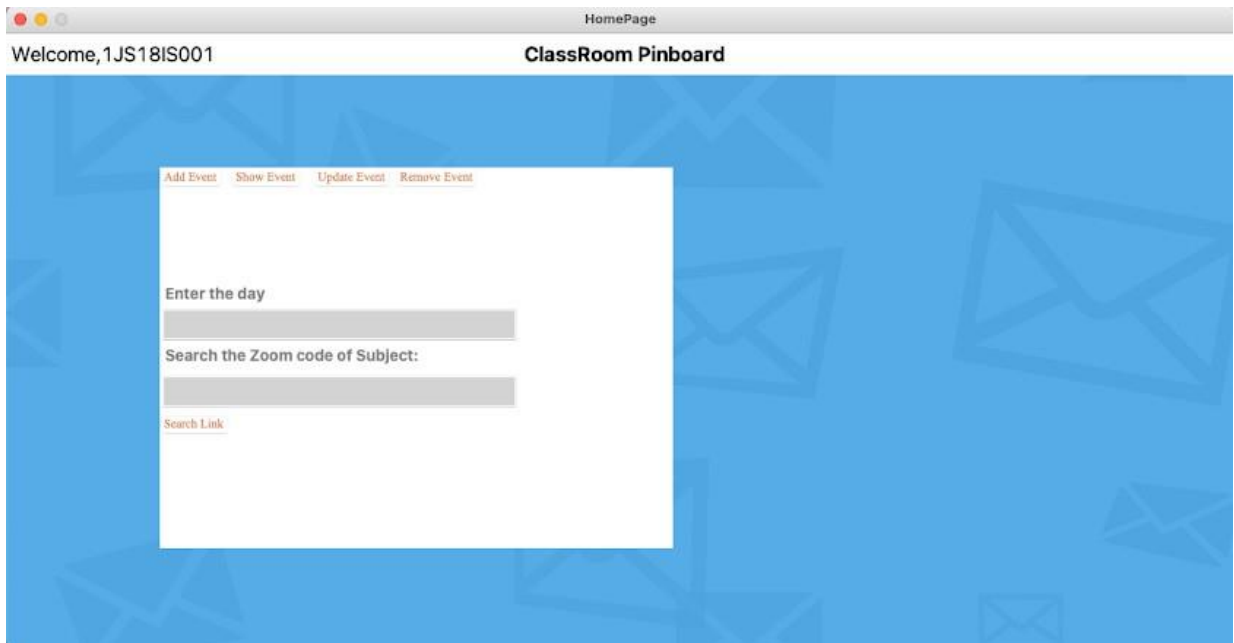


#### **➤ Login page**



- Login successful

### ➤ Home Page

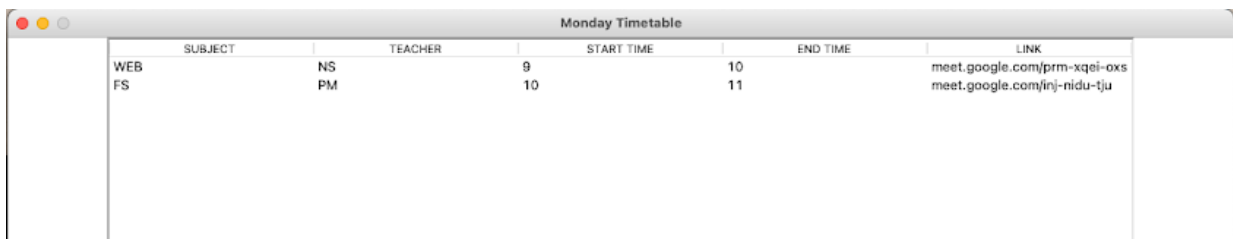


### ➤ Add Link



A web form titled "ClassRoom PinBoard" with a light gray header. Below the header, there are six input fields stacked vertically, each preceded by a green label: "Subject", "Teacher", "Start Time", "End Time", "Link", and "period". At the bottom of the form is a "Submit" button.

### Display TimeTable



A web table titled "Monday Timetable" with a light gray header. The table has six columns: "SUBJECT", "TEACHER", "START TIME", "END TIME", and "LINK". The first two rows contain data: "WEB" by "NS" from 9 to 10, and "FS" by "PM" from 10 to 11. The "LINK" column contains Google Meet links for each row.

SUBJECT	TEACHER	START TIME	END TIME	LINK
WEB	NS	9	10	meet.google.com/prm-xqei-oxs
FS	PM	10	11	meet.google.com/inj-nidu-tju

➤ **Update link**

## Meet Management System

ClassRoom Pinboard

Subject to be updated

Updated Subject

Updated Teacher

Updated Start Time

Update Subject

Updated End Time

Updated Zoom Link

### ➤ Delete Link

ClassRoom Pinboard

Subject to be deleted

Remove Subject

## **Chapter 9:**

# **CONCLUSION AND FUTURE WORKS**

### **9.1 Conclusion:**

This project provides a computerized version of Meet management system which will benefit the students as well as the teaching staff . It makes entire process online where student can search links, view timetable and edit links.

## **Chapter 10:**

### **BIBLIOGRAPHY**

#### **Book references:**

1. File structures- an Object-Oriented Approach with C++, Micheal J Folk, Bill Zoellick, Greg Riccardi, Third edition, Pearson Education, 1998.

#### **Web Link:**

1. <https://youtube.com/playlist?list=PLCC34OHNcOtoC6GglhF3ncJ5rLwQrLGnV>