

A Project Report
on
Automatic Exam Planning and Seating Tool

*Submitted in partial fulfillment of
the requirement for the award of
the degree of*

B.Tech CSE



**Under The Supervision of
Dr. S.Annamalai**

Submitted By

Raghav Sabharwal 18SCSE1180057
Yuvraj Mehta 18SCSE1050008

**SCHOOL OF COMPUTING SCIENCE AND
ENGINEERING DEPARTMENT OF COMPUTER
SCIENCE AND ENGINEERING GALGOTIAS
UNIVERSITY, GREATER NOIDA
INDIA 2022**



**SCHOOL OF COMPUTING SCIENCE AND
ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA**

CANDIDATE'S DECLARATION

We/I hereby certify that the work which is being presented in the thesis/project/dissertation, entitled “**Automatic Exam Planning and Seating Tool**” in partial fulfillment of the requirements for the award of the B.Tech CSE submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of January, 2022 to May 2022, under the supervision of Dr. S.Annamalai, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida.

The matter presented in the thesis/project/dissertation has not been submitted by me for the award of any other degree of this or any other places.

Raghav Sabharwal 18SCSE1180057
Yuvraj Mehta 18SCSE1050008

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr.S.Annamalai

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Raghav Sabharwal 18SCSE1180057, Yuvraj Mehta 18SCSE1050008 has been held on _____ and his work is recommended for the award of B.Tech Computer Science and Engineering.

Signature of Examiner(s)

Signature of Supervisor(s)

Signature of Project Coordinator

Signature of Dean

Date: May, 2022
Place: Greater Noida

Abstract

Conducting a quality based and coordinated examination is a very important task that has no room for errors. Since manual planning has more chances of human error and takes more time, it is best to avoid it. Hence, the idea is to introduce a website based on Python fully connected with a SQL Database of all Student records which includes the course, department, subject and so on is a wise choice. Along with this, a new feature that will set the project apart is taking into account the COVID-19 guidelines for social distancing. Different arrangements of seating patterns are revised in accordance to these guidelines and seating gaps of one's choice can be adjusted in the system. Manual Planning of arrangement of seats can be coordinated for students, only if great effort is put into it, but for staff it is usually not so well made. So, in this system we aim to provide invigilators a well made plan for smooth functioning of examinations. The system will make sure that professors from all types of departments are chosen for assignment to exam halls. Further, this project eliminates all factors such as overlapping of seat numbers for different students, cluttering of students on the notice board for checking assigned seats and even cutting down the huge waiting time that comes with it, making lists and printing them which requires loads of valuable resources like ink and paper and last but not the least, generation of reports becomes a chaotic procedure. Our system works upon removing all these factors and in turn provides QoL improvements to the Indian Education System.

List of Figures

Figure No.	Table Name	Page Number
1	UML Diagram	21
2	Data Flow Diagram	23
3	Code Screenshots	30

Acronyms

HTML	Hypertext Markup Language
SQL	Structured Query Language
IDE	Integrated Development Environment
VS Code	Visual Studio Code
COVID-19	Coronavirus Disease
GUI	Graphic User Interface
CSS	Cascading Style Sheets

Table of Contents

Abstract

List of Figures

Acronyms

Chapter 1	Introduction
	1.1 Introduction
	1.2 Formulation of Problem
	1.2.1 Tool and Technology Used
Chapter 2	Literature Survey/Project Design
Chapter 3	Working of Project
Chapter 4	Future Scope and Benefits
Chapter 5	Conclusion and References
	5.1 Conclusion
	5.2 Reference
	5.3 Publication/Acceptance Screenshots

Chapter 1

Introduction

Here we have made a project based on python for automating seat planning in education institutions. The project will have a website for easy navigation and will use dataset collected from colleges where this problem of manual seating plans exist. Based on some sorting algorithms, we will get seating arrangement of students as well as faculties. Along with this, we will also include seating arrangements for COVID-19 so that examinations can take place in offline mode. The website will make use of python, HTML and CSS. The prototype of the project will have GUI from in-built python library Tkinter.

The project will show how it improves the conduction of examination in colleges where they still use paper and ink for making a lot of lists which are extremely difficult to navigate through. Overall, this exploration offers an answer for stopping some test corridor cheating by orchestrating seats for enormous number of graduates and it likewise discovers the best blend of rooms to be allocated for the test to coordinate wonderful seating dependent on the room direction and size, number of understudies, separation of subjects.

Chapter 1.2

Formulation of Problem

Existing framework is exceptionally sluggish and wasteful. Report age is additionally not a simple assignment in the current circumstance. Additionally, if the report is created then computations are done physically that prompts more blunders. There is a ton of manual work engaged with current framework and mix-up in one detail can prompt wrong results. No appropriate assortment of necessities leads a massive issue for this framework. This framework is to improve manual work and furthermore more energy is used to do the planning. Along with this, no existing system has a seating plan for COVID-19 social distancing guidelines. Our framework aims to fix this problem as well.

There have been guidelines for social distancing seating even in trains and airplanes.

As per the guidelines issued by the Ministry of Civil Aviation, passengers are advised to not plan journey, in case the body temperature is high or they are symptomatic of Covid-19. They have to register on the Aarogya Setu App before starting a journey and always wear a face mask and hand gloves. Passengers have to maintain social distancing at all times.

Passenger shall have to do a web check-in and carry a print of boarding card from home. As they arrive at the airport, the guidelines state, keep the phone handy with travel documents like flight ticket, boarding pass and Identity proof for entry verification by a security officer standing behind a glass shield. After document verification, they have to follow the floor marking for baggage drop counter. Thermal scanners are installed to monitor the temperature of all passengers. Floor markings have been done at the check-in areas as CUSS Kiosk Machines to ensure social distancing. After reaching the airlines counter, they have to collect boarding pass and luggage tag touch-free. Passengers have to maintain social distancing - at least 4 feet gap from their

fellow passengers and airport staff members during the security check procedure.

They have to follow the seating arrangement markings as per social distancing norms in the waiting area.

Some food kiosks are open to serve at the waiting area with proper hygiene and social distancing. Food can also be delivered at the seat in the waiting area. The passengers are advised to take special precautions of social distancing while using restrooms.

High contact surfaces such as desks, chairs, elevators, railings, CUSS, trolleys, handles, trays and baggage belts are being sanitized regularly. During the lockdown, the Airports Authority of India has made arrangements at all airports to ensure safety of passengers amid the Covid-19 outbreak.

We have used this as an area of inspiration while coming up with the idea of solving covid based seating related problem.

Chapter 1.2.1

Tools and Technology used

The Project centers on seating understudies sequentially as indicated by their enlistment number in the lines and segments of empty rooms. This paper is created in **Python** utilizing **Jupyter Notebook/Visual Studio**. It includes fundamental capacities of Python language, for example, multi- dimensional exhibit, grids, capacities and code reusability.

This is utilized to designate seats to understudies naturally during an assessment simply by entering the quantity of rooms. This saves time and blunders if every understudy is physically designated a seat number in the framework. This ends up being an exceptionally productive and successful strategy for assessment arranging.

IDE is a product application that gives complete offices to software engineers for programming improvement. An IDE regularly comprises of a source code proofreader, assemble mechanization devices and a debugger. Generally, present day IDEs offer insightful code culmination highlights.

SQL is an amazing information base and it requires restricted projects and is utilized as back end. It upholds GUI and more application is created by help this server. Assortment of tables which holds the information is called data set. An amateur can make their own data set by click landing page. Here we have used a sorting algorithm to show outputs.

Sorting algorithms are a set of instructions that take an array or list as an input and arrange the items into a particular order. Sorts are most commonly in numerical or a form of alphabetical (called lexicographical) order, and can be in ascending (A-Z, 0-9) or descending (Z-A, 9-0) order.

Since sorting can often reduce the complexity of a problem, it is an important algorithm in Computer Science. These algorithms have direct applications in

searching algorithms, database algorithms, divide and conquer methods, data structure algorithms, and many more.

Trade-Offs of Algorithms

When using different algorithms some questions have to be asked. How big is the collection being sorted? How much memory is at disposal to be used? Does the collection need to grow? The answers to these questions may determine what algorithm is going to work best for the situation. Some algorithms like merge sort may need a lot of space to run, while insertion sort is not always the fastest but it doesn't require many resources to run. You should determine what the requirements of the system are and its limitations before deciding what algorithm to use.

Some of the most common sorting algorithms are:

Selection Sort

Bubble Sort

Insertion Sort

Merge Sort

Quick Sort

Heap Sort

Counting Sort

Radix Sort

Bucket Sort

But before we get into each of these, let's learn a bit more about what makes classifies a sorting algorithm.

Classification of a Sorting Algorithm

Sorting algorithms can be categorized based on the following parameters:

Based on Number of Swaps or Inversion This is the number of times the algorithm swaps elements to sort the input. Selection Sort requires the minimum number of swaps.

Based on Number of Comparisons This is the number of times the algorithm compares elements to sort the input. Using Big-O notation, the sorting algorithm examples listed above require at least $O(n \log n)$ comparisons in the best case and $O(n^2)$ comparisons in the worst case for most of the outputs.

Based on Recursion or Non-Recursion Some sorting algorithms, such as Quick Sort, use recursive techniques to sort the input. Other sorting algorithms, such as Selection Sort or Insertion Sort, use non-recursive techniques. Finally, some sorting algorithm, such as Merge Sort, make use of both recursive as well as non-recursive techniques to sort the input.

Based on Stability Sorting algorithms are said to be stable if the algorithm maintains the relative order of elements with equal keys. In other words, two equivalent elements remain in the same order in the sorted output as they were in the input. Insertion sort, Merge Sort, and Bubble Sort are stable Heap Sort and Quick Sort are not stable.

Based on Extra Space Requirement Sorting algorithms are said to be in place if they require a constant $O(1)$ extra space for sorting.

Insertion sort and Quick-sort are in place sort as we move the elements about the pivot and do not actually use a separate array which is NOT the case in merge sort where the size of the input must be allocated beforehand to store the output

during the sort.

Merge Sort is an example of out place sort as it require extra memory space for its operations.

The SQL database has long been the tried and true workhorse of the backend enterprise and at the heart of everything we do in this electronic age. SQL was created in the early 1970s at IBM as method of accessing IBM's System R database system.

The usefulness of being able to access several records from a single command that does not require specifying how to reach a given record was immediately recognized by the computing world. It was rapidly adopted as the basic query language for other relation database management systems, or RDBMS such as IBM's DB2 and in 1979 Relational Software Inc.'s (now known as Oracle Software) Oracle V2 database server for Vax systems. Eventually in 1986 SQL was adopted by ANSI and ISO standards organizations paving the way for Microsoft SQL Server and the various open source databases we have on the market today.

The RDBMS that we used today rely on SQL as the engine that allows us to perform all the operations required to create, retrieve, update, and delete data as needed. From an open source perspective these RDBMSs include MySQL, MariaDB, and PostgreSQL as the most commonly used open source RDBMS in production today. Many Fortune 100 companies across several different business sectors including financial, retail, healthcare, and others have turned to these open source alternatives to drastically lower their total cost of ownership when compared to pay-for-play offerings, such as Oracle Database server and Microsoft SQL Server.

A SQL database can:

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

Using SQL in a website:

To build a web site that shows data from a database, you will need:

- An RDBMS database program (i.e. MS Access, SQL Server, MySQL)
- To use a server-side scripting language, like PHP or ASP
- To use SQL to get the data you want
- To use HTML / CSS to style the page

Hardware Requirements

- Processor: Intel Dual Core
- Hard Disk: Minimum of 160 GB
- RAM: 4GB or more
- Integrated GPU

Software Requirements

- Operating System: Windows 8.1/10
- Browser: Preferably Google Chrome
- Language: Python, SQL
- Editor: Jupyter Notebook/VS Code

Chapter 2

Literature Survey

For the literature survey, we have done inspection of the existing automated systems and found research gaps. We have then tried to fix these research gaps in our project.

In Alam, A. F. (2016). Automatic seating arrangement tool for examinations in universities/colleges. *International Journal of Engineering Applied Sciences and Technology (IJEAST)*, 1(4), 2455-2143, the whole project has been developed in C/C++ so it lacks a interactive website or platform and it doesn't work well for large datasets.

Then in Aravinth, S., Pavithra, G., Myvizhimalar, M., Divya, D., & Rathinakrithika, M. (2014). Exam Hall Seating Arrangement System Using PHP. *International Journal of Innovative Research in Technology*, 1(11), 258-261, every student has to login to the website and register for their name to be in the seating plan. This is not centralized and very inconvenient for everybody.

The environment of the school building and the classrooms play a very important role in a student's day. The following review of literature reveals some of the research that has been done to show which areas in the school and classroom environments are the most beneficial towards the behavior and education of the students.

To better understand the different areas that are of importance to this topic, the review of literature has been divided into seven sections. School building environment, classroom environment, seating preference, classroom spatial needs, seating arrangement, the classroom environment and special needs students, and case studies. Student's degree of territoriality based on gender and seat preference in different types of classroom arrangements was studied by Kaya and Burgess. The types of classroom arrangements included rows of tablet-arm chairs, U-shaped, clusters, and rows of tables with individual chairs. The study

was carried out through a survey at a large public institution in the southeast region of the United States. Their results indicated that students who preferred seats at the end of rows of tables with individual chairs and tabletarm chair arrangement had higher scores on claiming a particular seat than those who preferred middle seats in a row. In rows of tables with individual chair arrangement, students who preferred seats at the end of rows also had more need to define their own territory than students who preferred middle seats in a row. No significant results were found in the U-shaped and cluster layouts. Females had higher scores on claiming a classroom particular seat than males regardless of seating arrangement.

This study also found that in classroom settings where students are required or elect to bring several items to class, for example, a backpack, jacket, and handbag, in addition to notebooks and textbooks, defining one's own territory may become increasingly important as a means of comfortably engaging in active learning. These studies were done in a college level classroom, and may be contradictory if the same studies were done at the elementary. The views on the arrangement of classrooms and seating arrangements have changed over the past few decades. In the 70's the classroom setting was mostly focused on the traditional row set-up classroom. Weinstein found that there was a tremendous surge of interest in determining the influence of the school setting on students. Based on the above research surveys, we have chosen to use Python as the base language for the project and SQL for data management.

Project Design

The relationships between and among the actors and the use cases of System:

Super Admin Entity: Use cases of Super Admin are Manage Student, Manage Block, Manage Room, Manage Exam, Manage Seat, Manage Teacher, Manage Users and Full Exam Seating Management System Operations

System User Entity: Use cases of System User are Manage Student, Manage Block, Manage Room, Manage Exam, Manage Seat, Manage Teacher

Examiner Entity: Use cases of Examiner are Check Seats, Allot Seats, Manage Candidates

Student Entity: Use cases of Student are Search Seats, Search Rooms, Search Exam

Administrator: In administrator panel, it having a unique id and password so that whenever it want to access the data that time admin must have to enter valid user id and password. After an authenticable id and password authentication process takes place. So that admin can fill up all the requirements regarding to seats allocation. After allocating a specific classrooms, buildings, exam dates and paper in a given section admin will upload a data in a server so that students will access the data.

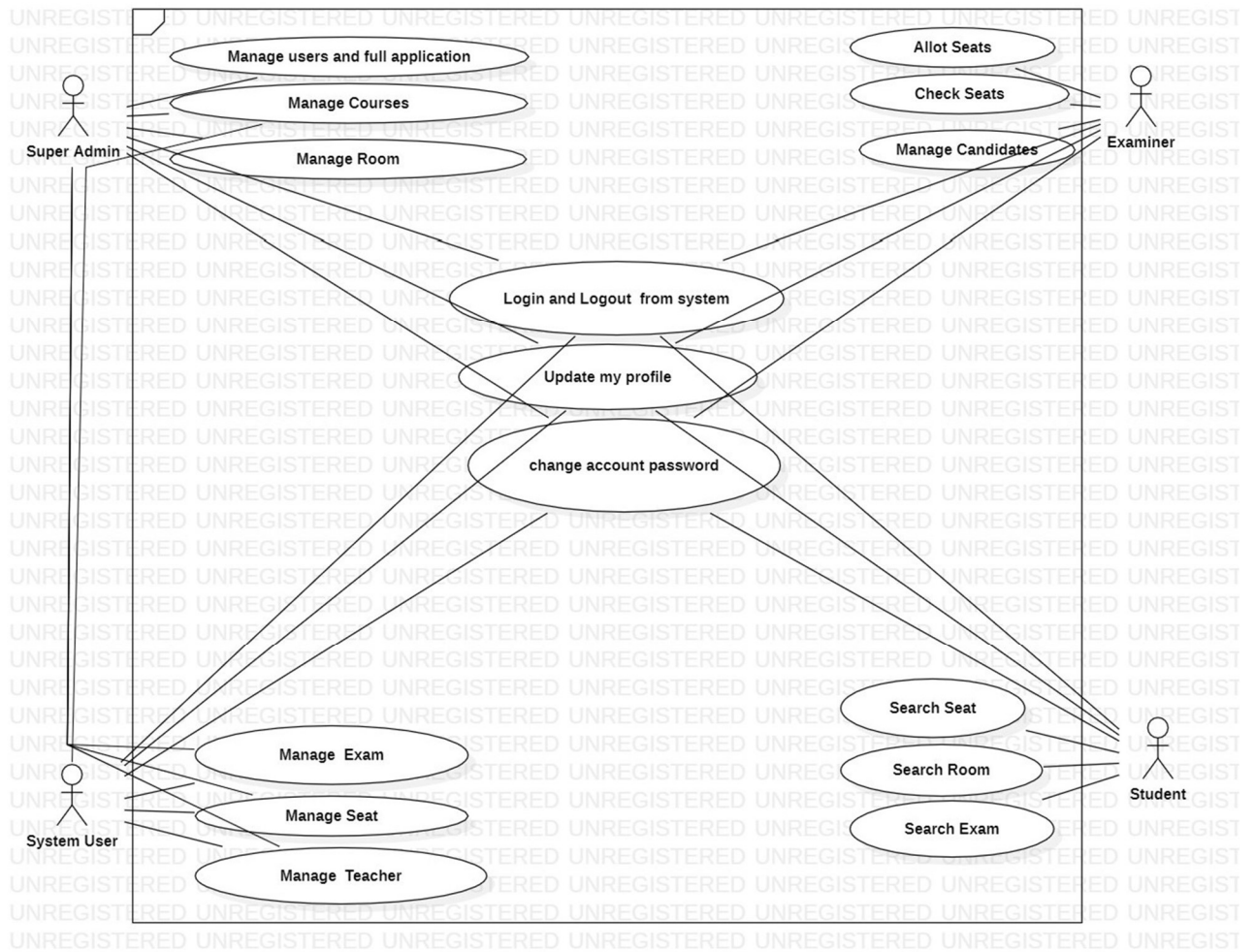
Step 1: Two types of data sets are needed for exam seating, one is room information having room id, size of columns and rows; another is exam information with subject id and number of students. By processing room and exam information it will calculate total number of seats and students.

Step 2: When the amount of extra seats is greater than any room capacity, then it may be possible of having one or more extra room. In this step it will search those rooms, which will be kept empty at the time of exam.

Step 3: Our main concept is to distribute column to the subject so at first it needs to calculate the number of columns and the capacity or number of rows for each column. To prevent copying from other it tries to ensure at least one column distance among students having same query set. That is why it will also calculate maximum usable column and its capacity for any subject.

Step 4: Then it will allot columns to the subjects. By using a combination of column size (row) we can ascertain the set of columns which will be assigned to subjects. After that it will provide a set of columns for each subject. In best case extra seats are not required, but in worst case it may require to assign some extra seats to subject for getting a set of column for exam seating.

Step 5: It have subject having different column size and number. We have also room identity which is represented by column number and row number as column size. So the common attribute is column size. By using this it assigns room and column number to the subject by maintaining one column distance between the columns having the same subject. After completing the column and room distribution it will provide a complete seating solution for the exam.



USE CASE DIAGRAM OF PROJECT

Architecture Diagram

There are many kinds of architecture diagrams, like a software architecture diagram, system architecture diagram, application architecture diagram, security architecture diagram, etc.

Levels of Abstractions of Architecture Diagrams

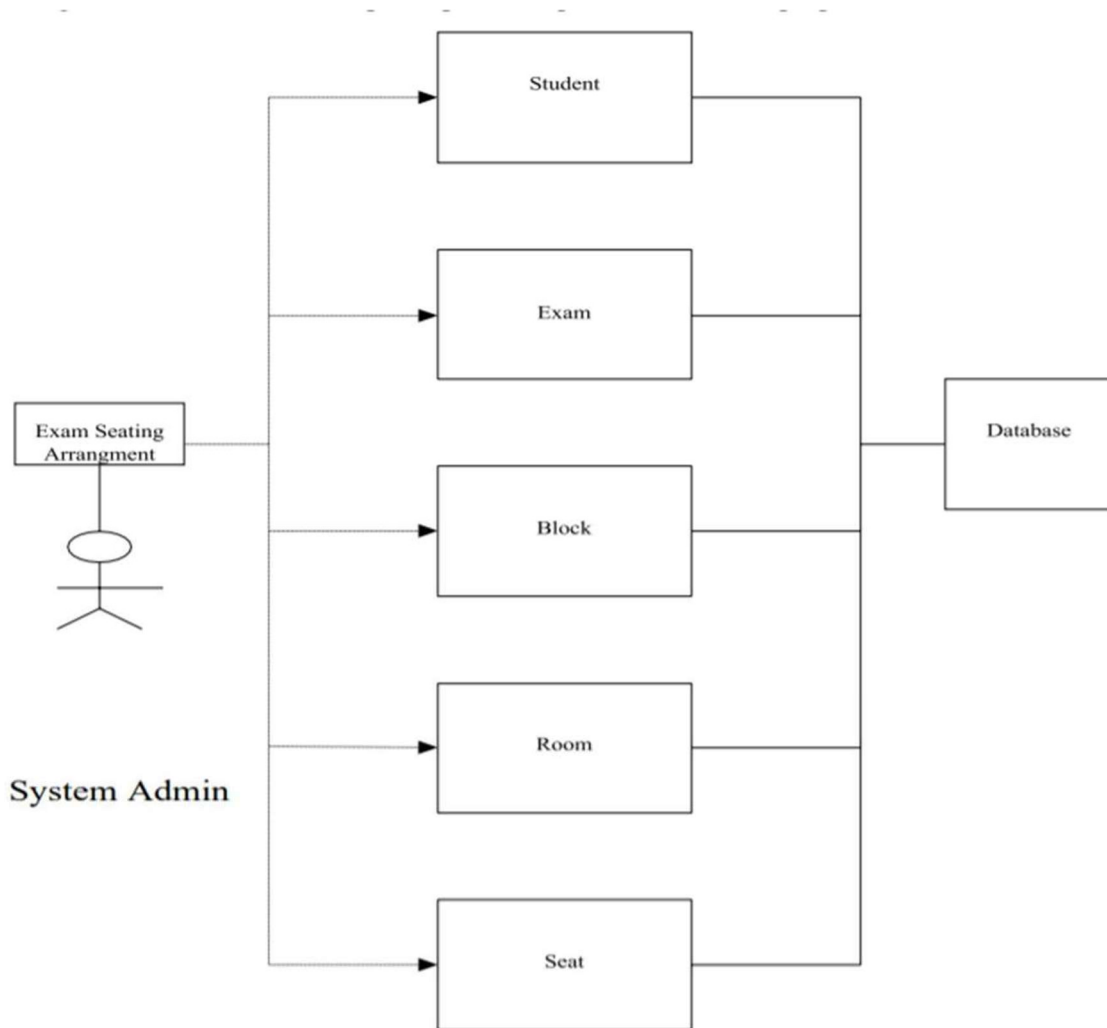
Architecture is a coherent set of concepts for a structure. These concepts are often visualized at four levels of abstraction. These are:

- Conceptual Level - showing an overview of concepts
- Logical Level - showing a logical design of one or more concepts, containing at least the key elements of concepts and showing the principles of the concepts (i.e. how the concepts work).
- Physical Level - showing a component design depicting the elements
- Implementational Level - showing the vendors and products with which the components will be implemented.

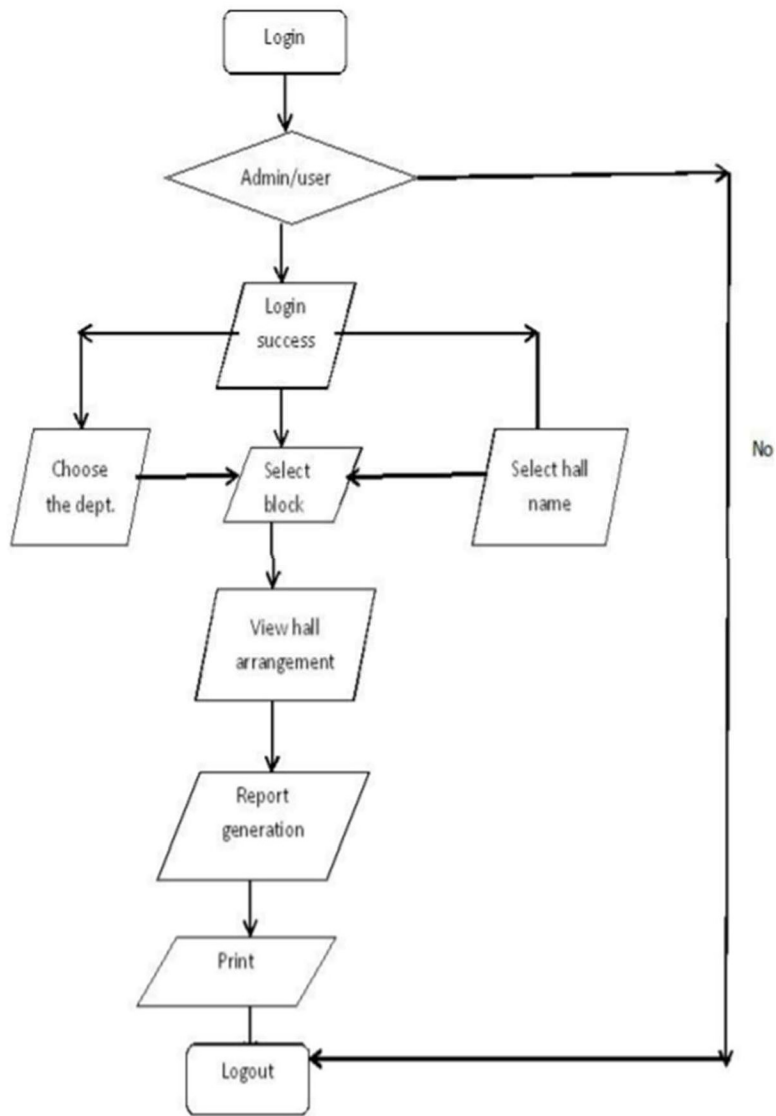
All the visualizations on this page are logical level architecture diagrams. Most often an architecture diagram contains a combination of logical and physical aspects, meaning that logical elements and physical components both are used in the visualization.

Organization building is the applied arrangement that portrays the construction and lead of a structure. A plan depiction be regular demarcation of a arrangement, sifted through with the end goal that supports contemplating the assistant properties of the method. It describes the organization parts or building deters to facilitate force participate near put to exist the all-purpose constitution.

The Figure provides the formation Architecture which contains admin who will manage the student details, exam details, block details, room details, and seat each and every student. Each and Every such detail will be stored in the database.



ARCHITECTUE DIAGRAM OF PROJECT



DATA FLOW DIAGRAM OF PROJECT

Chapter 3

Working of Project

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

The Tkinter framework provides Python users with a simple way to create GUI elements using the widgets found in the Tk toolkit. Tk widgets can be used to construct buttons, menus, data fields, etc. in a Python application. Once created, these graphical elements can be associated with or interact with features, functionality,

methods, data or even other widgets.

For example, a button widget can accept mouse clicks, and can also be programmed to perform some kind of action, such as exiting the application.

Each part has interesting functionalities in which it will perform their particular assignment. While playing out a task administrator having a full power to get to and display the information. On the SQL Login, it has an ID and Password so that at whatever point there is a need to get to the information then at that time the admin must enter substantial client ID and Password. After an authenticable ID and key validation process takes place. So that administrator can top off all the requirements regarding seats distribution. Subsequent to allotting specific classrooms, structures, test dates and paper in a given area administrator will transfer the information in a server so that students can access the information.

In the following example, Tkinter creates a Tk application frame and a button widget that exits the application when clicked.

```
from tkinter import *  
  
tk = Tk()  
frame = Frame(tk, borderwidth=2)  
frame.pack(fill=BOTH, expand=1)  
label = Label(frame, text="Button Example")  
label.pack(fill=X, expand=1)  
  
button = Button(frame, text="Exit", command=tk.destroy)  
button.pack(side=BOTTOM)  
tk.mainloop()
```

The best way to get the latest version of Tkinter is to install Python 3.7 or later. But Tkinter can also be downloaded and installed as part of any standard Python 3 installation.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps –

Import the Tkinter module.

Create the GUI application main window.

Add one or more of the above-mentioned widgets to the GUI application.

Enter the main event loop to take action against each event triggered by the user.

The Module description is as follows:

- Upload Database: Datasheet, Exam Data, Room Capacity and Teacher List is uploaded through this module into the database.
- While uploading datasheet, there is a provision to select odd or even semester and morning or evening shift through the program interface itself.
- SQL Login: includes Username, Password and Database Name.
- Teacher on Leave: A list of teachers is fed to the database and those individuals are not selected for invigilation.
- Arrangement: The system outputs the arrangement plan in an excel file.

Seating with Covid-19 in Mind

Coronavirus disease (COVID-19) is an infectious disease caused by the SARS-CoV-2 virus. Most people infected with the virus will experience mild to moderate respiratory illness and recover without requiring special treatment. However, some will become seriously ill and require medical attention. Older people and those with underlying medical conditions like cardiovascular disease, diabetes, chronic respiratory disease, or cancer are more likely to develop serious illness. Anyone can

get sick with COVID-19 and become seriously ill or die at any age.

The best way to prevent and slow down transmission is to be well informed about the disease and how the virus spreads. Protect yourself and others from infection by staying at least 1 metre apart from others, wearing a properly fitted mask, and washing your hands or using an alcohol-based rub frequently. Get vaccinated when it's your turn and follow local guidance.

The virus can spread from an infected person's mouth or nose in small liquid particles when they cough, sneeze, speak, sing or breathe. These particles range from larger respiratory droplets to smaller aerosols. It is important to practice respiratory etiquette, for example by coughing into a flexed elbow, and to stay home and self-isolate until you recover if you feel unwell.

Most people with COVID-19 have mild symptoms, but some people become severely ill. Some people including those with minor or no symptoms may suffer from post-COVID conditions — or “long COVID”. Older adults and people who have certain underlying medical conditions are at increased risk of severe illness from COVID-19. Hundreds of thousands of people have died from COVID-19 in the United States. Vaccines against COVID-19 are safe and effective. Vaccines teach our immune system to fight the virus that causes COVID-19.

Social distancing is a non-pharmaceutical infection prevention and control intervention implemented to avoid/decrease contact between those who are infected with a disease causing pathogen and those who are not, so as to stop or slow down the rate and extent of disease transmission in a community. This eventually leads to decrease in spread, morbidity and mortality due to the disease.

In addition to the proposed interventions, the State/UT Governments may prescribe such other measures as they consider necessary.

All these proposed interventions shall be in force till 31st of March, 2020. They will

be reviewed as per the evolving situation.

The following interventions are proposed:

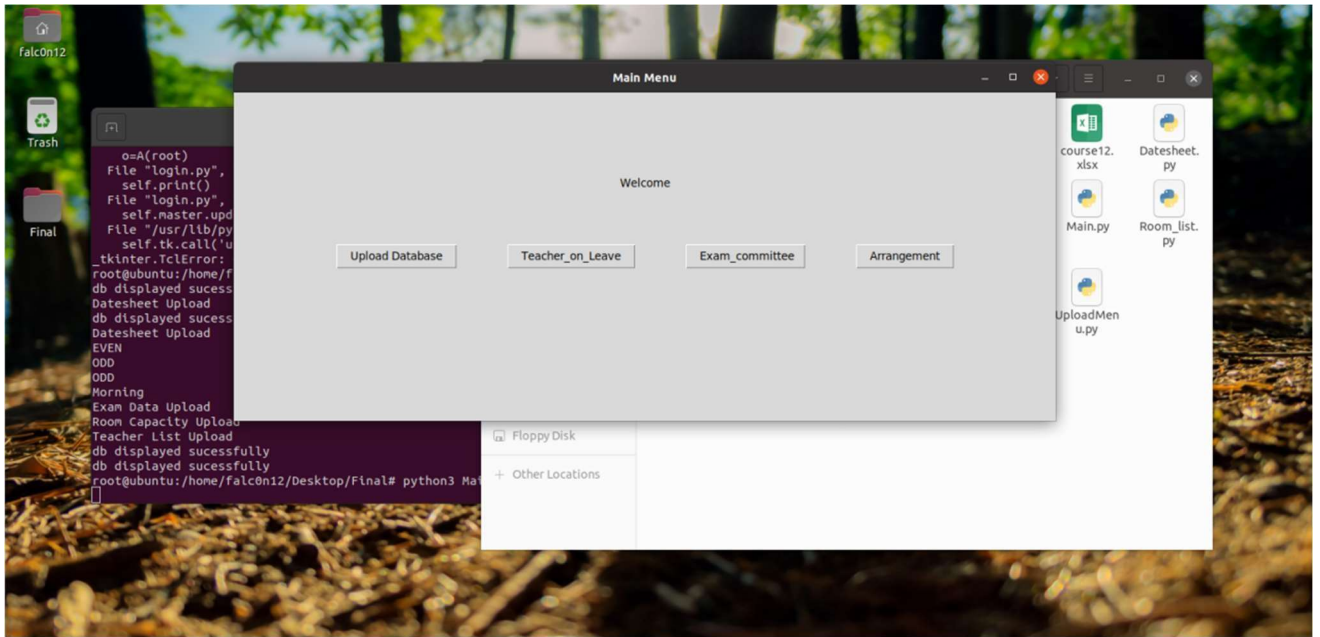
1. Closure of all educational establishments (schools, universities etc), gyms, museums, cultural and social centres, swimming pools and theatres. Students should be advised to stay at home. Online education to be promoted.
2. Possibility of postponing exams may be explored. Ongoing exams to be conducted only after ensuring physical distance of one meter amongst students.
3. Encourage private sector organizations/employers to allow employees to work from home wherever feasible.
4. Meetings, as far as feasible, shall be done through video conferences. Minimize or reschedule meetings involving large number of people unless necessary.
5. Restaurants to ensure handwashing protocol and proper cleanliness of frequently touched surfaces. Ensure physical distancing (minimum 1metre) between tables; encourage open air seating where practical with adequate distancing.
6. Keep already planned weddings to a limited gathering, postpone all non-essential social and cultural gatherings.
7. Local authorities to have a dialogue with organizers of sporting events and competitions involving large gatherings and they may be advised to postpone such events.
8. Local authorities to have a dialogue with opinion leaders and religious leaders to regulate mass gatherings and should ensure no overcrowding/at least one metre distance between people.
9. Local authorities to have meeting with traders associations and other stakeholders to regulate hours, exhibit Do's and Don'ts and take up a communication drive in market places like sabzi mandi, anaj mandi, bus depots, railway stations, post-offices etc., where essential services are provided.
10. All commercial activities must keep a distance of one meter between customers. Measures to reduce peak hour crowding in markets.
11. Non-essential travel should be avoided. Buses, Trains and aeroplanes to maximize

Covid based Seating Arrangement:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Date-2019-05-07							Session-EVENING										
2				Room no-67														
3		Row 1	Row 1	Row 2	Row 2	Row 3	Row 3											
4		PS - I	BOTANY - I	PS - I	BOTANY - I	PS - I	BOTANY - I											
5		16001582080	17001555968	18001582080	18001555456	18001582080	18001555456											
6																		
7		17001581568	17001555968	18001582080	18001555456	18001582080	18001555456											
8																		
9		17001581568	18001555456	18001582080	18001555456	18001582080	18001555456											
10																		
11		17001581568	18001555456	18001582080	18001555456	18001582080	18001555456											
12																		
13		17001582592	18001555456	18001582080	18001555456	18001582080	18001555456											
14																		
15		17001582592	18001555456	18001582080	18001555456	18001582080	18001555456											
16																		
17		17001582592	18001555456	18001582080	18001555456	18001582080	18001555456											
18																		
19		17001582592	18001555456	18001582080	18001555456	18001582080	18001555456											
20																		
21		17001582592	18001555456	18001582080	18001555456	18001582080	18001555456											
22																		
23																		
24		Left Side -	PS-I															
25		Right Side -	BOTANY-I															
26		On duty -	Ms. Sangeeta Relan ,Dr. Manisha Verma ,Dr. Rakesh Roshan															
27																		
28		Course	Total															
29		PS-I		27														
30		BOTANY-I		27														

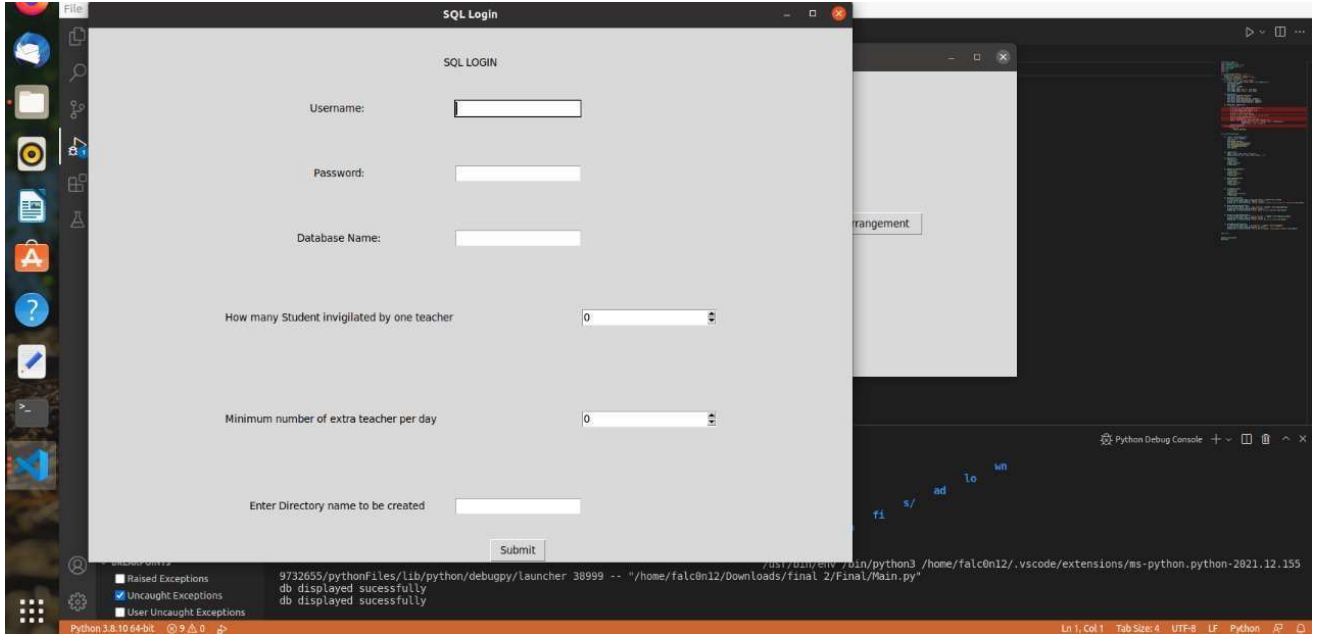
Exam Committee:

The screenshot displays a Python application in Visual Studio Code. The application window, titled 'Main Menu', features a 'Welcome' message and three buttons: 'Upload Database', 'Exam_committee', and 'Test'. A modal dialog box titled 'Select Members of Exam Committee' is open, showing a list of names with checkboxes. The 'Exam Committee' list includes: Mr. Abhay Pratap Singh, Ms. Hanisha Bhagat, Mr. Gurjinder Singh, Mr. Yask Saln, Mr. Ankit Singh, Mr. Jitender Goel, Ms. Hemant Yadav, Ms. Sneh Sagar, Mr. Vineet Girdharwal, Mr. Sanjay Kumar, Dr. Neelgagan Singh, Dr. Rahul Dev, Dr. Sweety Shrimali, Dr. Aparna Sharma, Dr. Rakesh Kumar Sonker, Dr. Satya Prakash Yadav, Dr. Siddhartha, Dr. Pradeep Kr. Gupta, Dr. Shashi Shekhar, Dr. Sanjay Kumar, Dr. Neelakshi N.K. Borah, Dr. Ambika, Dr. Rohtash Singh, and Dr. V. Bhasker Raj. The background shows the application's code and a terminal window with output.

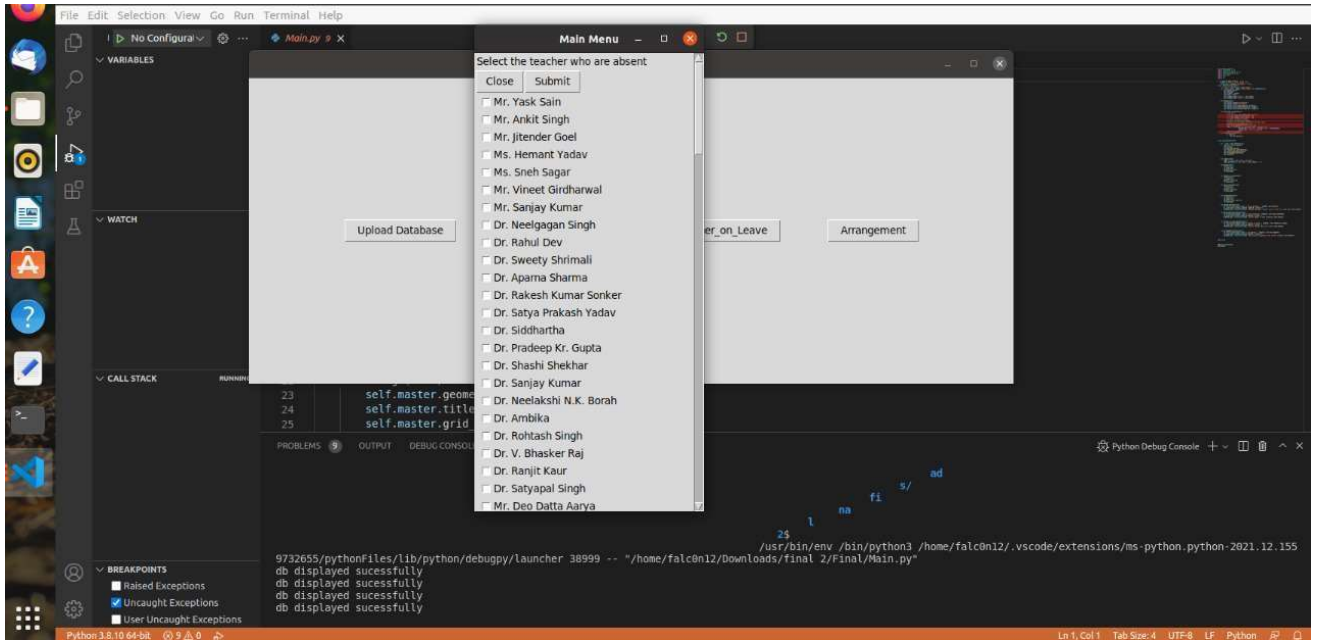


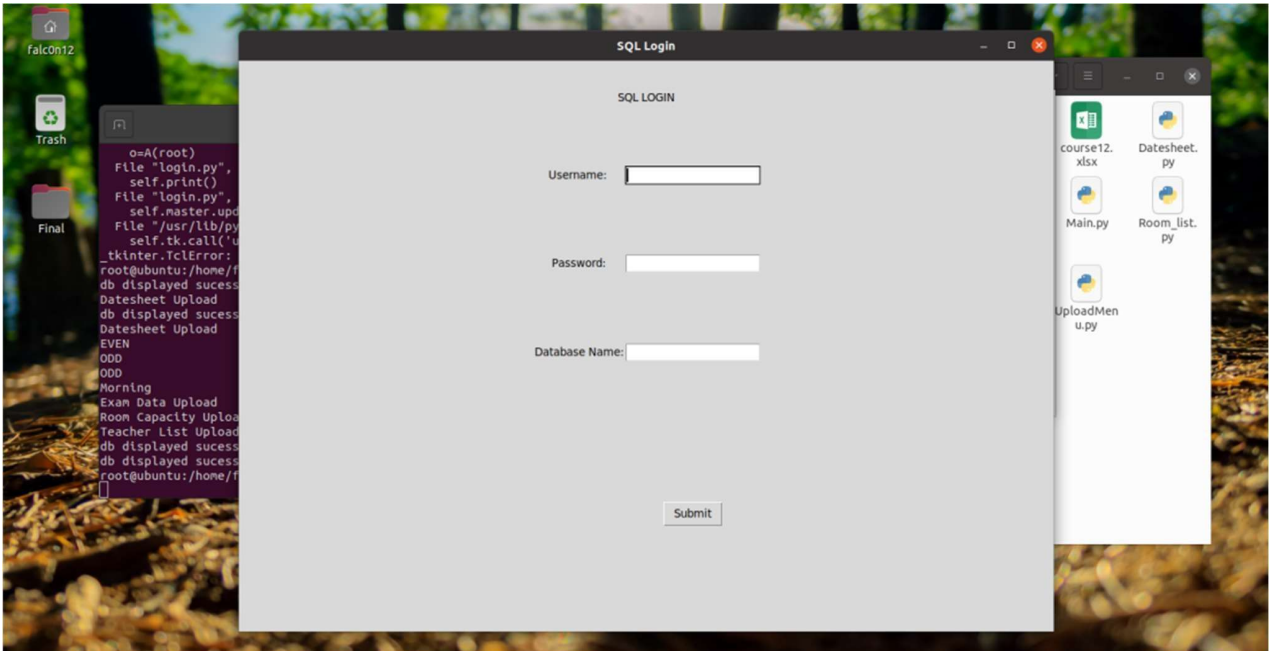
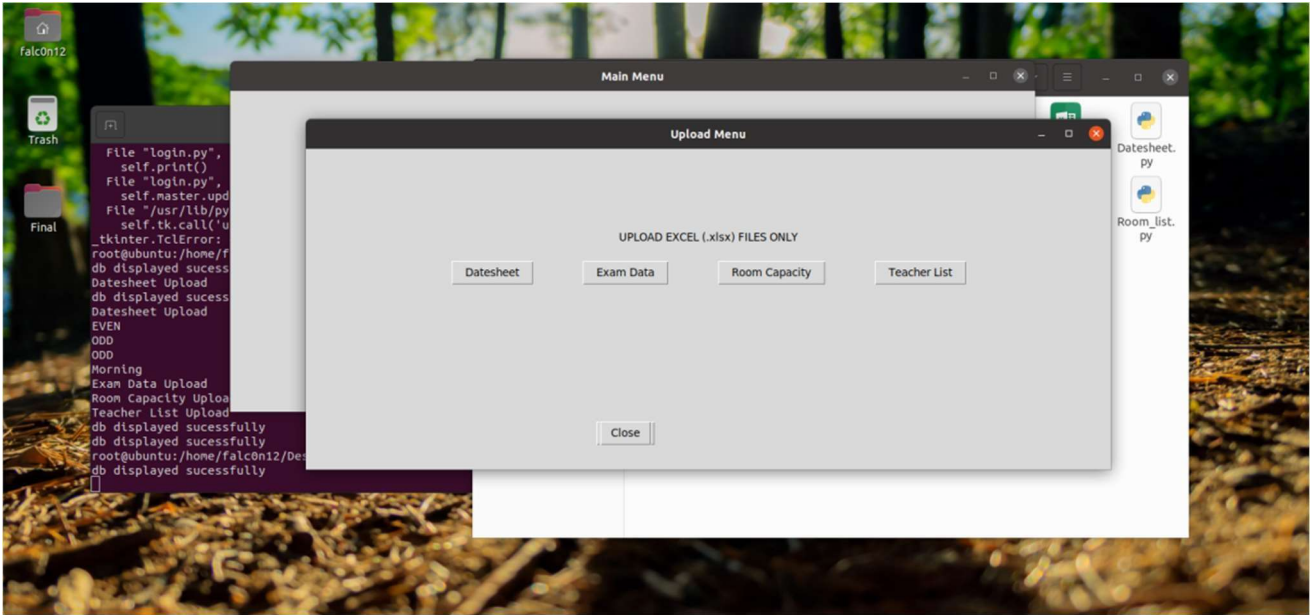
Normal Seating:

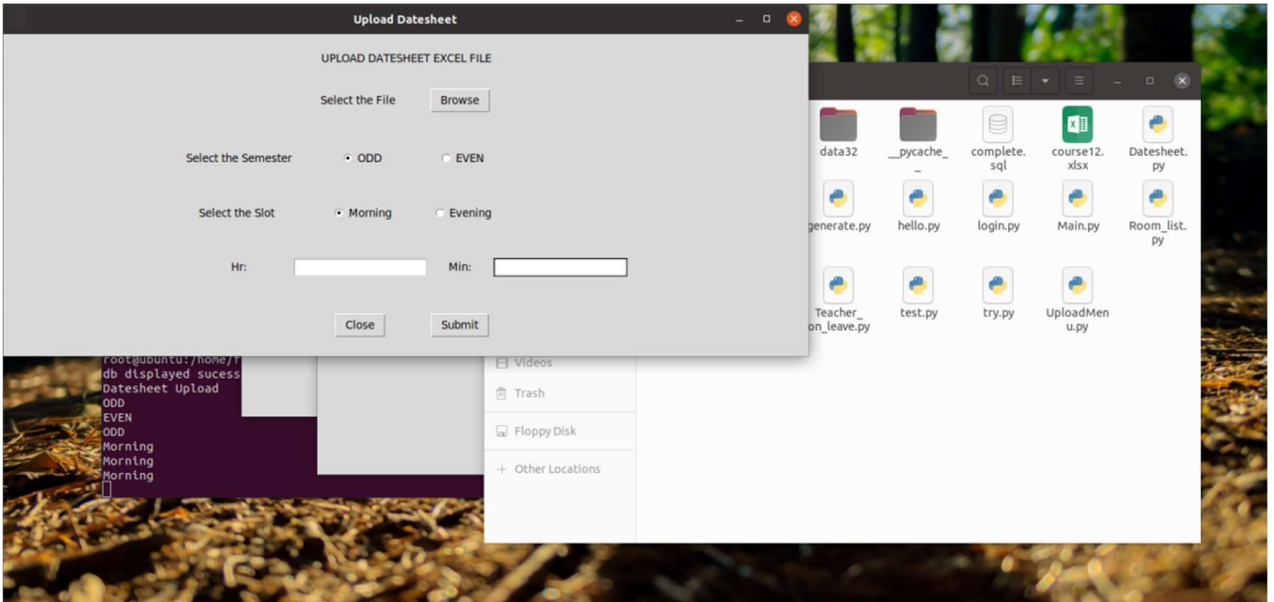
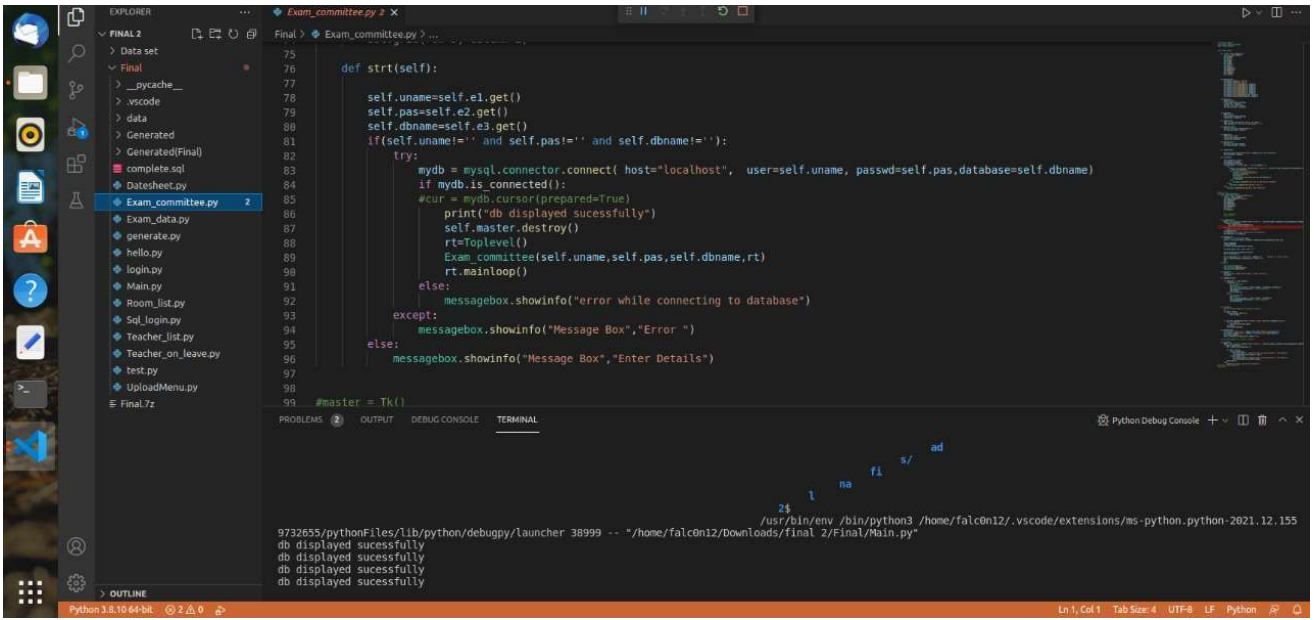
Date-2019-05-07									
A	B	C	D	E	F	G	H	I	J
Date-2019-05-07								Session-EVENING	
				Room no-67					
	Row 1	Row 1	Row 2	Row 2	Row 3	Row 3			
	PS - I	BOTANY - I	PS - I	BOTANY - I	PS - I	BOTANY - I			
	16001582080	17001555968	18001582080	18001555456	18001582080	18001555456			
	17001581568	17001555968	18001582080	18001555456	18001582080	18001555456			
	17001581568	18001555456	18001582080	18001555456	18001582080	18001555456			
	17001581568	18001555456	18001582080	18001555456	18001582080	18001555456			
	17001582592	18001555456	18001582080	18001555456	18001582080	18001555456			
	17001582592	18001555456	18001582080	18001555456	18001582080	18001555456			
	17001582592	18001555456	18001582080	18001555456	18001582080	18001555456			
	17001582592	18001555456	18001582080	18001555456	18001582080	18001555456			
	17001582592	18001555456	18001582080	18001555456	18001582080	18001555456			
	Left Side -	PS-I							
	Right Side -	BOTANY-I							
	On duty -	Ms. Sangeeta Relan ,Dr. Manisha Verma ,Dr. Rakesh Roshan							
	Course	Total							
	PS-I		27						
	BOTANY-I		27						



Teacher On-Leave:







```
falcon12@ubuntu: ~  
mysql  
+-----+  
| mysql |  
| performance_schema |  
| sys |  
+-----+  
5 rows in set (0.01 sec)  
  
mysql> use hhh;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
mysql> show tables;  
+-----+  
| Tables_in_hhh |  
+-----+  
| course |  
| mysamp |  
| room_data |  
| student_details |  
| teachers |  
+-----+  
5 rows in set (0.00 sec)  
  
mysql>
```

```
complete.sql generate.py Main.py Room_list.py  
1  
2 from tkinter import *  
3 import UploadMenu as um  
4 import Teacher_on_leave as tol  
5 import Exam_committee as ec  
6 import Sel_login  
7 import sys  
8 import os  
9  
10 if os.environ.get('DISPLAY','') == '':  
11     print("no display found. Using :0.0")  
12     os.environ.__setitem__('DISPLAY', ':0.0')  
13 class createToolTip(object):  
14     '''create a tooltip for a given widget'''  
15     def __init__(self, widget, text='widget info',master=None):  
16         self.master=master  
17         self.design()  
18         self.widget = widget  
19         self.text = text  
20         self.widget.bind("<Enter>", self.enter)  
21         self.widget.bind("<Leave>", self.close)  
22  
23     def design(self):  
24         self.master.geometry("1000x400")  
25         self.master.title("Main Menu")  
26         self.master.grid_rowconfigure(0, weight=1)  
27         self.master.grid_columnconfigure(0, weight=1)  
28         self.master.grid_columnconfigure(5, weight=1)  
29  
30     def enter(self, event=None):  
31         x = y = 0  
32         x, y, cx, cy = self.widget.bbox("insert")  
33         x += self.widget.winfo_rootx() + 25  
34         y += self.widget.winfo_rooty() + 20  
35         # creates a toplevel window  
36         self.tw = Toplevel(self.widget)  
37         # Leaves only the Label and removes the app window  
38         self.tw.wm_overrideredirect(True)  
39         self.tw.wm_geometry("+%d+%d" % (x, y))  
40         label = Label(self.tw, text=self.text, justify='left',  
41                     background='light blue', relief='solid', borderwidth=1,  
42                     font=("times", "10", "normal"))
```

Python Tab Width: 8 Ln 33, Col 52

Database:

```
falcon12@ubuntu: ~  
mysql> show tables;  
+-----+  
course  
mysamp  
room_data  
student_details  
teachers  
+-----+  
5 rows in set (0.00 sec)  
  
mysql> select * from teachers;  
+-----+  
tid | Name | Mobile_no | Email | Department | Status | availability  
+-----+  
1 | Dr. Sunita Hooda | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
2 | Dr. Manita Bhatia | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
3 | Dr. Sanjeeta Rani | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
4 | Ms. Sangeeta Relan | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
5 | Dr. Manisha Verna | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
6 | Ms. Sadhna Gupta | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
7 | Ms. Anupama | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
8 | Dr. Vatsla Kohli | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
9 | Dr. Poonam A. Kant | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
10 | Dr. Sarita Kumar | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
11 | Dr. Anil Garg | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
12 | Dr. Shalu Dhanda | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
13 | Dr. Pooja Bhagat | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
14 | Dr. Anju Agarwal | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
15 | Dr. Sharanjit Kaur | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
16 | Dr. Surinder Kaur | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
17 | Dr. Anita Garg | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
18 | Dr. Anupama Shukla | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
19 | Dr. Sunita Jetly | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
20 | Dr. Gagan Dhawan | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
21 | Dr. Archana Pandey | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
22 | Dr. Shalu Sachdeva | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
23 | Dr. Rashmi Thukral | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
24 | Dr. Anil Garg | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
25 | Mr. Gaurav Sharma | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
26 | Dr. Ravi Toteja | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
27 | Dr. Monica Misra | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
28 | Dr. Seema Makhtia | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
29 | Dr. Charu Khosla Gupta | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
30 | Dr. Manisha Jain | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
31 | Dr. Neeti Misra | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
32 | Dr. Vibha Gaur | 9560923136 | hello@gmail.com | Computer Science | Permanent | I  
+-----+
```

```
falcon12@ubuntu: ~  
mysql> select * from teachers;  
+-----+  
tid | Name | Mobile_no | Email | Department | Status | availability  
+-----+  
85 | Ms. Ntshu Singh | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
86 | Ms. Gunjan Rani | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
87 | Ms. Vandita Grover | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
88 | Ms. Gauri Ghat | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
89 | Dr. Anita Kumar | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
90 | Mr. Dinesh Kumar | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
91 | Dr. Monika Bhattacharya | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
92 | Dr. Jyoti Dhar Rakshita | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
93 | Dr. Neeti Goel | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
94 | Dr. Narayan Singh | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
95 | Mr. Brijendra Yadav | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
96 | Mr. Gurudatt Rao Anbedkar | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
97 | Mr. Dev Om | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
98 | Dr. Kalu Ram Meena | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
99 | Mr. Deo Datta Aarya | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
100 | Dr. Satyapal Singh | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
101 | Dr. Ranjit Kaur | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
102 | Dr. V. Bhasker Raj | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
103 | Dr. Rohtash Singh | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
104 | Dr. Ambika | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
105 | Dr. Neelakshi N.K. Borah | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
106 | Dr. Sanjay Kumar | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
107 | Dr. Shashi Shekhar | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
108 | Dr. Pradeep K. Gupta | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
109 | Dr. Sidhartha | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
110 | Dr. Satya Prakash Yadav | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
111 | Dr. Rakesh Kumar Sonker | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
112 | Dr. Aparna Sharma | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
113 | Dr. Suresh Shrinath | 9560923136 | hello@gmail.com | Computer Science | Adhoc | I  
114 | Dr. Rahul Dev | 9560924160 | hello@gmail.com | Computer Science | Adhoc | I  
115 | Dr. Neelgagan Singh | 9560924160 | hello@gmail.com | Computer Science | Adhoc | I  
116 | Mr. Sanjay Kumar | 9560924160 | hello@gmail.com | Computer Science | Adhoc | I  
117 | Mr. Vineet Giridarwal | 9560924160 | hello@gmail.com | Computer Science | Adhoc | I  
118 | Ms. Sneha Sagar | 9560924160 | hello@gmail.com | Computer Science | Adhoc | I  
119 | Ms. Hemant Yadav | 9560924160 | hello@gmail.com | Computer Science | Adhoc | I  
120 | Mr. Dittender Goel | 9560924160 | hello@gmail.com | Computer Science | Adhoc | I  
121 | Mr. Ankit Singh | 9560924160 | hello@gmail.com | Computer Science | Adhoc | I  
122 | Mr. Vask Satn | 9560924160 | hello@gmail.com | Computer Science | Adhoc | I  
123 | Mr. Gurjinder Singh | 9560924160 | hello@gmail.com | Computer Science | Adhoc | S  
124 | Ms. Manisha Bhagat | 9560924160 | hello@gmail.com | Computer Science | Adhoc | S  
125 | Mr. Abhay Pratap Singh | 9560924160 | hello@gmail.com | Computer Science | Adhoc | S  
+-----+  
125 rows in set (0.00 sec)  
  
mysql>
```

```

mysql> show tables;
+-----+
| Tables_in_hhh |
+-----+
| course        |
| mysamp        |
| room_data     |
| student_details |
| teachers      |
+-----+
5 rows in set (0.00 sec)

mysql> select * room_data;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'room_data' at line 1
mysql> select * from room_data;
+----+----+-----+----+
| Sno | RNo | Capacity | Row |
+----+----+-----+----+
| 1   | 67  | 54       | 3   |
| 2   | 68  | 54       | 3   |
| 3   | 69  | 54       | 3   |
| 4   | 61  | 36       | 3   |
| 5   | 60  | 36       | 3   |
| 6   | 64  | 42       | 3   |
| 7   | 65  | 42       | 3   |
| 8   | 33  | 32       | 4   |
| 9   | 34  | 32       | 4   |
| 10  | 35  | 32       | 4   |
| 11  | 36  | 32       | 4   |
| 12  | 37  | 56       | 4   |
| 13  | 39  | 32       | 4   |
| 14  | 47  | 32       | 4   |
| 15  | 48  | 32       | 4   |
| 16  | 49  | 32       | 4   |
| 17  | 50  | 32       | 4   |
| 18  | 53  | 48       | 4   |
| 19  | 54  | 42       | 3   |
| 20  | 55  | 24       | 4   |
| 21  | 56  | 38       | 3   |
| 22  | 57  | 48       | 4   |
| 23  | 58  | 36       | 3   |
+----+----+-----+----+
23 rows in set (0.00 sec)

mysql>

```

```

570 | I | II | 72182800 | ENVIRONMENTAL SCIENCE | 47 | 2019-05-07 | EVENING |
558 | I | II | 12035908 | READINGS IN INDIAN DIVERSITIES AND LITERARY MOVEMENT | 5 | 2019-05-10 | EVENING |
558 | I | II | 12555260 | FITNESS, WELLNESS AND NUTRITION | 7 | 2019-05-10 | EVENING |
558 | I | II | 32225200 | Mechanics | 9 | 2019-05-11 | EVENING |
558 | I | II | 32345200 | INTRODUCTION TO DATABASE SYSTEMS | 5 | 2019-05-11 | EVENING |
558 | I | II | 32355202 | LINEAR ALGEBRA | 23 | 2019-05-11 | EVENING |
558 | I | II | 32511204 | Semiconductor Devices | 46 | 2019-05-24 | MORNING |
558 | I | II | 32511204 | Applied Physics | 45 | 2019-05-22 | MORNING |
558 | I | II | 72182800 | ENVIRONMENTAL SCIENCE | 45 | 2019-05-07 | EVENING |
563 | I | II | 12275201 | INTRODUCTORY MACROECONOMICS | 5 | 2019-05-10 | EVENING |
563 | I | II | 12555260 | FITNESS, WELLNESS AND NUTRITION | 7 | 2019-05-10 | EVENING |
563 | I | II | 32175912 | CHEMISTRY OF S,P BLOCK ELEMENTS, STATES OF MATTER AND CHEMICAL KINETICS | 7 | 2019-05-11 | EVENING |
563 | I | II | 32225200 | Mechanics | 14 | 2019-05-11 | EVENING |
563 | I | II | 32345200 | INTRODUCTION TO DATABASE SYSTEMS | 21 | 2019-05-11 | EVENING |
563 | I | II | 32351200 | REAL ANALYSIS | 57 | 2019-05-24 | MORNING |
563 | I | II | 32351202 | Differential Equations | 56 | 2019-05-22 | MORNING |
563 | I | II | 72182800 | ENVIRONMENTAL SCIENCE | 53 | 2019-05-07 | EVENING |
567 | I | II | 12275201 | INTRODUCTORY MACROECONOMICS | 1 | 2019-05-10 | EVENING |
567 | I | II | 12555260 | FITNESS, WELLNESS AND NUTRITION | 1 | 2019-05-10 | EVENING |
567 | I | II | 32175902 | CHEMICAL ENERGETICS, EQUILIBRIA & FUNCTIONAL GROUP ORGANIC CHEMISTRY-I | 1 | 2019-05-11 | EVENING |
567 | I | II | 32175912 | CHEMISTRY OF S,P BLOCK ELEMENTS, STATES OF MATTER AND CHEMICAL KINETICS | 15 | 2019-05-11 | EVENING |
567 | I | II | 32221200 | Electricity and Magnetism | 59 | 2019-05-24 | MORNING |
567 | I | II | 32221202 | Waves and Optics | 60 | 2019-05-22 | MORNING |
567 | I | II | 32345200 | INTRODUCTION TO DATABASE SYSTEMS | 3 | 2019-05-11 | EVENING |
567 | I | II | 32355202 | LINEAR ALGEBRA | 40 | 2019-05-11 | EVENING |
567 | I | II | 72182800 | ENVIRONMENTAL SCIENCE | 57 | 2019-05-07 | EVENING |
569 | I | II | 12035908 | READINGS IN INDIAN DIVERSITIES AND LITERARY MOVEMENT | 17 | 2019-05-10 | EVENING |
569 | I | II | 12555260 | FITNESS, WELLNESS AND NUTRITION | 2 | 2019-05-10 | EVENING |
569 | I | II | 32165200 | Plant Ecology and Taxonomy | 1 | 2019-05-11 | EVENING |
569 | I | II | 32175912 | CHEMISTRY OF S,P BLOCK ELEMENTS, STATES OF MATTER AND CHEMICAL KINETICS | 26 | 2019-05-11 | EVENING |
569 | I | II | 32231200 | Non-chordates II: Coelomates | 46 | 2019-05-24 | MORNING |
569 | I | II | 32231202 | Cell Biology | 46 | 2019-05-22 | MORNING |
569 | I | II | 32345200 | INTRODUCTION TO DATABASE SYSTEMS | 1 | 2019-05-11 | EVENING |
569 | I | II | 32585902 | Biological Chemistry | 1 | 2019-05-11 | EVENING |
569 | I | II | 72182800 | ENVIRONMENTAL SCIENCE | 46 | 2019-05-07 | EVENING |
504 | I | II | 12035908 | READINGS IN INDIAN DIVERSITIES AND LITERARY MOVEMENT | 5 | 2019-05-10 | EVENING |
504 | I | II | 12275201 | INTRODUCTORY MACROECONOMICS | 93 | 2019-05-10 | EVENING |
504 | I | II | 12555260 | FITNESS, WELLNESS AND NUTRITION | 6 | 2019-05-10 | EVENING |
504 | I | II | 22411200 | Corporate Accounting | 103 | 2019-05-13 | MORNING |
504 | I | II | 22411202 | Corporate Laws | 110 | 2019-05-20 | MORNING |
504 | I | II | 72032800 | ENGLISH-A | 105 | 2019-05-07 | EVENING |
+----+----+-----+----+
224 rows in set (0.00 sec)

mysql>

```

```
25 self.master.grid_rowconfigure(0, weight=1)
PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL Python Debug Console + - [ ] [ ] ^ X
ad
na f1 s/
2s t
/usr/bin/env /bin/python3 /home/falc0n12/.vscode/extensions/ms-python.python-2021.12.155
9732655/pythonFiles/lib/python/debugpy/launcher 38999 -- "/home/falc0n12/Downloads/final 2/Final/Main.py"
db displayed sucessfully
db displayed sucessfully
db displayed sucessfully
db displayed sucessfully
```

Chapter 4

Future Scope and Benefits

This tool is especially valuable in the present time with the expanding number of individuals showing up for different examinations. Manually taking care of information isn't just dreary and tedious yet inclined to mistakes also. This product empowers the client to be accurate, quick and produce dependable outcomes. It deals with the framework productively and gets our work done. When the work has been executed, we can make changes physically also. It is a multi-client environment and can be effectively moved from one OS environment to another.

Some **benefits** of our system are:

1. When contrasted with existing framework our framework take very less time to execute every one of the information and show it on the user panel.
2. This framework lessens a great deal of weight to individuals in examination office. During the test, examiner has to check the whole room and furthermore count the number of seats and tables all alone.. So that's why making this application is extremely useful for the test controller.
3. The randomizing of individual's roll numbers is going to be exceptionally viable.
4. It very well may be utilized in different schools, organizations, school and at college level too.
5. A wide range of test can be directed under this framework so that there is no a particular breaking point of students.

In the **future system**, it can be implemented at an university level, wherein the system will add number of universities all at once so that different school overseers will have an authority to direct their test.

Apart from this, the existing system can be used in a college's ERP as a module or can be made into a website.

A Web application (Web app) is an application program that is stored on a remote server and delivered over the Internet through a browser interface. Web services are Web apps by definition and many, although not all, websites contain Web apps. According to Web.AppStorm editor Jarel Remick, any website component that performs some function for the user qualifies as a Web app.

Web applications can be designed for a wide variety of uses and can be used by anyone; from an organization to an individual for numerous reasons. Commonly used Web applications can include webmail, online calculators, or e-commerce shops. Some Web apps can be only accessed by a specific browser; however, most are available no matter the browser.

A Web application (Web app) is an application program that is stored on a remote server and delivered over the Internet through a browser interface. Web services are Web apps by definition and many, although not all, websites contain Web apps. According to Web.AppStorm editor Jarel Remick, any website component that performs some function for the user qualifies as a Web app.

Web applications can be designed for a wide variety of uses and can be used by anyone; from an organization to an individual for numerous reasons. Commonly used Web applications can include webmail, online calculators, or e-commerce shops. Some Web apps can be only accessed by a specific browser; however, most are

available no matter the browser.

How Web applications work

Web applications do not need to be downloaded since they are accessed through a network. Users can access a Web application through a web browser such as Google Chrome, Mozilla Firefox or Safari.

For a web app to operate, it needs a Web server, application server, and a database. Web servers manage the requests that come from a client, while the application server completes the requested task. A database can be used to store any needed information. Web applications typically have short development cycles and can be made with small development teams. Most Web apps are written in JavaScript, HTML5, or Cascading Style Sheets (CSS). Client-side programming typically utilizes these languages, which help build an applications front-end. Server-side programming is done to create the scripts a Web app will use. Languages such as Python, Java, and Ruby are commonly used in server-side programming.

Web applications have many different uses, and with those uses, comes many potential benefits. Some common benefits of Web apps include:

- Allowing multiple users access to the same version of an application.
- Web apps don't need to be installed.
- Web apps can be accessed through various platforms such as a desktop, laptop, or mobile.
- Can be accessed through multiple browsers.

Chapter 5

Conclusion and References

Chapter 5.1

Conclusion

This project has been an incredible learning experience because in addition to the fact that it is productive, it also provides an extraordinary strategy to reduce work. It facilitates our responsibility and gives us an accurate measure to determine seating arrangements. Most organizations ought to introduce this product and it can be an incredible assistance to them. Aside from that, it can be extended to use in other occasions and meetings where huge number of individuals are present. Some advantages of this model are that it is exceptionally quick, solid and strong. The significance of this product is to make our undertakings quicker and more dependable. In the present day and age work-diminishing applications are highly required in light of the expanding requests of different sectors.

Chapter 5.2 References

- [1] Hemalatha, L. (2020). Project Report On Automatic Exam Hall Seat Arrangement (Doctoral dissertation, CMR Institute of Technology. Bangalore).
- [2] Muhammad Ramees C. K 1 , Sherin Eliyas2," Efficient Seat Each and every singleocation Process in College Exam System" ,International Journal for Research in Applied Science and Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887 Volume 6 Issue V, May 2
- [3] Dinesh Chandewar, Mainak Saha, Pushpraj Deshkar, Pankaj Wankhede, "Programmed Seating Arrangement of University Exam", IJSTE - International Journal of Science Technology and Engineering | Volume 3 | Issue 09 | March 2017
- [4] Dhotre, D. R., Makwane, S., & Lahase, P. (2019). Automatic Exam Seating Arrangement System.
- [5] Alam, A. F. (2016). Automatic seating arrangement tool for examinations in universities/colleges. International Journal of Engineering Applied Sciences and Technology (IJEAST), 1(4), 2455- 2143.
- [6] Alawode, A. J., & Adegboye, A. J. (2019). AUTOMATIC SEATING ARRANGEMENT SYSTEM USING TABU SEARCH ALGORITHM. FEPI-JOPAS, 1(1), 127-133.
- [7] Chaki, P. K., & Anirban, S. (2016, March). Algorithm for efficient seating plan for centralized exam system. In 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT) (pp. 320-325). IEEE
- Mohler, Beth A. (2005). Citation analysis as an assessment tool. Science & Technology Librar-ies, 25(4), 57-64.
- Oppenheim, Charles, & Richard Smith. (2001). Student citation practices in an Information Sci-ence Department. Education for Information, 19, 299-323.
- Tunon, Johanna, & Bruce Brydges. (2005, August). Improving the quality of

university libraries through citation mining and analysis using two new dissertation bibliometric assessment tool. In World Library and Information Congress: 71th IFLA General Conference and Council. Retrieved October 1, 2006, Ursin, Lara, Elizabeth Blakesley Lindsay, & Corey M. Johnson. (2004). Assessing library instruction in the freshman seminar: A citation analysis study. *Reference Services Review*, 32(3), 284-292.

Young, Virginia E., & Linda G. Ackerson. (1994). Evaluating the impact of library instruction methods on the quality of student research. *Research Strategies*, 12(3), 132-144.

Young, Virginia E., & Linda G. Ackerson. (1995). Evaluation of student research paper bibliographies: Re-examining evaluation criteria. *Research Strategies*, 13(2), 80-93.

Chapter 5.3 Publication/Acceptance Screenshots

