St. John's Public School

Learning for Life

COMPUTER SCIENCE

INVESTIGATORY PROJECT

Submitted for

ALL INDIA SENIOR SCHOOL

CERTIFICATE EXAMINATION

2022-2023

STUDENT MARK ANALYSIS

Done By

KINGSTON RICHARD J.

SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my teacher **Mrs. S. Josephine Backiam** for her guidance, support throughout the duration of the project. We completed the project successfully by her motivation and her extended support for us.

As well as I would like to thank our **Correspondent Dr. R. Kishore Kumar**, our **Principal Mrs. Shanthi Samuel** and **Vice Principal Mrs. Angeline Christopher Roy** who gave me the golden opportunity to do this project, which also helped me in doing a lot of research and I came to know about so many new things when I was doing the project.

Finally, I would also like to **thank my parents and friends** who helped me a lot in finalizing this project within the limited time frame.

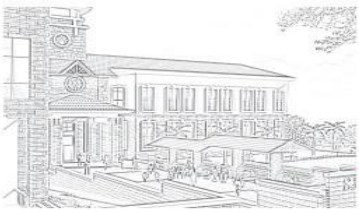
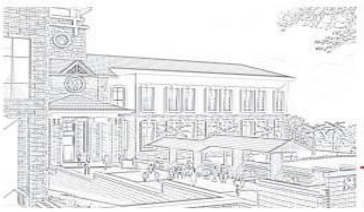


TABLE OF CONTENTS

S.NO.	CONTENTS	PAGE NO.
1.	Introduction	1
2.	Objective	2
3.	Modularization Approach	3
4.	Software & Hardware Requirements	4
5.	Why Python?	5
6.	Program Code	6 - 114
7.	Output	115 - 128
8.	Advantages of the Project	129
9.	Limitations of the Project	130
10.	Further Development Areas	131
11.	Bibliography	132

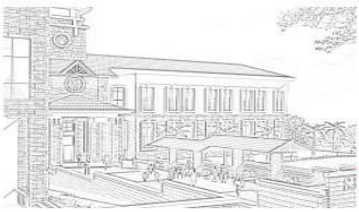


INTRODUCTION

Python is a **computer programming language** used for many purposes such as website development, software, task automation, data analysis, etc. It is a **high-level programming language** that has English-like syntax. This makes it easier to read and understand the code. It is an **interpreted language** which means that Python directly executes the code line by line. In case of any error, it stops further execution and reports back the error which has occurred.

Data management is very well important. It involves collection of data and using it securely and productively. Managing the data collected and updating it time to time is very important. A **classroom-level data management system** helps teachers track each student's performance. By examining the data plotted on each student's progress monitoring graph, the teachers can determine whether students are making adequate progress.

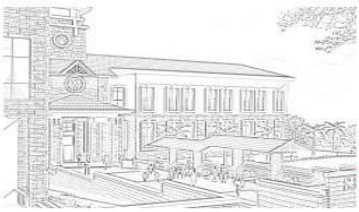
We have done a project using Python class XII knowledge. It is going to be on "**Students Mark Analysis**". It deals with more of the class XII concepts and also deals with Student Mark Management system in schools.



OBJECTIVE

The purpose of this project is to generate a **Mark Analysis for Class XII** students using Python programming language. Through this project, the teachers can execute various tasks like **searching**, **updating and deleting of marks scored** by each class XII student **in all the examinations**.

Parents of each student, **respective class teacher** of each class and **students** themselves will get the **overall report of the student's performance including,** marks scored, individual subject marks, individual subject grade obtained, overall grade obtained, total marks obtained and percentage of marks. **Comparison will be made between** the marks scored by the student and with those of - marks secured by him/her in previous examination, class average and marks of the top 5 students of the same class. Rank obtained by each student will also be printed. Finally, it states whether the student has passed or failed in the current examination.

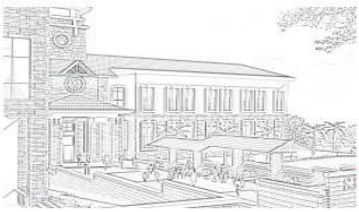


MODULARIZATION APPROACH

- Database – This contains the information of all the Students and Teachers. This Module contains the Marks of the Students, the Attendance, their personal details, the username and password of Teachers and the sections handled by them.

- Framework – This module is the heart of the program. It contains all the logics required to run the program. It contains the logics of logging in, creating and displaying graphs and report cards, editing and viewing marks of the students by Teachers.

- Frontend – This module contains the coding for the GUI of the project. It renders the "Student Mark Analysis" application.



SOFTWARE & HARDWARE REQUIREMENTS

Software Requirements:

→ Python IDLE

→ Google Chrome & Microsoft Edge

→ Windows 10

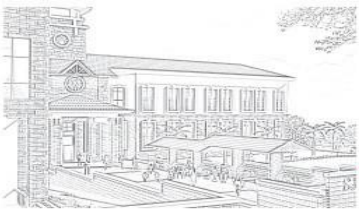
→ Microsoft Word

Hardware:

→ Laptop

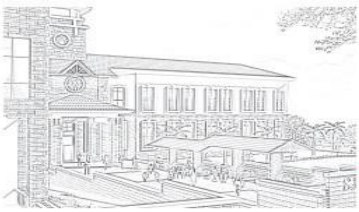
→ WIFI-Router

→ Printer



WHY PYTHON?

- ★ Python is a computer programming language often used to **build websites and software**, **automate tasks** and **conduct data analysis**.
- ★ It is a **general-purpose language**, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems.
- ★ It has a **simple syntax** that mimics English language, so it's easier to read and understand. This makes it quicker to build projects, and faster to improve them.
- ★ It's **open source**, which means it's free to use and distribute.
- ★ Python has a **large and active community** that contributes to Python's pool of modules and libraries and acts as a helpful resource for other programmers.
- ★ Python can be used by relative beginners very easily.
- ★ It is very **versatile** and **beginner-friendly**.
- ★ It can also be used to **automate simple tasks** on the computer.



PROGRAM CODE

Database.py:

```
import pickle

#The Marks are Out of 100

Marks_A={"Section":"A",

"aakash varma":{"Midterm-1":(56,68,66,58,50), "Terminal-1":(45,46,38,71,53), "Midterm-2":(45,89,34,97,25), "Terminal-2":(30,40,50,60,70)},

"abishek":{"Midterm-1":(92,70,68,59,52), "Terminal-1":(47,47,41,73,54), "Midterm-2":(46,91,38,95,29), "Terminal-2":(33,42,46,64,71)},

"anand raj":{"Midterm-1":(96,72,70,60,54), "Terminal-1":(52,51,44,75,55), "Midterm-2":(48,93,42,93,33), "Terminal-2":(35,44,48,66,72)},

"david washington":{"Midterm-1":(55,50,56,60,55), "Terminal-1":(88,82,80,66,77), "Midterm-2":(81,61,45,49,46), "Terminal-2":(54,70,64,54,46)},

"emannuel":{"Midterm-1":(97,74,73,64,59), "Terminal-1":(57,55,47,77,56), "Midterm-2":(50,34,46,91,37), "Terminal-2":(40,48,53,71,73)},

"harley sumesh":{"Midterm-1":(100,76,75,65,61), "Terminal-1":(59,57,50,79,57), "Midterm-2":(45,36,50,89,41), "Terminal-2":(43,50,57,75,74)},

"karthikeya airpirala":{"Midterm-1":(41,78,78,69,66), "Terminal-1":(64,61,53,81,58), "Midterm-2":(47,41,54,87,45), "Terminal-2":(45,54,62,80,75)},

"kaushik gautham":{"Midterm-1":(64,80,80,70,68), "Terminal-1":(69,63,56,83,59), "Midterm-2":(49,43,58,85,49), "Terminal-2":(47,56,65,82,76)},

"kavin raj":{"Midterm-1":(35,82,83,74,73), "Terminal-1":(74,67,59,85,60), "Midterm-2":(51,48,62,83,53), "Terminal-2":(49,60,31,87,77)},
```

"kevin raj":{"Midterm-1":(69,84,85,75,75), "Terminal-1":(79,71,62,34,61), "Midterm-2":(53,50,66,81,57), "Terminal-2":(51,64,33,89,78)},

"muhammad shayaan":{"Midterm-1":(92,86,87,76,77), "Terminal-1":(81,73,65,36,62), "Midterm-2":(55,52,70,79,61), "Terminal-2":(53,66,36,93,79)},

"padmesh prasadh":{"Midterm-1":(76,88,89,77,79), "Terminal-1":(86,77,68,38,63), "Midterm-2":(57,54,74,77,65), "Terminal-2":(55,70,38,95,80)},

"pravinesh ramprasad":{"Midterm-1":(89,90,91,78,81), "Terminal-1":(45,79,71,40,64), "Midterm-2":(59,56,78,75,69), "Terminal-2":(57,72,41,99,81)},

"rahul ganesh":{"Midterm-1":(80,92,94,82,86), "Terminal-1":(50,83,74,42,65), "Midterm-2":(61,31,82,73,73), "Terminal-2":(59,76,43,91,82)},

"rajan":{"Midterm-1":(24,94,96,83,88), "Terminal-1":(45,87,77,44,66), "Midterm-2":(63,33,86,71,77), "Terminal-2":(61,80,45,93,83)},

"rajesh sukuram":{"Midterm-1":(27,96,98,84,90), "Terminal-1":(50,89,80,46,67), "Midterm-2":(65,35,90,69,81), "Terminal-2":(63,82,48,87,84)},

"sabaresh sudhakar":{"Midterm-1":(81,32,63,45,31), "Terminal-1":(55,93,83,48,68), "Midterm-2":(67,37,94,67,85), "Terminal-2":(65,86,50,99,85)},

"sai samuel":{"Midterm-1":(32,34,66,49,36), "Terminal-1":(60,97,86,50,69), "Midterm-2":(69,42,98,65,89), "Terminal-2":(31,90,55,84,86)},

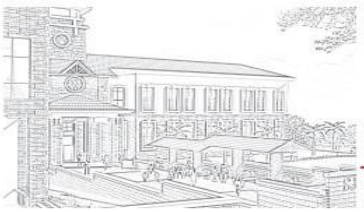
"shreyas sudip mitra":{"Midterm-1":(45,36,68,50,38), "Terminal-1":(65,99,89,52,70), "Midterm-2":(71,44,82,63,93), "Terminal-2":(33,92,58,96,87)},

"sundar raman":{"Midterm-1":(39,38,70,51,40), "Terminal-1":(70,73,92,54,71), "Midterm-2":(73,46,86,61,97), "Terminal-2":(35,96,60,88,88)},

"aarthi":{"Midterm-1":(78,40,72,52,42), "Terminal-1":(72,68,95,56,72), "Midterm-2":(74,48,80,59,100), "Terminal-2":(38,98,45,82,89)},

"aiswarya rani":{"Midterm-1":(82,42,74,53,44), "Terminal-1":(74,70,98,58,73), "Midterm-2":(75,50,24,57,95), "Terminal-2":(41,100,49,96,90)},

"amirtha varshini":{"Midterm-1":(46,44,76,54,46), "Terminal-1":(79,74,61,60,74), "Midterm-2":(77,52,58,55,34), "Terminal-2":(43,100,51,88,91)},



"avanthika":{"Midterm-1":(50,46,78,55,48), "Terminal-1":(81,76,64,62,75), "Midterm-2":(78,54,42,53,38), "Terminal-2":(46,84,55,45,45)},

"brighty salomi":{"Midterm-1":(51,48,81,59,53), "Terminal-1":(86,80,67,64,76), "Midterm-2":(80,59,96,51,42), "Terminal-2":(51,19,60,50,45)},

"diya sriram":{"Midterm-1":(56,52,59,64,60), "Terminal-1":(93,86,93,68,78), "Midterm-2":(83,66,49,47,50), "Terminal-2":(59,65,69,59,47)},

"gayathri sudha":{"Midterm-1":(60,54,61,65,62), "Terminal-1":(95,88,56,70,79), "Midterm-2":(84,68,53,45,54), "Terminal-2":(62,96,73,63,48)},

"hamsadhwani":{"Midterm-1":(64,56,63,66,64), "Terminal-1":(100,92,36,72,80), "Midterm-2":(86,70,56,43,58), "Terminal-2":(64,98,75,65,49)},

"hannah sumesh":{"Midterm-1":(67,58,65,67,66), "Terminal-1":(54,94,39,74,81), "Midterm-2":(88,72,60,41,62), "Terminal-2":(67,40,78,69,50)},

"harini raj":{"Midterm-1":(71,60,67,68,68), "Terminal-1":(74,96,42,76,82), "Midterm-2":(89,74,64,39,66), "Terminal-2":(70,65,82,73,51)},

"keerthana":{"Midterm-1":(46,62,69,69,70), "Terminal-1":(86,43,45,78,83), "Midterm-2":(90,76,68,37,45), "Terminal-2":(73,42,42,77,52)},

"keren david":{"Midterm-1":(47,64,72,73,75), "Terminal-1":(91,47,48,80,84), "Midterm-2":(92,81,72,35,49), "Terminal-2":(75,88,44,31,53)},

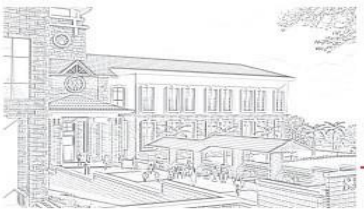
"madhumitha":{"Midterm-1":(91,66,74,74,77), "Terminal-1":(67,79,51,82,85), "Midterm-2":(93,83,76,33,53), "Terminal-2":(78,30,65,56,54)},

"monika sagar":{"Midterm-1":(92,68,77,78,82), "Terminal-1":(68,83,54,84,86), "Midterm-2":(95,88,80,31,57), "Terminal-2":(80,34,67,61,55)},

"nithya":{"Midterm-1":(23,70,79,79,84), "Terminal-1":(20,84,52,86,87), "Midterm-2":(96,90,84,29,61), "Terminal-2":(45,36,78,89,56)},

"preyanka shankar":{"Midterm-1":(27,72,81,80,86), "Terminal-1":(25,46,55,88,88), "Midterm-2":(98,92,88,27,65), "Terminal-2":(47,40,80,91,57)},

"rajeswari narayanan":{"Midterm-1":(30,74,83,81,88), "Terminal-1":(90,48,58,90,89), "Midterm-2":(100,94,92,25,69), "Terminal-2":(50,42,83,95,58)},



"soundarya rajan":{"Midterm-1":(34,76,85,82,90), "Terminal-1":(29,50,61,92,90), "Midterm-2":(71,96,96,23,73), "Terminal-2":(45,44,92,99,59)},

"sowmya sri":{"Midterm-1":(38,78,87,83,92), "Terminal-1":(34,54,64,94,91), "Midterm-2":(53,98,100,21,77), "Terminal-2":(47,48,94,91,60)},

"varsha":{"Midterm-1":(91,80,89,84,94), "Terminal-1":(39,58,67,96,92), "Midterm-2":(85,100,84,19,81), "Terminal-2":(49,50,96,93,61)}

}

Marks_B={"Section":"B",

"andrew ezhil":{"Midterm-1":(89,90,78,84,67), "Terminal-1":(78,88,65,89,100), "Midterm-2":(94,87,67,56,95), "Terminal-2":(78,43,79,90,65)},

"arvin samuel":{"Midterm-1":(99,94,92,97,100), "Terminal-1":(90,91,96,100,92), "Midterm-2":(93,94,99,90,95), "Terminal-2":(90,91,99,94,95)},

"benedict raj":{"Midterm-1":(89,83,79,90,88), "Terminal-1":(91,67,71,77,60), "Midterm-2":(81,86,92,98,100), "Terminal-2":(100,94,88,82,91)},

"chandru":{"Midterm-1":(77,60,91,54,97), "Terminal-1":(88,51,64,72,50), "Midterm-2":(74,92,60,76,62), "Terminal-2":(78,43,79,90,65)},

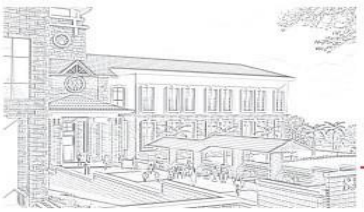
"daniel anto benjamine":{"Midterm-1":(82,56,69,42,98), "Terminal-1":(66,88,54,89,45), "Midterm-2":(41,56,61,67,55), "Terminal-2":(81,55,87,65,78)},

"elton craig samson":{"Midterm-1":(89,90,78,84,81), "Terminal-1":(48,77,65,62,78), "Midterm-2":(91,47,87,65,96), "Terminal-2":(78,100,89,65,77)},

"ezhil adhithya":{"Midterm-1":(94,95,98,92,90), "Terminal-1":(90,92,95,91,100), "Midterm-2":(91,95,91,96,99), "Terminal-2":(100,98,97,100,98)},

"ivaniel joshua":{"Midterm-1":(71,90,88,64,77), "Terminal-1":(68,90,55,89,91), "Midterm-2":(34,67,75,81,45), "Terminal-2":(96,57,49,90,88)},

"lokeshwar":{"Midterm-1":(100,53,78,94,91), "Terminal-1":(72,84,67,88,90), "Midterm-2":(92,91,47,65,79), "Terminal-2":(45,63,99,34,65)},



"karan raj":{"Midterm-1":(79,80,38,94,67), "Terminal-1":(78,58,62,49,90), "Midterm-2":(95,67,59,63,83), "Terminal-2":(38,43,69,37,79)},

"kingston richard":{"Midterm-1":(90,100,98,92,96), "Terminal-1":(65,97,90,93,100), "Midterm-2":(90,92,94,96,98), "Terminal-2":(91,93,95,90,97)},

"kommal":{"Midterm-1":(91,49,58,94,77), "Terminal-1":(89,92,61,75,40), "Midterm-2":(54,67,72,59,65), "Terminal-2":(43,67,79,34,95)},

"lebrio mascarnaz":{"Midterm-1":(69,93,88,89,67), "Terminal-1":(61,89,64,79,90), "Midterm-2":(74,87,73,45,75), "Terminal-2":(88,33,59,80,45)},

"sivanjan":{"Midterm-1":(67,95,58,74,64), "Terminal-1":(78,88,45,89,70), "Midterm-2":(91,41,65,66,95), "Terminal-2":(98,89,79,47,64)},

"porchelvan":{"Midterm-1":(39,94,58,74,64), "Terminal-1":(79,87,66,90,48), "Midterm-2":(100,72,57,90,94), "Terminal-2":(79,44,87,90,64)},

"manda sai nikhil":{"Midterm-1":(90,87,66,54,87), "Terminal-1":(88,79,64,69,79), "Midterm-2":(44,89,64,59,100), "Terminal-2":(48,66,80,94,57)},

"sanjay":{"Midterm-1":(67,50,100,74,43), "Terminal-1":(75,82,75,49,89), "Midterm-2":(54,67,87,56,97), "Terminal-2":(95,92,90,92,89)},

"siva kailash":{"Midterm-1":(89,91,52,80,64), "Terminal-1":(98,89,61,80,79), "Midterm-2":(34,87,77,66,94), "Terminal-2":(76,44,78,91,82)},

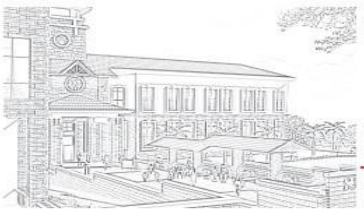
"sai pranav":{"Midterm-1":(79,88,65,89,100), "Terminal-1":(58,83,62,85,97), "Midterm-2":(100,87,47,86,94), "Terminal-2":(68,23,75,94,65)},

"tarunkumar":{"Midterm-1":(39,40,58,94,61), "Terminal-1":(67,89,65,90,49), "Midterm-2":(50,67,60,36,99), "Terminal-2":(86,44,71,67,52)},

"akshita patnaik":{"Midterm-1":(79,80,45,67,99), "Terminal-1":(90,56,67,79,60), "Midterm-2":(100,67,87,89,90), "Terminal-2":(78,80,56,46,92)},

"aparnashree":{"Midterm-1":(80,79,57,89,52), "Terminal-1":(93,90,92,99,96), "Midterm-2":(89,81,67,91,76), "Terminal-2":(90,96,83,79,51)},

"augustina noel":{"Midterm-1":(96,87,69,96,45), "Terminal-1":(67,89,61,56,90), "Midterm-2":(78,90,47,82,94), "Terminal-2":(65,90,71,48,60)},



"bharathi":{"Midterm-1":(89,56,67,34,80), "Terminal-1":(79,82,90,95,100), "Midterm-2":(90,98,67,47,81), "Terminal-2":(90,87,33,46,79)},

"bhavadharshni":{"Midterm-1":(56,61,35,89,92), "Terminal-1":(93,45,66,88,92), "Midterm-2":(92,45,74,87,43), "Terminal-2":(47,90,67,49,65)},

"deshna":{"Midterm-1":(90,92,95,89,60), "Terminal-1":(69,99,68,100,91), "Midterm-2":(93,95,97,99,73), "Terminal-2":(89,39,79,65,47)},

"diyashri":{"Midterm-1":(89,90,45,78,62), "Terminal-1":(89,72,90,37,51), "Midterm-2":(67,89,80,86,79), "Terminal-2":(89,98,76,83,78)},

"jeeva shree":{"Midterm-1":(79,62,84,92,47), "Terminal-1":(65,72,57,60,69), "Midterm-2":(80,45,33,69,90), "Terminal-2":(91,77,85,94,50)},

"jeevika":{"Midterm-1":(90,97,92,89,98), "Terminal-1":(97,91,82,87,100), "Midterm-2":(67,89,45,92,84), "Terminal-2":(99,95,90,93,96)},

"jenewar sarah singh":{"Midterm-1":(67,78,100,83,33), "Terminal-1":(39,47,87,90,79), "Midterm-2":(69,33,76,99,45), "Terminal-2":(94,79,82,33,92)},

"jyotsna":{"Midterm-1":(78,89,65,89,33), "Terminal-1":(56,89,90,99,78), "Midterm-2":(89,78,33,71,80), "Terminal-2":(33,67,89,57,90)},

"kandiz solita":{"Midterm-1":(56,79,41,33,90), "Terminal-1":(80,78,35,67,51), "Midterm-2":(89,90,71,35,60), "Terminal-2":(67,33,56,90,49)},

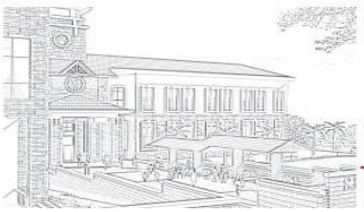
"kavya":{"Midterm-1":(91,98,94,96,67), "Terminal-1":(87,97,93,90,80), "Midterm-2":(86,82,90,94,75), "Terminal-2":(67,49,50,95,100)},

"leena":{"Midterm-1":(89,67,90,95,48), "Terminal-1":(79,75,80,33,100), "Midterm-2":(96,90,92,97,67), "Terminal-2":(79,90,49,84,60)},

"meher fathima":{"Midterm-1":(79,85,71,70,89), "Terminal-1":(90,95,96,58,71), "Midterm-2":(67,94,95,92,89), "Terminal-2":(60,33,57,83,45)},

"mohita":{"Midterm-1":(69,33,100,77,82), "Terminal-1":(82,100,73,90,33), "Midterm-2":(67,88,90,97,100), "Terminal-2":(37,35,89,100,47)},

"pon leela percy":{"Midterm-1":(67,78,90,34,49), "Terminal-1":(45,100,35,75,60), "Midterm-2":(60,98,100,47,33), "Terminal-2":(67,85,100,98,40)},



"shalini":{"Midterm-1":(39,78,90,45,100), "Terminal-1":(100,49,67,89,42), "Midterm-2":(89,94,78,90,100), "Terminal-2":(100,47,89,39,50)},

"sharon david":{"Midterm-1":(97,99,95,92,90), "Terminal-1":(94,98,90,67,89), "Midterm-2":(88,83,80,94,33), "Terminal-2":(67,89,79,80,37)},

"sindhuja":{"Midterm-1":(46,78,67,100,78), "Terminal-1":(97,100,87,90,89), "Midterm-2":(87,89,100,67,79), "Terminal-2":(78,90,46,89,97)}

}

Marks_C={"Section":"C",

"abishek joseph":{"Midterm-1":(56,78,49,76,47), "Terminal-1":(59,82,56,70,56), "Midterm-2":(54,82,78,69,70), "Terminal-2":(80,79,70,70,69)},

"alexander":{"Midterm-1":(96,54,19,64,45), "Terminal-1":(74,51,60,41,35), "Midterm-2":(19,41,21,35,23), "Terminal-2":(50,61,45,32,49)},

"ben laurie":{"Midterm-1":(56,21,13,62,10), "Terminal-1":(21,50,25,40,15), "Midterm-2":(45,25,24,52,26), "Terminal-2":(51,61,43,57,53)},

"calvin rodrigues":{"Midterm-1":(58,64,29,74,54), "Terminal-1":(59,64,39,42,56), "Midterm-2":(45,58,74,76,75), "Terminal-2":(54,65,95,43,61)},

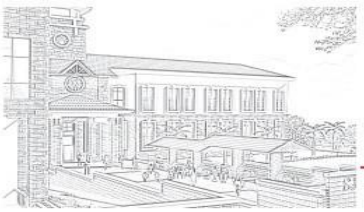
"dinesh kumar":{"Midterm-1":(57,69,84,54,65), "Terminal-1":(45,25,45,68,94), "Midterm-2":(47,58,96,54,67), "Terminal-2":(74,65,84,54,95)},

"ethan james":{"Midterm-1":(70,44,93,76,39), "Terminal-1":(40,22,44,96,90), "Midterm-2":(84,60,49,35,80), "Terminal-2":(96,77,82,62,55)},

"francis xavier":{"Midterm-1":(11,36,28,25,38), "Terminal-1":(28,34,31,38,36), "Midterm-2":(22,21,17,29,10), "Terminal-2":(32,21,17,12,22)},

"kobe jordan":{"Midterm-1":(38,57,35,58,31), "Terminal-1":(38,58,55,65,35), "Midterm-2":(65,57,34,31,64), "Terminal-2":(69,49,40,58,66)},

"koushik dev":{"Midterm-1":(50,58,60,54,53), "Terminal-1":(47,56,55,42,41), "Midterm-2":(40,53,57,51,48), "Terminal-2":(57,44,51,50,49)},



"logan sprite":{"Midterm-1":(75,62,73,58,87), "Terminal-1":(89,84,80,91,67), "Midterm-2":(69,79,71,68,72), "Terminal-2":(72,80,99,71,84)},

"nathan drake":{"Midterm-1":(64,73,99,77,100), "Terminal-1":(73,78,74,67,50), "Midterm-2":(54,91,80,68,71), "Terminal-2":(93,69,68,79,90)},

"naufal rizwan":{"Midterm-1":(50,61,68,56,43), "Terminal-1":(70,76,43,53,51), "Midterm-2":(67,49,40,61,63), "Terminal-2":(66,55,53,44,69)},

"pradeep kumar":{"Midterm-1":(82,59,62,87,77), "Terminal-1":(76,57,83,82,70), "Midterm-2":(70,62,85,72,74), "Terminal-2":(68,69,64,74,56)},

"samuel divine":{"Midterm-1":(33,28,36,27,38), "Terminal-1":(32,39,24,28,37), "Midterm-2":(30,23,37,36,26), "Terminal-2":(34,39,40,29,35)},

"shawn thomas":{"Midterm-1":(51,71,47,55,80), "Terminal-1":(67,72,43,41,63), "Midterm-2":(48,43,57,56,70), "Terminal-2":(70,76,74,57,50)},

"srikanth kumaraesan":{"Midterm-1":(38,54,67,46,56), "Terminal-1":(35,33,42,49,38), "Midterm-2":(57,42,58,55,43), "Terminal-2":(61,36,66,46,51)},

"victor sullivan":{"Midterm-1":(95,65,48,78,87), "Terminal-1":(94,45,78,93,67), "Midterm-2":(76,68,63,53,55), "Terminal-2":(60,100,65,98,50)},

"vijay yashwanth":{"Midterm-1":(47,42,75,44,69), "Terminal-1":(79,53,92,64,85), "Midterm-2":(79,74,63,94,67), "Terminal-2":(95,75,50,96,91)},

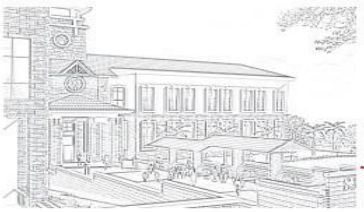
"vimal hassan":{"Midterm-1":(15,25,10,12,14), "Terminal-1":(34,21,24,39,16), "Midterm-2":(22,23,12,25,40), "Terminal-2":(29,20,34,17,33)},

"vishwa":{"Midterm-1":(21,15,11,31,14), "Terminal-1":(29,17,33,19,14), "Midterm-2":(39,14,19,33,10), "Terminal-2":(25,23,20,16,13)},

"alsey christina":{"Midterm-1":(55,40,59,94,89), "Terminal-1":(51,82,61,66,67), "Midterm-2":(89,57,53,75,45), "Terminal-2":(44,40,69,59,65)},

"brittany jules":{"Midterm-1":(45,78,47,96,50), "Terminal-1":(90,67,82,69,99), "Midterm-2":(95,59,61,47,40), "Terminal-2":(76,93,79,45,43)},

"catherine joel":{"Midterm-1":(85,37,92,34,82), "Terminal-1":(76,50,42,99,73), "Midterm-2":(75,36,53,66,45), "Terminal-2":(76,99,53,51,89)},



"daphne princy":{"Midterm-1":(84,31,58,77,80), "Terminal-1":(54,93,67,89,83), "Midterm-2":(72,73,99,79,63), "Terminal-2":(95,30,61,38,90)},

"deekshana":{"Midterm-1":(86,51,63,97,33), "Terminal-1":(38,74,51,80,75), "Midterm-2":(93,54,69,47,55), "Terminal-2":(72,62,77,58,64)},

"devashree":{"Midterm-1":(80,70,87,40,73), "Terminal-1":(82,85,43,46,69), "Midterm-2":(92,98,78,83,88), "Terminal-2":(80,42,52,51,50)},

"deepika neelamegan":{"Midterm-1":(88,65,68,48,62), "Terminal-1":(75,49,44,43,93), "Midterm-2":(69,43,52,82,64), "Terminal-2":(94,52,45,97,66)},

"dhisha vijay":{"Midterm-1":(80,46,57,70,47), "Terminal-1":(40,79,54,84,43), "Midterm-2":(76,75,54,85,61), "Terminal-2":(59,83,82,99,85)},

"dora maximoff":{"Midterm-1":(88,71,92,63,50), "Terminal-1":(76,64,74,92,87), "Midterm-2":(88,46,91,47,56), "Terminal-2":(45,83,95,40,56)},

"isha rajesh":{"Midterm-1":(71,62,65,96,58), "Terminal-1":(89,63,78,45,87), "Midterm-2":(78,64,73,65,94), "Terminal-2":(96,58,67,84,65)},

"indhu anand":{"Midterm-1":(47,57,95,41,89), "Terminal-1":(51,76,79,99,60), "Midterm-2":(61,52,89,44,83), "Terminal-2":(83,51,48,57,78)},

"jessica shammah":{"Midterm-1":(84,97,85,59,79), "Terminal-1":(91,82,59,42,60), "Midterm-2":(88,81,92,48,50), "Terminal-2":(46,41,94,91,90)},

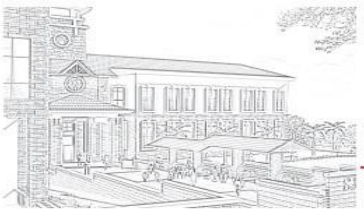
"manasa reddy":{"Midterm-1":(55,84,92,70,98), "Terminal-1":(66,73,87,57,56), "Midterm-2":(48,62,74,40,64), "Terminal-2":(79,95,81,56,89)},

"nidhi":{"Midterm-1":(98,53,76,49,62), "Terminal-1":(89,58,44,85,53), "Midterm-2":(76,71,63,61,95), "Terminal-2":(57,50,54,61,78)},

"pamila":{"Midterm-1":(81,65,96,80,63), "Terminal-1":(95,73,76,57,75), "Midterm-2":(64,48,56,67,69), "Terminal-2":(47,52,49,86,53)},

"prathipa":{"Midterm-1":(92,59,99,79,78), "Terminal-1":(87,68,96,65,47), "Midterm-2":(46,61,90,50,85), "Terminal-2":(76,55,87,62,45)},

"saishri":{"Midterm-1":(87,66,51,76,99), "Terminal-1":(48,72,56,81,78), "Midterm-2":(93,94,69,42,98), "Terminal-2":(91,55,51,56,52)},



"shreya":{"Midterm-1":(58,43,42,86,70), "Terminal-1":(61,76,56,66,62), "Midterm-2":(74,51,57,70,55), "Terminal-2":(81,69,62,68,73)},

"swathi":{"Midterm-1":(96,97,59,92,57), "Terminal-1":(98,84,51,64,65), "Midterm-2":(94,98,62,97,40), "Terminal-2":(45,60,52,66,90)},

"tessa thompson":{"Midterm-1":(49,70,98,97,66), "Terminal-1":(63,70,78,91,71), "Midterm-2":(53,59,85,45,70), "Terminal-2":(46,50,45,77,51)}

}

Marks_D={"Section":"D",

"alex luke":{"Midterm-1":(45,67,96,86,56), "Terminal-1":(34,68,78,95,30), "Midterm-2":(85,49,84,23,33), "Terminal-2":(67,45,34,66,76)},

"bob mark":{"Midterm-1":(45,68,69,58,48), "Terminal-1":(87,59,59,39,75), "Midterm-2":(85,94,74,35,48), "Terminal-2":(95,56,65,76,54)},

"carl engels":{"Midterm-1":(98,98,48,56,100), "Terminal-1":(85,95,48,67,83), "Midterm-2":(48,94,94,83,85), "Terminal-2":(87,94,94,37,38)},

"dan christ":{"Midterm-1":(75,95,95,94,94), "Terminal-1":(87,87,48,79,56), "Midterm-2":(100,95,86,48,38), "Terminal-2":(93,100,100,84,83)},

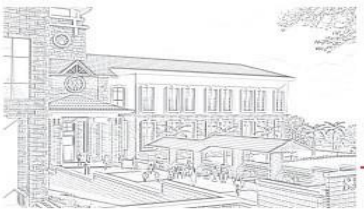
"ernest gayle":{"Midterm-1":(45,66,67,66,68), "Terminal-1":(45,54,33,54,34), "Midterm-2":(85,43,45,65,54), "Terminal-2":(95,99,43,76,88)},

"finn gates":{"Midterm-1":(55,65,65,65,45), "Terminal-1":(96,84,98,48,84), "Midterm-2":(83,100,38,58,94), "Terminal-2":(100,100,84,39,84)},

"grim roger":{"Midterm-1":(65,66,66,89,87), "Terminal-1":(56,67,87,87,56), "Midterm-2":(90,80,76,86,89), "Terminal-2":(67,100,100,100,85)},

"henderson fox":{"Midterm-1":(98,96,98,49,67), "Terminal-1":(97,69,59,96,85), "Midterm-2":(98,59,75,85,87), "Terminal-2":(76,86,95,95,85)},

"ivin grem":{"Midterm-1":(98,86,95,95,85), "Terminal-1":(95,58,58,74,85), "Midterm-2":(100,100,86,85,67), "Terminal-2":(100,98,100,76,97)},



"jack dorsen":{"Midterm-1":(87,96,60,58,84), "Terminal-1":(23,55,44,30,85), "Midterm-2":(76,95,84,83,94), "Terminal-2":(98,99,74,74,74)},

"kremlin kallus":{"Midterm-1":(85,85,85,94,76), "Terminal-1":(74,58,86,74,24), "Midterm-2":(85,84,63,65,75), "Terminal-2":(98,86,76,55,66)},

"lawson lewis":{"Midterm-1":(45,66,75,85,57), "Terminal-1":(34,100,57,86,85), "Midterm-2":(67,56,45,76,76), "Terminal-2":(34,100,86,48,74)},

"martin watts":{"Midterm-1":(56,78,54,65,54), "Terminal-1":(45,65,65,44,67), "Midterm-2":(23,45,55,65,66), "Terminal-2":(45,67,66,54,76)},

"newland warner":{"Midterm-1":(56,76,78,55,96), "Terminal-1":(56,77,76,76,86), "Midterm-2":(87,84,95,94,73), "Terminal-2":(94,56,23,43,49)},

"orlean cook":{"Midterm-1":(76,85,94,84,84), "Terminal-1":(76,86,94,94,65), "Midterm-2":(98,84,84,74,78), "Terminal-2":(76,94,75,74,49)},

"peter yottam":{"Midterm-1":(56,76,76,95,85), "Terminal-1":(94,84,87,84,75), "Midterm-2":(94,84,76,89,73), "Terminal-2":(98,74,88,84,54)},

"quinton joshua":{"Midterm-1":(98,94,88,94,91), "Terminal-1":(98,78,61,72,86), "Midterm-2":(94,50,29,74,71), "Terminal-2":(85,74,83,72,80)},

"rambo jackson":{"Midterm-1":(75,83,82,97,94), "Terminal-1":(93,97,87,72,61), "Midterm-2":(100,45,86,84,64), "Terminal-2":(98,98,37,73,21)},

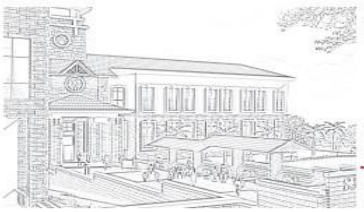
"stuart harley":{"Midterm-1":(88,84,70,40,28), "Terminal-1":(58,74,78,89,87), "Midterm-2":(75,83,84,76,72), "Terminal-2":(34,56,73,89,40)},

"tom alstair":{"Midterm-1":(98,98,73,73,67), "Terminal-1":(67,84,63,72,91), "Midterm-2":(98,67,86,74,24), "Terminal-2":(89,56,34,54,30)},

"angelina bramwell":{"Midterm-1":(84,83,38,39,63), "Terminal-1":(78,83,62,74,81), "Midterm-2":(98,97,96,34,45), "Terminal-2":(96,90,92,47,36)},

"bell rhode":{"Midterm-1":(85,98,78,67,32), "Terminal-1":(33,56,43,57,59), "Midterm-2":(89,78,37,74,21), "Terminal-2":(87,78,73,33,34)},

"christine puglisi":{"Midterm-1":(67,78,89,74,72), "Terminal-1":(98,63,72,67,43), "Midterm-2":(87,85,68,95,96), "Terminal-2":(98,64,73,98,100)},



"duerre watson":{"Midterm-1":(89,97,76,79,67), "Terminal-1":(87,65,96,56,46), "Midterm-2":(75,98,45,97,56), "Terminal-2":(87,54,76,98,67)},

"emma charlotte":{"Midterm-1":(98,97,76,98,100), "Terminal-1":(98,78,67,95,97), "Midterm-2":(89,86,98,70,60), "Terminal-2":(98,97,67,100,89)},

"francis red":{"Midterm-1":(87,98,78,68,70), "Terminal-1":(98,60,87,89,67), "Midterm-2":(86,56,93,30,67), "Terminal-2":(78,97,56,98,100)},

"gatsby joe":{"Midterm-1":(76,98,78,96,70), "Terminal-1":(97,96,87,56,90), "Midterm-2":(80,97,86,87,68), "Terminal-2":(74,82,20,98,34)},

"holly graf":{"Midterm-1":(85,84,86,83,90), "Terminal-1":(98,73,98,93,83), "Midterm-2":(99,77,73,76,78), "Terminal-2":(100,98,89,93,100)},

"julie orchard":{"Midterm-1":(78,98,91,65,73), "Terminal-1":(98,78,45,61,90), "Midterm-2":(65,78,89,40,67), "Terminal-2":(74,93,23,21,78)},

"kate brownwell":{"Midterm-1":(87,57,93,80,58), "Terminal-1":(67,80,84,73,87), "Midterm-2":(98,67,45,43,54), "Terminal-2":(98,91,57,83,78)},

"larson brown":{"Midterm-1":(92,63,78,93,94), "Terminal-1":(91,93,45,43,76), "Midterm-2":(98,84,82,93,31), "Terminal-2":(98,73,74,66,88)},

"marie joseph":{"Midterm-1":(87,74,82,92,57), "Terminal-1":(89,44,83,74,34), "Midterm-2":(87,97,74,83,32), "Terminal-2":(97,83,82,94,64)},

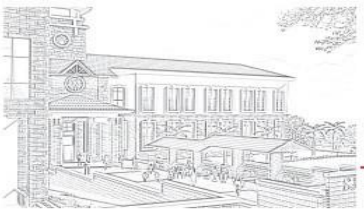
"oleson jinn":{"Midterm-1":(98,99,82,92,56), "Terminal-1":(74,83,95,33,24), "Midterm-2":(34,56,30,100,34), "Terminal-2":(95,47,38,56,43)},

"penrose walter":{"Midterm-1":(84,93,78,93,74), "Terminal-1":(73,56,43,67,45), "Midterm-2":(98,94,34,76,90), "Terminal-2":(98,78,67,84,33)},

"rosetta park":{"Midterm-1":(87,45,65,65,60), "Terminal-1":(87,74,38,32,39), "Midterm-2":(87,34,54,45,22), "Terminal-2":(87,33,43,33,33)},

"scarlett richard":{"Midterm-1":(87,95,85,48,67), "Terminal-1":(98,100,67,47,87), "Midterm-2":(98,95,74,75,100), "Terminal-2":(98,100,100,89,54)},

"sofia samuel":{"Midterm-1":(75,94,67,94,93), "Terminal-1":(83,92,89,83,90), "Midterm-2":(98,98,78,67,23), "Terminal-2":(87,67,74,82,45)},



"thebe kells":{"Midterm-1":(100,78,94,92,71), "Terminal-1":(76,82,92,19,74), "Midterm-2":(98,74,96,100,56), "Terminal-2":(87,89,82,46,76)},

"yolin goldie":{"Midterm-1":(78,100,67,87,89), "Terminal-1":(98,99,100,87,93), "Midterm-2":(87,98,99,89,100), "Terminal-2":(96,100,100,93,100)},

"zara hunter":{"Midterm-1":(89,94,85,83,34), "Terminal-1":(86,94,83,73,26), "Midterm-2":(67,34,37,65,34), "Terminal-2":(90,78,84,78,33)},

}

Details_A={"aakash varma":{"DOB":(11,3,2005),"Phone":"9790890914"},

"abishek":{"DOB":(31,4,2006),"Phone":"9091823045"},

"anand raj":{"DOB":(12,1,2005),"Phone":"9989067660"},

"david washington":{"DOB":(14,6,2005),"Phone":"8009023214"},

"emannuel":{"DOB":(13,7,2006),"Phone":"7456789019"},

"harley sumesh":{"DOB":(15,8,2005),"Phone":"6578909123"},

"karthikeya airpirala":{"DOB":(19,2,2005),"Phone":"7901256789"},

"kaushik gautham":{"DOB":(30,6,2006),"Phone":"6890921345"},

"kavin raj":{"DOB":(16,7,2005),"Phone":"7670098912"},

"kevin raj":{"DOB":(17,9,2005),"Phone":"9677295411"},

"muhammad shayaan":{"DOB":(13,11,2005),"Phone":"9798953599"},

"padmesh prasadh":{"DOB":(16,12,2006),"Phone":"7890989887"},

"pravinesh ramprasad":{"DOB":(18,1,2005),"Phone":"7677890567"},

"rahul ganesh":{"DOB":(19,4,2005),"Phone":"8909976775"},

"rajan":{"DOB":(23,6,2005),"Phone":"6756678798"},

"rajesh sukuram":{"DOB":(25,7,2005),"Phone":"9899789334"},

"sabaresh sudhakar":{"DOB":(24,10,2006),"Phone":"6789056732"},

"sai samuel":{"DOB":(6,9,2006),"Phone":"7899098224"},

"shreyas sudip mitra":{"DOB":(17,3,2005),"Phone":"7677878990"},

"sundar raman":{"DOB":(23,4,2005),"Phone":"7812345680"},

"aarthi":{"DOB":(25,4,2005),"Phone":"9088767889"},

"aiswarya rani":{"DOB":(28,6,2006),"Phone":"9788977123"},

"amirtha varshini":{"DOB":(30,2,2005),"Phone":"7098767898"},

"avanthika":{"DOB":(12,7,2006),"Phone":"7677898243"},

"brighty salomi":{"DOB":(15,8,2005),"Phone":"9087612345"},

"diya sriram":{"DOB":(11,9,2005),"Phone":"9086712543"},

"gayathri sudha":{"DOB":(9,10,2006),"Phone":"7890123456"},

"hamsadhwani":{"DOB":(18,11,2005),"Phone":"89055467812"},

"hannah sumesh":{"DOB":(4,12,2005),"Phone":"7878776767"},

"harini raj":{"DOB":(12,5,2006),"Phone":"8787945312"},

"keerthana":{"DOB":(14,3,2005),"Phone":"7654587654"},

"keren david":{"DOB":(3,8,2005),"Phone":"9034567129"},

"madhumitha":{"DOB":(16,9,2005),"Phone":"8907867855"},

"monika sagar":{"DOB":(19,4,2005),"Phone":"8901234567"},

"nithya":{"DOB":(17,2,2005),"Phone":"8788976712"},

"preyanka shankar":{"DOB":(31,11,2005),"Phone":"8765489012"},

"rajeswari narayanan":{"DOB":(14,9,2006),"Phone":"7677890123"},

"soundarya rajan":{"DOB":(5,10,2005),"Phone":"8234589012"},

"sowmya sri":{"DOB":(18,2,2005),"Phone":"9677294567"},

"varsha":{"DOB":(23,3,2005),"Phone":"9840411789"}

}

Details_B={"andrew ezhil":{"DOB":(1,1,2005),"Phone":"1234567890"},
"arvin samuel":{"DOB":(23,12,2005),"Phone":"9876543210"},
"benedict raj":{"DOB":(3,3,2005),"Phone":"2345678901"},
"chandru":{"DOB":(4,4,2005),"Phone":"9876543210"},
"daniel anto benjamine":{"DOB":(5,5,2005),"Phone":"3456789012"},
"elton craig samson":{"DOB":(6,6,2005),"Phone":"4567890123"},
"ezhil adhithya":{"DOB":(1,12,2005),"Phone":"5678901234"},
"ivaniel joshua":{"DOB":(8,8,2005),"Phone":"6789012345"},
"lokeshwar":{"DOB":(9,9,2005),"Phone":"7890123456"},
"karan raj":{"DOB":(10,10,2005),"Phone":"8901234567"},
"kingston richard":{"DOB":(28,6,2005),"Phone":"9012345678"},
"kommal":{"DOB":(12,12,2005),"Phone":"8765432109"},
"lebrio mascarnaz":{"DOB":(13,1,2005),"Phone":"7654321098"},
"sivanjan":{"DOB":(14,2,2005),"Phone":"6543210987"},
"porchelvan":{"DOB":(15,3,2005),"Phone":"5432109876"},
"manda sai nikhil":{"DOB":(16,4,2005),"Phone":"4321098765"},
"siva kailash":{"DOB":(17,5,2005),"Phone":"3210987654"},
"sai pranav":{"DOB":(18,6,2005),"Phone":"2109876543"},
"tarunkumar":{"DOB":(19,7,2005),"Phone":"1098765432"},
"akshita patnaik":{"DOB":(20,8,2005),"Phone":"6543217890"},
"aparnashree":{"DOB":(21,9,2005),"Phone":"5432167890"},
"augustina noel":{"DOB":(22,10,2005),"Phone":"8765009123"},
"bharathi":{"DOB":(23,11,2005),"Phone":"6512390678"},
"bhavadharshni":{"DOB":(24,12,2005),"Phone":"7899012456"},

"deshna":{"DOB":(25,1,2005),"Phone":"7226913402"},

"diyashri":{"DOB":(26,2,2005),"Phone":"8267980021"},

"jeevashree":{"DOB":(27,3,2005),"Phone":"4455889991"},

"jeevika":{"DOB":(28,4,2005),"Phone":"2658900325"},

"jenewar sarah singh":{"DOB":(29,5,2005),"Phone":"2367005912"},

"jyotsna":{"DOB":(30,6,2005),"Phone":"2300056793"},

"kandiz solita":{"DOB":(31,7,2005),"Phone":"9234561099"},

"kavya":{"DOB":(1,8,2005),"Phone":"6729841234"},

"leena":{"DOB":(2,9,2005),"Phone":"9443332167"},

"meher fathima":{"DOB":(3,10,2005),"Phone":"9123004578"},

"mohita":{"DOB":(4,11,2005),"Phone":"7999452190"},

"pon leela percy":{"DOB":(5,12,2005),"Phone":"6390032189"},

"shalini":{"DOB":(6,1,2005),"Phone":"9000656582"},

"sharon david":{"DOB":(7,2,2005),"Phone":"4519863001"},

"sindhuja":{"DOB":(8,3,2005),"Phone":"49871120023"}

}

Details_C={"abishek joseph":{"DOB":(16,10,2005),"Phone":"4911354074"},

"alexander":{"DOB":(1,5,2004),"Phone":"6190527961"},

"ben laurie":{"DOB":(9,10,2005),"Phone":"1478306483"},

"calvin rodrigues":{"DOB":(7,5,2005),"Phone":"4099991109"},

"dinesh kumar":{"DOB":(26,9,2005),"Phone":"8490218651"},

"ethan james":{"DOB":(1,3,2005),"Phone":"6296586901"},

"francis xavier":{"DOB":(12,7,2004),"Phone":"1927324753"},

"kobe jordan":{"DOB":(4,1,2005),"Phone":"6092254249"},

"koushik dev":{"DOB":(23,8,2005),"Phone":"9809384937"},

"logan sprite":{"DOB":(15,4,2004),"Phone":"2329697244"},

"nathan drake":{"DOB":(3,5,2004),"Phone":"6414127976"},

"naufal rizwan":{"DOB":(8,9,2005),"Phone":"2442977436"},

"pradeep kumar":{"DOB":(17,11,2005),"Phone":"2571642644"},

"samuel divine":{"DOB":(25,5,2004),"Phone":"150158082"},

"shawn thomas":{"DOB":(3,3,2005),"Phone":"8748583456"},

"srikanth kumaraesan":{"DOB":(11,7,2005),"Phone":"9784851537"},

"victor sullivan":{"DOB":(20,4,2005),"Phone":"3351846436"},

"vijay yashwanth":{"DOB":(14,9,2006),"Phone":"5544856052"},

"vimal hassan":{"DOB":(6,8,2005),"Phone":"1943505953"},

"vishwa":{"DOB":(18,2,2005),"Phone":"7981618910"},

"alsey christina":{"DOB":(30,3,2005),"Phone":"6731243768"},

"brittany jules":{"DOB":(25,11,2005),"Phone":"9658248368"},

"catherine joel":{"DOB":(7,9,2006),"Phone":"2897810072"},

"daphne princy":{"DOB":(19,4,2005),"Phone":"1496283579"},

"deekshana":{"DOB":(21,12,2006),"Phone":"4406426954"},

"devashree":{"DOB":(10,6,2005),"Phone":"8171551936"},

"deepika neelamegan":{"DOB":(15,3,2005),"Phone":"1820267056"},

"dhisha vijay":{"DOB":(23,5,2005),"Phone":"1352613956"},

"dora maximoff":{"DOB":(30,8,2005),"Phone":"2989935618"},

"isha rajesh":{"DOB":(4,11,2005),"Phone":"6372193191"},

"indhu anand":{"DOB":(25,12,2005),"Phone":"4088655726"},

"jessica shammah":{"DOB":(16,4,2004),"Phone":"5195032002"},

"manasa reddy":{"DOB":(27,8,2006),"Phone":"4044756258"},

"nidhi":{"DOB":(18,11,2005),"Phone":"4023335424"},

"pamila":{"DOB":(9,3,2005),"Phone":"5288559574"},

"prathipa":{"DOB":(28,5,2005),"Phone":"9477262052"},

"saishri":{"DOB":(17,6,2006),"Phone":"1360499768"},

"shreya":{"DOB":(6,2,2005),"Phone":"8294986719"},

"swathi":{"DOB":(31,7,2005),"Phone":"8882787460"},

"tessa thompson":{"DOB":(14,7,2005),"Phone":"8085705429"}

}

Details_D={"alex luke":{"DOB":(23,1,2005),"Phone":"9829181964"},

"bob mark":{"DOB":(21,3,2005),"Phone":"8475920473"},

"carl engels":{"DOB":(3,4,2006),"Phone":"9283728352"},

"dan christ":{"DOB":(5,2,2005),"Phone":"9281237475"},

"ernest gayle":{"DOB":(6,4,2006),"Phone":"8739728287"},

"finn gates":{"DOB":(10,6,2005),"Phone":"9827872878"},

"grim roger":{"DOB":(2,12,2005),"Phone":"9828773000"},

"henderson fox":{"DOB":(12,9,2006),"Phone":"9839828991"},

"ivin grem":{"DOB":(1,7,2006),"Phone":"8382727171"},

"jack dorsen":{"DOB":(13,5,2006),"Phone":"9839828721"},

"kremlin kallus":{"DOB":(7,6,2006),"Phone":"9872091989"},

"lawson lewis":{"DOB":(9,4,2004),"Phone":"9839847637"},

"martin watts":{"DOB":(29,1,2006),"Phone":"9872872677"},

"newland warner":{"DOB":(31,10,2005),"Phone":"9848574832"},

"orlean cook":{"DOB":(25,3,2006),"Phone":"9828837832"},

"peter yottam":{"DOB":(27,2,2005),"Phone":"9856525421"},

"quinton joshua":{"DOB":(18,11,2005),"Phone":"9462426567"},

"rambo jackson":{"DOB":(15,4,2005),"Phone":"9676265542"},

"stuart harley":{"DOB":(24,2,2005),"Phone":"8367265661"},

"tom alstair":{"DOB":(23,4,2006),"Phone":"7265255411"},

"angelina bramwell":{"DOB":(19,5,2005),"Phone":"8675783800"},

"bell rhode":{"DOB":(1,3,2006),"Phone":"8626554151"},

"christine puglisi":{"DOB":(16,9,2005),"Phone":"9261551562"},

"duerre watson":{"DOB":(14,2,2005),"Phone":"9254143256"},

"emma charlotte":{"DOB":(15,4,2006),"Phone":"7153525628"},

"francis red":{"DOB":(6,10,2006),"Phone":"8267716521"},

"gatsby joe":{"DOB":(2,7,2004),"Phone":"9882767181"},

"holly graf":{"DOB":(13,11,2005),"Phone":"8737265182"},

"sofia samuel":{"DOB":(25,7,2006),"Phone":"9867165121"},

"julie orchard":{"DOB":(8,4,2006),"Phone":"8727616511"},

"kate brownwell":{"DOB":(26,5,2006),"Phone":"7827176381"},

"larson brown":{"DOB":(30,8,2005),"Phone":"9882761651"},

"marie joseph":{"DOB":(8,3,2006),"Phone":"7261656352"},

"oleson jinn":{"DOB":(19,12,2005),"Phone":"9872652516"},

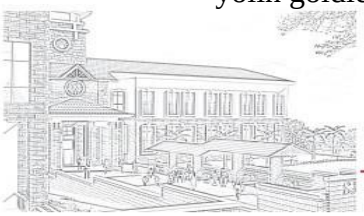
"penrose walter":{"DOB":(10,12,2005),"Phone":"9022725123"},

"rosetta park":{"DOB":(31,3,2005),"Phone":"8265265564"},

"scarlett richard":{"DOB":(4,8,2005),"Phone":"9562461563"},

"thebe kells":{"DOB":(10,9,2006),"Phone":"9652551562"},

"yolin goldie":{"DOB":(23,9,2006),"Phone":"7625254154"},



"zara hunter":{"DOB":(20,5,2005),"Phone":"8176265365"},

}

Attendance_A={"aakash varma":{"Midterm-1":(50,50),"Terminal-1":(42,50),"Midterm-2":(50,50),"Terminal-2":(44,50)},

"abishek":{"Midterm-1":(38,50),"Terminal-1":(43,50),"Midterm-2":(46,50),"Terminal-2":(45,50)},

"anand raj":{"Midterm-1":(31,50),"Terminal-1":(50,50),"Midterm-2":(50,50),"Terminal-2":(37,50)},

"david washington":{"Midterm-1":(41,50),"Terminal-1":(47,50),"Midterm-2":(36,50),"Terminal-2":(50,50)},

"emannuel":{"Midterm-1":(33,50),"Terminal-1":(41,50),"Midterm-2":(38,50),"Terminal-2":(41,50)},

"harley sumesh":{"Midterm-1":(48,50),"Terminal-1":(43,50),"Midterm-2":(45,50),"Terminal-2":(44,50)},

"karthikeya airpirala":{"Midterm-1":(46,50),"Terminal-1":(45,50),"Midterm-2":(33,50),"Terminal-2":(46,50)},

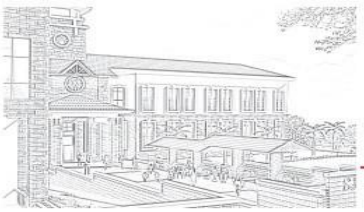
"kaushik gautham":{"Midterm-1":(50,50),"Terminal-1":(46,50),"Midterm-2":(31,50),"Terminal-2":(47,50)},

"kavin raj":{"Midterm-1":(42,50),"Terminal-1":(48,50),"Midterm-2":(50,50),"Terminal-2":(48,50)},

"kevin raj":{"Midterm-1":(48,50),"Terminal-1":(49,50),"Midterm-2":(39,50),"Terminal-2":(48,50)},

"muhammad shayaan":{"Midterm-1":(47,50),"Terminal-1":(33,50),"Midterm-2":(45,50),"Terminal-2":(47,50)},

"padmesh prasadh":{"Midterm-1":(49,50),"Terminal-1":(38,50),"Midterm-2":(38,50),"Terminal-2":(49,50)},



"pravinesh ramprasad":{"Midterm-1":(38,50),"Terminal-1":(39,50),"Midterm-2":(39,50),"Terminal-2":(38,50)},

"rahul ganesh":{"Midterm-1":(40,50),"Terminal-1":(37,50),"Midterm-2":(37,50),"Terminal-2":(40,50)},

"rajan":{"Midterm-1":(41,50),"Terminal-1":(40,50),"Midterm-2":(40,50),"Terminal-2":(41,50)},

"rajesh sukuram":{"Midterm-1":(47,50),"Terminal-1":(41,50),"Midterm-2":(41,50),"Terminal-2":(47,50)},

"sabaresh sudhakar":{"Midterm-1":(38,50),"Terminal-1":(35,50),"Midterm-2":(35,50),"Terminal-2":(38,50)},

"sai samuel":{"Midterm-1":(34,50),"Terminal-1":(49,50),"Midterm-2":(49,50),"Terminal-2":(34,50)},

"shreyas sudip mitra":{"Midterm-1":(50,50),"Terminal-1":(50,50),"Midterm-2":(50,50),"Terminal-2":(50,50)},

"sundar raman":{"Midterm-1":(45,50),"Terminal-1":(50,50),"Midterm-2":(50,50),"Terminal-2":(45,50)},

"aarthi":{"Midterm-1":(46,50),"Terminal-1":(50,50),"Midterm-2":(50,50),"Terminal-2":(46,50)},

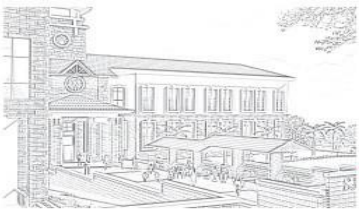
"aiswarya rani":{"Midterm-1":(47,50),"Terminal-1":(41,50),"Midterm-2":(41,50),"Terminal-2":(47,50)},

"amirtha varshini":{"Midterm-1":(44,50),"Terminal-1":(42,50),"Midterm-2":(42,50),"Terminal-2":(44,50)},

"avanthika":{"Midterm-1":(42,50),"Terminal-1":(45,50),"Midterm-2":(45,50),"Terminal-2":(42,50)},

"brighty salomi":{"Midterm-1":(41,50),"Terminal-1":(42,50),"Midterm-2":(42,50),"Terminal-2":(41,50)},

"diya sriram":{"Midterm-1":(34,50),"Terminal-1":(45,50),"Midterm-2":(45,50),"Terminal-2":(34,50)},



"gayathri sudha":{"Midterm-1":(35,50),"Terminal-1":(50,50),"Midterm-2":(50,50),"Terminal-2":(35,50)},

"hamsadhwani":{"Midterm-1":(37,50),"Terminal-1":(50,50),"Midterm-2":(50,50),"Terminal-2":(37,50)},

"hannah sumesh":{"Midterm-1":(35,50),"Terminal-1":(38,50),"Midterm-2":(38,50),"Terminal-2":(35,50)},

"harini raj":{"Midterm-1":(38,50),"Terminal-1":(39,50),"Midterm-2":(39,50),"Terminal-2":(38,50)},

"keerthana":{"Midterm-1":(31,50),"Terminal-1":(40,50),"Midterm-2":(40,50),"Terminal-2":(31,50)},

"keren david":{"Midterm-1":(40,50),"Terminal-1":(41,50),"Midterm-2":(41,50),"Terminal-2":(40,50)},

"madhumitha":{"Midterm-1":(41,50),"Terminal-1":(44,50),"Midterm-2":(44,50),"Terminal-2":(41,50)},

"monika sagar":{"Midterm-1":(42,50),"Terminal-1":(46,50),"Midterm-2":(46,50),"Terminal-2":(42,50)},

"nithya":{"Midterm-1":(43,50),"Terminal-1":(50,50),"Midterm-2":(50,50),"Terminal-2":(43,50)},

"preyanka shankar":{"Midterm-1":(50,50),"Terminal-1":(47,50),"Midterm-2":(47,50),"Terminal-2":(50,50)},

"rajeswari narayanan":{"Midterm-1":(50,50),"Terminal-1":(50,50),"Midterm-2":(50,50),"Terminal-2":(50,50)},

"soundarya rajan":{"Midterm-1":(43,50),"Terminal-1":(42,50),"Midterm-2":(42,50),"Terminal-2":(43,50)},

"sowmya sri":{"Midterm-1":(42,50),"Terminal-1":(45,50),"Midterm-2":(45,50),"Terminal-2":(42,50)},

"varsha":{"Midterm-1":(44,50),"Terminal-1":(46,50),"Midterm-2":(46,50),"Terminal-2":(44,50)}

}

Attendance_B={"andrew ezhil":{"Midterm-1":(32,50),"Terminal-1":(22,50),"Midterm-2":(37,50),"Terminal-2":(45,50)},

"arvin samuel":{"Midterm-1":(45,50),"Terminal-1":(33,50),"Midterm-2":(48,50),"Terminal-2":(27,50)},

"benedict raj":{"Midterm-1":(49,50),"Terminal-1":(44,50),"Midterm-2":(43,50),"Terminal-2":(49,50)},

"chandru":{"Midterm-1":(48,50),"Terminal-1":(47,50),"Midterm-2":(41,50),"Terminal-2":(37,50)},

"daniel anto benjamine":{"Midterm-1":(41,50),"Terminal-1":(36,50),"Midterm-2":(50,50),"Terminal-2":(32,50)},

"elton craig samson":{"Midterm-1":(50,50),"Terminal-1":(25,50),"Midterm-2":(34,50),"Terminal-2":(49,50)},

"ezhil adhithya":{"Midterm-1":(40,50),"Terminal-1":(32,50),"Midterm-2":(42,50),"Terminal-2":(29,50)},

"ivaniel joshua":{"Midterm-1":(56,50),"Terminal-1":(20,50),"Midterm-2":(32,50),"Terminal-2":(43,50)},

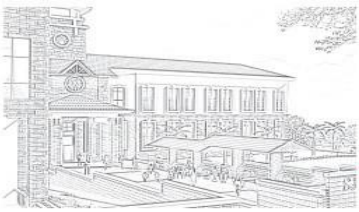
"lokeshwar":{"Midterm-1":(34,50),"Terminal-1":(31,50),"Midterm-2":(46,50),"Terminal-2":(39,50)},

"karan raj":{"Midterm-1":(23,50),"Terminal-1":(37,50),"Midterm-2":(40,50),"Terminal-2":(26,50)},

"kingston richard":{"Midterm-1":(43,50),"Terminal-1":(46,50),"Midterm-2":(45,50),"Terminal-2":(45,50)},

"kommal":{"Midterm-1":(22,50),"Terminal-1":(56,50),"Midterm-2":(47,50),"Terminal-2":(39,50)},

"lebrio mascarnaz":{"Midterm-1":(33,50),"Terminal-1":(32,50),"Midterm-2":(49,50),"Terminal-2":(40,50)},



"sivanjan":{"Midterm-1":(45,50),"Terminal-1":(40,50),"Midterm-2":(32,50),"Terminal-2":(42,50)},

"porchelvan":{"Midterm-1":(32,50),"Terminal-1":(45,50),"Midterm-2":(50,50),"Terminal-2":(41,50)},

"manda sai nikhil":{"Midterm-1":(37,50),"Terminal-1":(42,50),"Midterm-2":(41,50),"Terminal-2":(47,50)},

"siva kailash":{"Midterm-1":(23,50),"Terminal-1":(39,50),"Midterm-2":(27,50),"Terminal-2":(48,50)},

"sai pranav":{"Midterm-1":(39,50),"Terminal-1":(50,50),"Midterm-2":(45,50),"Terminal-2":(37,50)},

"tarunkumar":{"Midterm-1":(40,50),"Terminal-1":(42,50),"Midterm-2":(21,50),"Terminal-2":(36,50)},

"akshita patnaik":{"Midterm-1":(41,50),"Terminal-1":(43,50),"Midterm-2":(49,50),"Terminal-2":(41,50)},

"aparnashree":{"Midterm-1":(42,50),"Terminal-1":(49,50),"Midterm-2":(43,50),"Terminal-2":(33,50)},

"augustina noel":{"Midterm-1":(43,50),"Terminal-1":(34,50),"Midterm-2":(42,50),"Terminal-2":(27,50)},

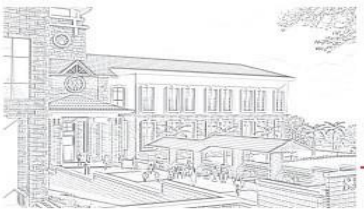
"bharathi":{"Midterm-1":(44,50),"Terminal-1":(31,50),"Midterm-2":(40,50),"Terminal-2":(50,50)},

"bhavadharshni":{"Midterm-1":(31,50),"Terminal-1":(47,50),"Midterm-2":(49,50),"Terminal-2":(29,50)},

"deshna":{"Midterm-1":(30,50),"Terminal-1":(36,50),"Midterm-2":(32,50),"Terminal-2":(31,50)},

"diyashri":{"Midterm-1":(27,50),"Terminal-1":(35,50),"Midterm-2":(40,50),"Terminal-2":(33,50)},

"jeevashree":{"Midterm-1":(50,50),"Terminal-1":(49,50),"Midterm-2":(43,50),"Terminal-2":(48,50)},



"jeevika":{"Midterm-1":(45,50),"Terminal-1":(34,50),"Midterm-2":(26,50),"Terminal-2":(50,50)},

"jenewar sarah singh":{"Midterm-1":(49,50),"Terminal-1":(50,50),"Midterm-2":(49,50),"Terminal-2":(45,50)},

"jyotsna":{"Midterm-1":(43,50),"Terminal-1":(23,50),"Midterm-2":(37,50),"Terminal-2":(33,50)},

"kandiz solita":{"Midterm-1":(37,50),"Terminal-1":(42,50),"Midterm-2":(48,50),"Terminal-2":(89,50)},

"kavya":{"Midterm-1":(30,50),"Terminal-1":(47,50),"Midterm-2":(27,50),"Terminal-2":(33,50)},

"leena":{"Midterm-1":(40,50),"Terminal-1":(50,50),"Midterm-2":(38,50),"Terminal-2":(20,50)},

"meher fathima":{"Midterm-1":(43,50),"Terminal-1":(35,50),"Midterm-2":(39,50),"Terminal-2":(42,50)},

"mohita":{"Midterm-1":(44,50),"Terminal-1":(31,50),"Midterm-2":(40,50),"Terminal-2":(47,50)},

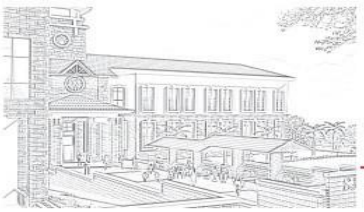
"pon leela percy":{"Midterm-1":(45,50),"Terminal-1":(39,50),"Midterm-2":(42,50),"Terminal-2":(33,50)},

"shalini":{"Midterm-1":(47,50),"Terminal-1":(28,50),"Midterm-2":(44,50),"Terminal-2":(44,50)},

"sharon david":{"Midterm-1":(48,50),"Terminal-1":(36,50),"Midterm-2":(50,50),"Terminal-2":(23,50)},

"sindhuja":{"Midterm-1":(21,50),"Terminal-1":(39,50),"Midterm-2":(52,50),"Terminal-2":(38,50)}

}



Attendance_C={"abishek joseph":{"Midterm-1":(33,50),"Terminal-1":(50,50),"Midterm-2":(42,50),"Terminal-2":(43,50)},

"alexander":{"Midterm-1":(35,50),"Terminal-1":(31,50),"Midterm-2":(32,50),"Terminal-2":(34,50)},

"ben laurie":{"Midterm-1":(50,50),"Terminal-1":(46,50),"Midterm-2":(31,50),"Terminal-2":(39,50)},

"calvin rodrigues":{"Midterm-1":(31,50),"Terminal-1":(31,50),"Midterm-2":(31,50),"Terminal-2":(39,50)},

"dinesh kumar":{"Midterm-1":(32,50),"Terminal-1":(33,50),"Midterm-2":(34,50),"Terminal-2":(39,50)},

"ethan james":{"Midterm-1":(45,50),"Terminal-1":(43,50),"Midterm-2":(43,50),"Terminal-2":(30,50)},

"francis xavier":{"Midterm-1":(34,50),"Terminal-1":(33,50),"Midterm-2":(49,50),"Terminal-2":(49,50)},

"kobe jordan":{"Midterm-1":(37,50),"Terminal-1":(47,50),"Midterm-2":(30,50),"Terminal-2":(30,50)},

"koushik dev":{"Midterm-1":(49,50),"Terminal-1":(47,50),"Midterm-2":(35,50),"Terminal-2":(39,50)},

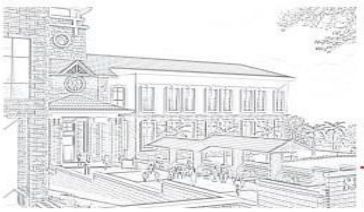
"logan sprite":{"Midterm-1":(31,50),"Terminal-1":(43,50),"Midterm-2":(42,50),"Terminal-2":(38,50)},

"nathan drake":{"Midterm-1":(43,50),"Terminal-1":(46,50),"Midterm-2":(33,50),"Terminal-2":(50,50)},

"naufal rizwan":{"Midterm-1":(44,50),"Terminal-1":(39,50),"Midterm-2":(35,50),"Terminal-2":(46,50)},

"pradeep kumar":{"Midterm-1":(49,50),"Terminal-1":(49,50),"Midterm-2":(34,50),"Terminal-2":(48,50)},

"samuel divine":{"Midterm-1":(50,50),"Terminal-1":(39,50),"Midterm-2":(37,50),"Terminal-2":(43,50)},



"shawn thomas":{"Midterm-1":(37,50),"Terminal-1":(39,50),"Midterm-2":(50,50),"Terminal-2":(42,50)},

"srikanth kumaraesan":{"Midterm-1":(32,50),"Terminal-1":(35,50),"Midterm-2":(37,50),"Terminal-2":(37,50)},

"victor sullivan":{"Midterm-1":(47,50),"Terminal-1":(32,50),"Midterm-2":(37,50),"Terminal-2":(30,50)},

"vijay yashwanth":{"Midterm-1":(30,50),"Terminal-1":(47,50),"Midterm-2":(30,50),"Terminal-2":(50,50)},

"vimal hassan":{"Midterm-1":(38,50),"Terminal-1":(43,50),"Midterm-2":(45,50),"Terminal-2":(46,50)},

"vishwa":{"Midterm-1":(42,50),"Terminal-1":(48,50),"Midterm-2":(45,50),"Terminal-2":(39,50)},

"alsey christina":{"Midterm-1":(30,50),"Terminal-1":(35,50),"Midterm-2":(37,50),"Terminal-2":(30,50)},

"brittany jules":{"Midterm-1":(40,50),"Terminal-1":(50,50),"Midterm-2":(37,50),"Terminal-2":(40,50)},

"catherine joel":{"Midterm-1":(30,50),"Terminal-1":(42,50),"Midterm-2":(45,50),"Terminal-2":(43,50)},

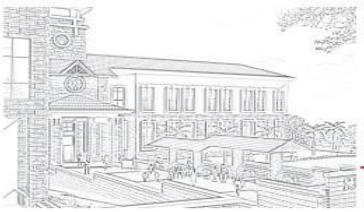
"daphne princy":{"Midterm-1":(31,50),"Terminal-1":(33,50),"Midterm-2":(38,50),"Terminal-2":(37,50)},

"deekshana":{"Midterm-1":(34,50),"Terminal-1":(41,50),"Midterm-2":(50,50),"Terminal-2":(33,50)},

"devashree":{"Midterm-1":(43,50),"Terminal-1":(49,50),"Midterm-2":(30,50),"Terminal-2":(46,50)},

"deepika neelamegan":{"Midterm-1":(32,50),"Terminal-1":(33,50),"Midterm-2":(42,50),"Terminal-2":(47,50)},

"dhisha vijay":{"Midterm-1":(40,50),"Terminal-1":(33,50),"Midterm-2":(46,50),"Terminal-2":(42,50)},



"dora maximoff":{"Midterm-1":(43,50),"Terminal-1":(43,50),"Midterm-2":(46,50),"Terminal-2":(31,50)},

"isha rajesh":{"Midterm-1":(40,50),"Terminal-1":(30,50),"Midterm-2":(32,50),"Terminal-2":(40,50)},

"indhu anand":{"Midterm-1":(32,50),"Terminal-1":(33,50),"Midterm-2":(34,50),"Terminal-2":(49,50)},

"jessica shammah":{"Midterm-1":(39,50),"Terminal-1":(34,50),"Midterm-2":(34,50),"Terminal-2":(37,50)},

"manasa reddy":{"Midterm-1":(45,50),"Terminal-1":(44,50),"Midterm-2":(48,50),"Terminal-2":(42,50)},

"nidhi":{"Midterm-1":(30,50),"Terminal-1":(48,50),"Midterm-2":(49,50),"Terminal-2":(30,50)},

"pamila":{"Midterm-1":(42,50),"Terminal-1":(43,50),"Midterm-2":(42,50),"Terminal-2":(37,50)},

"prathipa":{"Midterm-1":(37,50),"Terminal-1":(50,50),"Midterm-2":(41,50),"Terminal-2":(34,50)},

"saishri":{"Midterm-1":(33,50),"Terminal-1":(40,50),"Midterm-2":(40,50),"Terminal-2":(44,50)},

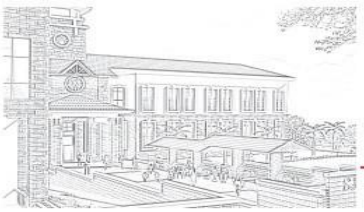
"shreya":{"Midterm-1":(39,50),"Terminal-1":(30,50),"Midterm-2":(40,50),"Terminal-2":(39,50)},

"swathi":{"Midterm-1":(43,50),"Terminal-1":(36,50),"Midterm-2":(34,50),"Terminal-2":(32,50)},

"tessa thompson":{"Midterm-1":(37,50),"Terminal-1":(44,50),"Midterm-2":(45,50),"Terminal-2":(37,50)}

}

Attendance_D={"alex luke":{"Midterm-1":(45,50),"Terminal-1":(32,50),"Midterm-2":(41,50),"Terminal-2":(33,50)},



"bob mark":{"Midterm-1":(45,50),"Terminal-1":(22,50),"Midterm-2":(50,50),"Terminal-2":(45,50)},

"carl engels":{"Midterm-1":(45,50),"Terminal-1":(37,50),"Midterm-2":(45,50),"Terminal-2":(35,50)},

"dan christ":{"Midterm-1":(50,50),"Terminal-1":(33,50),"Midterm-2":(39,50),"Terminal-2":(32,50)},

"ernest gayle":{"Midterm-1":(33,50),"Terminal-1":(23,50),"Midterm-2":(26,50),"Terminal-2":(33,50)},

"finn gates":{"Midterm-1":(50,50),"Terminal-1":(44,50),"Midterm-2":(46,50),"Terminal-2":(37,50)},

"grim roger":{"Midterm-1":(45,50),"Terminal-1":(24,50),"Midterm-2":(30,50),"Terminal-2":(49,50)},

"henderson fox":{"Midterm-1":(34,50),"Terminal-1":(42,50),"Midterm-2":(39,50),"Terminal-2":(38,50)},

"ivin grem":{"Midterm-1":(22,50),"Terminal-1":(44,50),"Midterm-2":(41,50),"Terminal-2":(41,50)},

"jack dorsen":{"Midterm-1":(42,50),"Terminal-1":(33,50),"Midterm-2":(26,50),"Terminal-2":(31,50)},

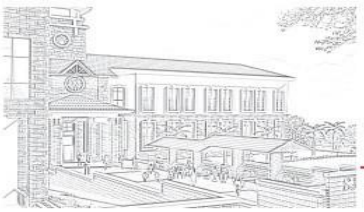
"kremlin kallus":{"Midterm-1":(32,50),"Terminal-1":(47,50),"Midterm-2":(22,50),"Terminal-2":(46,50)},

"lawson lewis":{"Midterm-1":(34,50),"Terminal-1":(30,50),"Midterm-2":(27,50),"Terminal-2":(40,50)},

"martin watts":{"Midterm-1":(22,50),"Terminal-1":(31,50),"Midterm-2":(45,50),"Terminal-2":(34,50)},

"newland warner":{"Midterm-1":(44,50),"Terminal-1":(50,50),"Midterm-2":(49,50),"Terminal-2":(41,50)},

"orlean cook":{"Midterm-1":(21,50),"Terminal-1":(30,50),"Midterm-2":(35,50),"Terminal-2":(21,50)},



"peter yottam":{"Midterm-1":(49,50),"Terminal-1":(34,50),"Midterm-2":(21,50),"Terminal-2":(32,50)},

"quinton joshua":{"Midterm-1":(29,50),"Terminal-1":(34,50),"Midterm-2":(22,50),"Terminal-2":(45,50)},

"rambo jackson":{"Midterm-1":(39,50),"Terminal-1":(39,50),"Midterm-2":(30,50),"Terminal-2":(47,50)},

"stuart harley":{"Midterm-1":(50,50),"Terminal-1":(42,50),"Midterm-2":(47,50),"Terminal-2":(32,50)},

"tom alstair":{"Midterm-1":(49,50),"Terminal-1":(45,50),"Midterm-2":(41,50),"Terminal-2":(43,50)},

"angelina bramwell":{"Midterm-1":(46,50),"Terminal-1":(33,50),"Midterm-2":(43,50),"Terminal-2":(25,50)},

"bell rhode":{"Midterm-1":(37,50),"Terminal-1":(37,50),"Midterm-2":(46,50),"Terminal-2":(49,50)},

"christine puglisi":{"Midterm-1":(50,50),"Terminal-1":(34,50),"Midterm-2":(21,50),"Terminal-2":(45,50)},

"duerre watson":{"Midterm-1":(37,50),"Terminal-1":(45,50),"Midterm-2":(43,50),"Terminal-2":(29,50)},

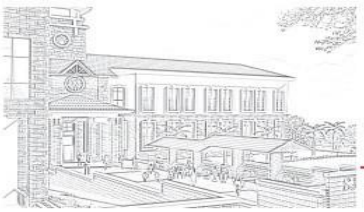
"emma charlotte":{"Midterm-1":(30,50),"Terminal-1":(30,50),"Midterm-2":(20,50),"Terminal-2":(37,50)},

"francis red":{"Midterm-1":(40,50),"Terminal-1":(42,50),"Midterm-2":(49,50),"Terminal-2":(29,50)},

"gatsby joe":{"Midterm-1":(39,50),"Terminal-1":(38,50),"Midterm-2":(40,50),"Terminal-2":(41,50)},

"holly graf":{"Midterm-1":(28,50),"Terminal-1":(32,50),"Midterm-2":(38,50),"Terminal-2":(29,50)},

"sofia samuel":{"Midterm-1":(48,50),"Terminal-1":(21,50),"Midterm-2":(33,50),"Terminal-2":(37,50)},



"julie orchard":{"Midterm-1":(40,50),"Terminal-1":(29,50),"Midterm-2":(48,50),"Terminal-2":(27,50)},

"kate brownwell":{"Midterm-1":(29,50),"Terminal-1":(24,50),"Midterm-2":(31,50),"Terminal-2":(42,50)},

"larson brown":{"Midterm-1":(34,50),"Terminal-1":(24,50),"Midterm-2":(25,50),"Terminal-2":(28,50)},

"marie joseph":{"Midterm-1":(24,50),"Terminal-1":(37,50),"Midterm-2":(27,50),"Terminal-2":(46,50)},

"oleson jinn":{"Midterm-1":(25,50),"Terminal-1":(32,50),"Midterm-2":(46,50),"Terminal-2":(32,50)},

"penrose walter":{"Midterm-1":(46,50),"Terminal-1":(47,50),"Midterm-2":(48,50),"Terminal-2":(26,50)},

"rosetta park":{"Midterm-1":(38,50),"Terminal-1":(49,50),"Midterm-2":(47,50),"Terminal-2":(37,50)},

"scarlett richard":{"Midterm-1":(39,50),"Terminal-1":(40,50),"Midterm-2":(36,50),"Terminal-2":(37,50)},

"thebe kells":{"Midterm-1":(50,50),"Terminal-1":(21,50),"Midterm-2":(24,50),"Terminal-2":(28,50)},

"yolin goldie":{"Midterm-1":(39,50),"Terminal-1":(27,50),"Midterm-2":(41,50),"Terminal-2":(39,50)},

"zara hunter":{"Midterm-1":(24,50),"Terminal-1":(23,50),"Midterm-2":(47,50),"Terminal-2":(20,50)},

}

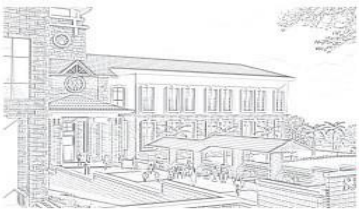
Teachers={"Swarup":("Gnana Swarup","Agniforever",("A","B"),"Physics"),

"Vijayan":("Vijayan","vijay@134",("A","B"),"Maths","CT"),

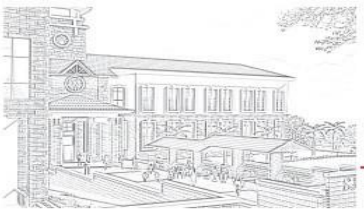
"Aravinth":("Aravinth Kalyani","AraKal90",("A","B"),"Chemistry","CT"),

"Elsa":("Elsa Packiam","elsa@93",("C","D"),"English"),

```
"Gemcy":("Gemcy Ebenezer","Gemeb@112",("B",),"Biology","CT"),
"Shiny":("Shiny Boaz","shinythebest",("A","B"),"English"),
"Sabina":("Sabina Begum","sab@45_9",("A",),"Computer Science"),
"Mehtab":("Mehtab","mehtab3000",("C",),"Accountancy"),
"Mary":("Mary Louis","marylovesgod",("C",),"Entrepreneurship"),
"Monica":("Monica Samuel","monica1974",("C","D"),"Economics","CT"),
"Deepika":("Deepika","Deepikapassionforlife",("C","D"),"Business Studies"),
"Ellmona":("Ellmona Joean","EllShawn123#",("D",),"History"),
"Justin":("Justin Selvaraj","justin6798^",("D",),"Geography")
}


Class_Teachers={"A":"Gemcy Ebenezer",
"B":"Aravinth Kalyani",
"C":"Vijayan",
"D":"Monica Samuel"
}


Sub={"A":("English", "Physics", "Chemistry", "Mathematics", "Biology"),
"B":("English", "Physics", "Chemistry", "Mathematics", "Computer Science"),
"C":("English", "Accountancy", "Entrepreneurship","Economics", "Business Studies"),
"D":("English", "History", "Geography", "Economics", "Buissness Studies")
}
```

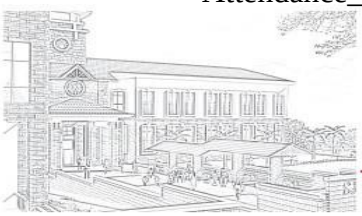


```
def Write():
    File=open("Data.bin", "wb")
    pickle.dump(Marks_A, File)
    pickle.dump(Marks_B, File)
    pickle.dump(Marks_C, File)
    pickle.dump(Marks_D, File)
    pickle.dump(Details_A, File)
    pickle.dump(Details_B, File)
    pickle.dump(Details_C, File)
    pickle.dump(Details_D, File)
    pickle.dump(Attendance_A, File)
    pickle.dump(Attendance_B, File)
    pickle.dump(Attendance_C, File)
    pickle.dump(Attendance_D, File)
    pickle.dump(Teachers, File)
    pickle.dump(Class_Teachers, File)
    pickle.dump(Sub, File)
    File.close()

Write()
```

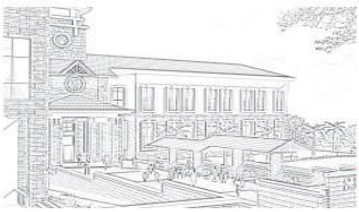


Framework.py:

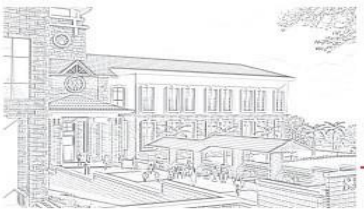
```
import pickle

import numpy as np

import plotly.graph_objects as go

import plotly.offline as pyo

import plotly.express as px

import pandas as pd

from jinja2 import Environment, FileSystemLoader

import webbrowser

from PIL import Image

from copy import deepcopy


File=open("Data.bin", "rb")


Marks_A=pickle.load(File)

Marks_B=pickle.load(File)

Marks_C=pickle.load(File)

Marks_D=pickle.load(File)

Details_A=pickle.load(File)

Details_B=pickle.load(File)

Details_C=pickle.load(File)

Details_D=pickle.load(File)

Attendance_A=pickle.load(File)

Attendance_B=pickle.load(File)
```



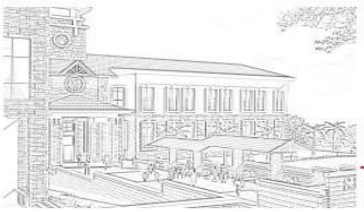
```
Attendance_C=pickle.load(File)
Attendance_D=pickle.load(File)
Teachers=pickle.load(File)
Class_Teachers=pickle.load(File)
Sub=pickle.load(File)

File.close()

class Student:
    def __init__(self, Student_Name):
        global _Name

        def Total():
            self.Total={}

            for i in self.Marks:
                Dup={}
                for j in self.Marks[i]:
                    try:
                        Dup[j]=sum(self.Marks[i][j])
                    except:
                        pass

                self.Total[i]=Dup
```



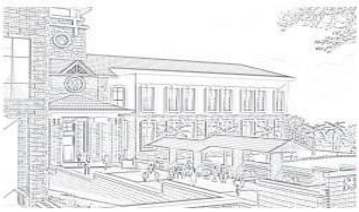
```
    del self.Total["Section"]

_Name=Student_Name
self.Name=Student_Name

if self.Name in Marks_A:
    self.Marks=Marks_A
    Total()
    self.Details=Details_A
    self.Attend=Attendance_A
    self.Sec=self.Marks["Section"]
    self.Login_S=True

elif self.Name in Marks_B:
    self.Marks=Marks_B
    Total()
    self.Details=Details_B
    self.Attend=Attendance_B
    self.Sec=self.Marks["Section"]
    self.Login_S=True

elif self.Name in Marks_C:
    self.Marks=Marks_C
    Total()
    self.Details=Details_C
```



```
        self.Attend=Attendance_C
        self.Sec=self.Marks["Section"]
        self.Login_S=True

    elif self.Name in Marks_D:
        self.Marks=Marks_D
        Total()
        self.Details=Details_D
        self.Attend=Attendance_D
        self.Sec=self.Marks["Section"]
        self.Login_S=True

    else:
        self.Login_S=False


def Login(self, DOB, Phone):
    global Login_S
    self.Dob=DOB
    self.Phone_No=Phone

    if self.Login_S:
        if self.Dob==self.Details[self.Name]["DOB"] and
        self.Phone_No==self.Details[self.Name]["Phone"]:
            Login_S=True
```

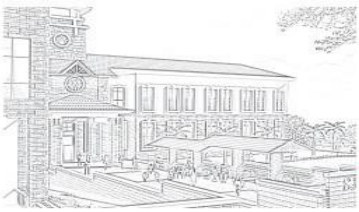


```
        else:
            Login_S=False

    else:
        Login_S=False

def Exam(self, Exam):
    self.Exam=Exam

def Percentage(self):
    Total_Mark=self.Total[self.Name][self.Exam]

    self.Percent=round((Total_Mark/500)*100, 2)

def Rank(self):
    Ranking=[]
    Ranks=[]
    Num=1

    for i in self.Total:
        Ranking.append((i,self.Total[i][self.Exam]))
        Ranks.append([Num])
        Num+=1

    Ranking=np.array(Ranking)
```



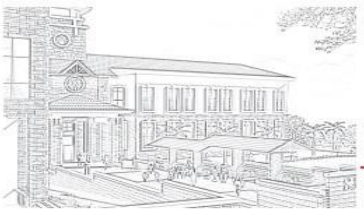
```
    Ranking=sorted(Ranking, key=lambda x:x[1], reverse=True)
    Ranking=np.concatenate((Ranking,Ranks), axis=1)

    Var=[0,0,0]

    for j in Ranking:
        if j[1]==Var[1]:
            Ranking[(int(j[2])-1),2]=Var[2]

        else:
            Var=j

    for k in Ranking:
        if self.Name in k:
            self.Rank_Info=k

def Attendance(self):
    self.Atten=self.Attend[self.Name][self.Exam]
    self.Atten_Percent=round((self.Atten[0]/self.Atten[1])*100,2)

def Grade(self):
    Subject_Marks=self.Marks[self.Name][self.Exam]
    self.Grade=()

    for x in Subject_Marks:
```

```
if 90<x<=100:
    self.Grade+=("A1",)

elif 80<x<=90:
    self.Grade+=("A2",)

elif 70<x<=80:
    self.Grade+=("B1",)

elif 60<x<=70:
    self.Grade+=("B2",)

elif 50<x<=60:
    self.Grade+=("C1",)

elif 40<x<=50:
    self.Grade+=("C2",)

elif 33<x<=40:
    self.Grade+=("D",)

else:
    self.Grade+=("E",)
```



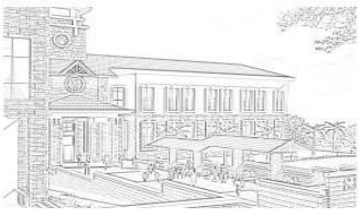
```
def Report_Card(self):
    Student.Grade(self)
    Student.Attendance(self)
    Student.Percentage(self)
    Student.Rank(self)

    Report=[{}]

    Report[0]["Exam"]=self.Exam
    Report[0]["Name"]=self.Name.title()
    Report[0]["Sec"]=self.Sec
    Report[0]["Teacher"]=Class_Teachers[self.Sec]
    Report[0]["Date"]=self.Details[self.Name]["DOB"][0]
    Report[0]["Month"]=self.Details[self.Name]["DOB"][1]
    Report[0]["Year"]=self.Details[self.Name]["DOB"][2]
    Report[0]["Subject"]=Sub[self.Sec]
    Report[0]["Marks"]=self.Marks[self.Name][self.Exam]
    Report[0]["Grade"]=self.Grade
    Report[0]["Total"]=self.Total[self.Name][self.Exam]
    Report[0]["Percent"]=self.Percent
    Report[0]["Rank"]=self.Rank_Info[2]
    Report[0]["Atten"]=self.Atten
    Report[0]["Atten_Percent"]=self.Atten_Percent
    Report=list(Report)
```

```
    file_loader = FileSystemLoader(r'C:\Users\dell\OneDrive\Desktop\IP\2022-2023\CSc IP')
    env = Environment(loader=file_loader)

    template = env.get_template('Report_Card.html')
    output = template.render(content=Report)

    file=open('Report.html', 'w')
    file.write(output)
    file.close()

    webbrowser.open('Report.html')


def Total_Bar(self):

    Data={"Total":[self.Total[self.Name]["Midterm-1"],self.Total[self.Name]["Terminal-
    1"],self.Total[self.Name]["Midterm-2"],self.Total[self.Name]["Terminal-2"]]}

    Df=pd.DataFrame(Data, index=["Midterm-1", "Terminal-1", "Midterm-2", "Terminal-2"])

    TotalBar=px.bar(Df,title="Total Marks of across all Examinations of " + self.Name.title(),
    text_auto=True)
    TotalBar.show()

def Overall_Radar(self):
```

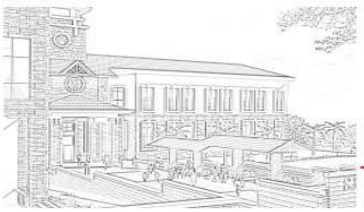


```
Subjects=np.array(Sub[self.Sec])
Subjects=np.array([*Subjects, Subjects[0]])

Midterm_1=np.array(self.Marks[self.Name]["Midterm-1"])
Terminal_1=np.array(self.Marks[self.Name]["Terminal-1"])
Midterm_2=np.array(self.Marks[self.Name]["Midterm-2"])
Terminal_2=np.array(self.Marks[self.Name]["Terminal-2"])

Midterm_1=np.array([*Midterm_1, Midterm_1[0]])
Terminal_1=np.array([*Terminal_1, Terminal_1[0]])
Midterm_2=np.array([*Midterm_2, Midterm_2[0]])
Terminal_2=np.array([*Terminal_2, Terminal_2[0]])

Radar=go.Figure(
    data=[go.Scatterpolar(r=Midterm_1, theta=Subjects, fill='toself', name="Midterm-1"),
        go.Scatterpolar(r=Terminal_1, theta=Subjects, fill='toself', name="Terminal-1"),
        go.Scatterpolar(r=Midterm_2, theta=Subjects, fill='toself', name="Midterm-2"),
        go.Scatterpolar(r=Terminal_2, theta=Subjects, fill='toself', name="Terminal-2")],

    layout=go.Layout(title=go.layout.Title(text="Overall Marks Comparison of " +
    self.Name.title() + " in the Academic Year 2022-23"), polar={"radialaxis": {"visible":
    True}}, showlegend=True)
    )

Radar.show()
```

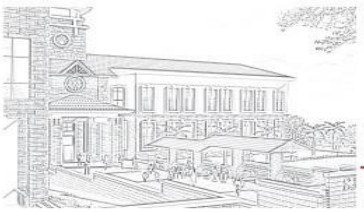
```
def Pie_Marks_Distribution(self):

    All_Marks=self.Marks[self.Name]["Midterm-1"] + self.Marks[self.Name]["Terminal-1"]
    + self.Marks[self.Name]["Midterm-2"] + self.Marks[self.Name]["Terminal-2"]

    Lower_Limit=[0,10,20,30,40,50,60,70,80,90]
    Max=100

    Var1=np.histogram(All_Marks, Lower_Limit+[Max])[0]

    Var2=[]
    Var3=["0-10","10-20","20-30","30-40","40-50","50-60","60-70","70-80","80-90","90-
    100"]

    for x in Var1:
        x=(x/20)*100
        Var2+=[x]

    Distribution=[]
    Class_Intervals=[]
    Counter=0

    while Counter<len(Var2):
        if Var2[Counter]>0:
```



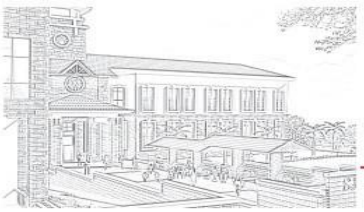
```
            Distribution.append(Var2[Counter])
            Class_Intervals.append(Var3[Counter])

        Counter+=1

    Distribution=Distribution[::-1]
    Class_Intervals=Class_Intervals[::-1]

    Data={"Distribution":Distribution,
            "Class_Intervals":Class_Intervals}

    Df=pd.DataFrame(Data)

    Pie=px.pie(Df, title="Marks Distribution of " + self.Name.title(), values="Distribution",
    names="Class_Intervals")
    Pie.update_traces(textposition='inside', textinfo='percent+label')
    Pie.show()

def Subject_Bar(self):

    Data={"Midterm-1":self.Marks[self.Name]["Midterm-1"],
        "Terminal-1":self.Marks[self.Name]["Terminal-1"],
        "Midterm-2":self.Marks[self.Name]["Midterm-2"],
        "Terminal-2":self.Marks[self.Name]["Terminal-2"]}
```



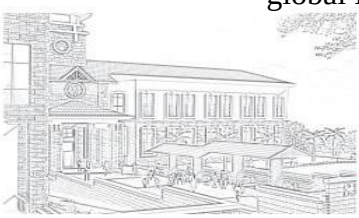
```
        Df=pd.DataFrame(Data, index=Sub[self.Sec])


        Bar=px.bar(Df, x=["Midterm-1","Terminal-1","Midterm-2","Terminal-2"],
        y=Sub[self.Sec], title="Total Marks of Each Subject of " + self.Name.title(),
        text_auto=True, barmode='group')


        Bar.show()



class Teacher:
    def __init__(self, Username, Password):
        global Login_T
        global Login_CT
        global User

        User=Username
        self.Teachers=Teachers

        if Username in self.Teachers and Password==self.Teachers[Username][1]:
            Login_T=True

            if self.Teachers[Username][-1]=="CT":
                Login_CT=True
            else:
                Login_CT=False
```



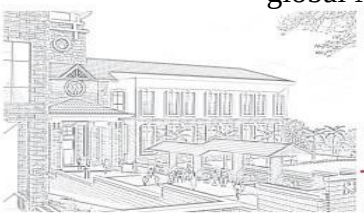
```
        else:
            Login_T=False
            Login_CT=False


class ST:
    def __init__(self, Section):
        global MARK1


        self.Teachers=Teachers
        self.Sub=Sub


        self.Sec=Section
        self.Class=self.Teachers[User][2]
        self.Subject=self.Teachers[User][3]
        self.Index=self.Sub[Section].index(self.Subject)


        if Marks_A["Section"]==self.Sec:
            self.Marks=Marks_A


        elif Marks_B["Section"]==self.Sec:
            self.Marks=Marks_B


        elif Marks_C["Section"]==self.Sec:
            self.Marks=Marks_C
```



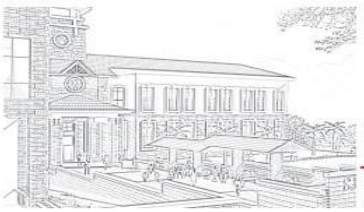
```
        else:
            self.Marks=Marks_D

        MARK1=self.Marks

    def Edit(self, Name, Exam, New_Marks):
        global Mistake1

        if 0<=New_Marks<=100:
            self.Marks[Name][Exam]=list(self.Marks[Name][Exam])
            self.Marks[Name][Exam][self.Index]=New_Marks
            self.Marks[Name][Exam]=tuple(self.Marks[Name][Exam])
            Mistake1=False

        else:
            Mistake1=True



    def View(self, Name, Exam):
        global Marks
        Marks=self.Marks[Name][Exam][self.Index]

class CT:
    def __init__(self):
        global MARK2
```



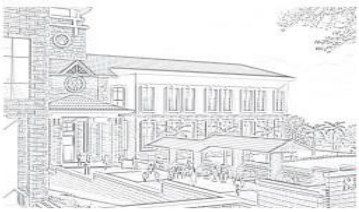
```
        global Subjects

        CT_Name=Teachers[User][0]

        for x in Class_Teachers:
            if Class_Teachers[x]==CT_Name:
                self.Sec=x

        if Marks_A["Section"]==self.Sec:
            self.Marks=Marks_A

        elif Marks_B["Section"]==self.Sec:
            self.Marks=Marks_B

        elif Marks_C["Section"]==self.Sec:
            self.Marks=Marks_C

        else:
            self.Marks=Marks_D

        MARK2=self.Marks
        Subjects=Sub[self.Sec]

    def Edit(self, Name, Exam, Subject, New_Marks):
        global Mistake2
```



```
        self.Index=Sub[self.Sec].index(Subject)
        if 0<=New_Marks<=100:
            self.Marks[Name][Exam]=list(self.Marks[Name][Exam])
            self.Marks[Name][Exam][self.Index]=New_Marks
            self.Marks[Name][Exam]=tuple(self.Marks[Name][Exam])
            Mistake2=False

        else:
            Mistake2=True

    def View(self, Name, Exam, Subject):
        global Marks
        self.Index=Sub[self.Sec].index(Subject)

        Marks=self.Marks[Name][Exam][self.Index]

class Update():
    def __init__(self):
        File=open("Data.bin", "wb")
        pickle.dump(Marks_A, File)
        pickle.dump(Marks_B, File)
        pickle.dump(Marks_C, File)
        pickle.dump(Marks_D, File)
```

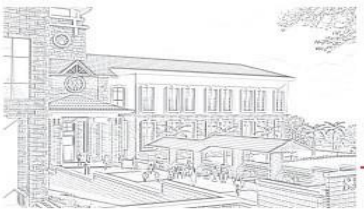


```
pickle.dump(Details_A, File)
pickle.dump(Details_B, File)
pickle.dump(Details_C, File)
pickle.dump(Details_D, File)
pickle.dump(Attendance_A, File)
pickle.dump(Attendance_B, File)
pickle.dump(Attendance_C, File)
pickle.dump(Attendance_D, File)
pickle.dump(Teachers, File)
pickle.dump(Class_Teachers, File)
pickle.dump(Sub, File)
File.close()
```



Frontend.py:

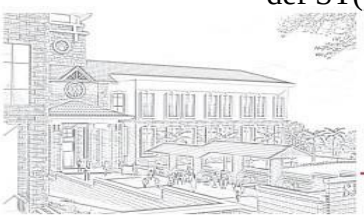
```
import Framework

import tkinter

from tkinter import *

from tkinter.ttk import *

from tkinter import messagebox

from PIL import Image,ImageTk


def Closing():

    if messagebox.askokcancel("Quit", "You are about to Quit Student Mark Analysis
   Application"):

        Obj=Framework.Update()

        SMA.destroy()


SMA=Tk()

SMA.title("Student Mark Analysis")

SMA.configure(background="white")

SMA.state("zoomed")

SMA.protocol("WM_DELETE_WINDOW", Closing)

SMA.resizable(False, False)


global Stu_Obj

global CT_Obj

global ST_Obj
```



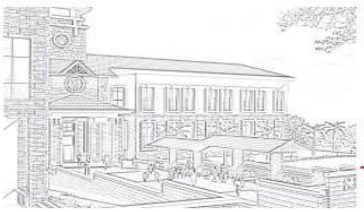
```
global _Name
global User

Home=tkinter.Frame(SMA)
Home.pack(fill=BOTH, expand=True)

def Next():
    Home.forget()
    Obj=Page1()

Home.configure(background="white")
Home.columnconfigure(0, weight=1)
Home.columnconfigure(1, weight=1)
Home.columnconfigure(2, weight=1)

School_Logo=PhotoImage(file="School_Logo.png")
Logo=Label(Home, image = School_Logo, relief="solid")

Text1=Label(Home, text="Student Mark Analysis of Class - XII", font=("Verdana", 20),
background="white")
Text2=Label(Home, text="Done By :", font=("Verdana", 20), background="white")

Arvin=Image.open("Arvin.png")
Ezhil=Image.open("Ezhil.png")
Kingston=Image.open("Kingston.png")
```

```
Arvin_Resize=Arvin.resize((150, 180), Image.ANTIALIAS)
Ezhil_Resize=Ezhil.resize((150, 180), Image.ANTIALIAS)
Kingston_Resize=Kingston.resize((150, 180), Image.ANTIALIAS)

Arvin=ImageTk.PhotoImage(Arvin_Resize)
Ezhil=ImageTk.PhotoImage(Ezhil_Resize)
Kingston=ImageTk.PhotoImage(Kingston_Resize)

Pic1=Label(Home, image = Arvin, relief="flat")
Pic2=Label(Home, image = Ezhil, relief="flat")
Pic3=Label(Home, image = Kingston, relief="flat")

Name1=Label(Home, text="Arvin Samuel A.", font=("Verdana", 10), background="white")
Name2=Label(Home, text="Ezhil Adhithya P.", font=("Verdana", 10), background="white")
Name3=Label(Home, text="Kingston Richard J.", font=("Verdana", 10), background="white")

Text=Label(Home, text="Click Here to Login", font=("Verdana", 18), background="white")
style=Style()
style.configure("Home.TButton", font=("Verdana", 15))

Login=Button(Home, text="Login", command=Next, style="Home.TButton")

Logo.grid(column=0, row=0, columnspan=4)
Text1.grid(column=0, row=1, columnspan=4, pady=10)
Text2.grid(column=0, row=2, columnspan=4, pady=10)
```

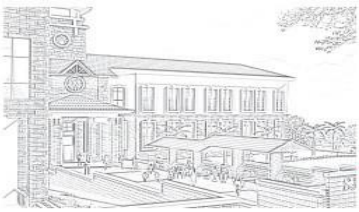
```
Pic1.grid(column=0, row=3, pady=10)

Pic2.grid(column=1, row=3, pady=10)

Pic3.grid(column=2, row=3, pady=10)


Name1.grid(column=0, row=4, pady=20)

Name2.grid(column=1, row=4, pady=20)

Name3.grid(column=2, row=4, pady=20)


Text.grid(column=0, row=5, columnspan=4, pady=10)

Login.grid(column=0, row=6, columnspan=4, pady=10)


class Menu_Bar:

    def __init__(self, Menu):

        SMA.config(menu=Menu)


class Page1:

    def __init__(self):

        Frame1=tkinter.Frame(SMA)

        Frame1.pack(fill=BOTH, expand=True)


        def Student():

            Frame1.forget()

            Obj=Login_Student()


        def ST():
```



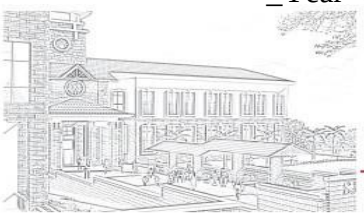
```
    Frame1.forget()
    Obj=Login_ST()

def CT():
    Frame1.forget()
    Obj=Login_CT()

def Previous():
    Frame1.forget()
    Menubar=Menu()
    Top_Menu=Menu_Bar(Menubar)
    Home.pack(fill=BOTH, expand=True)

Frame1.configure(background="white")
Frame1.columnconfigure(0, weight=1)
Frame1.columnconfigure(1, weight=1)

Menubar=Menu(Frame1)
Back=Menubar.add_command(label="Back", command=Previous)

Text=Label(Frame1, text="Are you a", font=("Verdana", 70), background="white")
Student=Button(Frame1, text="Student / Parent", command=Student, style="A.TButton")
Subject_Teacher=Button(Frame1, text="Subject Teacher", command=ST,
style="A.TButton")
Class_Teacher=Button(Frame1, text="Class Teacher", command=CT, style="A.TButton")
```



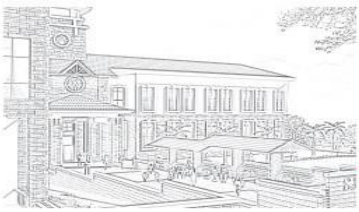
```
        style1=Style()
        style1.configure("A.TButton", font=("Verdana", 40))

        Text.grid(column=0, row=0, columnspan=2, pady=40)
        Student.grid(column=0, row=1, pady=40, ipadx=15, ipady=15)
        Subject_Teacher.grid(column=1, row=1, columnspan=2, pady=40, ipadx=15, ipady=15)
        Class_Teacher.grid(column=0, row=2, columnspan=2, pady=40, ipadx=15, ipady=15)

        Top_Menu=Menu_Bar(Menubar)

class Login_Student:
    def __init__(self):
        Frame2=tkinter.Frame(SMA)
        Frame2.pack(fill=BOTH, expand=True)

        def Login():
            if bool(_Name.get())==False or bool(_Date.get())==False or bool(_Month.get())==False
            or bool(_Year.get())==False or bool(_Phone.get())==False:
                messagebox.showerror("Error", "Some Field or Fields are Left Empty !!")

            else:
                login1=Framework.Student(_Name.get())
                login1.__init__(_Name.get())
```



```
        try:
            login1.Login((int(_Date.get()), int(_Month.get()), int(_Year.get())), _Phone.get())

            if Framework.Login_S==False:
                messagebox.showerror("Error", "Sorry the Entered Information is Wrong !!")

            else:
                Frame2.forget()
                Obj=Student1()

        except:
            messagebox.showerror("Error", "Sorry the Entered Information is Wrong !!")

def Previous():
    Frame2.forget()
    Obj=Page1()

Frame2.configure(background="white")
Frame2.columnconfigure(0, weight=1)
Frame2.columnconfigure(1, weight=1)

_Name=StringVar()
_Date=StringVar()
_Month=StringVar()
_Year=StringVar()
```

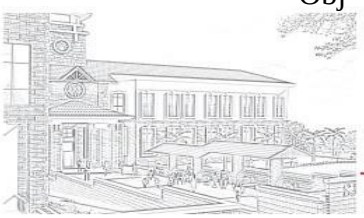


```
_Phone=StringVar()

Menubar=Menu(Frame2)
Back=Menubar.add_command(label="Back", command=Previous)

Text1=Label(Frame2, text="LOGIN", font=("Verdana", 40), background="white")
Text2=Label(Frame2, text="(For Student)", font=("Verdana", 20), background="white")

Name=Label(Frame2, text="Enter your Name : ", font=("Verdana", 20, "bold"),
background="white")
Name_Ext=Label(Frame2, text="(in lower case without initial)", font=("Verdana", 20,
"bold"), background="white")
Name_Input=Entry(Frame2, textvariable=_Name, font=("Verdana", 20, "normal"))

Date=Label(Frame2, text="Enter your \"Date\" of Birth (from 1 to 31) : ",
font=("Verdana", 20, "bold"), background="white")
Date_Input=Entry(Frame2, textvariable=_Date, font=("Verdana", 20, "normal"))

Month=Label(Frame2, text="Enter your \"Month\" of Birth (from 1 to 12) : ",
font=("Verdana", 20, "bold"), background="white")
Month_Input=Entry(Frame2, textvariable=_Month, font=("Verdana", 20, "normal"))

Year=Label(Frame2, text="Enter your \"Year\" of Birth (ex. 2005) : ", font=("Verdana",
20, "bold"), background="white")
Year_Input=Entry(Frame2, textvariable=_Year, font=("Verdana", 20, "normal"))
```

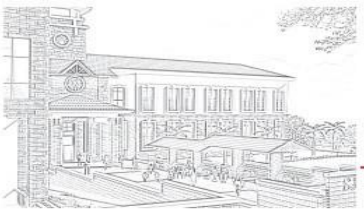


```
Phone=Label(Frame2, text="Enter your Phone Number :", font=("Verdana", 20, "bold"), background="white")

Phone_Input=Entry(Frame2, textvariable=_Phone, font=("Verdana", 20, "normal"))


style2=Style()

style2.configure("B.TButton", font=("Verdana", 15))


Clear=Button(Frame2, text="CLEAR", command=lambda:[Name_Input.delete(0, END), Date_Input.delete(0, END), Month_Input.delete(0, END), Year_Input.delete(0, END), Phone_Input.delete(0, END) ], style="B.TButton")

Submit=Button(Frame2, text="SUBMIT", command=Login, style="B.TButton")


Text1.grid(column=0, row=0, columnspan=2, pady=15)

Text2.grid(column=0, row=1, columnspan=2)

Name.grid(column=0, row=2, sticky=W, padx=40)

Name_Ext.grid(column=0, row=3, pady=15, sticky=W, padx=40)

Name_Input.grid(column=1, row=2, rowspan=2, pady=15)

Date.grid(column=0, row=4, pady=15, sticky=W, padx=40)

Date_Input.grid(column=1, row=4, pady=15)

Month.grid(column=0, row=5, pady=15, sticky=W, padx=40)

Month_Input.grid(column=1, row=5, pady=15)

Year.grid(column=0, row=6, pady=15, sticky=W, padx=40)

Year_Input.grid(column=1, row=6, pady=15)

Phone.grid(column=0, row=7, pady=15, sticky=W, padx=40)

Phone_Input.grid(column=1, row=7, pady=15)
```

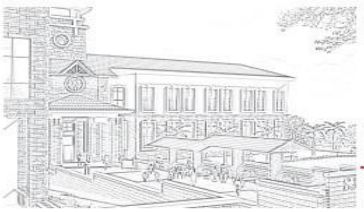
```
        Clear.grid(column=1, row=8, pady=15)
        Submit.grid(column=1, row=9)

        Top_Menu=Menu_Bar(Menubar)

class Student1:
    def __init__(self):
        Frame3=tkinter.Frame(SMA)
        Frame3.pack(fill=BOTH, expand=True)

        Stu_Obj=Framework.Student(Framework._Name)

        def Logout():
            if messagebox.askokcancel("Logout","You Are About to Logout"):
                Frame3.forget()
                Obj=Login_Student()

        def Logout1():
                if messagebox.askokcancel("Logout","You Are About to Logout"):
                    Frame3.forget()
                    Obj=Login_CT()

        def Previous1():
            Frame3.forget()
            Obj=CT_Review()
```



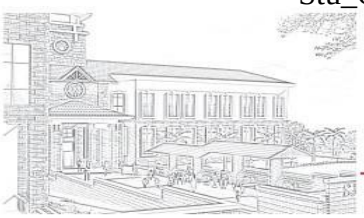
```
def Report():
    Frame3.forget()
    Obj=Exams()

def func1():
    Stu_Obj.Total_Bar()

def func2():
    Stu_Obj.Pie_Marks_Distribution()

def func3():
    Stu_Obj.Overall_Radar()

def func4():
    Stu_Obj.Subject_Bar()

Frame3.configure(background="white")
Frame3.columnconfigure(0, weight=1)
Frame3.columnconfigure(1, weight=1)

Menubar=Menu(Frame3)
Logout=Menubar.add_command(label="Logout", command=Logout)
Text1=Label(Frame3, text="Hello, " + Framework._Name.title(), font=("Verdana", 40),
background="white")
```



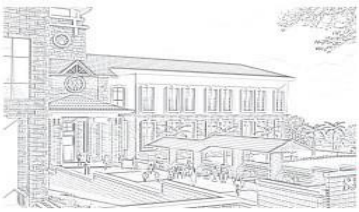
```
Text2=Label(Frame3, text="What do you want to see ?", font=("Verdana", 25),
background="white")


style3=Style()

style3.configure("C.TButton", font=("Verdana", 20))


Total_Bar=Button(Frame3, text="Comparison of Total Marks", command=func1,
style="C.TButton")

Pie=Button(Frame3, text="Distribution of subject marks", command=func2,
style="C.TButton")

Radar=Button(Frame3, text="Overall Academic Performance", command=func3,
style="C.TButton")

Group_Bar=Button(Frame3, text="Overall Performance in each subject",
command=func4, style="C.TButton")

Report_Card=Button(Frame3, text="Report Card", command=Report, style="C.TButton")


Text1.grid(column=0, row=0, columnspan=2, pady=20, padx=40, sticky=W)

Text2.grid(column=0, row=1, columnspan=2, pady=20, padx=100, sticky=W)

Total_Bar.grid(column=0, row=2, pady=20)

Pie.grid(column=1, row=2, pady=20)

Radar.grid(column=0, row=3, pady=20)

Group_Bar.grid(column=1, row=3, pady=20)

Report_Card.grid(column=0, row=4, columnspan=2, pady=20)


Top_Menu=Menu_Bar(Menubar)
```



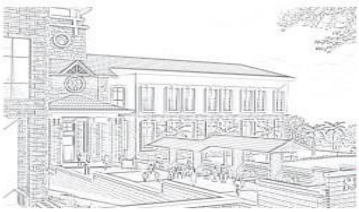
```
class Exams:
    def __init__(self):
        Frame4=tkinter.Frame(SMA)
        Frame4.pack(fill=BOTH, expand=True)

        Stu_Obj=Framework.Student(Framework._Name)

        def Previous():
            Frame4.forget()
            Obj=Student1()

        def func1():
            Stu_Obj.Exam("Midterm-1")
            Stu_Obj.Report_Card()

        def func2():
            Stu_Obj.Exam("Terminal-1")
            Stu_Obj.Report_Card()

        def func3():
            Stu_Obj.Exam("Midterm-2")
            Stu_Obj.Report_Card()

        def func4():
            Stu_Obj.Exam("Terminal-2")
```



```
    Stu_Obj.Report_Card()


Frame4.configure(background="white")

Frame4.columnconfigure(0, weight=1)

Frame4.columnconfigure(1, weight=1)


Menubar=Menu(Frame4)

Back=Menubar.add_command(label="Back", command=Previous)


style4=Style()

style4.configure("D.TButton", font=("Verdana", 30))


Text=Label(Frame4, text="Please select a Exam from Below", font=("Verdana", 40),
background="white")

Midterm1=Button(Frame4, text="Midterm-1", command=func1, style="D.TButton")

Terminal1=Button(Frame4, text="Terminal-1", command=func2, style="D.TButton")

Midterm2=Button(Frame4, text="Midterm-2", command=func3, style="D.TButton")

Terminal2=Button(Frame4, text="Terminal-2", command=func4, style="D.TButton")


Text.grid(column=0, row=0, columnspan=2, pady=40, padx=40, sticky=W)

Midterm1.grid(column=0, row=1, pady=40)

Terminal1.grid(column=1, row=1, pady=40)

Midterm2.grid(column=0, row=2, pady=40)

Terminal2.grid(column=1, row=2, pady=40)
```

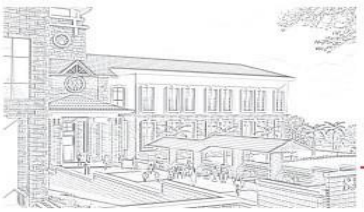


```
        Top_Menu=Menu_Bar(Menubar)


class Login_ST:
    def __init__(self):
        Frame5=tkinter.Frame(SMA)
        Frame5.pack(fill=BOTH, expand=True)


        def Login():
            login2=Framework.Teacher(_Username.get(), _Password.get())


            if Framework.Login_T==False:
                messagebox.showerror("Error", "Sorry the Entered Username or Password is Wrong !!")


            else:
                Frame5.forget()
                Obj=ST1()


        def Previous():
            Frame5.forget()
            Obj=Page1()


        Frame5.configure(background="white")
        Frame5.columnconfigure(0, weight=1)
        Frame5.columnconfigure(1, weight=1)
```

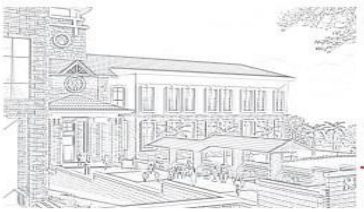


```
_Username=StringVar()

_Password=StringVar()


Menubar=Menu(Frame5)

Back=Menubar.add_command(label="Back", command=Previous)


Text1=Label(Frame5, text="LOGIN", font=("Verdana", 40), background="white")

Text2=Label(Frame5, text="(For Subject Teacher)", font=("Verdana", 20),
background="white")


Username=Label(Frame5, text="Enter the Username : ", font=("Verdana", 20, "bold"),
background="white")

Username_Input=Entry(Frame5, textvariable=_Username, font=("Verdana", 20,
"normal"))


Password=Label(Frame5, text="Enter the Password : ", font=("Verdana", 20, "bold"),
background="white")

Password_Input=Entry(Frame5, textvariable=_Password, font=("Verdana", 20, "normal"),
show="*")


style5=Style()

style5.configure("E.TButton", font=("Verdana", 15))


Clear=Button(Frame5, text="CLEAR", command=lambda:[Username_Input.delete(0,
END), Password_Input.delete(0, END)], style="E.TButton")

Submit=Button(Frame5, text="SUBMIT", command=Login, style="E.TButton")
```

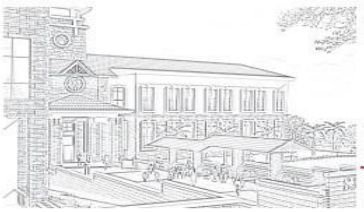
```
        Text1.grid(column=0, row=0, columnspan=2, pady=20)
        Text2.grid(column=0, row=1, columnspan=2)
        Username.grid(column=0, row=2, sticky=W, padx=40)
        Username_Input.grid(column=1, row=2, pady=20)
        Password.grid(column=0, row=3, sticky=W, padx=40)
        Password_Input.grid(column=1, row=3, pady=20)
        Clear.grid(column=1, row=4, pady=20)
        Submit.grid(column=1, row=5)

        Top_Menu=Menu_Bar(Menubar)

class ST1:
    def __init__(self):
        Frame6=tkinter.Frame(SMA)
        Frame6.pack(fill=BOTH, expand=True)

        def Logout():
            if messagebox.askokcancel("Logout","You Are About to Logout"):
                Frame6.forget()
                Obj=Login_ST()

        def func1():
            Frame6.forget()
            Obj=ST_Edit()
```



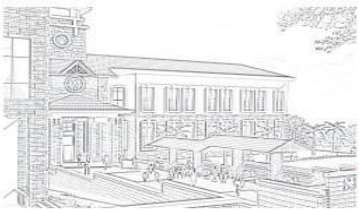
```
def func2():
    Frame6.forget()
    Obj=ST_View()

Frame6.configure(background="white")
Frame6.columnconfigure(0, weight=1)
Frame6.columnconfigure(1, weight=1)

Menubar=Menu(Frame6)
Logout=Menubar.add_command(label="Logout", command=Logout)

style6=Style()
style6.configure("F.TButton", font=("Verdana", 25))

Text1=Label(Frame6, text="Hello, " + Framework.Teachers[Framework.User][0].title(),
font=("Verdana", 40), background="white")
Text2=Label(Frame6, text="What do you want to do ?", font=("Verdana", 25),
background="white")
Edit=Button(Frame6, text="Edit", command=func1, style="F.TButton")
View=Button(Frame6, text="View", command=func2, style="F.TButton")

Text1.grid(column=0, row=0, columnspan=2, pady=20, padx=40, sticky=W)
Text2.grid(column=0, row=1, columnspan=2, pady=20, padx=100, sticky=W)
Edit.grid(column=0, row=2, pady=40)
View.grid(column=1, row=2, pady=40)
```



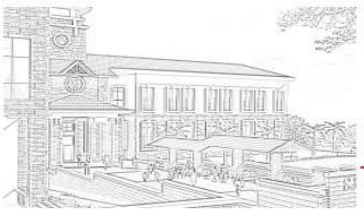
```
        Top_Menu=Menu_Bar(Menubar)


class ST_Edit:
    def __init__(self):
        Frame7=tkinter.Frame(SMA)
        Frame7.pack(fill=BOTH, expand=True)


        def Logout():
            if messagebox.askokcancel("Logout","You Are About to Logout"):
                Frame7.forget()
                Obj=Login_ST()


        def Previous():
            Frame7.forget()
            Obj=ST1()


        def func1():
            try:
                global ST_Obj
                ST_Obj=Framework.ST(_Section.get())
            except:
                pass


        def func2():
            if bool(_Section.get())==True:
```



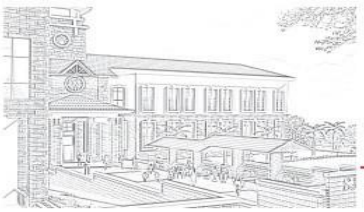
```
        try:
            Name_Input["values"]=tuple(Framework.MARK1.keys())


        except:
            pass


def func3():
    if bool(_Section.get())==False or bool(_Exam.get())==False or
    bool(_Name.get())==False or bool(_New_Marks.get())==False:
            messagebox.showerror("Error", "Some Field or Fields are Left Empty !!")
    else:
        ST_Obj.Edit(_Name.get(), _Exam.get(), _New_Marks.get())


        if Framework.Mistake1==True:
            messagebox.showerror("Error", "The Marks should be between 0 and 100 !!")


        else:
            messagebox.showinfo("Success", "The Marks of the Student is changed
            Successfully !!")


Frame7.configure(background="white")
Frame7.columnconfigure(0, weight=1)
Frame7.columnconfigure(1, weight=1)
Frame7.columnconfigure(2, weight=1)
Frame7.columnconfigure(3, weight=1)
```



```
Frame7.rowconfigure(0, weight=1)


Menubar=Menu(Frame7)

Logout=Menubar.add_command(label="Logout", command=Logout)

Back=Menubar.add_command(label="Back", command=Previous)


style7=Style()

style7.configure("G.TButton", font=("Verdana", 20))


_Section=StringVar()

_Exam=StringVar()

_Name=StringVar()

_New_Marks=IntVar()


Section=Label(Frame7, text="Section", font=("Verdana", 20), background="white")

Section_Input=Combobox(Frame7, textvariable=_Section, font=("Verdana", 20), state =
"readonly")

Section_Input["values"]=Framework.Teachers[Framework.User][2]


Exam=Label(Frame7, text="Exam", font=("Verdana", 20), background="white")

Exam_Input=Combobox(Frame7, textvariable=_Exam, font=("Verdana", 20),
postcommand=func1, state = "readonly")

Exam_Input["values"]=("Midterm-1", "Terminal-1", "Midterm-2", "Terminal-2")

Name=Label(Frame7, text="Name of the Student", font=("Verdana", 20),
background="white")
```

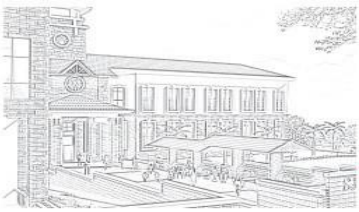


```
Name_Input=Combobox(Frame7, textvariable=_Name, font=("Verdana", 20),
postcommand=lambda: [func1(), func2()], state = "readonly")


New_Marks=Label(Frame7, text="New Marks", font=("Verdana", 20),
background="white")

New_Marks_Input=Entry(Frame7, textvariable=_New_Marks, font=("Verdana", 20,
"normal"))


Submit=Button(Frame7, text="Submit", command=func3, style="G.TButton")

Clear=Button(Frame7, text="Clear", command=lambda:[Section_Input.set(""),
Exam_Input.set(""), Name_Input.set(""), New_Marks_Input.delete(0, END)] ,
style="G.TButton")


Section.grid(column=0, row=1, pady=20, padx=10)

Exam.grid(column=1, row=1, pady=20, padx=10)

Name.grid(column=2, row=1, pady=20, padx=10)

New_Marks.grid(column=3, row=1, pady=20, padx=10)


Section_Input.grid(column=0, row=2, pady=20, padx=10)

Exam_Input.grid(column=1, row=2, pady=20, padx=10)

Name_Input.grid(column=2, row=2, pady=20, padx=10)

New_Marks_Input.grid(column=3, row=2, pady=20, padx=10)


Submit.grid(column=0, row=3, columnspan=2, pady=100)

Clear.grid(column=2, row=3, columnspan=2, pady=100)
```



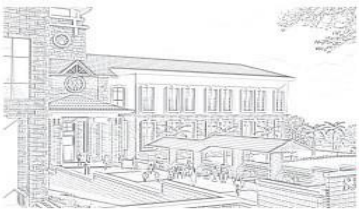
```
        Top_Menu=Menu_Bar(Menubar)


class ST_View:
    def __init__(self):
        Frame8=tkinter.Frame(SMA)
        Frame8.pack(fill=BOTH, expand=True)


        def Logout():
            if messagebox.askokcancel("Logout","You Are About to Logout"):
                Frame8.forget()
                Obj=Login_ST()


        def Previous():
            Frame8.forget()
            Obj=ST1()


        def func1():
            try:
                global ST_Obj
                ST_Obj=Framework.ST(_Section.get())
            except:
                pass


        def func2():
            try:
```

```
        if bool(_Section.get())==True:
            Name_Input["values"]=tuple(Framework.MARK1.keys())
    except:
        pass



def func3():
    if bool(_Section.get())==False or bool(_Exam.get())==False or
    bool(_Name.get())==False:
            messagebox.showerror("Error", "Some Field or Fields are Left Empty !!")
    else:
        ST_Obj.View(_Name.get(), _Exam.get())
        messagebox.showinfo("Marks",  _Name.get().title() + " have scored " +
        str(Framework.Marks) + " in " + _Exam.get())


Frame8.configure(background="white")
Frame8.columnconfigure(0, weight=1)
Frame8.columnconfigure(1, weight=1)
Frame8.columnconfigure(2, weight=1)
Frame8.rowconfigure(0, weight=1)


Menubar=Menu(Frame8)
Logout=Menubar.add_command(label="Logout", command=Logout)
Back=Menubar.add_command(label="Back", command=Previous)
```

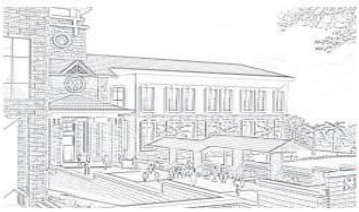


```
style8=Style()

style8.configure("H.TButton", font=("Verdana", 20))


_Section=StringVar()

_Exam=StringVar()

_Name=StringVar()

_New_Marks=IntVar()


Section=Label(Frame8, text="Section", font=("Verdana", 20), background="white")

Section_Input=Combobox(Frame8, textvariable=_Section, font=("Verdana", 20), state =
"readonly")

Section_Input["values"]=Framework.Teachers[Framework.User][2]


Exam=Label(Frame8, text="Exam", font=("Verdana", 20), background="white")

Exam_Input=Combobox(Frame8, textvariable=_Exam, font=("Verdana", 20),
postcommand=func1, state = "readonly")

Exam_Input["values"]=("Midterm-1", "Terminal-1", "Midterm-2", "Terminal-2")


Name=Label(Frame8, text="Name of the Student", font=("Verdana", 20),
background="white")

Name_Input=Combobox(Frame8, textvariable=_Name, font=("Verdana", 20),
postcommand=lambda: [func1(), func2()], state = "readonly")


Submit=Button(Frame8, text="Submit", command=func3, style="H.TButton")

Clear=Button(Frame8, text="Clear", command=lambda:[Section_Input.set(""),
Exam_Input.set(""), Name_Input.set("")] , style="H.TButton")
```

```
        Section.grid(column=0, row=1, pady=20, padx=10)
        Exam.grid(column=1, row=1, pady=20, padx=10)
        Name.grid(column=2, row=1, pady=20, padx=10)

        Section_Input.grid(column=0, row=2, pady=20, padx=10)
        Exam_Input.grid(column=1, row=2, pady=20, padx=10)
        Name_Input.grid(column=2, row=2, pady=20, padx=10)

        Submit.grid(column=0, row=3, pady=100)
        Clear.grid(column=2, row=3, pady=100)

        Top_Menu=Menu_Bar(Menubar)

class Login_CT:
    def __init__(self):
        Frame9=tkinter.Frame(SMA)
        Frame9.pack(fill=BOTH, expand=True)

        def Login():
            login3=Framework.Teacher(_Username.get(), _Password.get())

            if Framework.Login_T==False and Framework.Login_CT==False:
                messagebox.showerror("Error", "Sorry the Entered Username or Password is Wrong !!")
```

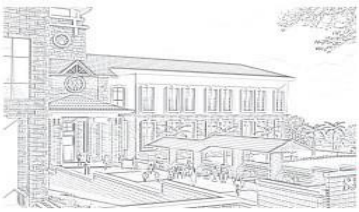


```
    elif Framework.Login_T==True and Framework.Login_CT==False:
        messagebox.showerror("Error", "You're Not a Class Teacher !!")

    else:
        Frame9.forget()
        Obj=CT1()

def Previous():
    Frame9.forget()
    Obj=Page1()

Frame9.configure(background="white")
Frame9.columnconfigure(0, weight=1)
Frame9.columnconfigure(1, weight=1)

_Username=StringVar()
_Password=StringVar()

Menubar=Menu(Frame9)
Back=Menubar.add_command(label="Back", command=Previous)

Text1=Label(Frame9, text="LOGIN", font=("Verdana", 40), background="white")
Text2=Label(Frame9, text="(For Class Teacher)", font=("Verdana", 20),
background="white")
```

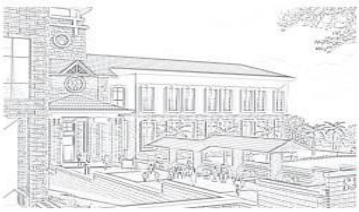


```
Username=Label(Frame9, text="Enter the Username : ", font=("Verdana", 20, "bold"), background="white")

Username_Input=Entry(Frame9, textvariable=_Username, font=("Verdana", 20, "normal"))


Password=Label(Frame9, text="Enter the Password : ", font=("Verdana", 20, "bold"), background="white")

Password_Input=Entry(Frame9, textvariable=_Password, font=("Verdana", 20, "normal"), show="*")


style9=Style()

style9.configure("I.TButton", font=("Verdana", 15))


Clear=Button(Frame9, text="CLEAR", command=lambda:[Username_Input.delete(0, END), Password_Input.delete(0, END)], style="I.TButton")

Submit=Button(Frame9, text="SUBMIT", command=Login, style="I.TButton")


Text1.grid(column=0, row=0, columnspan=2, pady=20)

Text2.grid(column=0, row=1, columnspan=2)

Username.grid(column=0, row=2, sticky=W, padx=40)

Username_Input.grid(column=1, row=2, pady=20)

Password.grid(column=0, row=3, sticky=W, padx=40)

Password_Input.grid(column=1, row=3, pady=20)

Clear.grid(column=1, row=4, pady=20)

Submit.grid(column=1, row=5)
```

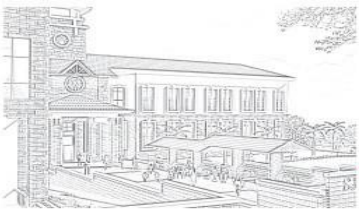
```
        Top_Menu=Menu_Bar(Menubar)


class CT1:
    def __init__(self):
        Frame10=tkinter.Frame(SMA)
        Frame10.pack(fill=BOTH, expand=True)


        def Logout():
            if messagebox.askokcancel("Logout","You Are About to Logout"):
                Frame10.forget()
                Obj=Login_CT()


        def func1():
            Frame10.forget()
            Obj=CT_Edit()


        def func2():
            Frame10.forget()
            Obj=CT_View()


        def func3():
            Frame10.forget()
            Obj=CT_Review()


        Frame10.configure(background="white")
```



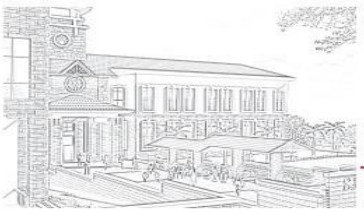
```
Frame10.columnconfigure(0, weight=1)

Frame10.columnconfigure(1, weight=1)


Menubar=Menu(Frame10)

Logout=Menubar.add_command(label="Logout", command=Logout)


style10=Style()

style10.configure("J.TButton", font=("Verdana", 25))


Text1=Label(Frame10, text="Hello, " + Framework.Teachers[Framework.User][0].title(),
font=("Verdana", 40), background="white")

Text2=Label(Frame10, text="What do you want to do ?", font=("Verdana", 25),
background="white")

Edit=Button(Frame10, text="Edit", command=func1, style="J.TButton")

View=Button(Frame10, text="View", command=func2, style="J.TButton")

Review=Button(Frame10, text="Review a Student's Performance", command=func3,
style="J.TButton")


Text1.grid(column=0, row=0, columnspan=2, pady=20, padx=40, sticky=W)

Text2.grid(column=0, row=1, columnspan=2, pady=20, padx=100, sticky=W)

Edit.grid(column=0, row=2, pady=40)

View.grid(column=1, row=2, pady=40)

Review.grid(column=0, row=3, columnspan=2, pady=40)


Top_Menu=Menu_Bar(Menubar)
```



```
class CT_Edit:
    def __init__(self):
        Frame11=tkinter.Frame(SMA)
        Frame11.pack(fill=BOTH, expand=True)

        global CT_Obj
        CT_Obj=Framework.CT()

        def Logout():
            if messagebox.askokcancel("Logout","You Are About to Logout"):
                Frame11.forget()
                Obj=Login_CT()

        def Previous():
            Frame11.forget()
            Obj=CT1()

        def func1():
            if bool(_Subject.get())==False or bool(_Exam.get())==False or
bool(_Name.get())==False or bool(_New_Marks.get())==False:
                messagebox.showerror("Error", "Some Field or Fields are Left Empty !!")

            else:
                CT_Obj.Edit(_Name.get(), _Exam.get(), _Subject.get(), _New_Marks.get())
```

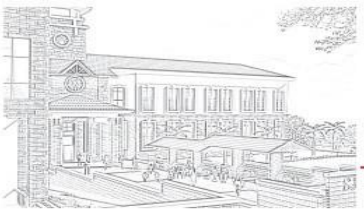


```
        if Framework.Mistake2==True:
            messagebox.showerror("Error", "The Marks should be between 0 and 100 !!")

        else:
            messagebox.showinfo("Success", "The Marks of the Student is changed
            Successfully !!")

Frame11.configure(background="white")
Frame11.columnconfigure(0, weight=1)
Frame11.columnconfigure(1, weight=1)
Frame11.columnconfigure(2, weight=1)
Frame11.columnconfigure(3, weight=1)
Frame11.rowconfigure(0, weight=1)

Menubar=Menu(Frame11)
Logout=Menubar.add_command(label="Logout", command=Logout)
Back=Menubar.add_command(label="Back", command=Previous)

style11=Style()
style11.configure("K.TButton", font=("Verdana", 20))

_Name=StringVar()
_Exam=StringVar()
_Subject=StringVar()
_New_Marks=IntVar()
```

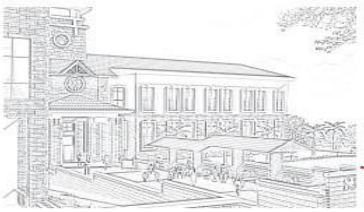


```
Name=Label(Frame11, text="Name of the Student", font=("Verdana", 20),
background="white")

Name_Input=Combobox(Frame11, textvariable=_Name, font=("Verdana", 20), state =
"readonly")

Name_Input["values"]=tuple(Framework.MARK2.keys())


Exam=Label(Frame11, text="Exam", font=("Verdana", 20), background="white")

Exam_Input=Combobox(Frame11, textvariable=_Exam, font=("Verdana", 20), state =
"readonly")

Exam_Input["values"]=("Midterm-1", "Terminal-1", "Midterm-2", "Terminal-2")


Subject=Label(Frame11, text="Subject", font=("Verdana", 20), background="white")

Subject_Input=Combobox(Frame11, textvariable=_Subject, font=("Verdana", 20), state =
"readonly")

Subject_Input["values"]=Framework.Subjects


New_Marks=Label(Frame11, text="New Marks", font=("Verdana", 20),
background="white")

New_Marks_Input=Entry(Frame11, textvariable=_New_Marks, font=("Verdana", 20,
"normal"))


Submit=Button(Frame11, text="Submit", command=func1, style="K.TButton")

Clear=Button(Frame11, text="Clear", command=lambda:[ Name_Input.set(""),
Subject_Input.set(""), Exam_Input.set(""), New_Marks_Input.delete(0, END)] ,
style="K.TButton")


Name.grid(column=0, row=1, pady=20, padx=10)
```

```
        Exam.grid(column=1, row=1, pady=20, padx=10)
        Subject.grid(column=2, row=1, pady=20, padx=10)
        New_Marks.grid(column=3, row=1, pady=20, padx=10)

        Name_Input.grid(column=0, row=2, pady=20, padx=10)
        Exam_Input.grid(column=1, row=2, pady=20, padx=10)
        Subject_Input.grid(column=2, row=2, pady=20, padx=10)
        New_Marks_Input.grid(column=3, row=2, pady=20, padx=10)

        Submit.grid(column=0, row=3, columnspan=2, pady=100)
        Clear.grid(column=2, row=3, columnspan=2, pady=100)

        Top_Menu=Menu_Bar(Menubar)

class CT_View:
    def __init__(self):
        Frame12=tkinter.Frame(SMA)
        Frame12.pack(fill=BOTH, expand=True)

        global CT_Obj
        CT_Obj=Framework.CT()

        def Logout():
            if messagebox.askokcancel("Logout","You Are About to Logout"):
                Frame12.forget()
```

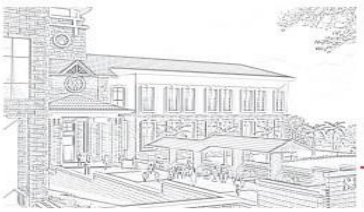


```
        Obj=Login_CT()


def Previous():
    Frame12.forget()
    Obj=CT1()


def func1():
    if bool(_Subject.get())==False or bool(_Exam.get())==False or
    bool(_Name.get())==False:
            messagebox.showerror("Error", "Some Field or Fields are Left Empty !!")


    else:
        CT_Obj.View(_Name.get(), _Exam.get(), _Subject.get())
        messagebox.showinfo("Marks",  _Name.get().title() + " have scored " +
        str(Framework.Marks) + " in " + _Subject.get() + " in " + _Exam.get())


Frame12.configure(background="white")
Frame12.columnconfigure(0, weight=1)
Frame12.columnconfigure(1, weight=1)
Frame12.columnconfigure(2, weight=1)
Frame12.rowconfigure(0, weight=1)


Menubar=Menu(Frame12)
Logout=Menubar.add_command(label="Logout", command=Logout)
Back=Menubar.add_command(label="Back", command=Previous)
```

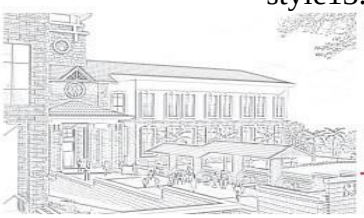


```
style12=Style()

style12.configure("L.TButton", font=("Verdana", 20))


_Name=StringVar()

_Exam=StringVar()

_Subject=StringVar()


Name=Label(Frame12, text="Name of the Student", font=("Verdana", 20),
background="white")

Name_Input=Combobox(Frame12, textvariable=_Name, font=("Verdana", 20), state =
"readonly")

Name_Input["values"]=tuple(Framework.MARK2.keys())


Exam=Label(Frame12, text="Exam", font=("Verdana", 20), background="white")

Exam_Input=Combobox(Frame12, textvariable=_Exam, font=("Verdana", 20), state =
"readonly")

Exam_Input["values"]=("Midterm-1", "Terminal-1", "Midterm-2", "Terminal-2")


Subject=Label(Frame12, text="Subject", font=("Verdana", 20), background="white")

Subject_Input=Combobox(Frame12, textvariable=_Subject, font=("Verdana", 20), state =
"readonly")

Subject_Input["values"]=Framework.Subjects


Submit=Button(Frame12, text="Submit", command=func1, style="L.TButton")

Clear=Button(Frame12, text="Clear", command=lambda:[ Name_Input.set(""),
Subject_Input.set(""), Exam_Input.set("")] , style="L.TButton")
```

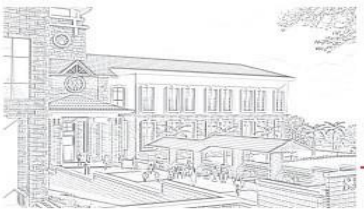
```
        Name.grid(column=0, row=1, pady=20, padx=10)
        Exam.grid(column=1, row=1, pady=20, padx=10)
        Subject.grid(column=2, row=1, pady=20, padx=10)

        Name_Input.grid(column=0, row=2, pady=20, padx=10)
        Exam_Input.grid(column=1, row=2, pady=20, padx=10)
        Subject_Input.grid(column=2, row=2, pady=20, padx=10)

        Submit.grid(column=0, row=3, pady=100)
        Clear.grid(column=2, row=3, pady=100)

        Top_Menu=Menu_Bar(Menubar)

class CT_Review:
    def __init__(self):
        Frame13=tkinter.Frame(SMA)
        Frame13.pack(fill=BOTH, expand=True)

        global CT_Obj
        CT_Obj=Framework.CT()


        def Logout():
            if messagebox.askokcancel("Logout","You Are About to Logout"):
                Frame13.forget()
```



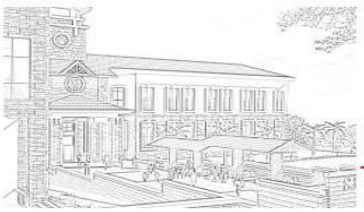
```
        Obj=Login_CT()


def Previous():
    Frame13.forget()
    Obj=CT1()


def func1():
    if bool(_Name.get())==False:
            messagebox.showerror("Error", "The Name is Left Empty !!")


    else:
        Framework._Name=_Name.get()
        Frame13.forget()
        Obj=CT_Student()


Frame13.configure(background="white")
Frame13.columnconfigure(0, weight=1)
Frame13.columnconfigure(1, weight=1)


Menubar=Menu(Frame13)
Logout=Menubar.add_command(label="Logout", command=Logout)
Back=Menubar.add_command(label="Back", command=Previous)


style13=Style()
style13.configure("M.TButton", font=("Verdana", 20))
```



```
        _Name=StringVar()

        Text=Label(Frame13, text="Please Select the Name of the Student from the Following : ",
        font=("Verdana", 40), background="white")

        Name_Input=Combobox(Frame13, textvariable=_Name, font=("Verdana", 20), state =
        "readonly")
        Name_Input["values"]=tuple(Framework.MARK2.keys())

        Submit=Button(Frame13, text="Submit", command=func1, style="M.TButton")
        Clear=Button(Frame13, text="Clear", command=lambda:[ Name_Input.set("")] ,
        style="M.TButton")

        Text.grid(column=0, row=0, columnspan=2, padx=40, pady=10, sticky=W)
        Name_Input.grid(column=0, row=1, columnspan=2, pady=100)

        Submit.grid(column=0, row=2, pady=20)
        Clear.grid(column=1, row=2, pady=20)

        Top_Menu=Menu_Bar(Menubar)

class CT_Student:
    def __init__(self):
        Frame14=tkinter.Frame(SMA)
```

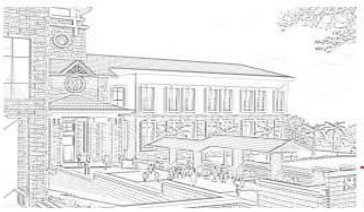


```
Frame14.pack(fill=BOTH, expand=True)

Stu_Obj=Framework.Student(Framework._Name)


def Logout():

    if messagebox.askokcancel("Logout","You Are About to Logout"):

        Frame14.forget()

        Obj=Login_CT()


def Previous():

    Frame14.forget()

    Obj=CT_Review()


def Report():

    Frame14.forget()

    Obj=CT_Exams()


def func1():

    Stu_Obj.Total_Bar()


def func2():

    Stu_Obj.Pie_Marks_Distribution()


def func3():

    Stu_Obj.Overall_Radar()
```



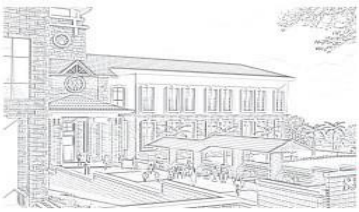
```
def func4():
    Stu_Obj.Subject_Bar()

Frame14.configure(background="white")
Frame14.columnconfigure(0, weight=1)
Frame14.columnconfigure(1, weight=1)

Menubar=Menu(Frame14)
Logout=Menubar.add_command(label="Logout", command=Logout)
Back=Menubar.add_command(label="Back", command=Previous)
Top_Menu=Menu_Bar(Menubar)

Text1=Label(Frame14, text="Hello, " + Framework._Name.title(), font=("Verdana", 40),
background="white")
Text2=Label(Frame14, text="What do you want to see ?", font=("Verdana", 25),
background="white")

style14=Style()
style14.configure("N.TButton", font=("Verdana", 20))

Total_Bar=Button(Frame14, text="Comparison of Total Marks", command=func1,
style="N.TButton")
Pie=Button(Frame14, text="Distribution of subject marks", command=func2,
style="N.TButton")
Radar=Button(Frame14, text="Overall Academic Performance", command=func3,
style="N.TButton")
```

```
        Group_Bar=Button(Frame14, text="Overall Performance in each subject",
        command=func4, style="N.TButton")

        Report_Card=Button(Frame14, text="Report Card", command=Report,
        style="N.TButton")


        Text1.grid(column=0, row=0, columnspan=2, pady=20, padx=40, sticky=W)

        Text2.grid(column=0, row=1, columnspan=2, pady=20, padx=100, sticky=W)

        Total_Bar.grid(column=0, row=2, pady=20)

        Pie.grid(column=1, row=2, pady=20)

        Radar.grid(column=0, row=3, pady=20)

        Group_Bar.grid(column=1, row=3, pady=20)

        Report_Card.grid(column=0, row=4, columnspan=2, pady=20)


        Top_Menu=Menu_Bar(Menubar)


class CT_Exams:

    def __init__(self):

        Frame15=tkinter.Frame(SMA)

        Frame15.pack(fill=BOTH, expand=True)


        Stu_Obj=Framework.Student(Framework._Name)


        def Previous():

            Frame15.forget()

            Obj=CT_Student()
```



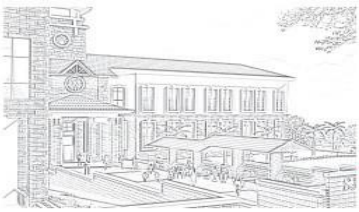
```
def func1():
    Stu_Obj.Exam("Midterm-1")
    Stu_Obj.Report_Card()
    Frame15.forget()
    Obj=CT_Student()

def func2():
    Stu_Obj.Exam("Terminal-1")
    Stu_Obj.Report_Card()
    Frame15.forget()
    Obj=CT_Student()

def func3():
    Stu_Obj.Exam("Midterm-2")
    Stu_Obj.Report_Card()
    Frame15.forget()
    Obj=CT_Student()

def func4():
    Stu_Obj.Exam("Terminal-2")
    Stu_Obj.Report_Card()
    Frame15.forget()
    Obj=CT_Student()

Frame15.configure(background="white")
```

```
Frame15.columnconfigure(0, weight=1)
Frame15.columnconfigure(1, weight=1)

Menubar=Menu(Frame15)
Back=Menubar.add_command(label="Back", command=Previous)

style15=Style()
style15.configure("O.TButton", font=("Verdana", 30))

Text=Label(Frame15, text="Please select a Exam from Below", font=("Verdana", 40),
background="white")
Midterm1=Button(Frame15, text="Midterm-1", command=func1, style="O.TButton")
Terminal1=Button(Frame15, text="Terminal-1", command=func2, style="O.TButton")
Midterm2=Button(Frame15, text="Midterm-2", command=func3, style="O.TButton")
Terminal2=Button(Frame15, text="Terminal-2", command=func4, style="O.TButton")

Text.grid(column=0, row=0, columnspan=2, pady=40, padx=40, sticky=W)
Midterm1.grid(column=0, row=1, pady=40)
Terminal1.grid(column=1, row=1, pady=40)
Midterm2.grid(column=0, row=2, pady=40)
Terminal2.grid(column=1, row=2, pady=40)

Top_Menu=Menu_Bar(Menubar)

SMA.mainloop()
```

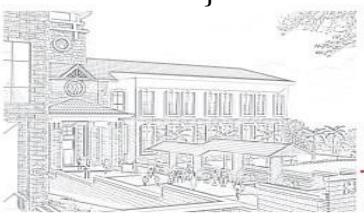


Report_Card.html:

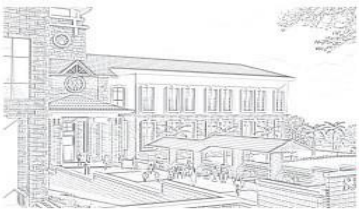
```
<!DOCTYPE html>
<html>

<head>
<style>

table, th, td {
  border:1px solid #000000;
  border-collapse:collapse !important;
  text-align:center;
  padding:3px
}

th {
  vertical-align:middle;
  text-transform:uppercase;
}

body {
  font-family:"calibri";
}
```



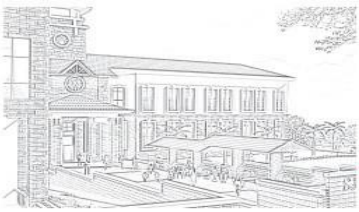
```
.shade1{
  background-color:#a6a6a6;
  color:#222222;
}

.shade2 {
  text-transform:capitalize;
  background-color:#555555;
  color:#ffffff;
}

.shade3 {
  background-color:#ff0080;
  text-transform:capitalize;
  color:#ffffff;
}

.shade4 {
  background-color:#ff99cc;
  text-transform:capitalize;
}

.shade5 {
  background-color:#ffe6f3;
}
```



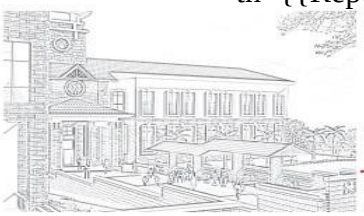
```
.width1 {
  width:230px;
}

.width2 {
  width:450px;
}

.width3 {
  width:100px;
}

.width4 {
  width:130px;
}

.width5 {
  width:50px;
}

.width6 {
  width:30px;
}
```



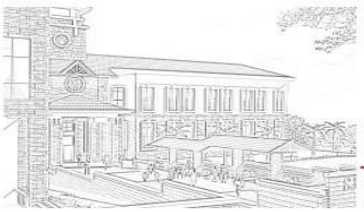
```
.width7 {
  width:200px;
}

.height1 {
  height:30px;
}

.align_right {
  text-align:right;
}

.bottom {
  border-bottom:1.5px dashed #000000;
}

</style>
</head>

<body>
<table style="margin-left:auto; margin-right:auto">
{% for Report in content %}
<tr>
<td colspan="6" style="border:1px solid #ffffff; border-top:1px solid #000000"><img src="School_Logo.png"></td>
```



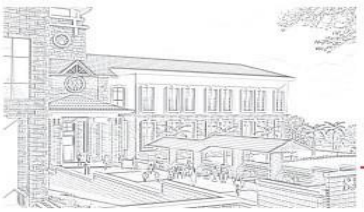
```
</tr>

<tr>
<td colspan="6" style="border:1px solid #ffffff">
<table id="inner">

  <tr>
  <td class="shade1 width1 height1">Name of the Examination</td>
  <th class="width2">{{Report.Exam}}</th>
  <td class="shade1 width3">Academic<br>Session</td>
  <th class="width4">2022 - 2023</th>
  </tr>

  <tr>
  <td class="shade1 height1">Name of the Student</td>
  <th>{{Report.Name}}</th>
  <td class="shade1">Class &<br>Sec</td>
  <th>XII - {{Report.Sec}}</th>
  </tr>

  <tr>
  <td class="shade1 height1">Name of the Class Teacher</td>
  <th>{{Report.Teacher}}</th>
  <td class="shade1">Date of<br>Birth</td>
  <th>{{Report.Date}}.{{Report.Month}}.{{Report.Year}}</th>
```



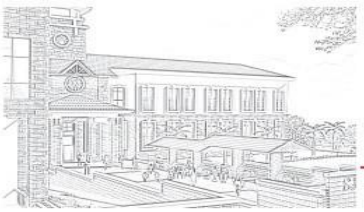
```
  </tr>

</table>
</td>
</tr>

<tr>
<th colspan="6" style="font-size:30px;border:1px solid #ffffff; border-bottom:1px solid #000000 !important;">SCHOLASTIC REPORT</th>
</tr>

<tr>
<th rowspan="2" class="shade2 width6">S.No.</th>
<th rowspan="2" class="shade2 width7">Subject</th>
<th rowspan="2" class="shade2 width5">Marks<br>Scored</th>
<th rowspan="2" class="shade2 width5">Grade</th>
<th colspan="2" class="shade3">8 Point Scale</th>
</tr>

<tr>
<th class="shade4 width6">Mark Range</th>
<th class="shade4 width6">Grade</th>
</tr>

<tr>
```



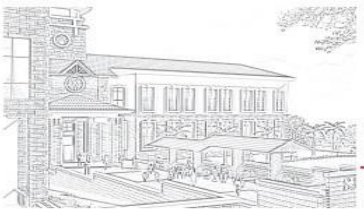
```
<td class="bottom height1">1.</td>
<th class="bottom" style="text-align:left; padding-left:15px">{{Report.Subject[0]}}</th>
<th class="bottom">{{Report.Marks[0]}}</th>
<th class="bottom">{{Report.Grade[0]}}</th>
<td class="shade5">91 - 100</td>
<td class="shade5">A1</td>
</tr>

<tr>
<td class="bottom height1">2.</td>
<th class="bottom" style="text-align:left; padding-left:15px">{{Report.Subject[1]}}</th>
<th class="bottom">{{Report.Marks[1]}}</th>
<th class="bottom">{{Report.Grade[1]}}</th>
<td class="shade5">81 - 90</td>
<td class="shade5">A2</td>
</tr>

<tr>
<td class="bottom height1">3.</td>
<th class="bottom" style="text-align:left; padding-left:15px">{{Report.Subject[2]}}</th>
<th class="bottom">{{Report.Marks[2]}}</th>
<th class="bottom">{{Report.Grade[2]}}</th>
<td class="shade5">71 - 80</td>
<td class="shade5">B1</td>
</tr>
```



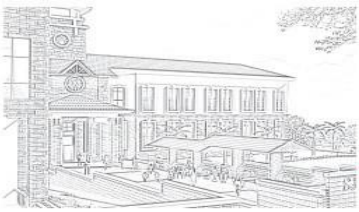
```
<tr>
<td class="bottom height1">4.</td>
<th class="bottom" style="text-align:left; padding-left:15px">{{Report.Subject[3]}}</th>
<th class="bottom">{{Report.Marks[3]}}</th>
<th class="bottom">{{Report.Grade[3]}}</th>
<td class="shade5">61 - 70</td>
<td class="shade5">B2</td>
</tr>

<tr>
<td class="bottom height1">5.</td>
<th class="bottom" style="text-align:left; padding-left:15px">{{Report.Subject[4]}}</th>
<th class="bottom">{{Report.Marks[4]}}</th>
<th class="bottom">{{Report.Grade[4]}}</th>
<td class="shade5">51 - 60</td>
<td class="shade5">C1</td>
</tr>

<tr>
<td class="bottom height1">6.</td>
<th class="bottom"></th>
<th class="bottom"></th>
<th class="bottom"></th>
<td class="shade5">41 - 50</td>
```



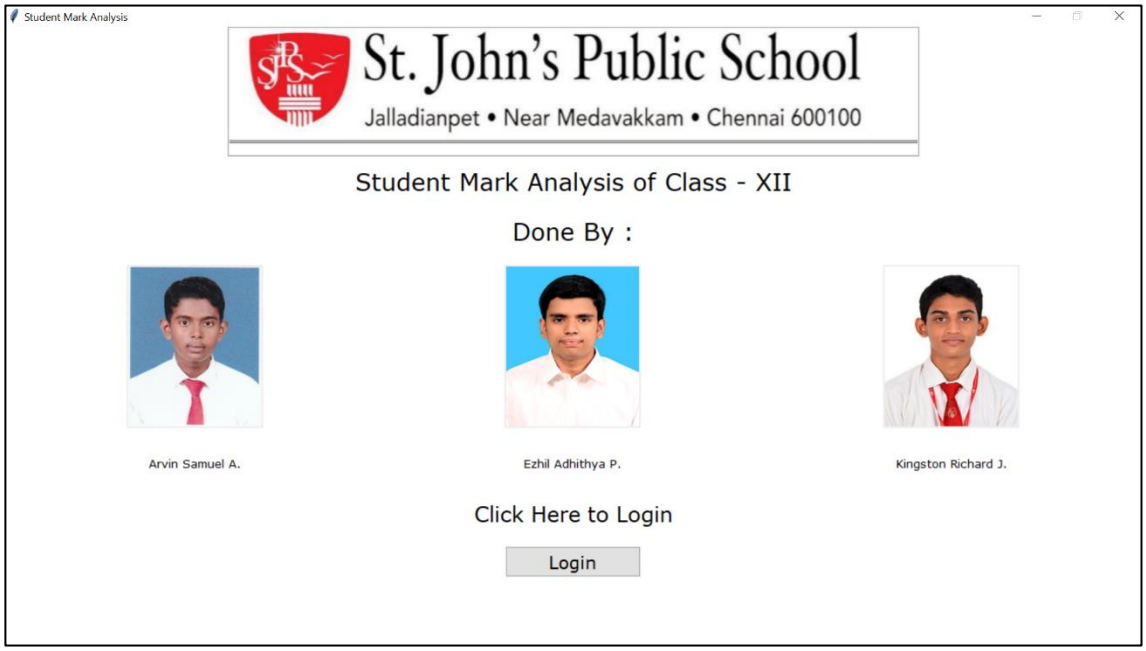
```
<td class="shade5">C2</td>
</tr>

<tr>
<td class="height1">7.</td>
<th></th>
<th></th>
<th></th>
<td class="shade5">33 - 40</td>
<td class="shade5">D</td>
</tr>

<tr>
<td colspan="2" class="shade1 height1">Total Marks</td>
<th colspan="2">{{Report.Total}}</th>
<td class="shade5">32 & Below</td>
<td class="shade5">E<br><font size="2">(Failed)</font></td>
</tr>

<tr>
<td colspan="2" class="shade1 height1">Overall %</td>
<th colspan="2">{{Report.Percent}}</th>
<td class="shade1">Maximum Marks</td>
<td>100 %</td>
</tr>
```



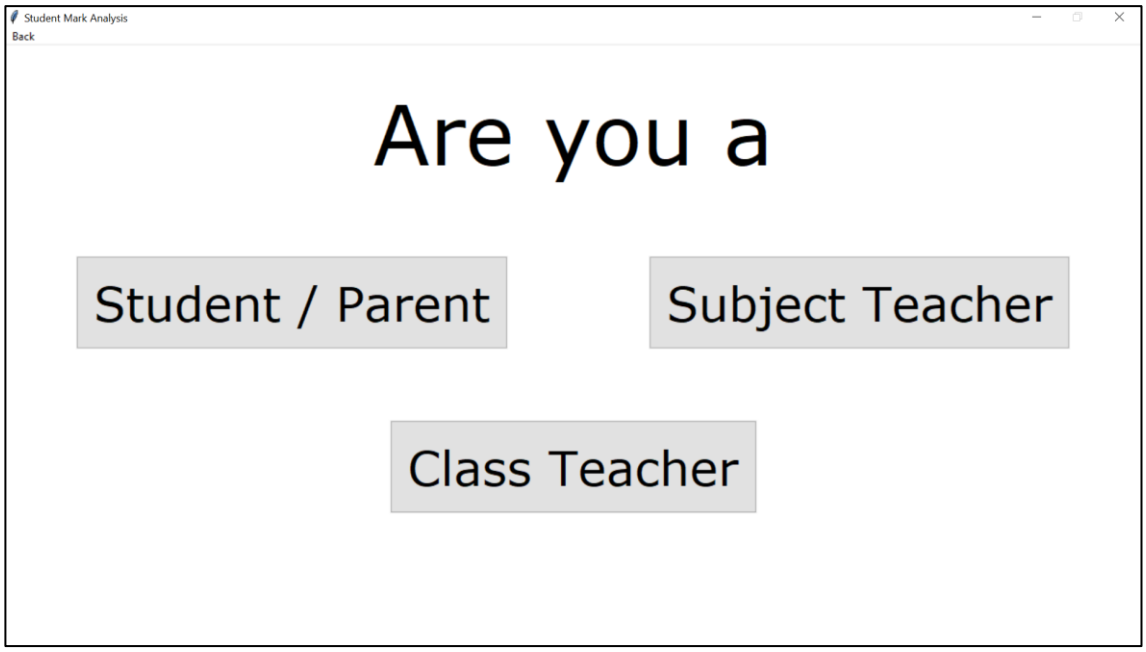
```
<tr>
<td colspan="2" class="shade1 height1">Rank Order</td>
<th colspan="2">{{Report.Rank}}</th>
<td class="shade1">Min. Pass Mark</td>
<td>33 %</td>
</tr>

<tr>
<td colspan="2" class="shade1 height1">Days</td>
<th colspan="2">{{Report.Atten[0]}}/{{Report.Atten[1]}} Days</th>
<td  class="shade1">Attendance %</td>
<th>{{Report.Atten_Percent}}%</th>
</tr>

<tr>
<th colspan="6" style="border-left:1px solid #ffffff; border-right:1px solid #ffffff"><img src="Signatures_Teachers.png"></th>
</tr>
{% endfor %}
</table>
</body>
</html>
```



OUTPUT

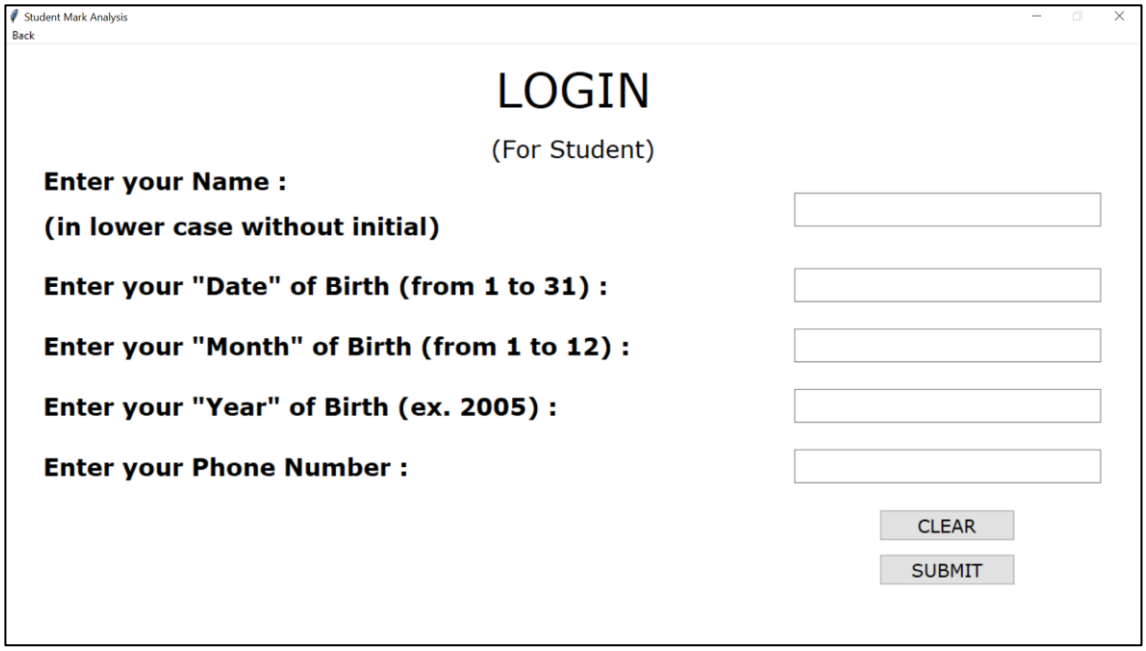


Home Page

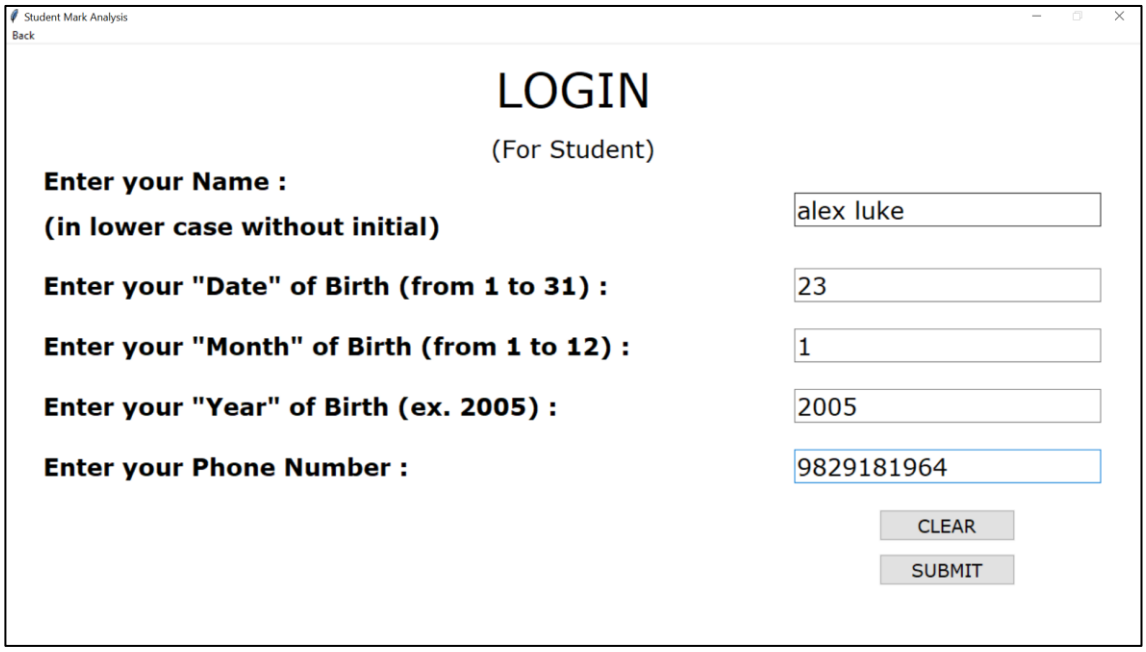


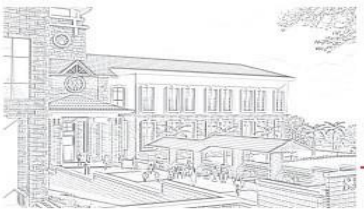
The Next Page after clicking Login Button

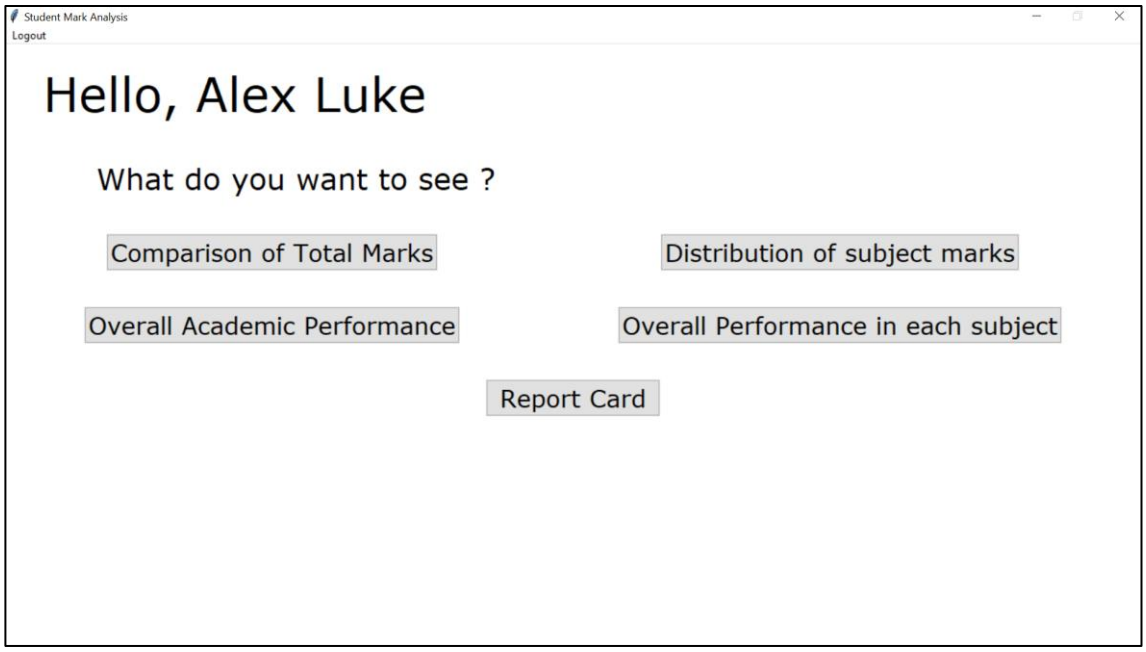
Student/Parent:



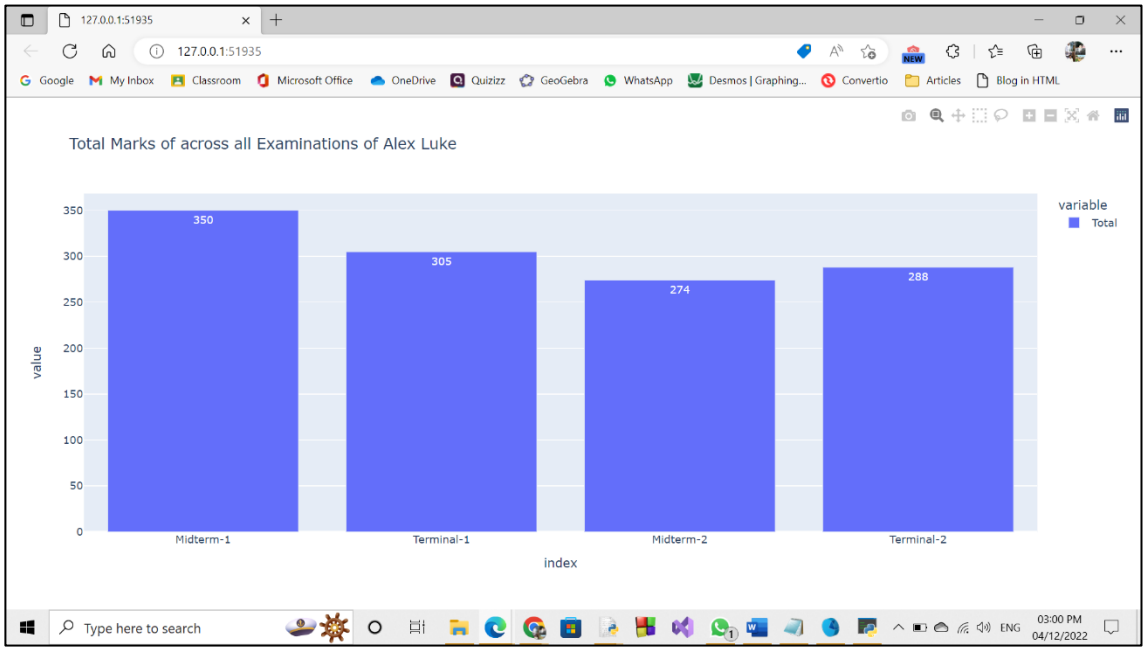
Login Page of the Student/Parent



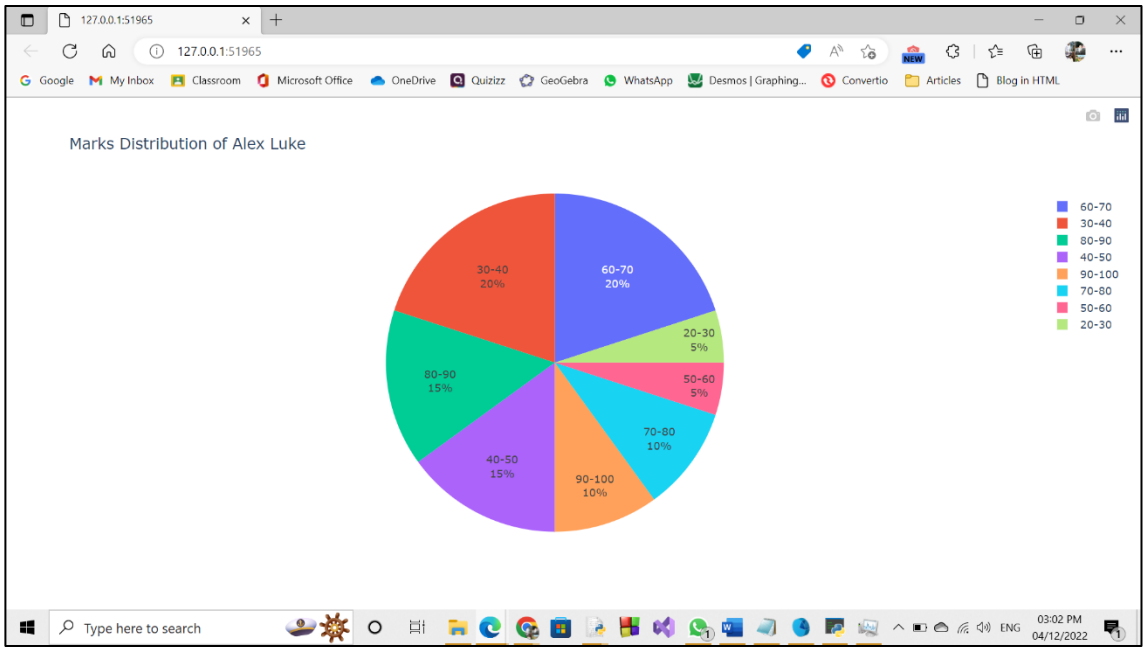




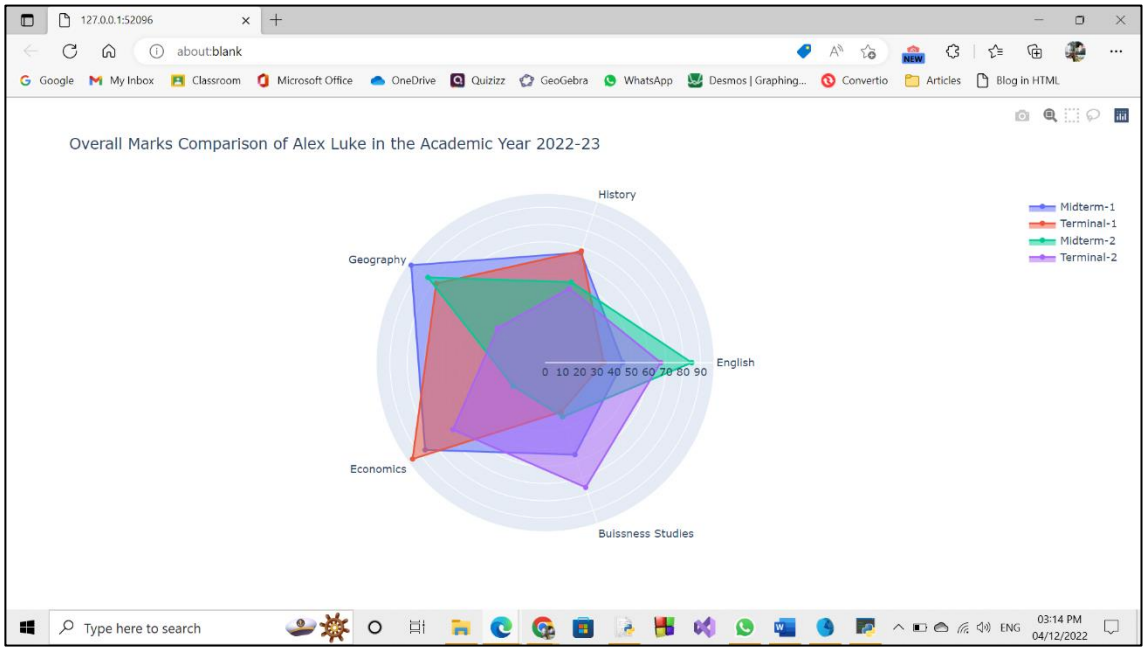
Page after clicking the Submit Button



The Graph after clicking Comparison of Total Marks Button

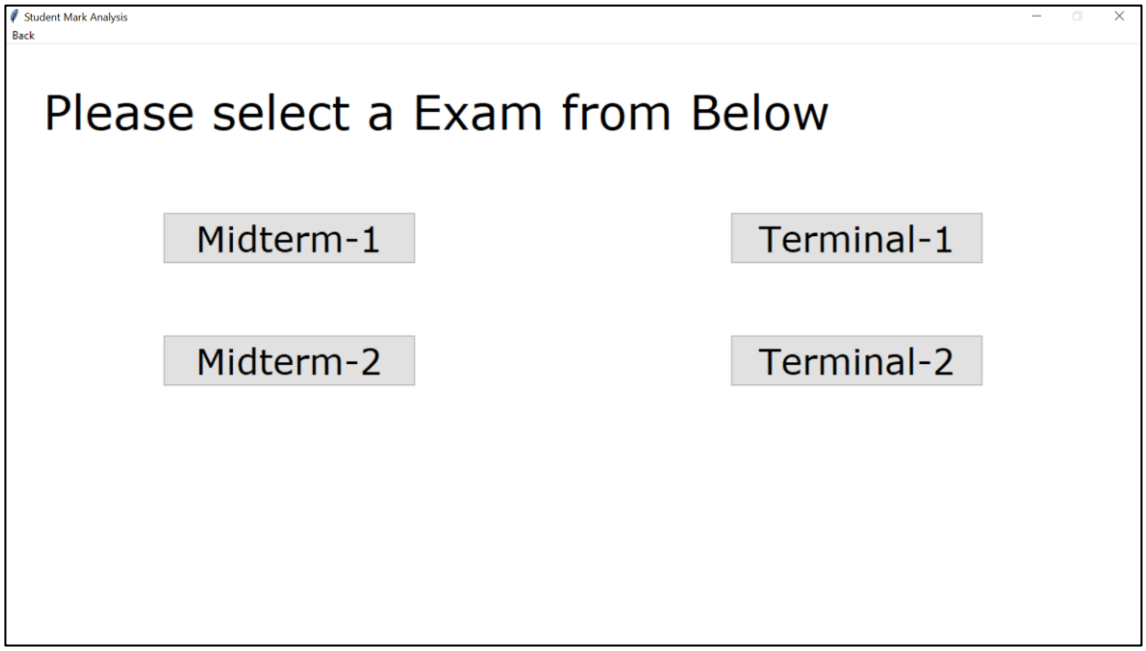


The Graph after clicking Distribution of subject marks Button

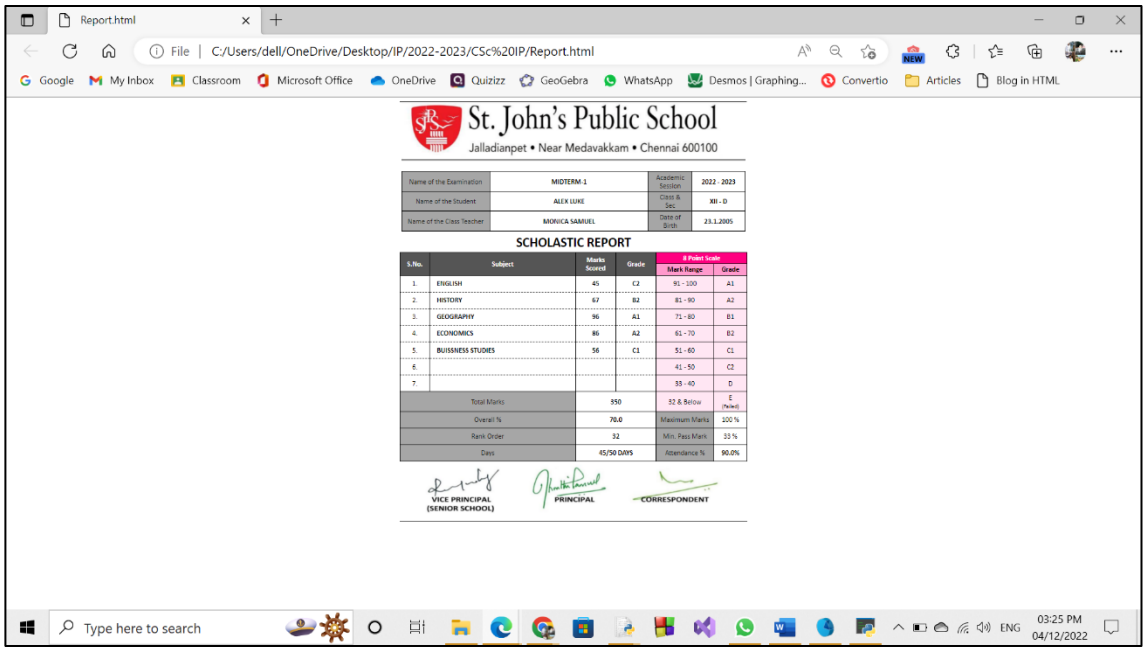


The Graph after clicking Overall Academic Performance Button

The Graph after clicking Overall Performance in each subject Button



The page after clicking Report Card Button



Report.html

File | C:/Users/dell/OneDrive/Desktop/IP/2022-2023/CSc%20IP/Report.html

Google | My Inbox | Classroom | Microsoft Office | OneDrive | Quizizz | GeoGebra | WhatsApp | Desmos | Graphing... | Convertio | Articles | Blog in HTML

St. John's Public School
Jalladianpet • Near Medavakkam • Chennai 600100

Name of the Examination	MIDTERM-1	Academic Session	2022 - 2023
Name of the Student	ALEX LUKE	Class & Sec	XII - D
Name of the Class Teacher	MONICA SAMUEL	Date of Birth	23.1.2005

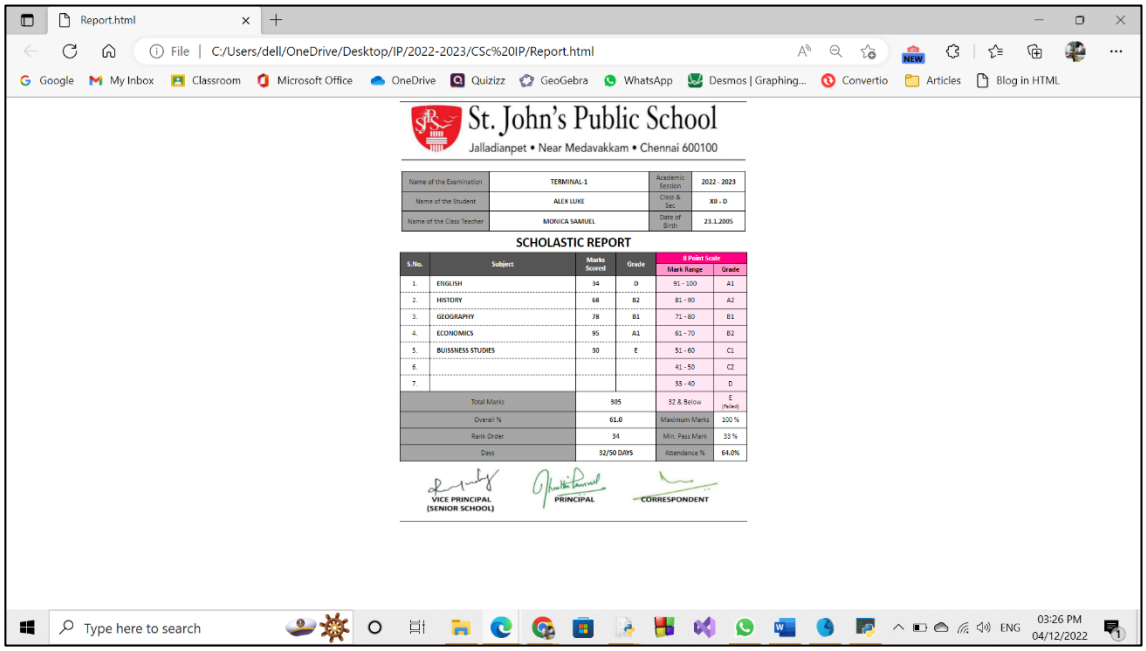
SCHOLASTIC REPORT

S.No.	Subject	Marks Scored	Grade	8 Point Scale	
				Mark Range	Grade
1.	ENGLISH	45	C2	91 - 100	A1
2.	HISTORY	67	B2	81 - 90	A2
3.	GEOGRAPHY	96	A1	71 - 80	B1
4.	ECONOMICS	86	A2	61 - 70	B2
5.	BUISSNESS STUDIES	56	C1	51 - 60	C1
6.				41 - 50	C2
7.				33 - 40	D
Total Marks		550		32 & Below	E (Failed)
Overall %		70.0		Maximum Marks	100 %
Rank Order		32		Min. Pass Mark	33 %
Days		45/50 DAYS		Attendance %	90.0%

VICE PRINCIPAL (SENIOR SCHOOL) | PRINCIPAL | CORRESPONDENT

Type here to search | ENG | 03:25 PM 04/12/2022

The Report Card after clicking Midterm-1



Report.html

File | C:/Users/dell/OneDrive/Desktop/IP/2022-2023/CSc%20IP/Report.html

Google | My Inbox | Classroom | Microsoft Office | OneDrive | Quizizz | GeoGebra | WhatsApp | Desmos | Graphing... | Convertio | Articles | Blog in HTML

St. John's Public School
Jalladianpet • Near Medavakkam • Chennai 600100

Name of the Examination	TERMINAL-1	Academic Session	2022 - 2023
Name of the Student	ALEX LUKE	Class & Sec	XII - D
Name of the Class Teacher	MONICA SAMUEL	Date of Birth	23.1.2005

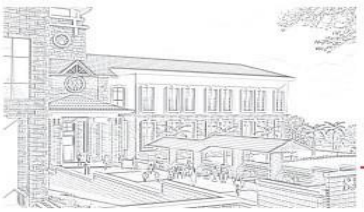
SCHOLASTIC REPORT

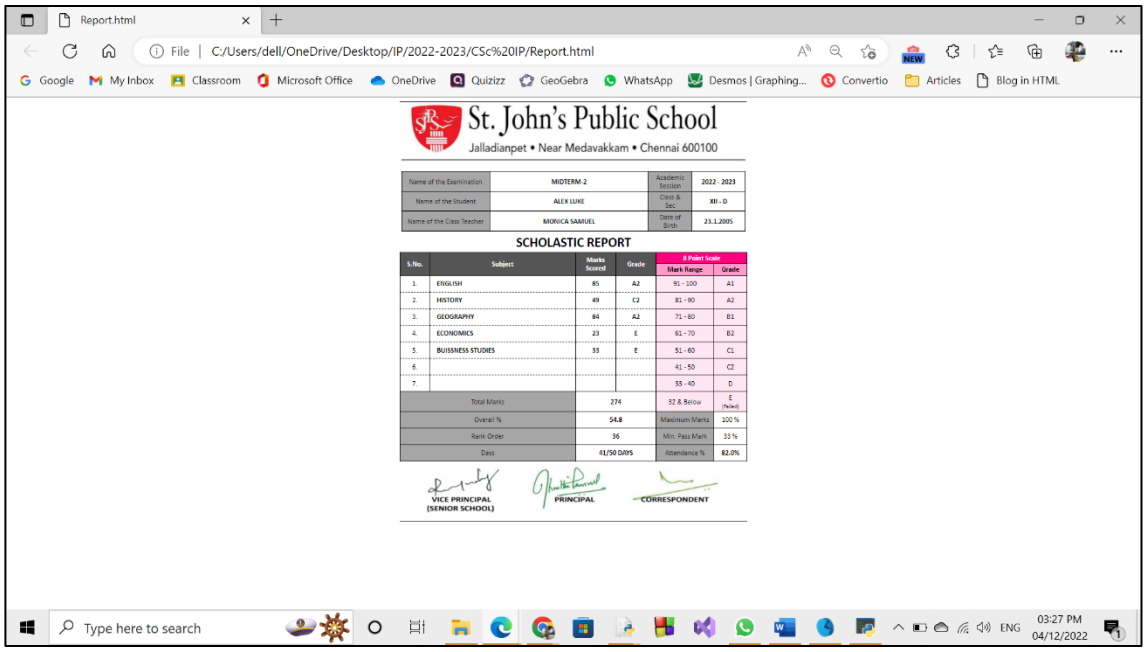
S.No.	Subject	Marks Scored	Grade	8 Point Scale	
				Mark Range	Grade
1.	ENGLISH	34	D	91 - 100	A1
2.	HISTORY	68	B2	81 - 90	A2
3.	GEOGRAPHY	78	B1	71 - 80	B1
4.	ECONOMICS	95	A1	61 - 70	B2
5.	BUISSNESS STUDIES	30	E	51 - 60	C1
6.				41 - 50	C2
7.				33 - 40	D
Total Marks		305		32 & Below	E (Failed)
Overall %		61.0		Maximum Marks	100 %
Rank Order		34		Min. Pass Mark	33 %
Days		32/50 DAYS		Attendance %	64.0%

VICE PRINCIPAL (SENIOR SCHOOL) | PRINCIPAL | CORRESPONDENT

Type here to search | ENG | 03:26 PM 04/12/2022

The Report Card after clicking Terminal-1





St. John's Public School
Jalladianpet • Near Medavakkam • Chennai 600100

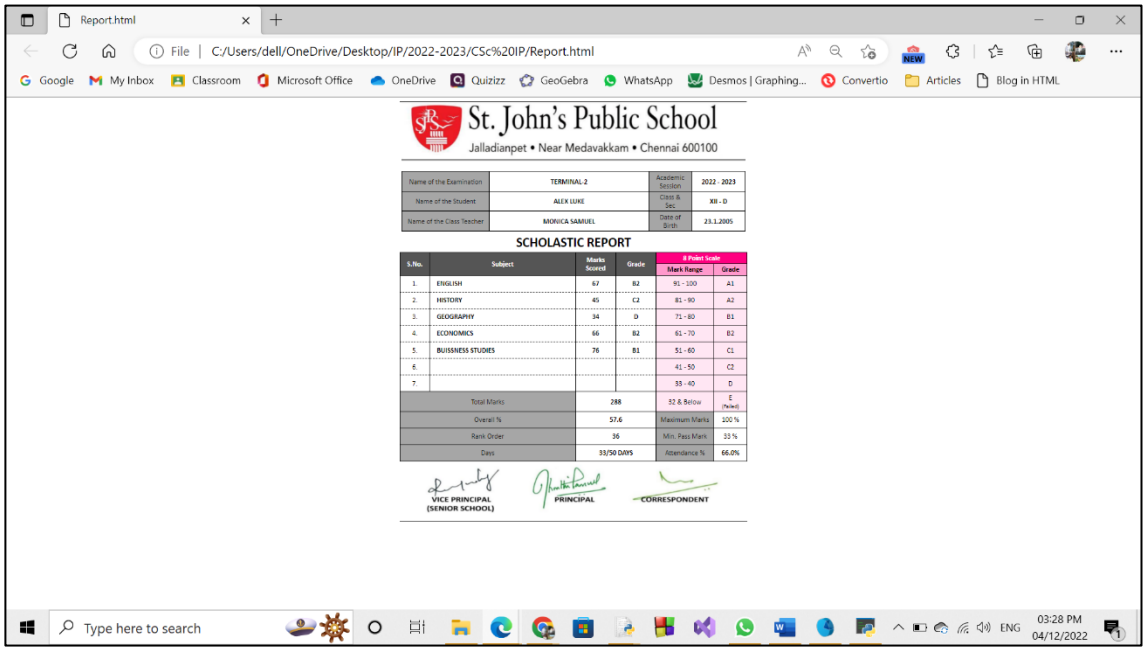
Name of the Examination	MIDTERM-2	Academic Session	2022 - 2023
Name of the Student	ALEX LUKE	Class & Sec	XII - D
Name of the Class Teacher	MONICA SAMUEL	Date of Birth	23.1.2005

SCHOLASTIC REPORT

S.No.	Subject	Marks Scored	Grade	8 Point Scale	
				Mark Range	Grade
1.	ENGLISH	85	A2	91 - 100	A1
2.	HISTORY	49	C2	81 - 90	A2
3.	GEOGRAPHY	84	A2	71 - 80	B1
4.	ECONOMICS	23	E	61 - 70	B2
5.	BUISSNESS STUDIES	33	E	51 - 60	C1
6.				41 - 50	C2
7.				33 - 40	D
Total Marks		274		32 & Below	E (Failed)
Overall %		54.8		Maximum Marks	100 %
Rank Order		36		Min. Pass Mark	33 %
Days		41/50 DAYS		Attendance %	82.0%

VICE PRINCIPAL (SENIOR SCHOOL) PRINCIPAL CORRESPONDENT

The Report Card after clicking Midterm-2



St. John's Public School
Jalladianpet • Near Medavakkam • Chennai 600100

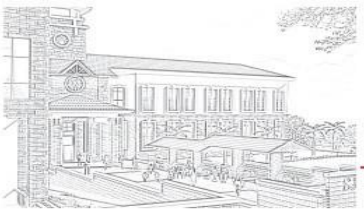
Name of the Examination	TERMINAL-2	Academic Session	2022 - 2023
Name of the Student	ALEX LUKE	Class & Sec	XII - D
Name of the Class Teacher	MONICA SAMUEL	Date of Birth	23.1.2005

SCHOLASTIC REPORT

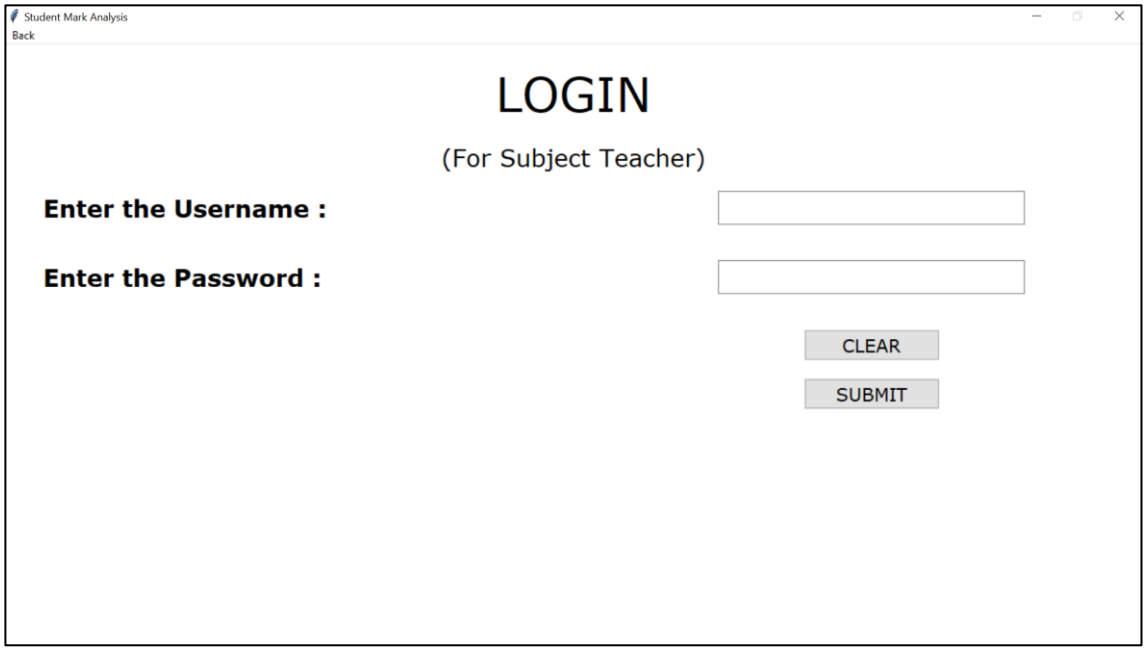
S.No.	Subject	Marks Scored	Grade	8 Point Scale	
				Mark Range	Grade
1.	ENGLISH	67	B2	91 - 100	A1
2.	HISTORY	45	C2	81 - 90	A2
3.	GEOGRAPHY	34	D	71 - 80	B1
4.	ECONOMICS	66	B2	61 - 70	B2
5.	BUISSNESS STUDIES	76	B1	51 - 60	C1
6.				41 - 50	C2
7.				33 - 40	D
Total Marks		288		32 & Below	E (Failed)
Overall %		57.6		Maximum Marks	100 %
Rank Order		36		Min. Pass Mark	33 %
Days		33/50 DAYS		Attendance %	66.0%

VICE PRINCIPAL (SENIOR SCHOOL) PRINCIPAL CORRESPONDENT

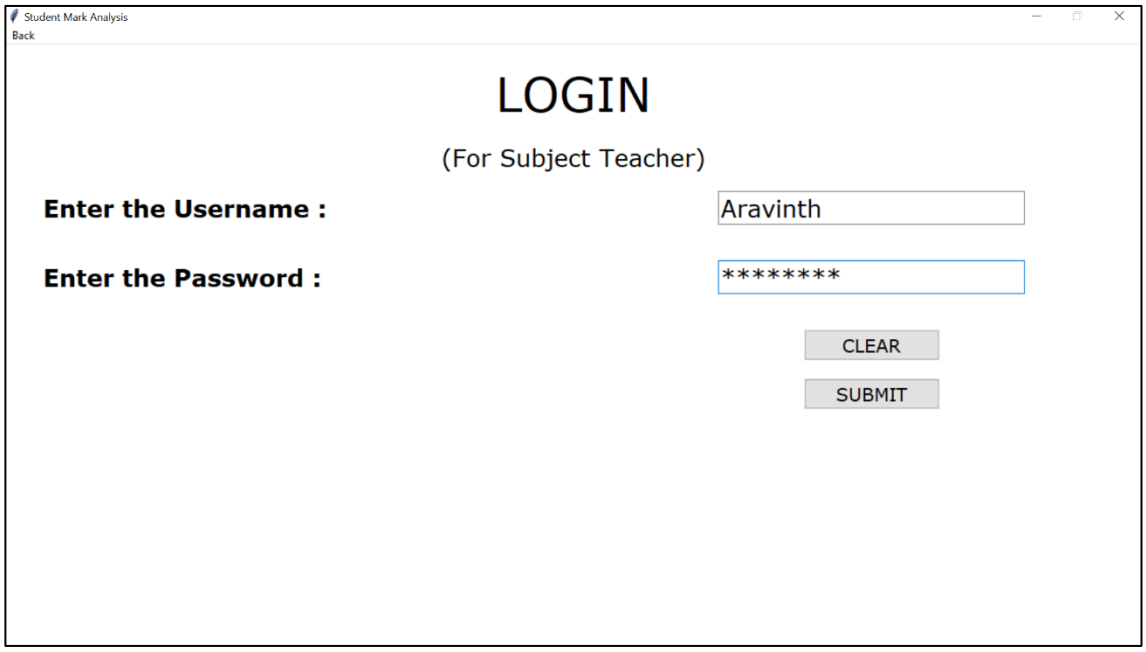
The Report Card after clicking Terminal-2

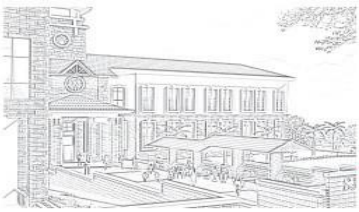


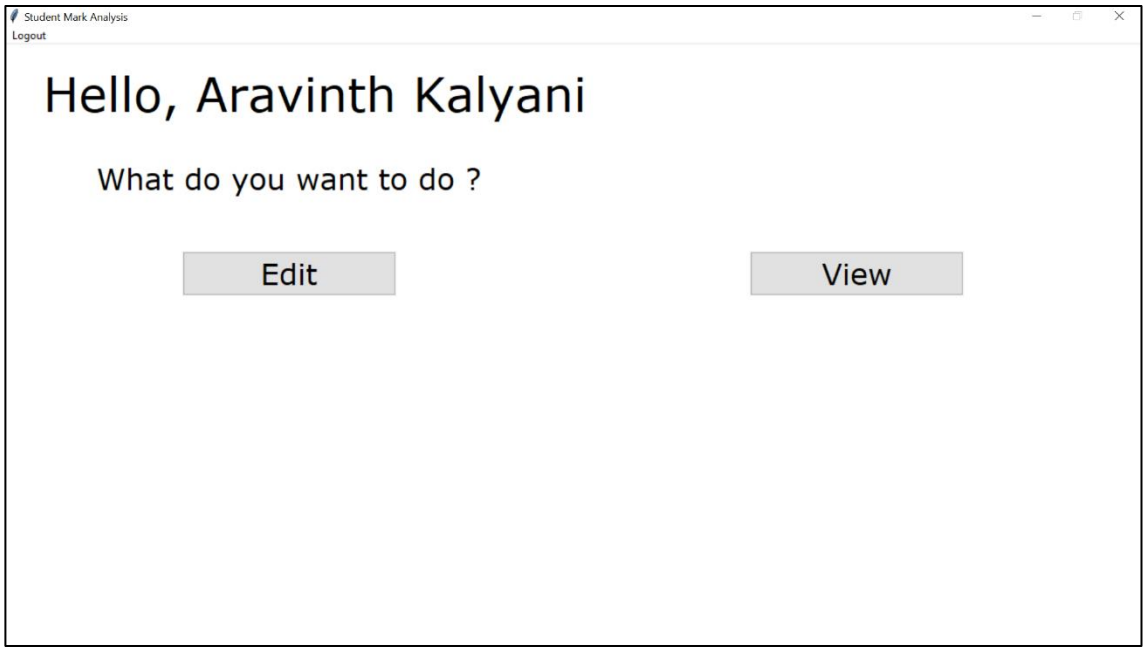
Subject Teacher:



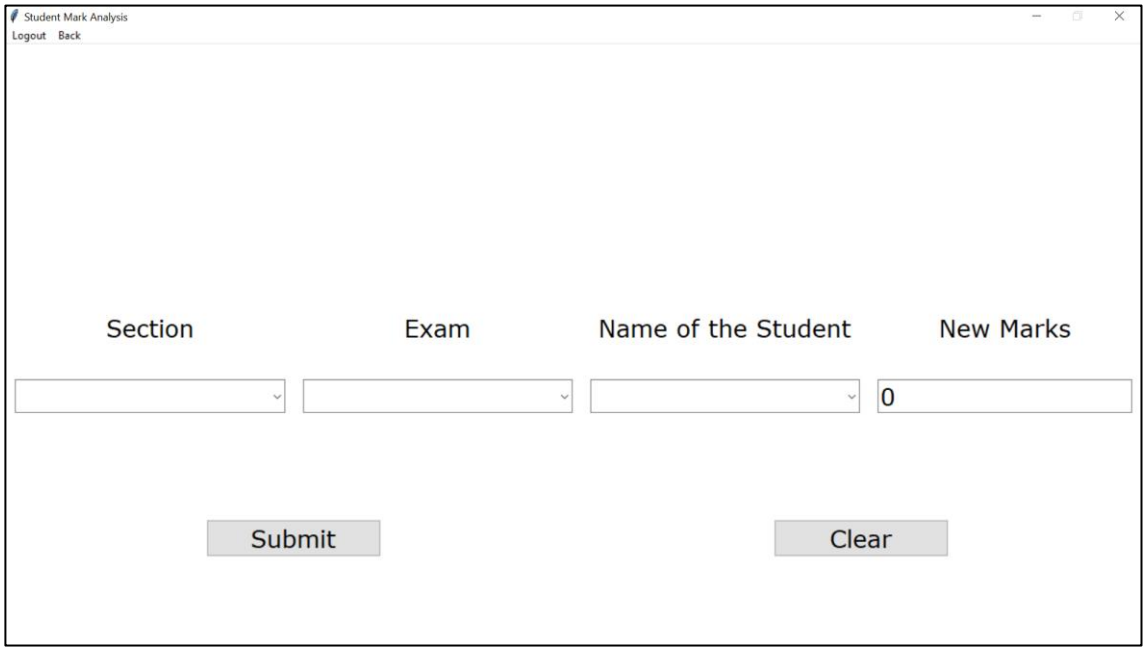
Login Page of the Subject Teacher



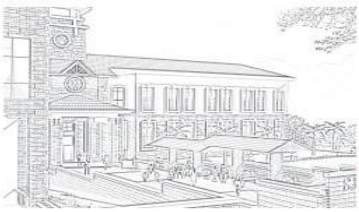


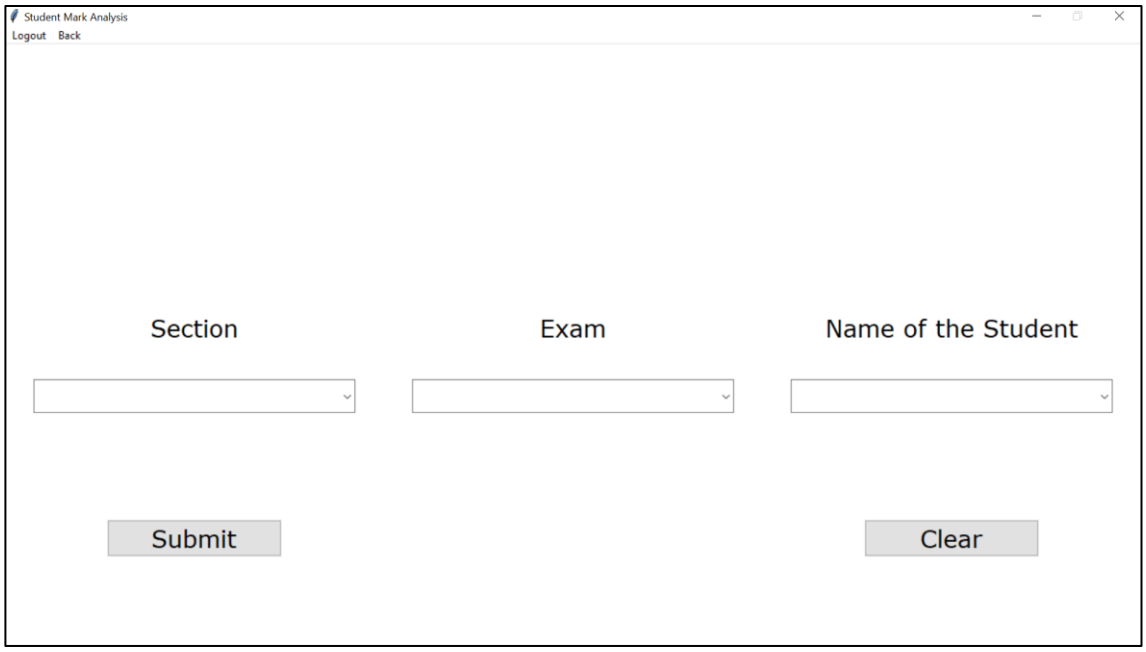


Page after clicking the Submit Button



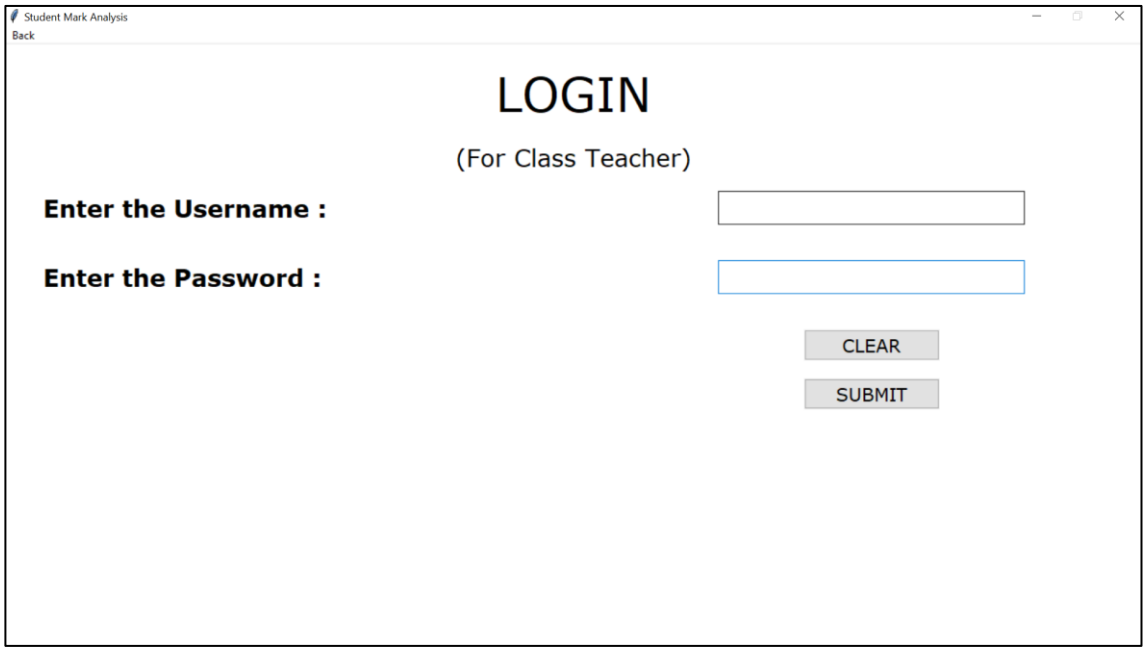
Page after clicking the Edit Button



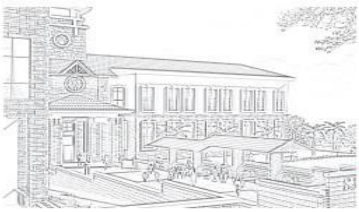


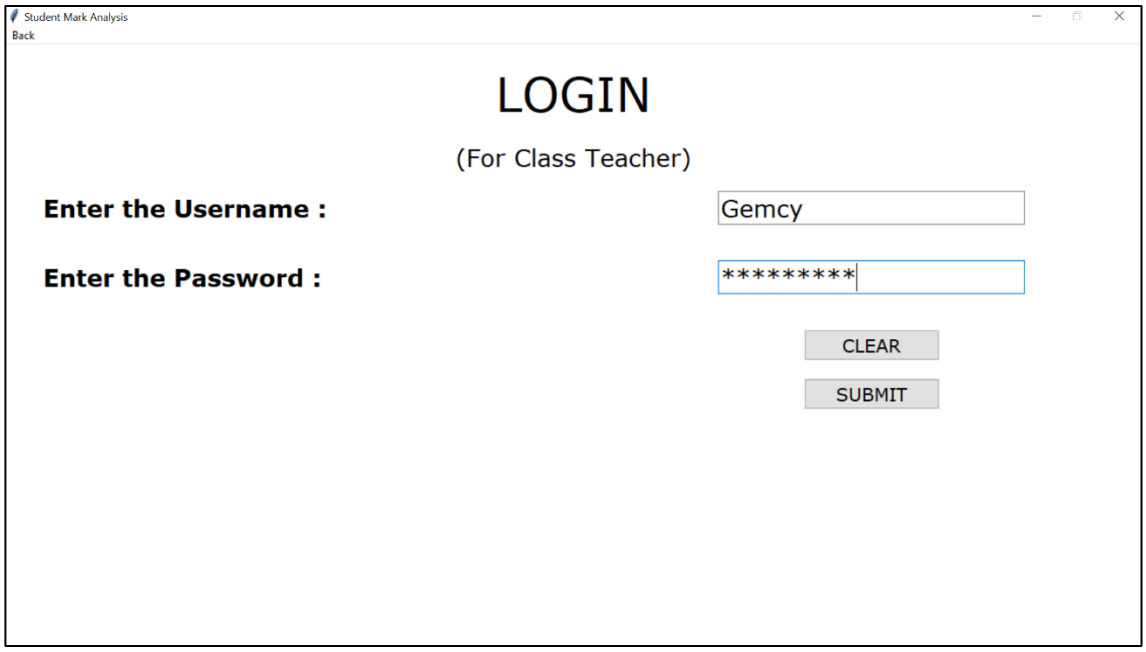
Page after clicking the View Button

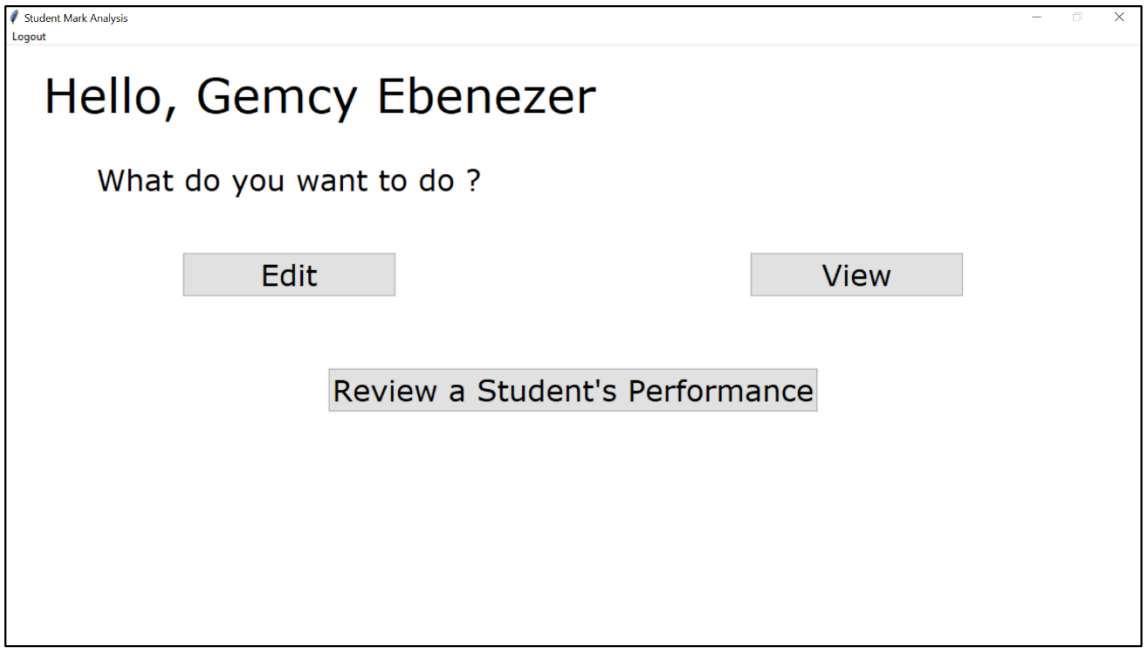
Class Teacher:



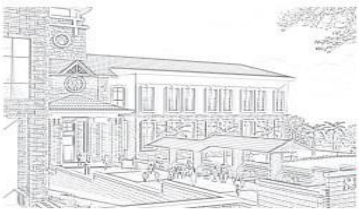
Login Page of the Class Teacher

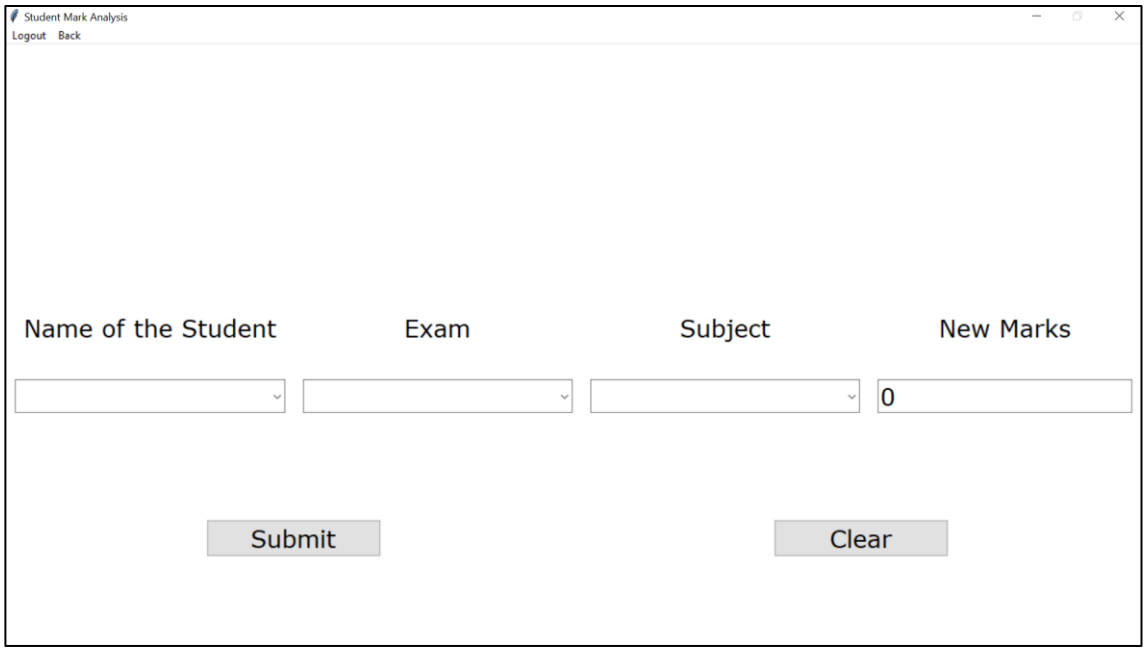




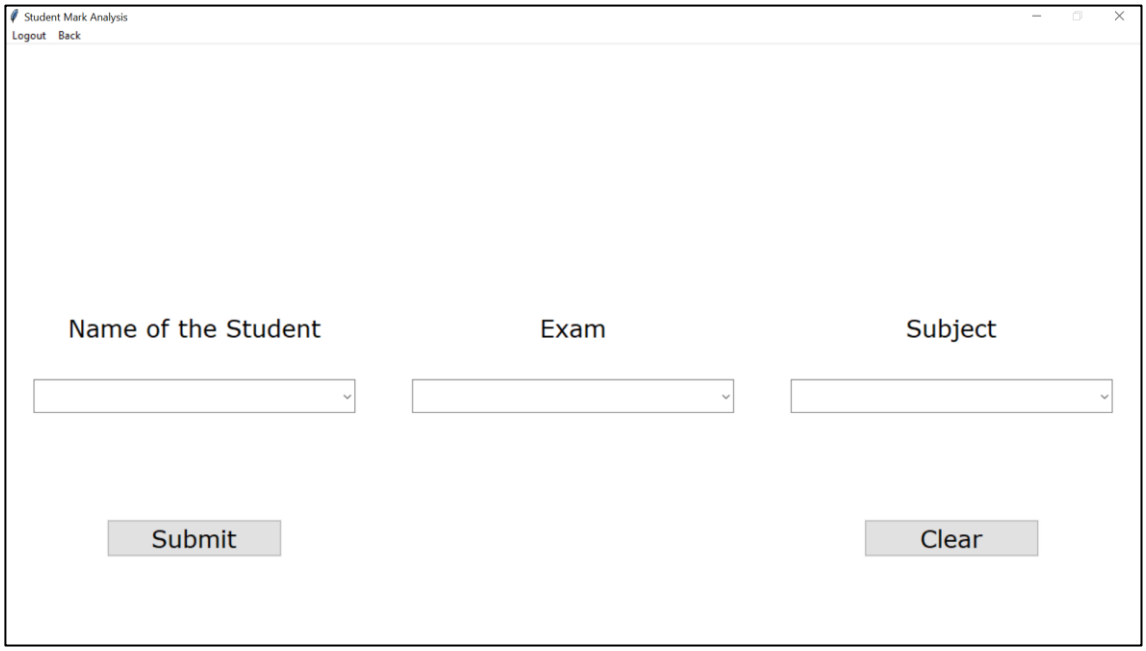


Page after clicking the Submit Button

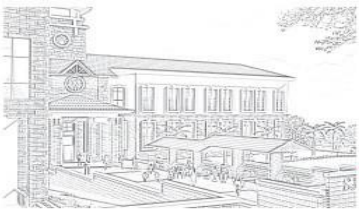


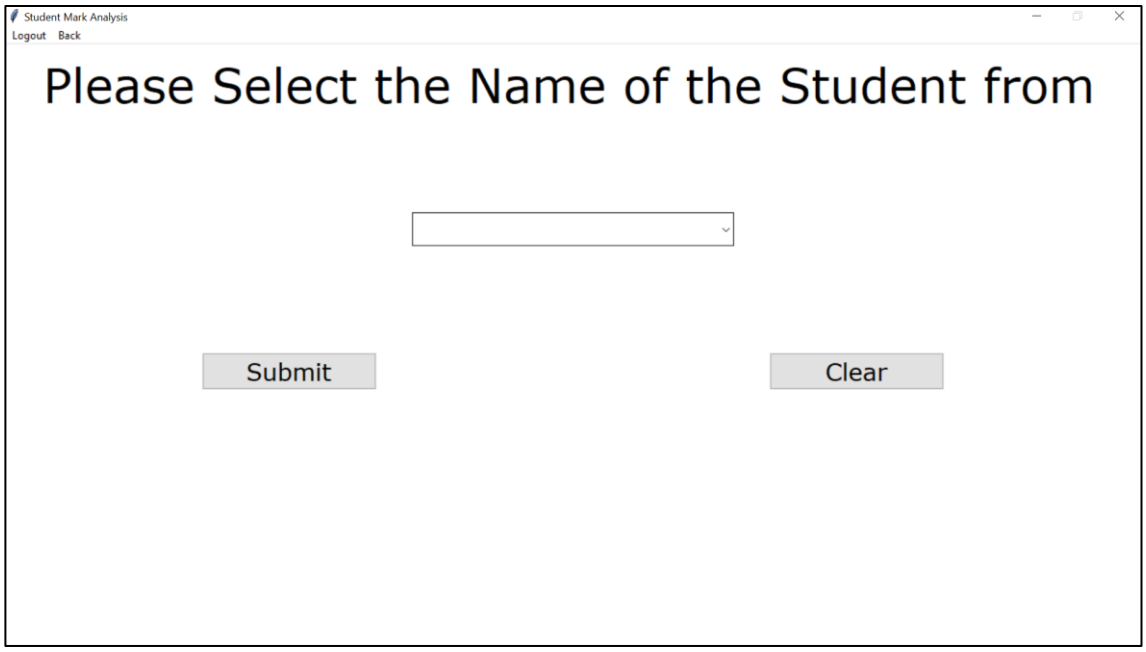


Page after clicking the Edit Button

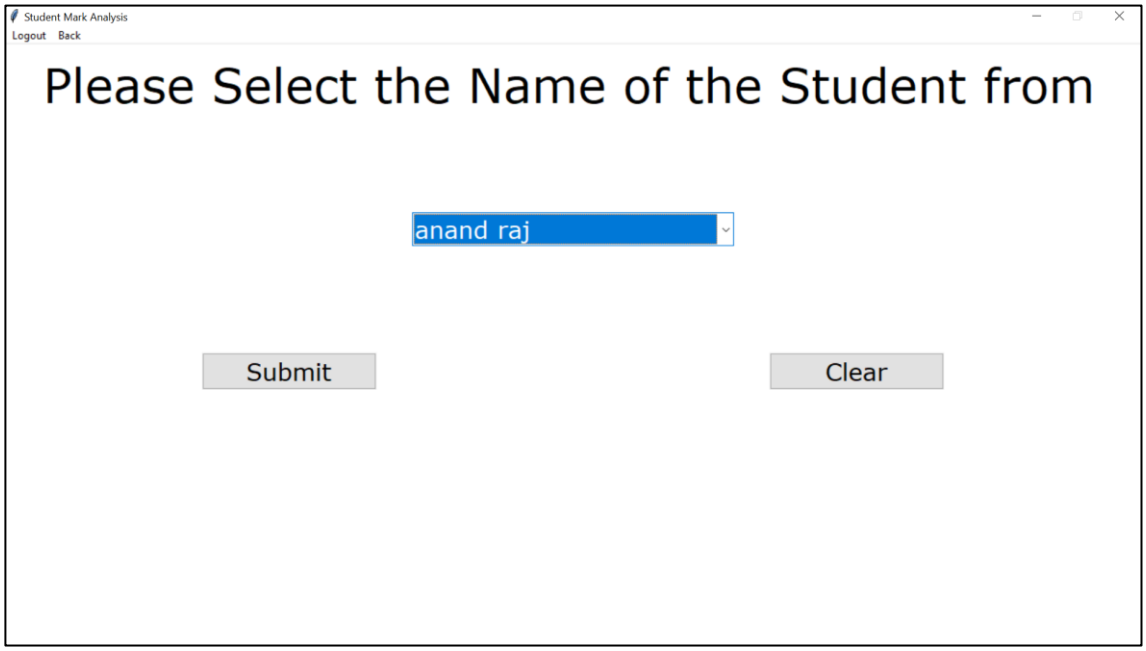


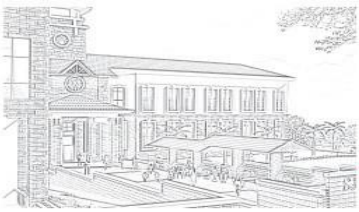
Page after clicking the View Button

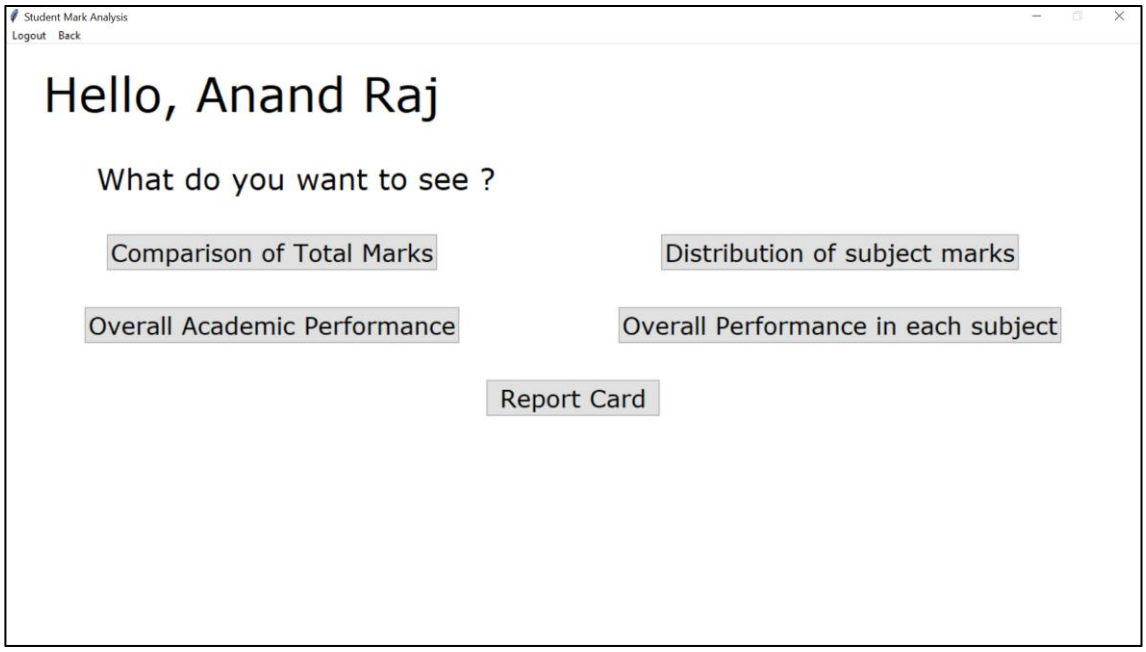




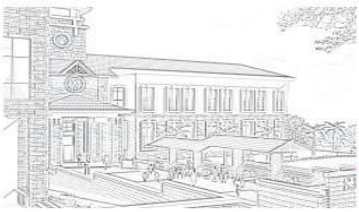
Page after clicking the Review a Student's Performance Button





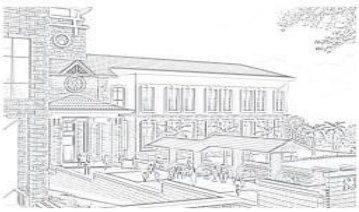


Page after clicking the Submit Button



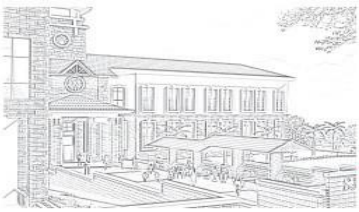
ADVANTAGES OF THE PROJECT

1. It helps the student to analyse their performance in the exam.
2. It helps the parents to understand the learning progress of the child.
3. It shows the improvement or downfall of the student in the current exam on comparing with the previous examinations.
4. It compares the performance of the student with others of the same class, so they could really know where they stand.
5. It helps the teachers to identify the potential of the students and train them accordingly.
6. It improves the technical aspects of all subject teachers as this allows them to know more about the usage of computers.
7. It is more efficient than the traditional report card/sheets.
8. It helps in reducing a lot of work and saves the time of teachers who usually sit with sheets of data, trying to tally the student's marks.
9. Since the data (marks) is represented in pictorial form, students, parents and teachers tend to understand the learning trend of the student very easily.



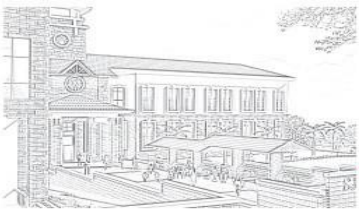
LIMITATIONS OF THE PROJECT

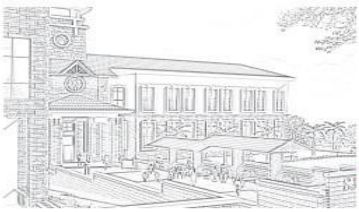
1) The Project is slower because the data retrieval and processing is slow as Python is Interpreted Programming Language.
2) Adding New data tends to be more difficult.
3) The Project cannot run on other computers as it requires the pre-installation of various modules in Python.



FURTHER DEVELOPMENT AREAS

- Use of the programming language SQL, instead of Binary File, improves the maintenance of the marks, scored by the class XII students in all the examinations, efficiently.
- Multi-Processing can be used which will exponentially increase the performance of the Application.
- Use of the programming language SQL also improves the number of marks to be stored and the marks scored by each student of class XII can also be retrieved easily.
- Use of the markup language such as HTML along with the usage of CSS and programming language like JavaScript improves the user - interface of the Student Mark Analysis for the teachers and for the students of class XII as well.
- The Student Mark Analysis can also be extended for all the students in the whole school.



BIBLIOGRAPHY

- https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python
- https://docs.python.org/
- https://www.w3schools.com/python/
- https://www.tutorialspoint.com/python/index.htm
- https://www.javatpoint.com/python-tutorial
- https://www.geeksforgeeks.org/
- https://www.codegrepper.com/
- https://www.coderslegacy.com/
- https://www.stackoverflow.com/
- https://pythonprogramming.altervista.org/
- https://www.pythontutorial.net/
- https://pythonguides.com/
- https://www.askpython.com/python/
- https://www.oreilly.com/library/view/python-gui-programming/