CHAPTER 1

INTRODUCTION

The objective of College Information Management System is to allow the administrator of any organization the ability to edit and find out the personal details of a student and allows the student to keep up to date his profile. It’ll also facilitate keeping all the records of students, such as their id, name, mailing address, phone number, DOB etc. So all the information about a student will be available in a few seconds. Overall, it’ll make Student Information an easier job for the administrator and the student of any organization. The main purpose of this project is to illustrate the requirements of the project College Information Management System and is intended to help any organization to maintain and manage personal data. It is a comprehensive project developed from the ground up to fulfill the needs of colleges as they guide their students. This integrated information management system connects daily operations in the college environment ranging from Attendance management to communicational means among students and teachers. This reduces data error and ensures that information is always up-to-date throughout the college. It provides a single source of data repository for streamlining your processes and for all reporting purposes. It has a simple user interface and is intuitive. This insures that the users spend less time in learning the system and hence, increase their productivity. Efficient security features provide data privacy and hence, increase their productivity.

**CHAPTER 2**

**SYSTEM ANALYSIS**

**2.1 EXISTING SYSTEM**

A basic ERP is based on a web application. Web applications need to be hosted on a server, and another storage space has to be allocated to store information. On the server side, the script or source code of the system is to be maintained securely, and the data needs to be encrypted.

On the client side, the user must have a stable internet connection to login, even if the server is located locally on the campus or university. You cannot use the ERP in offline mode. If you need to login to the system every time, the server needs to verify the user and give access to the client.

**2.1.1 Disadvantages**

* Need to host the system from a server
* The system is not accessible in offline
* If the server is offline we cannot connect to the system
* The internet connection must be strong enough to access a website

**2.2 PROPOSED SYSTEM**

The proposed system, the ERP system, aims to establish an efficient and effective mechanism for collecting, analyzing, and utilizing the student information. The system is going to be an executable software file thought there is no need to have a server to host a system. The system is designed based on offline portable application hence it also work on offline. The internet connection is required for user verification and data access only.

**2.2.1 Advantages**

* We can access the system in offline
* Data transfer even with unstable internet
* One-time login and logout process
* The data is secure
* No need to verify the user for every time

**CHAPTER 3**

**SYSTEM REQUIREMENTS**

**3.1 User Interfaces**

The User interface is made using Tkinter. Firstly, there will be a simple login page separate for students and teachers. Each student and teacher will have a unique interface. There will be a fixed sidebar with links to all the modules. The teachers will be able to view their respective students and update their attendance and marks using an effortless interface.

**3.2 Hardware Interfaces**

* A basic requirements for a system

Since the system have any designated hardware, it does not have any other direct hardware interfaces.

**3.3 Software Interfaces**

The system is completely independent of other software or applications. So there is no software requirement

Operating System: We have chosen Windows operating system for its best support and user- friendliness.

Tkinter: We have chosen to use Tkinter for the back-end of the software as Tkinter is a simple python framework and is suitable for beginners.

Database: We are using SQLite database, which comes as default with Tkinter.

**3.4 Communications Interfaces**

This project is to be deployed an online database centre. All the users can connect to the database server from anywhere and have access to their information.

**3.5 Operating Environment**

The operating environment for College ERP system are listed below:

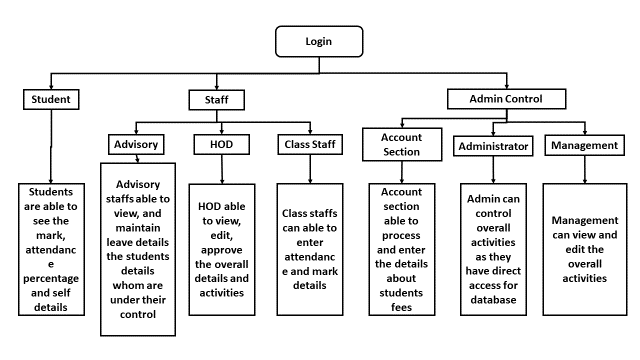
* Operating System: Windows 10
* Database: MySQL database
* Front-end: Tkinter(python)
* Back-end: RegEx(regular expression algorithm, mysql.connector)

**CHAPTER 4**

**SYSTEM DESIGN**

**4.1 ARCHITECTURAL DESIGN**

The ERP software requires the architectural design to represent the design of the software. Here we define a collection of hardware and software components and their interfaces to establish the framework for the development of this software. There exists number of components of the system which are integrated to form a system. The set of connectors will help in coordination, communication, and cooperation between the components. The ERP software is built for computer based system. It exhibits the data centric style of architecture.

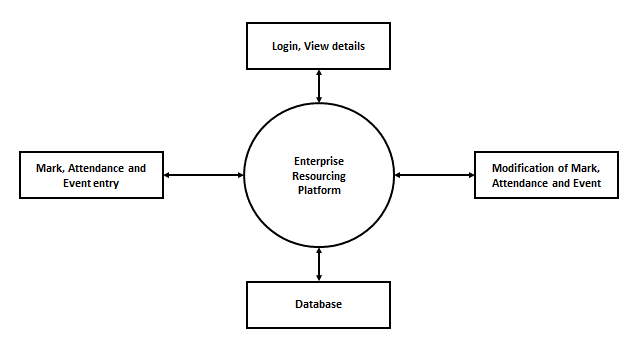
**4.1.1 Architectural Style**

**Figure No. 4.1 Architectural Diagram**

In the college ERP software, the database stores the data of all the students and faculties and the stored data is updated, added, deleted or modified. So it exhibits the **data centric architectural style.**

In this architecture different components communicate with the shared data repository. The components access a shared data structure and are relatively independent.

**4.2 DATA FLOW**



**Figure No. 4.2 Data flow Diagram**

**4.2.1 Central data**

Also known as data store or data repository, which is responsible for providing permanent data storage. It represents the current state. It stores the information of students, attendance of students and faculties of each day, salary of all the faculties etc...

**4.2.2 Data accessory**

Data accessory one of the components, they are also called as clients. A data accessory operates on the central data store, perform computations, and might put back the results. Which includes students, faculties and administrator. Students requests to access the data from the repository and gets the request serviced. Faculty members modify the data in the repository. Administrator can add or delete the clients.

**4.2.3 Interface**

Interface is the connecting component between data repository and clients’ client interact with the data through the web server. The operation of one client does not depend on the others. They are independent of each other. This data-centered architecture will promote inerrability. This means that the existing components can be changed and new client components can be added to the architecture without the permission or concern of other clients. Addition of removal of students and faculties can be done without the permission of other students and faculties.

**4.2.4 Login**

Each student in the college is assigned a unique username and password by the administrator. The user- name is the same as their USN and so is the password. They may change it later according to their wish.

**CHAPTER 5**

**MODULE DESCRIPTION**

**List of Modules**

* Students
* College Staff
* Teaching Staff
* Head of Department (HOD)

**5.1 Students Module**

Students are end users of ERP software. The attendance, internals marks uploaded by the teachers are viewed by students. It helps them track their attendance status. It also helps them to communicate with teachers and their classmates. So students make up another set of stakeholders of this software.

**5.2 Administrator Module**

College administrator is responsible for maintaining the database of the college. They will have the privilege to modify the database i.e., to add/remove. An admin holds all the privileges of the ERP system. The admin has access to all the databases in the system. These include student database, teacher database, courses database and several others. Their job is to maintain the systems and addressing the problems faced by the other users. The admin needs adequate resources and the right tools. The admin expects a simple interface where they can easily access the required information.

**5.3 College staff Module**

College staff are key stakeholders and use the ERP system the most. Thus, it is essential to cater to their needs first. Among the staff there are several different roles. For each role, The ERP system will have a different view based on the requirements of that group. Among the staff the requirements of the various groups are described below.

**5.4 Teaching staff Module**

Teaching staff make up most of the staff. A teacher expects the ERP system to be easy to use, reliable and reduce the work load. Each teacher belongs to department and are assigned to a class of students with a course. So, the teacher should only be able to view and manipulate the data of the students that they are assigned to. The teachers’ involvement In the ERP System, is to enter the attendance, the internal marks, the semester end examination marks. They will also have other features which include availing leave and managing a lecture plan for each course. For Attendance management, the teachers expect a compact and functional interface. An interface where teachers use minimal effort to manage the attendance status of the students. The features expected for the attendance are to ability to enter the attendance to the entire class at once, edit the attendance of each individual student. Also, in the event of leave, they should be given an option of assigning the class to another teacher, who takes a course for the same class. In the event of entering internal marks and semester end examination marks, the teacher enters the marks for each individual student. This is initial a draft and can be edited. The students review the marks and verify. If there are any mistakes, the student notifies the teacher and the mistakes are corrected. After certain amount of time, when all the marks are confirmed, the marks are ‘locked’. i.e., after locking, the marks cannot to change. When a teacher applies for a leave, there are many options for different categories of leave. The first category is casual leave, this is for general purposes. Restricted leave can be availed only on specific days given by the college. Then, Sick leave is for when the teacher is ill. Lastly, earned leave is an option given to each teacher for a period of 15 days.

**5.5 Head of Department (HOD) Module**

The head of department is a part of the teaching staff but has special privileges. They manage the operations of each department. The HOD could still conduct courses for students. So, they will have all the features given to a teacher. Also, as the HOD, they will have access to the records of every teacher, courses, students who belong to the same department.

**CHAPTER 6**

**SYSTEM FEATURE AND CLASSES**

**6.1 Product Features**

* Each teacher will be able to enter attendance and marks for their respective Students.
* Each student will be able to view the attendance status for their respective Courses.
* The teachers will be able to apply for various types of leave directly through the System.
* The students will be able to Communicate and provide feedback to their Teachers.
* The students will have access to a forum page where they are communicate will Each other.
* The administrator will be able to view and update information such asdepartments, classes, teachers, students, courses.

**6.2 User Classes and Characteristics**

There are several types of end users for the college ERP system. They are broadly divided as Students, Staff and the Administrator. Each of these classes have their own set of features.

**6.2.1 Student Module**

The student should have the following features:

* View the Attendance status of the courses to which they are enrolled.
* View the Marks of the courses to which they are enrolled.
* View the notification from the college administrator.
* Communicate or give feedback to their respective teachers.
* Communicate with other students of the same university.

**6.2.2 Staff Modules**

The staff should have the following features:

* Access to the information of all students that attend their courses.
* Add and edit the Attendance status of those students.
* Add and edit the exam marks of those students.
* Avail the different types of leave.
* Swap classes with other teachers who teach for the same class.

**6.2.3 Administrator Modules**

The administrator should have the following features:

* Add and update students, teachers and courses.
* Assign teachers and students to course

**CHAPTER 7**

**CONCLUSION AND FUTURE ENHANCEMENT**

**7.1 CONCLUSTION**

The education industry has witnessed various changes for several years like from pen and paper to phones, tablets, and laptops, for teaching and learning but there is still a big scope of change that can transform the level of education making all the operations simple, efficient, and relevant. I guess you may have understood what I am talking about? A Higher Education ERP – can transform the complete operations of the institution bringing automation and digital transformation of all the academic and administrative processes of the institution.

**7.2 FUTURE ENHANCEMENT**

The future of education is changing rapidly and will change in the upcoming years, but having an Education ERP Software you can make sure to adopt every change with the same consistency and flow and it will make all the processes easy and smooth providing access to every single thing from just one click. Let’s discuss how an ERP will help institutions to adopt any change in the education scenario easily.

**7.2.1 List to be achieved in future**

* BI-powered ERP
* Automation of complete activities
* Advances in technology – teaching and learning patterns
* Customization and personalization
* Better communication

**APPENDICES**

**Appendix 1:**

**Source code**

from tkinter import \*

from PIL import Image,ImageTk

import mysql.connector

main=Tk()

total\_height=main.winfo\_screenheight()

total\_width=main.winfo\_screenwidth()

main.title("M.I.E.T. ENGINEERING COLLEGE STUDENT MANAGEMENT SYSTEM")

main.geometry(f"{int(total\_width\*0.7)}x{int(total\_height\*0.7)}")

main.withdraw()

windowwidth=main.winfo\_width()

windowheight=main.winfo\_height()

logo = ImageTk.PhotoImage(Image.open('mietlogo.png').resize((250,150)))

loginlogo=ImageTk.PhotoImage(Image.open("login.png").resize((100,100)))

class Loginpage:

def \_\_init\_\_(self) -> None:

self.login\_window=Toplevel(main)

self.login\_window.geometry(f"{int(total\_width\*0.7)}x{int(total\_height\*0.7)}")

logoframe = Frame(self.login\_window,background="cadetblue3")

logoframe.pack(side=LEFT, fill=BOTH,expand=True)

loginframe=Frame(self.login\_window, background="cadetblue2",width=total\_width\*0.2)

loginframe.pack(side=RIGHT, fill=BOTH, expand=True)

logolabel=Label(logoframe,image=logo)

logolabel.place(relx=0.25,rely=0.4)

loginlogolabel=Label(loginframe,image=loginlogo,background="cadetblue2")

loginlogolabel.place(relx=0.5,rely=0.2,anchor=CENTER)

usernamelabel=Label(loginframe,text="Username :",background="cadetblue2",font=("Arial Rounded MT Bold",15))

usernamelabel.place(relx=0.36,rely=0.4,anchor=CENTER)

self.usernameentry=Entry(loginframe,border=2.5,font=('Arial Rounded MT',13))

self.usernameentry.place(relx=0.5,rely=0.45,anchor=CENTER,height=40,width=350)

passwordlabel=Label(loginframe,text="Password :",background="cadetblue2",font=("Arial Rounded MT Bold",15))

passwordlabel.place(relx=0.36,rely=0.54,anchor=CENTER)

self.passwordentry=Entry(loginframe,border=2.5,font=('Arial Rounded MT',13),show="•")

self.passwordentry.place(relx=0.5,rely=0.59,anchor=CENTER,height=40,width=350)

v = StringVar(loginframe, "1")

student\_radio\_button=Radiobutton(loginframe,text="Student",font=("Arial Rounded MT Bold",15),variable=v,value="1",background="cadetblue2")

student\_radio\_button.place(relx=0.4,rely=0.7,anchor=CENTER)

staff\_radio\_button=Radiobutton(loginframe,text="Staff",font=("Arial Rounded MT Bold",15),variable=v,value="2",background="cadetblue2")

staff\_radio\_button.place(relx=0.6,rely=0.7,anchor=CENTER)

login\_button=Button(loginframe,text="Login",font=("Arial Rounded MT Bold",15),background="cadetblue",width=20,command=self.show)

login\_button.place(relx=0.5,rely=0.8,anchor=CENTER)

self.login\_window.protocol("WM\_DELETE\_WINDOW", self.on\_closing)

def on\_closing(self):

try:

main.destroy()

except:

self.login\_window.destroy()

def show(self):

main.deiconify()

self.login\_window.destroy()

def checkuser(self):

pass

lp=Loginpage()

main.mainloop()

from tkinter import \*

from tkinter import ttk

from PIL import Image, ImageTk

main=Tk()

total\_height=main.winfo\_screenheight()

total\_width=main.winfo\_screenwidth()

class DoubleScrolledFrame:

def \_\_init\_\_(self, master, \*\*kwargs):

width = kwargs.pop('width', None)

height = kwargs.pop('height', None)

self.outer = Frame(master, \*\*kwargs)

self.vsb = ttk.Scrollbar(self.outer, orient=VERTICAL)

self.vsb.grid(row=0, column=1, sticky='ns')

self.hsb = ttk.Scrollbar(self.outer, orient=HORIZONTAL)

self.hsb.grid(row=1, column=0, sticky='ew')

self.canvas = Canvas(self.outer, highlightthickness=0, width=width, height=height,\*\*kwargs)

self.canvas.grid(row=0, column=0, sticky='nsew')

self.outer.rowconfigure(0, weight=1)

self.outer.columnconfigure(0, weight=1)

self.canvas['yscrollcommand'] = self.vsb.set

self.canvas['xscrollcommand'] = self.hsb.set

self.canvas.bind("<Enter>", self.\_bind\_mouse)

self.canvas.bind("<Leave>", self.\_unbind\_mouse)

self.vsb['command'] = self.canvas.yview

self.hsb['command'] = self.canvas.xview

self.inner = Frame(self.canvas)

self.canvas.create\_window(4, 4, window=self.inner, anchor='nw')

self.inner.bind("<Configure>", self.\_on\_frame\_configure)

self.outer\_attr = set(dir(Widget))

def \_\_getattr\_\_(self, item):

if item in self.outer\_attr:

return getattr(self.outer, item)

else:

return getattr(self.inner, item)

def \_on\_frame\_configure(self, event=None):

x1, y1, x2, y2 = self.canvas.bbox("all")

height = self.canvas.winfo\_height()

width = self.canvas.winfo\_width()

self.canvas.config(scrollregion = (0,0, max(x2, width), max(y2, height)))

def \_bind\_mouse(self, event=None):

self.canvas.bind\_all("<4>", self.\_on\_mousewheel)

self.canvas.bind\_all("<5>", self.\_on\_mousewheel)

self.canvas.bind\_all("<MouseWheel>", self.\_on\_mousewheel)

def \_unbind\_mouse(self, event=None):

self.canvas.unbind\_all("<4>")

self.canvas.unbind\_all("<5>")

self.canvas.unbind\_all("<MouseWheel>")

def \_on\_mousewheel(self, event):

func = self.canvas.xview\_scroll if event.state & 1 else self.canvas.yview\_scroll

if event.num == 4 or event.delta > 0:

func(-1, "units" )

elif event.num == 5 or event.delta < 0:

func(1, "units" )

def \_\_str\_\_(self):

return str(self.outer)

main.title("M.I.E.T Engineering College Student Management System")

main.geometry(f"{int(total\_width\*0.7)}x{int(total\_height\*0.7)}")

# main.state("zoomed")

style = ttk.Style(main)

style.configure('lefttab.TNotebook', tabposition='w')

profile = ImageTk.PhotoImage(Image.open('profile.png').resize((100,100)))

profiletest = ImageTk.PhotoImage(Image.open('login.png').resize((200,200)))

fees = ImageTk.PhotoImage(Image.open('fees.png').resize((100,100)))

result = ImageTk.PhotoImage(Image.open('result.png').resize((100,100)))

notebook = ttk.Notebook(main, style='lefttab.TNotebook')

f1 = DoubleScrolledFrame(notebook, width=300, borderwidth=2, relief=SUNKEN)

f2 = DoubleScrolledFrame(notebook, width=300, borderwidth=2, relief=SUNKEN, background="blue")

f3 = DoubleScrolledFrame(notebook, width=300, borderwidth=2, relief=SUNKEN, background="green")

profile\_label=Label(f1,text="PROFILE",font=("arial black",40))

profile\_label.grid(row=0,column=0,sticky="W",padx=40,pady=20)

profile\_picture=Label(f1,image=profiletest,font=("arial black",40))

profile\_picture.grid(row=0,column=4,sticky="W",padx=40,pady=20,rowspan=2)

name\_label=Label(f1,text="Name :",font=("arial",20))

name\_label.grid(row=1,column=0,sticky="W",padx=40,pady=20)

department\_label=Label(f1,text="Department :",font=("arial",20))

department\_label.grid(row=2,column=0,sticky="W",padx=40,pady=20)

dob\_label=Label(f1,text="D.O.B :",font=("arial",20))

dob\_label.grid(row=3,column=0,sticky="W",padx=40,pady=20)

father\_name\_label=Label(f1,text="Father Name :",font=("arial",20))

father\_name\_label.grid(row=4,column=0,sticky="W",padx=40,pady=20)

address\_label=Label(f1,text="Address :",font=("arial",20))

address\_label.grid(row=5,column=0,sticky="W",padx=40,pady=20)

name\_ans\_label=Label(f1,text="SHAIK MOHAMED FAHAD .T",font=("arial",20))

name\_ans\_label.grid(row=1,column=1,sticky="W",padx=40,pady=20)

department\_ans\_label=Label(f1,text="CSE",font=("arial",20))

department\_ans\_label.grid(row=2,column=1,sticky="W",padx=40,pady=20)

dob\_ans\_label=Label(f1,text="09/09/2002",font=("arial",20))

dob\_ans\_label.grid(row=3,column=1,sticky="W",padx=40,pady=20)

father\_name\_ans\_label=Label(f1,text="THAIYUB AHAMED .M",font=("arial",20))

father\_name\_ans\_label.grid(row=4,column=1,sticky="W",padx=40,pady=20)

address\_ans\_label=Label(f1,text="1/83 SEKU HOUSE STREET,\nKATTIMEDU,\nTHIRUTHURAIPOONDI TLK\nTHIRUVARUR DIST\n614716",font=("arial",20),justify=LEFT)

address\_ans\_label.grid(row=5,column=1,sticky="W",padx=40,pady=20)

notebook.add(f1, image=profile)

notebook.add(f2, image=result)

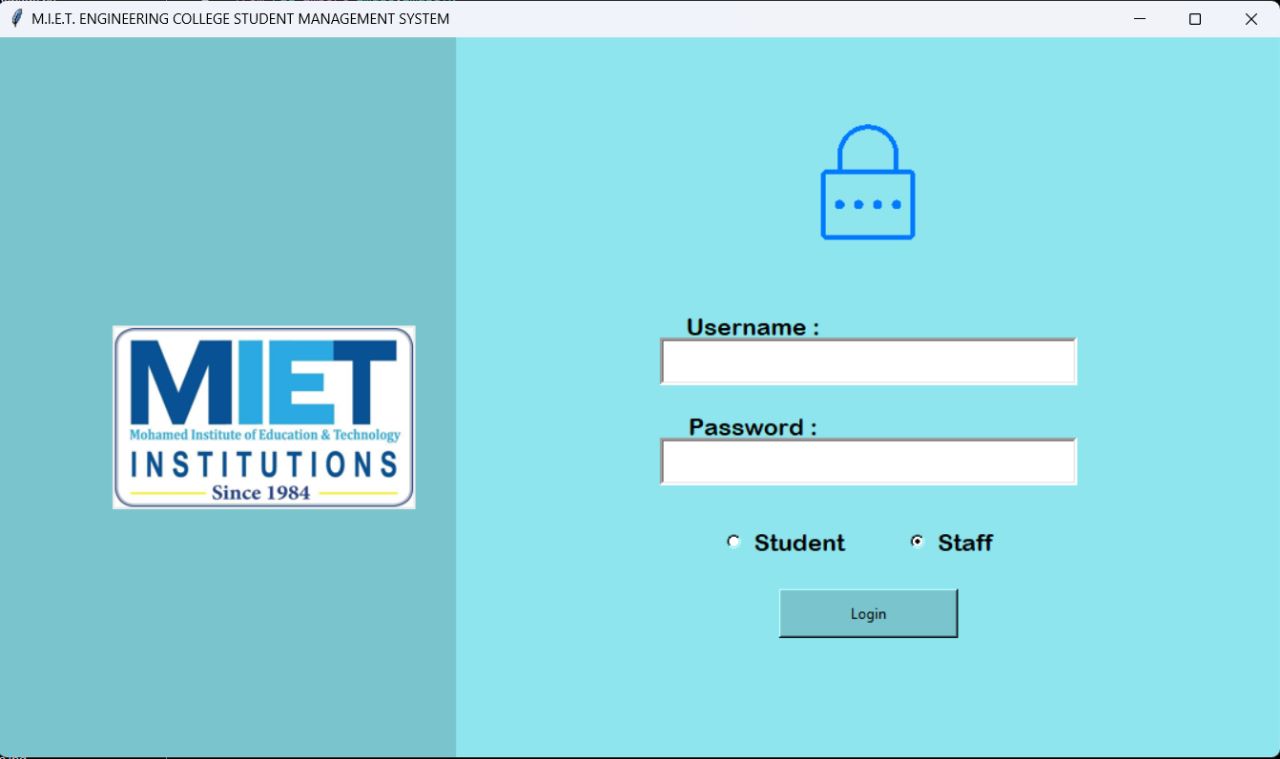
notebook.add(f3, image=fees)

notebook.pack(fill='both',expand=True)

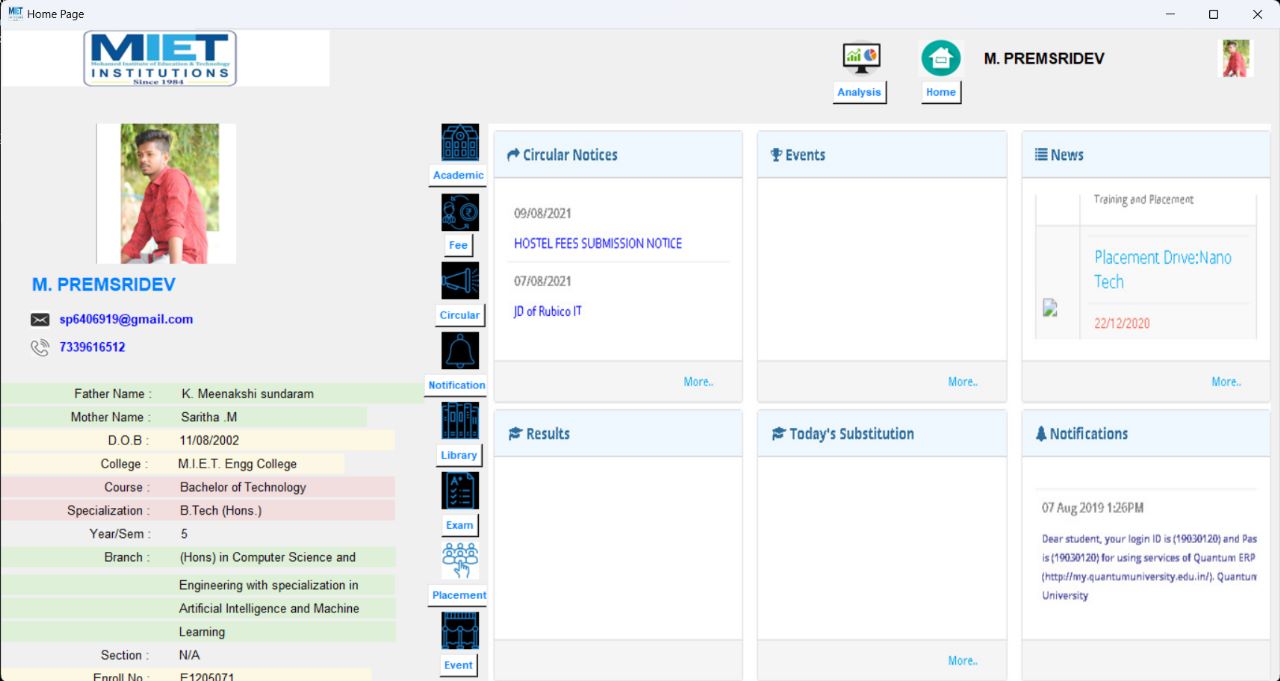
print(total\_height,total\_width)

main.mainloop()

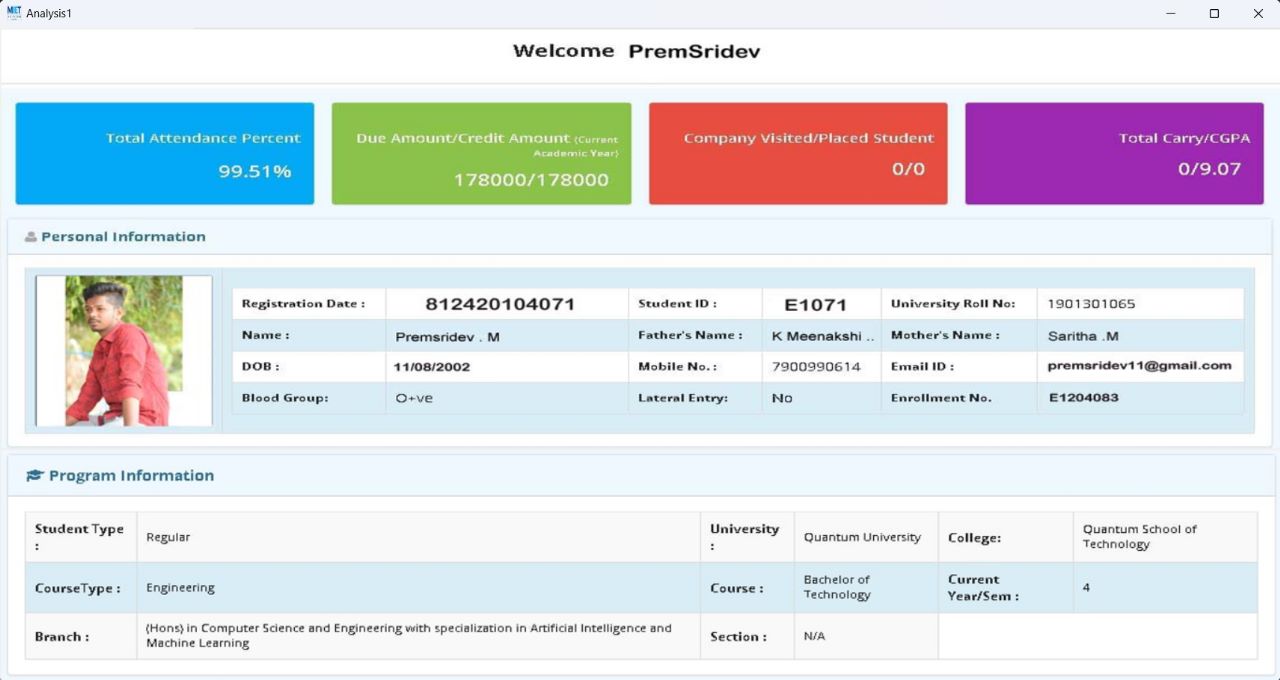
**Appendix 2:**

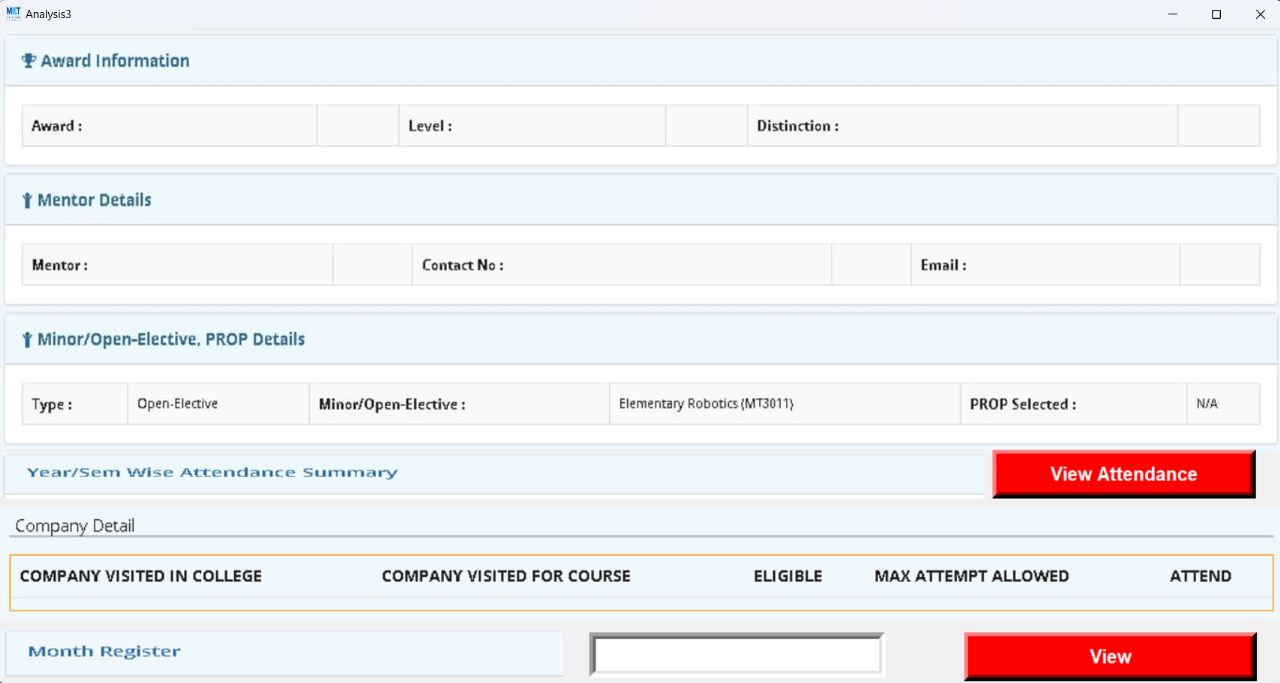
**Screenshots**

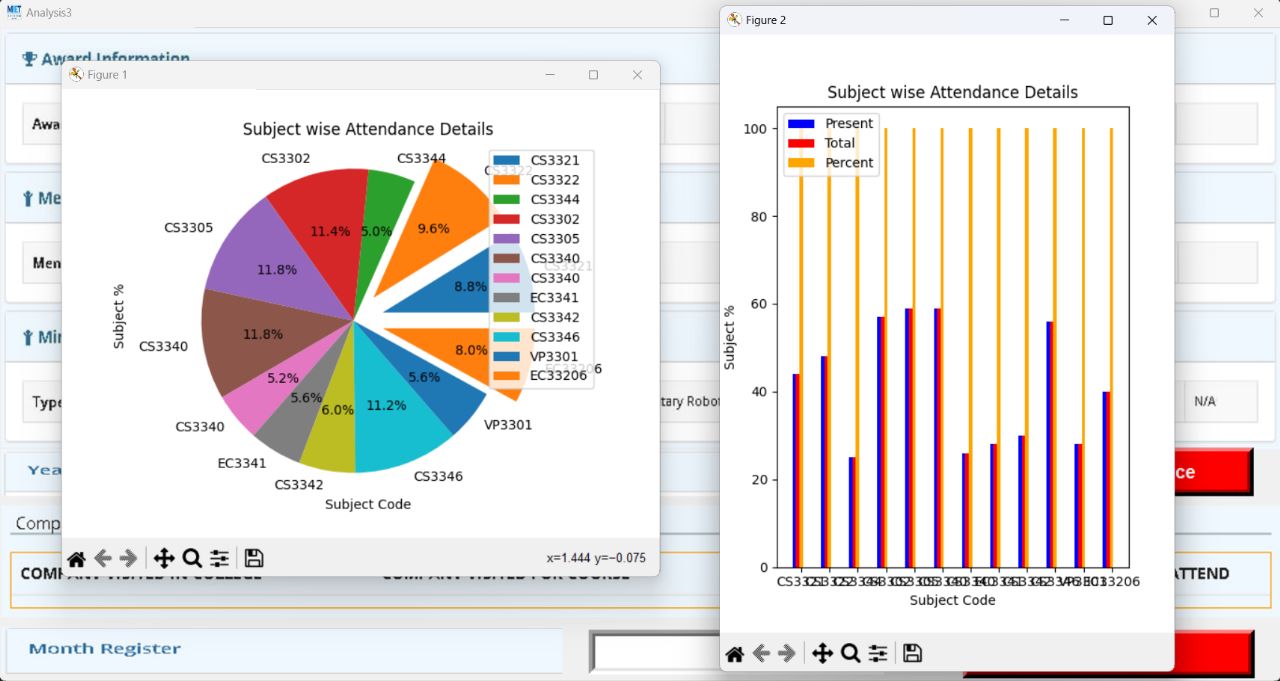
**Figure No. A. 2. 1: Login Page**



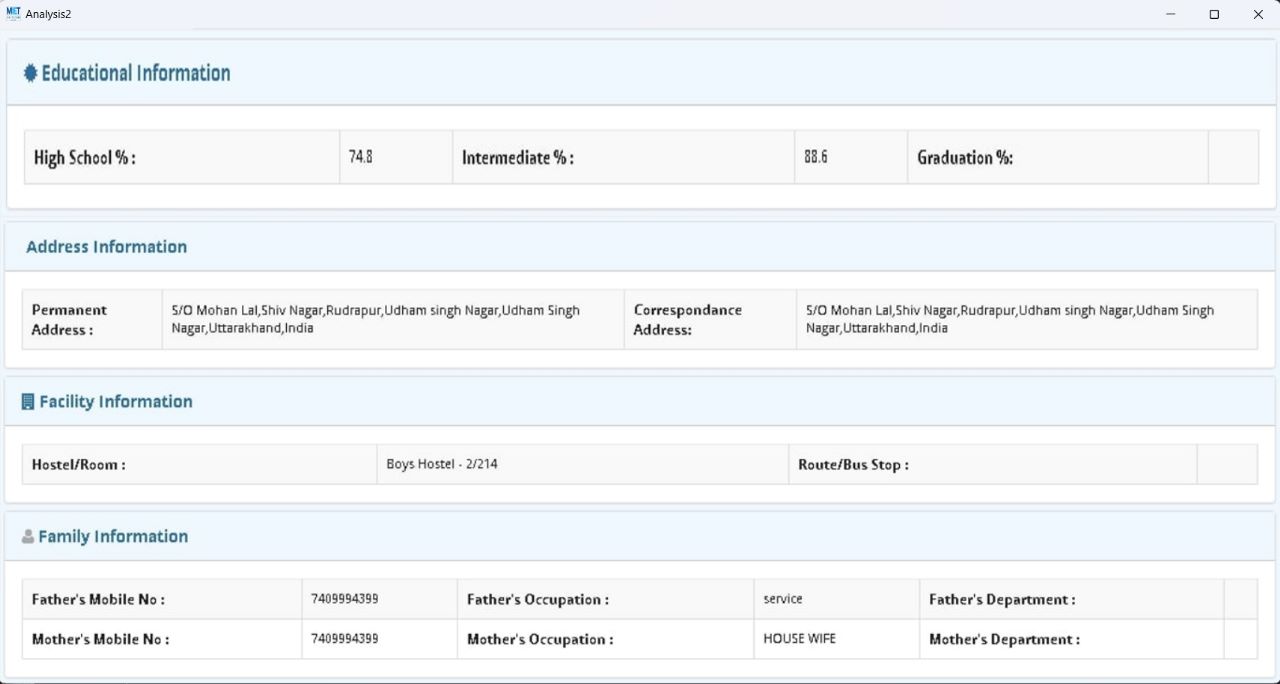
**Figure No. A. 2. 2: Home Page**

**Figure No. A. 2. 3: Student Info**

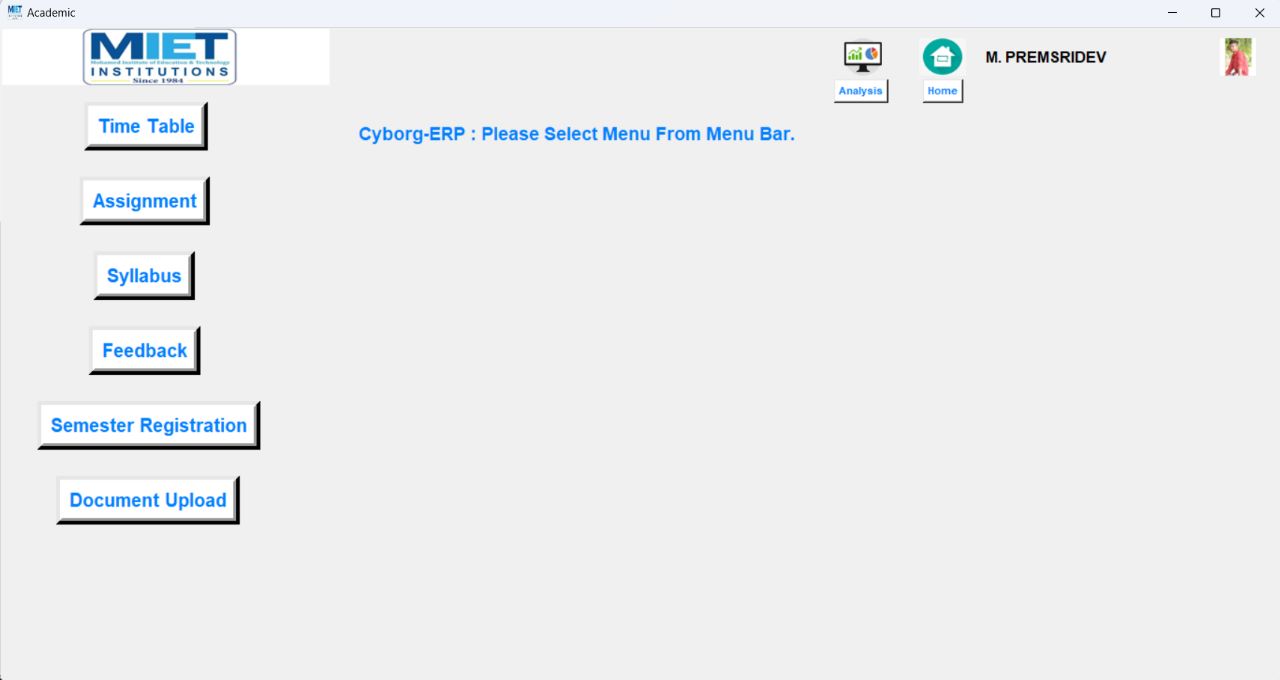
**Figure No. A. 2. 4: Attendace**

****

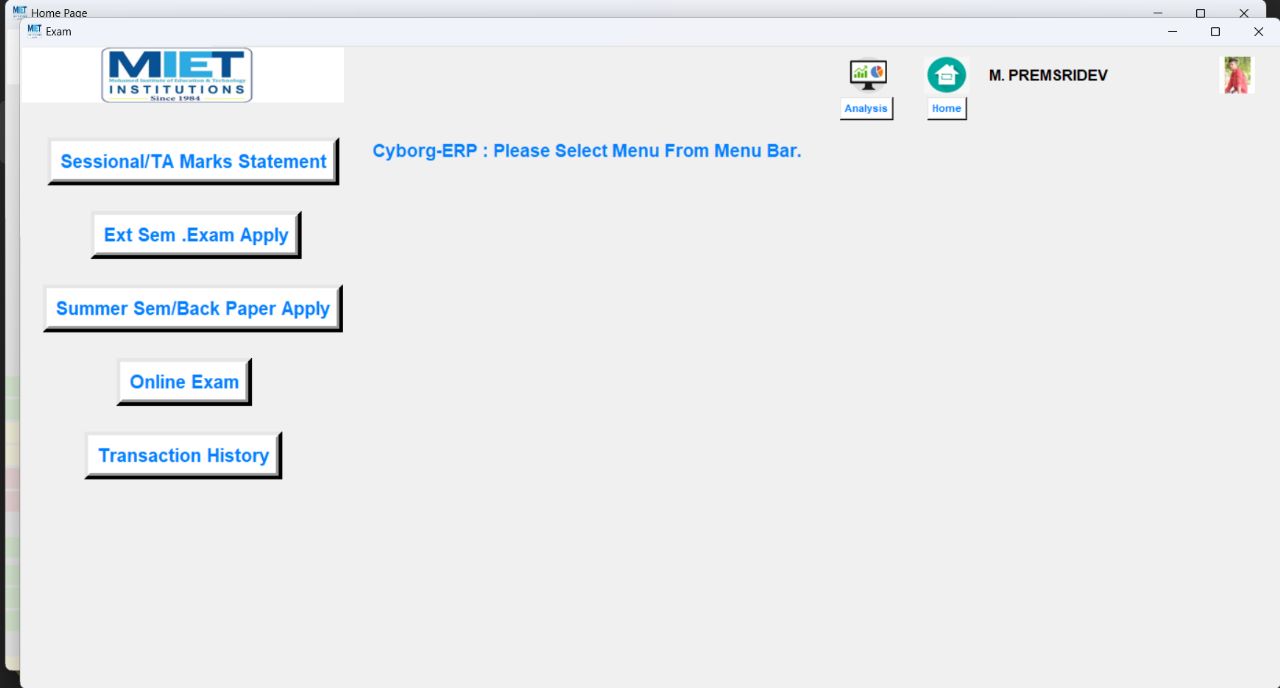
**Figure No. A. 2. 5: Subject Wise Attendance Details**

****

**Figure No. A. 2. 6: Student Past Info**

****

**Figure No. A. 2. 7: Academic Details**

****

**Figure No. A. 2. 8: Exam Details**

**Reference**

[1]. Albadri, A & Abdullah, S (2009). ERP Training and Evaluation: ERP Life-Cycle Approach to EndUsers’ Characterization and Competency Building in the Context of an Oil & Gas Company. IBIMA BUSINESS REVIEW.Vol. 3, pp.1926.

[2]. Bhatti, T (2005). Critical success factors for the implementation of enterprise resource planning (ERP): empirical validation. The Second International Conference on Innovation in Information Technology.

[3]. M. Markus, L, C. Tanis and P. Van Fenema, "Enterprise resource planning: multisite ERP implementations," Communications of ACM, vol. 43, pp. 42-46, 2000.

[4]. Y. van Everdingen, J. Hillegersberg and E. Waarts "Enterprise resource planning: ERP adoption by European midsize companies," Communications of ACM, vol. 43, pp. 27-31, 2000.

[5]. Wenjie Yang, Haoxue Liu, Jie Shi,” The Design of Printing Enterprise Resources Planning (ERP)