KLN COLLEGE OF ENGINEERING

MINI PROJECT REPORT

TITLE : E-PLACEMENT CELL

PROJECT MENTOR : SATEESH KUMAR

BY:

T.VENKATESH

T.H.VINOD

BONAFIDE CERTIFICATE

This is to certify that this project report "E-PLACEMENT CELL" is the bonafide work of **"Mr. T.VENKATESH (Reg 910619205068)" , "Mr. T.H.VINOD (Reg 910619205069)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Mr. SATHEESH KUMAR M ,	**Mr. DR. P. GANESH KUMAR ,**
B.E. (CSE), M.E. (CSE), (PH.D),	**B.E.(I&C), M.E, PH.D. ,**
ASSISTANT PROFESSOR,	**HEAD OF THE DEPARTMENT,**
INFORMATION TECHNOLOGY,	**INFORMATION TECHNOLOGY,**
K.L.N COLLEGE OF ENGINEERING,	**K.L.N COLLEGE OF ENGINEERING**
(AUTONOMOUS)	**(AUTONOMOUS)**

Submitted for the practical Examination held on

INTERNAL SIGN	**EXTERNAL SIGN**

ACKNOWLEDGEMENT

Any work would be unfullied without a word of thanks. We hereby take pleasure in acknowledging the persons who guided us throughout our work.

First and foremost, thanks to Almighty for providing us all his abundant blessings throughout the work. We all extend our heartfelt thanks to DR. K.N.K GANESH Ph.D(Hons), Secretary and Correspondent of our college and Dr. A.V. RAMPRASAD, M.E, Ph.D, Principal of our college for providing us all the provisions.

We esteem ourselves to articulate our sincere thanks to **Mr. DR. P. GANESH KUMAR,** Head of the department of Information Technology for leading us towards the zenith of success.

We express our grateful thanks to our Project Guide **Mr. SATHEESH KUMAR M, Assistant Professor** for their invaluable guidance and motivation. Their assistance and advice had been very helpful throughout our project. I would like to thank all teaching and non-teaching staff of our department who have been the sources of encouragement and ideas. I thank them for lending their support whenever needed.

ABSTRACT

Maintaining and managing the Student details and classifying it for the placement's is a very restless task and doing this manually would cause many human errors. The project we developed is a software that can automate this process of maintaining and managing the student details for the placements.

Here there are three types of login, one for students, another for placement coordinator and last one for TPO. Students can enter non spoof able information like date of birth, Aadhar details, Resume and more. Placement Coordinator has the task to update the student spoofable details like 10th mark, 12th mark, CGPA, etc. TPO can manage all the students and placement coordinator's and even provide login for that.

We have email alert for the drives updated from the TPO and TPO and placement coordinator can filter and classify the students and download their details as a CSV, They can even send mails via the software to the HR.

TABLE OF CONTENT

CHAPTER	TITLE	PAGE NO
	ACKNOWLEDGEMENT	**2**
	ABSTRACT	**3**
	LIST OF FIGURES	**5**
1	**INTRODUCTION** 1.1 Introduction 1.2 Problem statement 1.3 Project objective 1.4 Scope of the project 1.4.1 Existing system 1.4.2 Proposed system	**6**
2	**SOFTWARE LIFE CYCLE MODELS** 2.1 software life cycle model 2.1.1 Requirement gathering and analysis 2.1.2 Design 2.1.3 Implementation / coding 2.1.4 Testing 2.1.5 Deployment 2.1.6 Maintenance 2.2 Types 2.3 Incremental model 2.4 Reason for choosing the model	**8**
3	**LITERATURE REVIEW**	**11**
4	**MODULE DESCRIPTION** 4.1 Module description 4.2 Login dashboard module 4.3 Placement coordinator dashboard module 4.4 TPO dashboard module 4.5 Hardware requirements 4.6 Software requirements	**12**
5	**IMPLEMENTATION**	**17**
6	**OUTPUT**	**27**
7	**CONCLUSION**	**34**
8	**FUTURE SCOPE** 8.1 Future scope 8.2 Limitation of the project	**35**
9	**REFERENCE / BIBLIOGRAPHY**	**36**

LIST OF FIGURE

Figure No.	Figure Name	Page No.
2.4.1	Illustration of incremental model	11
4.1	Login module	14
4.2	Student module	15
4.3	Placement Coordinator	16
4.4	TPO module	16

CHAPTER 1

1.1 INTRODUCTION

Managing and maintaining students record for placement it not an easy job and doing that manually will cause many errors, this can even lead to failure of a student from attending the drive. The data’s are a huge set of volumes that must the processed computerized only, So we brought a Software that can the solution of this problem.

The software we created for this project can simplify the process of preserving and managing student information for placements.

There are three sorts of logins available here: one for students, one for placement coordinators, and one for TPOs. Students can add non-spoofable information such as their date of birth, Aadhar details, resume, and other personal information. The placement coordinator's job is to keep track of the student's spoofable information, such as 10th and 12th grades, CGPA, and so on. TPO can keep track of all of the students and placement coordinators, as well as give login credentials.

We have email alerts for TPO-updated drives, and TPOs and placement coordinators may filter and classify students, as well as download their information as a CSV file.

1.2 PROBLEM STATEMENT

In existing system all the data’s are collected using any forms to fill up or taking data from the forms already filled by the student. It will consume a lot of time and work and also cause many human errors.

Sending the eligible students to the respective company HR is also an heavy task because all the data’s are not computerized.

Let us say that a student has got 70% in his 10th and 6.99 cgpa in college but the company wants 70% throughout, while doing this in the manual method there is a possibility to leave this student behind because the company will accept this the near marks also therefore this student is eligible to attend the drive but his name is rejected while doing a manual process.

The other problem is that we need to submit reason letter’s for every changes we need, eg there is a change in my resume, or I want my marks to be changed, maintaining these letters are not easy and it is not handy too. So we provide a completely separate Database to store the files.

1.3 PROJECT OBJECTIVE

The main objective of the project is to automate the process of all the placement related works. This project is developed in user's state of mind, for a TPO he needs to manage all the students data and their documents also and post drive details also. The drive details will also have an email notification also.

Nextly the main component of this project is that the student's can only enter the non spoofable information and the placement coordinator the spoofable information like 10th and 12th marks. So that spoofing is prevented.

Then the TPO can filter students and make a report on them and can download it as a CSV or XLS file.

1.4 SCOPE OF THE PROJECT

1.4.1 Existing System

In the current approach, all data is acquired via filling out forms or extracting data from forms previously filled out by students. It will take a lot of time and effort, and it will also result in a lot of human blunders.

Managing and preserving a student's record for placement is a difficult task, and doing so by hand will result in several errors, which may result in a student's inability to attend the drive. Because all of the data is not electronic, sending the qualified students to the appropriate firm HR is also a difficult process.

Another issue is that we must submit justification letters for each modification we require, and keeping track of these letters is difficult.

1.4.2 Proposed System

The software we developed for this project can make maintaining and managing student information for placements much easier.

The project's major goal is to automate the process of all placement-related tasks. This project was created with the user in mind; as a TPO, he must manage all of the students' data, documents, and post drive information. There will be an email message with the drive details as well.

The major feature of this project is that students may only submit non-spoofable information, while the placement coordinator can only enter spoofable data such as 10th and 12th grades. So that spoofing isn't possible.

The TPO may then filter students and generate a report for them, which can be downloaded as a CSV file.

CHAPTER 2

2.1 SOFTWARE LIFE CYCLE MODEL

The period of time that start when a software is conceived and end when the product no longer available for use. The software Life cycle typically includes:

- ✓ Requirement gathering and analysis
- ✓ Design
- ✓ Implementation / coding
- ✓ Testing
- ✓ Deployment
- ✓ Maintenance

2.1.1 Requirement gathering and analysis

This phase gathers all of the business requirements. The project managers and stakeholders are primarily concerned with this phase. Following the collection of requirements, the validity of the requirements is evaluated, as well as the viability of implementing the requirements into the system under development.

2.1.2 Design

The system and software design is developed in this phase based on the requirements specifications that were researched in the first phase. System design aids in designing overall system architecture as well as describing hardware and system requirements. The testers create the Test strategy in this phase, which includes what to test and how to test.

2.1.3 Implementation / Coding

When the system design papers arrive, the work is separated into modules and the real coding begins. Because code is developed during this phase, it is the developers' primary emphasis.

2.1.4 Testing

Following the development of the code, it is tested against the requirement to ensure that the product is truly meeting the needs identified during the requirement phase.

2.1.5 Deployment

The product is provided / deployed to the customer for usage after successful testing. The beta testing will begin as soon as the product is delivered to the client. They will notify the engineering team if any adjustments are necessary or if any issues are discovered. The final deployment is completed after those adjustments are performed or the issues are addressed.

2.1.6 Maintenance

Once the customer starts using the developed software the actual problems comes up and that are needed to be solved from time to time. The process where the care of the system is take after deployment it is called as maintenance

2.2 TYPES

The common life cycles are

- Waterfall model
- Incremental model
- Spiral model
- Big bang model
- Prototyping model
- RAD model
- Agile model

2.3 INCREMENTAL MODEL

It's a step forward from the waterfall model. As a series of incremental builds, the product is conceived, implemented, integrated, and tested. Many commercial software businesses and system suppliers employ this approach of programme evolution. The incremental software development methodology may be appropriate for projects with well-defined software requirements, although implementation may be delayed if fundamental software functionality is required early on.

2.4 REASON FOR CHOOSING THE MODEL

- More flexible-less costly to change scope and requirements.
- Easy to test and debug during smaller iteration
- Easy to manage risk
- Phases are processed and completed one at a time
- Process and results are well documented

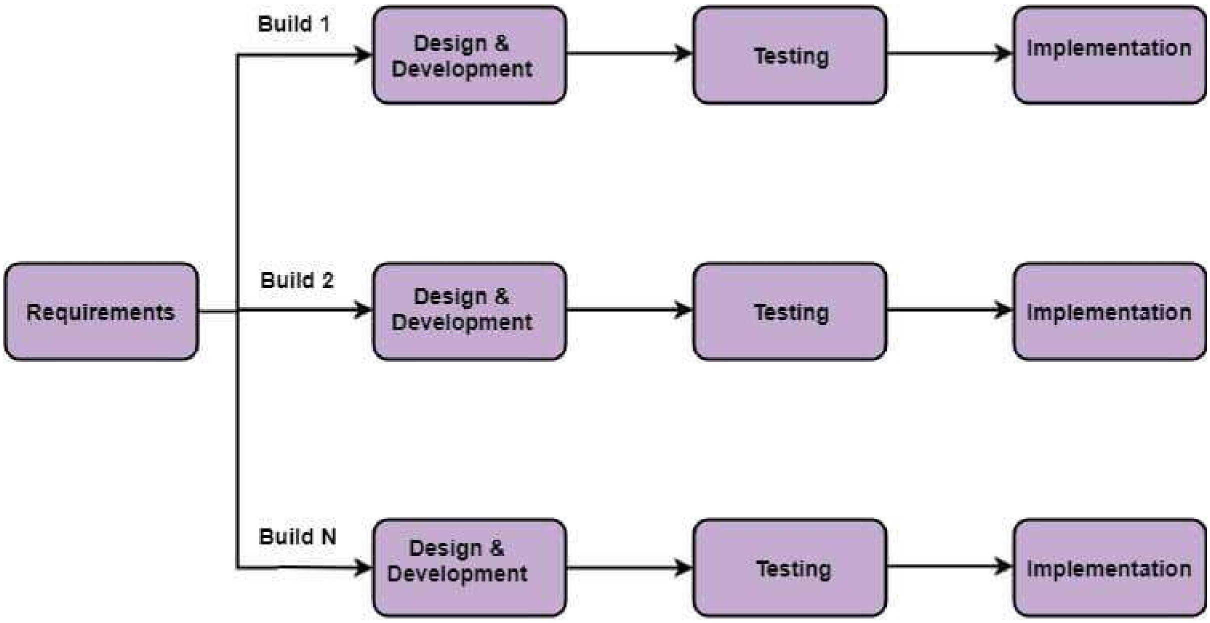


Fig 2.4.1 illustration of incremental model

CHAPTER 3

LITERATURE SURVEY

1. TnP Vision: Automation and Analysis of Campus Placements in Colleges

This paper depicts an ERP system for the Training and Placement Cell on the campuses of educational institutes. TnP Vision is a planned technology that will automate the entire placement process. It's an interactive software platform that focuses on student data management and analysis in order to digitise practises, provide visibility into students progress, and give employers a platform to streamline the hiring process. It gives information about placement activities to all participants. Aside from existing methods, the main focus is on tracking students achievement and forecasting areas for development. It's built on top of the Django framework and employs Python-based data analytics tools to look through the candidate's data. To scrape job postings from Linkedin, web scraping is inherited.

2. "Design Paper on Online Training and Placement System (OTAP)"

Online registration, system security, automated percentage computation, data sorting, and notification services are just a few of the many features offered by the system. However, it is missing the functionality of the Event Management System, as well as the ability to generate reports.

3. "Data mining paper for placement & training system"

The author's study on data mining explains how to handle student data more easily. Because the student and administrator interfaces are so close, the system takes a long time to load. Early in the year, students produced and submitted their curriculum vitae, thus freezing them in time. For each company, lists were created, and students were required to visit the notice board on a regular basis. The process was slowing down, and important academic time was being diverted away from more productive activities. At ACE, the records were saved in a customised, Excel sheet, so sorting was a difficulty.

4. " Training & Placement System"

The authors gives the basic idea of the existing system, all processes are handled manually. the administrator should refer all the records kept for years ago to simply known number of users increases. there are many limitations for the existing systems. in manual training and placement, all the work done at ACE is by human intervention due to which there were maximum chances of errors.

MODULE DESCRIPTION

In this project we have multiple modules with multiple features, each module has its own specific feature and set of operations too.

The modules are :

- ✓ **Login module**
  - ➢ Exception handling
- ✓ **Student dashboard module**
  - ➢ Personal details
  - ➢ Drive alert
  - ➢ Document upload
  - ➢ Change credentials
- ✓ **Placement Coordinator module**
  - ➢ Personal details
  - ➢ Students details update
  - ➢ Filtering
  - ➢ Send mail
- ✓ **TPO module**
  - ➢ Personal details
  - ➢ Filtering students
  - ➢ Manage login credentials
  - ➢ Send mails

LOGIN MODULE

All sorts of users, including students, class coordinators, and placement coordinators, can log in to the login module. The login information is checked against the database. We won't give them the remember me option since this programme includes a lot of important information, therefore we won't give them that choice to avoid physical intrusions.

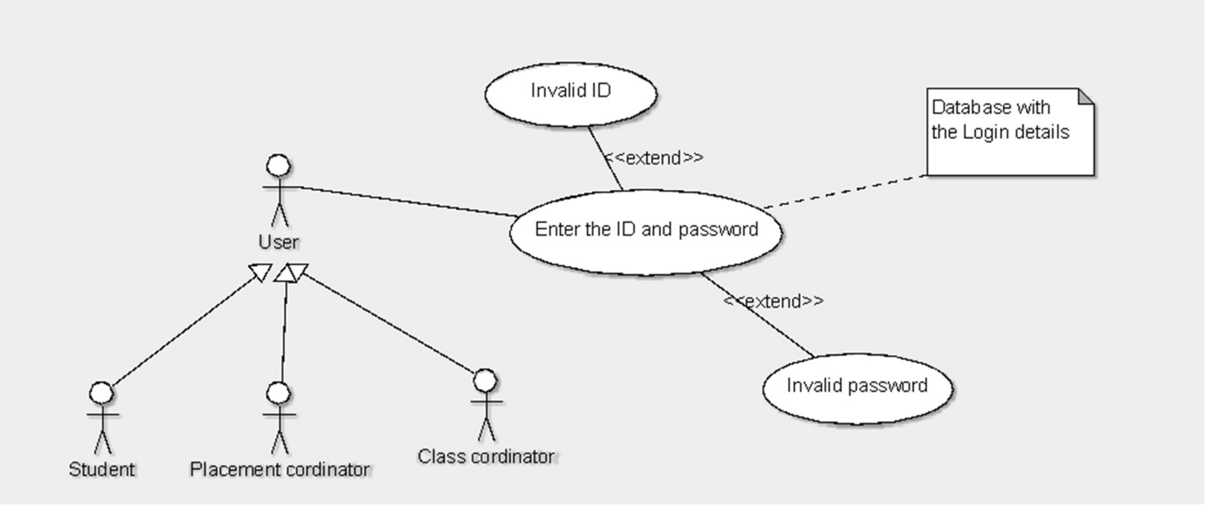


Fig 4.1 Login Module

STUDENT DASHBOARD MODULE

Students' personal information, such as name, date of birth, email, roll number, register number, address, and profile picture, are displayed in this section. There is a "upload document" button where students may submit crucial documents like as their Aadhar card, passport, 10th and 12th grade marksheets, and more, as well as a change credential and driving alert. He'll learn about the facts of the drive and find out if he meets the eligibility requirements in drive alert.

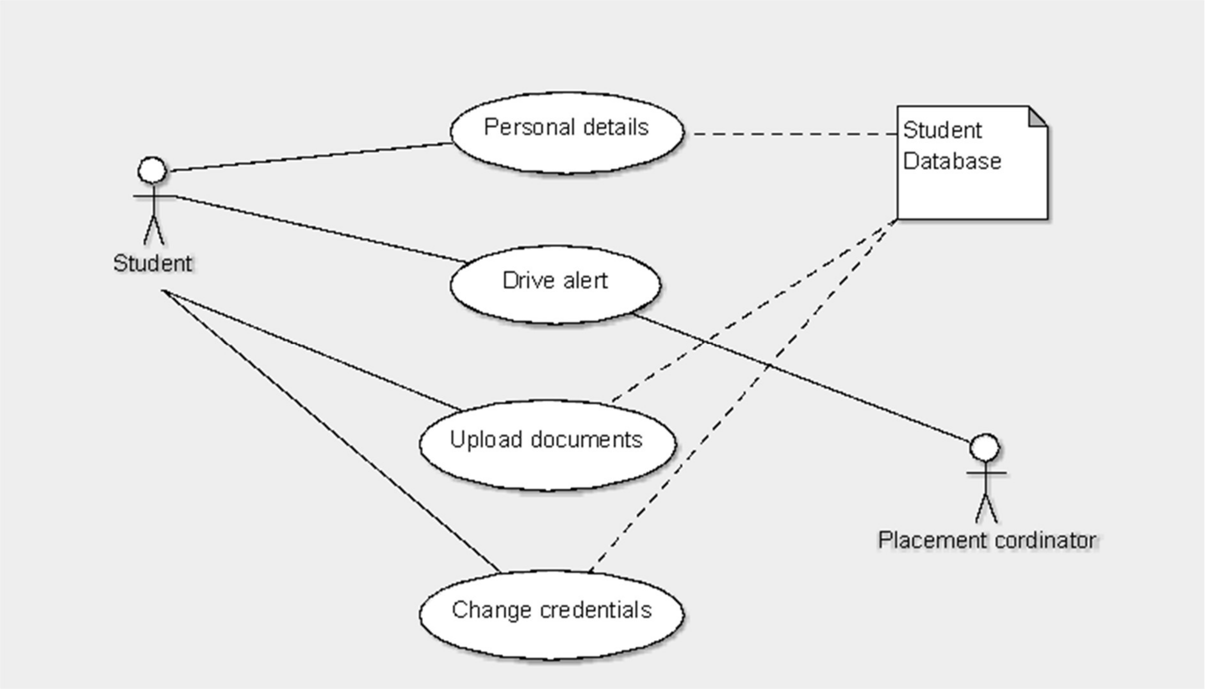


Fig 4.2 Student Module

PLACEMENT COORDINATOR DASHBOARD MODULE

It contains their personal information, including a photo, and then it has student information, including the ability to fill in their grades and check all of the information, as well as export it as a csv or pdf document. Depending on the class coordinator's mentality, the students might be grouped together or given a full report on each individual student.

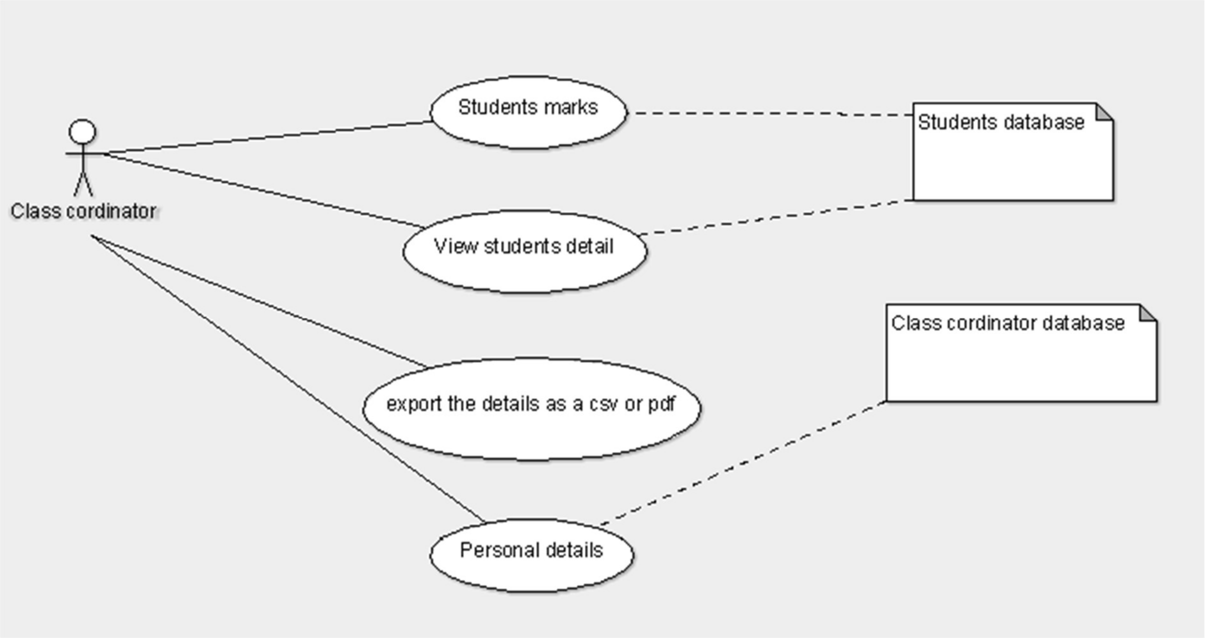


Fig 4.3 Placement Coordinator

TPO DASHBOARD MODULE

Placement coordinator or TPO plays an major part in this system. They can edit the details of a student and even delete an record of the student with proper reason stored in the database for the future reference. He can post the drive details and description of the drive also. He can send mails to HR with the eligible students list.

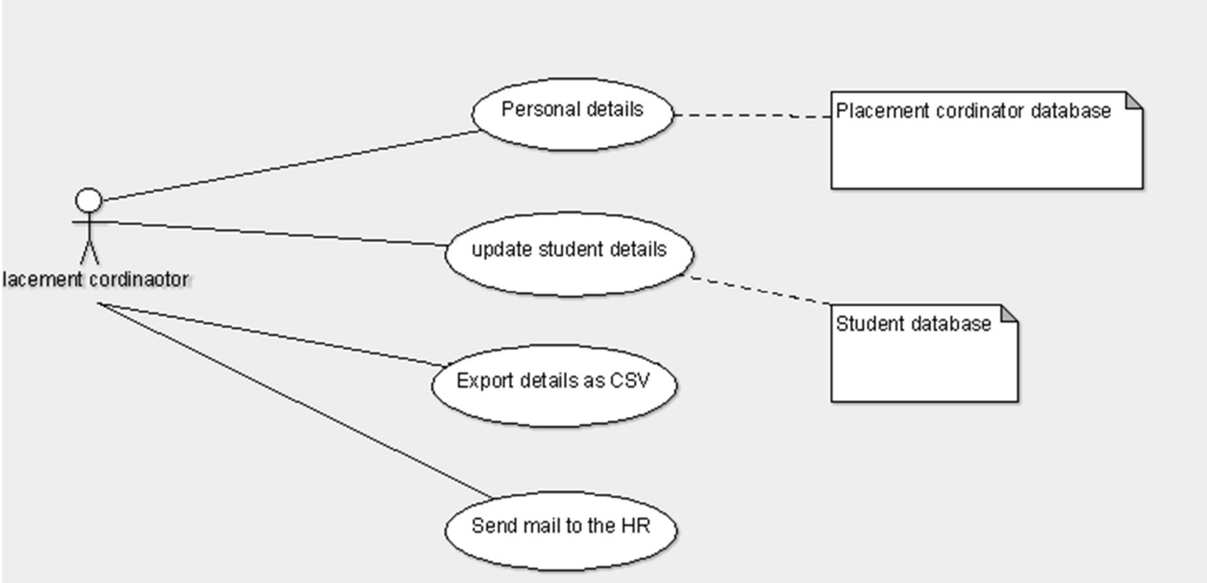


Fig 4.4 TPO Module

HARDWARE REQUIREMENTS

Processor – x64 based process

Speed – 2.10 GHz

RAM – 4Gb

Hard Disk – minimum 15gb

SOFTWARE REQUIREMENTS

OS – Windows

Language – Python

Framework – Tkinter

Database – Sqlite3

Image processing – PILLOW

Other modules – SYS, TIME, DATE, CSV, PyPDF2

CHAPTER 5

IMPLEMENTATION

login.py

```
from tkinter import *
from tkinter import messagebox as m
import sqlite3 as sq

from Studentdash import *

conn=sq.connect("DB/pdb.sqlite")
c=conn.cursor()


root=Tk()
root.geometry("1200x850+242+47")
bgc="#0085fa"
fgc="white"
root.resizable(False,False)
root.title("Signup page")
root.config(bg=bgc)

def getdata(a,b=None):
    if b == None:
        c.execute(f"SELECT * FROM {a} WHERE rollnum = {idev}")
        d=c.fetchall()
        # print(d)
        # print(d[0])
        # print(d[0][0])
        # print(d[0][1])
        if d==[]:
            return 0
    else:
        c.execute(f"SELECT * FROM {a} WHERE id = {idev}")
        d=c.fetchall()
        # print(d)
        # print(a)
        if d==[]:
            c.execute(f"SELECT * FROM {b} WHERE id = {idev}")
            d=c.fetchall()
            # print(d)
            # print(b)
            if d==[]:
                return 0
    if d[0][0]==int(idev) and d[0][1]==psev:
        return d
    else:
        return 0
def login_check():
    global idev
    global psev
    idev=ide.get()
    psev=pse.get()
    if(idev=="" and psev==""):
            m.showerror("Error","Enter both ID and Password")
    else:
        try:
            t=int(idev)
            # idev=str(idev)
        except:
            m.showerror("Error","Id must be an number")

        if len(idev)>=6:
            if (len(idev)>6):
                t=getdata("LPC","LF")
                if(t):
                    m.showinfo("Login sucessful","Click okay to view your dashboard")


                else:
                    m.showinfo("Alert","Your login details are not in the database check your 
credentials or contact the developer")
                    print("1")

            elif len(idev)==6:
                t=getdata("LST")
                if(t):
                    m.showinfo("Login sucessful","Welcome student click okay to continue")
                    root.destroy()
                    f()
                else:
                    print("2")
                    m.showinfo("Alert","Your login details are not in the database check your 
credentials or contact your adminsters")
        else:
            m.showwarning("Warning","Enter proper details")


wl=Label(root,fg=fgc,text="Welcome to E-Placement cell",font=("",35),bg=bgc)
```

```
wl.place(x=350,y=20)

idl=Label(root,fg=fgc,bg=bgc,text="ID",font=("Arial",20))
idl.place(x=400,y=250)

ide=Entry(root,width=25,font=("Arial",20))
ide.place(x=405,y=300)

pl=Label(root,fg=fgc,bg=bgc,text="Password",font=("Arial",20))
pl.place(x=400,y=350)

pse=Entry(root,show="*",width=25,font=("Arial",20))
pse.place(x=405,y=400)

fpb=Button(root,text="Forgot password",font=("Arial",15),bd=0,fg=fgc,bg=bgc)
fpb.place(x=402,y=450)

lb=Button(root,text="Login",font=("Arial",15),bd=0,fg=fgc,bg=bgc,command=login_check)
lb.place(x=725,y=450)

root.mainloop()
```

studentdash.py

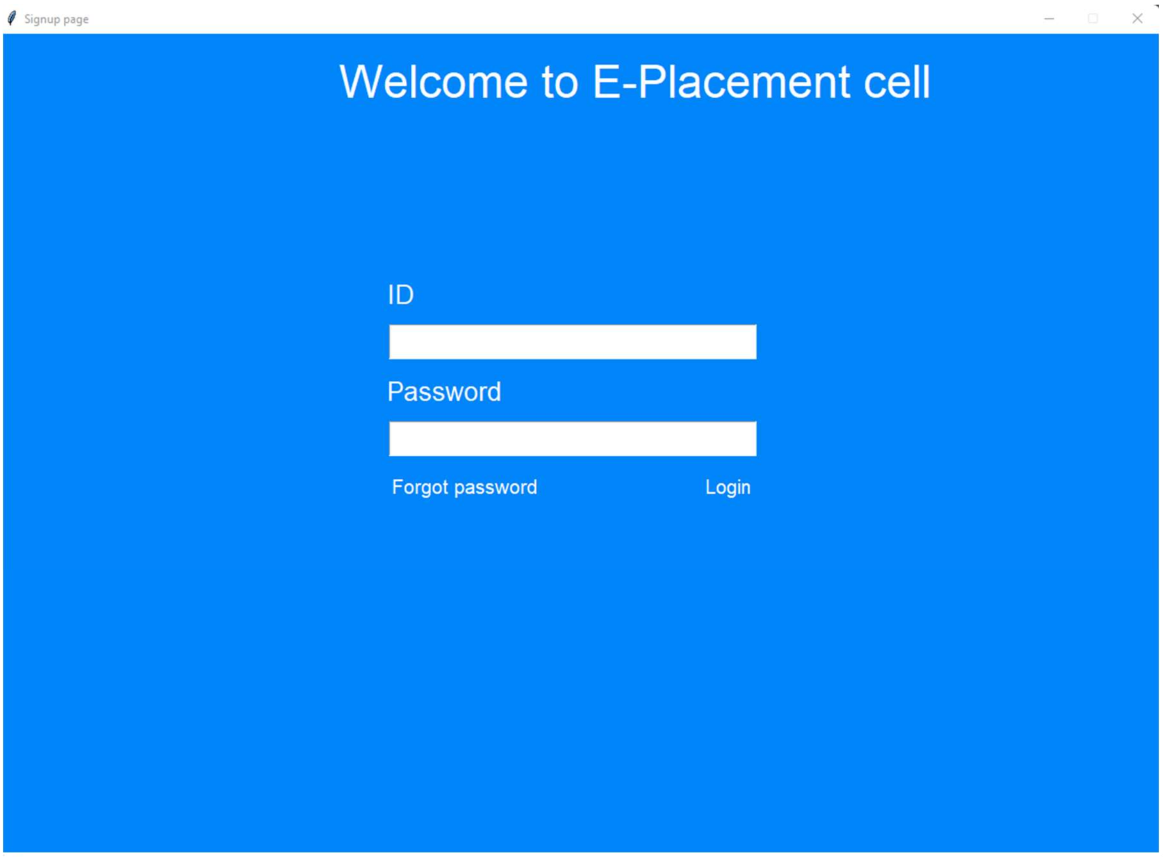
```
from tkinter import *
from tkinter import messagebox as m
import sqlite3 as sq
from tkinter import ttk
from PIL import ImageTk, Image
from tkinter import filedialog

conn=sq.connect("DB/pdb.sqlite")
c=conn.cursor()
fgc="#0085fa"
bgc="white"

def mailst():
    m.showinfo("Mail sent","Mail sent succesfully and your credentials are changed, If you
face any problem contact your administrator")

def changeCredential():
    ccf=Frame(root,width=955,height=845,bg=fgc)
    ccf.place(x=240,y=0)

    eidl=Label(ccf,text="Enter your Roll Number",font=(("Arial"),18),bg=fgc,fg=bgc)
    eidl.place(x=70,y=30)
    eide=Entry(ccf,width=25,font=(("Arial"),15),bd=0)
    eide.place(x=370,y=35)

    erel=Label(ccf,text="Enter your Register Number",font=(("Arial"),18),bg=fgc,fg=bgc)
    erel.place(x=20,y=90)

    eree=Entry(ccf,width=25,font=(("Arial"),15),bd=0)
    eree.place(x=370,y=95)

    eopl=Label(ccf,text="Enter your Old password",font=(("Arial"),18),bg=fgc,fg=bgc)
    eopl.place(x=50,y=150)

    eope=Entry(ccf,width=25,font=(("Arial"),15),bd=0)
    eope.place(x=370,y=155)

    enpl=Label(ccf,text="Enter your New password",font=(("Arial"),18),bg=fgc,fg=bgc)
    enpl.place(x=45,y=210)

    enpe=Entry(ccf,width=25,font=(("Arial"),15),bd=0)
    enpe.place(x=370,y=215)

    ernpl=Label(ccf,text="Retype the New password",font=(("Arial"),18),bg=fgc,fg=bgc)
    ernpl.place(x=35,y=270)

    ernpe=Entry(ccf,width=25,font=(("Arial"),15),bd=0)
    ernpe.place(x=370,y=275)

    ercl=Label(ccf,text="The reason to change",font=(("Arial"),18),bg=fgc,fg=bgc)
    ercl.place(x=80,y=330)

    erce=Entry(ccf,width=25,font=(("Arial"),15),bd=0)
    erce.place(x=370,y=335)

    ccb=Button(ccf,width=25,text="Commit
changes",font=(("Arial"),15),bd=2,command=mailst)
    ccb.place(x=370,y=390)


def uploadfilepath(i,m,n):
    f=["aadhar","tms","twms","diploma","pan","collegemarksht"]
    a =  filedialog.askopenfilename(initialdir = "/",title = "Select file",filetypes = (("pdf
files","*.pdf"),("all files","*.*")))
    statusc="Uploaded" if a else "NIL"
    print(a)
    print(statusc)
    ads=Label(udf,text=statusc,font=(("Arial"),18),bg=fgc,fg=bgc)
    ads.place(x=m,y=n)
    # c.execute(f"UPDATE SUD SET {f[i]}={a} WHERE rollnum={195007}")
    # c.commit()
    # d=c.execute(f"SELECT {f[i]} FROM SUD WHERE rollnum={195007}")
    # print(d.fetchall())


def uploaddocs():
    statusc="NIL" if 1 else "Uploaded"
    global udf

    udf=Frame(root,width=955,height=845,bg=fgc)
    udf.place(x=240,y=0)

    adl=Label(udf,text="Adhaar card upload",font=(("Arial"),18),bg=fgc,fg=bgc)
    adl.place(x=20,y=30)


adb=Button(udf,text="Upload",width=18,font=(("Arial"),15),bg=bgc,fg=fgc,command=lamb
da: uploadfilepath(0,600,30))
    adb.place(x=350,y=30)

    tml=Label(udf,text="10th marksheet upload",font=(("Arial"),18),bg=fgc,fg=bgc)
    tml.place(x=20,y=90)


tmb=Button(udf,text="Upload",width=18,font=(("Arial"),15),bg=bgc,fg=fgc,command=lamb
da: uploadfilepath(1,600,90))
    tmb.place(x=350,y=90)

    twml=Label(udf,text="12th marksheet upload",font=(("Arial"),18),bg=fgc,fg=bgc)
    twml.place(x=20,y=150)


twmu=Button(udf,text="Upload",width=18,font=(("Arial"),15),bg=bgc,fg=fgc,command=la
mbda: uploadfilepath(2,600,150))
    twmu.place(x=350,y=150)

    dml=Label(udf,text="Diploma marksheet upload",font=(("Arial"),18),bg=fgc,fg=bgc)
    dml.place(x=20,y=210)


dmb=Button(udf,text="Upload",width=18,font=(("Arial"),15),bg=bgc,fg=fgc,command=lam
bda: uploadfilepath(3,600,210))
    dmb.place(x=350,y=210)
    dll=Label(udf,text="Driving license upload",font=(("Arial"),18),bg=fgc,fg=bgc)
    dll.place(x=20,y=270)


dlb=Button(udf,text="Upload",width=18,font=(("Arial"),15),bg=bgc,fg=fgc,command=lamb
da: uploadfilepath(4,600,270))
    dlb.place(x=350,y=270)

    pcl=Label(udf,text="Pan card upload",font=(("Arial"),18),bg=fgc,fg=bgc)
    pcl.place(x=20,y=330)


pcu=Button(udf,text="Upload",width=18,font=(("Arial"),15),bg=bgc,fg=fgc,command=lamb
da: uploadfilepath(5,600,330))
    pcu.place(x=350,y=330)

    cml=Label(udf,text="College marksheet upload\n[upto current
semester]",font=(("Arial"),18),bg=fgc,fg=bgc)
    cml.place(x=20,y=390)


cmb=Button(udf,text="Upload",width=18,font=(("Arial"),15),bg=bgc,fg=fgc,command=lam
bda: uploadfilepath(6,600,390))
    cmb.place(x=350,y=390)

def temps():
    a =  filedialog.askopenfilename(initialdir = "/",title = "Select file",filetypes = (("pdf
files","*.pdf"),("all files","*.*")))
def temps1():
    a =  filedialog.askopenfilename(initialdir = "/",title = "Select file",filetypes = (("jpg
files","*.jpg"),("all files","*.*")))

def personal():
    global ui
    pf=Frame(root,width=955,height=845,bg=fgc)
    pf.place(x=240,y=0)
    def btd(data, file_name):
        # global pf
        global ui
        with open(file_name, 'wb') as file:
            file.write(data)
        ui = Image.open(file_name)
        ui=ui.resize((200,200))
        ui=ImageTk.PhotoImage(ui)
        label = Label(pf, image = ui)
        label.place(x=10,y=10)
    # pf=Frame(root,width=955,height=845,bg=fgc)
    # pf.place(x=240,y=0)

    c.execute(f"SELECT * FROM SPD WHERE rollnum = {195007}")
    d=c.fetchall()
    d=d.pop()
    c.execute(f"SELECT * FROM SP WHERE rollnum = {195007}")
    p=c.fetchall()
    p=p.pop()
    btd(p[1],"Images/temp.jpg")

    cpb=Button(pf,text="Change
pic",font=(("Arial"),15),fg=fgc,bg=bgc,width=18,command=temps1)
    cpb.place(x=10,y=230)

    fnl=Label(pf,text="First name",font=(("Arial"),15),bg=fgc,fg=bgc)
    fnl.place(x=290,y=40)

    fnb=Entry(pf,width=45,font=(("Arial"),15),bd=0)
    fnb.place(x=410,y=40)
    fnb.insert(INSERT,d[2])
    fnb.config(state="disabled")

    lnl=Label(pf,text="Last name",font=(("Arial"),15),bg=fgc,fg=bgc)
    lnl.place(x=290,y=120)

    lnb=Entry(pf,width=45,font=(("Arial"),15),bd=0)
    lnb.place(x=410,y=120)
    lnb.insert(INSERT,d[3])
    lnb.config(state="disabled")

    enl=Label(pf,text="Email",font=(("Arial"),15),bg=fgc,fg=bgc)
    enl.place(x=330,y=200)

    enb=Entry(pf,width=45,font=(("Arial"),15),bd=0)
    enb.place(x=410,y=200)
    enb.insert(INSERT,d[4])
    enb.config(state="disabled")

    rnl=Label(pf,text="Rollnumber",font=(("Arial"),15),bg=fgc,fg=bgc)
    rnl.place(x=280,y=280)

    rnb=Entry(pf,font=(("Arial"),15),bd=0)
    rnb.place(x=410,y=280)
    rnb.insert(INSERT,d[0])
    rnb.config(state="disabled")

    renl=Label(pf,text="Register number",font=(("Arial"),15),bg=fgc,fg=bgc)
    renl.place(x=235,y=360)

    renb=Entry(pf,width=20,font=(("Arial"),15),bd=0)
    renb.place(x=410,y=360)
    renb.insert(INSERT,d[1])
    renb.config(state="disabled")

    dnl=Label(pf,text="Date of birth",font=(("Arial"),15),bg=fgc,fg=bgc)
    dnl.place(x=270,y=440)

    dnb=Entry(pf,font=(("Arial"),15),bd=0)
    dnb.place(x=410,y=440)
    dnb.insert(INSERT,d[5])
    dnb.config(state="disabled")

    anl=Label(pf,text="Address",font=(("Arial"),15),bg=fgc,fg=bgc)
    anl.place(x=300,y=520)

    anb=Text(pf,font=(("Arial"),15),height=10,width=45,bd=0)
    anb.place(x=410,y=520)
    anb.insert(INSERT,d[6])
    anb.config(state="disabled")

    # disableentry()
    eb=Button(root,text="EDIT",font=(("Arial"),15),fg=fgc,bg=bgc,width=10)
    eb.place(x=900,y=795)

    sb=Button(root,text="SAVE",font=(("Arial"),15),fg=fgc,bg=bgc,width=10)
    sb.place(x=1050,y=795)

def drivealert():
    daf=Frame(root,width=955,height=845,bg=fgc)
    daf.place(x=240,y=0)

    dah=Label(daf,text="Shop Up",font=(("Arial"),15),fg=fgc,bg=bgc)
    dah.place(x=20,y=60)

    dah=Label(daf,text="It is a Fullstack B2B company providing various
roles",font=(("Arial"),15),fg=fgc,bg=bgc)
    dah.place(x=20,y=100)

    dah=Label(daf,text="Eligibility : 60% throughout \n Arrears : No standing
arrears",font=(("Arial"),15),fg=fgc,bg=bgc)
```

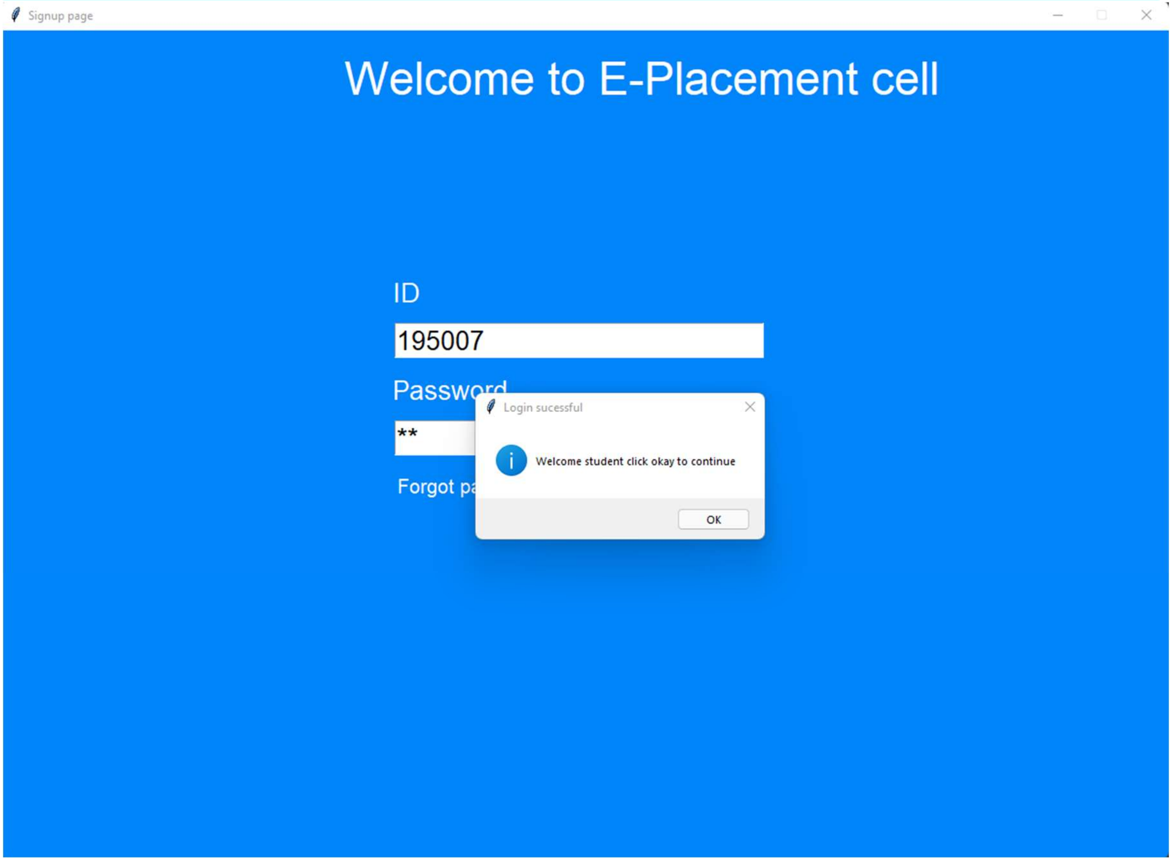
```
    dah.place(x=20,y=140)

    dah=Label(daf,text="For more details",font=(("Arial"),15),fg=fgc,bg=bgc)
    dah.place(x=20,y=220)

    dah=Button(daf,text="click here",font=(("Arial"),12),fg=fgc,bg=bgc,width=15)
    dah.place(x=180,y=220)

    dah=Label(daf,text="ZOHO",font=(("Arial"),15),fg=fgc,bg=bgc)
    dah.place(x=20,y=320)

    dah=Label(daf,text="It is a Product based company providing various
roles",font=(("Arial"),15),fg=fgc,bg=bgc)
    dah.place(x=20,y=360)

    dah=Label(daf,text="Eligibility : Candidates currently enrolled in BE/Btech \n Arrears :
No standing arrears",font=(("Arial"),15),fg=fgc,bg=bgc)
    dah.place(x=20,y=400)

    dah=Label(daf,text="For more details",font=(("Arial"),15),fg=fgc,bg=bgc)
    dah.place(x=20,y=460)

    dah=Button(daf,text="click here",font=(("Arial"),12),fg=fgc,bg=bgc,width=15)
    dah.place(x=180,y=460)

def f(n="Vinod"):

    global root
    root=Tk()
    root.geometry("1200x850+242+47")

    root.resizable(False,False)
    root.title("Student dashboard")
    root.config(bg=fgc)

    ll=Label(root,text=f"Welcome {n}",font=(("Arial"),20),bd=2,fg=fgc,bg=bgc)
    ll.grid(row=0,column=0,sticky='nw',padx=12,pady=12)

    pb=Button(root,text="Personal
details",font=(("Arial"),15),fg=fgc,bg=bgc,width=20,command=personal)
    pb.grid(row=1,column=0,pady=15)

    db=Button(root,text="Drive
alert",font=(("Arial"),15),fg=fgc,bg=bgc,width=20,command=drivealert)
    db.grid(row=2,column=0,pady=15)
```

```
   ub=Button(root,text="Upload
documents",font=(("Arial"),15),fg=fgc,bg=bgc,width=20,command=uploaddocs)
   ub.grid(row=3,column=0,pady=15)

   ub=Button(root,text="Change
credential",font=(("Arial"),15),fg=fgc,bg=bgc,width=20,command=changeCredential)
   ub.grid(row=4,column=0,pady=15)
   changeCredential()
   root.mainloop()
f()
```

CHAPTER 6

OUTPUT



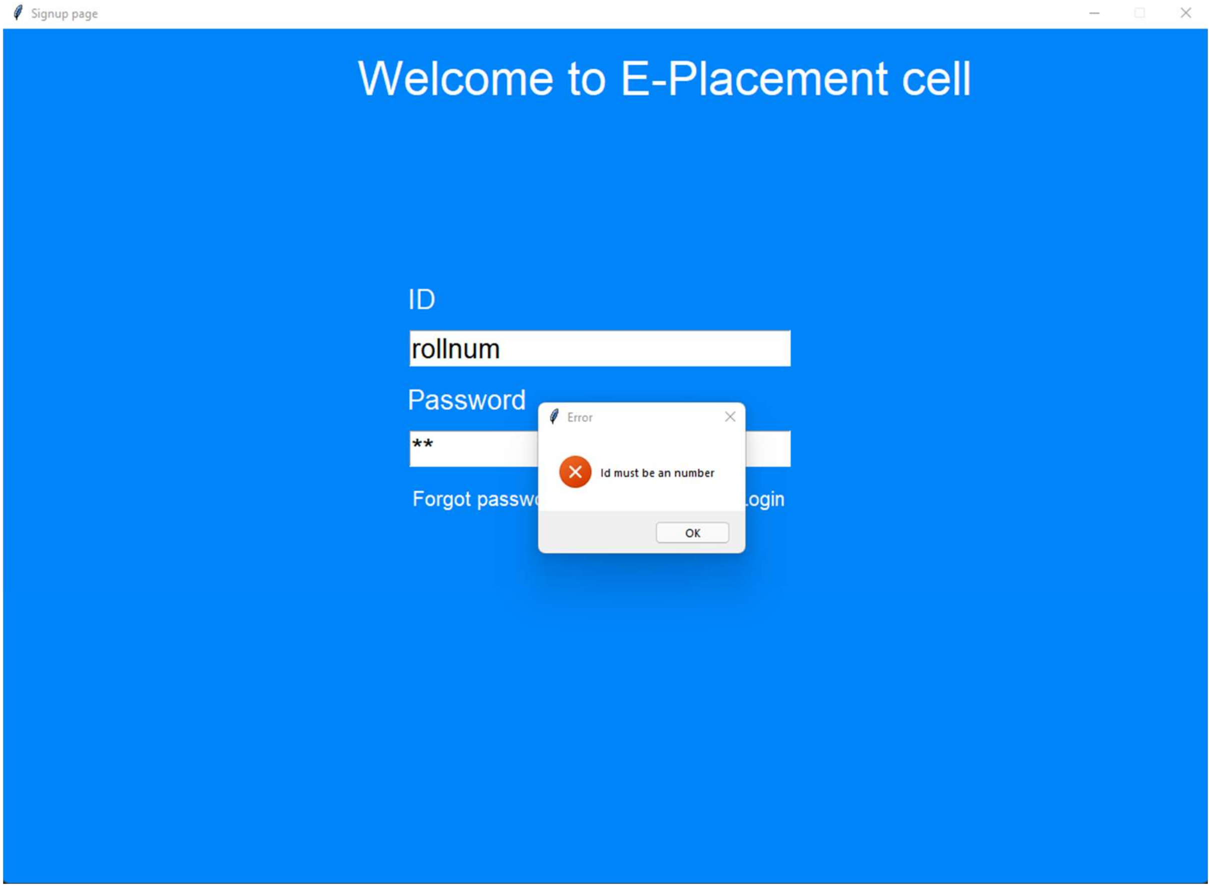


ID: 195007

Password: **

Login sucessful

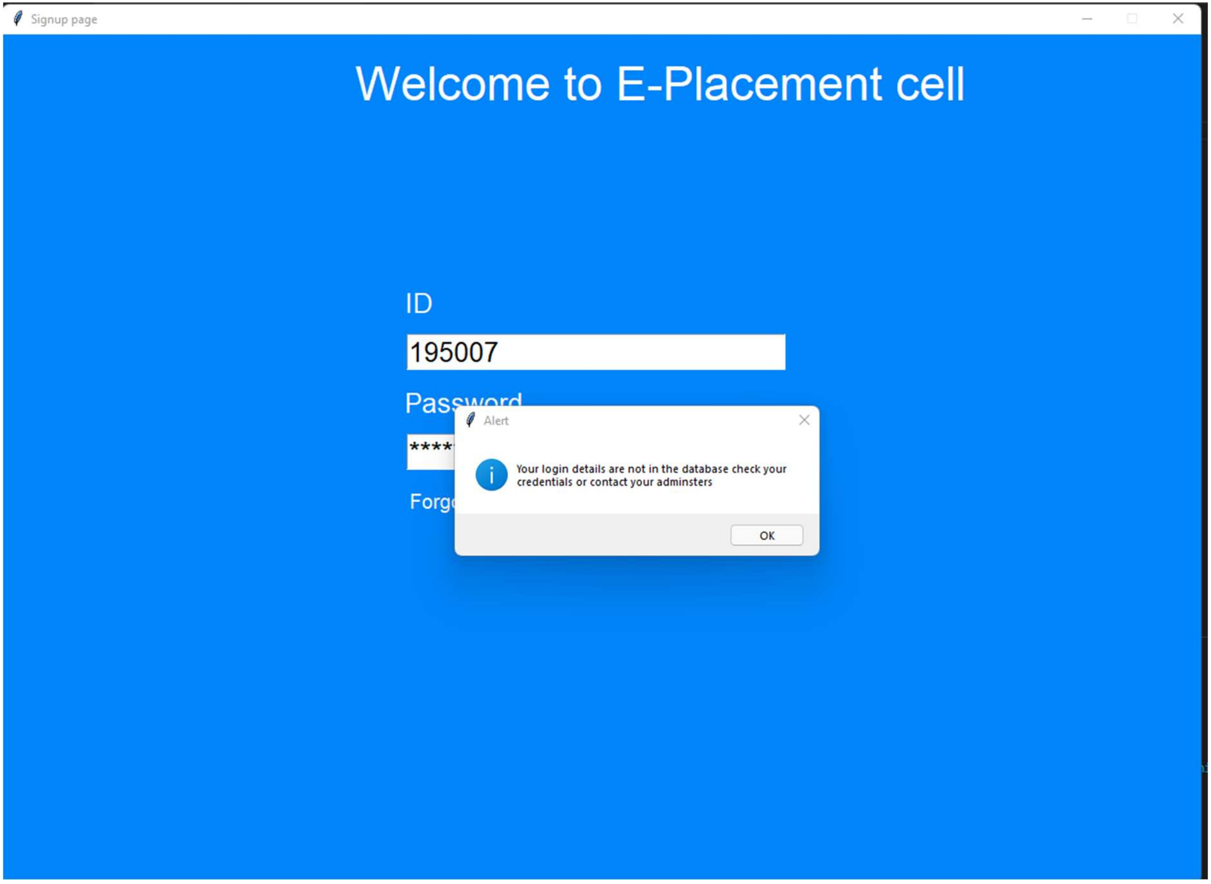
Welcome student click okay to continue



rollnum

**

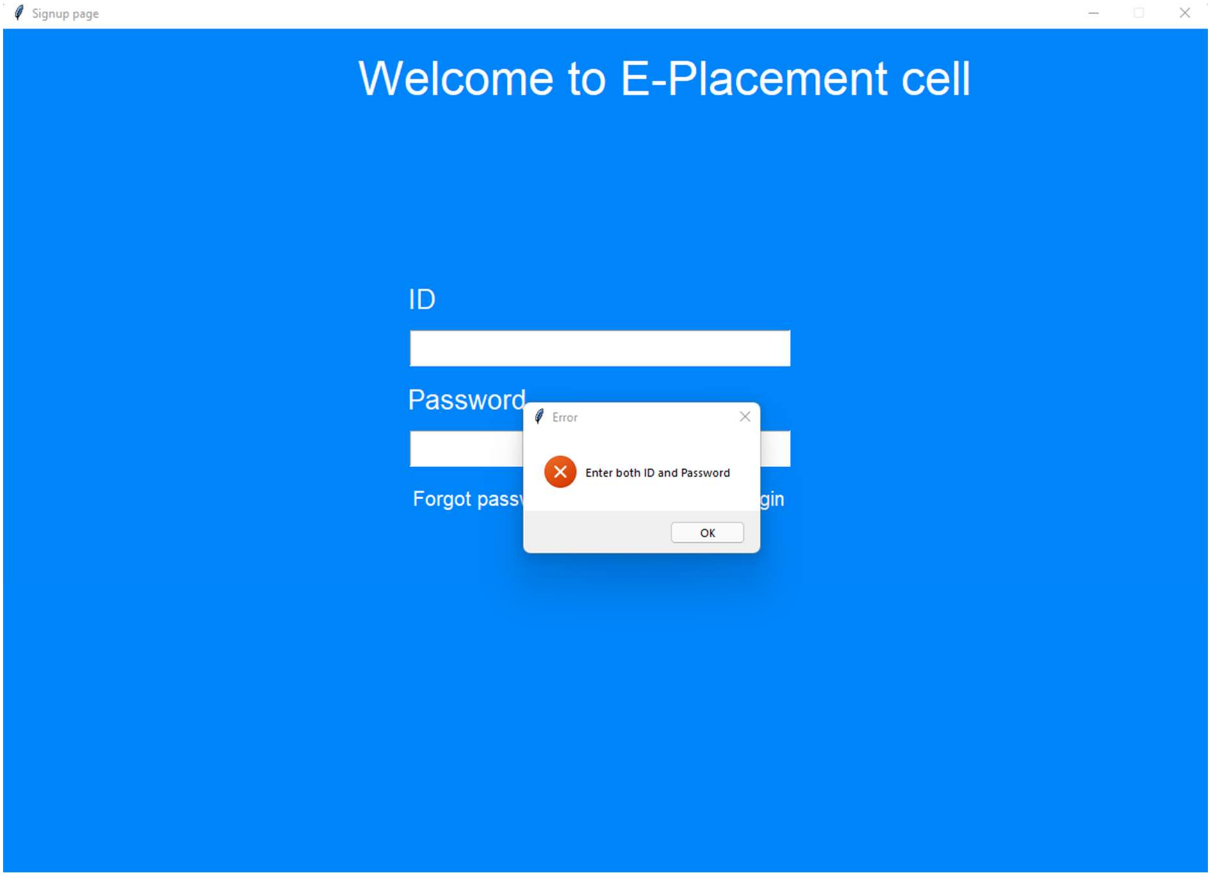
Id must be an number



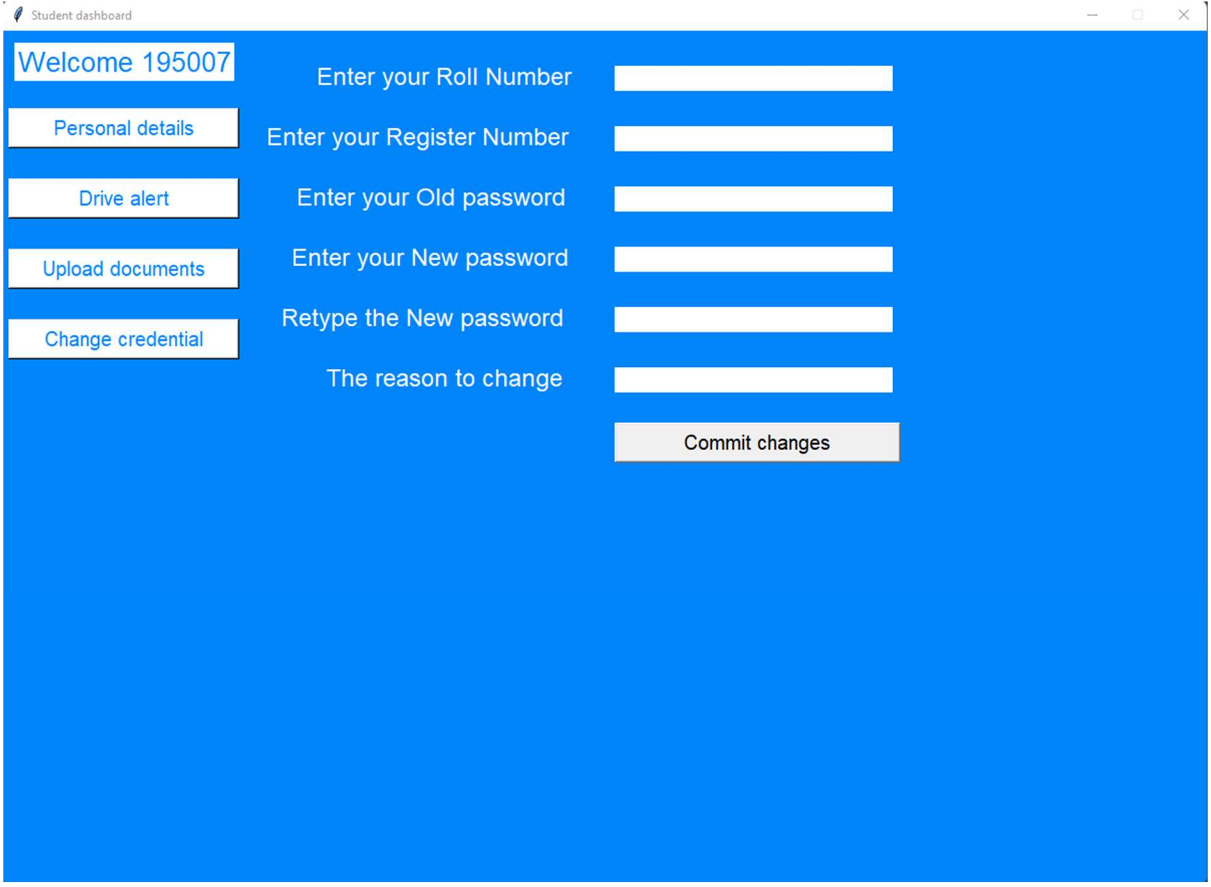
195007

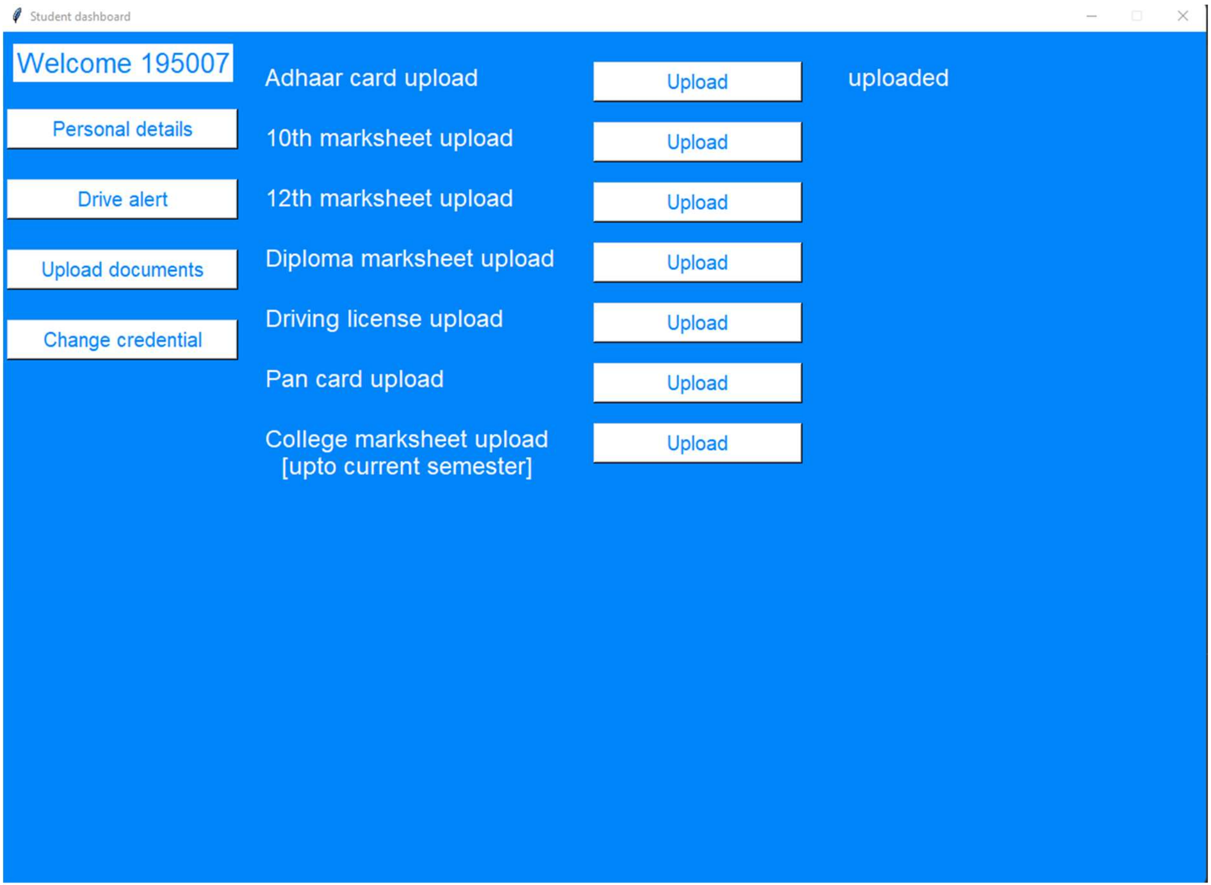
****

Your login details are not in the database check your credentials or contact your adminsters



Enter both ID and Password

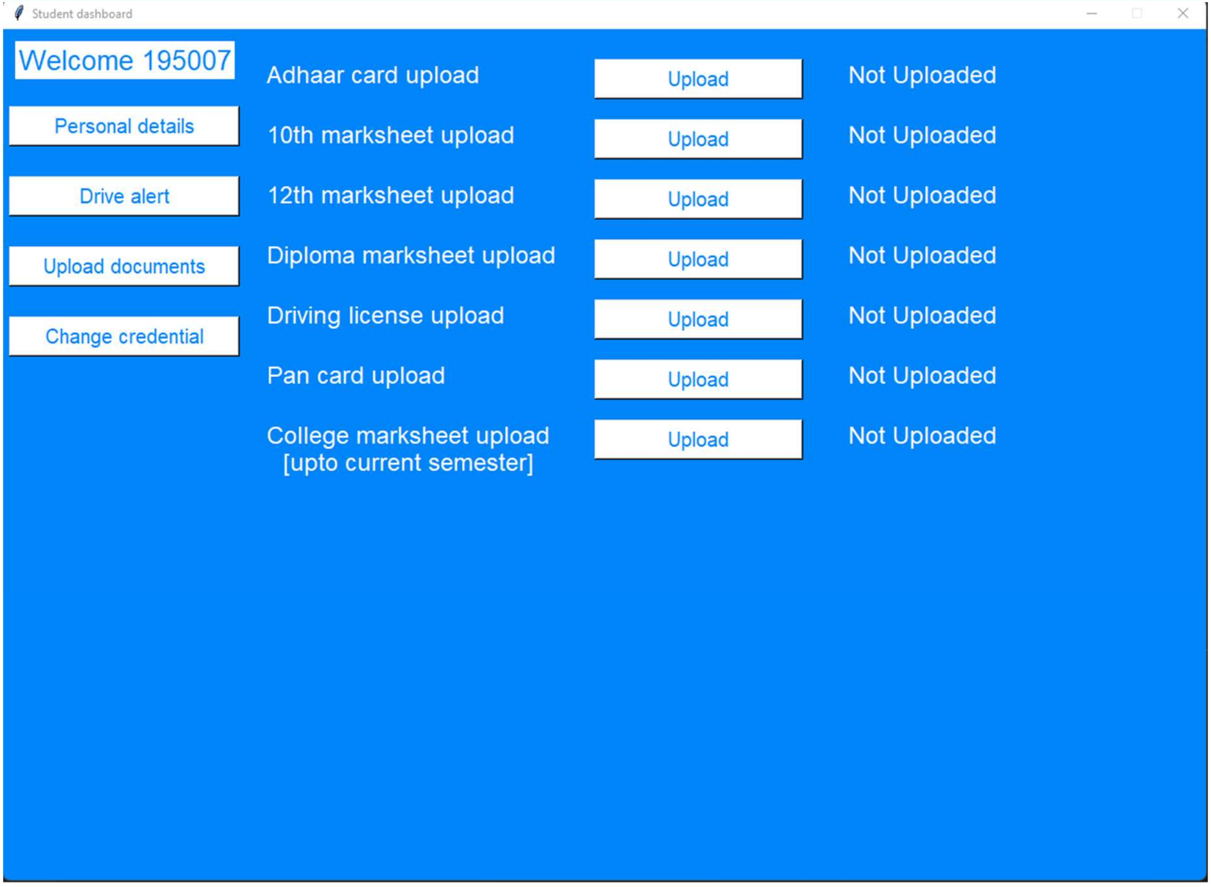




Student dashboard

Welcome 195007

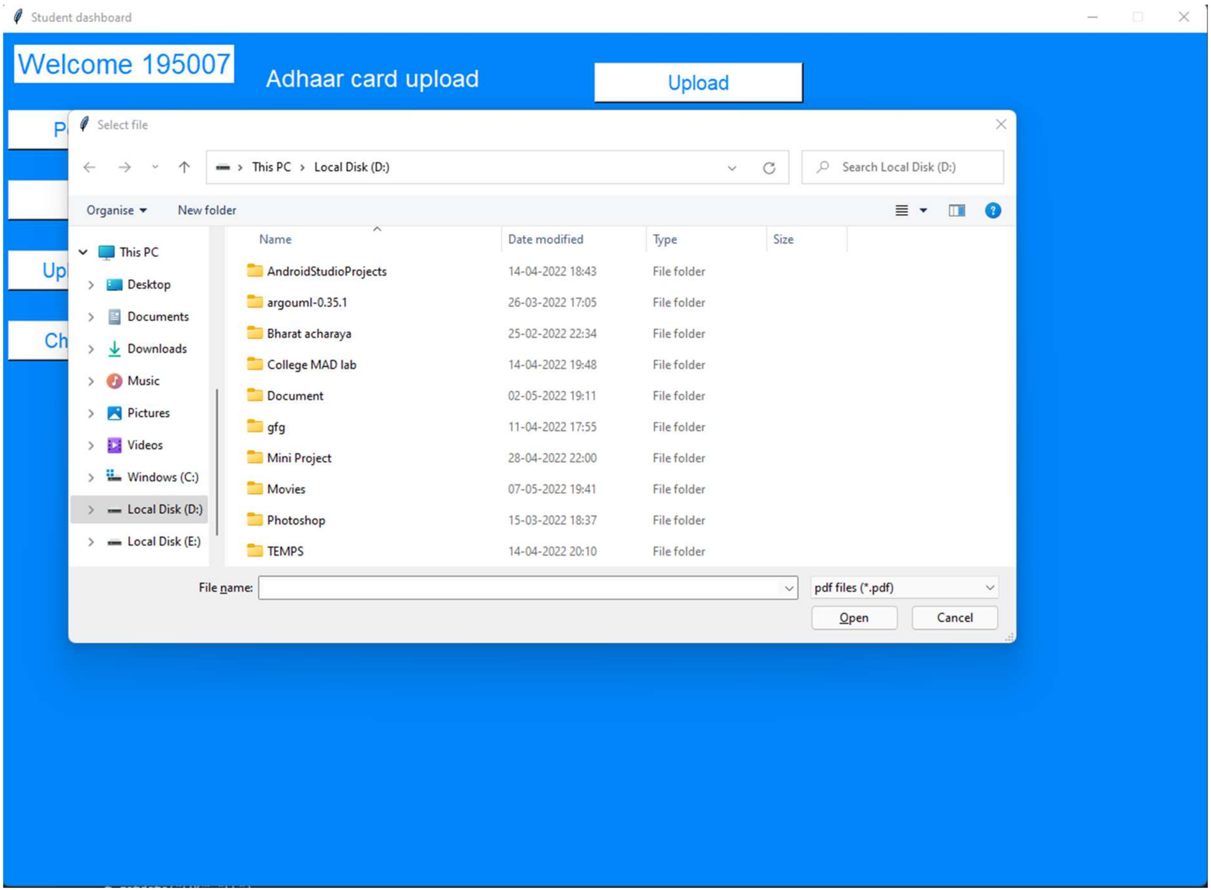
Adhaar card upload	Upload	uploaded
10th marksheet upload	Upload	
12th marksheet upload	Upload	
Diploma marksheet upload	Upload	
Driving license upload	Upload	
Pan card upload	Upload	
College marksheet upload [upto current semester]	Upload	



Student dashboard

Welcome 195007

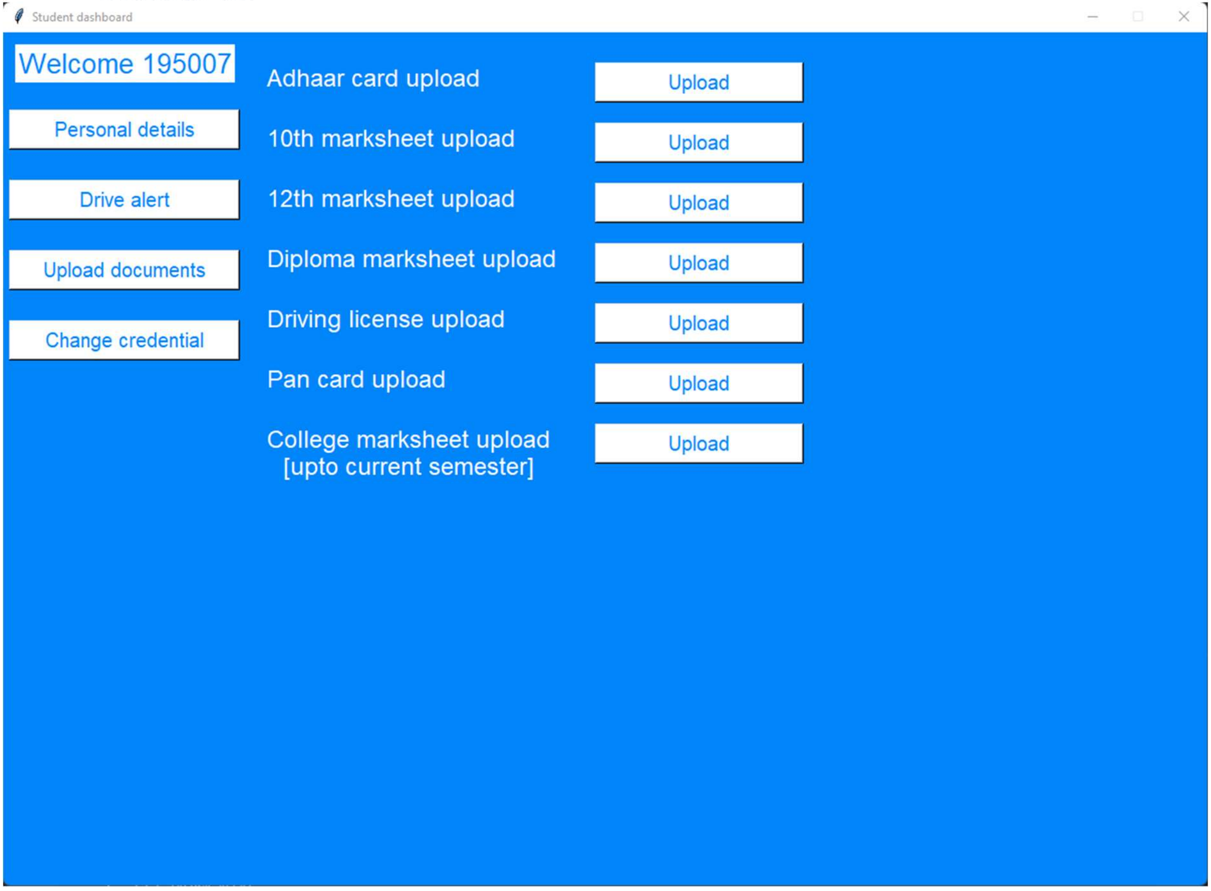
Adhaar card upload	Upload	Not Uploaded
10th marksheet upload	Upload	Not Uploaded
12th marksheet upload	Upload	Not Uploaded
Diploma marksheet upload	Upload	Not Uploaded
Driving license upload	Upload	Not Uploaded
Pan card upload	Upload	Not Uploaded
College marksheet upload [upto current semester]	Upload	Not Uploaded

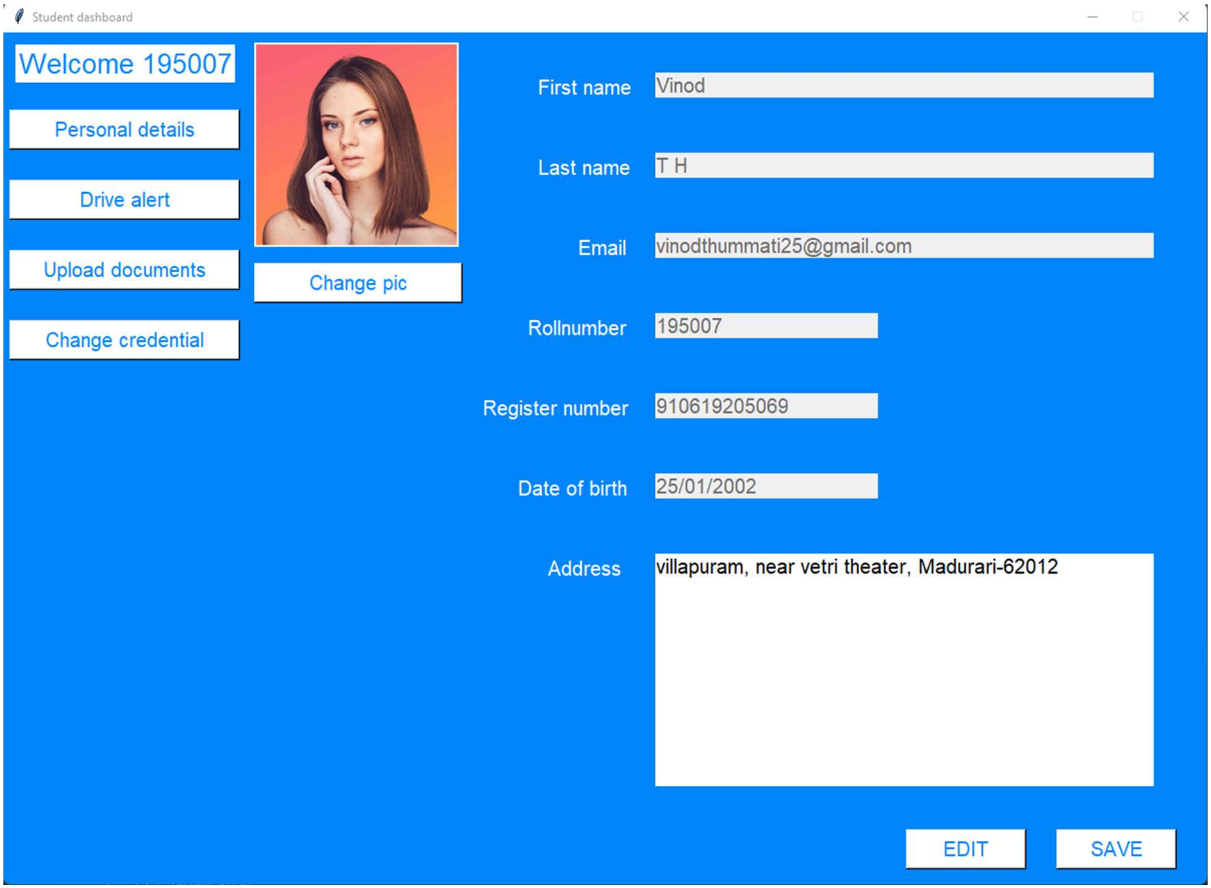


Name	Date modified	Type	Size
AndroidStudioProjects	14-04-2022 18:43	File folder	
argouml-0.35.1	26-03-2022 17:05	File folder	
Bharat acharaya	25-02-2022 22:34	File folder	
College MAD lab	14-04-2022 19:48	File folder	
Document	02-05-2022 19:11	File folder	
gfg	11-04-2022 17:55	File folder	
Mini Project	28-04-2022 22:00	File folder	
Movies	07-05-2022 19:41	File folder	
Photoshop	15-03-2022 18:37	File folder	
TEMPS	14-04-2022 20:10	File folder	

File name:

pdf files (*.pdf)





First name Vinod

Last name T H

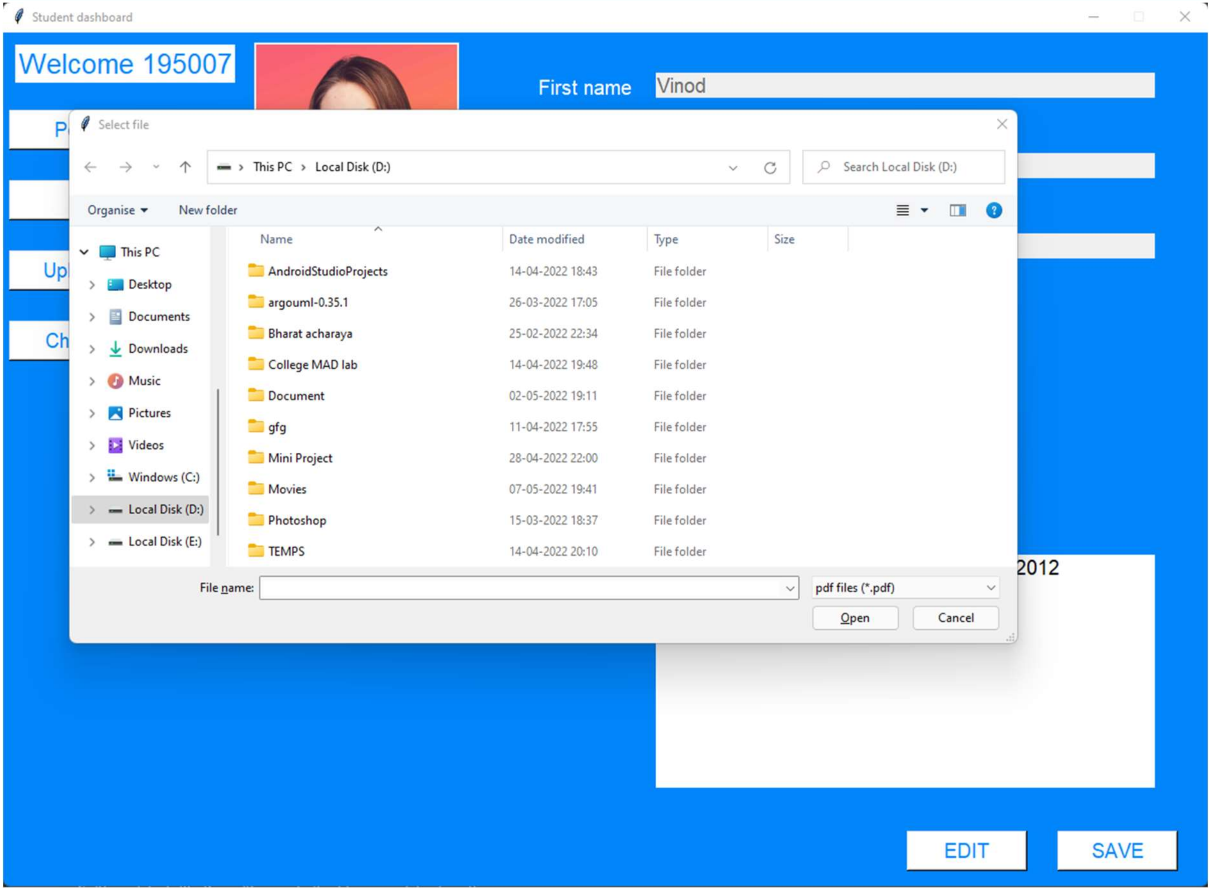
Email vinodthummati25@gmail.com

Rollnumber 195007

Register number 910619205069

Date of birth 25/01/2002

Address villapuram, near vetri theater, Madurari-62012



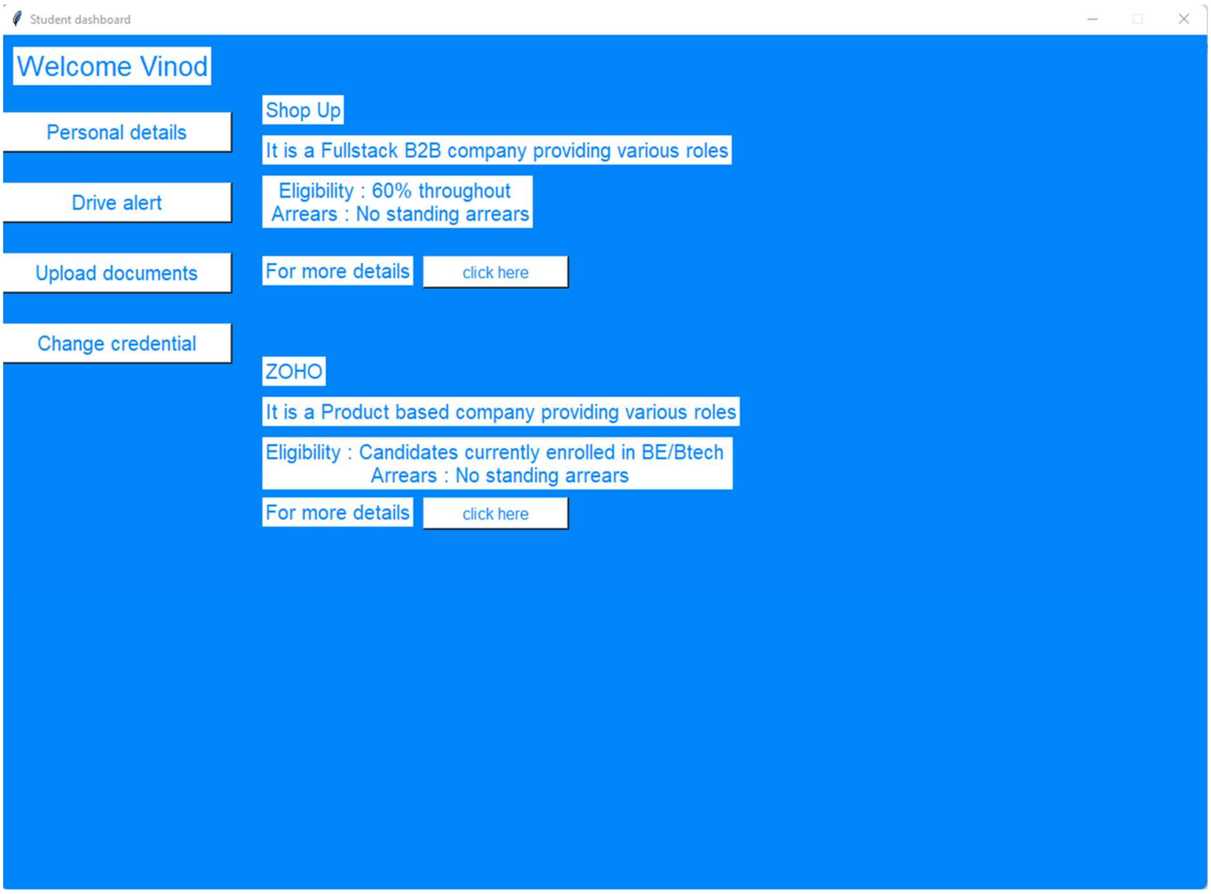
First name Vinod

2012

Name	Date modified	Type	Size
AndroidStudioProjects	14-04-2022 18:43	File folder	
argouml-0.35.1	26-03-2022 17:05	File folder	
Bharat acharaya	25-02-2022 22:34	File folder	
College MAD lab	14-04-2022 19:48	File folder	
Document	02-05-2022 19:11	File folder	
gfg	11-04-2022 17:55	File folder	
Mini Project	28-04-2022 22:00	File folder	
Movies	07-05-2022 19:41	File folder	
Photoshop	15-03-2022 18:37	File folder	
TEMPS	14-04-2022 20:10	File folder	

File name:

pdf files (*.pdf)



Shop Up

It is a Fullstack B2B company providing various roles

Eligibility : 60% throughout
Arrears : No standing arrears

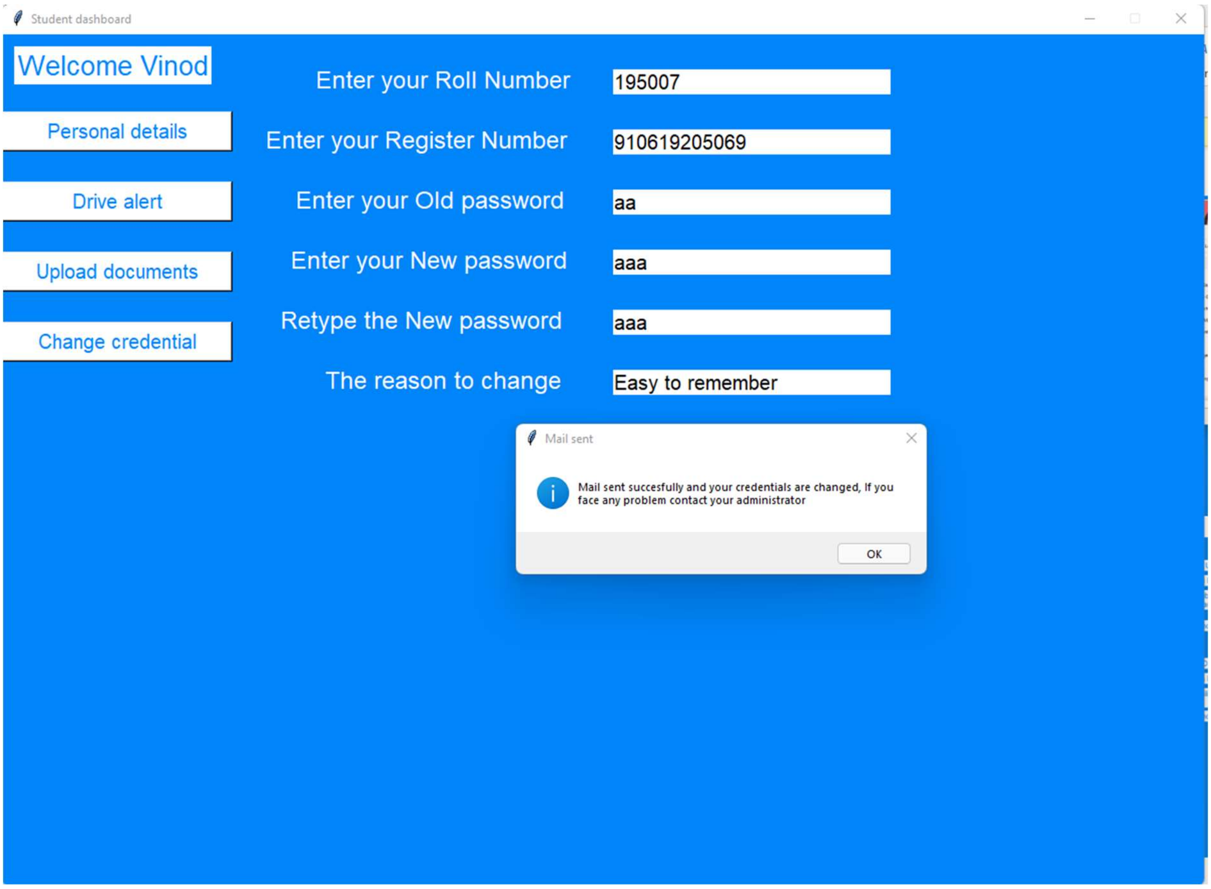
For more details

ZOHO

It is a Product based company providing various roles

Eligibility : Candidates currently enrolled in BE/Btech
Arrears : No standing arrears

For more details



Enter your Roll Number	195007
Enter your Register Number	910619205069
Enter your Old password	aa
Enter your New password	aaa
Retype the New password	aaa
The reason to change	Easy to remember

Mail sent succesfully and your credentials are changed, If you face any problem contact your administrator

CHAPTER 7

CONCLUSION

Our proposed system functions in accordance with the requirements. It can successfully login and register authorised users on the system. In our system, the administrator may check the student list for those who are qualified based on the criteria set by the company, notify them immediately, and change the information at any time. For all three components, our solution is secure and user-friendly.

The growing need for comfort and the integration of all data in one location has always been a difficult task for everyone. We hope that by introducing this software-based training and placement portal, we will make students' and administrators' life a bit simpler by providing an alternative to the present approach.

This portal's easy accessibility and functionality will allow for simple control of the allocation process during the placement phase. With the rising desire for digitization in all aspects of daily life, we may predict a high demand for such portals in the near future, as well as the convenience it will bring to everyone's lives. Also, with the fast rising concerns about global warming as a result of increased deforestation for the vast quantity of paper that it takes, we have a little part to play in saving Mother Nature.

CHAPTER 8

FUTURE SCOPE

The primary goal of designing this software was to minimise the possibility of mistakes in manual operations.

Make time for the procedure. Students are also immediately contacted through SMS.

Other features, such as notifying students about potential employment both on and off campus, can be added to the updated versions. The SMS integration is not supported by the system. As a result, it may be updated to include SMS integration. Other capabilities, such as analytics, may be added to this site in the future to track students' development in certain areas. Following an analysis, this system will warn pupils of the areas in which they fall short.

Limitations of project

- Runs on Local Database.
- Minimum system screen resolution needed (1366x720 and more)
- Poor UI

CHAPTER 9

Reference / Bibliography

- https://docs.python.org/3/tutorial/
- https://www.programiz.com/python-programming
- https://www.w3schools.com/python/
- https://www.javatpoint.com/python-tkinter
- https://www.geeksforgeeks.org/python-tkinter-tutorial/
- https://www.pythontutorial.net/tkinter/
- https://realpython.com/python-gui-tkinter/
- https://www.sqlite.org/index.html
- https://www.youtube.com/watch?v=WnMQ8HlmeXc
- https://www.youtube.com/watch?v=yQSEXcf6s2I&list=PLCC34OHNcOtoC6GglhF3ncJ5rLwQrLGnV
- Gunderloy, Jorden BPB Publications (2000) - “Mastering SQL Server”
- Thereon Willis Worx publications (2000) - “Beginning SQL Server”.