CAPSTONE PROJECT REPORT

(Project Term January-May 2023)

Student Registration Management System

Submitted by

Name: Satwik Uppada **Registration Number: 12111298**

Course Code: INT 216

Under the Guidance of

Waseem Ud Din Wani Sir

School of Computer Science and Engineering

LOVELY PROFESSIONAL UNIVERSITY

DECLARATION

We hereby declare that the project work entitled "Student Registration Management" is an authentic record of our own work carried out as requirements of Capstone Project for the award of B. Tech degree in **C.S.E** (Programme Name) from Lovely Professional University, Phagwara, under the guidance of **Waseem Ud Din Wani Sir**, during August to November 2022. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Name of Student: Satwik Uppada

Registration Number: 12111298

Date:25/04/2023

CERTIFICATE

This is to certify that the declaration statement made by this this student is correct to the best of my knowledge and belief. He has completed this Capstone Project under my guidance and supervision. The present work is the result of his original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Capstone Project is fit for the submission and partial fulfillment of the conditions for the award of B. Tech degree in CSE from Lovely Professional University, Phagwara.

Signature and Name of the Mentor: Waseem Ud Din Wani Sir

Designation: Teaching Assistent

School of Computer Science and Engineering,
Lovely Professional University,
Phagwara, Punjab.

Date: 25/04/2023

ACKNOWLEDGEMENT

"I would like to express my sincere gratitude and appreciation to all those who have contributed to the successful completion of this capstone project. First and foremost, I would like to thank "**Waseem Ud Din Wani Sir**", whose guidance, encouragement, and feedback have been invaluable throughout this project. Their insights and suggestions have helped me to improve my work and bring it to completion.

I would also like to express my gratitude to the faculty members and staff of Lovely Professional University, who have provided me with the necessary resources and support to undertake this project. Their unwavering support and encouragement have been critical in making this project a success.

I am also grateful to my friends, who have been a constant source of support and motivation throughout my academic journey. Their love, encouragement, and belief in me have been instrumental in helping me achieve my goals.

Lastly, I would like to express my gratitude to all the participants and stakeholders who generously shared their time, knowledge, and expertise with me during the course of this project. Their insights and contributions have been crucial in shaping the direction of my research and have helped me to develop a deeper understanding of the subject matter.

Once again, I would like to thank everyone who has contributed to this project in any way, and I am truly grateful for the opportunity to undertake this work."

TABLE OF CONTENTS

1. INTRODUCTION

"Are you tired of managing student registrations manually? Are you struggling to keep track of important student details, such as their marks, educational history, and personal information? If so, then you'll be interested in the Student Registration Management System I have developed. This system is designed to simplify the registration process by taking inputs of student details such as name, gender, date of birth, email, mobile number, religion, and educational background, including marks obtained in 10th and 12th grade, school and college details, and registration number. The system can also store student pictures and manually input the date of registration.

One of the most significant advantages of the Student Registration Management System is that it can save, update, and search for student records using their unique registration number. The system is user-friendly and can automate many of the tasks associated with student registration. In this capstone project, I explore the design and implementation of this system and demonstrate its effectiveness in simplifying the registration process for students and administrators alike."

The Student Registration Management system is a capstone project designed to facilitate the management of student registration records in an educational institution. The project is developed using the Python programming language and the Tkinter GUI toolkit.

1.1 Context and Background

"Student registration is a critical process in any educational institution. It involves collecting and managing student data, ensuring compliance with regulatory requirements, and allocating resources to courses and faculties. However, manual registration systems can be cumbersome and error-prone, leading to delays and inaccuracies in the registration process. Moreover, the increasing number of students and courses in educational institutions further complicates the registration process. As a result, there is a growing need for a reliable, efficient, and user-friendly registration management system that can simplify the process for students and administrators alike.

In response to this need, I have developed a Student Registration Management System. This system is designed to automate many of the tasks associated with student registration, including data collection, record-keeping, and course allocation. It is also designed to be user-friendly, with an intuitive interface that allows students and administrators to access and manage their information easily.

The Student Registration Management System is built using modern software development tools and follows industry best practices in software engineering. It is scalable, robust, and secure, and can be customized to meet the specific needs of educational institutions. In this capstone project, I explore the design and implementation of the system and demonstrate how it can improve the registration process for students and administrators alike."

2.OBJECTIVE OF THE PROJECT

1. Create a user-friendly interface that allows students and administrators to input and manage student registration data, including personal information and academic records.
2. Automate the data collection and record-keeping process to improve accuracy and efficiency and reduce errors and delays.
3. Enable easy retrieval and search of student data based on registration numbers, making it easier to manage and allocate resources.
4. Ensure compliance with regulatory requirements by maintaining accurate and up-to-date student records.
5. Enhance the security and scalability of the registration process by leveraging modern software development tools and practices.
6. Streamline the course allocation process by automating the assignment of courses to students and managing course schedules.
7. Improve the overall efficiency of the registration process, enabling educational institutions to allocate resources more effectively.

In summary, the objective of your project is to create a comprehensive and user-friendly Student Registration Management System using Python and Tkinter that simplifies the registration process for students and administrators, improves accuracy and efficiency, and provides educational institutions with valuable data and insights to allocate resources more effectively.

3.DESCRIPTION OF THE PROJECT

The Student Registration Management System is a comprehensive software solution designed to facilitate the management of student registration records in educational institutions. The project is developed using the Python programming language and the Tkinter GUI toolkit, making it user-friendly and easy to navigate.

The system can take inputs of student details, such as name, gender, date of birth, email, mobile number, religion, and educational background, including marks obtained in 10th and 12th grade, school and college details, and registration number. In addition, the system can store student pictures and manually input the date of registration.

The primary objective of the Student Registration Management System is to simplify the registration process for both students and administrators. One of its most significant advantages is that it can save, update, and search for student records using their unique registration number, which helps reduce the time and effort required to manage student records. The system can also generate reports and provide real-time updates to help administrators make informed decisions.

Overall, the Student Registration Management System is an efficient and effective tool for managing student registration records in educational institutions. In this capstone project, the design and implementation of the system are explored, demonstrating its effectiveness in simplifying the registration process and improving efficiency for both students and administrators.

3.1 KEY FEATURES

1. **User-friendly interface**: The Student Registration Management System has an easy-to-use graphical user interface that allows users to navigate through the system seamlessly. It is designed to be intuitive and straightforward, making it easy for users to interact with the system and input data without any difficulties.
2. **Automated registration process**: The system automates many of the tasks associated with the student registration process, such as generating unique registration numbers, storing student data, and updating records as necessary. This feature significantly reduces the time and effort required for manual record-keeping and processing.

3. **Comprehensive student record-keeping**: The Student Registration Management System can store and manage a wide range of student data, including personal details, educational qualifications, and contact information. This feature makes it easier for administrators to access and analyse student records quickly.
4. **Efficient search functionality**: The system's search functionality is another essential feature. It enables users to search for student records quickly and efficiently using their unique registration number. This feature saves time and effort, particularly when searching for specific records.
5. **Picture uploading and viewing**: The Student Registration Management System also allows for picture uploading, which makes it easy to associate student pictures with their records. It also has a viewing feature that allows users to see the pictures of the student for easy identification.
6. **Robust security features**: The system has robust security features to ensure that student data is protected from unauthorized access. This feature helps to maintain the confidentiality and privacy of student records.

Overall, these features make the Student Registration Management System a powerful tool for educational institutions that need to manage student registration records efficiently.

3.2 KEY FUNCTIONALITIES

1. **Student Registration:** The system allows for the registration of new students by capturing their personal details, educational background, and registration number.

2. **Record Management:** The system can store and manage student records, including their personal details, academic performance, and registration status.

3. **Record Update:** The system enables the update of student records, such as change of personal information, addition of new academic records, and change of registration status.

4. **Record Search:** The system allows for the search of student records using their unique registration number, enabling administrators to quickly locate and retrieve student information.

5. **Automation:** The system automates many tasks associated with student registration, such as the creation of registration numbers, which reduces the workload of administrators.
6. **User-Friendly**: The system is designed to be user-friendly, with an easy-to-use interface that allows for easy navigation and data input.
7. **Secure:** The system employs robust security measures to protect student data, ensuring the confidentiality and integrity of student records.

3.3 UNIQUE FEATURE

1. **Time-saving and cost-effective:** The Student Registration Management System is a time-saving and cost-effective solution that eliminates the need for manual record-keeping and reduces administrative workload.

3.4 HOW IT IS DEVELOPED

The Student Registration Management System is developed using the Python programming language and the Tkinter GUI toolkit for designing the user interface. The system is also integrated with Microsoft Excel for data storage and management.

The development process involves several stages, including requirements gathering, system design, implementation, and testing. During the requirements gathering stage, the system's functional and non-functional requirements are identified and documented. The design stage involves creating the system's architecture, user interface, and database structure.

After the design stage, the implementation process begins, where the system is coded using Python and the Tkinter library to create the user interface. Microsoft Excel is used to store and manage the student registration data. The system is tested thoroughly to ensure that it is free of errors and meets the functional and non-functional requirements.

Overall, the development of the Student Registration Management System is a comprehensive process that involves several stages, tools, and technologies to create a functional and user-friendly system for managing student registration records.

3.4 HOW IT WORKS

The Student Registration Management System greets the user with a student portal homepage upon launching, featuring a heading and a single button. When the button is clicked, a new window for the student registration portal opens, which includes seven buttons, such as the home icon, update icon, search icon, upload image button, save button, reset button, and exit button.

Clicking the home icon will hide the current window and return the user to the homepage. When the update icon is clicked, the user can update the information for a specific registration number entered in the search entry. A message will pop up once the update is completed, stating that the update was successful.

Clicking the search icon will display the information of the student with that specific registration number. The save button will be disabled in this case. If the user clicks the upload image button, the system will take them to the file manager to select a picture.

When the user clicks the save button, the information will be saved to the Excel file, and a message will appear indicating that the data has been successfully saved. If the reset button is clicked, all entries in the entry boxes will be cleared. Finally, if the user clicks the exit button, the GUI will close completely.

3.5 HOW IT IMPLEMENTED

The Student Registration Management System is implemented using the Python programming language and the Tkinter GUI toolkit. The system is designed to be modular and uses a Model-View-Controller (MVC) architecture to separate the different components of the application.

The Model component is responsible for managing the student data and storing it in an Excel spreadsheet. The Controller component interacts with the Model and the View components and manages the flow of information between them. The View component is responsible for displaying the graphical user interface (GUI) to the user.

The GUI was designed using Tkinter, which provides a set of widgets and tools for creating user interfaces. The home page consists of a heading and a single button that opens the student registration portal. The registration portal contains seven buttons, including the Home, Update, and Search icons, an Upload Image button, a Save button, a Reset button, and an Exit button.

When the user clicks on the Home icon, it hides the current window and displays the home page. When the user clicks on the Update icon, it prompts the user to enter the registration number of the student whose details they wish to update. After clicking the Update button, a message pops up indicating that the details have been successfully updated.

When the user clicks on the Search icon, it prompts the user to enter the registration number of the student whose details they wish to view. The system then displays the student's information, and the Save button is disabled. When the user clicks on the Upload Image button, it takes them to the file manager to select a picture of the student.

4.SOURCE CODE

```
from tkinter import *
import customtkinter
from PIL import Image, ImageTk
import os

customtkinter.set_default_color_theme("blue")
app = customtkinter.CTk()
app.state("zoomed")
app.resizable(False, False)
app.title("Home Page")


def cover_page():
    img1 = ImageTk.PhotoImage(Image.open("Images/bg1.jpg"))
    l1 = customtkinter.CTkLabel(master=app, image=img1)
    l1.pack()

    frame = customtkinter.CTkFrame(master=l1, width=1020, height=820, corner_radius=15, fg_color="white")
    frame.place(relx=0.5, rely=0.5, anchor=CENTER)

    l2 = customtkinter.CTkLabel(master=frame, text="Registration Management System", font=('arial', 48, "bold"),
                                text_color="#186677")
    l2.place(x=120, y=45)

    login_img = Image.open("Images/login pic.jpg")
    resized_login_img = login_img.resize((400, 400))
    img2 = ImageTk.PhotoImage(resized_login_img)
    l2 = customtkinter.CTkLabel(master=frame, image=img2,text="")
    l2.place(x=300, y=150)

    student_login_image = Image.open("Images1/student_login.png")
    # Resize the image
    resized_student_image = student_login_image.resize((90, 90))
    photo1 = ImageTk.PhotoImage(resized_student_image)

    student_button = customtkinter.CTkButton(master=frame, image=photo1, text="Student Portal",
fg_color="#d1d2d7",corner_radius=15, font=("arial", 20, "bold"), text_color="black",
command=student_portal_app)
    student_button.place(x=370, y=600)

    app.mainloop()
```

```
img1 = ImageTk.PhotoImage(Image.open("Images1/upload photo.png"))
imageicon3 = ImageTk.PhotoImage(Image.open("Images1/search.png"))
imageicon4 = ImageTk.PhotoImage(Image.open("Images1/Layer 4.png"))
img = ImageTk.PhotoImage(Image.open("Images1/upload photo.png"))
home_icon = ImageTk.PhotoImage(Image.open("Images1/home_icon.png"))


def student_portal_app():

    # STUDENT REGISTRATION SYSTEM
    from datetime import date
    import customtkinter
    from tkinter import messagebox
    from tkinter import filedialog
    from PIL import Image, ImageTk
    import os
    from tkinter.ttk import Combobox
    import openpyxl, xlrd
    from openpyxl import Workbook
    import pathlib

    background = "#282631"
    framebg = "#EDEDED"
    framefg = "#06283D"

    root = Toplevel(app)
    # Hide the main page window
    app.withdraw()
    root.title("Student Registration System")
    root.state("zoomed")
    root.resizable(False, False)

    root.geometry("1270x700+210+100")
    root.config(bg=background)

    file = pathlib.Path("Student_data.xlsx")
    if file.exists():
        pass
    else:
        file = Workbook()
        sheet = file.active
        sheet['A1'] = "Registration No"
        sheet['B1'] = "Name"
        sheet['C1'] = "Email"
        sheet['D1'] = "Gender"
        sheet['E1'] = "DOB"
        sheet['F1'] = "Date of Registration"
        sheet['G1'] = "Religion"
        sheet['H1'] = "Mobile Number"
        sheet['I1'] = "Father Name"
        sheet['J1'] = "Mother Name"
        sheet['K1'] = "Father's Occupation"
        sheet['L1'] = "Mother's Occupation"
        sheet['M1'] = "Tenth Class Marks"
        sheet['N1'] = "Tenth Passout Year "
        sheet['O1'] = "School"
        sheet['P1'] = "Intermediate Marks"
        sheet['Q1'] = "Inter Passout Year"

        sheet['R1'] = "College"

    file.save('Student_data.xlsx')
```

```
def home_page():
    root.withdraw()
    app.deiconify()

# EXIT
def Exit():
    root.destroy()

#######################search#####################3
def search():
    text = Search.get()
    Clear()
    savebutton.configure(state='disable')
    file = openpyxl.load_workbook("Student_data.xlsx")
    sheet = file.active
    if text == "":
        messagebox.showerror("Error","Lack of Information")
    else:
        for row in sheet.rows:
            if row[0].value == int(text):
                name = row[0]
                #       print(str(name))
                reg_no_position = str(name)[14:-1]
                reg_number = str(name)[15:-1]
        #       print(reg_no_position)
        #       print(reg_number)
        try:
            print(str(name))
        except:
            messagebox.showerror("Invalid", "Invalid registration nummber!!!")

        x1 = sheet.cell(row=int(reg_number), column=1).value
        x2 = sheet.cell(row=int(reg_number), column=2).value
        x3 = sheet.cell(row=int(reg_number), column=3).value
        x4 = sheet.cell(row=int(reg_number), column=4).value
        x5 = sheet.cell(row=int(reg_number), column=5).value
        x6 = sheet.cell(row=int(reg_number), column=6).value
        x7 = sheet.cell(row=int(reg_number), column=7).value
        x8 = sheet.cell(row=int(reg_number), column=8).value
        x9 = sheet.cell(row=int(reg_number), column=9).value
        x10 = sheet.cell(row=int(reg_number), column=10).value
        x11 = sheet.cell(row=int(reg_number), column=11).value
        x12 = sheet.cell(row=int(reg_number), column=12).value
        x13 = sheet.cell(row=int(reg_number), column=13).value
        x14 = sheet.cell(row=int(reg_number), column=14).value
        x15 = sheet.cell(row=int(reg_number), column=15).value
        x16 = sheet.cell(row=int(reg_number), column=16).value
        x17 = sheet.cell(row=int(reg_number), column=17).value
        x18 = sheet.cell(row=int(reg_number), column=18).value
```

```
    Registration.set(x1)
    Name.set(x2)
    Email.set(x3)

    if x4 == 'female':
       R2.select()
    else:
       R1.select()

    DOB.set(x5)
    Date.set(x6)
    Religion.set(x7)
    Mobile_no.set(x8)
    F_Name.set(x9)
    M_Name.set(x10)
    Father_Occupation.set(x11)
    Mother_Occupation.set(x12)
    Tenth_Marks.set(x13)
    Tenth_pass_year.set(x14)
    School_name.set(x15)
    Inter_Marks.set(x16)
    Inter_pass_year.set(x17)
    college_name.set(x18)

    img = (Image.open("Student Images/" + str(x1) + ".jpg"))
    resized_image = img.resize((190, 190))
    photo2 = ImageTk.PhotoImage(resized_image)
    lbl.configure(image=photo2)
    lbl.image = photo2

############################## update ################################
def Update():
   R1 = Registration.get()
   N1 = Name.get()
   C1 = Email.get()
   selection()
   G1 = gender
   D2 = DOB.get()
   D1 = Date.get()
   Re = Religion.get()
   S1 = Mobile_no.get()
   fathername = F_Name.get()
   mothername = M_Name.get()
   F1 = Father_Occupation.get()
   M1 = Mother_Occupation.get()
   T1 = Tenth_Marks.get()
   P1 = Tenth_pass_year.get()
   Scl = School_name.get()
   I1 = Inter_Marks.get()
   P2 = Inter_pass_year.get()
   Clg = college_name.get()
```

```
file = openpyxl.load_workbook("Student_data.xlsx")
sheet = file.active
for row in sheet.rows:
    if row[0].value == R1:
        name = row[0]
        reg_no_position = str(name)[14:-1]
        reg_number = str(name)[15:-1]

#  sheet.cell(column=1,row=int(reg_number),value=R1)
sheet.cell(column=2, row=int(reg_number), value=N1)
sheet.cell(column=3, row=int(reg_number), value=C1)
sheet.cell(column=4, row=int(reg_number), value=G1)
sheet.cell(column=5, row=int(reg_number), value=D2)
sheet.cell(column=6, row=int(reg_number), value=D1)
sheet.cell(column=7, row=int(reg_number), value=Re)
sheet.cell(column=8, row=int(reg_number), value=S1)
sheet.cell(column=9, row=int(reg_number), value=fathername)
sheet.cell(column=10, row=int(reg_number), value=mothername)
sheet.cell(column=11, row=int(reg_number), value=F1)
sheet.cell(column=12, row=int(reg_number), value=M1)
sheet.cell(column=13, row=int(reg_number), value=T1)
sheet.cell(column=14, row=int(reg_number), value=P1)
sheet.cell(column=15, row=int(reg_number), value=Scl)
sheet.cell(column=16, row=int(reg_number), value=I1)
sheet.cell(column=17, row=int(reg_number), value=P2)
sheet.cell(column=18, row=int(reg_number), value=Clg)

file.save(r'Student_data.xlsx')

try:
        img.save("Student Images/" + str(R1) + ".jpg")

    except:
        pass
    messagebox.showinfo("Update", "Update Successfully!!")
    Clear()

# upload images
def showimage():
    global filename
    global img
    filename = filedialog.askopenfilename(initialdir=os.getcwd, title="Select image file",
                                          filetype=(("Jpg File", "*.jpg"),
                                                    ("PNG File", "*.png"),
                                                    ("All Files", "*.txt")))
    img = (Image.open(filename))
    resized_image = img.resize((190, 190))
    photo2 = ImageTk.PhotoImage(resized_image)
    lbl.configure(image=photo2)
    lbl.image = photo2
```

```
##########################Registration no automatic function#####################
def registration_no():
    file = openpyxl.load_workbook('Student_data.xlsx')
    sheet = file.active
    row = sheet.max_row

    max_row_value = sheet.cell(row=row, column=1).value

    try:
        Registration.set(max_row_value + 1)
    except:
        Registration.set("1")

################################ Clear ###################################
def Clear():
    global img
    global filename

    Name.set("")
    DOB.set("")
    Religion.set("")
    Mobile_no.set("")
    F_Name.set("")
    M_Name.set("")
    Father_Occupation.set("")
    Mother_Occupation.set("")
    Email.set("")
    Tenth_Marks.set("")
    Tenth_pass_year.set("")
    School_name.set("")
    Inter_Marks.set("")
    Inter_pass_year.set("")
    college_name.set("")
    registration_no()

    savebutton.configure(state="normal")

    # img1 = PhotoImage(file="Images/upload photo.png")
    lbl.configure(image=img1)
    lbl.image = img1

    img = ""
```

```
############################## Save ##########################
def save():
    R1 = Registration.get()
    N1 = Name.get()
    C1 = Email.get()
    try:
        G1 = gender
    except:
        messagebox.showerror("Error", "Select Gender")

    D2 = DOB.get()
    D1 = Date.get()
    Re = Religion.get()
    S1 = Mobile_no.get()
    fathername = F_Name.get()
    mothername = M_Name.get()
    F1 = Father_Occupation.get()
    M1 = Mother_Occupation.get()
    T1 = Tenth_Marks.get()
    P1 = Tenth_pass_year.get()
    Scl = School_name.get()
    I1 = Inter_Marks.get()
    P2 = Inter_pass_year.get()
    Clg = college_name.get()

    if N1 == "" or C1 == "Select Class" or D2 == "" or Re == "" or S1 == "" or fathername == "" or mothername == "" or F1 == "" or M1 == "" or T1 == "" or P1 == "" or Scl == "" or P2 == "" or I1 == "" or Clg == "":
        messagebox.showerror("Error", "Few Data is Missing")
    else:
        file = openpyxl.load_workbook('Student_data.xlsx')
        sheet = file.active
        sheet.cell(column=1, row=sheet.max_row + 1, value=R1)
        sheet.cell(column=2, row=sheet.max_row, value=N1)
        sheet.cell(column=3, row=sheet.max_row, value=C1)
        sheet.cell(column=4, row=sheet.max_row, value=G1)
        sheet.cell(column=5, row=sheet.max_row, value=D2)
        sheet.cell(column=6, row=sheet.max_row, value=D1)
        sheet.cell(column=7, row=sheet.max_row, value=Re)
        sheet.cell(column=8, row=sheet.max_row, value=S1)
        sheet.cell(column=9, row=sheet.max_row, value=fathername)
        sheet.cell(column=10, row=sheet.max_row, value=mothername)
        sheet.cell(column=11, row=sheet.max_row, value=F1)
        sheet.cell(column=12, row=sheet.max_row, value=M1)
        sheet.cell(column=13, row=sheet.max_row, value=T1)
        sheet.cell(column=14, row=sheet.max_row, value=P1)
        sheet.cell(column=15, row=sheet.max_row, value=Scl)
        sheet.cell(column=16, row=sheet.max_row, value=I1)
        sheet.cell(column=17, row=sheet.max_row, value=P2)
        sheet.cell(column=18, row=sheet.max_row, value=Clg)
        file.save(r'Student_data.xlsx')
```

```
    try:
        img.save("Student Images/" + str(R1) + ".jpg")
    except:
        messagebox.showinfo("Info", "Profile Picture is not available !")

    messagebox.showinfo("Info Message", message="Information Saved Successfully")

    Clear()
    registration_no()

  ############################### gender function ######################

def selection():
  global gender
  value = radio.get()
  if value == 1:
    gender = "Male"
  else:
    gender = "Female"

# top frames
Label(root, text="Email: satwikuppada@gmail.com", width=10, height=3, bg="#2b7a90", anchor="e",
    font="arial 14 bold").pack(side=TOP, fill=X)
Label(root, text="STUDENT REGISTRATION PORTAL", width=10, height=2, bg="#539baf", fg="#fff",
    font="arial 20 bold").pack(
  side=TOP, fill=X)

# search box to update
Search = StringVar()
customtkinter.CTkEntry(root, textvariable=Search,text_color="black",
border_color="#2b7a90",width=240,border_width=3, font=("arial"
,20),fg_color="white",bg_color="#539baf").place(x=1500, y=88)

# imageicon3 = PhotoImage(file="Images/search.png")
Home = customtkinter.CTkButton(root,text="", compound=LEFT, image=home_icon, width=65,
bg_color="#539baf",fg_color="#539baf", command=home_page, height=60)
Home.place(x=10, y=74)

Srch = customtkinter.CTkButton(root, text="Search", compound=LEFT, image=imageicon3, width=65,
bg_color="#539baf",fg_color="#539baf", font=("arial", 13 ,"bold"),
        command=search)
Srch.place(x=1750, y=84)
# imageicon4 = PhotoImage(file="Images/Layer 4.png")
update_button = customtkinter.CTkButton(root,text="", image=imageicon4, fg_color="#539baf",
command=Update,bg_color="#539baf",width=65)
update_button.place(x=110, y=80)
```

```
# Registation and date
customtkinter.CTkLabel(root, text="Registration No:", font=("arial" ,12,"bold"),
fg_color=background).place(x=60, y=150)
customtkinter.CTkLabel(root, text="Date:", font=("arial ",12,"bold"), fg_color=background,).place(x=1650,
y=150)

Registration = IntVar()
Date = StringVar()

# Reg_no
reg_entry = customtkinter.CTkEntry(root, textvariable=Registration, width=150,
font=("arial",12),fg_color="white",text_color="black")
reg_entry.place(x=160, y=152)

registration_no()

# date
today = date.today()
d1 = today.strftime("%d/%m/%Y")
date_entry = customtkinter.CTkEntry(root, textvariable=Date, width=150, font=("arial",
12),fg_color="white",text_color="black")
date_entry.place(x=1700, y=152)

Date.set(d1)

# Student Details
obj = LabelFrame(root, text="Student's Details", font=20, bd=2, width=1500, bg=framebg, height=250,
relief=GROOVE)
obj.place(x=60, y=200)

Label(obj, text="Full Name:", font="arial 16", bg=framebg, fg=framefg).place(x=30, y=50)
Label(obj, text="Date of Birth:", font="arial 16", bg=framebg, fg=framefg).place(x=30, y=100)
Label(obj, text="Gender:", font="arial 16", bg=framebg, fg=framefg).place(x=30, y=150)

Label(obj, text="Email:", font="arial 16", bg=framebg, fg=framefg).place(x=1030, y=50)
Label(obj, text="Religion:", font="arial 16", bg=framebg, fg=framefg).place(x=1030, y=100)
Label(obj, text="Mobile.No:", font="arial 16", bg=framebg, fg=framefg).place(x=1030, y=150)

# entries
Name = StringVar()
name_entry = customtkinter.CTkEntry(obj, textvariable=Name, width=200, font=("arial ", 16),
text_color="black",
                            fg_color="white")
name_entry.place(x=240, y=50)

DOB = StringVar()
dob_entry = customtkinter.CTkEntry(obj, textvariable=DOB, width=200, font=("arial ", 16), text_color="black",
                           fg_color="white")
dob_entry.place(x=240, y=100)
```

```
radio = IntVar()
R1 = Radiobutton(obj, text="Male", variable=radio, value=1, bg=framebg, fg=framefg, command=selection,
            font="arial 16")
R1.place(x=240, y=150)

R2 = Radiobutton(obj, text="Female", variable=radio, value=2, bg=framebg, fg=framefg, command=selection,
            font="arial 16")
R2.place(x=330, y=150)

Religion = StringVar()
rel_entry = customtkinter.CTkEntry(obj, textvariable=Religion, width=200, font=("arial ", 16), text_color="black",
                           fg_color="white")
rel_entry.place(x=1230, y=100)

Mobile_no = StringVar()
skill_entry = customtkinter.CTkEntry(obj, textvariable=Mobile_no, width=200, font=("arial ", 16),
                             text_color="black", fg_color="white")
skill_entry.place(x=1230, y=150)

Email = StringVar()
Email_entry = customtkinter.CTkEntry(obj, textvariable=Email, width=200, font=("arial ", 16),
text_color="black",
                             fg_color="white")
Email_entry.place(x=1230, y=50)
# Class = customtkinter.CTkComboBox(obj, values=['1', '2', '3', '4', '5', '7', '8', '9', '10', '11', '12'], font=("Roboto",
16), width=200,text_color="black",fg_color="white",bg_color="black"
#  ,state='r')
# Class.place(x=1230, y=50)
# Class.set("Select Class")

# Parent Details
obj2 = LabelFrame(root, text="Parent's Details", font="48", bd=2, width=1500, bg=framebg, height=220,
relief=GROOVE)
obj2.place(x=60, y=470)

Label(obj2, text="Father's Name:", font="arial 16", bg=framebg, fg=framefg).place(x=30, y=50)
Label(obj2, text="Occupation:", font="arial 16", bg=framebg, fg=framefg).place(x=30, y=100)

F_Name = StringVar()
f_entry = customtkinter.CTkEntry(obj2, textvariable=F_Name, width=200, font=("arial ", 16), text_color="black",
                           fg_color="white")
f_entry.place(x=240, y=50)

Father_Occupation = StringVar()
FO_Entry = customtkinter.CTkEntry(obj2, textvariable=Father_Occupation, width=200, font=("arial ", 16),
                             text_color="black", fg_color="white")
FO_Entry.place(x=240, y=100)

Label(obj2, text="Mother's Name:", font="arial 16", bg=framebg, fg=framefg).place(x=1030, y=50)
Label(obj2, text="Occupation:", font="arial 16", bg=framebg, fg=framefg).place(x=1030, y=100)
m_entry.place(x=1230, y=50)
```

```
M_Name = StringVar()
m_entry = customtkinter.CTkEntry(obj2, textvariable=M_Name, width=200, font=("arial ", 16), text_color="black",
                         fg_color="white")
m_entry.place(x=1230, y=50)

Mother_Occupation = StringVar()
MO_Entry = customtkinter.CTkEntry(obj2, textvariable=Mother_Occupation, width=200, font=("arial ", 16),
                          text_color="black", fg_color="white")
MO_Entry.place(x=1230, y=100)

# image
f = Frame(root, bd=3, bg="black", width=200, height=200, relief=GROOVE)
f.place(x=1650, y=200)

obj3 = LabelFrame(root, text="Acadamic Details", font="48", bd=2, width=1500, bg=framebg, height=220,
relief=GROOVE)
obj3.place(x=60, y=710)

Label(obj3, text="Tenth Marks:", font="arial 16", bg=framebg, fg=framefg).place(x=30, y=50)
Label(obj3, text="Tenth Pass Out Year:", font="arial 16", bg=framebg, fg=framefg).place(x=30, y=100)
Label(obj3, text="School Name:", font="arial 16", bg=framebg, fg=framefg).place(x=30, y=150)

Label(obj3, text="Inter Marks:", font="arial 16", bg=framebg, fg=framefg).place(x=1030, y=50)
Label(obj3, text="Inter Pass Out Year:", font="arial 16", bg=framebg, fg=framefg).place(x=1030, y=100)
Label(obj3, text="College Name:", font="arial 16", bg=framebg, fg=framefg).place(x=1030, y=150)

# entries
Tenth_Marks = StringVar()
Tenth_Marks_entry = customtkinter.CTkEntry(obj3, width=200, textvariable=Tenth_Marks, fg_color="white",
                                   text_color="black", font=("arial", 16))
Tenth_Marks_entry.place(x=240, y=50)

Tenth_pass_year = StringVar()
Tenth_pass_year_entry = customtkinter.CTkEntry(obj3, textvariable=Tenth_pass_year, width=200, font=("arial ",
16),
                                       text_color="black", fg_color="white")
Tenth_pass_year_entry.place(x=240, y=100)

School_name = StringVar()
School_name_entry = customtkinter.CTkEntry(obj3, textvariable=School_name, width=200, font=("arial ", 16),
                                   text_color="black", fg_color="white")
School_name_entry.place(x=240, y=150)

Inter_Marks = StringVar()
Inter_Marks_entry = customtkinter.CTkEntry(obj3, textvariable=Inter_Marks, width=200, font=("arial ", 16),
                                   text_color="black", fg_color="white")
Inter_Marks_entry.place(x=1230, y=50)

Inter_pass_year = StringVar()
Inter_pass_year_entry = customtkinter.CTkEntry(obj3, textvariable=Inter_pass_year, width=200, font=("arial ", 16),
                                       text_color="black", fg_color="white")
Inter_pass_year_entry.place(x=1230, y=100)
```

```
college_name = StringVar()
   college_name_entry = customtkinter.CTkEntry(obj3, textvariable=college_name, width=200, font=("arial ", 16),
                                    text_color="black", fg_color="white")
   college_name_entry.place(x=1230, y=150)

   # img = PhotoImage(file="Images/upload photo.png")
   lbl = Label(f, bg="black", image=img)
   lbl.place(x=0, y=0)

   # button
   # Button(root,text="Take Picture",width=10,height=2,font=("arial 12 bold"),bg="light
yellow").place(x=977,y=370)
   customtkinter.CTkButton(root, text="Upload", width=180, text_color="black", height=50, font=("arial", 16,
"bold"),
                           fg_color="pink", bg_color="#282631", command=showimage).place(x=1650, y=450)

   savebutton = customtkinter.CTkButton(root, text="Save", width=180, text_color="black", height=50,
                                font=("arial", 16, "bold"), fg_color="light green", bg_color="#282631",
                                command=save)
   savebutton.place(x=1650, y=530)

   customtkinter.CTkButton(root, text="Reset", width=180, text_color="black", height=50, font=("arial", 16,
"bold"),
                           fg_color="orange", bg_color="#282631", command=Clear).place(x=1650, y=610)

   customtkinter.CTkButton(root, text="Exit", width=180, text_color="black", height=50, font=("arial", 16,
"bold"),
                           fg_color="crimson", bg_color="#282631", cursor="pirate", command=Exit).place(x=1650,
y=690)

   root.mainloop()


cover_page()
```

5.OUTPUT

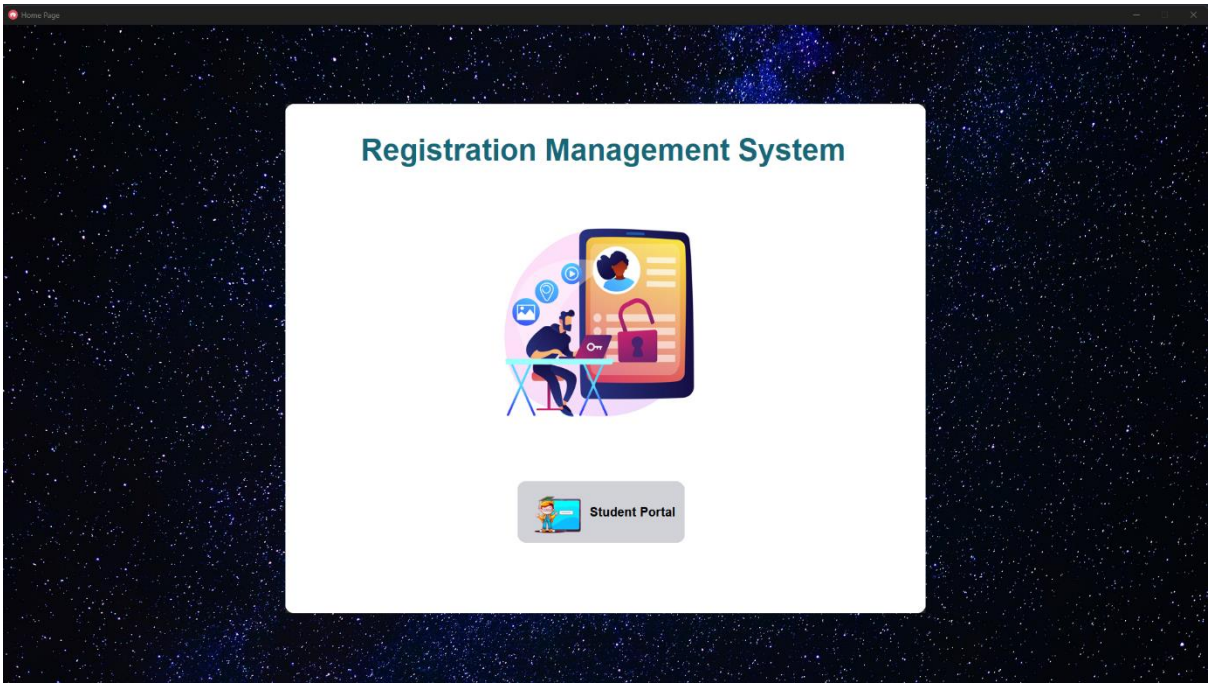


Figure 1: Home page

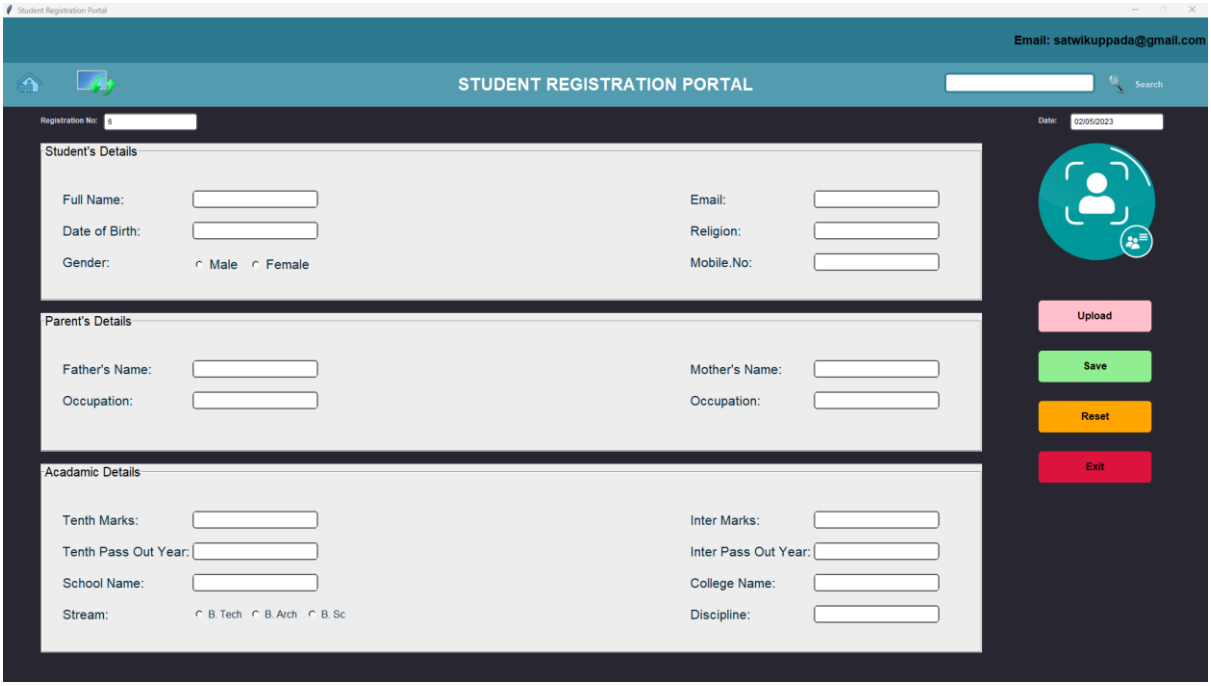


Figure 2: Student Portal Page

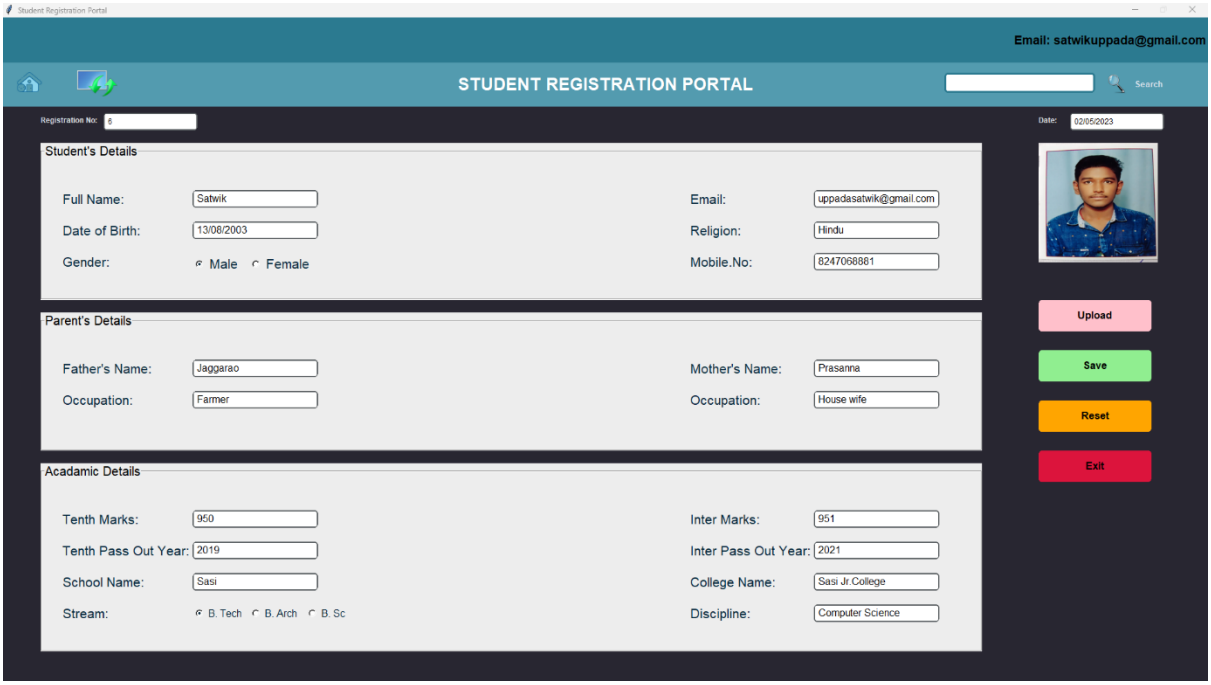


Figure 3: Filling Details in Entry fields

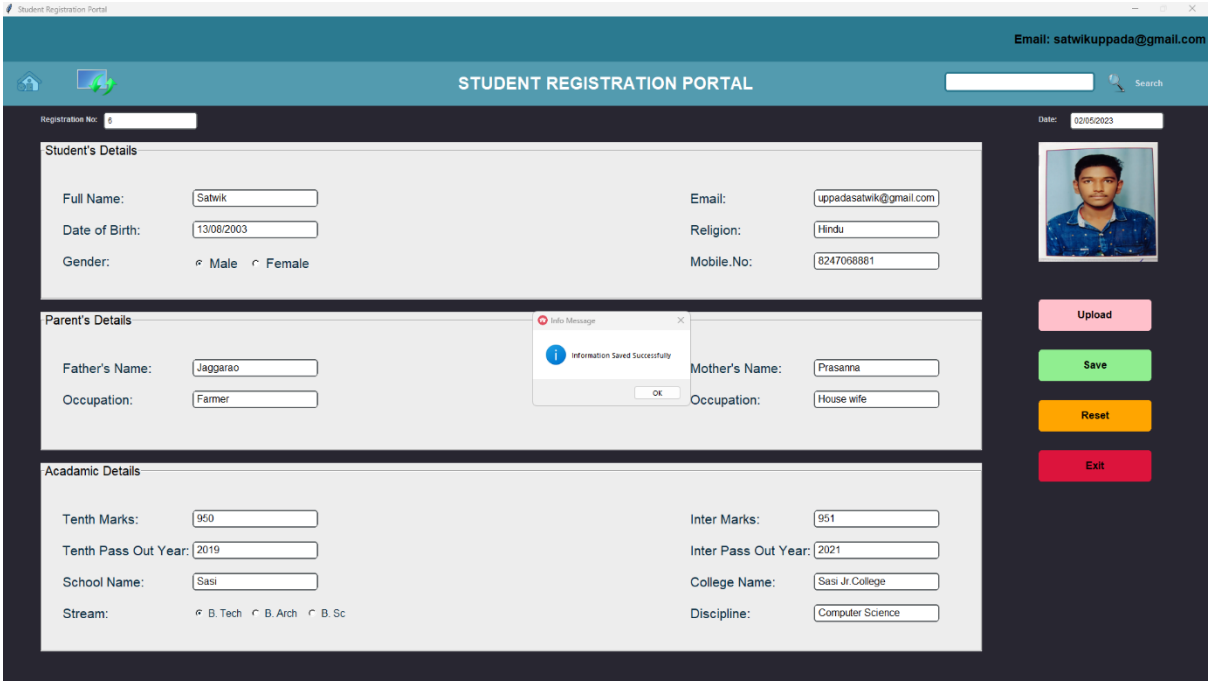


Figure 4: Pop-Up Message after saving data

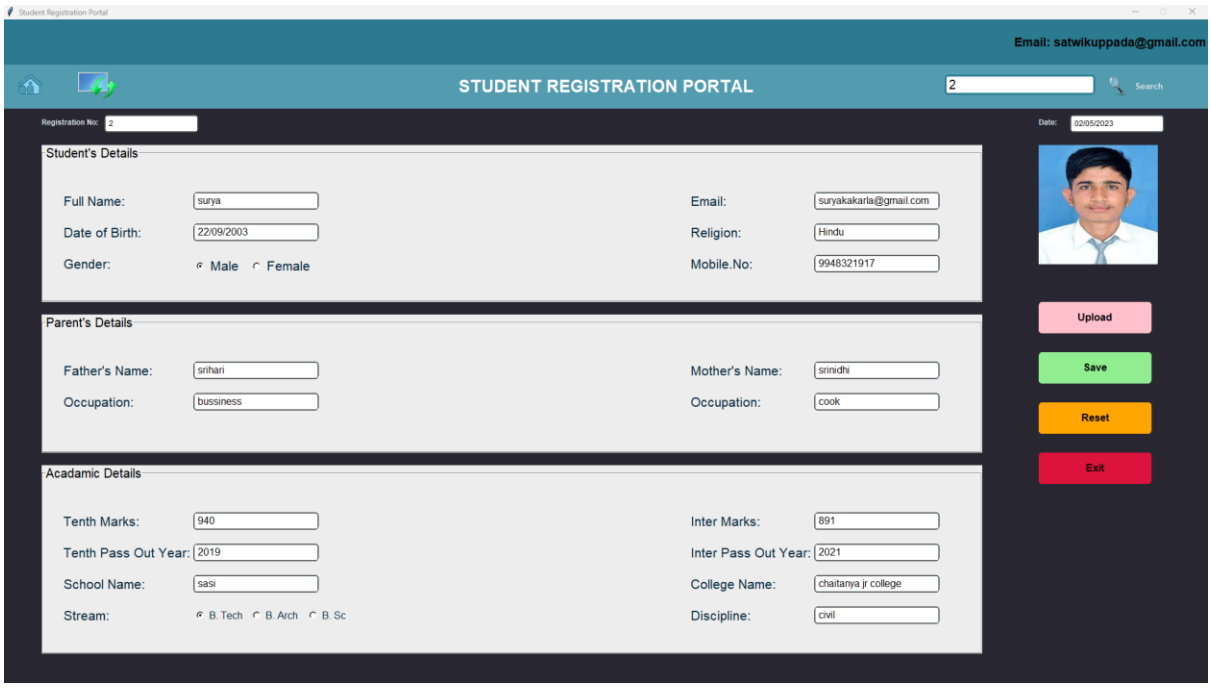


Figure 5: Searched data using search bar

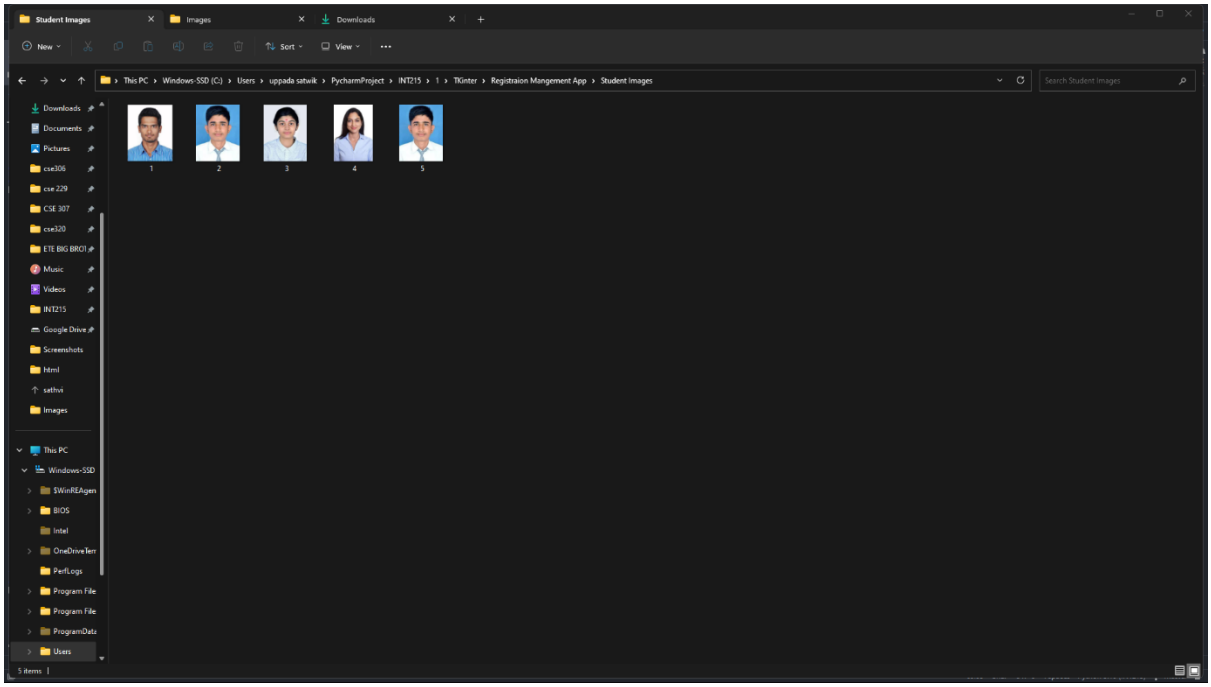
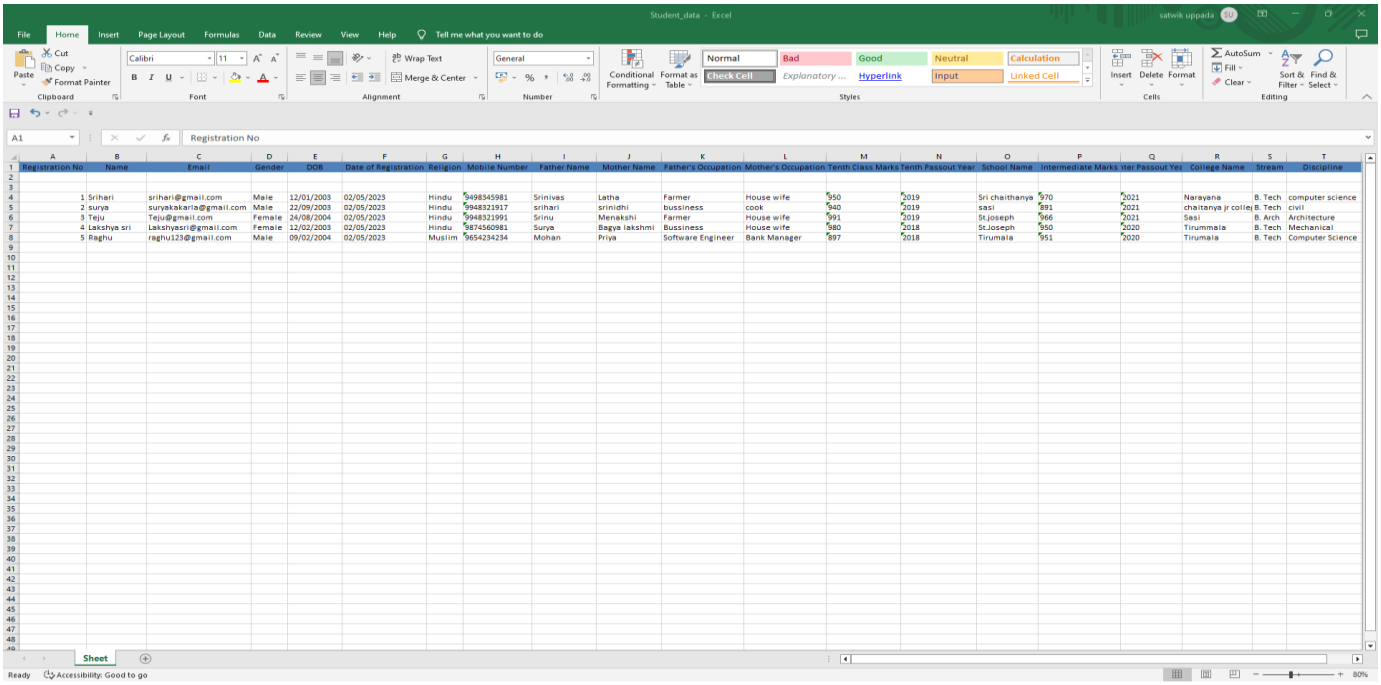


Figure 6: Uploaded images saved in this folder (Student Images)



Registration No	Name	Email	Gender	DOB	Date of Registration	Religion	Mobile Number	Father Name	Mother Name	Father's Occupation	Mother's Occupation	Tenth Class Marks	Tenth Passout Year	School Name	Intermediate Marks	ıter Passout Yea	College Name	Stream	Discipline
1	Srihari	srihari@gmail.com	Male	12/01/2003	02/05/2023	Hindu	9498345981	Srinivas	Latha	Farmer	House wife	950	2019	Sri chaithanya	970	2021	Narayana	B. Tech	computer science
2	surya	suryakakarla@gmail.com	Male	22/09/2003	02/05/2023	Hindu	9948321917	srihari	srinidhi	bussiness	cook	940	2019	sasi	891	2021	chaitanya jr colle	B. Tech	civil
3	Teju	Teju@gmail.com	Female	24/08/2004	02/05/2023	Hindu	9948321991	Srinu	Menakshi	Farmer	House wife	991	2019	St.joseph	966	2021	Sasi	B. Arch	Architecture
4	Lakshya sri	Lakshyasri@gmail.com	Female	12/02/2003	02/05/2023	Hindu	9874560981	Surya	Bagya lakshmi	Bussiness	House wife	980	2018	St.joseph	950	2020	Tirummala	B. Tech	Mechanical
5	Raghu	raghu123@gmail.com	Male	09/02/2004	02/05/2023	Muslim	9654234234	Mohan	Priya	Software Engineer	Bank Manager	897	2018	Tirumala	951	2020	Tirumala	B. Tech	Computer Science

Figure 7: Data stored in Excel file (Student_data.xlsx)

6.FUTURE SCOPE

The Student Registration Management System is a powerful tool that can streamline the student registration process. As technology continues to evolve, there are several areas where this system can be enhanced and improved. Here are some potential future scope areas for this project:

1. **Integration with Student Information Systems (SIS)**

   The Student Registration Management System can be integrated with an institution's existing Student Information System (SIS) to further streamline the registration process. This integration can allow for seamless data exchange between the two systems, reducing the time and effort required for data entry and minimizing the chances of errors.

2. **Online registration**

   With the increasing trend of online education and remote learning, the system can be modified to support online registration. This functionality can allow students to register for courses and programs online from anywhere, making the process more convenient and accessible.

3. **Student dashboard**

   A student dashboard can be added to the system, providing students with a central location to view their registration details, course schedules, grades, and other academic information. This feature can improve the student experience by allowing them to easily access their academic information and track their progress.

4. **Automated notifications**

   The system can be enhanced to provide automated notifications to students and administrators regarding registration deadlines, course changes, and other important updates. This feature can reduce the chances of missed deadlines and ensure that everyone is up to date with the latest information.

5. **Analytics and reporting**

   The system can be modified to include analytics and reporting capabilities, providing administrators with insights into student registration trends, course popularity, and other useful information. This data can be used to inform decision-making and improve the overall registration process.

6. **Integration with payment systems**

Integration with payment systems can be added to allow students to pay for courses and programs directly through the Student Registration Management System. This feature can simplify the payment process and reduce the chances of errors.

In conclusion, the Student Registration Management System has immense potential for future development and improvement. As technology continues to evolve, there will be several opportunities to enhance the system's functionality and improve the student registration experience. By embracing these future scope areas, institutions can continue to streamline their registration processes and provide a better experience for students and administrators alike.

7.CONCLUSION

In conclusion, the Student Registration Management System is a useful software application that can significantly improve the efficiency of student registration in educational institutions. With its user-friendly interface and advanced features, it simplifies the process of recording and managing student information, making it easier for administrators to access, update and search for records.

Through this capstone project, I have explored the design and implementation of the Student Registration Management System, demonstrating its effectiveness in streamlining the registration process for students and administrators alike. Moreover, the project's future scope shows that there is a lot of potential for expanding the system's functionality and improving its performance, such as adding payment gateways, integrating attendance systems, or implementing AI-based analytics for student performance monitoring.

Overall, the Student Registration Management System is a vital tool for educational institutions looking to automate their registration process and improve their efficiency. It not only simplifies administrative tasks but also saves time and resources, enabling institutions to focus more on education and student success.