PROJECT

**MESS RECORD TRACKER BY EXCEL FILE HANDLING AND EMAIL AUTOMATION USING OPENPYXL AND SMTPLIB**

****

**DEPARTMENT OF CSE**

**DR. B.R. AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY**

**JALANDHAR**

**(A2C) – DATA SCIENCE AND ENGINEERING**

**Navroop Kaur 23117025**

**INTRODUCTION**

In today's fast-paced world, the need for efficient and automated record-keeping systems is paramount, especially in environments like educational institutions, hostels, and workplaces where attendance tracking is crucial. The efficient management of attendance records not only streamlines administrative processes but also ensures accuracy, transparency, and accountability.

The "Mess Record Tracker by Excel File Handling and Email Automation using OpenPyXL and SMTPlib" project addresses this need by offering a comprehensive solution that leverages the power of Python programming along with the capabilities of OpenPyXL and SMTPlib libraries. This project aims to revolutionize attendance management by providing a seamless and automated system that simplifies the task of recording attendance and enhances communication channels through email automation.

**Need of the project:**

Attendance tracking in environments like hostels, especially in educational institutions such as the National Institute of Technology (NIT) Jalandhar, is a critical aspect of administrative operations. Traditional methods of attendance management involving manual entry and maintenance of records are not only time-consuming but also prone to errors and inaccuracies. Additionally, the lack of automation often leads to inefficiencies in communication between stakeholders, such as hostel administrators, mess staff, and students.

**The Mess Record Tracker project addresses these needs by offering the following key features:**

1. **AUTOMATION:** By automating the process of recording attendance, the project reduces the manual effort required by hostel staff and administrators. This automation ensures that accurate attendance data is consistently captured without the risk of human error.

2. **EFFICIENCY** With the integration of Excel file handling capabilities, the project facilitates efficient storage, organization, and retrieval of attendance records. Hostel administrators can easily access and analyze attendance data, enabling them to make informed decisions and identify trends.

3. **TRANSPARENCY:** The project promotes transparency in attendance management by providing a centralized platform where all stakeholders can access real-time attendance information. This transparency fosters trust and accountability among hostel staff, mess workers, and students.

4. **COMMUNICATION:** Through email automation using the SMTPlib library, the project enhances communication channels between stakeholders. Hostel administrators can automatically generate and send attendance reports to relevant parties, keeping them informed about attendance patterns and any notable updates or announcements.

**WHY PYTHON?**

Python, with its simplicity, versatility, and robust ecosystem of libraries, is the ideal programming language for implementing this project. Here are some reasons why Python is chosen:

1. **Ease of Learning and Use:** Python's clean and readable syntax makes it accessible to developers of all skill levels. This ease of learning and use accelerates the development process and allows for rapid prototyping and iteration.

2. **Rich Ecosystem**: Python boasts a vast array of libraries and frameworks that provide solutions for various tasks, from data manipulation and analysis to web development and automation. OpenPyXL and SMTPlib are prime examples of libraries that seamlessly integrate with Python, enabling efficient Excel file handling and email automation, respectively.

3. **Community Support**: Python enjoys robust community support, with a vast community of developers contributing to its growth and evolution. This active community ensures that developers have access to resources, documentation, and community forums for assistance and collaboration.

4. **Cross-Platform Compatibility**: Python is a cross-platform language, meaning that code written in Python can run on various operating systems without modification. This cross-platform compatibility ensures that the Mess Record Tracker project can be deployed seamlessly across different environments, including Windows, macOS, and Linux.

**Impact on NIT Jalandhar Hostels and Beyond:**

The implementation of the Mess Record Tracker project at NIT Jalandhar hostels holds the potential to make a significant impact on attendance management and administrative efficiency. By streamlining attendance tracking processes, the project frees up valuable time and resources for hostel administrators, allowing them to focus on more strategic tasks and initiatives.

Furthermore, the project's transparency and communication features foster a culture of accountability and engagement among hostel staff, mess workers, and students. By providing stakeholders with access to real-time attendance data and automated communication channels, the project enhances collaboration and ensures that everyone remains informed and aligned.

Beyond NIT Jalandhar, the Mess Record Tracker project serves as a scalable solution that can be implemented in hostels and similar environments across educational institutions and workplaces. Its adaptable design and reliance on open-source technologies make it accessible to a wide range of organizations seeking to improve their attendance management processes.

**Challenges Faced During Project Development:**

While the Mess Record Tracker project holds immense promise in revolutionizing attendance management, its development journey was not without its share of challenges. Overcoming these hurdles required creativity, perseverance, and collaboration among the project team members. Below are some of the key difficulties encountered during the making of this project:

1. **Complexity of Excel File Handling**: One of the primary challenges faced during the project development was the complexity associated with handling Excel files programmatically using the OpenPyXL library. Working with large datasets and implementing features such as data validation, conditional formatting, and formula calculations within Excel required a deep understanding of the library's functionalities and careful consideration of performance optimization techniques.

2. **Integration of Email Automation**: Integrating email automation functionality using the SMTPlib library posed another significant challenge. Configuring SMTP servers, handling email authentication, and ensuring the security of email communications were intricate tasks that required meticulous attention to detail. Additionally, ensuring compatibility with various email service providers and handling potential issues such as spam filters and email delivery failures added complexity to the implementation.

3. **Testing and Debugging**: Testing and debugging the Mess Record Tracker project to ensure its reliability, robustness, and compatibility across different environments proved to be a time-consuming endeavour. Identifying and addressing edge cases, handling exceptions gracefully, and conducting thorough regression testing were critical steps in ensuring the quality and stability of the application. Additionally, debugging issues related to interoperability between Python, OpenPyXL, SMTPlib, and other dependencies required careful troubleshooting and collaboration among developers.

4. **Deployment and Maintenance**: Deploying the Mess Record Tracker project in production environments and ensuring its smooth operation posed challenges related to deployment configurations, system dependencies, and ongoing maintenance. Automating deployment processes, documenting installation instructions, and establishing procedures for monitoring, updates, and troubleshooting were essential aspects of ensuring the project's long-term sustainability and scalability.

Despite these challenges, the dedication and expertise of the project team members enabled the successful development and deployment of the Mess Record Tracker project. By overcoming these difficulties, the project not only demonstrates the capabilities of Python and related libraries but also underscores the importance of collaboration, innovation, and perseverance in tackling complex real-world problems.

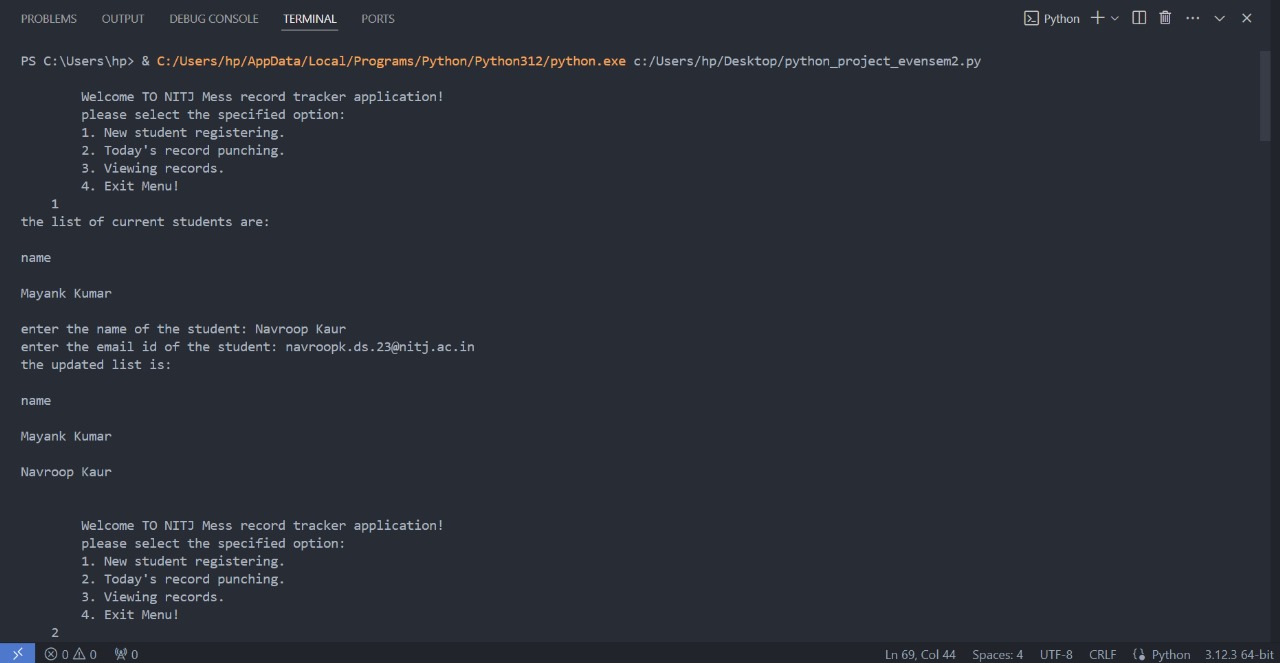
**Conclusion:**

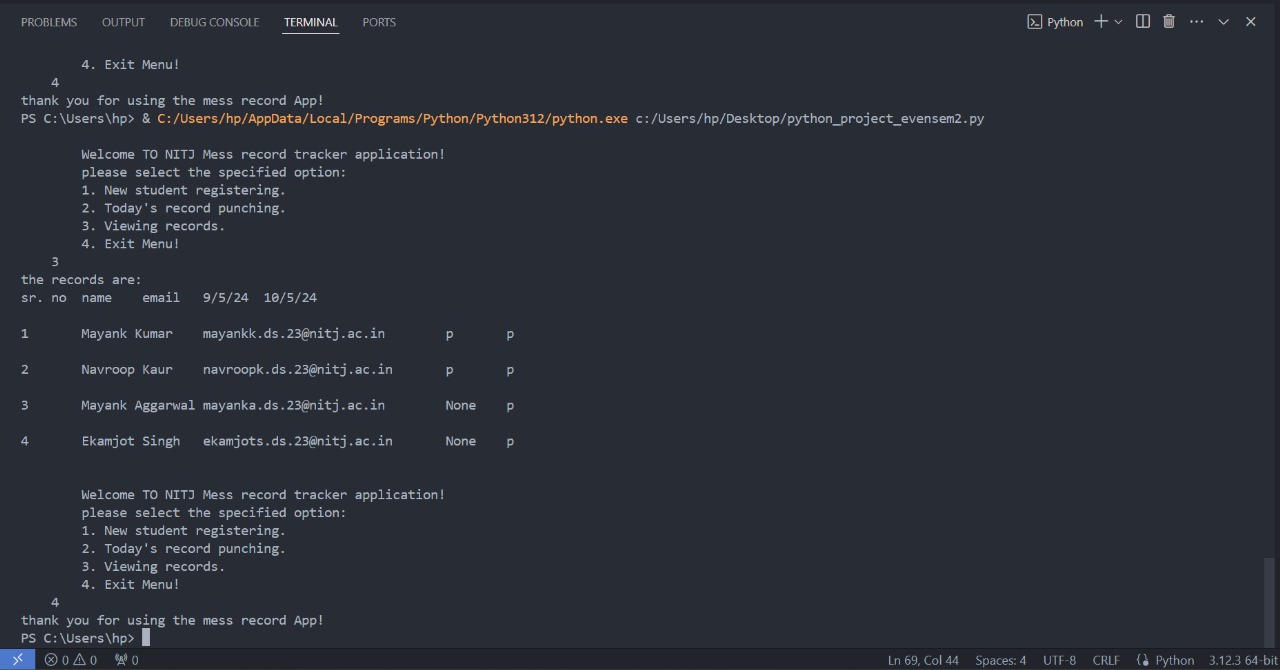
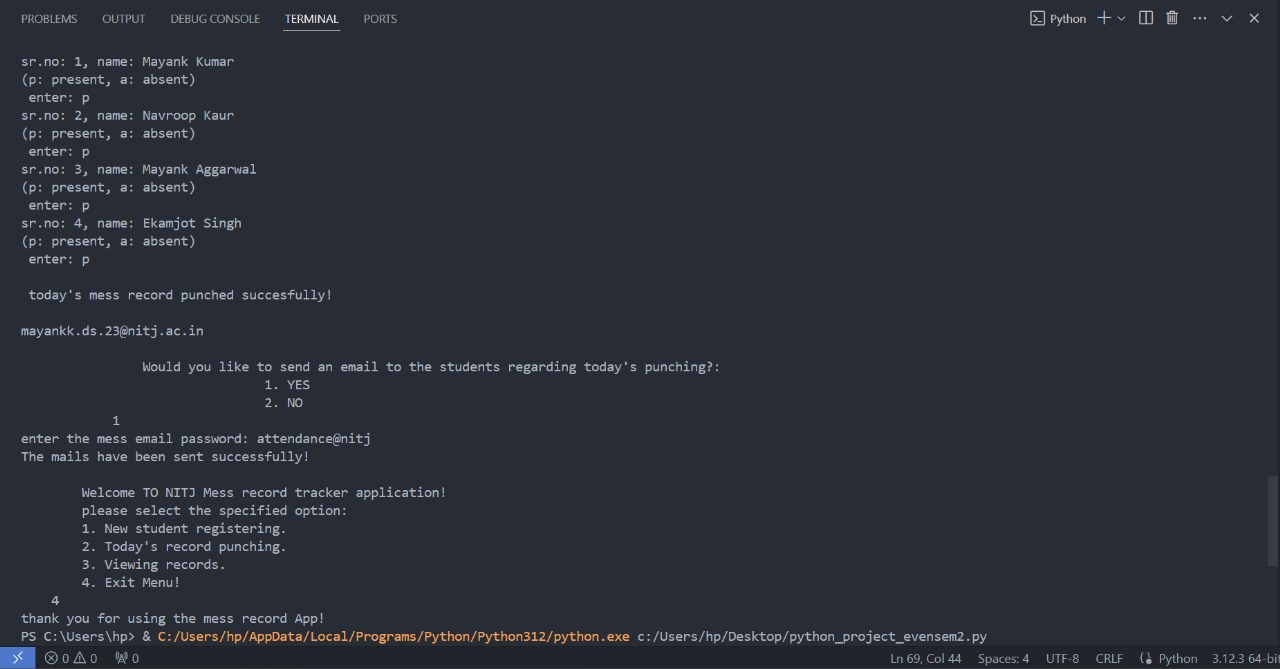
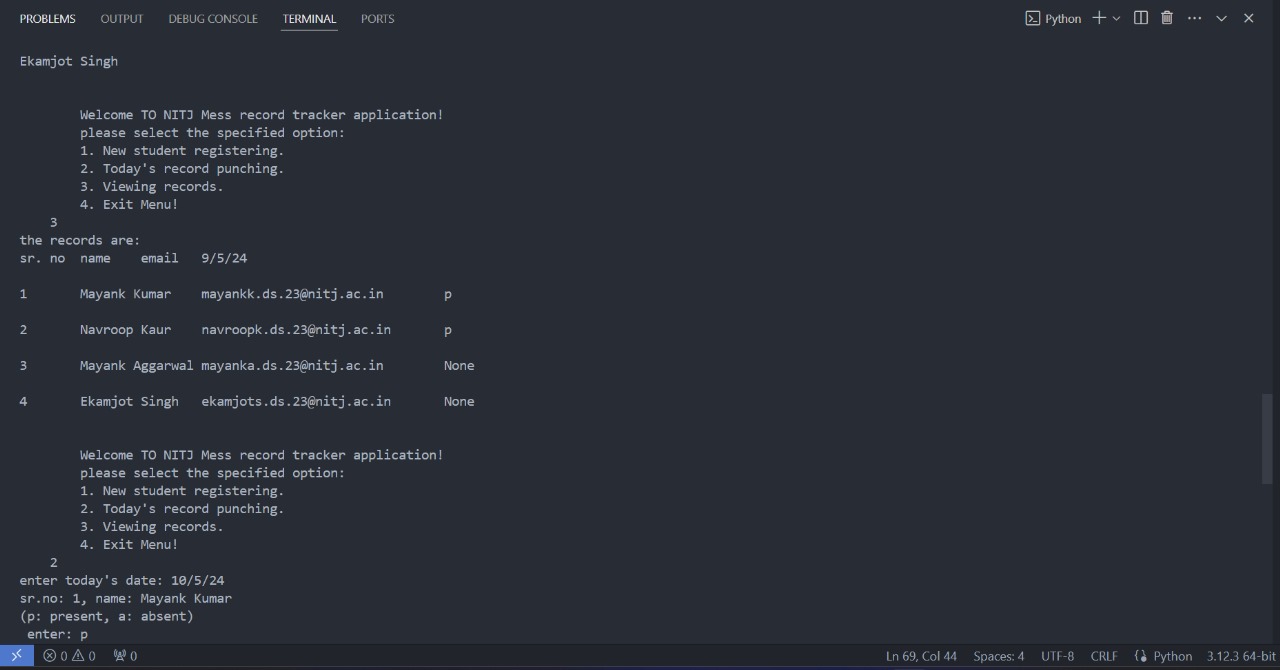
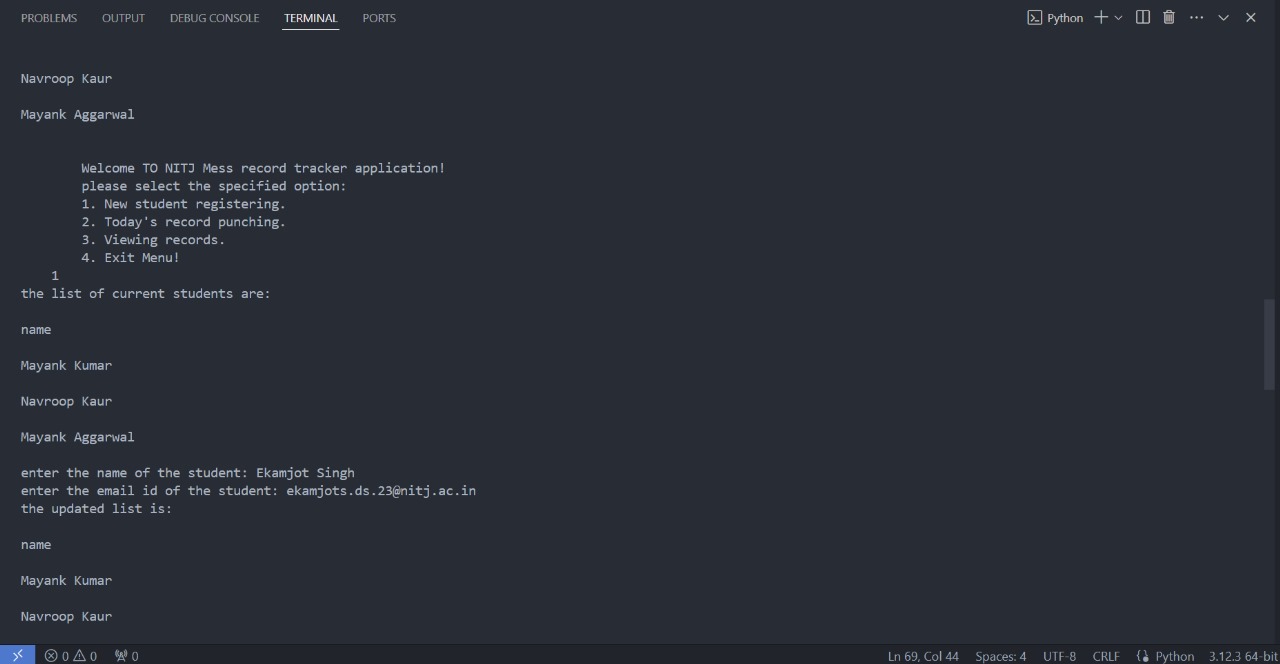
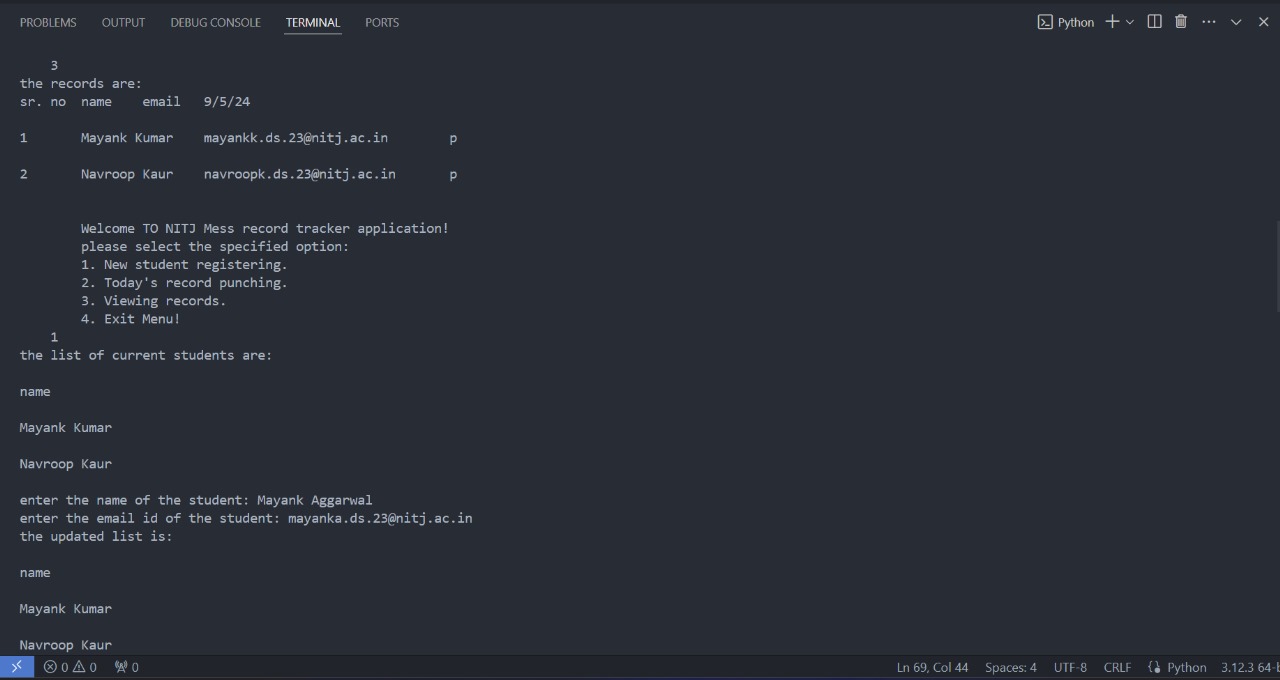
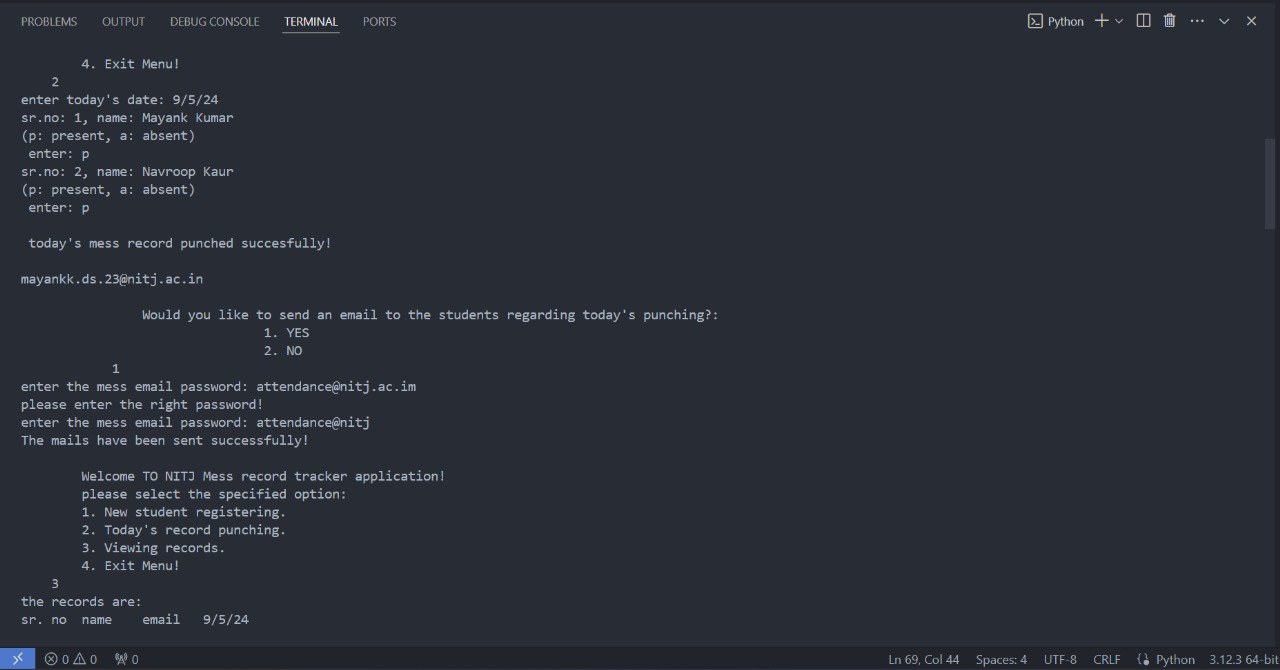
In conclusion, the Mess Record Tracker project represents a significant step forward in attendance management through its innovative use of Python, OpenPyXL, and SMTPlib. By addressing the needs of hostel administration and leveraging the power of automation and communication, the project promises to streamline operations, enhance transparency, and make a lasting impact on NIT Jalandhar hostels and beyond.

**PYTHON CODE**

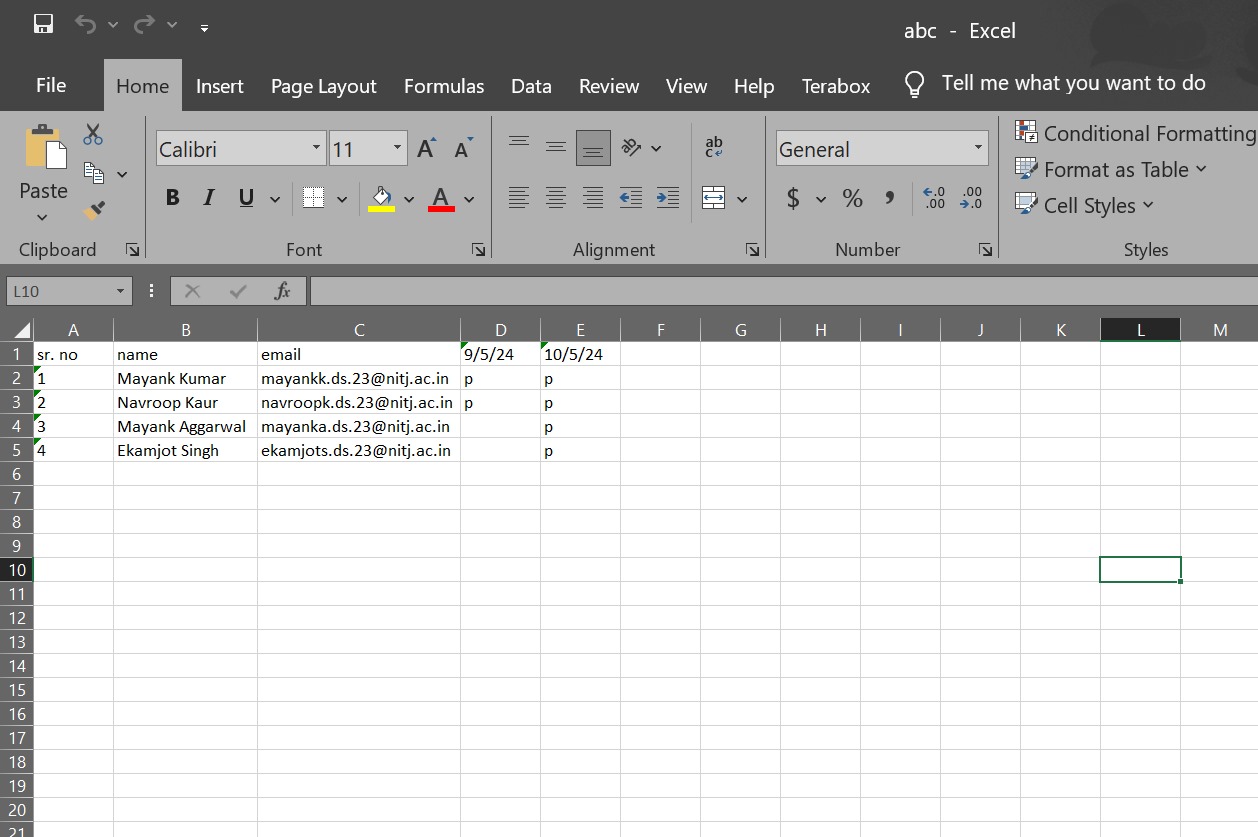
|  |
| --- |
| import smtplib  import getpass  from openpyxl import load\_workbook,workbook  from openpyxl.utils import get\_column\_letter  wb = load\_workbook(r"C:\Users\hp\Desktop\abc.xlsx")  ws = wb.active  server = smtplib.SMTP("smtp-mail.outlook.com",587)  server.ehlo()  server.starttls()  i = 1  while(i):      option = int(input("""          Welcome TO NITJ Mess record tracker application!          please select the specified option:          1. New student registering.          2. Today's record punching.          3. Viewing records.          4. Exit Menu!      """))      if option == 1:          print("the list of current students are: \n")          for stu in ws["B"]:              print(f"{stu.value} \n")          name\_stu = input("enter the name of the student: ")          email\_stu = input("enter the email id of the student: ")          r\_end = str(len(ws["B"])+1)          ws["A"+r\_end] = str(len(ws["B"]))          ws["B"+r\_end] = name\_stu          ws["C"+r\_end] = email\_stu          print("the updated list is: \n")          for stu in ws["B"]:              print(f"{stu.value} \n")          # for ema in ws["C"]:          #     print(f"{ema.value} \n")        elif option == 2:          date = input("enter today's date: ")          col\_no = get\_column\_letter(len(ws["1"])+1)          ws[col\_no+'1'] = date          for attendance in range(2,len(ws["A"])+1):              print(f"sr.no: {attendance - 1}, name: {ws["B"+str(attendance)].value}")              ws[col\_no+str(attendance)] = input("(p: present, a: absent) \n enter: ")          print("\n today's mess record punched succesfully!\n")          print(ws["C"+str(2)].value)          y=1          while(y):              option\_1 = int(input("""                  Would you like to send an email to the students regarding today's punching?:                                  1. YES                                  2. NO              """))              if option\_1 == 1:                  x = 1                  while(x):                      password = input("enter the mess email password: ")                      if password != "attendance@nitj":                          print("please enter the right password!")                      else:                          break                  server.login('mayank6g23@gmail.com',password)                  for i in range(1,len(ws["C"])):                      server.sendmail('mayank6g23@gmail.com',ws["C2":],'''                          Dear Student,                              You have been marked present in todays mess.                          regards,                          mess coordinator,                          NIT jalandhar                      ''')                  print("The mails have been sent successfully!")                  break                elif option\_1 == 2:                  break              else:                  print("please enter a valid value!")      elif option == 3:          print("the records are: ")          for row in range(1,len(ws["A"])+1):              for col in range(1,len(ws["1"])+1):                  i = get\_column\_letter(col)                  j = str(row)                  print(f"{ws[i+j].value}\t", end = "")              print("\n")      elif option == 4:          break      else:          print("please enter valid option!")  print("thank you for using the mess record App!")  wb.save(r"C:\Users\hp\Desktop\abc.xlsx") |

**OUTPUT**

**NOTE: THE OUTPUT HAVE BEEN TAKEN FROM A SINGLE TERMINAL AND THE INTERACTION WITH THE PROGRAMME HAS BEEN RECORDED AS MULTIPLE SCREENSHOTS:- **

****

**THE EXCEL SPREAD SHEEET (abc.xlsx) IS SHOWN BELOW AFTER BEING HANDLED BY THE PROGRAMME:-**

****

**The emails received by the students are:-**

|  |
| --- |
| **mayankk.ds.23@nitj.ac.in** |
| **ekamjots.ds.23@nitj.ac.in** |
| **navroopk.ds.23@nitj.ac.in** |
| **mayanka.ds.23@nitj.ac.in** |

**APPENDIX**

**For the project, 2 modules are used :-**

1. **OPENPYXL:** It is a powerful Python library for working with Excel files. It allows users to create, manipulate, and extract data from Excel spreadsheets programmatically. OpenPyXL supports a wide range of Excel file formats, including .xlsx and .xlsm, making it a versatile tool for data processing tasks. With OpenPyXL, users can perform operations such as reading and writing cell values, formatting cells, and creating charts and graphs. Its intuitive API and extensive documentation make it accessible to developers of all skill levels. OpenPyXL seamlessly integrates with other Python libraries, enabling complex data manipulation and analysis workflows. Overall, OpenPyXL is an asset for anyone working with Excel files in Python, offering efficient and flexible solutions for data management tasks.

2. **SMTPLIB:** This is a module in Python provides a simple and efficient way to send emails from a Python script. It interfaces with SMTP (Simple Mail Transfer Protocol) servers to send emails over the internet. With `SMTPlib`, developers can easily construct and send emails programmatically, including specifying recipients, subject, body, and attachments. This module supports authentication mechanisms for secure email transmission, ensuring data integrity and confidentiality. `SMTPlib` is a reliable choice for automating email communication in various applications, from simple notifications to complex email campaigns.

**These modules were installed in the system via pip install command in PowerShell :-**

|  |
| --- |
|  |