STUDENT MANAGEMENT SYSTEM (SMS)

DESCRIPTION OVERVIEW

Allowing teachers to log in with a username and password. It maintains a dictionary of teachers' credentials for validation. Once authenticated, teachers can perform various operations, including providing attendance, allotting marks, and viewing student details. These functionalities are encapsulated in the **teacher_operations** method.

For students, the system provides a range of operations accessible through the **student_operations** method. These operations include accepting, displaying, searching, updating, and deleting student data. Additionally, students can view their results and attendance details, with permissions enforced to ensure students only access their own information.

The data management features allow the addition, modification, and deletion of student records, capturing essential details such as name, roll number, department, gender, date of birth, phone number, and parents' names. The code exhibits a well-organized and modular structure, enhancing readability and maintainability.

PROBLEM STATEMENT:

In this project our goal is to achieve the problem of Design and implement a Student Management System that allows both students and teachers to perform various operations related to student data, attendance, and results. The system should be menu-driven.

TECHNOLOGY USED:

Here we have used Google collab which was introduced by Google research team and Python programming language where it’s easy to code & learn.

INSTALLATION

Step 01: Go to website of JetBrains and click on the **"DOWNLOAD"** link of the Community section. https://www.jetbrains.com/pycharm/download/

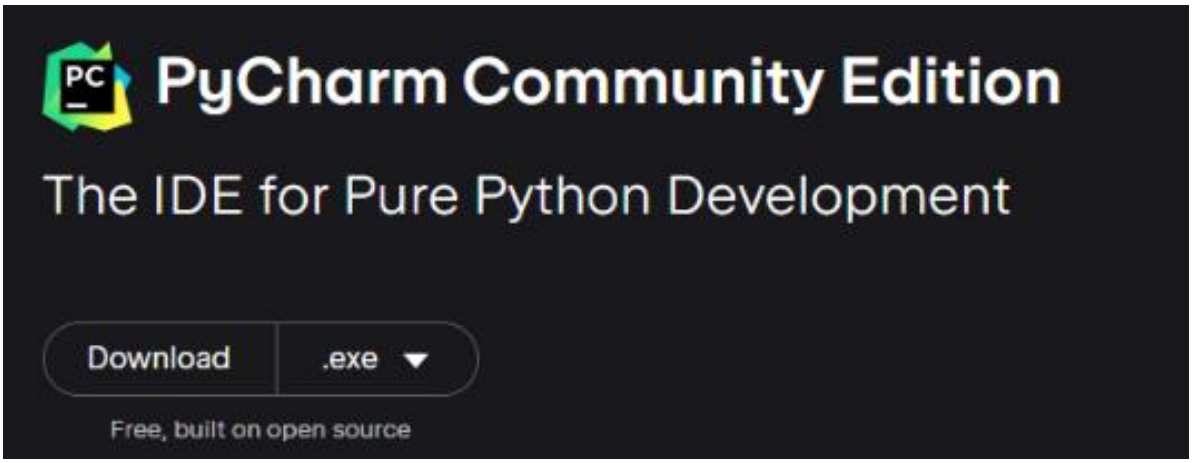


Fig 1.1

Step 02: After clicking on Download Click on Next

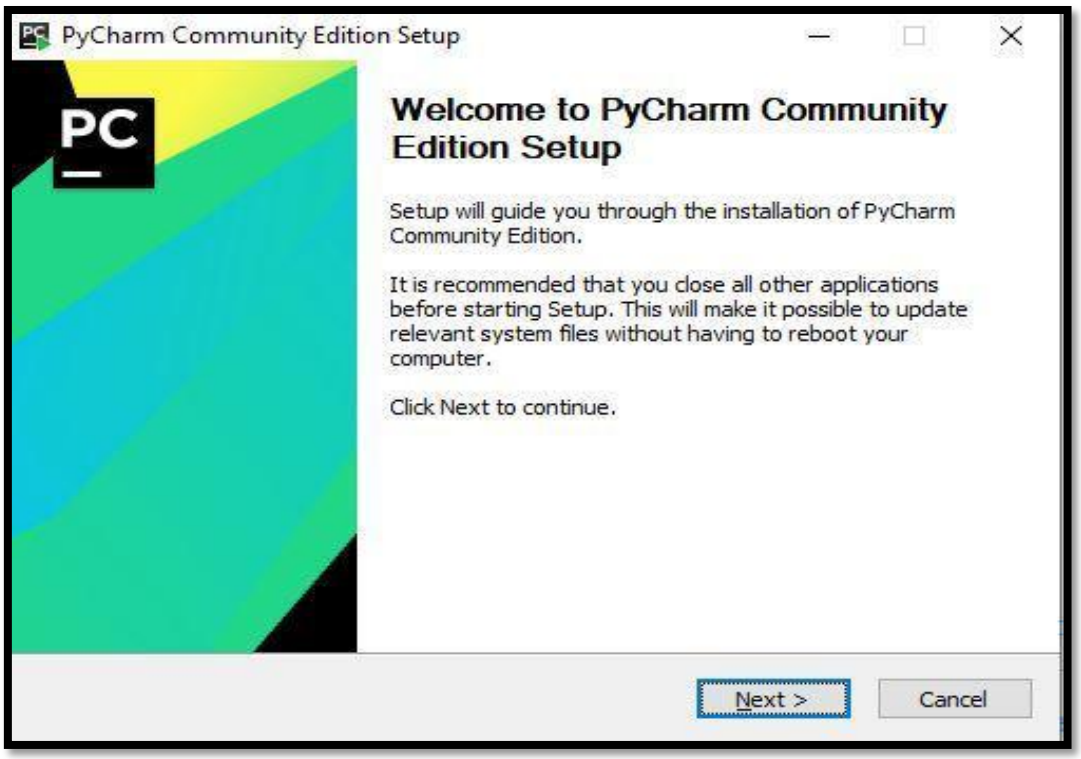


Fig 1.2

Step 03: After Click on Next, you need to choose the destination folder according to your choice.

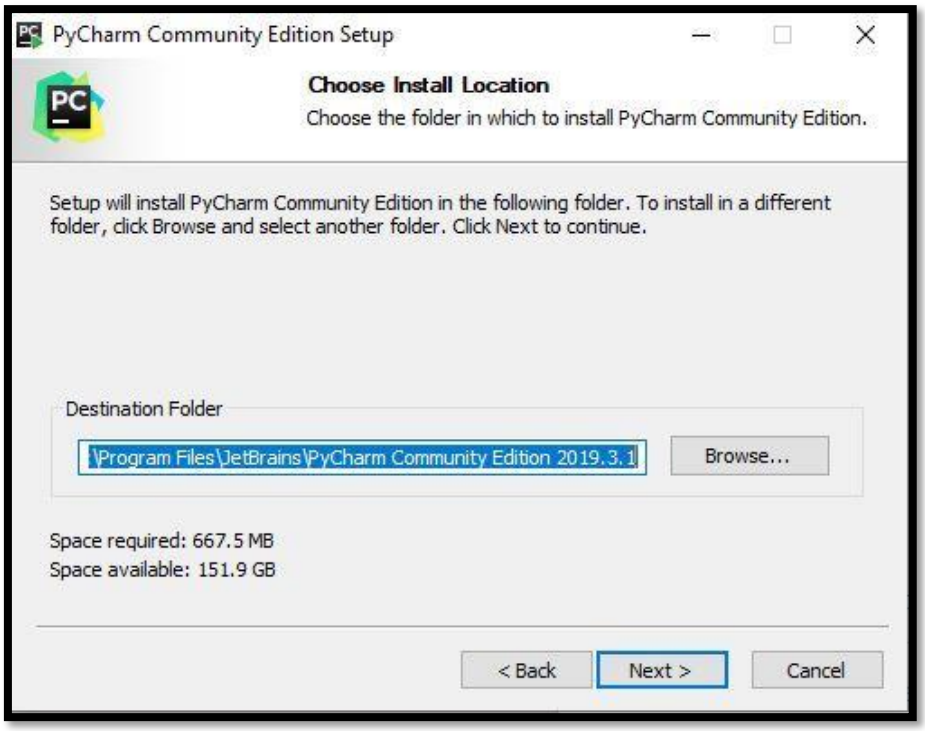


Fig 1.3

Step 04: Choose options of installation according to your choice.

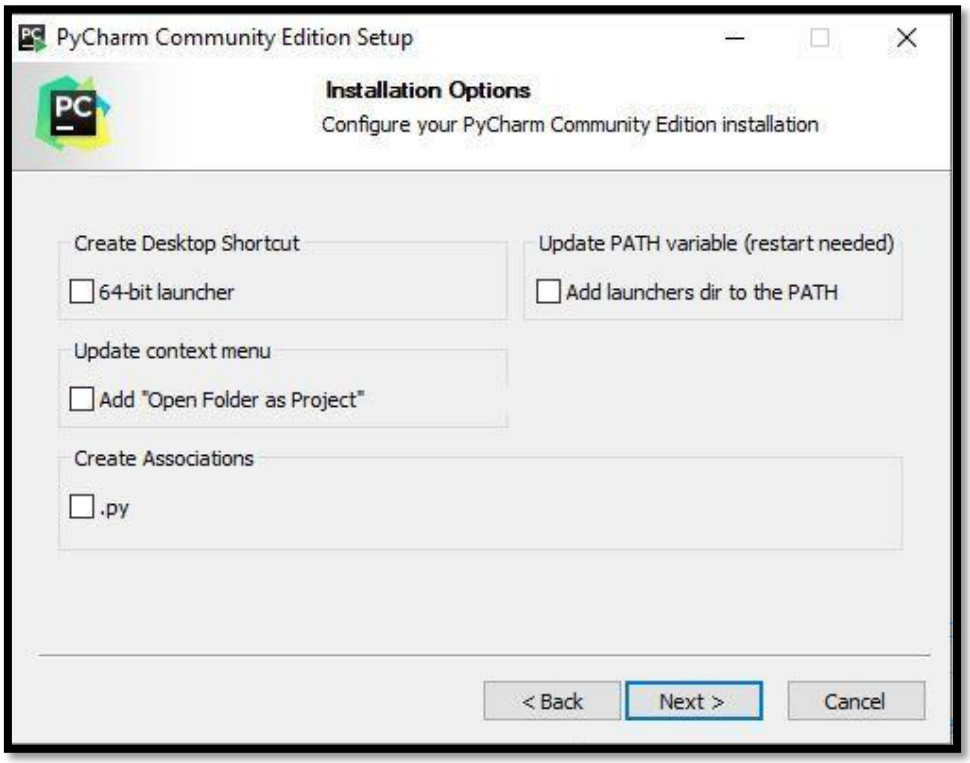


Fig 1.4

Step 05: Choose JetBrains and Click on “Install”.

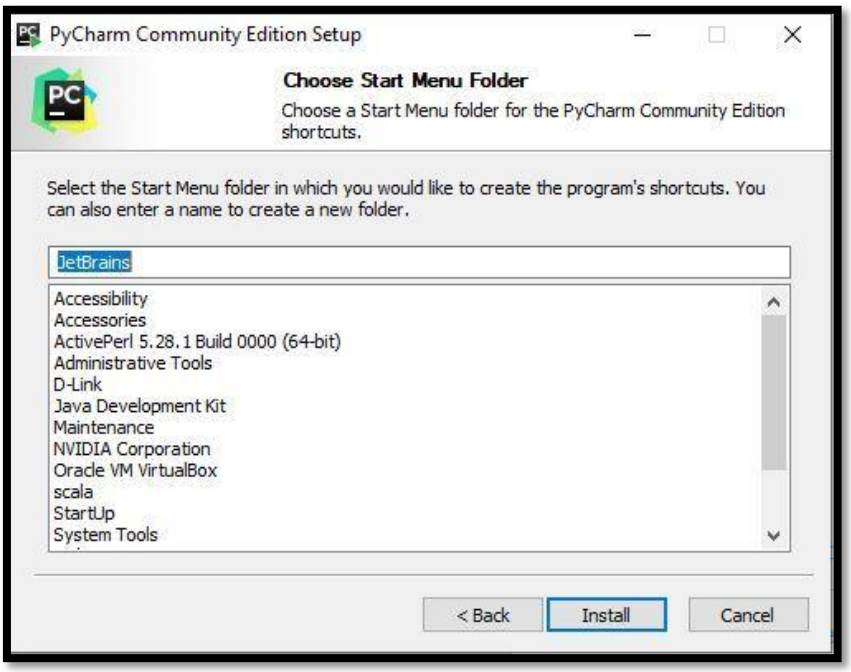


Fig 1.5

Step 06: Let the installation finished.

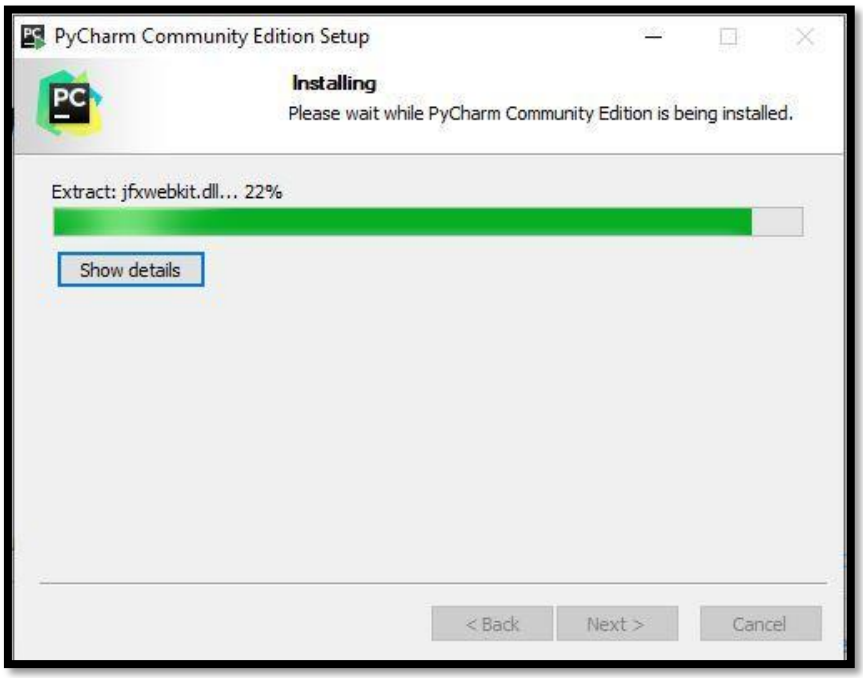


Fig 1.6

Step 07: After Installation completed, It will show that PyCharm is installed successfully, then click on "I want to manually reboot later". Click on Finish and then the process is completed.

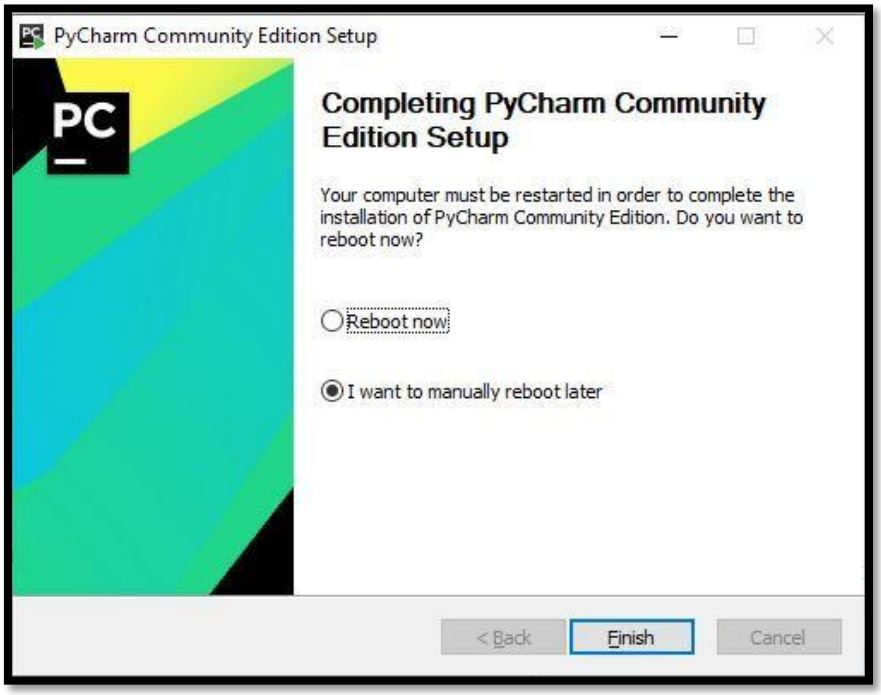


Fig 1.7

Step 08: Once the Installation is over, PyCharm can be searched and started from the Start Menu. Follow the steps given below to do the same:

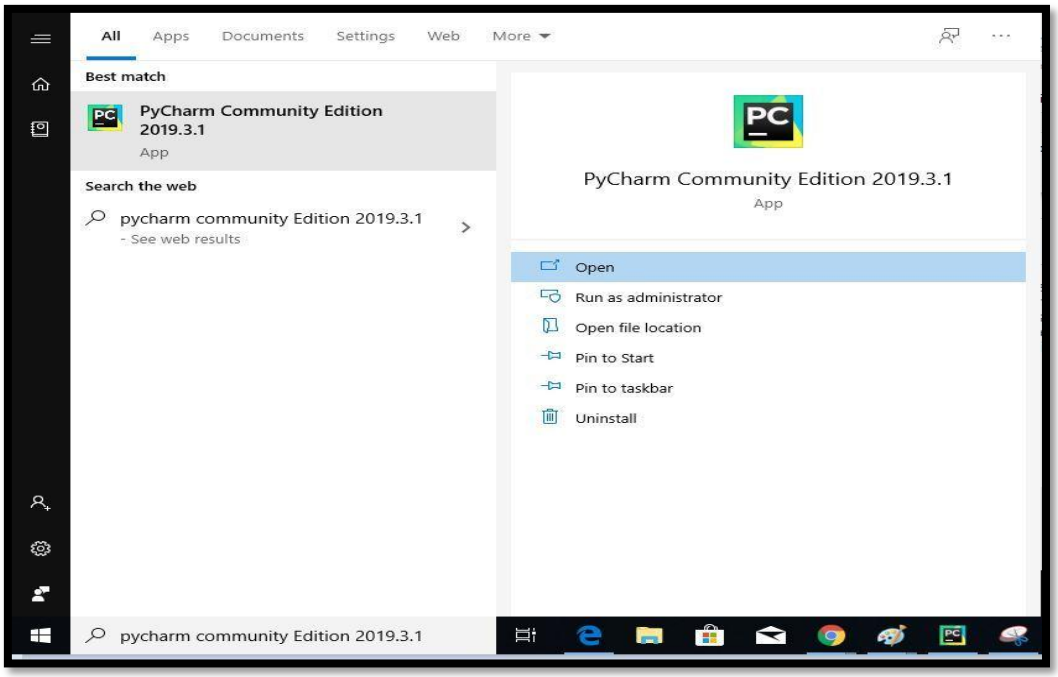


Fig 1.8

Step 09: Installation of PyCharm community edition is completed successfully

WORKFLOW DIAGRAM

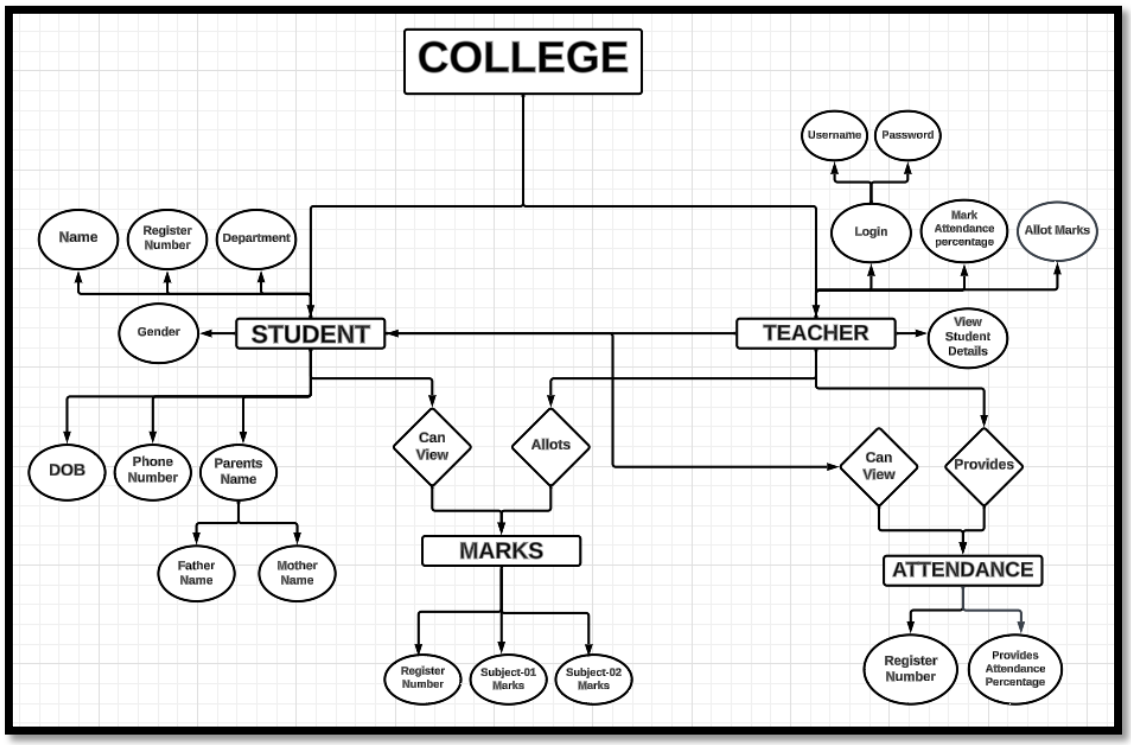


Fig 1.9

IMPLEMENTATION

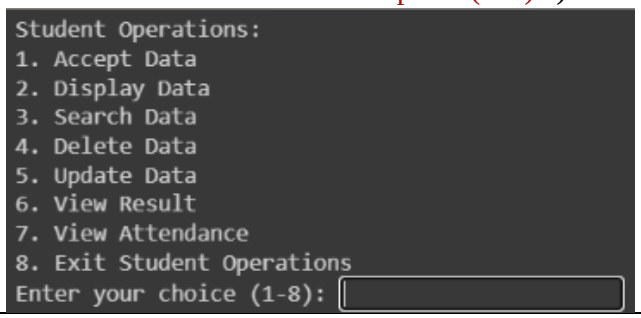
```
# Initializing and managing passwords like username and password in Dictionary format:
class StudentManagementSystem:
    def --init-- (self):
        self.students = {}
        self.teachers = {'vishal': 'vishal123'}
        self.logged_in_user = None

    def login(self, username, password):
    if username in self.teachers and self.teachers[username] == password:
        print(f"Welcome, Teacher {username}!")
        self.logged_in_user = 'teacher'
    else:
        print("Invalid username or password. Please try again.")
```

O/P: -

```
Student Management System Menu:
1. Student
2. Teacher
3. Exit
Enter your choice (1-3): 2
Enter teacher username: vishal
Enter teacher password: vishal123
Welcome, Teacher vishal!
```

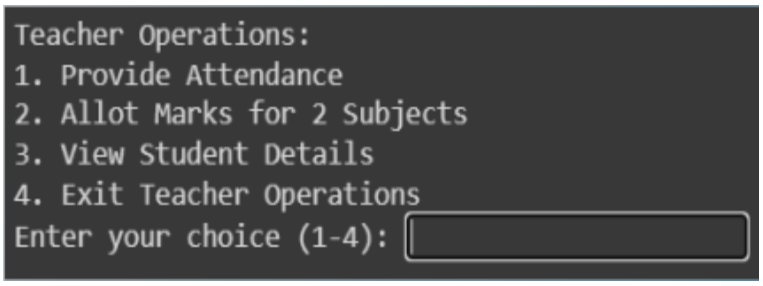
```python
# Student Operation:
def student_operations(self):
    while True:
        print("\nStudent Operations:")
        print("1. Accept Data")
        print("2. Display Data")
        print("3. Search Data")
        print("4. Delete Data")
        print("5. Update Data")
        print("6. View Result")
        print("7. View Attendance")
        print("8. Exit Student Operations")

        student_choice = input("Enter your choice (1-8): ")

        if student_choice == '1':
            self.accept_data()
        elif student_choice == '2':
            self.display_data()
        elif student_choice == '3':
            roll_number = input("Enter roll number to search: ")
            self.search_data(roll_number)
        elif student_choice == '4':
            roll_number = input("Enter roll number to delete: ")
            self.delete_data(roll_number)
        elif student_choice == '5':
            roll_number = input("Enter roll number to update: ")
            self.update_data(roll_number)
        elif student_choice == '6':
            roll_number = input("Enter your roll number to view result: ")
            self.logged_in_user = roll_number
            self.view_result(roll_number)
        elif student_choice == '7':
            roll_number = input("Enter your roll number to view attendance: ")
            self.logged_in_user = roll_number
            self.view_attendance(roll_number)
        elif student_choice == '8':
            break
        else:
            print("Invalid choice. Please enter a valid option (1-8).")
```

O/P: -



```python
# Teacher Operation:
def teacher_operations(self):
    while True:
        print("\nTeacher Operations:")
        print("1. Provide Attendance")
        print("2. Allot Marks for 2 Subjects")
        print("3. View Student Details")
        print("4. Exit Teacher Operations")

        teacher_choice = input("Enter your choice (1-4): ")

        if teacher_choice == '1':
            if self.logged_in_user == 'teacher':
                roll_number = input("Enter student roll number to provide attendance: ")
                self.provide_attendance(roll_number)
            else:
                print("Permission denied. Please login as a teacher.")
        elif teacher_choice == '2':
            if self.logged_in_user == 'teacher':
                roll_number = input("Enter student roll number to allot marks: ")
                self.allot_marks(roll_number)
            else:
                print("Permission denied. Please login as a teacher.")
        elif teacher_choice == '3':
            if self.logged_in_user == 'teacher':
                roll_number = input("Enter student roll number to view details: ")
                self.view_student_details(roll_number)
            else:
                print("Permission denied. Please login as a teacher.")
        elif teacher_choice == '4':
            break
        else:
            print("Invalid choice. Please enter a valid option (1-4).")
```

O/P:-



```python
# Accept Operation:
def accept_data(self):
    name = input("Enter student name: ")
    roll_number = input("Enter roll number: ")
    department = input("Enter department (CSE/ME/EEE): ")
    gender = input("Enter Gender: ")
    dob = input("Enter Date of Birth (DD-MM-YYYY): ")
    phone_number = input("Enter phone number: ")
    father_name = input("Enter Father's name(As per 10th marksheet): ")
    mother_name = input("Enter Mother's name(As per 10th marksheet): ")

    self.students[roll_number] = {
        'name': name,
        'Registration number': roll_number,
        'department': department,
        'gender': gender,
        'dob': dob,
        'phone number': phone_number,
        'father name': father_name,
        'mother name': mother_name,
        'result': None
    }
    print(f"Data for {name} (Roll No: {roll_number}) added successfully.")
```

O/P: -

```
Enter your choice (1-8): 1
Enter student name: N M SUHAS
Enter roll number: 408CS21029
Enter department (CSE/ME/EEE): CSE
Enter Gender: Male
Enter Date of Birth (DD-MM-YYYY): 26-06-2005
Enter phone number: 7892331225
Enter Father's name(As per 10th marksheet): N B MOHAN
Enter Mother's name(As per 10th marksheet): N M VINUTHA
Data for N M SUHAS (Roll No: 408CS21029) added successfully.
```

```python
# Display Operation:
def display_data(self):
    for roll_number, data in self.students.items():
        print(f"\nStudent Details for Roll No {roll_number}:")
        for key, value in data.items():
            if key not in ('result', 'attendance'):
                print(f"{key}: {value}")
```

O/P:-

```
Enter your choice (1-8): 2

Student Details for Roll No 408CS21029:
name: N M SUHAS
Registration number: 408CS21029
department: CSE
gender: Male
dob: 26-06-2005
phone number: 7892331225
father name: N B MOHAN
mother name: N M VINUTHA
```

```
# Search Operation:
def search_data(self, roll_number):
        if roll_number in self.students:
            print(f"\nStudent Details for Roll No {roll_number}:")
            for key, value in self.students[roll_number].items():
                print(f"{key}: {value}")
        else:
            print(f"Student with Roll No {roll_number} not found.")
```

O/P:-

```
Enter your choice (1-8): 3
Enter roll number to search: 408CS21029

Student Details for Roll No 408CS21029:
name: N M SUHAS
Registration number: 408CS21029
department: CSE
gender: Male
dob: 26-06-2005
phone number: 7892331225
father name: N B MOHAN
mother name: N M VINUTHA
result: None
```

```
# Delete Operation:
def update_data(self, roll_number):
        if roll_number in self.students:
            print(f"\nUpdate Details for Roll No {roll_number}:")
            name = input("Enter updated name: ")
            department = input("Enter updated department (CSE/ME/EEE): ")
            gender = input("Enter Gender: ")
            dob = input("Enter updated Date of Birth (YYYY-MM-DD): ")
            phone_number = input("Enter new phone number: ")
            father_name = input("Enter updated Father's name: ")
            mother_name = input("Enter updatedMother's name: ")

            self.students[roll_number]['name'] = name
            self.students[roll_number]['department'] = department
            self.students[roll_number]['Gender'] = gender
            self.students[roll_number]['DOB'] = dob
            self.students[roll_number]['phone number'] = phone_number
            self.students[roll_number]['father name'] = father_name
            self.students[roll_number]['mother name'] = mother_name

            print(f"Record for Roll No {roll_number} updated successfully.")
        else:
            print(f"Student with Roll No {roll_number} not found.")
```

O/P: -

```
Enter your choice (1-8): 5
Enter roll number to update: 408CS21029

Update Details for Roll No 408CS21029:
Enter updated name: SUHAS N M
Enter updated department (CSE/ME/EEE): MECHANICAL
Enter Gender: Male
Enter updated Date of Birth (YYYY-MM-DD): 27-06-2005
Enter new phone number: 1234567891
Enter updated Father's name: MOHAN N B
Enter updatedMother's name: VINUTHA N M
Record for Roll No 408CS21029 updated successfully.
```

```
# Delete Operation:
def delete_data(self, roll_number):
    if roll_number in self.students:
        del self.students[roll_number]
        print(f"Record for Roll No {roll_number} deleted successfully.")
    else:
        print(f"Student with Roll No {roll_number} not found.")
```

O/P: -

```
Student Operations:
1. Accept Data
2. Display Data
3. Search Data
4. Delete Data
5. Update Data
6. View Result
7. View Attendance
8. Exit Student Operations
Enter your choice (1-8): 4
Enter roll number to delete: 408CS21029
Record for Roll No 408CS21029 deleted successfully.
```

```
# Providing attendance to students:
def provide_attendance(self, roll_number):
    if roll_number in self.students:
        attendance_percentage = float(input("Enter attendance percentage for the student: "))
        self.students[roll_number]['attendance'] = attendance_percentage
        print(f"Attendance filled successfully for Roll No {roll_number}.")
    else:
        print(f"Student with Roll No {roll_number} not found.")
```

O/P: -

```
Teacher Operations:
1. Provide Attendance
2. Allot Marks for 2 Subjects
3. View Student Details
4. Exit Teacher Operations
Enter your choice (1-4): 1
Enter student roll number to provide attendance: 408CS21029
Enter attendance percentage for the student: 82
Attendance filled successfully for Roll No 408CS21029.
```

```python
# Allot marks to students:
def allot_marks(self, roll_number):
    if roll_number in self.students:
        subject1_marks = float(input("Enter marks for Subject 1: "))
        subject2_marks = float(input("Enter marks for Subject 2: "))
        self.students[roll_number]['subject1_marks'] = subject1_marks
        self.students[roll_number]['subject2_marks'] = subject2_marks

        print(f"Marks assigned successfully for Roll No {roll_number}.")
    else:
        print(f"Student with Roll No {roll_number} not found.")
```

O/P:-

```
Enter your choice (1-4): 2
Enter student roll number to allot marks: 408CS21029
Enter marks for Subject 1: 86
Enter marks for Subject 2: 92
Marks assigned successfully for Roll No 408CS21029.
```

```python
# View student details in teacher table:
def view_student_details(self, roll_number):
    if roll_number in self.students:
        print(f"\nStudent Details for Roll No {roll_number}:")
        for key, value in self.students[roll_number].items():
            print(f"{key}: {value}")
        self.view_result(roll_number)
    else:
        print(f"Student with Roll No {roll_number} not found.")
```

O/P:-

```
Enter your choice (1-4): 3
Enter student roll number to view details: 408CS21029

Student Details for Roll No 408CS21029:
name: N M SUHAS
Registration number: 408CS21029
department: CSE
gender: Male
dob: 26-06-2005
phone number: 7892331225
father name: N B MOHAN
mother name: N M VINUTHA
result: None
attendance: 82.0
subject1_marks: 86.0
subject2_marks: 92.0
Gender: Male
DOB: 26-06-2005

Result for Roll No 408CS21029:
Subject 1 Marks: 86.0
Subject 2 Marks: 92.0
Total Marks: 178.0
```

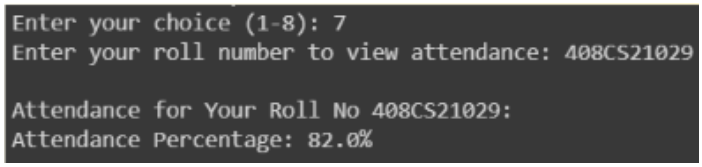
```
# View results as a student:
def view_result(self, roll_number):
    if roll_number in self.students:
        # Check if the logged-in user is a student and matches the provided roll_number
        if self.logged_in_user == roll_number:
            if 'subject1_marks' in self.students[roll_number] and 'subject2_marks' in self.students[roll_number]:
                subject1_marks = self.students[roll_number]['subject1_marks']
                subject2_marks = self.students[roll_number]['subject2_marks']
                total_marks = subject1_marks + subject2_marks
                print(f"\nResult for Your Roll No {roll_number}:")
                print(f"Subject 1 Marks: {subject1_marks}")
                print(f"Subject 2 Marks: {subject2_marks}")
                print(f"Total Marks: {total_marks}")
            else:
                print("Marks not available. Please ask the teacher to allot marks.")
        elif self.logged_in_user == 'teacher':
            if 'subject1_marks' in self.students[roll_number] and 'subject2_marks' in self.students[roll_number]:
                subject1_marks = self.students[roll_number]['subject1_marks']
                subject2_marks = self.students[roll_number]['subject2_marks']
                total_marks = subject1_marks + subject2_marks
                print(f"\nResult for Roll No {roll_number}:")
                print(f"Subject 1 Marks: {subject1_marks}")
                print(f"Subject 2 Marks: {subject2_marks}")
                print(f"Total Marks: {total_marks}")
            else:
                print("Marks not available. Please allot marks for the student.")
        else:
            print("Permission denied. You can only view your own result.")
    else:
        print(f"Student with Roll No {roll_number} not found.")
```

O/P: -

```
Enter your choice (1-8): 6
Enter your roll number to view result: 408CS21029

Result for Your Roll No 408CS21029:
Subject 1 Marks: 86.0
Subject 2 Marks: 92.0
Total Marks: 178.0
```

```
# View attendance in Student table:
def view_attendance(self, roll_number):
    if roll_number in self.students:
        # Check if the logged-in user is a student and matches the provided roll_number
        if self.logged_in_user == roll_number:
            if 'attendance' in self.students[roll_number]:
                attendance_percentage = self.students[roll_number]['attendance']
                print(f"\nAttendance for Your Roll No {roll_number}:")
                print(f"Attendance Percentage: {attendance_percentage}%")
            else:
                print("Attendance not available. Please ask the teacher to provide attendance.")
        else:
            print("Permission denied. You can only view your own attendance.")
    else:
        print(f"Student with Roll No {roll_number} not found.")
```

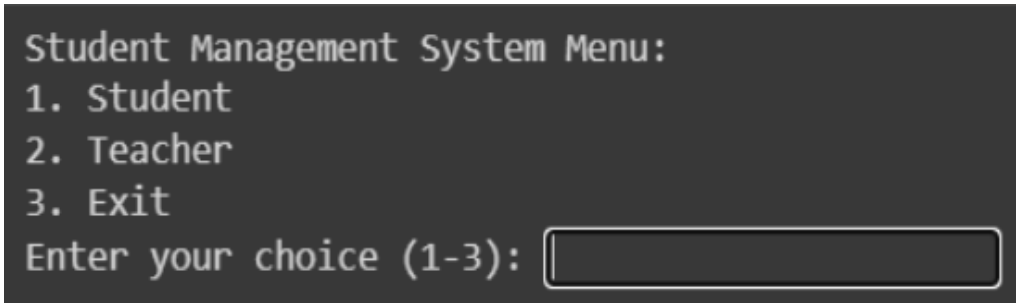
O/P:-




```
# Demonstration of code:
sms = StudentManagementSystem()

while True:
    print("\nStudent Management System Menu:")
    print("1. Student")
    print("2. Teacher")
    print("3. Exit")

    choice = input("Enter your choice (1-3): ")

    if choice == '1':
        # Set the logged-in user as a student when entering the student menu
        sms.logged_in_user = 'student'
        sms.student_operations()
    elif choice == '2':
        username = input("Enter teacher username: ")
        password = input("Enter teacher password: ")
        sms.login(username, password)
        if sms.logged_in_user == 'teacher':
            sms.teacher_operations()
```

```
elif choice == '3':
    print("Exiting Student Management System. Goodbye!")
    break
else:
    print("Invalid choice. Please enter a valid option (1-3).")
```

O/P: -



CONCLUSION

1. **Structured System:** The StudentManagementSystem class offers a well-organized and interactive framework for managing student information.
2. **Comprehensive Operations:** The class incorporates methods that handle a diverse set of operations, including accepting, displaying, searching, updating, and deleting student data.
3. **Data Acceptance:** The accept _data method facilitates the input of detailed student information, encompassing aspects such as name, roll number, age, gender, etc...
4. **Data Display:** The display_data method provides a comprehensive overview of all stored student records.
5. **Search Functionality:** The search_data method allows for efficient searching, supporting queries based on either the full roll number or the last two digits of the roll number.
6. **Data Modification:** The update_data method enables the modification of a student's name and marks, utilizing the provided roll number.
7. **Data Deletion:** The delete_data method removes a student's record from the system based on the specified roll number.

REFERENCES

1. https://www.canva.com/design/DAF7VEWhqNE/eTDtumi9yn5yXlAN0VrjnQ/edit?utm_content=DAF7VEWhqNE&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

2. https://lucid.app/lucidchart/ef69d560-137d-4f77-bd5a-eff442a7f7d8/edit?viewport_loc=-793%2C-469%2C3200%2C1412%2C0_0&invitationId=inv_c231c50d-b347-46a0-bef1-1516c9a2ab17

3. https://lucid.app/lucidchart/3ef47b5d-a2a4-4d0d-9821-7bb3834d9f0e/edit?viewport_loc=-802%2C-43%2C3046%2C1345%2C0_0&invitationId=inv_111cc28d-1b24-4954-a280-78771a9044fd

4. https://www.geeksforgeeks.org/how-to-install-python-pycharm-on-windows/