Project Report
On
Student Management System

Submitted By:
Jiya Rawat
Class: XII

Submitted To:
Mr. Vikram Singh Rawat
St. Benedict's School, Shivpuri

Certificate

This is to certify that the project titled "**Student Management System**" submitted by **Jiya Rawat**, a student of Class XII, has been completed under my guidance and supervision.

The project is a partial fulfillment of the requirement for the Computer Science course, CBSE Class XII.

Teacher's Name: Mr. Vikram Singh Rawat
School: St. Benedict's School, Shivpuri

Table of Contents

Objective

The objective of this project is to create a simple Student Management System using Python and CSV as the backend. This project provides features to add, view, update, delete, and search student records.

Tools and Technologies Used

- **Programming Language**: Python
- **Data Storage**: CSV (Comma-Separated Values)
- **Editor**: VS Code / PyCharm / IDLE

Features

1. Add Student
2. View Students
3. Search Student
4. Delete Student
5. Update Student

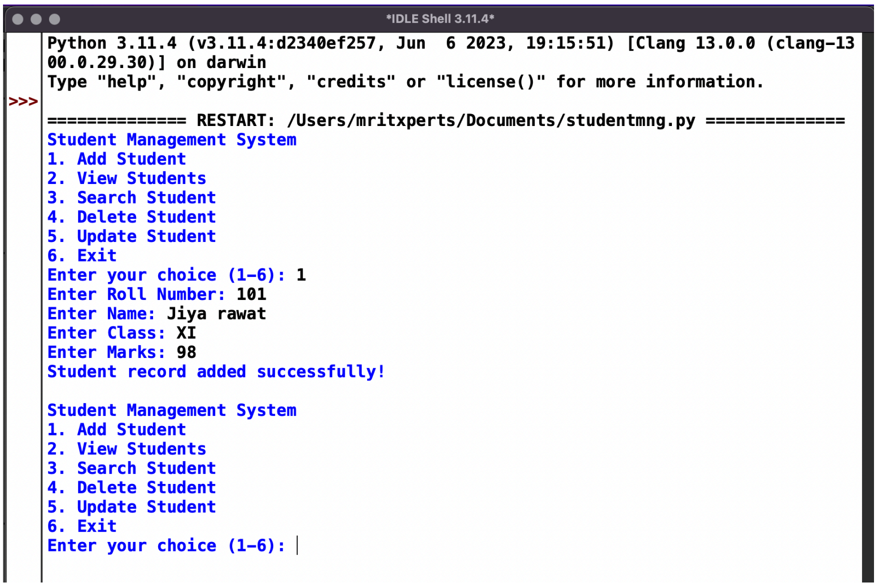
Python Code

```
import csv
import os

def initialize_csv(file_name):
    """Initialize the CSV file with headers if it does not exist."""
    if not os.path.exists(file_name):
        with open(file_name, mode='w', newline='') as file:
            writer = csv.writer(file)
            writer.writerow(["Roll Number", "Name", "Class", "Marks"])



def add_student(file_name):
    """Add a new student record to the CSV file."""
    roll_number = input("Enter Roll Number: ")
    name = input("Enter Name: ")
    student_class = input("Enter Class: ")
    marks = input("Enter Marks: ")

    with open(file_name, mode='a', newline='') as file:
        writer = csv.writer(file)
        writer.writerow([roll_number, name, student_class, marks])

    print("Student record added successfully!\n")


def view_students(file_name):
    """Display all student records from the CSV file."""
    try:
        with open(file_name, mode='r') as file:
            reader = csv.reader(file)
            print("\nStudent Records:")
            print("----------------------------------------")
            for row in reader:
                print("\t".join(row))
            print("----------------------------------------\n")
    except FileNotFoundError:
        print("No records found. Please add some students first.\n")


def search_student(file_name):
    """Search for a student record by roll number."""
    roll_number = input("Enter Roll Number to search: ")
    found = False

    try:
        with open(file_name, mode='r') as file:
            reader = csv.reader(file)
            for row in reader:
                if row[0] == roll_number:
                    print("\nStudent Record Found:")
                    print("Roll Number: ", row[0])
                    print("Name: ", row[1])
                    print("Class: ", row[2])
                    print("Marks: ", row[3])
                    found = True
                    break
            if not found:
                print("\nStudent record not found!\n")
    except FileNotFoundError:
        print("No records found. Please add some students first.\n")


def delete_student(file_name):
    """Delete a student record by roll number."""
    roll_number = input("Enter Roll Number to delete: ")
    rows = []
    found = False

    try:
        with open(file_name, mode='r') as file:
            reader = csv.reader(file)
            for row in reader:
                if row[0] != roll_number:
                    rows.append(row)
                else:
                    found = True

        if found:
            with open(file_name, mode='w', newline='') as file:
                writer = csv.writer(file)
                writer.writerows(rows)
            print("\nStudent record deleted successfully!\n")
        else:
            print("\nStudent record not found!\n")
    except FileNotFoundError:
        print("No records found. Please add some students first.\n")


def update_student(file_name):
    """Update a student's record by roll number."""
    roll_number = input("Enter Roll Number to update: ")
    rows = []
    found = False
    try:
        with open(file_name, mode='r') as file:
            reader = csv.reader(file)
            for row in reader:
                if row[0] == roll_number:
                    print("\nCurrent Details:")
                    print("Roll Number: ", row[0])
                    print("Name: ", row[1])
                    print("Class: ", row[2])
                    print("Marks: ", row[3])
                    row[1] = input("Enter new Name: ")
                    row[2] = input("Enter new Class: ")
                    row[3] = input("Enter new Marks: ")
                    found = True
                rows.append(row)

        if found:
            with open(file_name, mode='w', newline='') as file:
                writer = csv.writer(file)
                writer.writerows(rows)
            print("\nStudent record updated successfully!\n")
        else:
            print("\nStudent record not found!\n")
    except FileNotFoundError:
        print("No records found. Please add some students first.\n")


def main():
    """Main function to drive the program."""
    file_name = "students.csv"
    initialize_csv(file_name)

    while True:
        print("Student Management System")
        print("1. Add Student")
        print("2. View Students")
        print("3. Search Student")
        print("4. Delete Student")
        print("5. Update Student")
        print("6. Exit")
        choice = input("Enter your choice (1-6): ")

        if choice == '1':
            add_student(file_name)
        elif choice == '2':
            view_students(file_name)
        elif choice == '3':
            search_student(file_name)
        elif choice == '4':
            delete_student(file_name)
        elif choice == '5':
            update_student(file_name)
        elif choice == '6':
            print("Exiting the program. Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.\n")

if __name__ == "__main__":
    main()
```

Output Screenshots

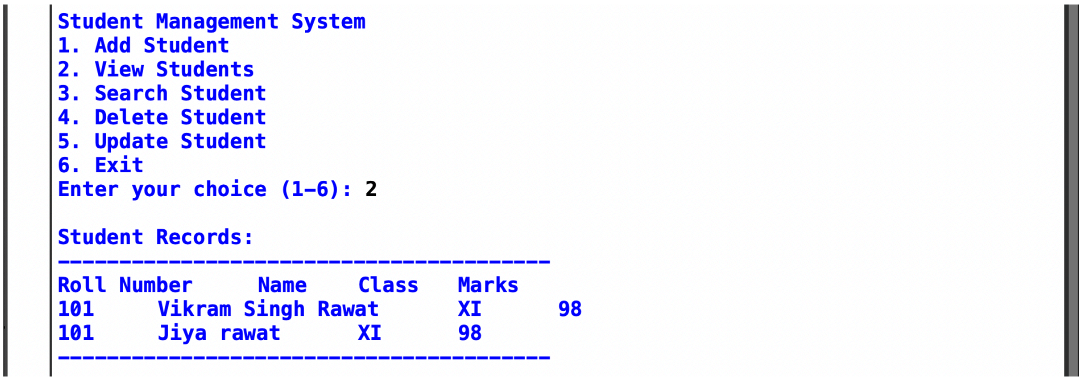
Attach screenshots of the following:

1. Adding a Student



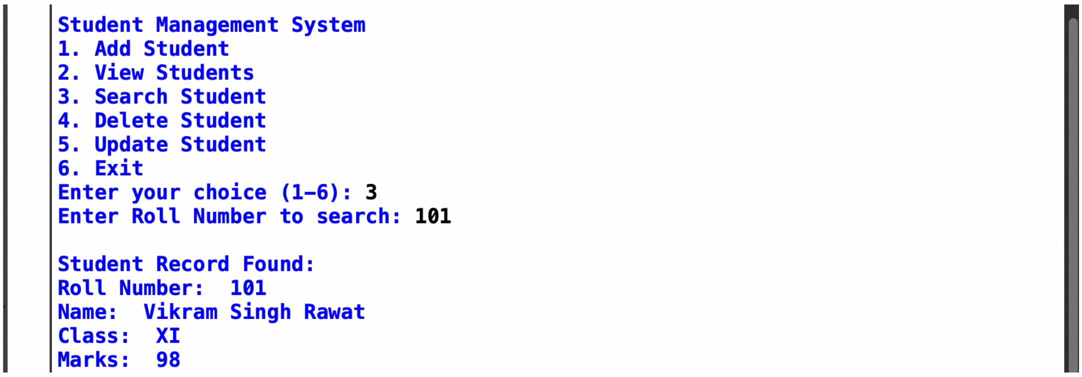
```
Python 3.11.4 (v3.11.4:d2340ef257, Jun  6 2023, 19:15:51) [Clang 13.0.0 (clang-13
00.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: /Users/mritxperts/Documents/studentmng.py =============
Student Management System
1. Add Student
2. View Students
3. Search Student
4. Delete Student
5. Update Student
6. Exit
Enter your choice (1-6): 1
Enter Roll Number: 101
Enter Name: Jiya rawat
Enter Class: XI
Enter Marks: 98
Student record added successfully!

Student Management System
1. Add Student
2. View Students
3. Search Student
4. Delete Student
5. Update Student
6. Exit
Enter your choice (1-6):
```

2. Viewing Students



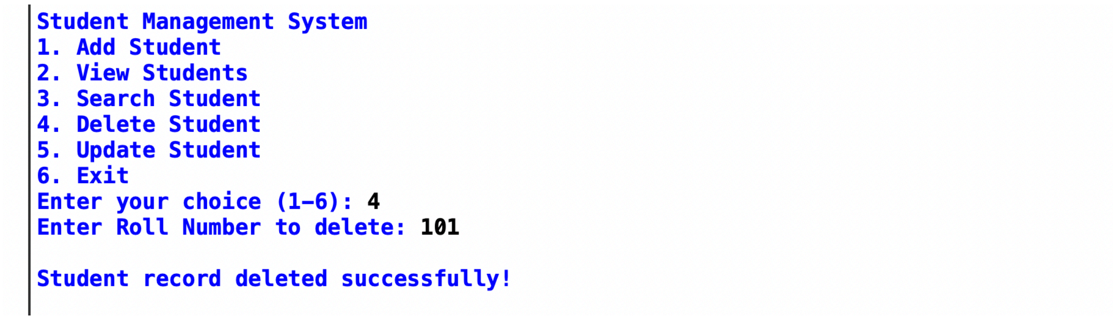
```
Student Management System
1. Add Student
2. View Students
3. Search Student
4. Delete Student
5. Update Student
6. Exit
Enter your choice (1-6): 2

Student Records:
----------------------------------------
Roll Number     Name    Class   Marks
101     Vikram Singh Rawat      XI      98
101     Jiya rawat      XI      98
----------------------------------------
```

3. Searching a Student



```
Student Management System
1. Add Student
2. View Students
3. Search Student
4. Delete Student
5. Update Student
6. Exit
Enter your choice (1-6): 3
Enter Roll Number to search: 101

Student Record Found:
Roll Number:  101
Name:  Vikram Singh Rawat
Class:  XI
Marks:  98
```

4. Deleting a Student



```
Student Management System
1. Add Student
2. View Students
3. Search Student
4. Delete Student
5. Update Student
6. Exit
Enter your choice (1-6): 4
Enter Roll Number to delete: 101

Student record deleted successfully!
```

5. Updating a Student

Conclusion

This project demonstrates the ability to perform basic CRUD operations using Python and a CSV backend. It is simple, efficient, and user-friendly.

Let me know if you'd like a formatted PDF or Word version of this report!

Bibliography

1. **Python Documentation**
   - URL: https://docs.python.org/3/
   - Description: Official Python documentation used for understanding core libraries and syntax.
2. **CSV Module Documentation**
   - URL: https://docs.python.org/3/library/csv.html
   - Description: Reference for using the `csv` module to handle CSV file operations.
3. **W3Schools - Python Tutorials**
   - URL: https://www.w3schools.com/python/
   - Description: Tutorials and examples for learning Python programming basics and advanced concepts.
4. **GeeksforGeeks - Python Programming**
   - URL: https://www.geeksforgeeks.org/python-programming-language/
   - Description: Examples and explanations of Python functionalities relevant to the project.
5. **Stack Overflow**
   - URL: https://stackoverflow.com/
   - Description: Community forum used for solving coding issues and referencing solutions for CSV and file handling.
6. **Programiz - Python CSV Handling**
   - URL: https://www.programiz.com/python-programming/csv
   - Description: Guide for handling CSV files in Python, including reading, writing, and updating data.

This bibliography section ensures proper credit to all the resources used during the project development.