Student Marks Management System

This project allows users to **store, update, and analyze student marks** using arrays (lists in Python).

Features:

☑ Add marks for students
☑ Update marks
☑ Calculate average, highest, and lowest marks
☑ Display all marks

Code Implementation:

python

```
def display_marks(marks):
    print("Student Marks:", marks)


def add_marks(marks):
    mark = float(input("Enter the new mark: "))
    marks.append(mark)
    print("Mark added successfully!")


def update_mark(marks):
    index = int(input("Enter the student index to update (0-based): "))
    if 0 <= index < len(marks):
        new_mark = float(input("Enter the new mark: "))
        marks[index] = new_mark
        print("Mark updated successfully!")
    else:
        print("Invalid index!")

def calculate_statistics(marks):
    if marks:
        print(f"\nAverage Marks: {sum(marks) / len(marks):.2f}")
        print(f"Highest Mark: {max(marks)}")
        print(f"Lowest Mark: {min(marks)}")
    else:
        print("\nNo marks available!")

marks = []

while True:
    print("\n===== Student Marks Management =====")
    print("1. Display Marks")
    print("2. Add Marks")
    print("3. Update Marks")
    print("4. Calculate Statistics")
    print("5. Exit")

    choice = input("Enter your choice: ")

    if choice == '1':
        display_marks(marks)
    elif choice == '2':
        add_marks(marks)
    elif choice == '3':
        update_mark(marks)
    elif choice == '4':
        calculate_statistics(marks)
    elif choice == '5':
        print("Exiting program. Goodbye!")
        break
    else:
        print("Invalid choice! Please enter a valid option.")
```

The **Student Marks Management System** is a simple **Python-based project** that utilizes **arrays (lists)** to store and manage student marks. It provides functionalities to **add, update, display, and analyze** student marks efficiently. The system is designed to help teachers or students keep track of marks and perform basic statistical analysis such as calculating the **average, highest, and lowest marks**.

The program runs in a **menu-driven format**, allowing users to interact through command-line inputs. It provides the following functionalities:

1. **Display Marks** – Shows all stored marks.
2. **Add Marks** – Allows the user to add a new student's mark.
3. **Update Marks** – Enables updating an existing mark.
4. **Calculate Statistics** – Computes and displays the **average, highest, and lowest** marks.
5. **Exit** – Terminates the program.