A

Project Report

On

“Bike Information Management System”

SUBMITTED BY:

MR.OMSAISH DHOKCHAULE (2124UCEM1017).

SUBJECT:

PROGRAMING AND PRACTICE

SOLVING USING C++

Under the Guidance of

Miss. ISHWARI TIRSE.

Department of Computer Science and Engineering

Sanjivani Rural Education Society’s

SANJIVANI UNIVERSITY

KOPARGAON – 423603, DIST : AHMEDNAGAR

2024-2025

INDEX

INTRODUCTION

The Bike Information System is a simple yet effective application designed to manage and store information about bicycles. As the popularity of cycling continues to grow, having a structured way to keep track of bike details has become increasingly important for both retailers and enthusiasts. This project aims to provide users with a convenient interface to add, display, and search for bike information. The system utilizes a user-friendly, menu-driven approach, making it accessible even for those with minimal programming experience. By employing a vector-based storage system, the application efficiently manages multiple entries of bike data, including brand, model, year, and price.

CODE

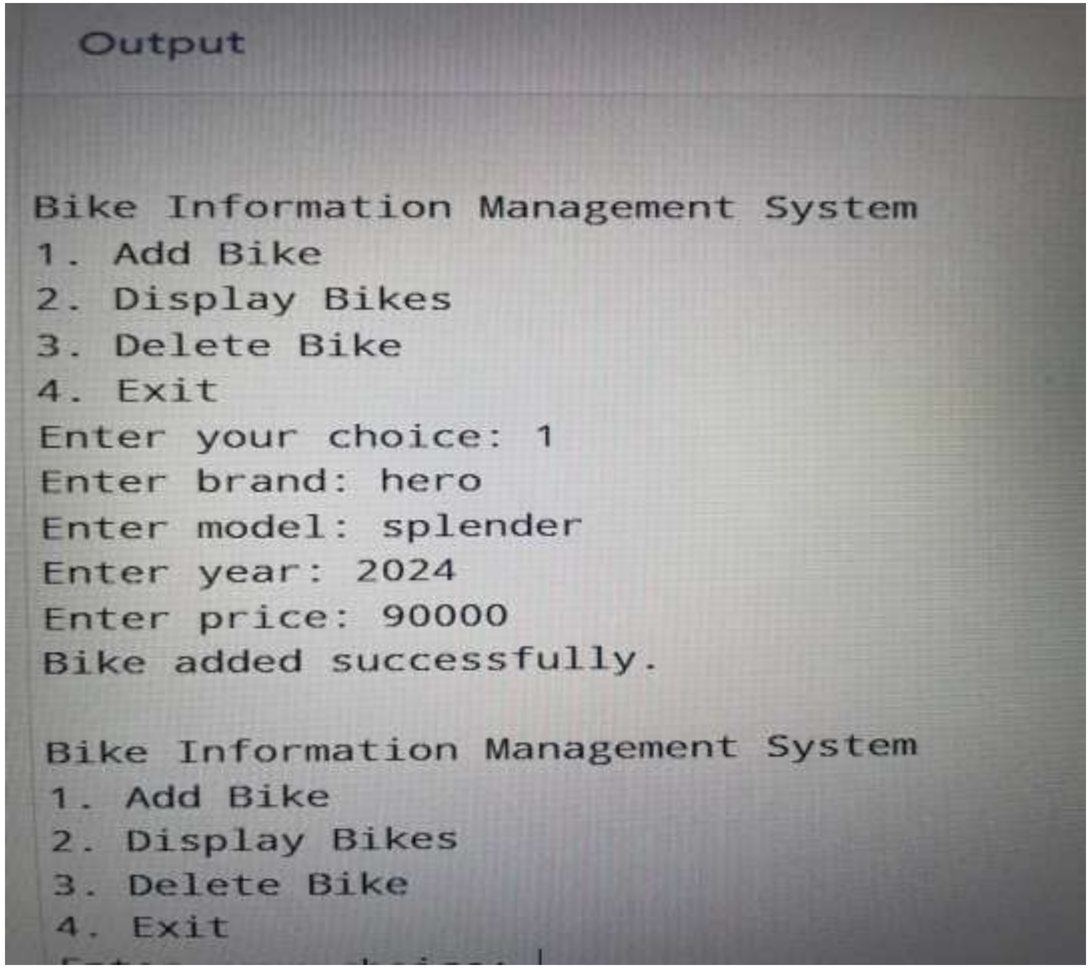
```
#include <iostream>
#include <vector>
#include <string>
class Bike
{
public:
    std::string brand;
    std::string model;
    int year;
    float price;

    Bike(std::string b, std::string m, int y, float p)
        : brand(b), model(m), year(y), price(p) {}
};

class BikeManagementSystem {
private:
    std::vector<Bike> bikes;
```

```
public:
    void addBike(const Bike& bike) {
        bikes.push_back(bike);
        std::cout << "Bike added successfully.\n";
    }

    void displayBikes() const {
        if (bikes.empty()) {
            std::cout << "No bikes available.\n";
            return;
        }
        for (const auto& bike : bikes) {
            std::cout << "Brand: " << bike.brand
                      << ", Model: " << bike.model
                      << ", Year: " << bike.year
                      << ", Price: $" << bike.price << "\n";
        }
    }
```

```
void deleteBike(int index) {
        if (index < 0 || index >= bikes.size()) {
            std::cout << "Invalid index.\n";
            return;
        }
        bikes.erase(bikes.begin() + index);
        std::cout << "Bike deleted successfully.\n";
    }
};

int main() {
    BikeManagementSystem system;
    int choice;

    do {
        std::cout << "\nBike Information Management
System\n";
        std::cout << "1. Add Bike\n";
        std::cout << "2. Display Bikes\n";
        std::cout << "3. Delete Bike\n";
        std::cout << "4. Exit\n";
```

```
std::cout << "Enter your choice: ";
    std::cin >> choice;

    switch (choice) {
        case 1: {
            std::string brand, model;
            int year;
            float price;
            std::cout << "Enter brand: ";
            std::cin >> brand;
            std::cout << "Enter model: ";
            std::cin >> model;
            std::cout << "Enter year: ";
            std::cin >> year;
            std::cout << "Enter price: ";
            std::cin >> price;
            system.addBike(Bike(brand, model, year, price));
            break;
        }
        case 2:
            system.displayBikes();
            break;
case 3: {
                int index;
                std::cout << "Enter bike index to delete: ";
                std::cin >> index;
                system.deleteBike(index);
                break;
            }
            case 4:
                std::cout << "Exiting...\n";
                break;
            default:
                std::cout << "Invalid choice. Please try
again.\n";
        }
    } while (choice != 4);

    return 0;
}
```

OUTPUT:



```
Bike Information Management System
1. Add Bike
2. Display Bikes
3. Delete Bike
4. Exit
Enter your choice: 1
Enter brand: hero
Enter model: splender
Enter year: 2024
Enter price: 90000
Bike added successfully.

Bike Information Management System
1. Add Bike
2. Display Bikes
3. Delete Bike
4. Exit
Enter your choice:
```

CONCLUSION

The Bike Information System project successfully demonstrates a practical application of C++ programming concepts, showcasing how software can facilitate the management of bike-related data. By providing a user-friendly interface for adding, displaying, and searching bike information, this project highlights the importance of structured data handling in software development.Through the implementation of key features, users can efficiently manage their bike records, which is beneficial for both individual cyclists and bike retailers. The use of object-oriented principles, such as structures and classes, not only enhances code organization but also promote and ability and maintainability.