BIKE RENT APPLICATION

TEAM MEMBERS :

V.LEKHYA SREEYA	*(22BFA12192)*
V.S.HIMABINDU	*(22BFA12190)*
S. MUSKAN BHANU	*(22BFA12181)*

INTRODUCTION

- The Bike Rental System is a console-based application designed to provide an efficient and user-friendly platform for renting bikes.

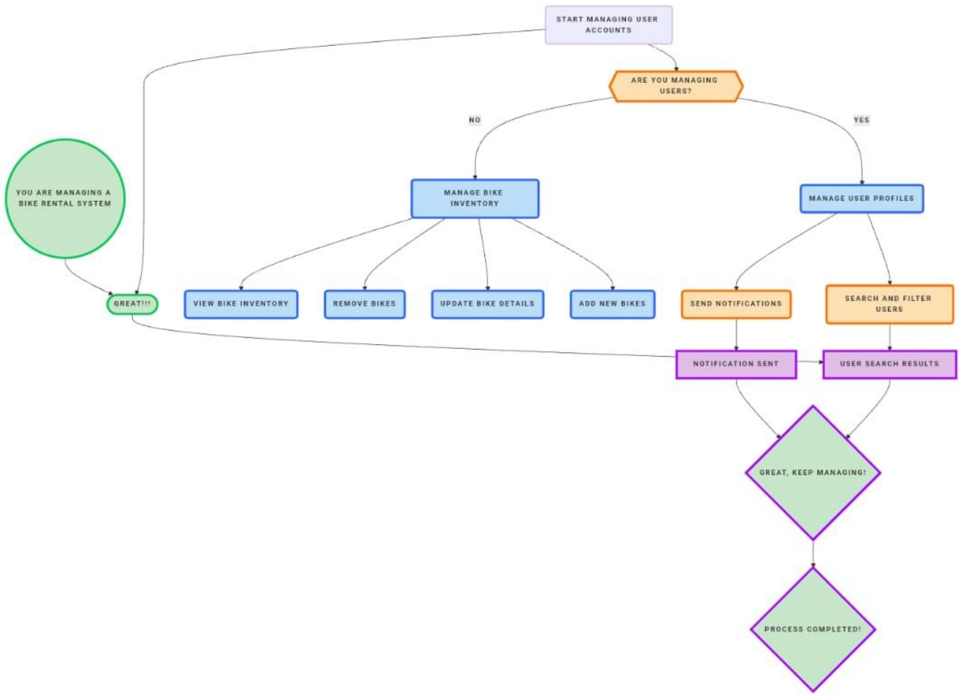
- Built using C programming, the system follows object-oriented principles to manage bike inventory, user rentals, and administrative functions seamlessly.

- The program allows users to register, search for available bikes, rent them for a specified duration, and return them after use.

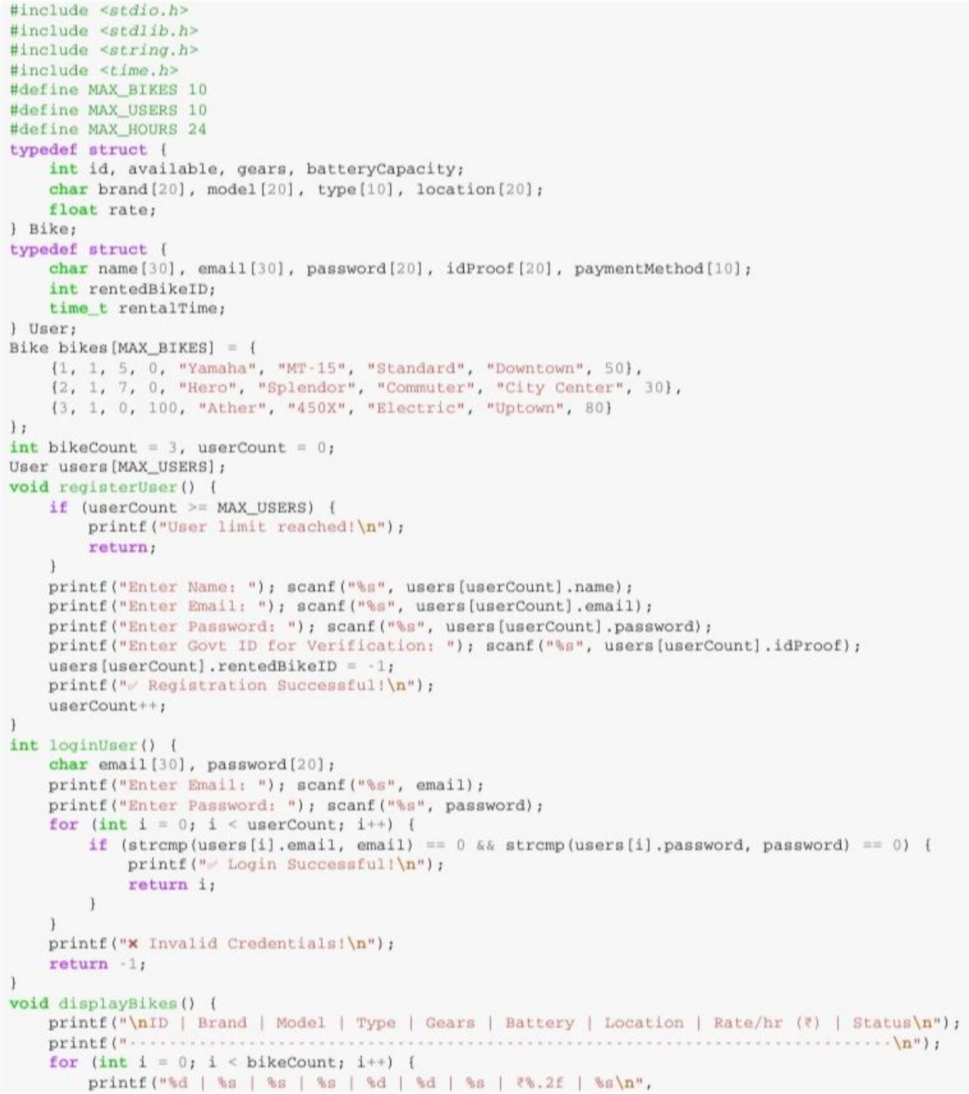
- It also includes an admin panel for managing the bike inventory, tracking rentals, and maintaining records.

KEY FEATURES

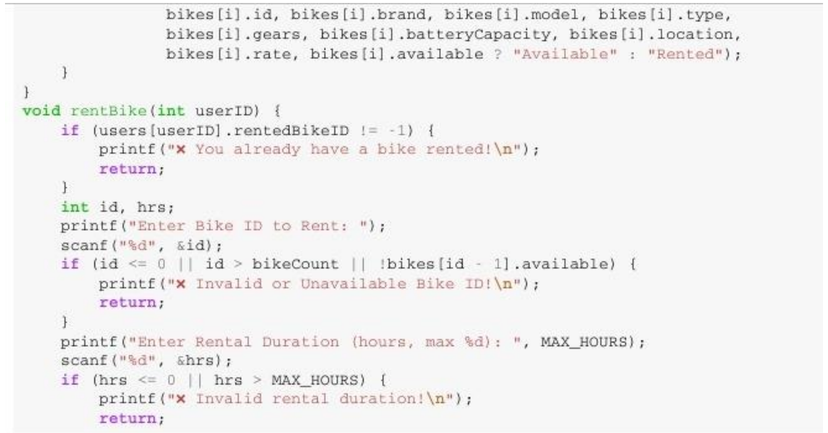
- ***User Management***
- ***Bike Inventory Management***
- ***Rental and Payment System***
- ***Search and Filter Options***
- ***Error Handling and Notifications***
- ***Admin Panel***



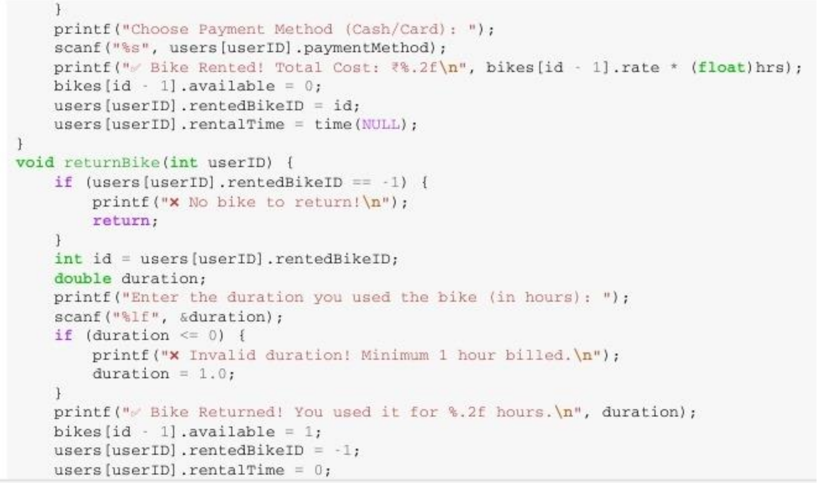
SOURCE CODE



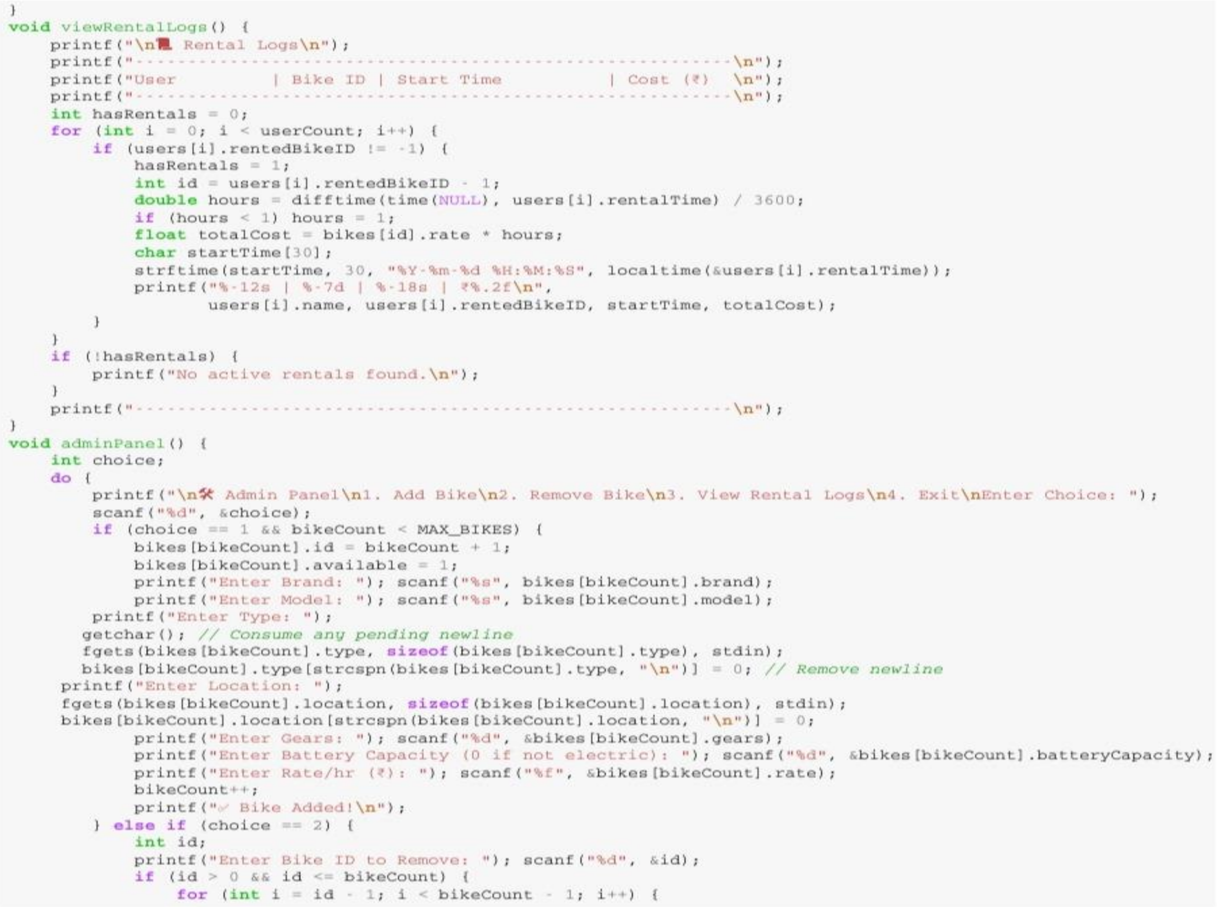
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#define MAX_BIKES 10
#define MAX_USERS 10
#define MAX_HOURS 24
typedef struct {
    int id, available, gears, batteryCapacity;
    char brand[20], model[20], type[10], location[20];
    float rate;
} Bike;
typedef struct {
    char name[30], email[30], password[20], idProof[20], paymentMethod[10];
    int rentedBikeID;
    time_t rentalTime;
} User;
Bike bikes[MAX_BIKES] = {
    {1, 1, 5, 0, "Yamaha", "MT-15", "Standard", "Downtown", 50},
    {2, 1, 7, 0, "Hero", "Splendor", "Commuter", "City Center", 30},
    {3, 1, 0, 100, "Ather", "450X", "Electric", "Uptown", 80}
};
int bikeCount = 3, userCount = 0;
User users[MAX_USERS];
void registerUser() {
    if (userCount >= MAX_USERS) {
        printf("User limit reached!\n");
        return;
    }
    printf("Enter Name: "); scanf("%s", users[userCount].name);
    printf("Enter Email: "); scanf("%s", users[userCount].email);
    printf("Enter Password: "); scanf("%s", users[userCount].password);
    printf("Enter Govt ID for Verification: "); scanf("%s", users[userCount].idProof);
    users[userCount].rentedBikeID = -1;
    printf("✓ Registration Successful!\n");
    userCount++;
}
int loginUser() {
    char email[30], password[20];
    printf("Enter Email: "); scanf("%s", email);
    printf("Enter Password: "); scanf("%s", password);
    for (int i = 0; i < userCount; i++) {
        if (strcmp(users[i].email, email) == 0 && strcmp(users[i].password, password) == 0) {
            printf("✓ Login Successful!\n");
            return i;
        }
    }
    printf("× Invalid Credentials!\n");
    return -1;
}
void displayBikes() {
    printf("\nID | Brand | Model | Type | Gears | Battery | Location | Rate/hr (₹) | Status\n");
    printf("--------------------------------------------------------------------------------\n");
    for (int i = 0; i < bikeCount; i++) {
        printf("%d | %s | %s | %s | %d | %d | %s | ₹%.2f | %s\n",
```



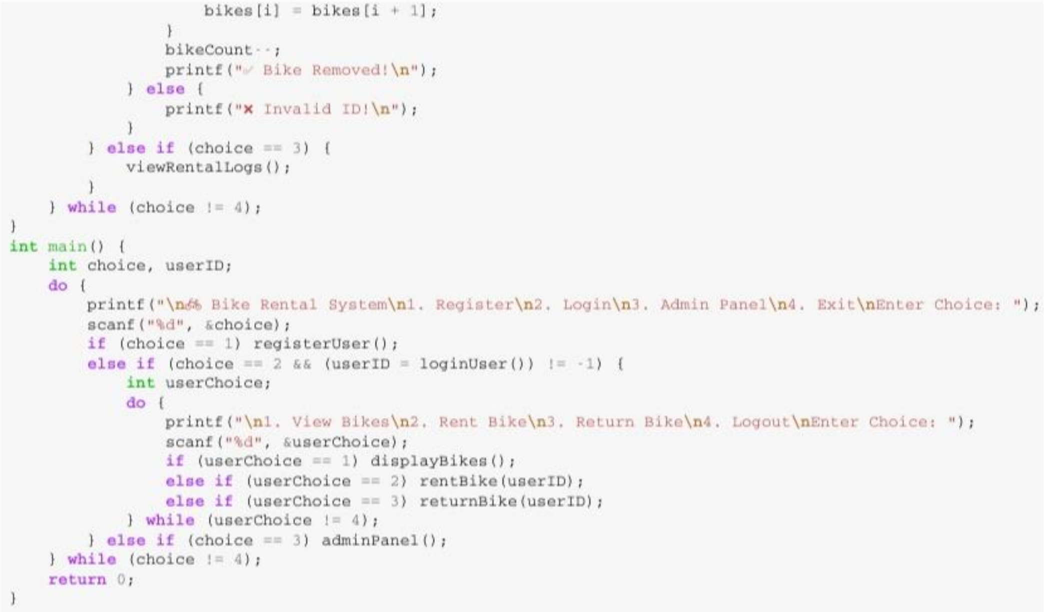
```c
               bikes[i].id, bikes[i].brand, bikes[i].model, bikes[i].type,
               bikes[i].gears, bikes[i].batteryCapacity, bikes[i].location,
               bikes[i].rate, bikes[i].available ? "Available" : "Rented");
    }
}
void rentBike(int userID) {
    if (users[userID].rentedBikeID != -1) {
        printf("× You already have a bike rented!\n");
        return;
    }
    int id, hrs;
    printf("Enter Bike ID to Rent: ");
    scanf("%d", &id);
    if (id <= 0 || id > bikeCount || !bikes[id - 1].available) {
        printf("× Invalid or Unavailable Bike ID!\n");
        return;
    }
    printf("Enter Rental Duration (hours, max %d): ", MAX_HOURS);
    scanf("%d", &hrs);
    if (hrs <= 0 || hrs > MAX_HOURS) {
        printf("× Invalid rental duration!\n");
        return;
```



```c
    }
    printf("Choose Payment Method (Cash/Card): ");
    scanf("%s", users[userID].paymentMethod);
    printf("✓ Bike Rented! Total Cost: ₹%.2f\n", bikes[id - 1].rate * (float)hrs);
    bikes[id - 1].available = 0;
    users[userID].rentedBikeID = id;
    users[userID].rentalTime = time(NULL);
}
void returnBike(int userID) {
    if (users[userID].rentedBikeID == -1) {
        printf("× No bike to return!\n");
        return;
    }
    int id = users[userID].rentedBikeID;
    double duration;
    printf("Enter the duration you used the bike (in hours): ");
    scanf("%lf", &duration);
    if (duration <= 0) {
        printf("× Invalid duration! Minimum 1 hour billed.\n");
        duration = 1.0;
    }
    printf("✓ Bike Returned! You used it for %.2f hours.\n", duration);
    bikes[id - 1].available = 1;
    users[userID].rentedBikeID = -1;
    users[userID].rentalTime = 0;
```

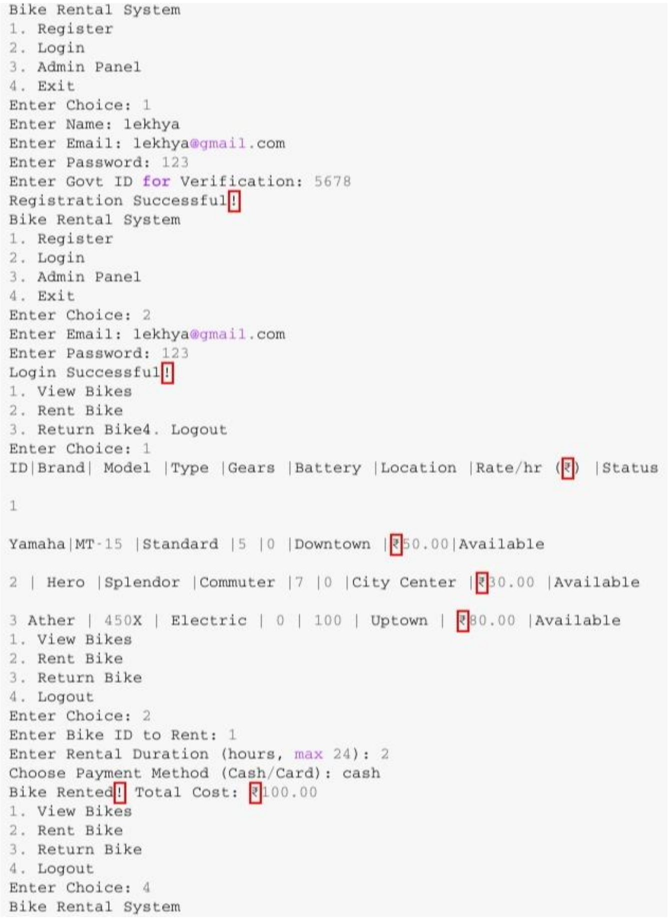


```c
}
void viewRentalLogs() {
    printf("\n📜 Rental Logs\n");
    printf("------------------------------------------------------------\n");
    printf("User          | Bike ID | Start Time           | Cost (₹)  \n");
    printf("------------------------------------------------------------\n");
    int hasRentals = 0;
    for (int i = 0; i < userCount; i++) {
        if (users[i].rentedBikeID != -1) {
            hasRentals = 1;
            int id = users[i].rentedBikeID - 1;
            double hours = difftime(time(NULL), users[i].rentalTime) / 3600;
            if (hours < 1) hours = 1;
            float totalCost = bikes[id].rate * hours;
            char startTime[30];
            strftime(startTime, 30, "%Y-%m-%d %H:%M:%S", localtime(&users[i].rentalTime));
            printf("%-12s | %-7d | %-18s | ₹%.2f\n",
                   users[i].name, users[i].rentedBikeID, startTime, totalCost);
        }
    }
    if (!hasRentals) {
        printf("No active rentals found.\n");
    }
    printf("------------------------------------------------------------\n");
}
void adminPanel() {
    int choice;
    do {
        printf("\n🛠 Admin Panel\n1. Add Bike\n2. Remove Bike\n3. View Rental Logs\n4. Exit\nEnter Choice: ");
        scanf("%d", &choice);
        if (choice == 1 && bikeCount < MAX_BIKES) {
            bikes[bikeCount].id = bikeCount + 1;
            bikes[bikeCount].available = 1;
            printf("Enter Brand: "); scanf("%s", bikes[bikeCount].brand);
            printf("Enter Model: "); scanf("%s", bikes[bikeCount].model);
        printf("Enter Type: ");
       getchar(); // Consume any pending newline
       fgets(bikes[bikeCount].type, sizeof(bikes[bikeCount].type), stdin);
       bikes[bikeCount].type[strcspn(bikes[bikeCount].type, "\n")] = 0; // Remove newline
     printf("Enter Location: ");
     fgets(bikes[bikeCount].location, sizeof(bikes[bikeCount].location), stdin);
     bikes[bikeCount].location[strcspn(bikes[bikeCount].location, "\n")] = 0;
            printf("Enter Gears: "); scanf("%d", &bikes[bikeCount].gears);
            printf("Enter Battery Capacity (0 if not electric): "); scanf("%d", &bikes[bikeCount].batteryCapacity);
            printf("Enter Rate/hr (₹): "); scanf("%f", &bikes[bikeCount].rate);
            bikeCount++;
            printf("✅ Bike Added!\n");
        } else if (choice == 2) {
            int id;
            printf("Enter Bike ID to Remove: "); scanf("%d", &id);
            if (id > 0 && id <= bikeCount) {
                for (int i = id - 1; i < bikeCount - 1; i++) {
```

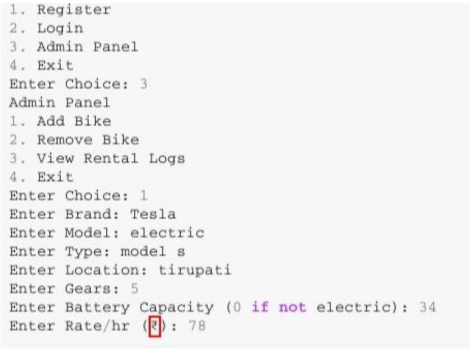


```c
                    bikes[i] = bikes[i + 1];
                }
                bikeCount--;
                printf("✅ Bike Removed!\n");
            } else {
                printf("❌ Invalid ID!\n");
            }
        } else if (choice == 3) {
            viewRentalLogs();
        }
    } while (choice != 4);
}
int main() {
    int choice, userID;
    do {
        printf("\n🚲 Bike Rental System\n1. Register\n2. Login\n3. Admin Panel\n4. Exit\nEnter Choice: ");
        scanf("%d", &choice);
        if (choice == 1) registerUser();
        else if (choice == 2 && (userID = loginUser()) != -1) {
            int userChoice;
            do {
                printf("\n1. View Bikes\n2. Rent Bike\n3. Return Bike\n4. Logout\nEnter Choice: ");
                scanf("%d", &userChoice);
                if (userChoice == 1) displayBikes();
                else if (userChoice == 2) rentBike(userID);
                else if (userChoice == 3) returnBike(userID);
            } while (userChoice != 4);
        } else if (choice == 3) adminPanel();
    } while (choice != 4);
    return 0;
}
```

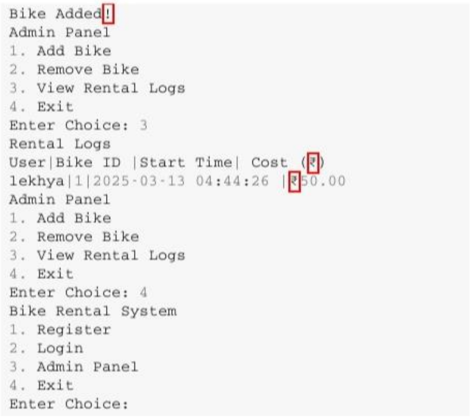
OUTPUT



```
Bike Rental System
1. Register
2. Login
3. Admin Panel
4. Exit
Enter Choice: 1
Enter Name: lekhya
Enter Email: lekhya@gmail.com
Enter Password: 123
Enter Govt ID for Verification: 5678
Registration Successful!
Bike Rental System
1. Register
2. Login
3. Admin Panel
4. Exit
Enter Choice: 2
Enter Email: lekhya@gmail.com
Enter Password: 123
Login Successful!
1. View Bikes
2. Rent Bike
3. Return Bike4. Logout
Enter Choice: 1
ID|Brand| Model |Type |Gears |Battery |Location |Rate/hr (₹) |Status

1

Yamaha|MT-15 |Standard |5 |0 |Downtown |₹50.00|Available

2 | Hero |Splendor |Commuter |7 |0 |City Center |₹30.00 |Available

3 Ather | 450X | Electric | 0 | 100 | Uptown | ₹80.00 |Available
1. View Bikes
2. Rent Bike
3. Return Bike
4. Logout
Enter Choice: 2
Enter Bike ID to Rent: 1
Enter Rental Duration (hours, max 24): 2
Choose Payment Method (Cash/Card): cash
Bike Rented! Total Cost: ₹100.00
1. View Bikes
2. Rent Bike
3. Return Bike
4. Logout
Enter Choice: 4
Bike Rental System
```



```
1. Register
2. Login
3. Admin Panel
4. Exit
Enter Choice: 3
Admin Panel
1. Add Bike
2. Remove Bike
3. View Rental Logs
4. Exit
Enter Choice: 1
Enter Brand: Tesla
Enter Model: electric
Enter Type: model s
Enter Location: tirupati
Enter Gears: 5
Enter Battery Capacity (0 if not electric): 34
Enter Rate/hr (₹): 78
```



```
Bike Added!
Admin Panel
1. Add Bike
2. Remove Bike
3. View Rental Logs
4. Exit
Enter Choice: 3
Rental Logs
User|Bike ID |Start Time| Cost (₹)
lekhya|1|2025-03-13 04:44:26 |₹50.00
Admin Panel
1. Add Bike
2. Remove Bike
3. View Rental Logs
4. Exit
Enter Choice: 4
Bike Rental System
1. Register
2. Login
3. Admin Panel
4. Exit
Enter Choice:
```

CONCLUSION

- The Bike Rental System is a well-structured, efficient, and user-friendly application that streamlines the bike rental process.
- It includes essential features such as user management, bike inventory control, rental and payment processing, search and filtering, error handling, and an admin panel

THANK YOU