CAMERA RENTAL APPLICATION

Main.java

```
package camerarental;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        CameraRentalApp cameraRentalApp = new
CameraRentalApp();
        cameraRentalApp.displayWelcomeScreen();
        cameraRentalApp.login();
    }
}
```

CameraRentalApp.java

```
package camerarental;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;


public class CameraRentalApp {

    private List<Camera> cameraList = new ArrayList<>();

    private double walletAmount = 0.0;

    private String username = "admin";

    private String password = "admin123";


    public void displayWelcomeScreen() {

        System.out.println("======WELCOME TO CAMERA RENTAL
APP==========\n");

    }


    public void login() {

        Scanner scanner = new Scanner(System.in);


        System.out.println("PLEASE LOGIN TO CONTINUE...");

        System.out.print("USERNAME :");

        String enteredUsername = scanner.nextLine();

        System.out.print("PASSWORD :");

        String enteredPassword = scanner.nextLine();
```

```
        if (enteredUsername.equals(username) &&
enteredPassword.equals(password)) {

            System.out.println("\nLogin successful!");

            showMainMenu(scanner);

        } else {

            System.out.println("Invalid username or password.
Exiting the application.");

        }

    }



    public void showMainMenu(Scanner scanner) {

        int choice;



        do {

            System.out.println("\nMAIN MENU:");

            System.out.println("1. My Cameras");

            System.out.println("2. Rent a Camera");

            System.out.println("3. View All Cameras");

            System.out.println("4. My Wallet");

            System.out.println("5. Exit");



            System.out.print("\nEnter your choice: ");

            choice = scanner.nextInt();

            scanner.nextLine(); // consume newline character
```

```
            switch (choice) {
                case 1:
                    MyCamerasMenu myCamerasMenu = new
MyCamerasMenu(cameraList);

myCamerasMenu.handleMyCamerasMenu(scanner);
                    break;
                case 2:
                    rentCamera(scanner);
                    break;
                case 3:
                    listCameras();
                    break;
                case 4:
                    handleWalletMenu(scanner);
                    break;
                case 5:
                    System.out.println("Thank you for using
the Camera Rental App. Goodbye!");
                    System.exit(0);
                default:
                    System.out.println("Invalid choice. Please
try again.");
            }
        } while (choice != 5);
    }
```

```
    public void rentCamera(Scanner scanner) {
        if (cameraList.isEmpty()) {
            System.out.println("No cameras available for rent
at the moment.");
            return;
        }

        System.out.println("\nAVAILABLE CAMERAS:");
        listCameras();

        System.out.print("\nEnter the camera ID you want to
rent: ");
        int cameraId = scanner.nextInt();
        scanner.nextLine(); // consume newline character

        boolean cameraFound = false;
        for (Camera camera : cameraList) {
            if (camera.getId() == cameraId) {
                if (camera.isRented()) {
                    System.out.println("Camera is already
rented.");
                } else {
                    double rentalAmount =
camera.getRentalAmount();
                    if (rentalAmount <= walletAmount) {
                        walletAmount -= rentalAmount;
                        camera.setRented(true);
```

```
                        System.out.println("Your transaction
for camera " + camera.getBrand() + " " +

                                camera.getModel() + " with
rent INR " + rentalAmount + " has been successfully
completed.");

                    } else {

                        System.out.println("Insufficient
balance in your wallet.");

                    }

                }

                cameraFound = true;

                break;

            }

        }



        if (!cameraFound) {

            System.out.println("Invalid camera ID.");

        }

    }



    public void handleWalletMenu(Scanner scanner) {

        System.out.println("\nMY WALLET:");

        System.out.println("Your current wallet balance is INR
" + walletAmount);



        System.out.print("\nDo you want to deposit more amount
to your wallet? (1. Yes, 2. No): ");

        int choice = scanner.nextInt();
```

```
        scanner.nextLine(); // consume newline character


        switch (choice) {
            case 1:
                System.out.print("Enter the amount (INR): ");
                double depositAmount = scanner.nextDouble();
                scanner.nextLine(); // consume newline
character
                walletAmount += depositAmount;
                System.out.println("Your wallet balance has
been updated successfully. Current wallet balance: INR " +
walletAmount);
                break;
            case 2:
                break;
            default:
                System.out.println("Invalid choice.");
        }
    }


    public void listCameras() {
        if (cameraList.isEmpty()) {
            System.out.println("No cameras available for rent
at the moment.");
        } else {

System.out.println("==================================================
========================");
```

```
            System.out.println("CAMERA ID    BRAND
MODEL           PRICE (PER DAY)      STATUS");


System.out.println("==========================================
========================");

            for (Camera camera : cameraList) {

                System.out.printf("%-13d%-14s%-16s%-
22.2f%s\n",

                        camera.getId(), camera.getBrand(),
camera.getModel(), camera.getRentalAmount(),

                        camera.isRented() ? "Rented" :
"Available");

            }


System.out.println("==========================================
========================");

        }

    }

}
```

Camera.java

```
package camerarental;

public class Camera {
    private static int nextId = 1;

    private int id;
    private String brand;
    private String model;
    private double rentalAmount;
    private boolean rented;

    public Camera(String brand, String model, double
rentalAmount) {
        this.id = nextId++;
        this.brand = brand;
        this.model = model;
        this.rentalAmount = rentalAmount;
        this.rented = false;
    }

    public int getId() {
        return id;
    }

    public String getBrand() {
        return brand;
    }

    public String getModel() {
        return model;
    }

    public double getRentalAmount() {
        return rentalAmount;
    }

    public boolean isRented() {
        return rented;
    }

    public void setRented(boolean rented) {
        this.rented = rented;
    }
}
```

MyCamerasMenu.java

```
package camerarental;
import java.util.List;
import java.util.Scanner;
public class MyCamerasMenu {
    private List<Camera> cameraList;

    public MyCamerasMenu(List<Camera> cameraList) {
        this.cameraList = cameraList;
    }
    public void handleMyCamerasMenu(Scanner scanner) {
        int choice;

        do {
            System.out.println("\nMY CAMERAS MENU:");
            System.out.println("1. Add Camera");
            System.out.println("2. Remove Camera");
            System.out.println("3. View My Cameras");
            System.out.println("4. Go to Previous Menu");
            System.out.print("\nEnter your choice: ");
            choice = scanner.nextInt();
            scanner.nextLine(); // consume newline character

            switch (choice) {
```

```
                case 1:
                    addCamera(scanner);
                    break;
                case 2:
                    removeCamera(scanner);
                    break;
                case 3:
                    viewMyCameras();
                    break;
                case 4:
                    return;
                default:
                    System.out.println("Invalid choice. Please
try again.");
            }
        } while (choice != 4);
    }

    private void addCamera(Scanner scanner) {
        System.out.print("\nEnter the camera brand: ");
        String brand = scanner.nextLine();
        System.out.print("Enter the camera model: ");
        String model = scanner.nextLine();
        System.out.print("Enter the per day price (INR): ");
        double rentalAmount = scanner.nextDouble();
        scanner.nextLine(); // consume newline character
```

```
        Camera camera = new Camera(brand, model,
rentalAmount);

        cameraList.add(camera);



        System.out.println("Your camera has been successfully
added to the list.");

    }



    private void removeCamera(Scanner scanner) {

        System.out.println("\nMY CAMERAS:");

        viewMyCameras();

        System.out.print("Enter the camera ID to remove: ");

        int cameraId = scanner.nextInt();

        scanner.nextLine(); // consume newline character

        boolean cameraRemoved = false;

        for (Camera camera : cameraList) {

            if (camera.getId() == cameraId) {

                cameraList.remove(camera);

                cameraRemoved = true;

                break;

            }

        }

        if (cameraRemoved) {

            System.out.println("Camera successfully removed
from the list.");

        } else {
```

```
            System.out.println("Invalid camera ID.");

        }

    }

private void viewMyCameras() {

System.out.println("\nMY CAMERAS:");


System.out.println("==========================================
========================");

        System.out.println("CAMERA ID    BRAND         MODEL
PRICE (PER DAY)      STATUS");


System.out.println("==========================================
========================");

        for (Camera camera : cameraList) {

            System.out.printf("%-13d%-14s%-16s%-22.2f%s\n",

                    camera.getId(), camera.getBrand(),
camera.getModel(), camera.getRentalAmount(),

                    camera.isRented() ? "Rented" :
"Available");

        }


System.out.println("==========================================
========================");

    }

}
```