Phase1: **Camera Project**

: ***Source Code*** :

```
package com.camera.rental;
        import java.util.ArrayList;
        import java.util.List;
        import java.util.Scanner;


        class Camera {
            private String brand;
            private String model;
            private double perDayPrice;
            private boolean isRented;

            public Camera(String brand, String model, double perDayPrice) {
                this.brand = brand;
                this.model = model;
                this.perDayPrice = perDayPrice;
                this.isRented = false;
            }

            public String getBrand() {
                return brand;
            }

            public String getModel() {
                return model;
            }

            public double getPerDayPrice() {
                return perDayPrice;
            }

            public boolean isRented() {
                return isRented;
            }

            public void setRented(boolean rented) {
                isRented = rented;
            }

            @Override
            public String toString() {
                return brand + "  " + model + " - " + perDayPrice;
            }
        }

        class Wallet {
            private double balance;
```

```java
    public Wallet() {
        this.balance = 0.0;
    }

    public double getBalance() {
        return balance;
    }

    public void deposit(double amount) {
        balance += amount;
    }

    public boolean withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            return true;
        }
        return false;
    }
}

public class CameraRentalApp {
    private static List<Camera> cameraList = new ArrayList<>();
    private static Wallet wallet = new Wallet();
      private static Scanner sc;

    public static void main(String[] args) {
        displayWelcomeScreen();

        boolean exit = false;
        Scanner scanner = new Scanner(System.in);

        while (!exit) {
            int choice = getUserChoice(scanner);
            switch (choice) {
                case 1:
                    manageMyCamera(scanner);
                    break;
                case 2:
                    rentCamera(scanner);
                    break;
                case 3:
                    viewAllCameras();
                    break;
                case 4:
                    manageWallet(scanner);
                    break;
                case 5:
                    exit = true;
                    System.out.println("Exiting the application...");
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        }
        scanner.close();
    }

    private static void displayWelcomeScreen() {
       System.out.println("+------------------------------------------+");
        System.out.println("|   WELCOME TO THE RENTAL CAMERA APP   |");
        System.out.println("+------------------------------------------+");
        System.out.println("PLEASE LOGIN TO CONTINUE -");
        sc = new Scanner(System.in);
        String str1=sc.next();
        String str2=sc.next();
        System.out.println("USERNAME -"+str1);
        System.out.println("PASSWORD -"+str2);
        System.out.println("1. MY CAMERA");
        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");
    }

    private static int getUserChoice(Scanner scanner) {
        System.out.print("Enter your choice: ");
        return scanner.nextInt();
    }

    private static void manageMyCamera(Scanner scanner) {
        boolean backToMain = false;

        while (!backToMain) {
            System.out.println("\n1. ADD");
            System.out.println("2. REMOVE");
            System.out.println("3. VIEW MY CAMERAS");
            System.out.println("4. GO TO PREVIOUS MENU");

            int choice = getUserChoice(scanner);
            switch (choice) {
                case 1:
                    addCamera(scanner);
                    break;
                case 2:
                    removeCamera(scanner);
                    break;
                case 3:
                    viewMyCameras();
                    break;
                case 4:
                    backToMain = true;
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        }
    }
    private static void addCamera(Scanner scanner) {
        System.out.print("Enter the camera brand: ");
        String brand = scanner.next();
        System.out.print("Enter the camera model: ");
        String model = scanner.next();
        System.out.print("Enter the per day price (INR): ");
        double perDayPrice = scanner.nextDouble();

        Camera camera = new Camera(brand, model, perDayPrice);
        cameraList.add(camera);

        System.out.println("YOUR CAMERA HAS BEEN SUCCESSFULLY ADDED TO THE
LIST.");
    }

    private static void removeCamera(Scanner scanner) {
        viewMyCameras();

        System.out.print("Enter the camera ID to remove: ");
        int cameraId = scanner.nextInt();

        if (cameraId >= 0 && cameraId < cameraList.size()) {
            cameraList.remove(cameraId);
            System.out.println("Camera successfully removed from the list.");
        } else {
            System.out.println("Invalid camera ID.");
        }
    }

    private static void viewMyCameras() {
        if (cameraList.isEmpty()) {
            System.out.println("No cameras present at this moment.");
        } else {
            System.out.printf("%-10s %-10s %-10s %-10s %-10s\n",
                    "CAMERA ID", "BRAND", "MODEL", "PRICE", "STATUS");
            int id = 0;
            for (Camera camera : cameraList) {
                System.out.printf("%-10s %-10s %-10s %-10.2f %-10s\n",
                        id++, camera.getBrand(), camera.getModel(),
                        camera.getPerDayPrice(), camera.isRented() ? "Rented"
: "Available");
            }
        }
    }

    private static void rentCamera(Scanner scanner) {
        viewAllCameras();

        if (cameraList.isEmpty()) {
            System.out.println("No cameras available for rent at this
moment.");
            return;
        }

        System.out.print("Enter the camera ID you want to rent: ");
```

```java
                int cameraId = scanner.nextInt();

                if (cameraId >= 0 && cameraId < cameraList.size()) {
                    Camera camera = cameraList.get(cameraId);
                    if (camera.isRented()) {
                        System.out.println("Camera is already rented.");
                    } else {
                        if (wallet.getBalance() >= camera.getPerDayPrice()) {
                            wallet.withdraw(camera.getPerDayPrice());
                            camera.setRented(true);
                            System.out.println("Camera rented successfully.");
                        } else {
                            System.out.println("Insufficient wallet balance. Please 
deposit the amount to your wallet.");
                        }
                    }
                } else {
                    System.out.println("Invalid camera ID.");
                }
            }

            private static void viewAllCameras() {
                System.out.println("\nFOLLOWING IS THE LIST OF AVAILABLE 
CAMERA(S)\n");
                if (cameraList.isEmpty()) {
                    System.out.println("No cameras available at this moment.");
                } else {
                    System.out.printf("%-10s %-10s %-10s %-10s %-10s\n",
                            "CAMERA ID", "BRAND", "MODEL", "PRICE", "STATUS");
                    int id = 0;
                    for (Camera camera : cameraList) {
                        System.out.printf("%-10s %-10s %-10s %-10.2f %-10s\n",
                                id++, camera.getBrand(), camera.getModel(),
                                camera.getPerDayPrice(), camera.isRented() ? "Rented" 
: "Available");
                    }
                }
            }

            private static void manageWallet(Scanner scanner) {
                System.out.println("\nMY WALLET\n");
                System.out.printf("Your current wallet balance is INR %.2f\n", 
wallet.getBalance());

                System.out.println("Do you want to deposit more amount to your 
wallet?");
                System.out.println("1. Yes");
                System.out.println("2. No");

                int choice = scanner.nextInt();

                switch (choice) {
                    case 1:
                        System.out.print("Enter the amount (INR): ");
                        double amount = scanner.nextDouble();
                        wallet.deposit(amount);
                        System.out.printf("Your wallet balance updated successfully. 
Current wallet balance: INR %.2f\n", wallet.getBalance());
                        break;
                    case 2:
                        break;
                    default:
                        System.out.println("Invalid choice. Please try again.");
                }
            }

            @SuppressWarnings("unused")
                private static void exitApplication() {
                 System.out.println("Exiting the application... Goodbye!");
                 System.exit(0);
            }
        }
```