CAMERA RENTAL APPLICATION

SOURCE CODE:

```
package Assignedp;


import java.util.ArrayList;
import java.util.Scanner;

class Camera {
    int id;
    String brand;
    String model;
    double pricePerDay;
    String status;

    public Camera(int id, String brand, String model, double
pricePerDay, String status) {
        this.id = id;
        this.brand = brand;
        this.model = model;
        this.pricePerDay = pricePerDay;
        this.status = status;
    }
}

class Wallet {
    double balance;

    public Wallet(double balance) {
        this.balance = balance;
    }
}

class User {
    String username;
    String password;
    ArrayList<Camera> myCameras = new ArrayList<>();
    Wallet wallet;

    public User(String username, String password, double
initialBalance) {
        this.username = username;
        this.password = password;
        this.wallet = new Wallet(initialBalance);
    }
}

public class CameraRentalApp {
    static Scanner scanner = new Scanner(System.in);
    static User currentUser;

    public static void main(String[] args) {

        User demoUser = new User("bhavs1", "bhavs123", 10000);
        currentUser = demoUser;

        displayWelcomeScreen();
        login();
```

```
              mainMenu();
          }

          private static void displayWelcomeScreen() {
              System.out.println("+----------------------------------------
-------------------+");
              System.out.println("|           WELCOME TO CAMERA RENTAL APP
|");
              System.out.println("+----------------------------------------
-------------------+");
              System.out.println("PLEASE LOGIN TO CONTINUE -");
          }

          private static void login() {
              System.out.print("USERNAME - ");
              String username = scanner.nextLine();
              System.out.print("PASSWORD - ");
              String password = scanner.nextLine();

              if (username.equals(currentUser.username) &&
password.equals(currentUser.password)) {
                  System.out.println("Login successful!\n");
              } else {
                  System.out.println("Invalid credentials. Exiting...\n");
                  System.exit(0);
              }
          }

          private static void mainMenu() {
              while (true) {
                  System.out.println("**Options to be displayed:**");
                  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");

                  int choice = getIntInput("Enter your choice: ");

                  switch (choice) {
                      case 1:
                          myCameraMenu();
                          break;
                      case 2:
                          rentCamera();
                          break;
                      case 3:
                          viewAllCameras();
                          break;
                      case 4:
                          myWalletMenu();
                          break;
                      case 5:
                          System.out.println("Exiting...");
                          System.exit(0);
                      default:
                          System.out.println("Invalid choice. Please try
again.");
                  }
              }
          }
          private static void myCameraMenu() {
              while (true) {
                  System.out.println("1. ADD");
                  System.out.println("2. REMOVE");
                  System.out.println("3. VIEW ALL CAMERAS");
                  System.out.println("4. GO TO PREVIOUS MENU");

                  int choice = getIntInput("Enter your choice: ");

                  switch (choice) {
                      case 1:
                          addCamera();
                          break;
                      case 2:
                          removeCamera();
                          break;
                      case 3:
                          viewAllCameras();
                          break;
                      case 4:
                          return;
                      default:
                          System.out.println("Invalid choice. Please try
again.");
                  }
              }
          }

          private static void addCamera() {
             System.out.print("ENTER THE CAMERA BRAND - ");
                    String brand = scanner.next();

                    System.out.print("ENTER THE MODEL - ");
                    String model = scanner.next();

                    System.out.print("ENTER THE PER DAY PRICE (INR) - ");
                    double pricePerDay = scanner.nextDouble();
              int id = currentUser.myCameras.size() + 1;

              Camera newCamera = new Camera(id, brand, model, pricePerDay,
"Available");
              currentUser.myCameras.add(newCamera);

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

          private static void removeCamera() {
              viewAllCameras();
              int cameraId = getIntInput("ENTER THE CAMERA ID TO REMOVE -
");

              for (Camera camera : currentUser.myCameras) {
                  if (camera.id == cameraId) {
                      currentUser.myCameras.remove(camera);
                      System.out.println("CAMERA SUCCESSFULLY REMOVED FROM
THE LIST\n");
                      return;
                  }
              }
              System.out.println("Camera not found with ID: " + cameraId +
"\n");
          }

          private static void rentCamera() {
              viewAllCameras();
              int cameraId = getIntInput("ENTER THE CAMERA ID YOU WANT TO
RENT - ");

              for (Camera camera : currentUser.myCameras) {
                  if (camera.id == cameraId &&
camera.status.equals("Available")) {
                      double rentalAmount = camera.pricePerDay;
                      if (rentalAmount <= currentUser.wallet.balance) {
                          currentUser.wallet.balance -= rentalAmount;
                          camera.status = "Rented";
                          System.out.println("YOUR TRANSACTION FOR CAMERA -
" + camera.brand + " " + camera.model +
                                  " with rent INR." + rentalAmount + " HAS
SUCCESSFULLY COMPLETED\n");
                      } else {
                          System.out.println("ERROR : TRANSACTION FAILED
DUE TO INSUFFICIENT WALLET BALANCE." +
                                  " PLEASE DEPOSIT THE AMOUNT TO YOUR
WALLET\n");
                      }
                      return;
                  }
              }

              System.out.println("Camera not available for rent with ID: "
+ cameraId + "\n");
          }

          private static void viewAllCameras() {

System.out.println("=======================================================
==============================================");
              System.out.printf("%-15s%-30s%-25s%-20s%-20s\n", "CAMERA ID",
"BRAND", "MODEL", "PRICE(PER DAY)", "STATUS");

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

              for (Camera camera : currentUser.myCameras) {
                  System.out.printf("%-15d%-30s%-25s%-20.2f%-20s\n",
camera.id, camera.brand, camera.model,
                          camera.pricePerDay, camera.status);
              }


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

          private static void myWalletMenu() {
              System.out.println("YOUR CURRENT WALLET BALANCE IS - INR. " +
currentUser.wallet.balance);
```

```
              String depositChoice = getStringInput("DO YOU WANT TO DEPOSIT
MORE AMOUNT TO YOUR WALLET? (1.YES    2.NO) - ");
              if (depositChoice.equals("1")) {
                  double depositAmount = getDoubleInput("ENTER THE AMOUNT
(INR) - ");
                  currentUser.wallet.balance += depositAmount;
                  System.out.println("YOUR WALLET BALANCE UPDATED
SUCCESSFULLY. CURRENT WALLET BALANCE - INR. " +
                          currentUser.wallet.balance + "\n");
              }
          }

          private static int getIntInput(String prompt) {
              System.out.print(prompt);
              while (!scanner.hasNextInt()) {
                  System.out.println("Invalid input. Please enter a valid
integer.");
                  scanner.next(); // consume the invalid input
              }
              return scanner.nextInt();
          }

          private static double getDoubleInput(String prompt) {
              System.out.print(prompt);
              while (!scanner.hasNextDouble()) {
                  System.out.println("Invalid input. Please enter a valid
double.");
                  scanner.next(); // consume the invalid input
              }
              return scanner.nextDouble();
          }

          private static String getStringInput(String prompt)
          {
              System.out.print(prompt);
              return scanner.next();
          }
      }
```