CAMERA RENTAL APPLICATION CODE:

```
package LMS_Projects;

import java.util.*;

class Camera {
    private int cameraId;
    private String brand;
    private String model;
    boolean isAvailable;
    private double rentalPricePerDay;


    public Camera(int cameraId, String brand, String model, boolean isAvailable,  double rentalPricePerDay) {
        this.cameraId = cameraId;
        this.brand = brand;
        this.model = model;
        this.isAvailable = isAvailable;
        this.rentalPricePerDay = rentalPricePerDay;
    }

    public int getCameraId() {
        return cameraId;
    }

    public String getBrand() {
        return brand;
    }
    public String getModel() {
        return model;
    }

    public boolean isAvailable() {
        return isAvailable;
    }

    public double getRentalPricePerDay() {
        return rentalPricePerDay;
    }
}

class User {
    private String username;
    private String password;
    private double walletBalance;
    private List<Camera> myCameras;

    public User(String username, String password) {
        this.username = username;
        this.password = password;
        this.walletBalance = 0.0;
        this.myCameras = new ArrayList<>();
    }

    public String getUsername() {
        return username;
    }

    public boolean authenticate(String password) {
        return this.password.equals(password);
    }

    public double getWalletBalance() {
        return walletBalance;
    }

    public void depositToWallet(double amount) {
        walletBalance += amount;
    }

    public List<Camera> getMyCameras() {
        return myCameras;
    }

    public void addCamera(Camera camera) {
        myCameras.add(camera);
    }

    public void removeCamera(Camera camera) {
        myCameras.remove(camera);
    }
}

public class CameraRentalApplication {
    private static Scanner scanner = new Scanner(System.in);
    private static User loggedInUser;
    private static List<Camera> cameraList = new ArrayList<>();

    public static void main(String[] args) {
            cameraList.add(new Camera(1, "Canon", "EOS 5D Mark IV", true, 70.0));
        cameraList.add(new Camera(2, "Nikon", "D850", true, 60.0));
        cameraList.add(new Camera(3, "Sony", "A7 III", true, 50.0));
        cameraList.add(new Camera(4, "Fujifilm", "X-T4", false, 90.0));
        cameraList.add(new Camera(5, "Panasonic", "Lumix GH5", true, 70.0));



        login();
    }

    private static void login() {
            System.out.println("---------------------------------");
            System.out.println("Welcome to Camera Rental App");
            System.out.println("---------------------------------");
            System.out.print("Enter username: ");
        String username = scanner.nextLine();
        System.out.print("Enter password: ");
        String password = scanner.nextLine();

        if (username.equals("Admin") && password.equals("admin123")) {
            loggedInUser = new User(username, password);
            displayMainMenu();
        } else {
            System.out.println("Invalid credentials. Please try again.");
            login();
        }
    }

    private static void displayMainMenu() {
        System.out.println("-----------");
        System.out.println("Main Menu");
        System.out.println("-----------");
        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");
        System.out.print("Enter your choice: ");

        int choice = scanner.nextInt();
        scanner.nextLine(); // Consume newline character

        switch (choice) {
            case 1:
                displayMyCameraMenu();
                break;
            case 2:
                rentCamera();
                break;
            case 3:
                viewAllCameras();
                break;
            case 4:
                displayMyWalletMenu();
                break;
            case 5:
                System.out.println("Thank you for using the Camera Rental Application. Goodbye!");
                System.exit(0);
            default:
                System.out.println("Invalid choice. Please try again.");
                displayMainMenu();
        }
    }
```

```
private static void displayMyCameraMenu() {
    System.out.println("-----------");
    System.out.println("My Camera");
    System.out.println("-----------");
    System.out.println("1. Add a camera");
    System.out.println("2. Remove a camera");
    System.out.println("3. View my Cameras");
    System.out.println("4. Previous menu");
    System.out.print("Enter your choice: ");

    int choice = scanner.nextInt();
    scanner.nextLine(); // Consume newline character

    switch (choice) {
        case 1:
            addCameraToMyList();
            break;
        case 2:
            removeCameraFromMyList();
            break;
        case 3:
            viewMyCameras();
            break;
        case 4:
            displayMainMenu();
            break;
        default:
            System.out.println("Invalid choice. Please try again.");
            displayMyCameraMenu();
    }
```

```
}

private static void addCameraToMyList() {
        System.out.print("Enter the camera ID: ");
    int cameraId = scanner.nextInt();
    scanner.nextLine(); // Consume newline character

    System.out.print("Enter the brand: ");
    String brand = scanner.nextLine();

    System.out.print("Enter the model: ");
    String model = scanner.nextLine();

    System.out.print("Is the camera available? (true/false): ");
    boolean isAvailable = scanner.nextBoolean();
    scanner.nextLine(); // Consume newline character

    System.out.print("Enter the rental price per day: ");
    double rentalPricePerDay = scanner.nextDouble();
    scanner.nextLine(); // Consume newline character

    Camera newCamera = new Camera(cameraId, brand, model, isAvailable, rentalPricePerDay);
    cameraList.add(newCamera);
    System.out.println("Camera added to the list.");



    displayMyCameraMenu();
}

private static void removeCameraFromMyList() {
        System.out.print("Enter the camera ID to remove: ");
    int cameraId = scanner.nextInt();
    scanner.nextLine(); // Consume newline character

    Camera selectedCamera = findCameraById(cameraId);
    if (selectedCamera != null && cameraList.contains(selectedCamera)) {
        cameraList.remove(selectedCamera);
        loggedInUser.removeCamera(selectedCamera);
        System.out.println(selectedCamera.getBrand() + " " + selectedCamera.getModel() +
                " removed from the camera database.");
    } else {
        System.out.println("Invalid camera ID or the camera does not exist. Please try again.");
    }


    displayMyCameraMenu();
}

private static void viewMyCameras() {
          List<Camera> myCameras = loggedInUser.getMyCameras();

            if (myCameras.isEmpty()) {
                System.out.println("You have no cameras in your list.");
            } else {
                System.out.println("-----------");
                System.out.println("My Cameras");
                System.out.println("-----------");
                for (Camera camera : myCameras) {
                    System.out.println("Camera ID: " + camera.getCameraId() +
                            ", Brand: " + camera.getBrand() +
                            ", Model: " + camera.getModel());
                }
```

```
       }

       System.out.print("Enter the camera ID to add: ");
       int cameraId = scanner.nextInt();
       scanner.nextLine(); // Consume newline character

       Camera selectedCamera = findCameraById(cameraId);
       if (selectedCamera != null) {
         if (!selectedCamera.isAvailable()) {
           System.out.println("Camera not available for rent.");
         } else {
           loggedInUser.addCamera(selectedCamera);
           System.out.println(selectedCamera.getBrand() + " " + selectedCamera.getModel() +
               " added to your camera list.");
         }
       } else {
         System.out.println("Invalid camera ID. Please try again.");
       }

  displayMyCameraMenu();
}

private static void rentCamera() {
  System.out.println("-----------");
  System.out.println("Available Cameras");
  System.out.println("-----------");

  boolean availableCameras = false;
  for (Camera camera : cameraList) {
    if (camera.isAvailable()) {
      availableCameras = true;
```

```
                System.out.println("Camera ID: " + camera.getCameraId() +
                        ", Brand: " + camera.getBrand() +
                        ", Model: " + camera.getModel());
            }
        }

        if (!availableCameras) {
            System.out.println("No cameras available for rent at the moment.");
        } else {
            System.out.print("Enter the camera ID to rent: ");
            int cameraId = scanner.nextInt();
            scanner.nextLine(); // Consume newline character

            Camera selectedCamera = findCameraById(cameraId);
            if (selectedCamera != null && selectedCamera.isAvailable()) {
                double rentAmount = selectedCamera.getRentalPricePerDay();
                if (loggedInUser.getWalletBalance() >= rentAmount) {
                    loggedInUser.depositToWallet(-rentAmount);
                    selectedCamera = findCameraById(cameraId);
                    selectedCamera.isAvailable = false;
                    System.out.println("Your transaction for " + selectedCamera.getBrand() + " " +
selectedCamera.getModel() +
                            " is successfully completed.");
                } else {
                    System.out.println("Insufficient wallet balance. Please add money to your wallet.");
                }
            } else {
                System.out.println("Invalid camera ID or the camera is not available for rent. Please try
again.");
            }
        }
        displayMainMenu();
    }

    private static void viewAllCameras() {
           System.out.println("--------------------------------");
        System.out.println("All Cameras");
        System.out.println("--------------------------------");

        if (cameraList.isEmpty()) {
            System.out.println("No Data Present at This Moment.");
        } else {
            System.out.println("-------------------------------------------------------");
            System.out.printf("| %-10s | %-15s | %-15s | %-9s | %-9s |\n", "Camera ID", "Brand",
"Model", "Status", "Rental Price");
            System.out.println("-------------------------------------------------------");

            for (Camera camera : cameraList) {
                System.out.printf("| %-10s | %-15s | %-15s | %-9s | $%-8.2f |\n",
                        camera.getCameraId(), camera.getBrand(), camera.getModel(),
                        (camera.isAvailable() ? "Available" : "Not Available"), camera.getRentalPricePerDay());
            }
            System.out.println("-------------------------------------------------------");
        }



        displayMainMenu();
    }

    private static void displayMyWalletMenu() {
        System.out.println("-----------");
        System.out.println("My Wallet");
        System.out.println("-----------");
        System.out.println("1. View Wallet Balance");
        System.out.println("2. Add Amount to Wallet");
        System.out.println("3. Previous menu");
        System.out.print("Enter your choice: ");

        int choice = scanner.nextInt();
        scanner.nextLine(); // Consume newline character

        switch (choice) {
            case 1:
                viewWalletBalance();
                break;
            case 2:
                addAmountToWallet();
                break;
            case 3:
                displayMainMenu();
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
                displayMyWalletMenu();
        }
    }

    private static void viewWalletBalance() {
        System.out.println("Your wallet balance is: $" + loggedInUser.getWalletBalance());
        displayMyWalletMenu();
    }

    private static void addAmountToWallet() {
        System.out.print("Enter the amount to deposit: ");
        double amount = scanner.nextDouble();
        scanner.nextLine(); // Consume newline character

        loggedInUser.depositToWallet(amount);
        System.out.println("$" + amount + " added to your wallet.");
        displayMyWalletMenu();
    }

    private static Camera findCameraById(int cameraId) {
        for (Camera camera : cameraList) {
            if (camera.getCameraId() == cameraId) {
                return camera;
            }
        }
        return null;
    }
}
```