SOURCE CODE FOR CAMERA RENTAL APPLICATION

```
public class Camera {
        private String model;
        private String brand;
        private double price;
        private boolean available;
        public Camera(String model,String brand,double price,boolean available)
{
                this.model=model;
                this.brand=brand;
                this.price=price;
                this.available=available;


        }
        public String getModel() {
                return model;
        }
        public String getBrand() {
                return brand;
        }
        public double getPrice() {
                return price;
        }
        public boolean isAvailable() {
                return available;
        }
        public void setAvailable(boolean available) {
                this.available=available;
```

```
	}

}

import java.util.ArrayList;
import java.util.List;

public class User {
	private String username;
	private String password;
	private double wallet;
	private List<Camera> bookings;
	public User(String username,String password, double wallet) {
		this.username=username;
		this.password=password;
		this.wallet=wallet;
		this.bookings=new ArrayList<>();
	}
	public String getUsername() {
		return username;

	}
	public String getPassword() {
		return password;

	}
	public double getWallet() {
```

```
            return wallet;
        }
        public void setWallet(double wallet) {
            this.wallet=wallet;
        }
        public List<Camera> getBookings() {
            return bookings;

        }
        public void addBooking(Camera camera) {
            bookings.add(camera);
        }
}


import java.util.Scanner;
import java.util.List;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;


public class CameraRentalaApp {
   private List<Camera> cameras;
   private List<User> users;
   private User loggedInUser;
```

```
    public CameraRentalaApp() {
        cameras = new ArrayList<>();
        users = new ArrayList<>();
        loggedInUser = null;

        // Initialize some sample cameras
        cameras.add(new Camera("Camera1", "SONY", 10000.0, true));
        cameras.add(new Camera("Camera2", "NIKON", 15000.0, true));
        cameras.add(new Camera("Camera3", "CANON", 20000.0, true));
        cameras.add(new Camera("Camera4", "PANASONIC", 2000.0, true));
        cameras.add(new Camera("Camera5", "CANON 500", 20000.0, true));

        // Initialize some sample users
        users.add(new User("janishya", "jani@09", 10000.00));
    }

    public void displayMenu() {
        System.out.println("\n==== Camera Rental App Menu ====");
        System.out.println("1. List Cameras");
        System.out.println("2. Sort Cameras by Model");
        System.out.println("3. Sort Cameras by Price");
        System.out.println("4. Book a Camera");
        System.out.println("5. Display User Bookings");
        System.out.println("6. Log In");
        System.out.println("7. Log Out");
        System.out.println("8. Add Amount to Wallet");
        System.out.println("9. Exit");
        System.out.print("Enter your choice: ");
    }

    public void displayCameras() {
        System.out.println("Camera List:");
        for (int i = 0; i < cameras.size(); i++) {
            Camera camera = cameras.get(i);
            System.out.println(i + ". Model: " + camera.getModel() + ", Brand: " + camera.getBrand() + ", Price: $" + camera.getPrice() + ", Available: " + (camera.isAvailable() ? "Yes" : "No"));
        }
    }

    public void bookCamera(int cameraIndex) {
        if (cameraIndex >= 0 && cameraIndex < cameras.size()) {
            Camera camera = cameras.get(cameraIndex);
            if (camera.isAvailable()) {
                camera.setAvailable(false);
                loggedInUser.addBooking(camera);
                System.out.println("Camera booked successfully");
            } else {
                System.out.println("Camera is not available");
            }
        } else {
            System.out.println("Invalid camera index");
        }
    }
    public void displayUserBookings() {
        System.out.println("User Bookings:");
        List<Camera> bookings = loggedInUser.getBookings();
        if (bookings.isEmpty()) {
            System.out.println("No bookings found");
        } else {
            for (Camera camera : bookings) {
                System.out.println("- Model: " + camera.getModel() + ", Brand: " +
camera.getBrand() + ", Price: $" + camera.getPrice());
            }
        }
    }

    public void login(String username, String password) {
        for (User user : users) {
            if (user.getUsername().equals(username) &&
user.getPassword().equals(password)) {
                loggedInUser = user;
                System.out.println("Logged in as " + loggedInUser.getUsername());
                return;
            }
        }
        System.out.println("Invalid username or password");
    }

    public void run() {
        Scanner scanner = new Scanner(System.in);
        while (true) {
            displayMenu();
            int choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    displayCameras();
                    break;
                case 2:
                    // Sort cameras by model
                    // Implement your sorting logic here
                    System.out.println("Sorting by model");
Collections.sort(cameras,Comparator.comparing(Camera::getModel));
                    displayCameras();
                    break;
                case 3:
                    // Sort cameras by price
                    // Implement your sorting logic here
                    System.out.println("Sorting by price");
Collections.sort(cameras,Comparator.comparingDouble(Camera::getPrice));
                    displayCameras();
                    break;
                case 4:
                    if (loggedInUser == null) {
                        System.out.println("Please log in to book a camera");
```

```
    } else {
        displayCameras();
        System.out.print("Enter the camera index to book: ");
        int cameraIndex = scanner.nextInt();
        scanner.nextLine();
        bookCamera(cameraIndex);
    }
    break;
case 5:
    if (loggedInUser == null) {
        System.out.println("Please log in to view user bookings");
    } else {
        displayUserBookings();
    }
    break;
case 6:
    System.out.print("Enter username: ");
    String username = scanner.nextLine();
    System.out.print("Enter password: ");
    String password = scanner.nextLine();
    login(username, password);
    break;
case 7:
    loggedInUser = null;
    System.out.println("Logged out");
    break;
case 8:
```

```
                    if (loggedInUser == null) {
                        System.out.println("Please log in to add amount to wallet");
                    } else {
                        System.out.print("Enter amount to add to wallet: ");
                        double amount = scanner.nextDouble();
                        scanner.nextLine();
                        double currentWallet = loggedInUser.getWallet();
                        loggedInUser.setWallet(currentWallet + amount);
                        System.out.println("Amount added successfully. Current wallet balance: $" + loggedInUser.getWallet());
                    }
                    break;
                case 9:
                    System.out.println("Exiting the program. Goodbye!");
                    System.exit(0);
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
                    break;
            }
        }
    }

    public static void main(String[] args) {
        CameraRentalaApp app = new CameraRentalaApp();
        app.run();
    }
}
```