

# SmartCity Citizen Service Portal Project:

This Java-based SmartCity Citizen Service Portal allows citizens to register, log complaints, and track the status of their complaints. It also includes admin functionalities like viewing complaints and performing basic analytics. The project demonstrates the use of object- oriented principles, exception handling, input validation, data serialization, and stream processing.

## Key Functionalities:

### User Registration (Citizen and Admin):

- 1. Citizens and admins can register by providing details such as name, email, mobile, and password.
- 2. Passwords are encoded using Base64 for security.
- 3. Citizens can register complaints by providing a description.

## Complaint Submission and Tracking:

- 1. Citizens can submit complaints, and these complaints are stored in a list.
- 2. Complaints have a status, which can be updated (for simplicity, it is initially set as "Open").

## Data Serialization and Deserialization:

- 1. The list of complaints is serialized into a file (complaints.ser) so that data can be saved and reloaded, ensuring persistence across application runs.

## Phone Number Validation:

- 1. The mobile number is validated using a custom exception, InvalidPhoneException. The validation ensures that the number consists of exactly 10 digits.
- 2. Object-Oriented Programming (OOP)
- 3. Exception Handling
- 4. Data Persistence (Serialization)
- 5. Base64 Encodin

### Program:

#### Main.java (main file)

```
1 package com.smartcity;
2
3 import com.smartcity.model.Citizen;
4
5
6
7
8
9
10
11
12
13 public class Main {
14     private static List<Complaint> complaints = Collections.synchronizedList(new ArrayList<>());
15     private static List<Citizen> citizens = new ArrayList<>();
16
17     public static void main(String[] args) {
18         Scanner sc = new Scanner(System.in);
19         System.out.println("Welcome to SmartCity Portal");
20
21         System.out.print("Enter name: ");
22         String name = sc.nextLine();
23         System.out.print("Enter email: ");
24         String email = sc.nextLine();
25         System.out.print("Enter mobile: ");
26         String mobile = sc.nextLine();
27         System.out.print("Enter password: ");
28         String password = sc.nextLine();
29
30         try {
31             validatePhone(mobile);
32             String encodedPassword = Base64.getEncoder().encodeToString(password.getBytes());
33             Citizen citizen = new Citizen(name, email, mobile, encodedPassword);
34             citizens.add(citizen);
35             System.out.println("Citizen registered: " + citizen);
36
37             System.out.print("Enter complaint description: ");
38             String desc = sc.nextLine();
39             Complaint complaint = new Complaint(desc);
40             complaints.add(complaint);
41             System.out.println("Complaint submitted: " + complaint);
42
43             saveData();
44             loadData();
45             performAnalytics();
46         }
```

```

47     } catch (InvalidPhoneException e) {
48         System.out.println("Error: " + e.getMessage());
49     }
50 }
51
52 private static void validatePhone(String phone) throws InvalidPhoneException {
53     if (!phone.matches("\\d{10}")) {
54         throw new InvalidPhoneException("Phone number must be 10 digits");
55     }
56 }
57
58 private static void saveData() {
59     try (ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream("complaints.ser"))) {
60         out.writeObject(complaints);
61     } catch (IOException e) {
62         e.printStackTrace();
63     }
64 }
65
66 private static void loadData() {
67     try (ObjectInputStream in = new ObjectInputStream(new FileInputStream("complaints.ser"))) {
68         List<Complaint> loaded = (List<Complaint>) in.readObject();
69         System.out.println("Loaded Complaints:");
70         loaded.forEach(System.out::println);
71     } catch (IOException | ClassNotFoundException e) {
72         e.printStackTrace();
73     }
74 }
75
76 private static void performAnalytics() {
77     long openCount = complaints.stream().filter(c -> "Open".equals(c.getStatus())).count();
78     System.out.println("Total open complaints: " + openCount);
79 }
80 }

```

Main.java Citizen.java Complaint.java User.java InvalidPhoneException.java X Admin.java

```

1 package com.smartcity.exception;
2
3 public class InvalidPhoneException extends Exception {
4     public InvalidPhoneException(String message) {
5         super(message);
6     }
7 }

```

#### Entities User.java

Main.java Citizen.java Complaint.java User.java X InvalidPhoneException.java Admin.java

```

1 package com.smartcity.model;
2
3 public abstract class User {
4     protected String name;
5     protected String email;
6     protected String mobile;
7
8     public User(String name, String email, String mobile) {
9         this.name = name;
10        this.email = email;
11        this.mobile = mobile;
12    }
13
14    public String getName() { return name; }
15    public String getEmail() { return email; }
16    public String getMobile() { return mobile; }
17 }

```

#### Admin.java



Main.java
Citizen.java
Complaint.java
User.java
InvalidPhoneException.java
Admin.java ×

```

1 package com.smartcity.model;
2
3 public class Admin extends User {
4
5     public Admin(String name, String email, String mobile) {
6         super(name, email, mobile);
7     }
8
9     public void viewComplaints() {
10         System.out.println("Admin viewing complaints...");
11     }
12
13     @Override
14     public String toString() {
15         return "Admin{name='" + name + "', email='" + email + "', mobile='" + mobile + "'}";
16     }
17 }

```

Main.java
Citizen.java ×
Complaint.java
User.java
InvalidPhoneException.java
Admin.java

```

1 package com.smartcity.model;
2
3 import java.util.Objects;
4
5 public class Citizen extends User {
6     private String password;
7
8     public Citizen(String name, String email, String mobile, String password) {
9         super(name, email, mobile);
10         this.password = password;
11     }
12
13     public String getPassword() { return password; }
14
15     @Override
16     public String toString() {
17         return "Citizen{name='" + name + "', email='" + email + "', mobile='" + mobile + "'}";
18     }
19
20     @Override
21     public boolean equals(Object o) {
22         if (this == o) return true;
23         if (o == null || getClass() != o.getClass()) return false;
24         Citizen citizen = (Citizen) o;
25         return Objects.equals(email, citizen.email);
26     }
27
28     @Override
29     public int hashCode() {
30         return Objects.hash(email);
31     }
32 }

```

Complain.java

Main.java Citizen.java Complaint.java × User.java InvalidPhoneException.java Admin.java

```
1 package com.smartcity.model;
2
3 import java.io.Serializable;
4
5 public class Complaint implements Serializable {
6     private String description;
7     private String status;
8
9     public Complaint() {}
10
11     public Complaint(String description) {
12         this.description = description;
13         this.status = "Open";
14     }
15
16     public String getDescription() { return description; }
17     public String getStatus() { return status; }
18     public void setStatus(String status) { this.status = status; }
19
20     @Override
21     public String toString() {
22         return "Complaint{" + "description='" + description + "', status='" + status + "'";
23     }
24 }
```

Javadoc Console ×

<terminated> Main (13) [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (04-May-2025, 6:49:08 pm – 6:50:00 pm) [pid: 19500]

Welcome to SmartCity Portal  
Enter name: Mrunali Kale  
Enter email: mrunalikale76@gmail.com  
Enter mobile: 8956413765  
Enter password: 11012004  
Citizen registered: Citizen{name='Mrunali Kale', email='mrunalikale76@gmail.com', mobile='8956413765'}  
Enter complaint description: for electricity issue  
Complaint submitted: Complaint{description='for electricity issue', status='Open'}  
Loaded Complaints:  
Complaint{description='for electricity issue', status='Open'}  
Total open complaints: 1