User Notification Preference & Campaign Service

1. Objective

Marketing and product teams want to send notifications (email/SMS/push) only to users who have opted in for specific categories such as offers, order updates, or newsletters.

The system should allow:
- Managing users and their notification preferences
- Creating notification campaigns
- Sending notifications to eligible users

2. User Management

The system should support full lifecycle management of users.

Functionalities:
- Create, view, edit, and delete users
- Bulk upload users via CSV

Each user has the following attributes:
- user_id
- name
- email
- phone
- city
- is_active

3. User Roles and Permissions

The system must support role-based access control.

Roles:
Admin:
- Full access to users, preferences, and campaigns

Creator:
- Create and update users
- Create and update campaigns

Viewer:
- View campaigns
- Download campaign recipient lists

4. Notification Preferences

Each user can opt in or opt out of the following notification categories:
- Promotional Offers
- Order Updates
- Newsletters

Sample preference object:

```
{
  "user_id": "12345",
  "offers": true,
  "order_updates": true,
  "newsletter": false
}
```

5. Campaign Management

Campaigns allow targeting users based on notification preferences and optional filters.

Campaign attributes:
- campaign_id
- campaign_name
- notification_type (offers / order_updates / newsletter)
- city_filter (optional)
- created_by
- status (draft / sent)

Example:
Send promotional offers to users in Bangalore who have opted in.

6. Target User Resolution

The system should automatically determine eligible users for a campaign based on the following rules:
- User must be active
- User must have opted in for the campaign notification type
- User must satisfy optional filters such as city

The system should allow previewing the final recipient list before sending notifications.

7. Sending Notifications (Mocked)

No real SMS, email, or push notification integration is required.

For each notification sent, log the following details in the database:
- user_id
- campaign_id

- sent_at
- status (success / failed)

8. Data Upload

API Upload (≤ 50 users):

```
{
  "users": [
    {
      "user_id": "12345",
      "name": "Amit",
      "email": "amit@test.com",
      "city": "Delhi"
    }
  ]
}
```

CSV Upload (> 50 users):

```
user_id,name,email,phone,city,is_active
12345,Amit,amit@test.com,9999999999,Delhi,true
23456,Riya,riya@test.com,8888888888,Mumbai,true
```

9. Suggested Data Models

User:

```
{
  "user_id": "string",
  "name": "string",
  "email": "string",
  "city": "string",
  "is_active": true
}
```

Preference:

```
{
  "user_id": "string",
  "offers": true,
  "order_updates": false,
  "newsletter": true
}
```

Campaign:

```
{
  "campaign_id": "string",
  "campaign_name": "Diwali Offers",
  "notification_type": "offers",
  "city_filter": "Delhi",
  "status": "draft"
}
```

10. Technology Stack

Backend: Python or Java
Frontend: Choice of framework based on preference
Database: MySQL