1. **Access Keycloak Admin Console**:
    - Once Keycloak is running, access the Keycloak Admin Console through your web browser. The default address is usually http://localhost:8080/auth/admin.
2. **Create a Realm**: **College**
    - A realm is a security domain where your users, applications, and authentication settings will be defined. By default, Keycloak has a master realm, but it's better to create a new one for your application. You can do this from the Keycloak Admin Console.
    - For our application
3. **Create a Client**: **CollegeApplication**
4. **Valid redirect URIs: http://localhost:8082/***
5. **Client Role: user and admin**
    - A client represents your application or service that will be protected by Keycloak. Within the realm you created, go to "Clients" and add a new client. Configure the client settings as per your application's requirements.
6. **Define User Attributes Realm Role**: **CourseManager and Student**
7. **Add Associatd Roles: CourseManager as admin and Student as user**
    - Before adding users, determine the user attributes you need. Keycloak provides some default attributes like username, email, etc. You can also add custom attributes to suit your application's needs.
8. **Create Users**: user1, user2, and user3.
    - Within the realm, navigate to "Users" and click "Add User" to create new users. Provide the required details like username, email, and password. You can also set up two-factor authentication or configure other authentication options depending on your requirements.
9. **User Federation**: user1 as Student. user2 as CourseManager and user3 as Student and CourseManager.
    - If you already have user data in an external system (LDAP, Active Directory, etc.), you can set up User Federation to synchronize users from that system into Keycloak. This way, you can centralize user management.
10. **Integrate Keycloak with Your Application**: **application.properties**
    - Depending on the programming language and framework you are using, you will need to integrate Keycloak's authentication into your application

code. Keycloak provides various client adapters and libraries to ease the integration process.

- **application.properties file**

-
  ```
  #server port number for run Application

  server.port=8082 (role setting valid redirect URLs)
  ```

-
  ```
  # Keycloak Configuration for Application

  keycloak.realm=College (Realm Name)
  keycloak.resource=CollegeApplication (Client Name)
  keycloak.credentials.secret=jK5r9CZTYnu01bqJ4EKwJKAJgyB3pdua (Client
  secret key)
  keycloak.auth-server-url=http://localhost:8080 (keycloak admin URL)
  keycloak.ssl-required=external
  keycloak.public-client=true
  keycloak.principal-attribute=preferred_username
  ```

-
  ```
  #keycloak.use-resource-role-mappings=true for Application Users

  keycloak.security-constraints[0].authRoles[0]=Student
  keycloak.security-constraints[0].authRoles[1]=CourseManager
  keycloak.security-constraints[0].securityCollections[0].name=students
  keycloak.security-
  constraints[0].securityCollections[0].patterns[0]=/students/*
  ```

-
  ```
  #keycloak.use-resource-role-mappings=true for Application Admin
  ```

-
  ```
  keycloak.securityConstraints[1].authRoles[0] = CourseManager
  keycloak.securityConstraints[1].securityCollections[0].name =
  CourseManager
  keycloak.securityConstraints[1].securityCollections[0].patterns[0] =
  /manager/*
  ```

-
  ```
  # Spring Security Configuration for keycloak Realms as like keycloak

  spring.security.oauth2.client.registration.keycloak.client-
  id=CollegeApplication
  spring.security.oauth2.client.registration.keycloak.client-
  secret=jK5r9CZTYnu01bqJ4EKwJKAJgyB3pdua
  spring.security.oauth2.client.registration.keycloak.scope=openid
  spring.security.oauth2.client.provider.keycloak.issuer-
  uri=http://localhost:8080/auth/realms/CollegeApplication
  ```

-
  ```
  # Database configuration for Application

  spring.datasource.url= jdbc:postgresql://localhost:5432/test_data_set
  #spring.datasource.url=
  jdbc:postgresql://192.168.8.186:5432/cyberskills
  #spring.datasource.url= jdbc:postgresql://192.168.56.3:5432/cyberskills
  #spring.datasource.url= jdbc:postgresql://localhost:5432/cyberskills

  spring.datasource.username= postgres
  spring.datasource.password= user1234
  ```

```
spring.jpa.properties.hibernate.jdbc.lob.non_contextual_creation= true
spring.jpa.properties.hibernate.dialect=
org.hibernate.dialect.PostgreSQLDialect
#spring.jpa.open-in-view=false
```

-
  ```
  # Hibernate ddl auto (create, create-drop, validate, update)
  spring.jpa.hibernate.ddl-auto= update
  logging.level.root= WARN
  logging.lavel.ie.cyberskills.application.controller= INFO

  #spring.main.allow-bean-definition-overriding=true
  ```

11. **Test the Authentication**:
    - After integrating Keycloak, test the authentication flow for your application to ensure everything is working as expected.