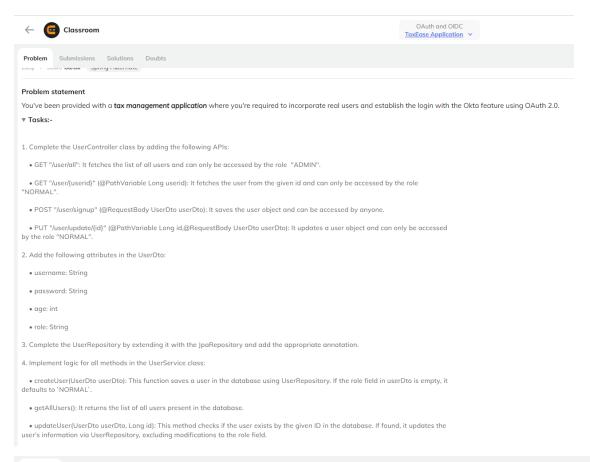
Tax Ease Application

Problem Statement:



Problem Submissions Solutions Doubts

- username: String
- password: String
- age: int
- role: String
- 3. Complete the UserRepository by extending it with the JpaRepository and add the appropriate annotation.
- 4. Implement logic for all methods in the UserService class:
- createUser(UserDto userDto): This function saves a user in the database using UserRepository. If the role field in userDto is empty, it defaults to `NORMAL`.
 - getAllUsers(): It returns the list of all users present in the database.
- updateUser(UserDto userDto, Long id): This method checks if the user exists by the given ID in the database. If found, it updates the user's information via UserRepository, excluding modifications to the role field.
- \bullet getUserByld(Long userid): This method retrieves the user from the UserRepository in the database. If the user is found, it returns the user; otherwise, it throws a `UserNotFoundException` with a custom message.
- ${\tt 5.\ Complete\ the\ UserNotFoundException\ by\ extending\ it\ with\ the\ appropriate\ class.}$
- 6. Create a LoginController and a login.html file as described below:
- Create a `login.html` file within the templates folder. Set up a login form with a "login with Okta" button that redirects to Okta's redirect URL.
 - Create a method login() and use mapping GET "/login": It returns login as a string.
- 7. Lastly, add the required configuration for oauth2 in the application.yml and TaxSecurityConfig file.
- 8. Finally, test your application.
- $\,\blacktriangleright\,$ Special Instructions for submitting the solution:
- ▶ Note:-