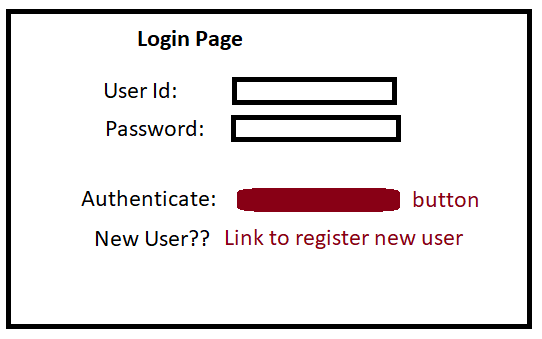
**Assignment:02**

**Front End:**

1. Create Login page. Authenticate from backend and then allow to Dashboard Page
2. Provide registration page for user when user click on link “New User”.



1. Dashboard:
2. Create a Menu bar.
3. Add one menu button in Menu bar named “**Component**”
4. Component button will open after clicking on Component button.
5. There is table on **Components page** where Item can be ADD/REMOVE. Table should be reflecting data dynamically.
6. Bottom of the table there will be on “Submit” button which’ll send all data in table and reflect updated data from Database.
7. ..

**Back-end:**

**Login Page:**

1. Use Spring Security for maintaining authentication.
2. Create **REST API** to add new user.

**Components Page**:

1. **Create REST API** which receive String as below from UI and save/update in DB:

"501-R:EXCLUDE,502-R:INCLUDE,503-R:INCLUDE,7685-TS:EXCLUDE,13433-TC:EXCLUDE,277427-S:EXCLUDE,502-R:EXCLUDE, 503-R:EXCLUDE, 509-:EXCLUDE "

1. Using JPA save data in table :

Table name: USERFLAG

Columns:

ID : LONG

TYPE : VARCHAR

VALUE : VARCHAR

1. Split String in below format:

ID: 501

Type: R

Value: EXCLUDE

ID:502

Type: R

Value: INCLUDE

ID: 13433

Type: TC

Value: EXCLUDE

1. Type “R” should be saved in DB as “ROUTINE”, Type “TC” should be saved in DB as “TESTCASE”, Type “TS” should be saved in DB as “TESTSTEP” and Type “S” should be saved in DB as “SEQUENCE”.
2. Last ID and its value should be saved in table If ID is repeating in STRING from UI.
3. Throw exception if any set has id/type/value is NULL coming from UI.
4. **Create REST API** by which we can fetch data from table on basis of id and Type using Swagger.
   1. While fetching data user will provide id and type as R/S/TS/TC.
5. Add dependencies for SWAGGER.
6. Explain how you’ll host this application in Azure cloud. (Scaling up services, CI/CD pipeline, storing image of API).

**Follow below steps building Spring Boot REST API:**

01: Initializing a Spring Boot Project

02: Connecting Spring Boot to the Database

03: Creating a User Model

04: Creating Repository Classes

05: Creating a Controller/Resource

06: Testing the REST APIs using Swagger.