User Registration Formik & Yup

IIHT

Time To Complete: 2 hr

CONTENTS

1 Problem Statement

User Registration is SPA (Single Page Application) for showing a user registration form with multiple details created with formik and yup.

The core modules of User Registration app are:

1. User registration form

2 Proposed User Registration Wireframe

UI needs improvisation and modification as per given use case and to make test cases passed.

2.1 User Registration form

User Registration

First Name*
First name is required

Last Name

Email*
Email is required

Contact

Department*
Department is required

Designation*
Designation is required

Experience*
Experience is required

Add User

3 BUSINESS-REQUIREMENT:

As an application developer, develop the Health Club App (Single Page App) with below guidelines:

User Story #	User Story Name	User Story
US_01	User Registration	As a user I should be able to create a user Acceptance criteria: 1. As a user I should be able to see a user registration form with fields as: 1.1 First Name 1.2 Last Name 1.3 Email 1.4 Contact 1.5 Department 1.6 Designation 1.7 Experience 1.8 Add User button. 2. If any field does not match with its validations, then a proper error message should be shown. 3. Add User button should be enabled only when all validations are fulfilled. 4. On click of “Add User” button, we must see an alert message as “User added” and on closing that alert, all fields must be cleared.

4 VALIDATIONS

- All required fields must be fulfilled with valid data.
- First Name field is required and on invalid first name field, “First name is required” message should be shown.
- Email field is required and on invalid email field, “Invalid Email” message should be shown.
- Contact field must accept only numeric values with max length of 10.
- Department field is required and on invalid department field, “Department is required” message should be shown.
- Designation field is required and on invalid designation field, “Designation is required” message should be shown.
- Experience field is required and on invalid experience field, “Experience is required” message should be shown.
- In starting “Add User” button should be disabled.
- Only after validating fields, “Add User” button should be enabled.

5 GUIDELINES

- All fields must have “name” and “id” attribute with value as same of that field.
    - For example: for First Name input field
      <input type=”text” name=”firstName” id=”firstName”/>
- After executing test command, you can check out “output_boundary_revised.txt” file to see, which test cases are passed and which are not.

6 CONSTRAINTS

- On the page load, a form must be there.
- You should be able to press the “TAB” key and “SHIFT + TAB” to navigate from top field to bottom field and vice-versa.
- If we click on required field and moved to next field without entering any valid data, an appropriate error message should be shown.
- “Add User” button will be disabled until all validations are fulfilled.
- On click of “Add User” button, we must see an alert message as “User added” and on closing that alert, all fields must be cleared.

7 MANDATORY ASSESSMENT GUIDELINES

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to**
   Application menu (Three horizontal lines at left top) -> Terminal ->New Terminal.
3. **This editor Auto Saves the code.**
4. **If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.**
5. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**

6. **To test any Restful application, the last option on the left panel of IDE, you can find ThunderClient, which is the lightweight equivalent of POSTMAN.**
7. **This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.**
   Note: The application will not run in the local browser

8. **You can follow series of command to setup React environment once you are in your project-name folder:**
   a. **npm install -> Will install all dependencies -> takes 10 to 15 min**
   b. **npm run start -> To compile and deploy the project in browser. You can press <Ctrl> key while clicking on localhost:8080/8081 to open project in browser -> takes 2 to 3 min**
   c. **npm run test -> to run all test cases. It is mandatory to run this command before submission of workspace -> takes 5 to 6 min**
9. **You may also run "npm run test" while developing the solution to re-factor the code to pass the test-cases.**
10. **Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on "Submit Assessment" after you are done with code.**
11. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**