React Form Handling Lab:

In this lab we will create a user registration form which will accept two user inputs from the User. The form will be validated if there is no input received or empty input from the user.

- Create a new React App
- Install necessary dependencies for the Lab.
- Create a SignUp Form Component
- Import the Component to the root of your App.
- Write Logic to handle the Form Submission and Form Validation

As the developer in the team, you are supposed to work on the following tasks:

Solution

Task 1: Create new React App

1. In this task, create a new react app with the following commands:

<table>
<tr>
<td>1
2</td>
<td>

//

$ npx create-react-app <application-name>

OR

// Optional Command

$ npm create vite@latest <application-name>

// select react as template

// It is recommended to use the first command only

</td>
</tr>
</table>

```
● (base) dev$ npm create vite@latest react-form-handling-lab
  Need to install the following packages:
  create-vite@5.2.0
  Ok to proceed? (y) y
  ✔ Select a framework: › React
  ✔ Select a variant: › JavaScript + SWC

  Scaffolding project in /Users/mithileshtarkar/IT/Code/React Projects/react-form-handling-lab...

  Done. Now run:

    cd react-form-handling-lab
    npm install
    npm run dev

○ (base) dev$
○ (base) dev$ ▮
```

Task 2: Install Dependencies

2. In this task, let us install the dependencies we need for this lab.

1 2	$ npm i react-hook-form yup @hookform/resolvers --save

```
(base) dev$ npm i react-hook-form yup @hookform/resolvers --save

added 7 packages, and audited 228 packages in 4s

97 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
(base) dev$
```

Task 3: Create SignUp Form Component

3. In this task, we will create the signUp form component.

```html
<div class="container">
    <h2>Sign Up</h2>
    <form>
        <div class="form-group">
            <label>
                First Name
            </label>
            <input
                type="text"
                name="first_name"

            />
            <p></p>
        </div>
        <div class="form-group">
            <label>
                Last Name
            </label>
            <input
                type="text"
                name="last_name"
            />
            <p></p>
        </div>
        <input type="submit" value="Sign Up" />
    </form>
</div>
```

Task 4: Add CSS to the SignUp Form

4. In this task, we will add css in the assets folder and link it with the SignUp Component.

1 2	[click here to download the css](#)

Task 5: Add the Form Handling and Validation Logic.

5. In this task, we will add the form handling and validation logic to the SignUp Component.

```jsx
import React from 'react'
import '../assets/css/sign_up_form.css'
import { useForm } from 'react-hook-form'
import { yupResolver } from "@hookform/resolvers/yup"
import * as yup from "yup"
const SignUpForm = () => {

    const formSchema = yup.object({
        first_name: yup.string().required('First Name is Required'),
        last_name: yup.string().required('Last Name is Required'),
    })

    const {
        register,
        handleSubmit,
        formState: { errors },
    } = useForm({
        resolver: yupResolver(formSchema)
    })

    const onSubmit = (data) => {
        console.log(data);
    }
```

```
<form onSubmit={handleSubmit(onSubmit)}>
    <div class="form-group">
        <label>
            First Name
        </label>
        <input
            type="text"
            name="first_name"
            {...register('first_name')}
        />
        <p>{errors.first_name?.message}</p>
    </div>
    <div class="form-group">
        <label>
            Last Name
        </label>
        <input
            type="text"
            name="last_name"
            {...register('last_name')}
        />
        <p>{errors.last_name?.message}</p>
    </div>
    <input type="submit" value="Sign Up" />
</form>
```

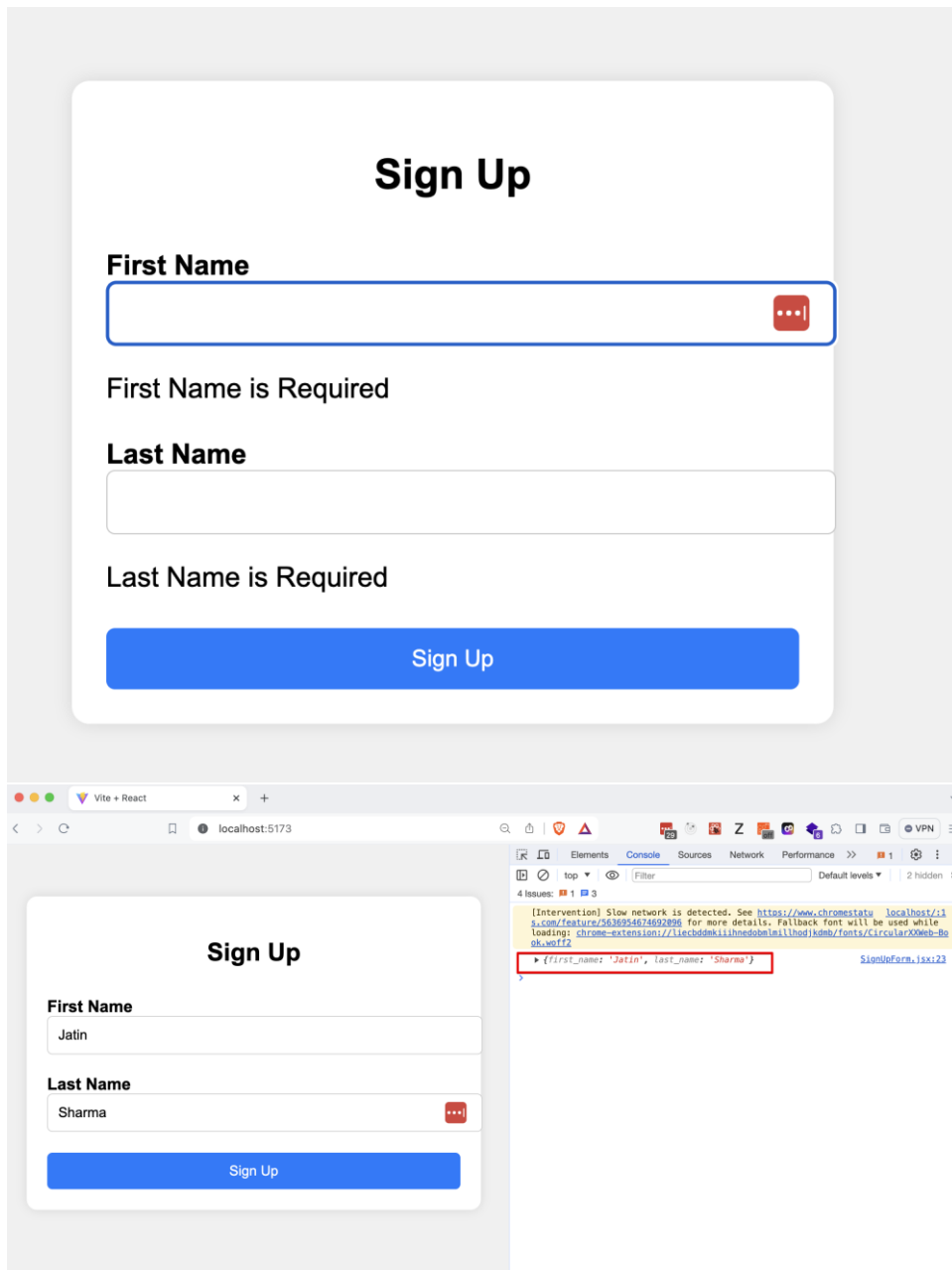# Task 9: Start the React Server

9. In this task, start the react server with the following command.

| 1 2 | $ npm run start |

Task 10: Your Output

10. Your output should look like the screenshot as mentioned below.

Task 11: Conclusion

11. In this lab, we have completed the following tasks:

- We have created a React App
- We have successfully implemented Form Validation and Form handling by installing dependencies like yup, react hook form.
- We have successfully submitted the Form data after validation.
- Discuss the Solution with your Teammates, Trainers and Mentors.
- Is there anything where you can improve?
- Get Feedback on the Output.