Worksheet-5

Student Name: Harsh **UID: 24MCA20045**
Branch: MCA (General) **Section/Group: 1/A**
Semester: 2nd **Date of Performance: 11-04-25**
Subject Name: Front End Technologies **Subject Code: 24CAH-655**

1. Aim/Overview of the practical:

Create login and signup forms using different React controls.

2. Objective:

• Implement login and signup formsin React using various UI controls.

•Use controlled componentsto handle form input dynamically.

Implement validation using React state and validation libraries (e.g., Yup,

React Hook Form).

•Manage authentication states using local storage or context API.

3. Overview:

You will create a **Login Form** and a **Signup Form** in React using different UI controls,

such as:

a) Text fieldsfor username/email and password.

b) Checkboxesfor "Remember Me" options.

c) Radio buttons or dropdowns for user roles(optional).

d) Toggle buttonsfor password visibility.

e) Submit buttons with loading indicators.

You will also implement form validation, manage state, and handle authentication logic.

Coding:

(app.js)

```
import React, { useState } from 'react'; import Login from './component/login'; import Signup from './component/signup';

function App() {

const [showLogin, setShowLogin] = useState(true);

return (

<div className="App">

<h1>{showLogin ? 'Login' : 'Signup'} Form</h1>

<button onClick={() => setShowLogin(!showLogin)}> Switch to {showLogin ? 'Signup' : 'Login'}

</button>

{showLogin ? <Login />: <Signup />}

</div>

);

}

export default App;
```

(login.js)

```
import React from 'react';

import { useForm } from 'react-hook-form';

import { yupResolver } from '@hookform/resolvers/yup';

import * as yup from 'yup';

import './login.css';

// Validation schema

const loginSchema = yup.object().shape({

  email: yup.string().email('Invalid email').required('Email is required'),
  password: yup.string().min(4).required('Password is required'),
});
const Login = () => {
  const {
    register,
    handleSubmit,
    formState: { errors },
  } = useForm({ resolver: yupResolver(loginSchema) });

  const onSubmit = (data) => {
    const storedUser = JSON.parse(localStorage.getItem('user'));

    if (storedUser && data.email === storedUser.email && data.password === storedUser.password) {
      alert('Login successful!');
    } else {
      alert('Invalid credentials!');
    }
  };
  const handleForgotPassword = () => {
    const email = prompt('Enter your registered email:');
    if (!email) return;
    // Simulate checking if user exists
    const storedUser = JSON.parse(localStorage.getItem('user'));
    if (storedUser && storedUser.email === email) {
      alert(`A password reset link has been sent to ${email}.`);
    } else {
      alert('No account found with this email.');
    }
  };
  return (
    <form onSubmit={handleSubmit(onSubmit)}>
      <input placeholder="Email" {...register('email')} />
      <p>{errors.email?.message}</p>

      <input type="password" placeholder="Password" {...register('password')} />
      <p>{errors.password?.message}</p>

      <div className="forgot-password">
        <a href="#" onClick={handleForgotPassword}>Forgot password?</a>
      </div>

      <button type="submit">Login</button>
    </form>
  );
};
export default Login;
```

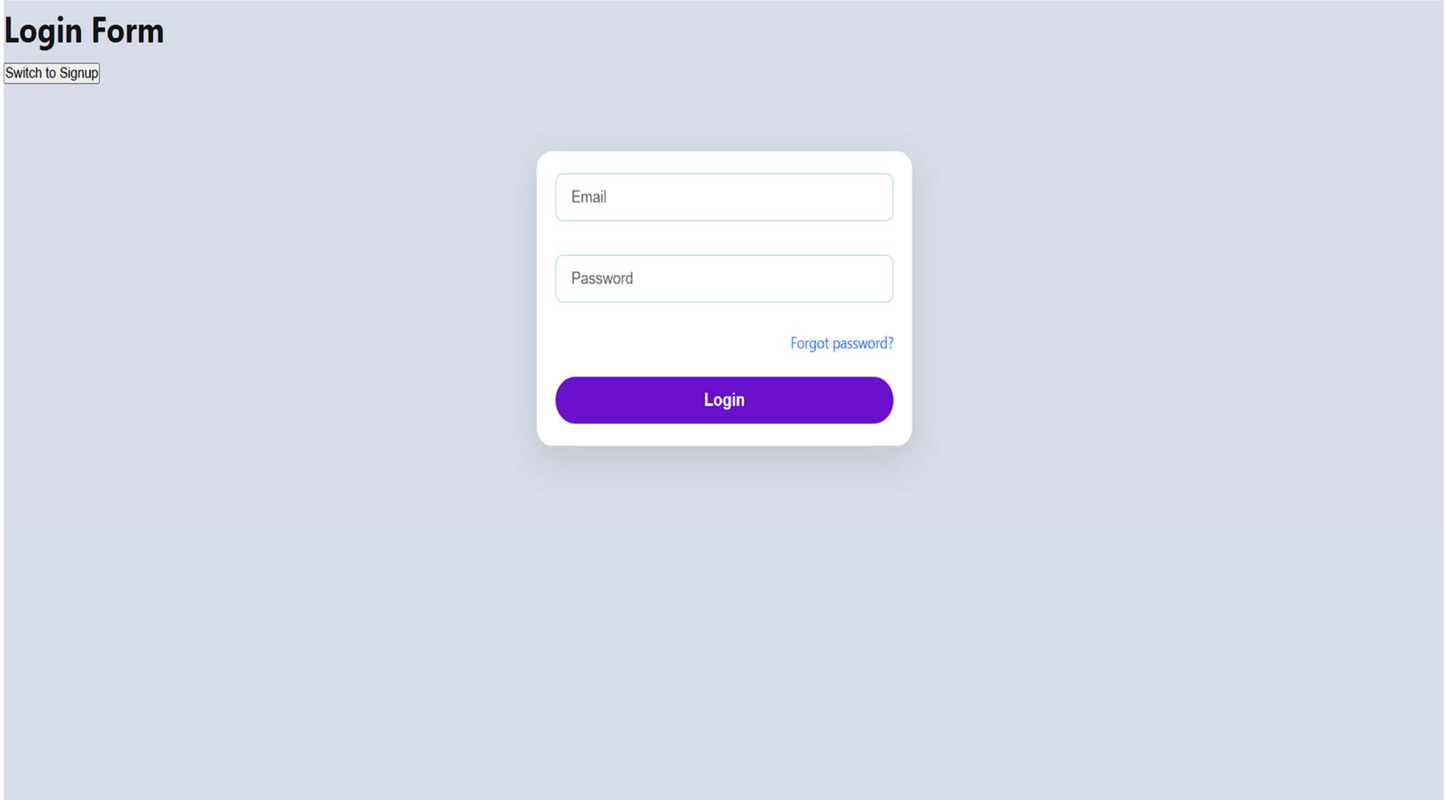
Signup.js

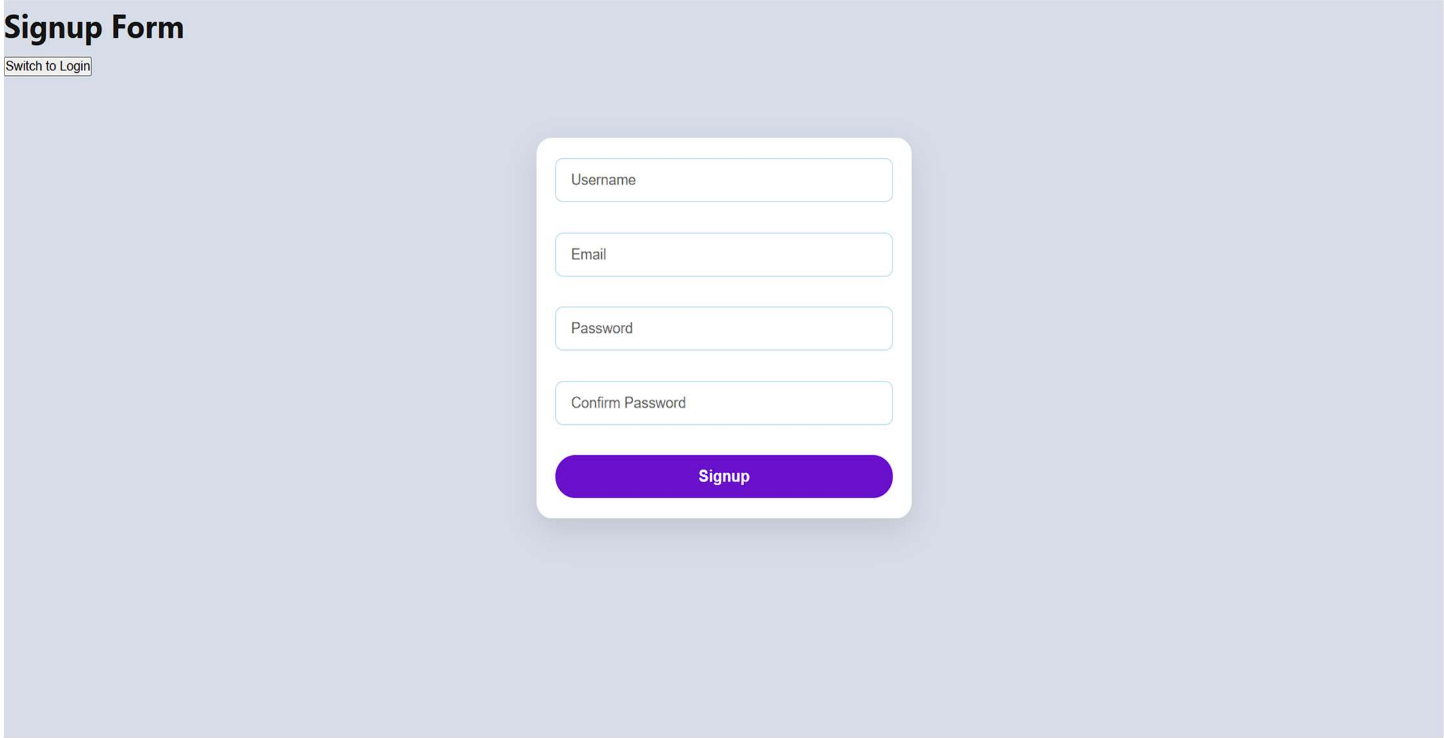
```
import React from 'react';
import { useForm } from 'react-hook-form';
import { yupResolver } from '@hookform/resolvers/yup'; import * as yup from 'yup';
import './signup.css';
const signupSchema = yup.object().shape({
username: yup.string().required('Username is required'),
email: yup.string().email('Invalid email').required('Email is required'), password:
```

```
yup.string().min(4).required('Password is required'), confirmPassword: yup
.string()
.oneOf([yup.ref('password')], 'Passwords do not match')
.required('Confirm password is required'),
});
const Signup = () => { const {
register, handleSubmit, formState: { errors },
} = useForm({ resolver: yupResolver(signupSchema) });
const onSubmit = (data) => { localStorage.setItem(
'user',
JSON.stringify({ email: data.email, password: data.password })
);
alert('User registered successfully!');
};
return (
<form onSubmit={handleSubmit(onSubmit)}>
<input placeholder="Username" {...register('username')} />
<p>{errors.username?.message}</p>
<input placeholder="Email" {...register('email')} />
<p>{errors.email?.message}</p>
<input type="password" placeholder="Password" {...register('password')} />
<p>{errors.password?.message}</p>
<input type="password"
placeholder="Confirm Password"
{...register('confirmPassword')}
/>
<p>{errors.confirmPassword?.message}</p>
```

```
<button type="submit">Signup</button>

</form>

);

};

export default Signup;
```

Output:





4. Learning outcomes (What I have learnt):

- Understanding how to create and manage forms in React.
- Implementing controlled and uncontrolled form components.
- Validating form input using React Hook Form and Yup.
- Managing authentication state using localstorage or context.
- Styling UI components with CSS frameworks.

5. Evaluation Grid :

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Worksheet Completion		08 marks
2.	Conduct		12 marks
3.	Viva Voice		10 marks
	Total		30 marks