

Learner Assignment Submission Format

Learner Details

- **Name:**H O Akash
 - **Enrollment Number:**SU624MR015
 - **Batch / Class:**MERN Stack
 - **Assignment:**Typescript
 - **Date of Submission:**12/08/2025
-

Problem Solving Activity 1.1

1. Program Statement

Create a registration form that collects user information (first name, last name and email) and validates the input data. If the data is valid, submit the form and display a success message.

2. Algorithm

1. Initialize State:

- Initialize state variables to store form data and errors.

2. Handle Input Changes:

- Update the form data state when the user types in the input fields.

3. Validate Form Data:

- Check if the form data is valid (e.g., first name, last name, email, and password are required, and email format is valid).
- If the form data is invalid, display error messages.

4. Submit the Form:

- If the form data is valid, submit the form and display a success message.
-

3. Pseudocode

FUNCTION RegistrationForm

 INITIALIZE formData state with default values

 INITIALIZE errors state with default values

 FUNCTION handleChange

 UPDATE formData state with new input values

 FUNCTION handleSubmit

 VALIDATE formData

 IF formData is valid

 SUBMIT form data

 DISPLAY success message

 ELSE

 DISPLAY error messages

 RENDER registration form with input fields and error messages

4. Program Code

```
import React, { useState } from 'react';
import './register.css'

interface RegistrationFormData {
  firstName: string;
  lastName: string;
  email: string;
  password: string;
```

```
}

const RegistrationForm = () => {

  const [formData, setFormData] = useState<RegistrationFormData>({

    firstName: '',

    lastName: '',

    email: '',

    password: '',

  });

  const [errors, setErrors] = useState({

    firstName: '',

    lastName: '',

    email: '',

    password: '',

  });

  const handleChange = (e: React.ChangeEvent<HTMLInputElement>) => {

    const { name, value } = e.target;

    setFormData({

      ...formData,

      [name]: value,

    });

  };

};
```

```
const handleClick=()=>{  
  
  alert("submitted succesfull")  
  
}  
  
const handleSubmit = (e: React.FormEvent<HTMLFormElement>) => {  
  
  e.preventDefault();  
  
  let isValid = true;  
  
  const newErrors = {  
  
    firstName: '',  
  
    lastName: '',  
  
    email: '',  
  
    password: '',  
  
  };  
  
  if (!formData.firstName) {  
  
    newErrors.firstName = 'First name is required';  
  
    isValid = false;  
  
  }  
  
  if (!formData.lastName) {  
  
    newErrors.lastName = 'Last name is required';  
  
    isValid = false;  
  
  }  
  
}
```

```

if (!formData.email) {

    newErrors.email = 'Email is required';

    isValid = false;

} else if
(!/^([A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,})$/i.test(formData.email)) {

    newErrors.email = 'Invalid email address';

    isValid = false;

}

setErrors(newErrors);

if (isValid) {

    // Submit the form data

    console.log(formData);

}

};

return (

    <div className='a'>

    <form onSubmit={handleSubmit}>

        <div>

            <label>First Name:</label>

            <input type="text" name="firstName" value={formData.firstName}
onChange={handleChange} />

```

```
    {errors.firstName && <div style={{ color: 'red'
}}>{errors.firstName}</div>}

</div><br />

<div>

  <label>Last Name:</label>

  <input

    type="text"

    name="lastName"

    value={formData.lastName}

    onChange={handleChange}

  />

  {errors.lastName && <div style={{ color: 'red'
}}>{errors.lastName}</div>}

</div><br />

<div>

  <label>Email:</label>

  <input

    type="email"

    name="email"

    value={formData.email}

    onChange={handleChange}

  />

  {errors.email && <div style={{ color: 'red'
}}>{errors.email}</div>}

</div><br />

<button type="submit" onClick={handleClick}>Register</button>
```

```
</form>

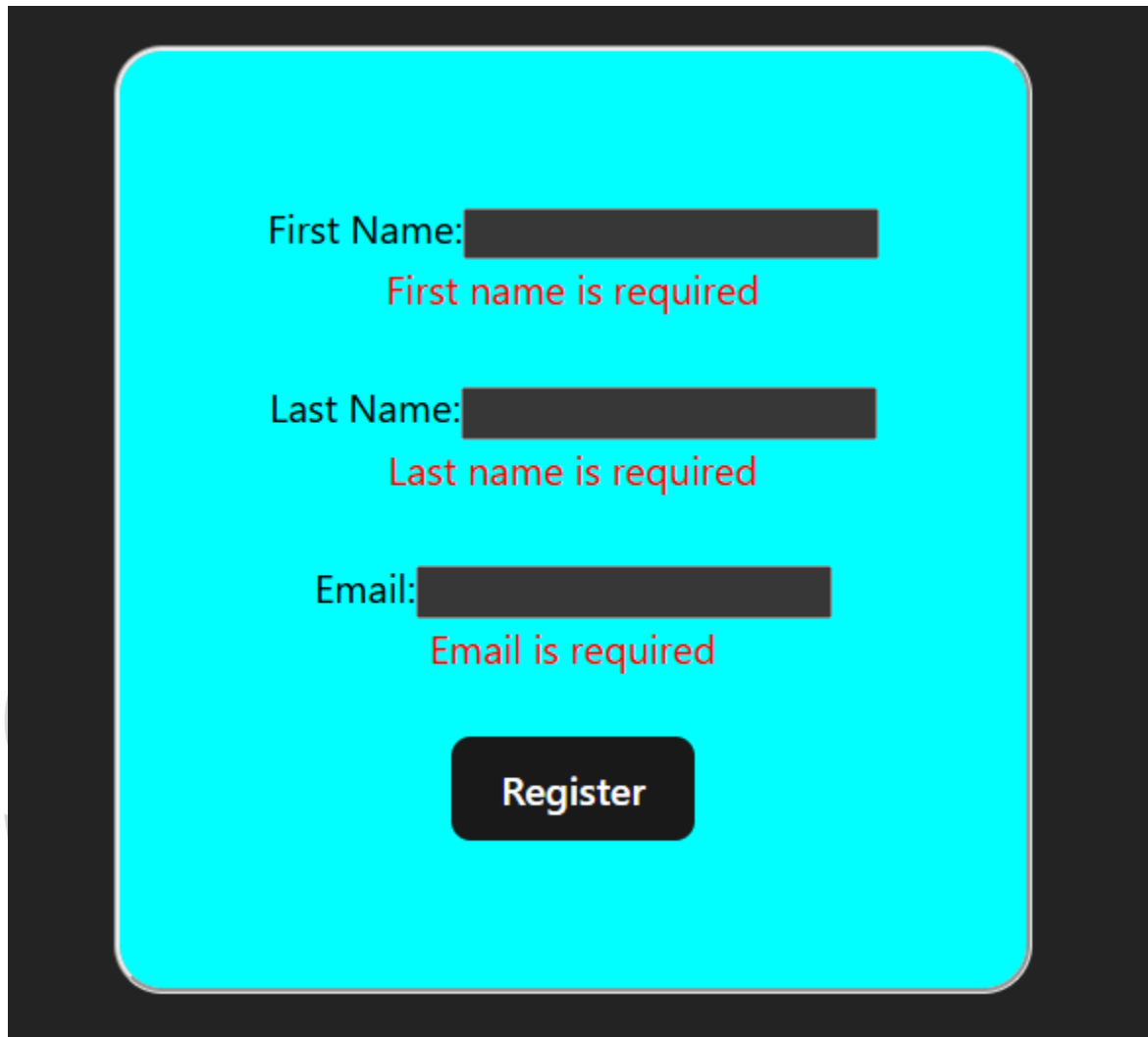
</div>

);

};

export default RegistrationForm;
```

6. Screenshots of Output



The screenshot shows a registration form with a light blue background and rounded corners. It contains three input fields: 'First Name:', 'Last Name:', and 'Email:'. Each field is followed by a red error message: 'First name is required', 'Last name is required', and 'Email is required' respectively. Below the input fields is a dark blue 'Register' button. The form is set against a dark background.

7. Observation / Reflection

The given React code is a simple implementation of a registration form with validation. Here are some reflections:

- The code uses the useState hook to manage the form data and error states.
- The handleChange function updates the form data state when the user types in the input fields.
- The handleSubmit function validates the form data and submits it if it's valid.
- The code uses a simple validation logic to check if the form data is valid.

