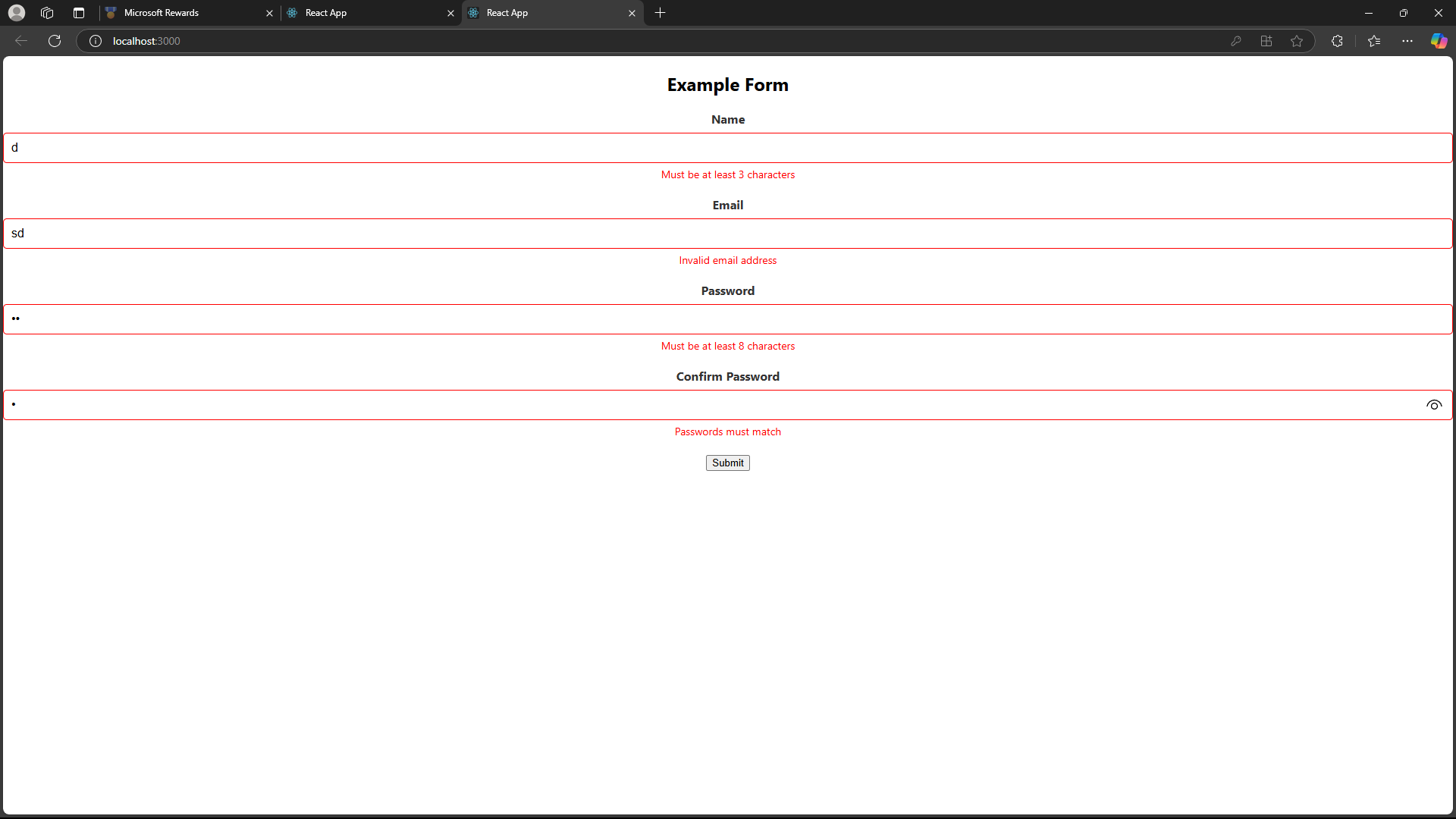
1. **Task Description**

The purpose of this project was to build a **Reusable Input Component** for forms using **Formik** and **Yup**. The input component supports various validation types and can be reused across multiple forms. This simplifies form development and enhances maintainability. The key features of this project include:

* A reusable input component (CustomInput) with built-in error handling.
* Integration of **Formik** for form state management.
* Use of **Yup** for validation schema to enforce form input rules.
* Example form (ExampleForm) demonstrating the use of the reusable input component.
* Responsive design and styled components for a user-friendly interface.

1. **Task Output Screenshot:**



1. **Widget/Algorithm Used In Task**

#### **CustomInput Component**

* A reusable input field that handles user input and validation errors dynamically.
* Uses Formik's useField hook to connect the input field to Formik's form state.
* Displays an error message below the input field if validation fails.

 Includes fields for:

* Name
* Email
* Password
* Confirm Password
*  Handles form submission and displays input values upon validation.

**Algorithms**

#### **Validation Schema (Yup)**

* Validates form input using the following rules:
  + **Name**: Must be between 3 and 15 characters and is required.
  + **Email**: Must follow a valid email format and is required.
  + **Password**: Must include at least 8 characters, one uppercase letter, one lowercase letter, and one number.
  + **Confirm Password**: Must match the password field.

#### **Formik for Form Management**

* Manages form state, validation, and error handling efficiently.
* Automatically updates form state as the user interacts with the fields.