FORMIK VALIDATION (DAY 31)

BASH

npm install react formik yup

JSX

// App.js

import React from 'react';

import { BrowserRouter as Router, Switch, Route } from 'react-router-dom';

import BookManagement from './components/BookManagement';

import AuthorManagement from './components/AuthorManagement';

function App() {

return (

<Router>

<div>

<Switch>

<Route path="/books">

<BookManagement />

</Route>

<Route path="/authors">

<AuthorManagement />

</Route>

</Switch>

</div>

</Router>

);

}

export default App;

JSX

// components/BookManagement.js

import React from 'react';

import { Formik, Form, Field, ErrorMessage } from 'formik';

import \* as Yup from 'yup';

const BookSchema = Yup.object().shape({

title: Yup.string().required('Title is required'),

author: Yup.string().required('Author is required'),

year: Yup.number().required('Year is required').positive().integer(),

});

const BookManagement = () => {

const handleSubmit = (values, actions) => {

console.log(values);

// You can add logic here to handle form submission like API calls, database updates, etc.

actions.setSubmitting(false);

};

return (

<div>

<h2>Manage Books</h2>

<Formik

initialValues={{ title: '', author: '', year: '' }}

validationSchema={BookSchema}

onSubmit={handleSubmit}

>

{({ isSubmitting }) => (

<Form>

<div>

<label htmlFor="title">Title</label>

<Field type="text" name="title" />

<ErrorMessage name="title" component="div" />

</div>

<div>

<label htmlFor="author">Author</label>

<Field type="text" name="author" />

<ErrorMessage name="author" component="div" />

</div>

<div>

<label htmlFor="year">Year</label>

<Field type="number" name="year" />

<ErrorMessage name="year" component="div" />

</div>

<div>

<button type="submit" disabled={isSubmitting}>

Submit

</button>

</div>

</Form>

)}

</Formik>

</div>

);

};

export default BookManagement;

JSX

// components/AuthorManagement.js

import React from 'react';

import { Formik, Form, Field, ErrorMessage } from 'formik';

import \* as Yup from 'yup';

const AuthorSchema = Yup.object().shape({

name: Yup.string().required('Name is required'),

nationality: Yup.string().required('Nationality is required'),

});

const AuthorManagement = () => {

const handleSubmit = (values, actions) => {

console.log(values);

// You can add logic here to handle form submission like API calls, database updates, etc.

actions.setSubmitting(false);

};

return (

<div>

<h2>Manage Authors</h2>

<Formik

initialValues={{ name: '', nationality: '' }}

validationSchema={AuthorSchema}

onSubmit={handleSubmit}

>

{({ isSubmitting }) => (

<Form>

<div>

<label htmlFor="name">Name</label>

<Field type="text" name="name" />

<ErrorMessage name="name" component="div" />

</div>

<div>

<label htmlFor="nationality">Nationality</label>

<Field type="text" name="nationality" />

<ErrorMessage name="nationality" component="div" />

</div>

<div>

<button type="submit" disabled={isSubmitting}>

Submit

</button>

</div>

</Form>

)}

</Formik>

</div>

);

};

export default AuthorManagement;