**Task 1**

**Title:** Formik

**Description:** Create custom common component for the input using formik with all kind of validation

**Project Overview**

This project implements a user login form using React and Formik, with comprehensive validation using Yup. The form captures user inputs for a username, email, and password, ensuring that all fields meet specific criteria before submission. The implementation also includes a user-friendly popup notification that confirms successful form submission.

**Key Features**

1. **User Input Fields**: The form consists of three input fields:
   * **Username**: Must be at least 5 characters long and is required.
   * **Email**: Must be a valid email format and is required.
   * **Password**: Must meet the following criteria:
     + At least 8 characters long.
     + Contains at least one uppercase letter (A-Z).
     + Contains at least one lowercase letter (a-z).
     + Contains at least one digit (0-9).
     + Contains at least two special characters from the set @#$%&.
2. **Validation**: The form uses Yup for validation, ensuring that all input fields adhere to the specified rules. If the validation fails, appropriate error messages are displayed to guide the user.
3. **Popup Notification**: Upon successful submission, a popup notification appears, confirming that the form has been submitted successfully. The notification automatically disappears after 3 seconds.
4. **Responsive Design**: The form is designed to be responsive, adapting to different screen sizes for an optimal user experience.

**Technical Details**

* **Framework**: React
* **State Management**: React's **useState** hook is used to manage the visibility of the popup notification.
* **Form Handling**: Formik is utilized to handle form state and submission, simplifying the process of managing form inputs and validation.
* **Validation Library**: Yup is used for schema-based validation, providing a robust way to enforce rules on user inputs.

**Code Structure**

The main components of the code include:

* **Login Component**: This is the primary component that renders the form, handles submission, and manages the popup notification.
* **Custom Input Component**: This reusable component is used for rendering input fields with labels and error handling.

**Example Code Snippet**

const validationSchema = Yup.object({

username: Yup.string().required('Username is required').min(5, 'Too short!'),

email: Yup.string().email('Invalid email').required('Email is required'),

password: Yup.string()

.required('Password is required')

.min(8, 'Password must be at least 8 characters long')

.matches(/[A-Z]/, 'It must contain at least one A-Z letter')

.matches(/[a-z]/, 'It must contain at least one a-z letter')

.matches(/\d/, 'It must contain at least one digit 0-9')

.matches(/[^a-zA-Z0-9]/g, 'Password must contain at least two @#$%& ')

.test('two-symbols', 'Password must contain at least two symbols', (value) => {

return (value.match(/[^a-zA-Z0-9]/g) || []).length >= 2; }),

});

**Conclusion**

This project successfully implements a user login form with robust validation with React.js as framework, Formik as a form handling and Yup as validation library.

**Future Enhancements**

* Integration with a backend authentication service.
* Additional features such as "Remember Me" functionality.
* Enhanced user experience with animations for the popup notification.
* Accessibility improvements to ensure compliance with standards.

**Submitted By**:  
*Arnava Tivari*