

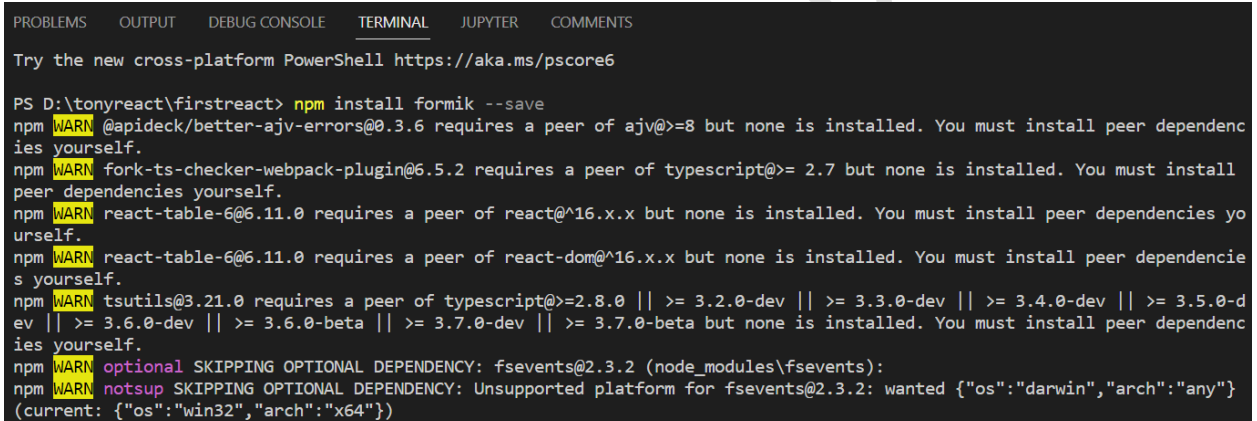
# Hướng dẫn validate form trong React với Formik

## Giáo viên: Nguyễn Hùng Cường

Formik là một thư viện mở rộng của React, nó cung cấp các tính năng cho phép thao tác với form, đồng thời cung cấp khả năng validate form. Formik giúp tiết kiệm thời gian làm việc với form.

### Bước 1: Cài đặt thư viện

Để sử dụng Formik, ta cần cài đặt thư viện. Ta mở VS Code, mở Project, rồi mở Terminal, sau đó gõ lệnh `npm install formik --save` để cài đặt thư viện Formik.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER COMMENTS
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\tonyreact\firstreact> npm install formik --save
npm WARN @apideck/better-ajv-errors@0.3.6 requires a peer of ajv@>=8 but none is installed. You must install peer dependencies yourself.
npm WARN fork-ts-checker-webpack-plugin@6.5.2 requires a peer of typescript@>= 2.7 but none is installed. You must install peer dependencies yourself.
npm WARN react-table-6@6.11.0 requires a peer of react@^16.x.x but none is installed. You must install peer dependencies yourself.
npm WARN react-table-6@6.11.0 requires a peer of react-dom@^16.x.x but none is installed. You must install peer dependencies yourself.
npm WARN tsutils@3.21.0 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
```

### Bước 2: Định nghĩa css

Tiếp theo, ta tạo file css, rồi trong file này, ta định nghĩa các class để có thể hiển thị giao diện của form nhập trong bước sau.

# FormDangKy.css U X

src > # FormDangKy.css > #expenseForm span

```
1  input[type=text], input[type=number], input[type=date], select {
2      width: 100%;
3      padding: 12px 20px;
4      margin: 8px 0;
5      display: inline-block;
6      border: 1px solid #ccc;
7      border-radius: 4px;
8      box-sizing: border-box;
9  }
10 input[type=submit] {
11     width: 100%;
12     background-color: #4CAF50;
13     color: white;
14     padding: 14px 20px;
15     margin: 8px 0;
16     border: none;
17     border-radius: 4px;
18     cursor: pointer;
19 }
```

```
21 input[type=submit]:hover {
22     background-color: #45a049;
23 }
24 input:focus {
25     border: 1px solid #d9d5e0;
26 }
27 #expenseForm div {
28     border-radius: 5px;
29     background-color: #f2f2f2;
30     padding: 20px;
31 }
32 #expenseForm span {
33     color: red;
34 }
```

### Bước 3: Định nghĩa form đăng ký

Trong bước tiếp theo, ta sẽ định nghĩa một component có chứa một form đăng ký người dùng. Tạo form này ta sẽ import thư viện Formik để phục vụ cho thao tác validate form với mã nguồn như sau.

```
JS FormDangKy.js U X
src > components > JS FormDangKy.js > FormDangKy > render
1  import React from 'react';
2  import '../FormDangKy.css'
3  import { Formik } from 'formik';
4
5  class FormDangKy extends React.Component {
6    constructor(props) {
7      super(props);
8
9      this.initialValues = { name: '', amount: '', date: '', category: '' }
10   }
11   validate = (values) => {
12     const errors = {};
13     if (!values.name) {
14       errors.name = 'Bạn cần nhập tên !';
15     }
16     if (!values.amount) {
17       errors.amount = 'Bạn cần nhập số lượng !';
18     }
19     if (!values.date) {
20       errors.date = 'Bạn cần nhập ngày !';
21     }
22     if (!values.category) {
23       errors.category = 'Bạn cần nhập loại !';
24     }
25     return errors;
26   }
```

```

JS FormDangKy.js U ●
src > components > JS FormDangKy.js > FormDangKy > validate
28   handleSubmit = (values, setSubmitting) => {
29     setTimeout(() => {
30       alert(JSON.stringify(values, null, 2));
31       setSubmitting(false);
32     }, 400);
33   }
34
35   render() {
36     return (
37       <div id="expenseForm">
38         <Formik
39           initialValues={this.initialValues}
40           validate={values => this.validate(values)}
41           onSubmit={values, { setSubmitting }} => this.handleSubmit(values, setSubmitting)} >
42           {
43             ({
44               values,
45               errors,
46               touched,
47               handleChange,
48               handleBlur,
49               handleSubmit,
50               isSubmitting,
51               /* and other goodies */

```

```

JS FormDangKy.js U ●
src > components > JS FormDangKy.js > FormDangKy > validate
52   }) =>
53   (
54     <form onSubmit={handleSubmit}>
55       <label for="name">Title <span>{errors.name && touched.name && errors.name}</span></label>
56       <input type="text" id="name" name="name" placeholder="Enter expense title"
57         onChange={handleChange}
58         onBlur={handleBlur}
59         value={values.name} />
60
61       <label for="amount">Amount <span>{errors.amount && touched.amount && errors.amount}</span></label>
62       <input type="number" id="amount" name="amount" placeholder="Enter expense amount"
63         onChange={handleChange}
64         onBlur={handleBlur}
65         value={values.amount} />
66
67       <label for="date">Spend Date <span>{errors.date && touched.date && errors.date}</span></label>
68       <input type="date" id="date" name="date" placeholder="Enter date"
69         onChange={handleChange}
70         onBlur={handleBlur}
71         value={values.date} />

```

```

JS FormDangKy.js U
src > components > JS FormDangKy.js > FormDangKy > validate
72
73     <label for="category">Category <span>{errors.category && touched.category && errors.category}</span></label>
74     <select id="category" name="category"
75         onChange={handleChange}
76         onBlur={handleBlur}
77         value={values.category}>
78         <option value="">Select</option>
79         <option value="Food">Food</option>
80         <option value="Entertainment">Entertainment</option>
81         <option value="Academic">Academic</option>
82     </select>
83
84     <input type="submit" value="Submit" disabled={isSubmitting} />
85 </form>
86 )

```

```

84     <input type="submit" value="Submit" disabled={isSubmitting} />
85 </form>
86 )
87 }
88 </Formik>
89 </div>
90 )
91 }
92 }
93 export default FormDangKy;

```

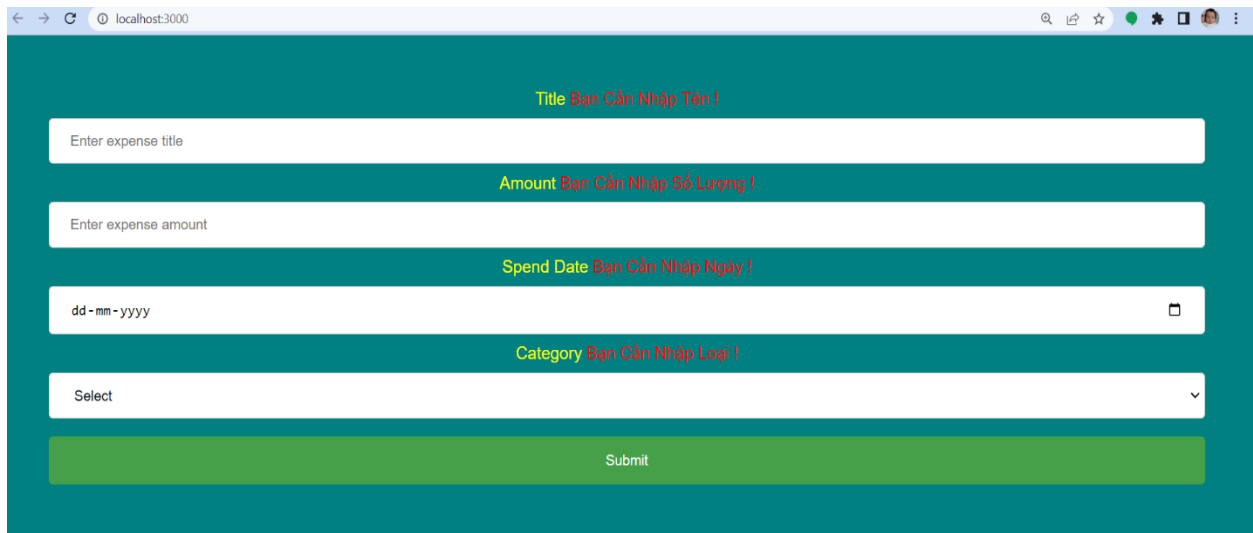
### Bước 3: Thực thi ứng dụng và xem kết quả

Sau khi đã tạo component xong, hãy thực thi ứng dụng và xem kết quả. Đầu tiên ta sẽ nhìn thấy form đăng ký đã được tạo và có giao diện như hình bên dưới.

The screenshot shows a web browser window at localhost:3000 displaying a registration form. The form is styled with a teal background and white input fields. It includes the following elements:

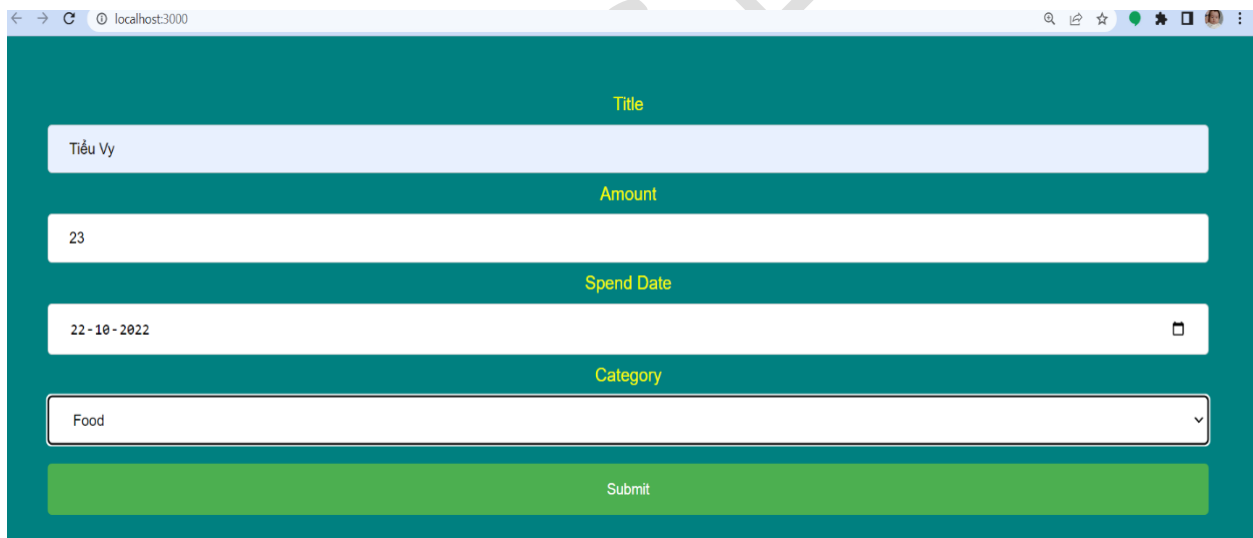
- Title:** A text input field with the placeholder "Enter expense title".
- Amount:** A text input field with the placeholder "Enter expense amount".
- Spend Date:** A date input field with the placeholder "dd-mm-yyyy" and a calendar icon on the right.
- Category:** A dropdown menu with the placeholder "Select".
- Submit:** A green button with the text "Submit".

Tiếp theo, mỗi khi người dùng chưa nhập dữ liệu và submit form, trên màn hình component sẽ hiển thị các thông báo lỗi mà ta đã khai báo trong component ở bước trên.



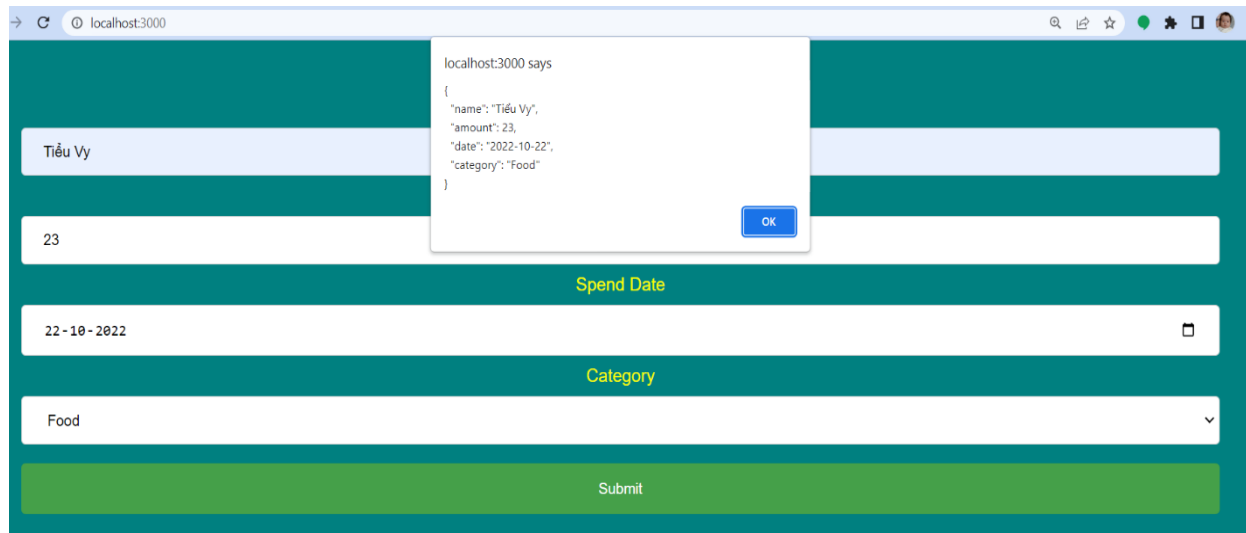
A screenshot of a web browser showing a form with four input fields and a submit button. The form is titled "Expense Form" and has a teal background. The input fields are labeled "Enter expense title", "Enter expense amount", "dd - mm - yyyy", and "Select". Above each input field is a red error message: "Title Bạn Cần Nhập Tên !", "Amount Bạn Cần Nhập Số Lượng !", "Spend Date Bạn Cần Nhập Ngày !", and "Category Bạn Cần Nhập Loại !". The submit button is green and labeled "Submit".

Sau khi ta nhập đầy đủ dữ liệu vào form như hình sau.



A screenshot of the same web browser showing the form with valid data. The input fields are filled with "Tiểu Vy", "23", "22 - 10 - 2022", and "Food". The error messages are gone. The submit button is still green and labeled "Submit".

Sau khi nhấn submit, ta có thể thấy dữ liệu của form đã được hiển thị trên hộp thoại alert() như hình bên dưới đúng như mong muốn.



The screenshot shows a web browser window with the address bar displaying 'localhost:3000'. The browser window contains a form with the following fields and labels:

- Name:** A text input field containing 'Tiểu Vy'.
- Amount:** A text input field containing '23'.
- Spend Date:** A date input field containing '22-10-2022'.
- Category:** A dropdown menu with 'Food' selected.
- Submit:** A green button at the bottom of the form.

An alert dialog box is displayed in the center of the browser window, titled 'localhost:3000 says'. The alert contains the following JSON data:

```
{
  "name": "Tiểu Vy",
  "amount": 23,
  "date": "2022-10-22",
  "category": "Food"
}
```

The alert dialog has an 'OK' button.