实验：内容评分应用程序以喜欢或不喜欢内容

预计所需时间： 40分钟

您将学习的内容

在本实验中，您将创建一个名为 'ContentRating' 的 React 组件，该组件允许用户通过点击 '喜欢' 或 '不喜欢' 按钮来对材料进行评分。当组件第一次渲染时，喜欢和不喜欢的计数都设置为零。您将创建一个方法，当用户选择 '喜欢' 按钮时，该方法会将喜欢的数量加一。同样，点击 '不喜欢' 按钮时，该方法会将不喜欢的数量加一。这个操作使用户能够通过喜欢或不喜欢按钮以互动的方式对内容进行评分，并向内容创作者提供反馈。

学习目标

完成本实验后，您将能够：

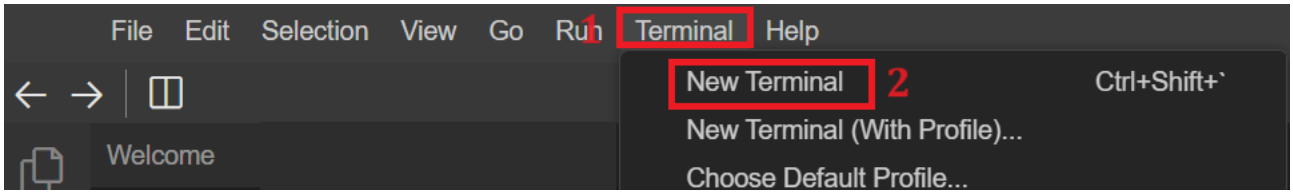
- 在 React 中使用状态变量处理组件状态，以跟踪喜欢和不喜欢的数量
- 在用户选择按钮以喜欢或不喜欢内容时执行事件处理
- 创建组件并通过构建一个包含内容和评分功能的切换组件来重用它们
- 使用 React 创建一个可以处理状态和用户事件并渲染复杂 UI 元素的互动用户界面

先决条件

- 基础的 HTML 知识
- 中级 JavaScript 知识
- 基础的 React 类组件和状态管理知识

步骤 1：设置环境

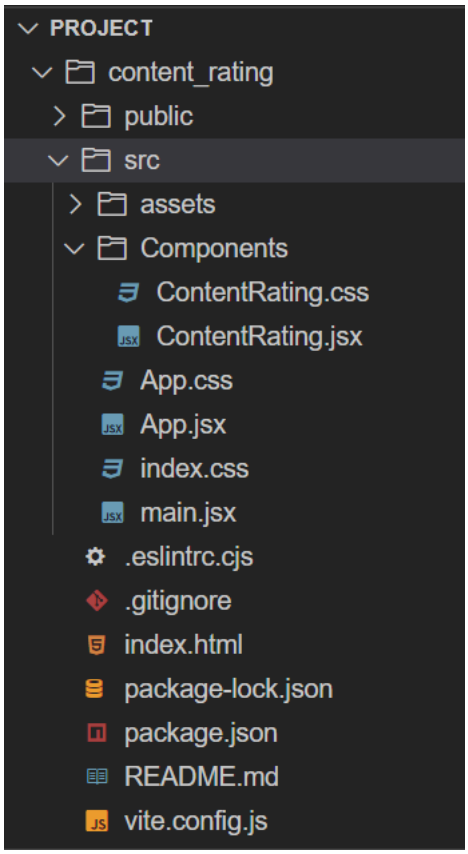
1. 从实验室顶部菜单中，点击窗口右上角的**终端**标签，如给定截图中编号 1 所示，然后点击**新建终端**，如编号 2 所示。



2. 现在，在终端中输入以下命令以克隆此 React 应用程序的模板，并按 Enter。

```
git clone https://github.com/ibm-developer-skills-network/content_rating.git
```

3. 上述命令将在“项目”文件夹下创建一个名为“content_rating”的文件夹，您可以在截图中看到其结构。这包括一个名为“CourseRating.jsx”的类组件和一个名为“CourseRating.css”的 CSS 文件。



4. 接下来，您需要在终端的路径中进入“content_rating”文件夹。为此，您需要在终端中输入给定的命令。此操作将设置您的终端路径，以便在 content_rating 文件夹中运行 React 应用程序。

   ```
   cd content_rating
   ```

5. 为确保您克隆的代码正常工作，您需要执行以下步骤：

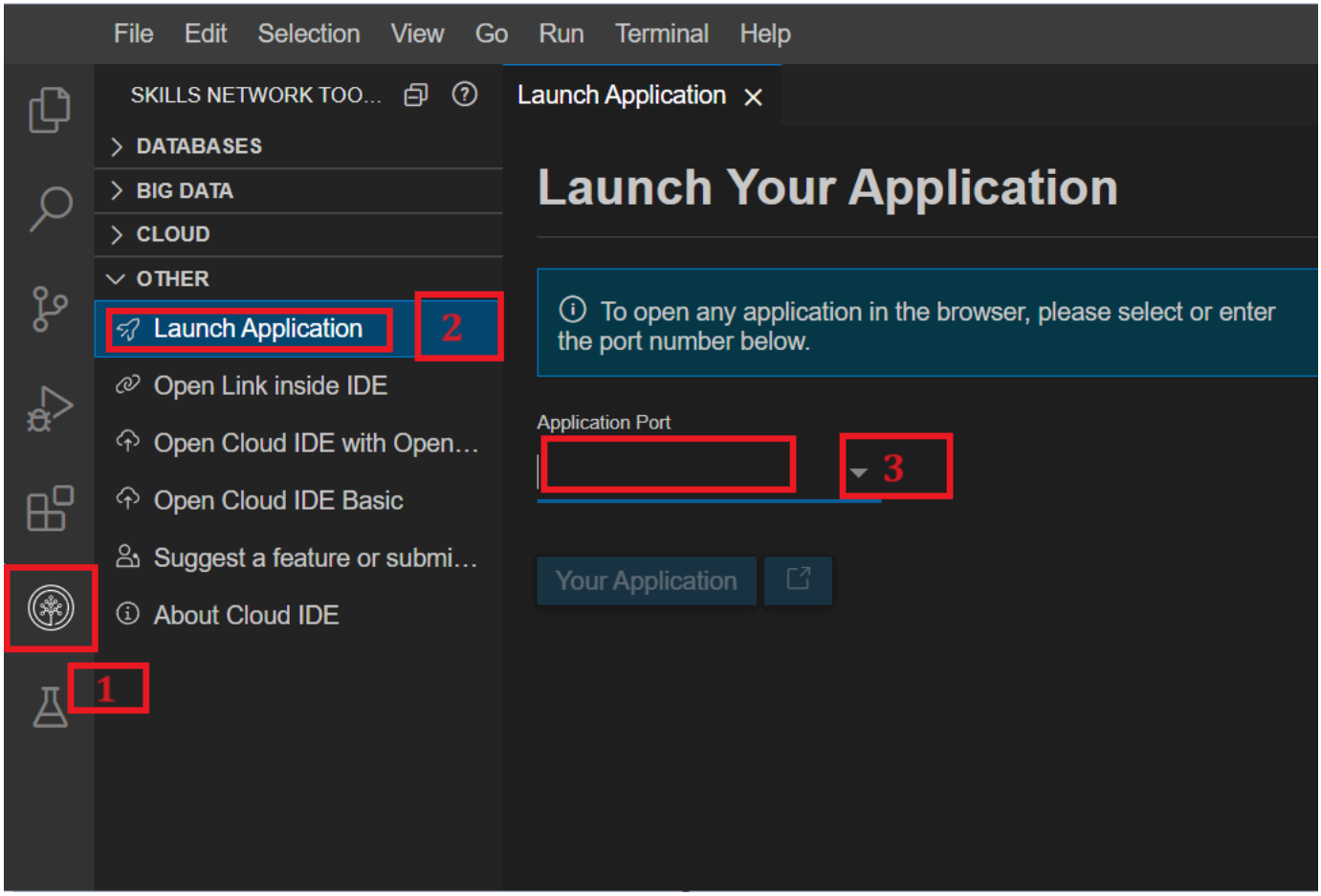
   - 在终端中输入给定的命令并按 Enter。此命令将安装执行应用程序所需的所有必要软件包。

     ```
     npm install
     ```

   - 然后，执行以下命令以运行应用程序，提供端口号 4173。

     ```
     npm run preview
     ```

6. 要查看您的 React 应用程序，请点击左侧的技能网络图标（参见编号 1）。此操作将打开**技能网络工具箱**。接下来，点击**启动应用程序**（参见编号 2）。在**应用程序端口**中输入端口号**4173**（参见编号 3），然后点击 。



7. 输出将显示在给定截图中，如下所示。



8. 您可以通过添加、提交并推送到您的 GitHub 存储库来保存您在此实验室中的最新工作。这确保了即使您没有持续进行任务，您的进度也会被保存，允许您从中断的地方继续。

注意：步骤 8 是可选的。

步骤 2：设置初始状态

1. 接下来，通过导航到 **src** 目录中克隆的 **content_rating** 文件夹里的 **Components** 文件夹中的 **ContentRating.jsx** 组件，打开评分组件。
2. 该组件的基本结构将如截图所示。

```
import React, { Component } from 'react'
import './ContentRating.css'
export class ContentRating extends Component {
    constructor() {
        super();
    }
  render() {
    return (
      <>
      <h1>Text Content Rating</h1>
      </>
    )
  }
}
export default ContentRating
```

3. 你需要在当前组件的构造函数中，在 **super()** 方法之后初始化喜欢和不喜欢计数的状态。创建 **this.state** 对象，并将喜欢和不喜欢的值初始化为 `0` 作为初始状态。

```
this.state = {
     likes: 0,
     dislikes: 0
   };
```

4. 你可以移除 `<h1>Text Content Rating</h1>` 元素，并在类组件的 `return` 中使用给定命令创建带有类名 **content-rating** 的 `div` 标签。

```
<div className='content-rating'></div>
```

这个 `div` 将作为其他标签的父 div。将给定命令包含在片段内。

5. 然后在 `div` 标签内创建一个 `<p> Add text here</p>` 标签，内容与用户可以喜欢或不喜欢的任何主题相关。

```
<div className='content-rating'>
    <p>
    //Add text here
    </p>
</div>
```

6. 在父 div 标签内的段落标签之后，再创建一个带有类名 **rating-buttons** 的 `<div>` 标签。

```
<div className='content-rating'>
    <p>
    //Add text here
    </p>
    <div className='rating-buttons'></div>
</div>
```

7. 在类名为 **rating-buttons** 的 div 内创建两个按钮，一个用于喜欢，另一个用于不喜欢。然后，在这些按钮中显示你在 `this.state` 对象下初始化的变量值作为文本。

```
    <div className='content-rating'>
        <p>
        ---Add text here---
        </p>
        <div className='rating-buttons'>
        <button className="like-button">
            Like ({this.state.likes})
          </button>
          <button className="dislike-button">
            Dislike ({this.state.dislikes})
          </button>
        </div>
    </div>
```

步骤 3：创建事件处理

1. 在此步骤中，您将创建处理点赞和点踩按钮点击事件的事件。

2. 在 `this.state` 对象下创建两个事件处理程序，分别命名为 **handleLike** 用于点赞按钮和 **handleDislike** 用于点踩按钮。

```
constructor() {
   super();
   this.state = {
     likes: 0,
     dislikes: 0,
     handleLike:() => {
     },
     handleDislike:() => {
     }
 }
   }
```

3. 现在，在这两个事件处理程序中编写代码逻辑，以增加和减少点赞和点踩按钮的值。

```
constructor() {
   super();
   this.state = {
     likes: 0,
     dislikes: 0,
     handleLike:() => {
       this.setState((prevState) => ({
         likes: prevState.likes + 1
       }));
     },
     handleDislike:() => {
       this.setState((prevState) => ({
         dislikes: prevState.dislikes + 1
       }));
     }
 }
   }
```

   - 在上述代码中，箭头函数 handleLike 和 handleDislike 在用户点击 "点赞" 或 "点踩" 按钮时被调用。
   - 这些函数利用 setState 方法更新组件的状态。在 setState 内部，访问先前的状态 (prevState)，它保存了更新之前组件的先前状态。
   - 然后，点赞或点踩计数增加一并设置为新的状态值。
   - 这确保每次按钮点击都会准确更新组件状态中的相应计数。

步骤 4：调用事件处理程序

1. 现在您需要使用喜欢和不喜欢按钮的点击来调用这些事件处理程序。

2. 为此，您需要在按钮上使用 `onClick` 事件，并分别在 **Likes** 和 **Dislikes** 按钮上调用 **handleLike** 和 **handleDislike**，如给定代码所示。

```
 <button className="like-button" onClick={this.state.handleLike}>
           喜欢 ({this.state.likes})
         </button>
         <button className="dislike-button" onClick={this.state.handleDislike}>
           不喜欢 ({this.state.dislikes})
</button>
```

步骤 5：检查输出

1. 现在通过在终端中执行 `ctrl+c` 来停止 React 应用程序的执行。

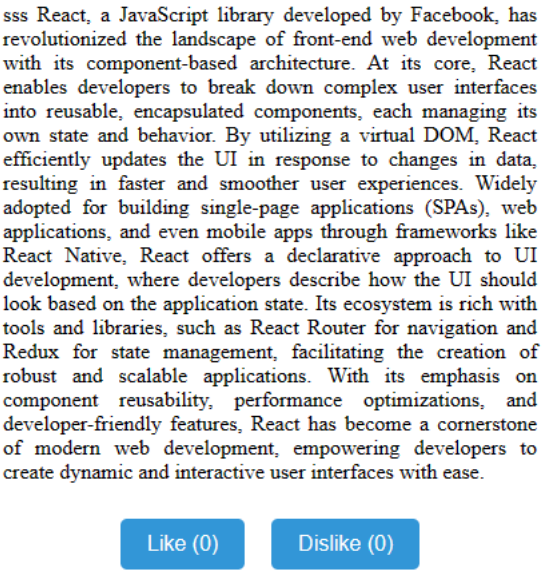
2. 然后，在终端中输入给定的命令并按 Enter。

```
npm run preview
```

3. 要查看您的 React 应用程序，请在浏览器中刷新已经打开的 React 应用程序网页。如果没有打开，请点击左侧面板上的 Skills Network 图标。

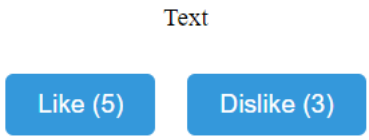
   此操作将打开“SKILLS NETWORK TOOLBOX”。接下来，选择“启动应用程序”。在“应用程序端口”中输入端口号 **4173** 并点击 。

4. 输出将根据给定的截图显示。



   此截图显示了 React 库的内容。您可以在 `<p>` 标签中写入自己的内容，以替代占位符 `//Add text here`。您的输出将基于您在 `//Add text here` 占位符中添加的文本。

5. 通过点击按钮检查点赞和点踩的功能。如果您点击 **like** 按钮 5 次和 **dislike** 按钮 3 次，则将根据给定的截图显示结果。



6. 这表示该文本获得了 5 个赞和 3 个踩。

7. ▼ 点击这里查看 "ContentRating.jsx" 的完整示例解决方案

```
import React, { Component } from 'react';
import './ContentRating.css';
class ContentRating extends Component {
  constructor() {
    super();
    this.state = {
        likes: 0,
        dislikes: 0,
      handleLike:() => {
        this.setState((prevState) => ({
            likes: prevState.likes + 1
          }));
      },
      handleDislike:() => {
        this.setState((prevState) => ({
            dislikes: prevState.dislikes + 1
          }));
      }
      };
  }
  render() {
    return (
     <>
     <h1>文本内容评分</h1>
     <div className='content-rating'>
        <p>文本</p>
        <div className='rating-buttons'>
        <button className="like-button" onClick={this.state.handleLike}>
            点赞 ({this.state.likes})
          </button>
          <button className="dislike-button" onClick={this.state.handleDislike}>
            点踩 ({this.state.dislikes})
          </button>
        </div>
     </div>
     </>
    );
  }
}
export default ContentRating;
```

8. ▼ 点击这里查看 "App.jsx" 父组件的完整示例解决方案

```
import ContentRating from "./Components/ContentRating"
 function App() {
   return (
    <>
   <ContentRating/>
    </>
   )
 }
 export default App
```

第6步：练习

1. 在这个练习中，您需要创建逻辑来计算在 `ContentRating.jsx` 组件中结合喜欢和不喜欢的总评分数量。

2. 为此，您需要在 `this.state` 中声明一个名为 `totalRatings` 的变量，并将其初始化为 0。

3. 现在创建逻辑来计算内容的喜欢和不喜欢的总数，并将总数存储在变量 `totalRatings` 中。

   提示：在增加总评分数量时，将 `totalRatings` 变量包含在喜欢和不喜欢的函数中。

▼ 点击这里查看答案

```
handleLike: () => {
       this.setState((prevState) => ({
         likes: prevState.likes + 1,
         totalRatings: prevState.totalRatings + 1
       }));
     },
     handleDislike: () => {
       this.setState((prevState) => ({
         dislikes: prevState.dislikes + 1,
         totalRatings: prevState.totalRatings + 1
       }));
     }
```

4. 现在在按钮标签后面使用 jsx 语法显示总评分。
   提示：使用 `{}` 来显示总评分变量，使用 **this.state**。

▼ 点击这里查看答案

```
<p>总评分: {this.state.totalRatings}</p>
```

5. 通过再次在终端中重新运行应用程序来检查输出。

6. 输出如提供的截图所示，当有 5 个喜欢和 4 个不喜欢时，总评分将显示为 9。点击喜欢或不喜欢按钮将相应地增加总评分。



Total Ratings: 9

7. ▼ 点击这里查看 "ContentRating.jsx" 的完整示例解决方案

```
import React, { Component } from 'react';
import './ContentRating.css';
class ContentRating extends Component {
  constructor() {
    super();
    this.state = {
      likes: 0,
      dislikes: 0,
      totalRatings: 0,
      handleLike: () => {
        this.setState((prevState) => ({
          likes: prevState.likes + 1,
          totalRatings: prevState.totalRatings + 1
        }));
      },
      handleDislike: () => {
        this.setState((prevState) => ({
          dislikes: prevState.dislikes + 1,
          totalRatings: prevState.totalRatings + 1
        }));
      }
    };
  }
  render() {
    return (
      <>
        <h1>文本内容评分</h1>
        <div className='content-rating'>
          <p>文本</p>
          <div className='rating-buttons'>
            <button className="like-button" onClick={this.state.handleLike}>
              喜欢 ({this.state.likes})
            </button>
            <button className="dislike-button" onClick={this.state.handleDislike}>
              不喜欢 ({this.state.dislikes})
            </button>
          </div>
          <p>总评分: {this.state.totalRatings}</p>
        </div>
      </>
    );
  }
}
export default ContentRating;
```

注意： – 要查看最新更改，您需要在终端中再次执行 `npm run preview`。

恭喜！您已创建了第二个 React 应用程序来对内容进行评分！

结论

- 在这个实验中，您学习了如何在 React 类组件的构造函数中设置和创建状态变量。状态变量使组件能够处理和跟踪动态数据，如“喜欢”和“不喜欢”。
- 您已经理解了如何在组件的状态中创建箭头函数，这些函数包含在用户执行某些操作（例如点击按钮）时更改特定状态变量的代码。
- 通过事件处理，您学习了如何将这些箭头函数连接到 React 中相关 UI 元素的 onClick 事件。这些函数使状态能够根据用户的交互动态变化。
- 您还获得了将动态状态数据添加到渲染的 UI 中的信息，使用 JSX 标记。这显示了当前的喜欢和不喜欢的计数以及描述性内容，为用户提供了完整的内容评分方式。

Author(s)

Richa Arora

© IBM Corporation. All rights reserved.