Chapter 2

Create simple React app with Props (properties)

Prosp system in React JS

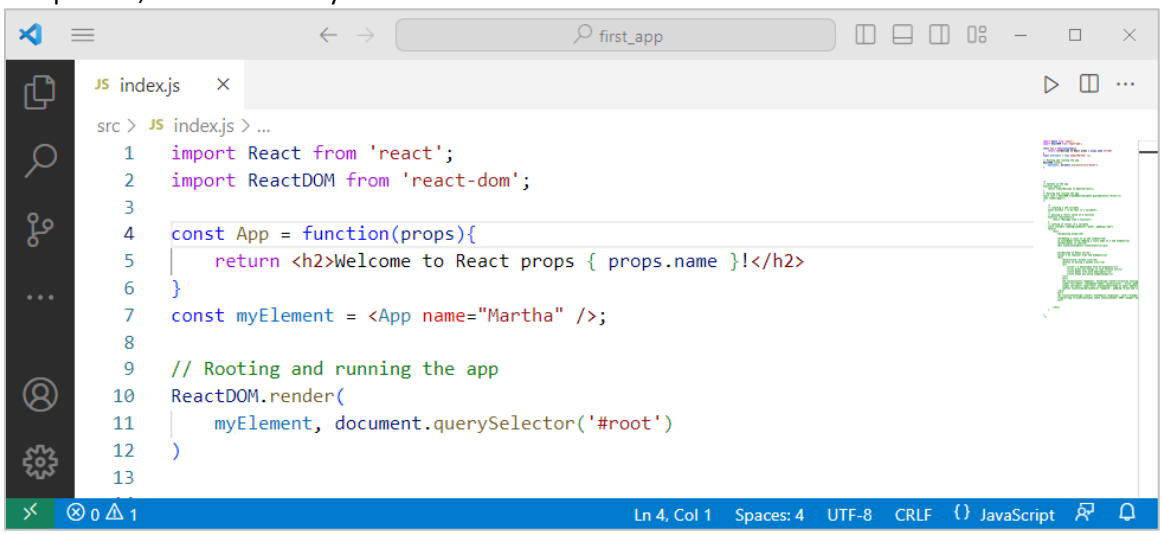
props stands for **'properties'** and are arguments passed into React components via HTML attributes. **props** are also pass data from one component (parent) to another (child) component as parameters. Props purpose is to customize or configure a child component. Conceptually, components are like JavaScript functions. They accept arbitrary inputs (called "props") and return React elements describing what should appear on the screen.

The simplest way to define a component using props is to write a JavaScript function and pass props as an event:

```
function Welcome(props) {
  return <h2>Welcome to React props {props.name}!</h2>;
}
```

This function accepts a single "props" object argument with data and returns a React element.

React **props** are like function arguments in JavaScript and attributes in HTML. To send props into a component, use the same syntax as HTML attributes:



React Props can help programmers to reuse and nest component.

- Component nesting is when a component can be shown inside of another component.
- Component reusability helps programmers to reduce the code lines of repeating component.
- Programmers should be able to configure a component when it is created.

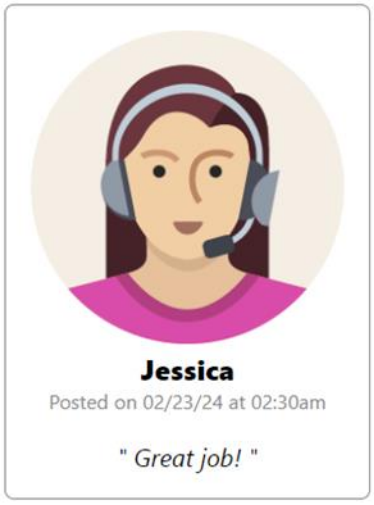
How to use props to create reusable components?

To use Props to create reusable components by following steps:

1. Identify the JSX that appears to be duplicated.
2. Think of a descriptive name for reusable component.
3. Create a new file for reusable components.
4. Create a new component in the new file.
5. Make the new component configurable by using **props** system.

Let's create the following app:

For this app, we can identify that the JSX element that appears to the duplicate is a card with an image at the top and three text components at the bottom:



Setting up the React environment

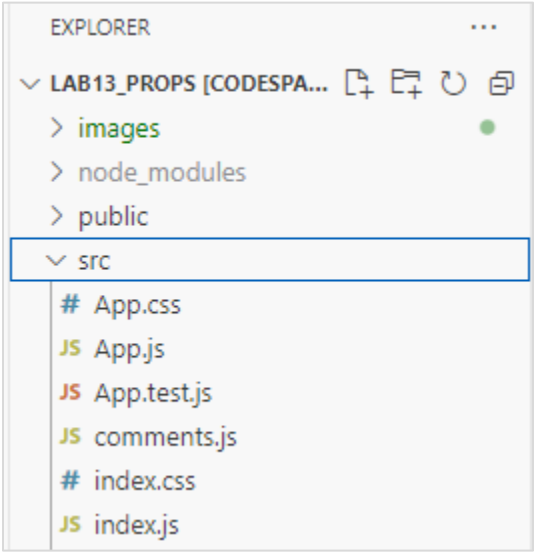
For first time React user, install a React app through the Terminal by typing:

```
npm install -g create-react-app
```

Once the 'create-react-app' is installed, create a react app project named 'lab13_props' as:

```
create-react-app lab13_props
```

After it, we can create a new .js file named 'comments.js' to create the UI of the user card. The 'comments.js' file should go in the 'src' folder:



In the *comments.js* file, we are going to create the structure of the main JSX component. This main JSX component is created in a function named *User*. Once the structure is complete, the *User* function is exported to the **index.js** file using the link: `import User from './comments';`

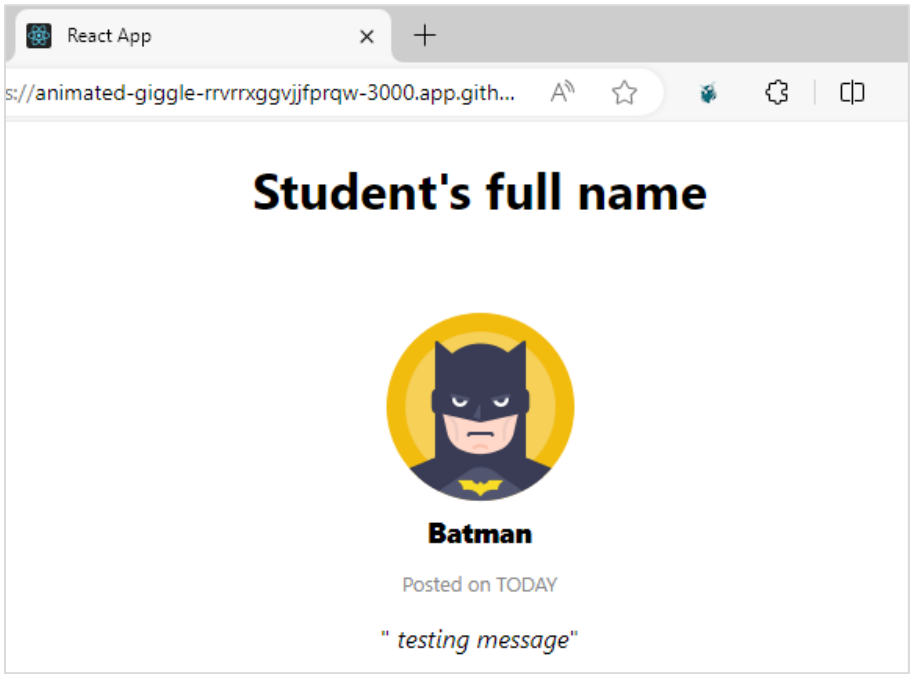
comments.js

```
export default User;
import React from "react";
import avataruser1 from './images/batman.png';

const User = function(props){
    return(
        <>
            <section className="card">
                <a> <img src={ avataruser1 }/> </a>
                <div className="content">
                    <a className="author">Batman</a>
                </div>
                <div className="metadata">
                    <p>Posted on <span  className="date"> TODAY </span></p>
                </div>
                <div className="comment">
                    <p>"<i> testing message</i>"</p>
                </div>
            </section>
        </>
    )
}

export default User;
```

Now, let us create the main JSX component in the **index.js** file.

```
import React from 'react';
import ReactDOM from 'react-dom/client';
// import files
import './index.css';
import User from './comments'

const App = function(){
  return(
    <>
      <h1 style={{textAlign:"center"}}>Student's full name</h1>
      <section className='container'>
        {/* user 1 */}
       <User />
      </section>
    </>
  )
}
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

Running the server, **npm start**, should show the following:



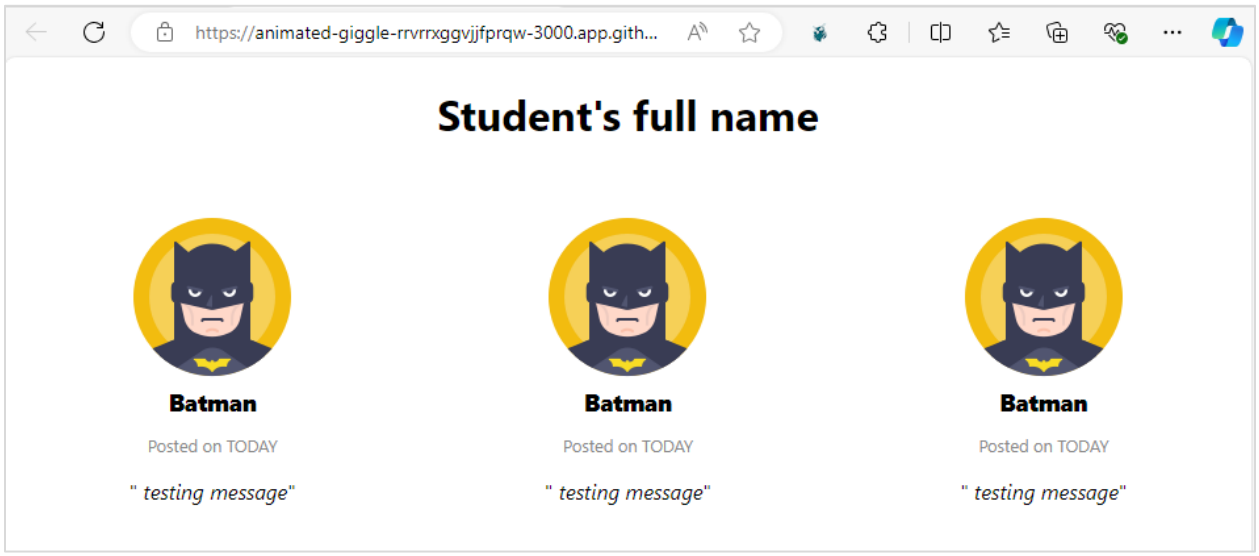
The styling of the react app, *index.css,* is:

```
/* CARD - COMMENTS.JS */
.card{
  padding: 1em;
  width: 90%;
  text-align: center;
  margin: 1em auto;
  border-radius: 10px;
  transition: ease-in-out 100ms;
}
.content{
  font-size: 1.1em; font-weight: 900;
}
.metadata{
  font-size:0.8em; color: #888;
}
.container{
  display: flex;
  justify-content: center;
  flex-wrap: nowrap;
}
```

If you want to use the same JSX structure <User />, we call <User /> again in the ***index.js*** file:

```
import React from 'react';
import ReactDOM from 'react-dom/client';
// import files
import './index.css';
import User from './comments'

const App = function () {
  return (
    <>
      <h1 style={{ textAlign: "center" }}>Student's full name</h1>
      <section className='container'>
        {/* user 1 */}
        <User />
        {/* user 2 */}
        <User />
        {/* user 3 */}
        <User />
      </section>
    </>
  ) }
```

The internet browser, *localhost:3000,* should look as:



After the main JSX component is created, we move to create a child component in the **index.js** file. In the component, we will create properties called *name, msg,* and *image* and assign them to three different components, *Users:*

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';

// import all images
import avataruser1 from './images/batman.png';
import avataruser2 from './images/businessman.png';
import avataruser3 from './images/lady.png';

// import files
import './index.css';
import User from './comments'

const App = function(){
  return(
   <>
    <h1 style={{textAlign:"center"}}>Student's full name</h1>
    <section className='container'>
    {/* user 1 */}
    <User name="Batman" image={avataruser1} date="11/12/24" msg="I am Batman"/>

    {/* user 2 */}
    <User name="Peter" image={avataruser2} date="10/03/24" msg="I am late!"/>
    {/* user 3 */}
    <User name="Lady" image={avataruser3} date="08/20/24" msg="Get more rest"/>
   </section>
 </>
 )
}
```

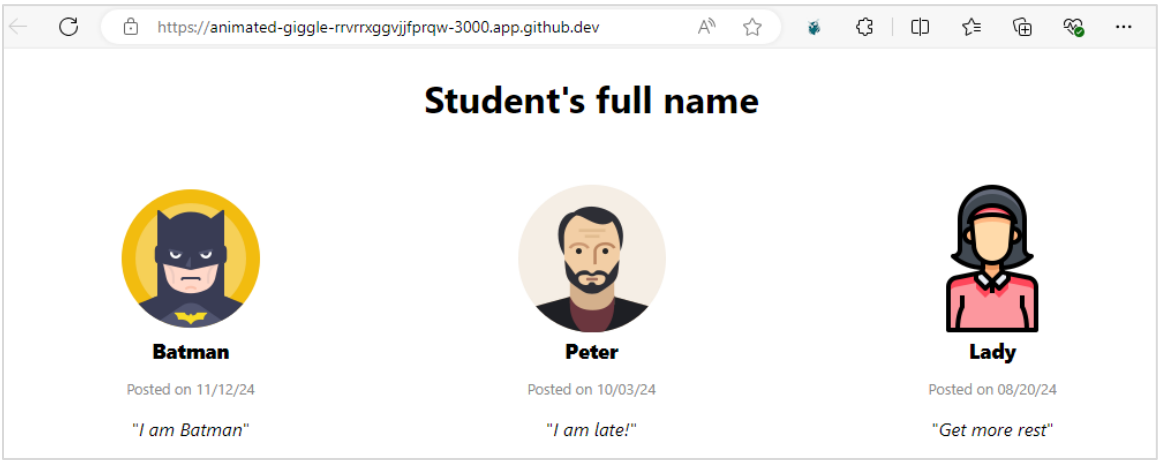
```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

Now we are going to pass the properties **name, msg, date,** and **image** from the *index.js* file into the component in *comments.js* file.

comments.js

```
import React from "react";

const User = function(props){
    return(
        <>
            <section className="card">
                <a> <img src={props.image}/> </a>
                <div className="content">
                    <a className="author">{props.name}</a>
                </div>
                <div className="metadata">
                    <p>Posted on <span  className="date">{props.date} </span></p>
                </div>
                <div className="comment">
                    <p>"<i>{props.msg}</i>"</p>
                </div>
            </section>

        </>
    )
}

export default User;
```

In the browser, *localhost:3000,* it should see an update of each props' name, msg, date, and image properties:



Adding props children in React

We almost done! The last component we are adding to our app is the link that allows the user to add a comment. To do so, we will create a file that is used to create a window to add comment to an existing component. Let's create a new file called ***user_feedback.js*** in the **src** folder to set up the feedback component. We are creating and test the feedback component first:

User_feedback.js

```
import React from "react";
import './index.css'
const User_feedback = function(){
    return(
       <>
          <div className="feedbackcontainer">
            <section className="feedbackcard">
                <div className="content"><p>Name of user</p></div>
                <div className="description">Some feedback</div>
                <div className="cardfooter">
                  <p className="addicon"><span>&#10011;</span>Add feedback</p>
                </div>
            </section>
          </div>
        </>
    )
}
export default User_feedback
```

We can update the style of the feedback card in the 'index.css' file as:

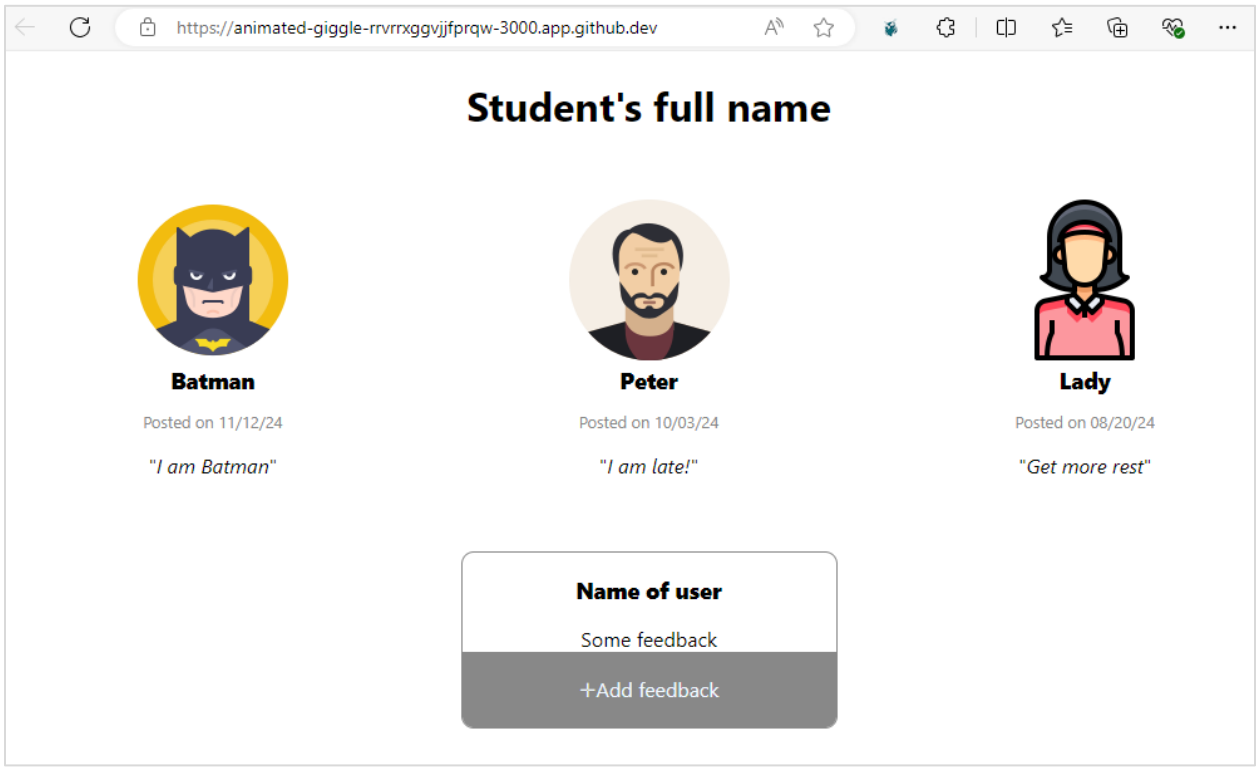
```
/* USER_FEEDBACK.JS*/
.feedbackcard{
  width: 300px;
  border: #888 solid 1px;
  border-radius: 10px;
  text-align: center;
  overflow: hidden;
}
.cardfooter{
  background-color: #888;
  padding: 0.2em;
  color: aliceblue;
}
.feedbackcontainer:hover{
  box-shadow: 0px 0px 10px 1px gray;
  transform: scale(1.01);
  overflow: hidden;
  border-radius: 10px;
}
.feedbackcontainer{
  margin: 1%;
}
```

We import the u*ser_feedback* to the index.js file using the *import* keyword.

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';

// import all images
import avataruser1 from './images/batman.png';
import avataruser2 from './images/businessman.png';
import avataruser3 from './images/lady.png';

// import files
import User_feedback from './user_feedback';
import './index.css';
import User from './comments'

const App = function(){
  return(
   <>
    <h1 style={{textAlign:"center"}}>Student's full name</h1>
    <section className='container'>
    {/* user 1 */}
    <User name="Batman" image={avataruser1} date="11/12/24" msg="I am Batman"/>

    {/* user 2 */}
    <User name="Peter" image={avataruser2} date="10/03/24" msg="I am late!"/>
    {/* user 3 */}
    <User name="Lady" image={avataruser3} date="08/20/24" msg="Get more rest"/>
   </section>

   <section className='container'>
     <User_feedback></User_feedback>
   </section>
 </>
 )
}

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

The browser should look as:



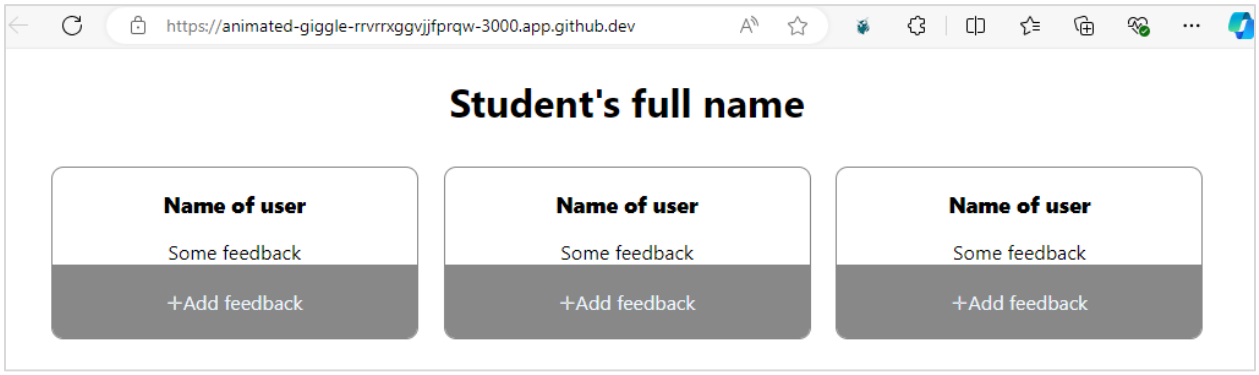
We can also make a component to a child component by passing the child component using props. For this, in the *index.js* file, we are going to use the `<User>` component as the child of the `<User_feedback>` component:

comments.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';

// import all images
import avataruser1 from './images/batman.png';
import avataruser2 from './images/businessman.png';
import avataruser3 from './images/lady.png';

// import files
import User_feedback from './user_feedback';
import './index.css';
import User from './comments'

const App = function(){
  return(
   <>
    <h1 style={{textAlign:"center"}}>Student's full name</h1>

    <section className='container'>
```

```
      {/* user 1 */}
      <User_feedback><User name="Batman" image={avataruser1} date="11/12/24"
msg="I am Batman"/></User_feedback>
      {/* user 2 */}
      <User_feedback><User name="Peter" image={avataruser2} date="10/03/24" msg="I
am late!"/></User_feedback>
      {/* user 3 */}
      <User_feedback><User name="Lady" image={avataruser3} date="08/20/24"
msg="Get more rest"/></User_feedback>

    </section>
  </>
  )
}

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

The browser will look as:



After it, we need to use 'props' in the 'User_feedback.js' file to pass the children UI into the feedback cards:

User_feedback.js

```
import React from "react";
import './index.css'

const User_feedback = function(props){
    return(
        <>
          <div className="feedbackcontainer">
            <section className="feedbackcard">
              <div className="content"><p>{props.username}</p></div>
              <div className="description">{props.children}</div>
              <div className="cardfooter">
                 <p className="addicon"><span>&#10011;</span>Add feedback</p>
              </div>
```

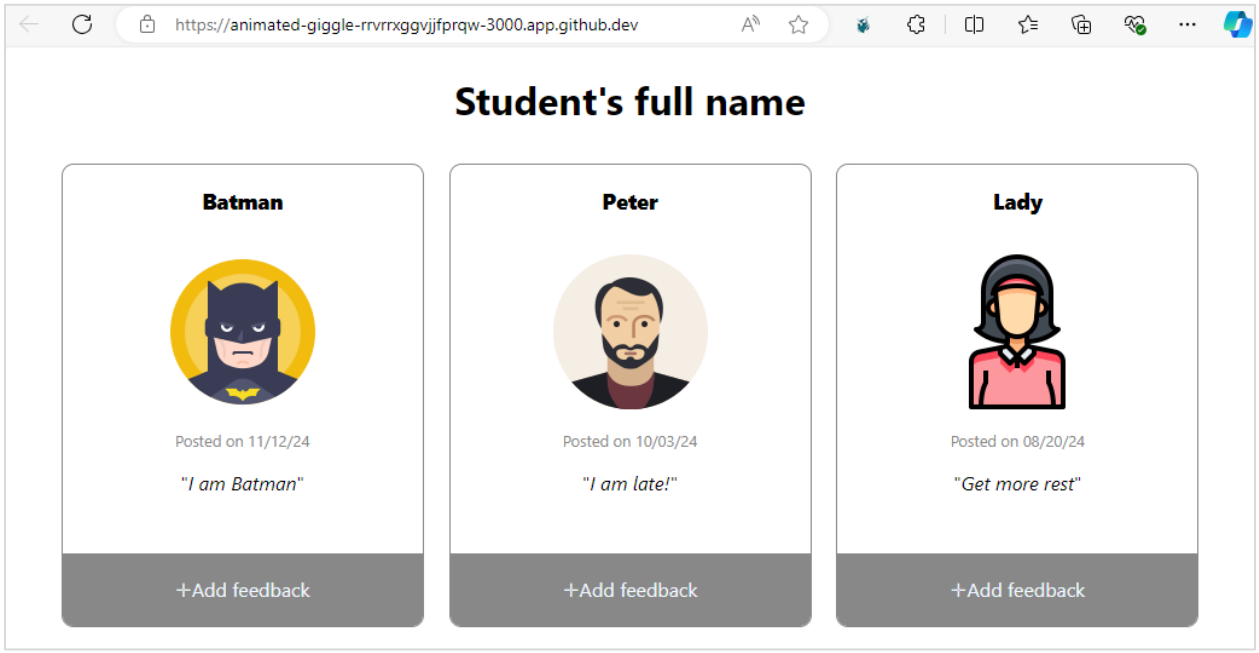
```
            </section>
          </div>
        </>
    )
}
export default User_feedback
```

Now, we need to pass the 'props' in 'user_feedback.js' into the 'index.js' as:

Index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';

// import all images
import avataruser1 from './images/batman.png';
import avataruser2 from './images/businessman.png';
import avataruser3 from './images/lady.png';

// import files
import User_feedback from './user_feedback';
import './index.css';
import User from './comments'

const App = function(){
  return(
   <>
    <h1 style={{textAlign:"center"}}>Student's full name</h1>
    <section className='container'>

     {/* user 1 */}
     <User_feedback username="Batman"><User image={avataruser1} date="11/12/24"
msg="I am Batman"/></User_feedback>
     {/* user 2 */}
     <User_feedback username="Peter"><User image={avataruser2} date="10/03/24"
msg="I am late!"/></User_feedback>
     {/* user 3 */}
     <User_feedback username="Lady"><User image={avataruser3} date="08/20/24"
msg="Get more rest"/></User_feedback>

   </section>
 </>
 )
}

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

The final app will look as the following in the browser:



Adding event listener

To add an event in React, you'll attach event handlers to your React components. Event handlers in React are defined as functions, and they can be added as props to elements.

Here's a step-by-step guide:

---

1. Set Up Your React Component

Create a React component where the event will be added.
import React from 'react';

```
function App() {
    return (
      <div>
        <button>Click Me</button>
      </div>
    );
  }

  export default App;
```

2. Define an Event Handler

Create a function in your component to handle the event.

```
function App() {
    const handleClick = () => {
      alert('Button clicked!');
    };

    return (
      <div>
        <button>Click Me</button>
      </div>
    );
  }
```

3. Attach the Event

Pass your event handler to the element as a prop. The event name should be camelCase (e.g., `onClick`).

```
function App() {
    const handleClick = () => {
      alert('Button clicked!');
    };

    return (
      <div>
        <button onClick={handleClick}>Click Me</button>
      </div>
    );
  }
```

4. Use Inline Handlers (Optional)

You can define the event handler inline, though this is generally not recommended for more complex handlers.

```
function App() {
    return (
      <div>
        <button onClick={() => alert('Button clicked!')}>Click Me</button>
      </div>
    );
  }
```

5. Pass Arguments to Event Handlers

Use an arrow function or .bind() to pass arguments to the event handler.

```
function App() {
    const handleClick = (message) => {
      alert(message);
    };

    return (
      <div>
        <button onClick={()=>handleClick('Hello from React!')}>Click Me
        </button>
      </div>
    );
  }
```

Supported Events in React

React supports many DOM events, such as:

- Mouse events: `onClick`, `onDoubleClick`, `onMouseEnter`, `onMouseLeave`, etc.
- Form events: `onChange`, `onSubmit`, `onFocus`, `onBlur`, etc.
- Keyboard events: `onKeyDown`, `onKeyPress`, `onKeyUp`.
- And more: `onScroll`, `onLoad`, `onError`.

Exercise:

For our app, we are going to add an event to the 'Add Feedback' button so when the button is clicked it will open a modal window to add a Feedback.

Modalwindow.js

```
import React from "react";
import './index.css'

const Modalwindow = function (props) {
  // function to close the modal window
  const closemodalwindow = () => {
     const modalwindow = document.querySelector(".modalwindow")
     modalwindow.style.display = "none";
  }

  // function to collect comment
  const collectcomment = () => {
    let commentarea = document.querySelector(".commentarea")
    const commentlist = document.querySelector(".commentlist")
```

```
    commentlist.innerHTML += `<li>${props.usercomment} -
${commentarea.value}</li>`
    const modalwindow = document.querySelector(".modalwindow")
    modalwindow.style.display = "none";
    commentarea.value = ""
 }

 return (
   <>
     {/* Modal window */}
     <section className="modalwindow">
       <div className="modalcontent">
         <header className="modalheader">
            <p>Add Feedback</p>
            <p className="closemodal" onClick={closemodalwindow}>&#x58;</p>
         </header>
         <main className='modalbody'>
            <input placeholder='Type your comments...' className='commentarea'
            type="text" />
            <p className="description_comment">Max 200 characters</p>
            <button className="btnpostfeedback" onClick={collectcomment} >
            Post Feedback</button>
         </main>
        </div>
       </section>
  </>
    )
}

export default Modalwindow
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';

// import all images
import avataruser1 from './images/batman.png';
import avataruser2 from './images/businessman.png';
import avataruser3 from './images/lady.png';

// import files
import User_feedback from './user_feedback';
import './index.css';
import User from './comments'

const App = function () {
  return (
    <>
      <h1 style={{ textAlign: "center" }}>Student's full name</h1>

      <section className='container'>
        {/* user 1 */}
        <User_feedback username="Batman"><User image={avataruser1}
date="11/12/24" msg="I am Batman" /></User_feedback>
        {/* user 2 */}
        <User_feedback username="Peter"><User image={avataruser2} date="10/03/24"
msg="I am late!" /></User_feedback>
        {/* user 3 */}
        <User_feedback username="Lady"><User image={avataruser3} date="08/20/24"
msg="Get more rest" /></User_feedback>
      </section>

      {/* POST COMMENTS */}
      <div className="postcomment">
        <ul className="commentlist"></ul>
      </div>
    </>
  )
}

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

index.css

```
.modalwindow{
  background-color: rgba(200,200,200,0.7);
  position: fixed;
  top:0%; bottom: 0%; right: 0%; left: 0%;
}
.modalcontent{
  width: 500px; height: 300px; margin: 100px auto; background-color: white
}
.modalheader{
  border-bottom: ridge 1px lightgray;
  display: flex;
  flex-wrap: wrap;
  padding: 0.1em 2em;
  color: #888;
  font-weight: bolder;
}
.modalheader p:nth-child(1){ font-size: 1.1em; }
.modalheader p:nth-child(2){ margin-left: auto;}
.commentarea{
  margin: 2em;   padding: 10px 15px;
  border-radius: 5px;
  border: solid 1px lightgray;
  width: 90%;
}
.btnpostfeedback{
  margin: 2em;
  background-color: #888;
  text-align: center;
  border: none;
  padding: 10px 20px;
  color:white;
  border-radius: 5px;
}
.btnpostfeedback:hover{
  background-color: crimson;
  cursor: pointer;}
/* POST COMMENTS */
.postcomment{
  display: block;
  width: 1200px;
  margin: 2em auto;
  padding: 2em;
  background-color: rgba(200,200,200,0.5);
}
```

user_feedback.js

```
import React from "react";
import './index.css';
import Modalwindow from "./modalwindow";

const User_feedback = function (props) {
    // open modal window
    const openmodalwindow = () => {
        const modalwindow = document.querySelector(".modalwindow")
        modalwindow.style.display = "block";
    }

    return (
      <>
        <div className="feedbackcontainer">
           <section className="feedbackcard">
             <div className="content"><p>{props.username}</p></div>
             <div className="description">{props.children}</div>
             <div className="cardfooter">
                <p className="addicon" onClick={openmodalwindow}><span>
                   &#10011;</span>Add feedback</p>
             </div>
           </section>
        </div>

        {/* MODAL WINDOW */}

        <Modalwindow /> </>
    )
}

export default User_feedback
```