COP 4808-001: Full-Stack Web Development

PreWork

Due: Jan 21 at 11:59pm

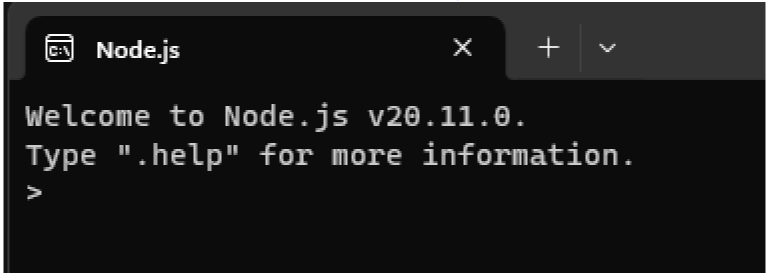
GitHub Username: ccabreragarc

GitHub Repo URL: https://github.com/ccabreragarc/Lab1.1

Step 0: Create a New React Project Using Vite

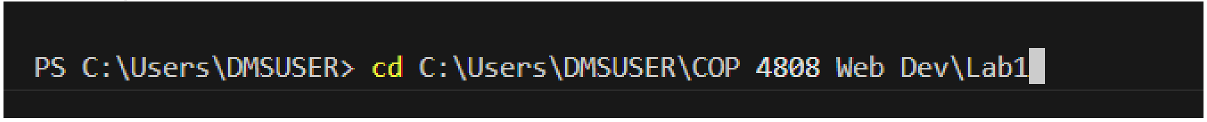
In this step, we will create a new React project using Vite.

- Download and install Node.js



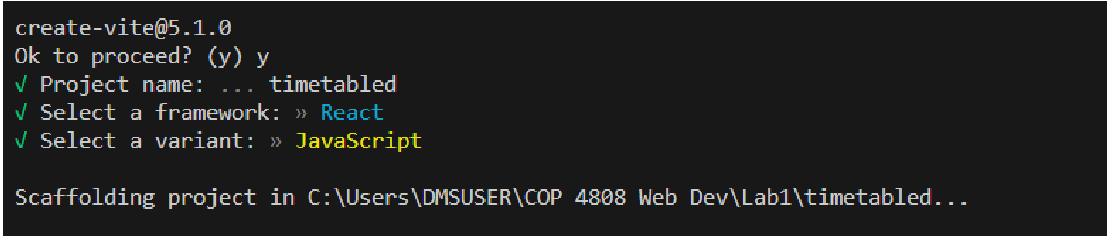
```
Welcome to Node.js v20.11.0.
Type ".help" for more information.
>
```

- Open the Terminal in VS Code



```
PS C:\Users\DMSUSER> cd C:\Users\DMSUSER\COP 4808 Web Dev\Lab1
```

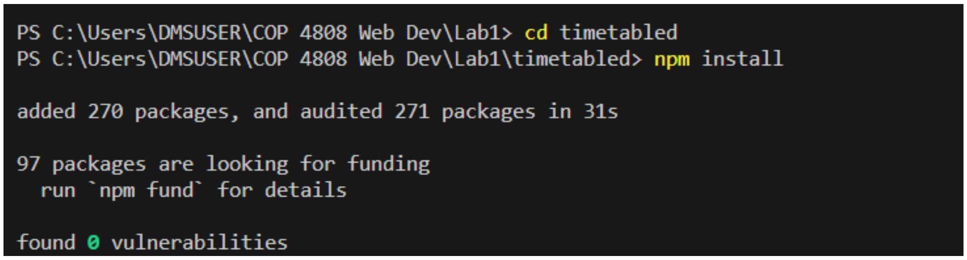
- Initialize a new React project using Vite:
  - In the Terminal, enter the command `npm create vite@latest`
  - Name the project `timetabled`
  - Select **React** as the framework (use the arrow keys and `enter` to navigate the menu)
  - Select **JavaScript** for the language variant



```
create-vite@5.1.0
Ok to proceed? (y) y
√ Project name: ... timetabled
√ Select a framework: » React
√ Select a variant: » JavaScript

Scaffolding project in C:\Users\DMSUSER\COP 4808 Web Dev\Lab1\timetabled...
```

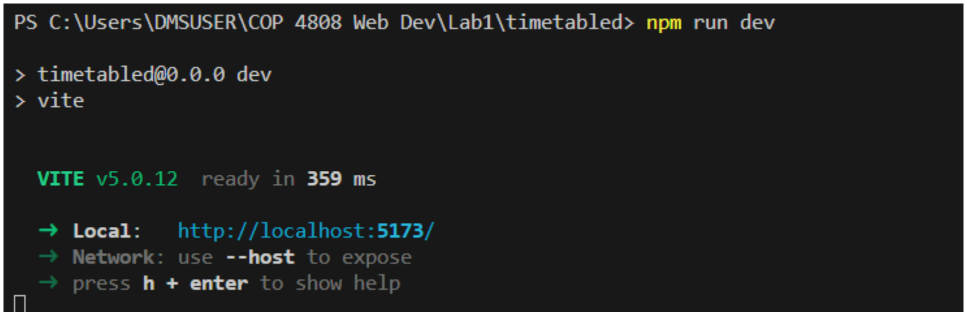
Now, let's install the required dependencies and run the app!

- Move into the `timetabled` directory: `cd timetabled`
- Install the dependencies by running the command `npm install`



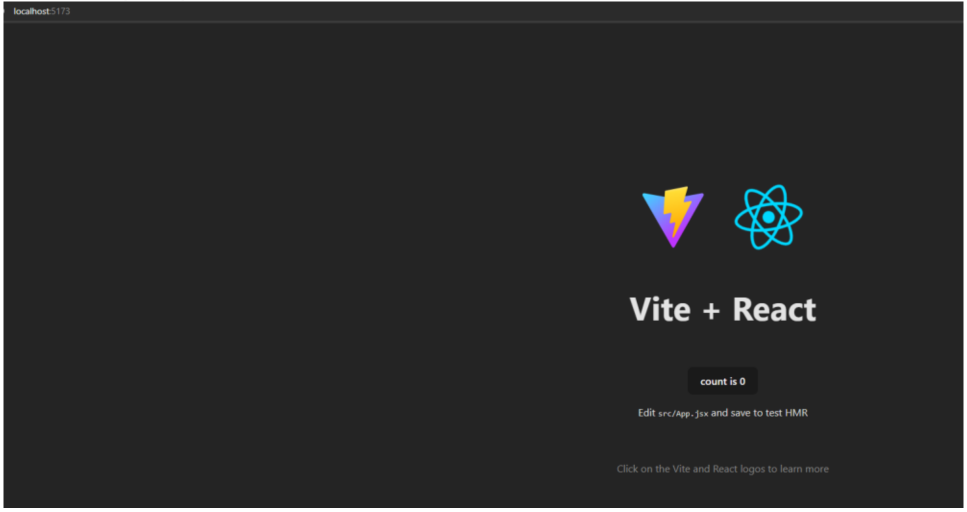
```
PS C:\Users\DMSUSER\COP 4808 Web Dev\Lab1> cd timetabled
PS C:\Users\DMSUSER\COP 4808 Web Dev\Lab1\timetabled> npm install

added 270 packages, and audited 271 packages in 31s

97 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

- Run the application in developer mode by running the command `npm run dev`



```
PS C:\Users\DMSUSER\COP 4808 Web Dev\Lab1\timetabled> npm run dev

> timetabled@0.0.0 dev
> vite


  VITE v5.0.12  ready in 359 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

- Open project in the browser



Vite + React

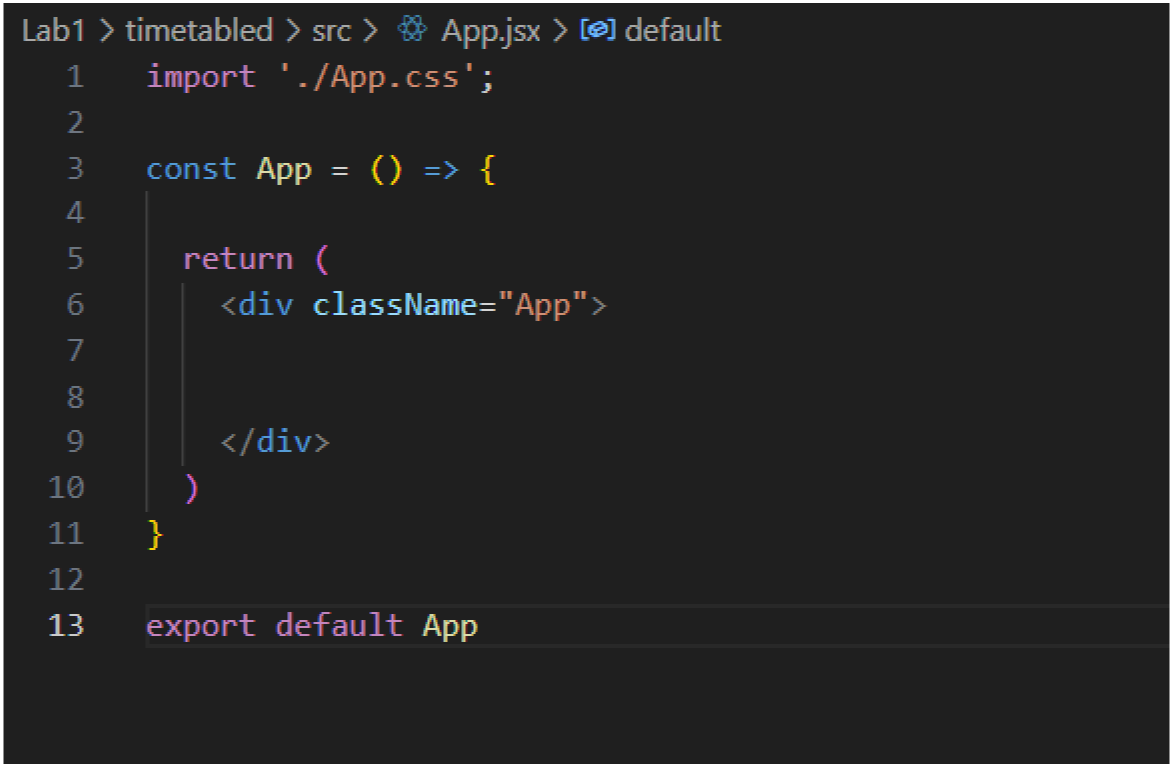
Edit `src/App.jsx` and save to test HMR

Click on the Vite and React logos to learn more

Step 1: Update the Starter Code for Root File `App.jsx`

- Go to the root file `App.jsx` in the `src` directory.

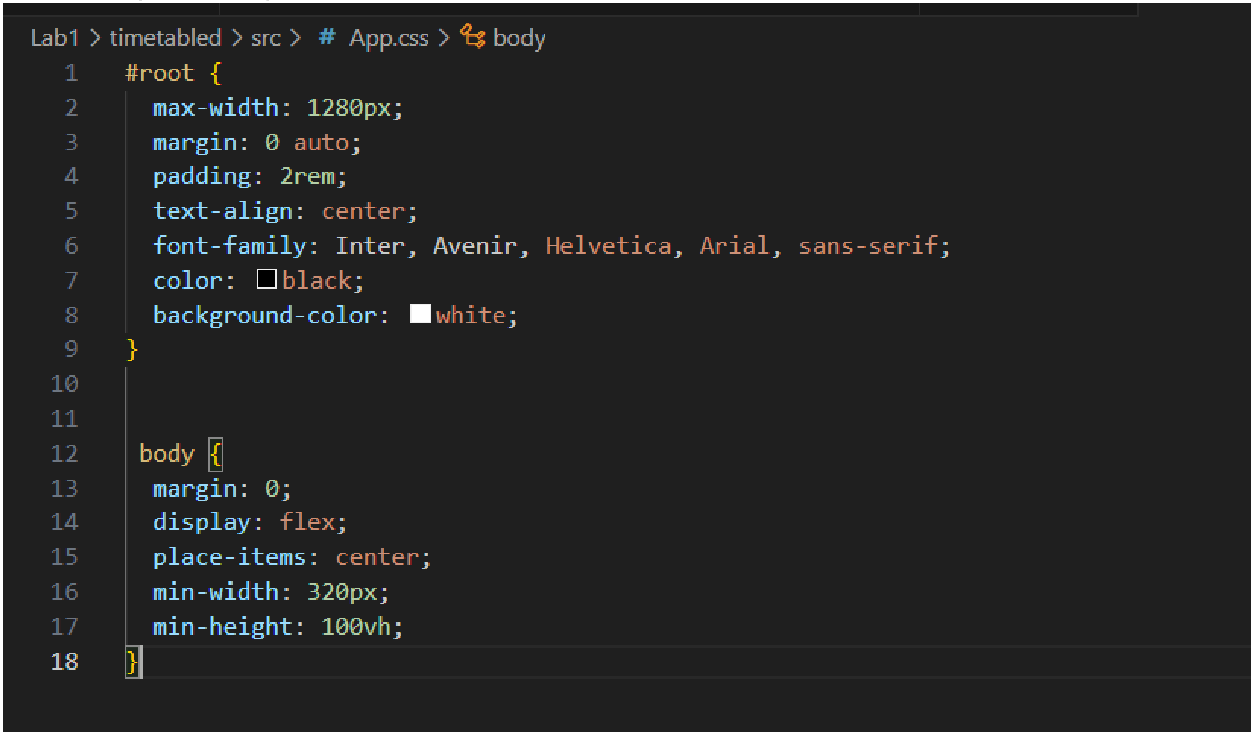
- Replace the provided starter code in `App.jsx`



Lab1 > timetabled > src > App.jsx > default

```jsx
import './App.css';

const App = () => {

  return (
    <div className="App">


    </div>
  )
}

export default App
```

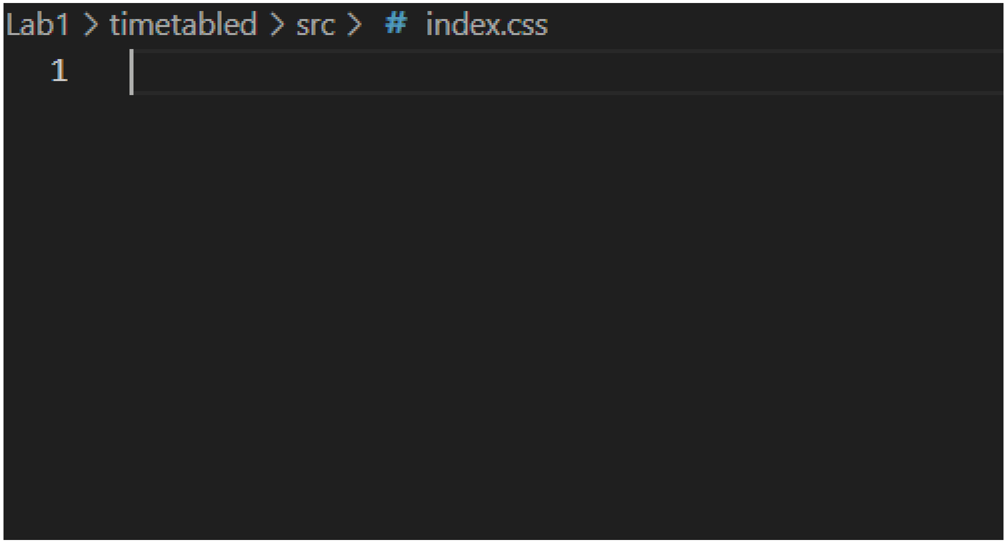
- Open the `App.css` file in the `src` directory.
- Replace the provided starter code in `App.css`



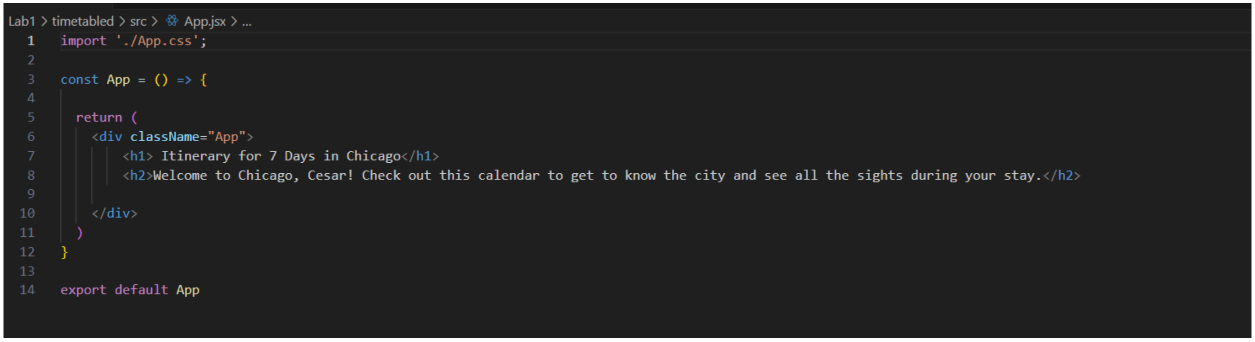
Lab1 > timetabled > src > # App.css > body

```css
#root {
  max-width: 1280px;
  margin: 0 auto;
  padding: 2rem;
  text-align: center;
  font-family: Inter, Avenir, Helvetica, Arial, sans-serif;
  color: black;
  background-color: white;
}


 body {
  margin: 0;
  display: flex;
  place-items: center;
  min-width: 320px;
  min-height: 100vh;
}
```

- Remove all of the CSS style rules in `index.css`.



- In the `div` in `App.jsx`
- Create an `h1` element with the title of your project.
- Create an `h2` element with a subtitle containing a message for your schedule recipient.



```
import './App.css';

const App = () => {

  return (
    <div className="App">
        <h1> Itinerary for 7 Days in Chicago</h1>
        <h2>Welcome to Chicago, Cesar! Check out this calendar to get to know the city and see all the sights during your stay.</h2>

    </div>
  )
}

export default App
```

Step 2: Create the `Calendar` Component

In this step, you will create a `Calendar` component with tester code. You'll then import and add it to your web application.

- In the `src` directory, create a subdirectory called `components`.
- In the `components` directory, create a file called `Calendar.jsx`.

In `Calendar.jsx`:

- Import `React` from the `react` library.
- Create a React functional component
- Between the curly braces, write a return statement followed by a pair of parentheses.

Between the parentheses, we'll write JSX statements to construct the component.

- Create a `div` where `className="Calendar"`
- Inside the `div`, write the text `"Testing the calendar component"`.
- At the end of the file, export the component by writing:

```
export default Calendar;
```

This exports the component so we can import it into our `App`.

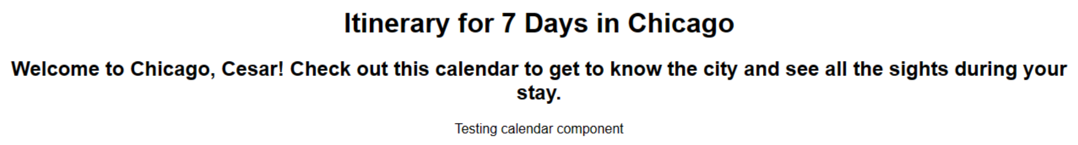
In `App.jsx`:

- Import the `Calendar` component.

```
import Calendar from './components/Calendar'
```

- After the `h2` element, add a `Calendar` component.

```
<Calendar />
```



Itinerary for 7 Days in Chicago

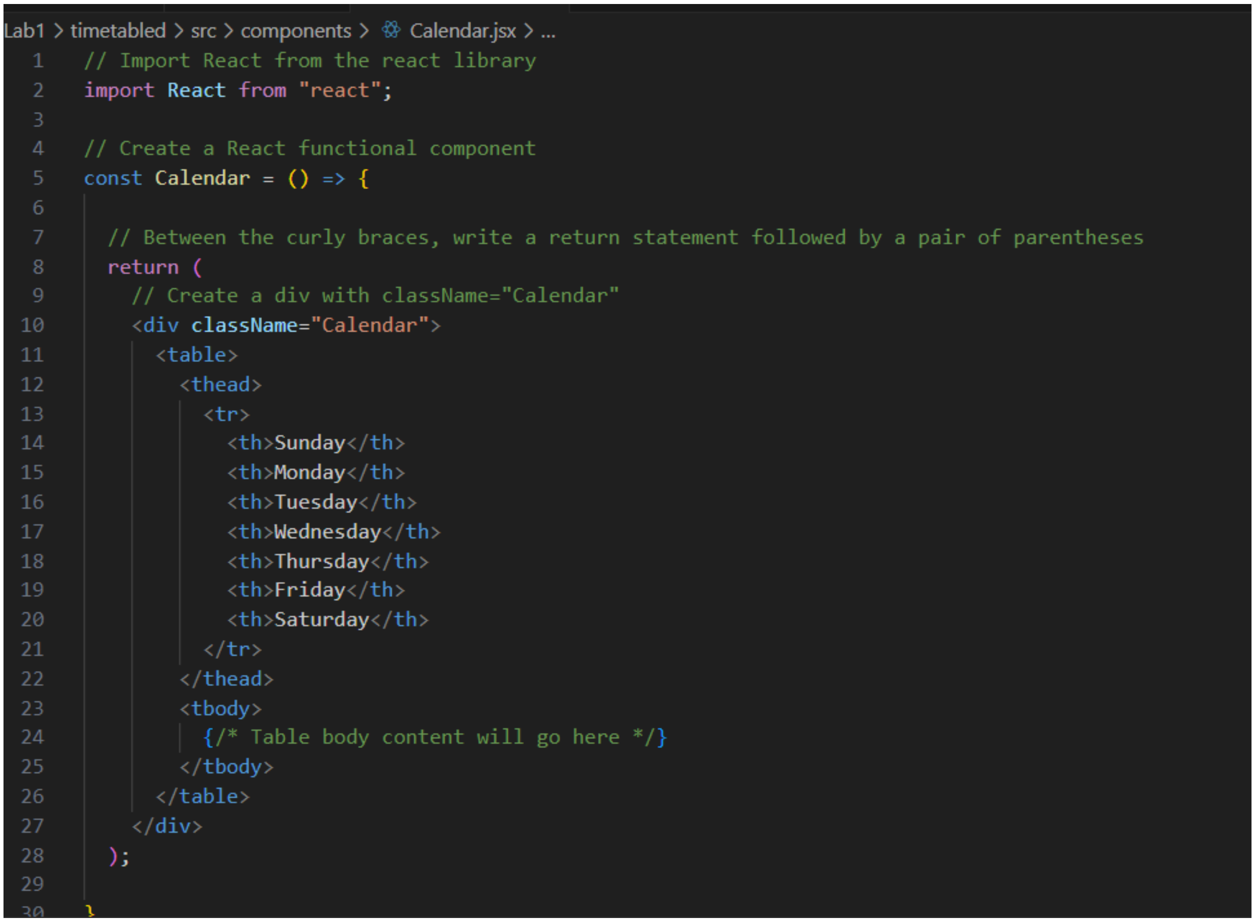
Welcome to Chicago, Cesar! Check out this calendar to get to know the city and see all the sights during your stay.

Testing calendar component

Step 3: Create a Grid in the `Calendar` Component

In `Calendar.jsx`:

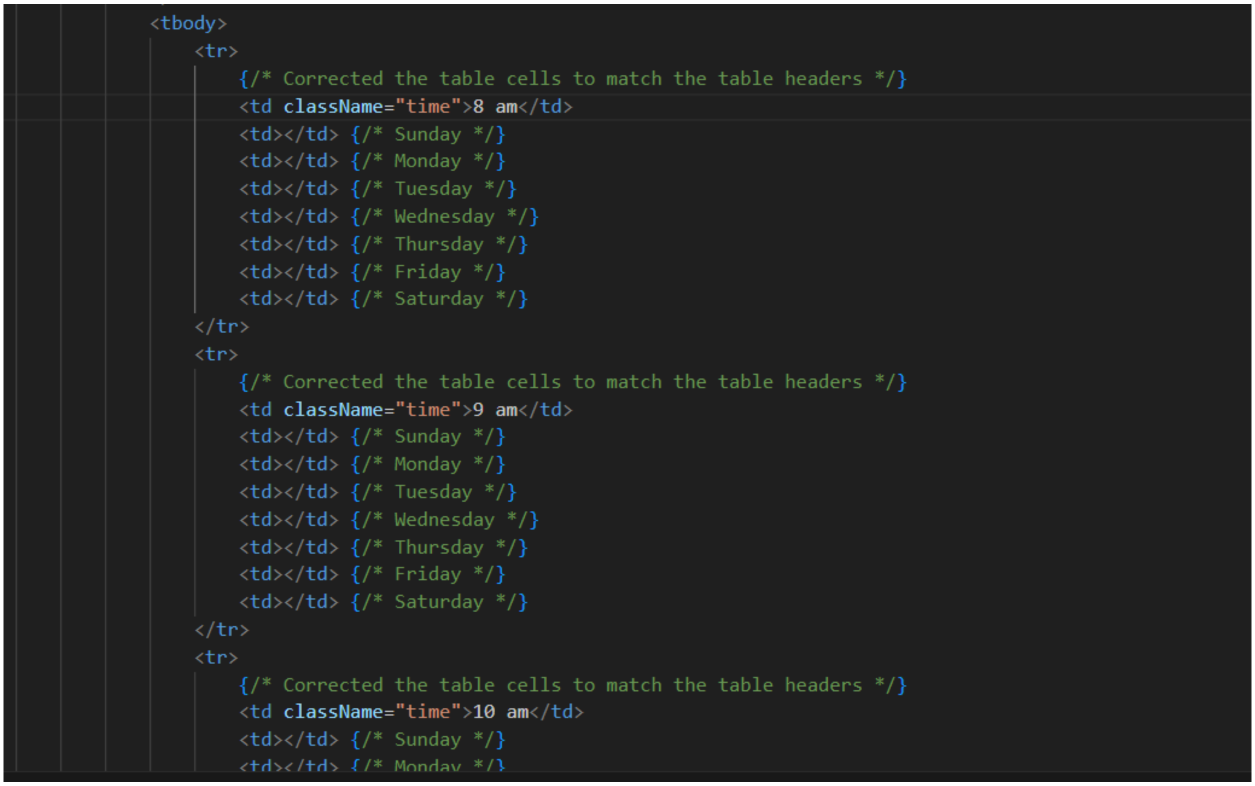
- Remove the test message `Testing the calendar`, but leave the `div`.
- Between the parentheses `()`, create a table with a table header.
- In the table header, create a table row containing the following 8 table headings: `""`, `"Sunday"`, `"Monday"`, `"Tuesday"`, `"Wednesday"`, `"Thursday"`, `"Friday"`, `"Saturday"`.



Lab1 > timetabled > src > components > Calendar.jsx > ...

```
// Import React from the react library
import React from "react";

// Create a React functional component
const Calendar = () => {

  // Between the curly braces, write a return statement followed by a pair of parentheses
  return (
    // Create a div with className="Calendar"
    <div className="Calendar">
      <table>
        <thead>
          <tr>
            <th>Sunday</th>
            <th>Monday</th>
            <th>Tuesday</th>
            <th>Wednesday</th>
            <th>Thursday</th>
            <th>Friday</th>
            <th>Saturday</th>
          </tr>
        </thead>
        <tbody>
          {/* Table body content will go here */}
        </tbody>
      </table>
    </div>
  );

}
```

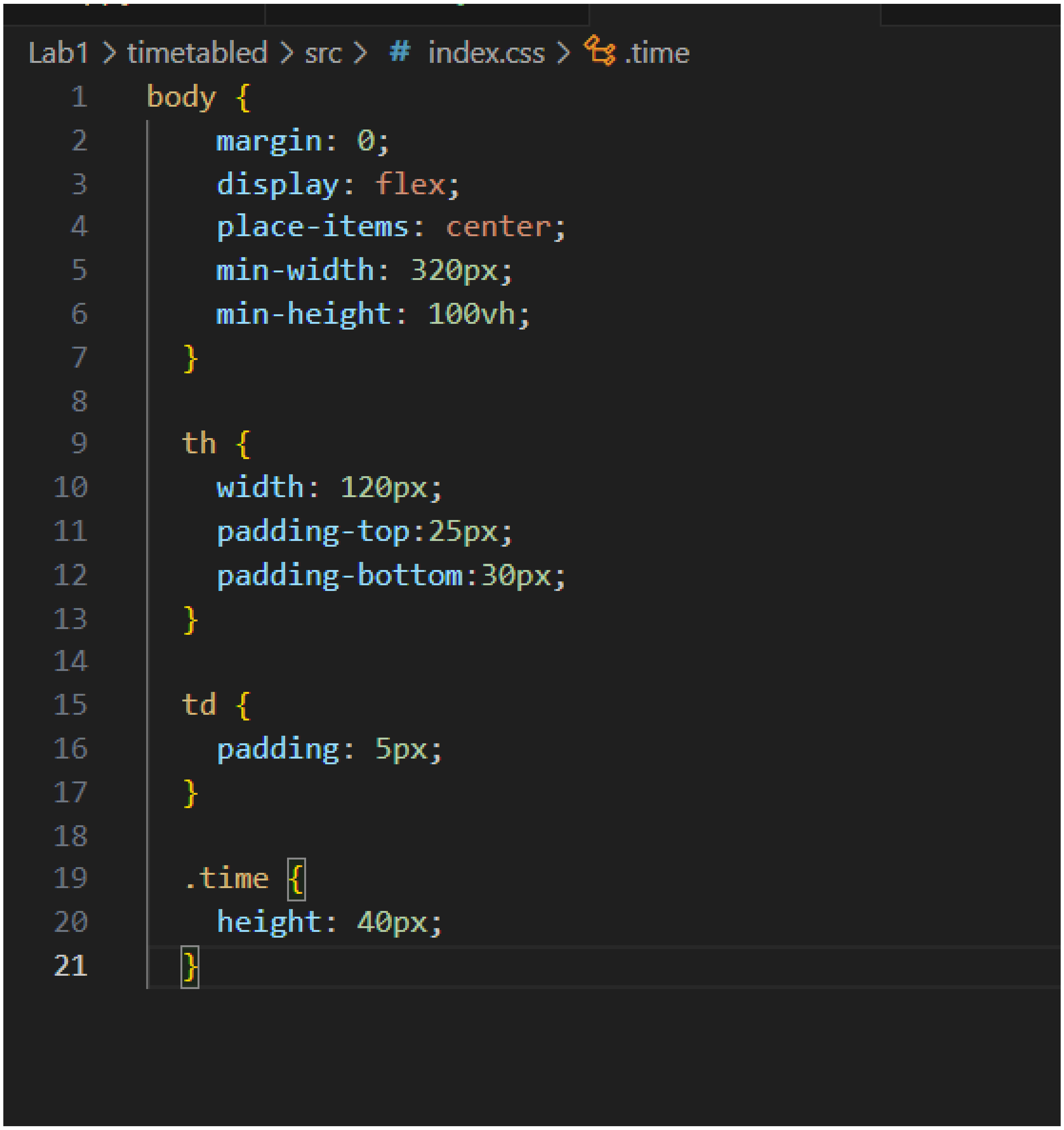
- After the table header, create a table body.
- Inside the table body, create 10 table rows, each with 8 table data cells.



```
            <tbody>
                <tr>
                    {/* Corrected the table cells to match the table headers */}
                    <td className="time">8 am</td>
                    <td></td> {/* Sunday */}
                    <td></td> {/* Monday */}
                    <td></td> {/* Tuesday */}
                    <td></td> {/* Wednesday */}
                    <td></td> {/* Thursday */}
                    <td></td> {/* Friday */}
                    <td></td> {/* Saturday */}
                </tr>
                <tr>
                    {/* Corrected the table cells to match the table headers */}
                    <td className="time">9 am</td>
                    <td></td> {/* Sunday */}
                    <td></td> {/* Monday */}
                    <td></td> {/* Tuesday */}
                    <td></td> {/* Wednesday */}
                    <td></td> {/* Thursday */}
                    <td></td> {/* Friday */}
                    <td></td> {/* Saturday */}
                </tr>
                <tr>
                    {/* Corrected the table cells to match the table headers */}
                    <td className="time">10 am</td>
                    <td></td> {/* Sunday */}
                    <td></td> {/* Monday */}
```

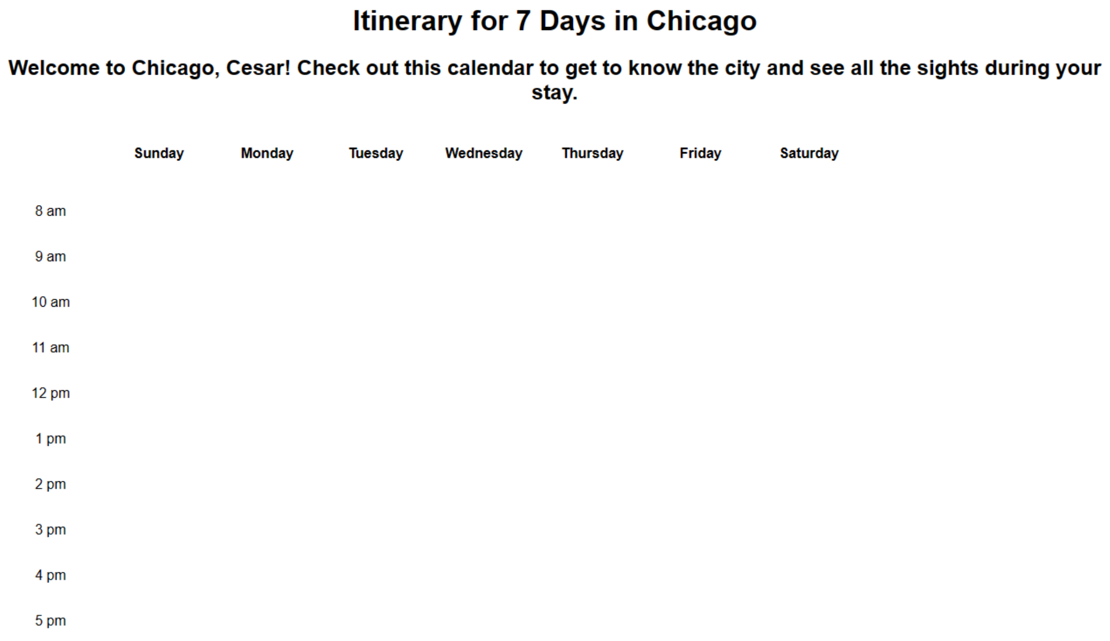
In `index.css`:

- Create more white space in the table:



Lab1 > timetabled > src > # index.css > .time

```css
body {
    margin: 0;
    display: flex;
    place-items: center;
    min-width: 320px;
    min-height: 100vh;
  }

  th {
    width: 120px;
    padding-top:25px;
    padding-bottom:30px;
  }

  td {
    padding: 5px;
  }

  .time {
    height: 40px;
  }
```



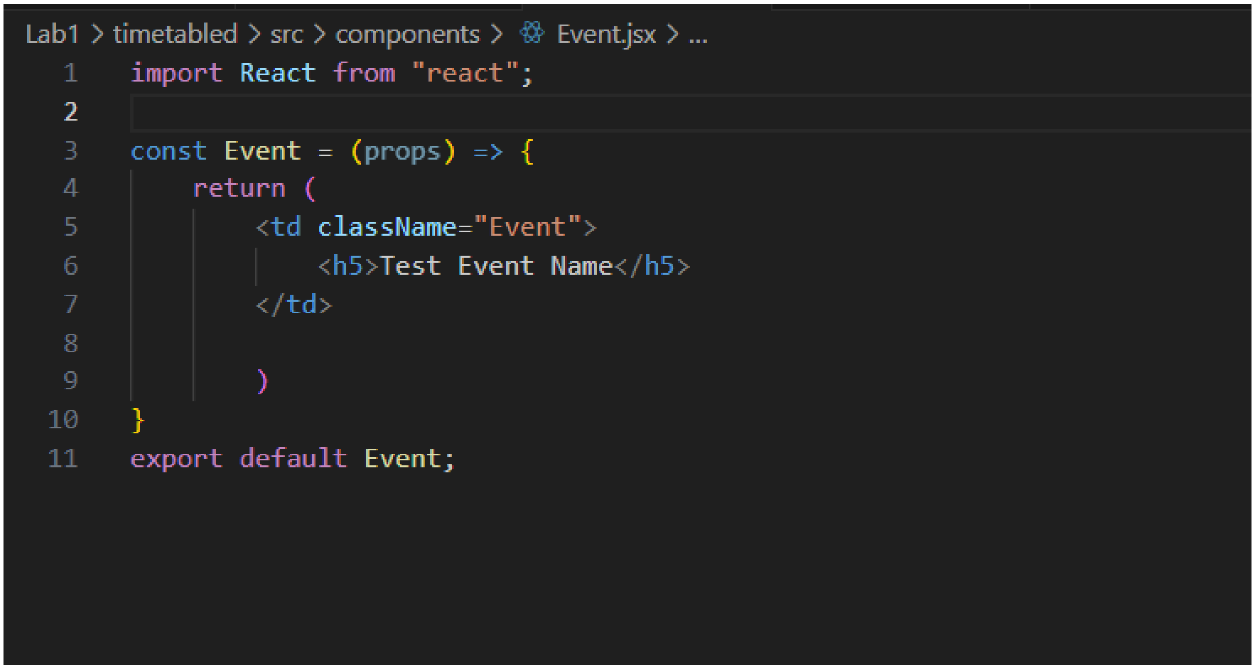
Itinerary for 7 Days in Chicago

Welcome to Chicago, Cesar! Check out this calendar to get to know the city and see all the sights during your stay.

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
8 am							
9 am							
10 am							
11 am							
12 pm							
1 pm							
2 pm							
3 pm							
4 pm							
5 pm							

Step 4: Create the `Event` Component

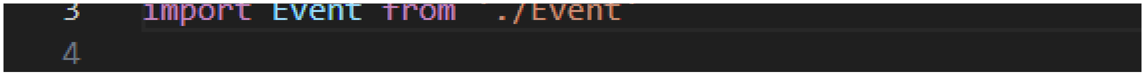
- In the `components` directory, create a file called `Event.jsx`.
- Import `React` from the `react` library
- Create a React functional component using arrow function notation called `Event`. It should have a single parameter called `props`.
- Between the curly braces, write a return statement followed by a pair of parentheses.
- Create a `td` element where `className="Event"`.
- Inside the `td` element, create an `h5` element with the text `Test Event Name`.



```
import React from "react";

const Event = (props) => {
    return (
        <td className="Event">
            <h5>Test Event Name</h5>
        </td>

        )
}
export default Event;
```

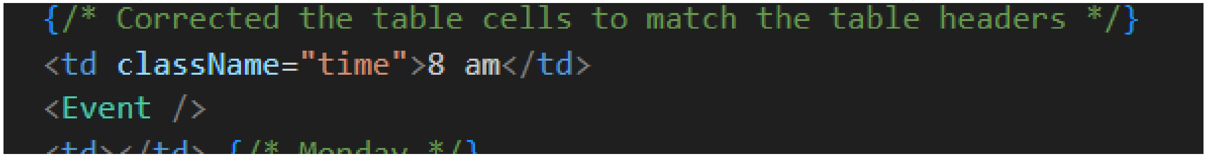
In `Calendar.jsx`:

- Import the `Event` component.



```
import Event from './Event'

```

- Replace one of the `td` elements in the table with an `Event` component.



```
{/* Corrected the table cells to match the table headers */}
<td className="time">8 am</td>
<Event />
<td></td> {/* Monday */}
```

Step 5: Pass props to the `Event` Component

In this step, you will pass props to the `Event` component with the location and the color.

In the `Event` component you added to the `Calendar`:

- Create an `event` prop and set it to `'Fancy Dinner 🎩'`

```
<Event event='Fancy Dinner 🎩' />
```

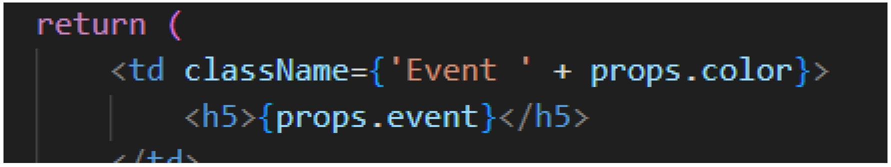
- Create `color` prop and set it to `'green'`

```
<Event event='Fancy Dinner 🎩' color ='green'/>
```

```
<Event event='Fancy Dinner 🎩' color='green' />
```

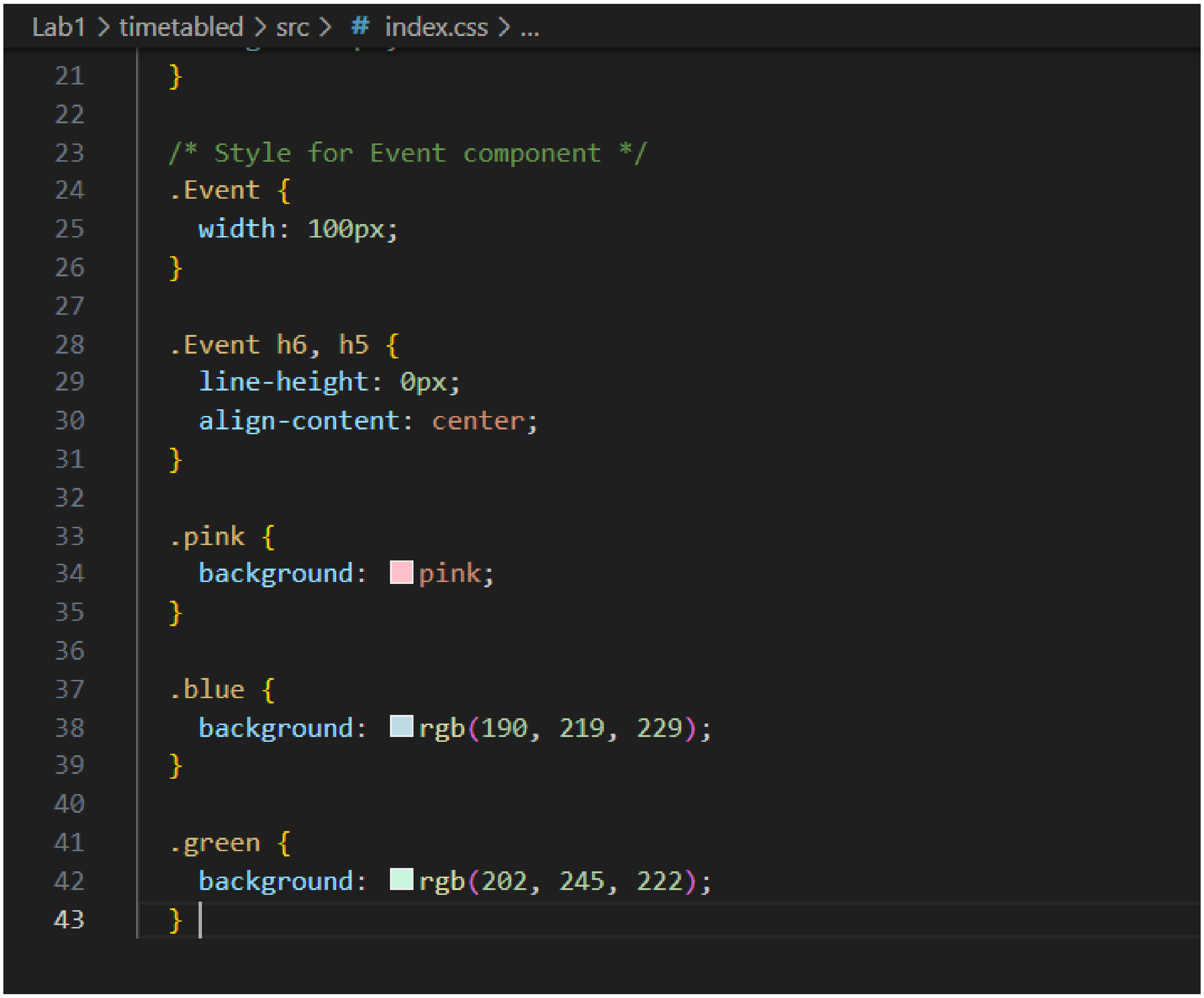
In `Event.jsx`:

- Replace the text within the `h5` element with `{props.event}`.

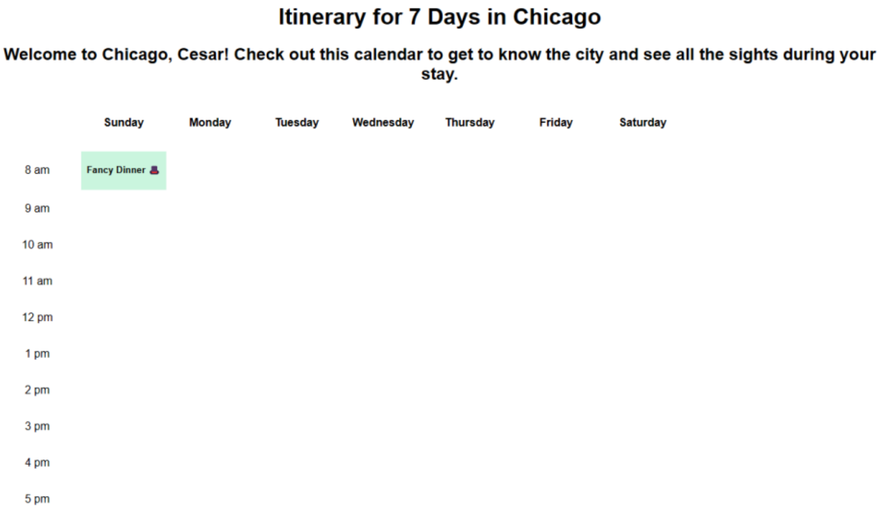


```
return (
    <td className={'Event ' + props.color}>
        <h5>{props.event}</h5>
    </td>
```

- In `index.css`:
- Create rules to style the `Event` components and set the background for the different colors that will be passed as props.



```css
}

/* Style for Event component */
.Event {
  width: 100px;
}

.Event h6, h5 {
  line-height: 0px;
  align-content: center;
}

.pink {
  background: pink;
}

.blue {
  background: rgb(190, 219, 229);
}

.green {
  background: rgb(202, 245, 222);
}
```



Itinerary for 7 Days in Chicago

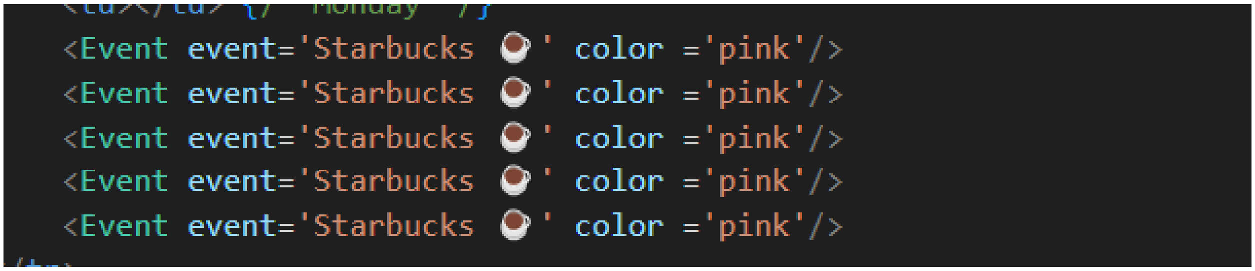
Welcome to Chicago, Cesar! Check out this calendar to get to know the city and see all the sights during your stay.

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
8 am	Fancy Dinner 🎩						
9 am							
10 am							
11 am							
12 pm							
1 pm							
2 pm							
3 pm							
4 pm							
5 pm							

Step 6: Add More Events

In this step, you will add more events to the calendar.

- To add an event to the calendar, replace one of the `td` elements in the table body with an `Event` component. Be sure to pass a value for `event` and a `color`. Currently, we only have CSS rules for `green`, `blue`, and `pink`



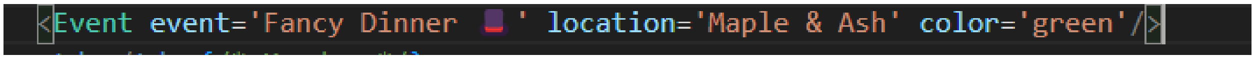
```
<td></td> {/* Monday */}
<Event event='Starbucks ☕' color ='pink'/>
<Event event='Starbucks ☕' color ='pink'/>
<Event event='Starbucks ☕' color ='pink'/>
<Event event='Starbucks ☕' color ='pink'/>
<Event event='Starbucks ☕' color ='pink'/>
```

Step 7: Add the Location to Event

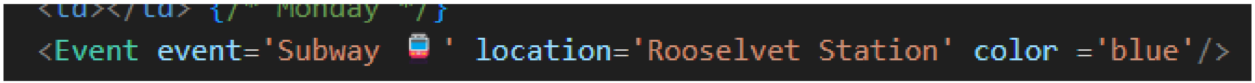
In this step, you will add pass another prop to the Event component with the location of the event.

In the `Event` component you added to the Calendar:

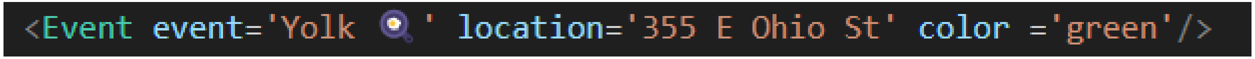
- Create a `location` prop and set it to string with the location name or address.



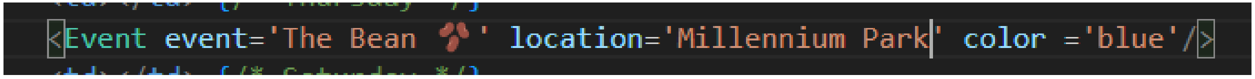
```
<Event event='Fancy Dinner 🎩' location='Maple & Ash' color='green'/>
```



```
<td></td> {/* Monday */}
<Event event='Subway 🚈' location='Rooselvet Station' color ='blue'/>
```



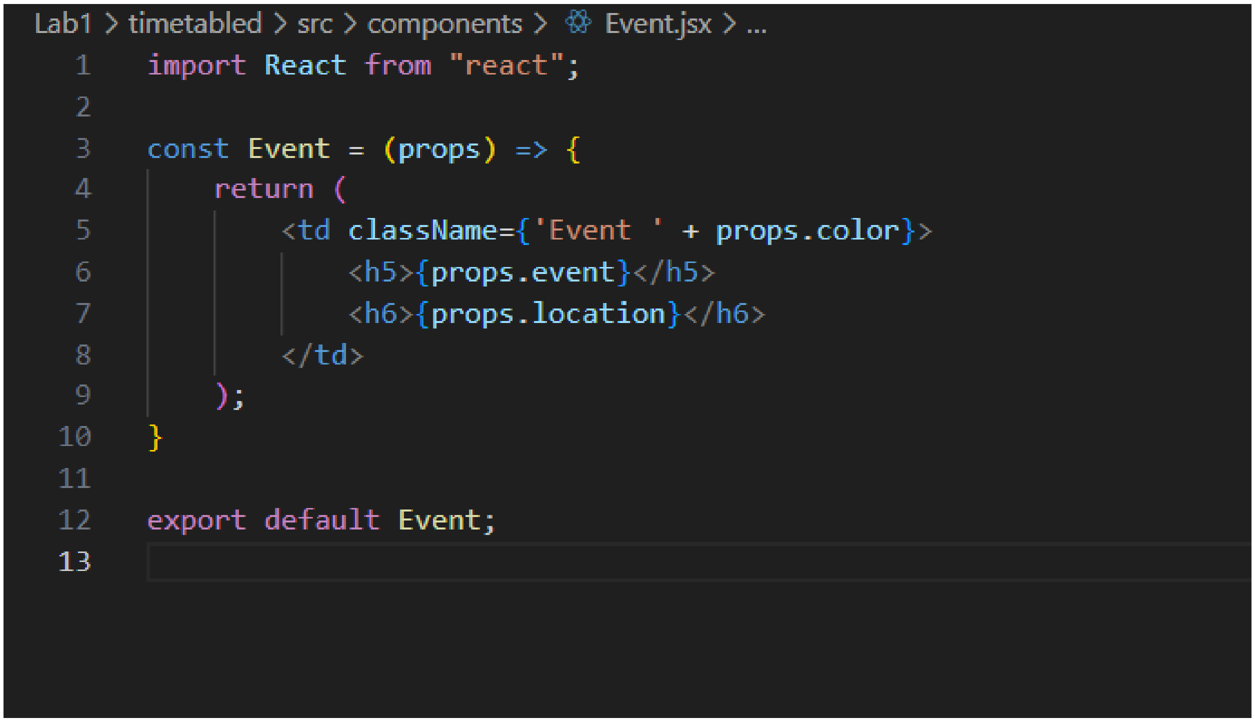
```
<Event event='Yolk 🍳' location='355 E Ohio St' color ='green'/>
```



```
<Event event='The Bean 🫘' location='Millennium Park' color ='blue'/>
```

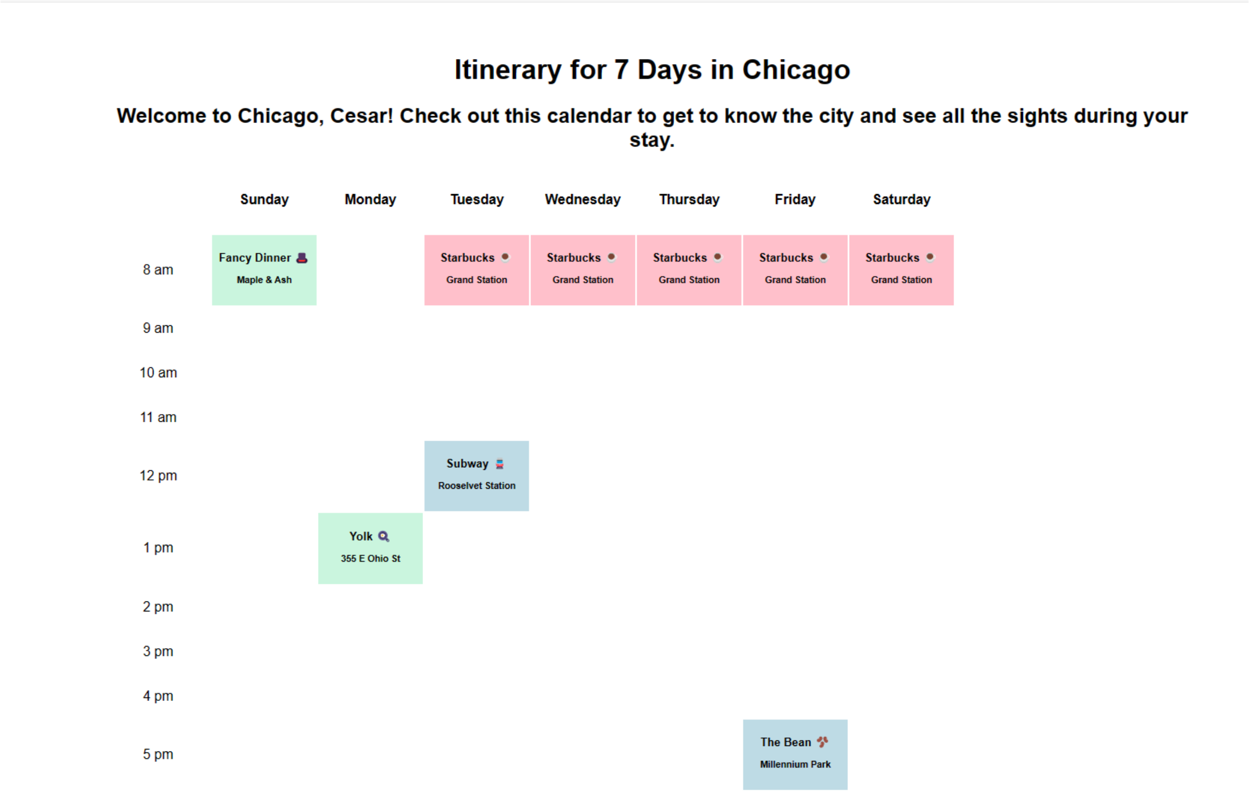
In `Event.jsx`:

- Add the `{props.location}` in an `h6` element.



Lab1 > timetabled > src > components > Event.jsx > ...

```jsx
import React from "react";

const Event = (props) => {
    return (
        <td className={'Event ' + props.color}>
            <h5>{props.event}</h5>
            <h6>{props.location}</h6>
        </td>
    );
}

export default Event;

```



Itinerary for 7 Days in Chicago

Welcome to Chicago, Cesar! Check out this calendar to get to know the city and see all the sights during your stay.

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
8 am	Fancy Dinner 🎩 Maple & Ash		Starbucks ☕ Grand Station	Starbucks ☕ Grand Station	Starbucks ☕ Grand Station	Starbucks ☕ Grand Station	Starbucks ☕ Grand Station
9 am							
10 am							
11 am							
12 pm			Subway 🚇 Roosevelt Station				
1 pm		Yolk 🍳 355 E Ohio St					
2 pm							
3 pm							
4 pm							
5 pm						The Bean 🫘 Millennium Park	