



Description Submission History

Step 3/3

Your assignment has been checked by a code reviewer. Some mistakes have been found.

Update the code and resubmit it



GitHub is set up



Assignment has been submitted



Some mistakes have been found

↓Expand to see the whole summary↓

Hello, Sabrina!

Great job on submitting your project 10! This is your first project related to React ! I hope, it was interesting for you!

Parts I really liked in your project:

- Good job that you `catch` possible errors at the end of server requests
- Good job that you've used `_id` for the `key` rather than the `index` .
- Good job that the submit button is only in `ModalWithForm` , and its text is controlled

There are few things that need to be corrected in your project. They're mostly minor issues that are easy to fix.

Needs correcting:

- All comments in the code marked `Needs correcting` and the comments below:
- Please, reuse the `handleCloseModal` function for any closing, not to duplicate the same code
- `<>` empty tags are not needed if you have a common tag that wraps the whole markup of the component
- The `text` near the `input` should be in a `label` tag, according to the semantics

- In Footer you need to use the footer **tag**, according to the semantics

There are some comments in your project that can help you make it even better. Please check the Could be improved comments.

A reminder: It's necessary to fix all Needs correcting comments to get the project accepted. And Could be improved comments are optional, they will be up to you if you want to implement them.

Your hard work on this project is truly commendable. While there are still some adjustments needed. If you need any assistance with the feedback or understanding certain concepts, our tutors are here to help. Keep moving forward and maintain your dedication – you are doing an amazing job! ✨

reviewer

Collapse

How was the review?



Your project with feedback

Expand all



Seyonce12-se_project_react-b902cce

12 | 1 | 5



src

12 | 1 | 5



components

12 | 1 | 5



AddClothes

2



AddClothes.css

No comments



AddClothes.jsx

2

```
1 import './AddClothes.css';
2
3 const AddClothes = () => {
4   return (
5     <>
6       <div className="modal__input">
```

div

The **text** near the `input` should be in a `label` tag, according to the semantics

```
<label className="modal__input">
  Name
  <input
    id="modal__input-name"
    className="modal__input-form"
    name="name"
    type="text"
    placeholder="Name"
    minLength="1"
    maxLength="30"
    required
  />
</label>
```

NEEDS CORRECTING

Mark as completed

reviewer

```
7       Name
8       <input
9         id="modal__input-name"
10        className="modal__input-form"
11        name="name"
12        type="text"
13        placeholder="Name"
14        minLength="1"
15        maxLength="30"
16        required
17      />
18    </div>
19    <span className=""></span>
20    {/* Image URL Input */}
21    <div className="modal__input">
```

div

The **text** near the `input` should be in a `label` tag, according to the semantics

```
<label className="modal__input">
```

NEEDS CORRECTING

Mark as completed

reviewer

```
22      Image
23      <input
24          id="modal__input-url"
25          className="modal__input-form"
26          name="link"
27          type="url"
28          placeholder="Image URL"
29          required
30      />
31  </div>
32  <span className=""></span>
33  {/* Select weather and radio inputs */}
34  <p className="modal__radio-title">Select weather type:</p>
35  <div className="modal__radio-container">
36      <div className="radio__btn-container">
37          <input
38              className="radio__btns"
39              type="radio"
40              id="hot"
41              name="weather"
42              value="hot"
43          />
44          <label
45              htmlFor="hot"
46              className="radio__btn-label"
47          >
48              Hot
49          </label>
50      </div>
51      <div className="radio__btn-container">
52          <input
53              className="radio__btns"
54              type="radio"
55              id="warm"
56              name="weather"
57              value="warm"
58          />
59          <label
```

```

60         htmlFor="warm"
61         className="radio__btn-label"
62     >
63         Warm
64     </label>
65 </div>
66 <div className="radio__btn-container">
67     <input
68         className="radio__btns"
69         type="radio"
70         id="cold"
71         name="weather"
72         value="cold"
73     />
74     <label
75         htmlFor="cold"
76         className="radio__btn-label"
77     >
78         Cold
79     </label>
80 </div>
81 </div>
82 </>
83 );
84 };
85
86 export default AddClothes;

```



App

7 | 1



App.css

No comments



App.jsx

7 | 1

```

1  import { useEffect, useState } from 'react';
2  import './App.css';
3  import Header from '../Header/Header';
4  import Main from '../Main/Main';
5  import Footer from '../Footer/Footer';
6  import ModalWithForm from '../ModalWithForm/ModalWithForm';
7  import ItemModal from '../ItemModal/ItemModal';
8  import AddClothes from '../AddClothes/AddClothes';
9  import { getWeather } from '../utils/weatherApi';

```

```

10 import { defaultClothingItems } from '../utils/constants';
11
12 function App() {
13     const [modalOpened, setModalOpened] = useState('');
14     const [selectedCard, setSelectedCard] = useState({});
15     const [location, setLocation] = useState('');
16     const [temp, setTemp] = useState(0);
17     const [weatherType, setWeatherType] = useState('');
18     const [sunrise, setSunrise] = useState();
19     const [sunset, setSunset] = useState();
20     const [clothingItems, setClothingItems] = useState(defaultClothingItems || [])
21     const dateNow = Date.now() * 0.001;
22
23     useEffect(() => {
24         getWeather()
25         .then((data) => {
26             const weatherTemperature = data.main.temp;
27             setTemp(weatherTemperature);
28             const locationName = data.name;
29             setLocation(locationName);
30             const weatherType = data.weather[0].main;
31             setWeatherType(weatherType.toLowerCase());
32             const sunriseData = data.sys.sunrise;
33             setSunrise(sunriseData);
34             const sunsetData = data.sys.sunset;
35             setSunset(sunsetData);
36         })
37         .catch((err) => {
38             console.log(err);

```

Good job that you **catch** possible errors at the end of server requests. You can simplify it with `.catch(console.error);` because `console.error` is meant to log errors

GOOD

Mark as completed

reviewer

```

39     });
40 }, []);
41
42 const timeOfDay = () => {
43     if (dateNow >= sunrise && dateNow < sunset) {
44         return true;
45     } else {
46         return false;
47     }
48 };
49
50 /* Modal functions */

```

51 `useEffect(() => {`

You should add an `Escape` listener like this to be able to remove it **correctly** when modals are closed:

```
useEffect(() => {
```

```
  if (!modalOpened) return; // stop the effect not to add the listener if
```



NEEDS CORRECTING

Mark as completed

reviewer

```
52      const handleEscClose = (evt) => {  
53          if (evt.key === 'Escape') {  
54              setModalOpened('');
```

```
setModalOpened('');
```

Please, reuse the `handleCloseModal` function for any closing, not to duplicate the same code

NEEDS CORRECTING

Mark as completed

reviewer

```
55          }  
56      };  
57      window.addEventListener('keydown', handleEscClose);  
58      return () => {  
59          window.removeEventListener('keydown', handleEscClose);  
60      };  
61      }, []);
```

You should add an `Escape` listener like this to be able to remove it **correctly** when modals are closed:

```
    }, [modalOpened]); // watch modalOpened here
```

NEEDS CORRECTING

Mark as completed

reviewer

```
62  
63      const handleSubmit = (e) => {  
64          e.preventDefault();  
65
```

```

66 // Get form values
67 const form = e.target;
68 const name = form.elements.name.value;
69 const link = form.elements.link.value;
70 const weather = form.elements.weather.value;
71
72 // Create new clothing item
73 const newItem = {
74   _id: Date.now().toString(), // Simple unique ID
75   name,
76   weather,
77   link
78 };
79
80 // Add item to the state
81 setClothingItems([newItem, ...clothingItems]);
82
83 // Close the modal
84 setModalOpened('');

```

setModalOpened('')

Please, reuse the `handleCloseModal` function for any closing, not to duplicate the same code

NEEDS CORRECTING

Mark as completed

reviewer

```

85 };
86
87 const handleCloseModal = () => {
88   setModalOpened('');
89 };
90
91 const handleOpenModal = () => {
92   setModalOpened('new-clothes-modal');
93 };
94
95 /* Item Card Image Modal functions */
96 const handleSelectedCard = (card) => {
97   setModalOpened('open');

```

open

There are 2 modals in the project. You need to **name** variables and functions the way all could understand what they are: `item-modal`

```
setModalOpened('item-modal');
```


NEEDS CORRECTING

Mark as completed

reviewer

```
98     setSelectedCard(card);
99   };
100
101   return (
102     <>
```

<> empty tags are not needed if you have a common tag that wraps the whole markup of the component

NEEDS CORRECTING

Mark as completed

reviewer

```
103     <div className="App">
104       <Header locationData={location} openAddClothesModal={handleOpenModal} />
105       <Main
106         weatherTemp={temp}
107         weatherType={weatherType}
108         onSelectCard={handleSelectedCard}
109         timeOfDay={timeOfDay()}
110         clothingItems={clothingItems}
111       />
112       <Footer />
113       {modalOpened === 'new-clothes-modal' && (
114         <ModalWithForm
115           title="New clothes"
116           name="clothes"
117           buttonText="Add clothes"
118           onClose={handleCloseModal}
119           handleSubmitForm={handleSubmit}
120         >
121           <AddClothes />
122         </ModalWithForm>
123       )}
124       {modalOpened === 'open' && (
```

open

There are 2 modals in the project. You need to name variables and functions the way all could understand what they are: item-modal

NEEDS CORRECTING

Mark as completed

reviewer

```
125       <ItemModal onClose={handleCloseModal} selectedCard={selectedCard} />
126     )}
127   </div>
```

```
128     </>
129   );
130 }
131
132 export default App;
```



Footer

1 | 1



Footer.css

No comments



Footer.jsx

1 | 1

```
1   import './Footer.css';
2
3   const Footer = () => {
4     return (
5       <section className="footer">
```

In Footer you need to use the footer tag, according to the semantics

NEEDS CORRECTING

Mark as completed

reviewer

```
6       <p className="footer__name">Developed by: Sabrina Florence</p>
7       <p className="footer__year">{new Date().getFullYear()}</p>
```

Nice year getting

GOOD

Mark as completed

reviewer

```
8       </section>
9     );
10  };
11
12  export default Footer;
```



Header

No comments



ItemCard

1



ItemCard.css

No comments



ItemCard.jsx

1

```
1   import './ItemCard.css';
2
3   const ItemCard = ({ data, onSelectCard }) => {
4     return (
5       <ul className="clothing__cards-wrapper">
```

ul can't be used for a card because it's not a list. It's one item of a section, according to the semantics

And li will not be needed anymore

You can use article here

```
return (
  <article className="clothing__cards-wrapper">
    <div
      className="clothing__card-items"
      id={`clothing__card-items_${data.name}`}
    >
      <p className="clothing-card-title">{data.name}</p>
      <img
        className="clothing-card-individual"
        id={`clothing-card-individual_${data.name}`}
        alt={data.name}
        src={data.link}
        onClick={() => onSelectCard(data)}
      />
    </div>
  </article>
);
```

NEEDS CORRECTING

Mark as completed

reviewer

```
6     <li
7       className="clothing__card-items"
8       id={`clothing__card-items_${data.name}`}
9     >
10      <p className="clothing-card-title">{data.name}</p>
```

```

11         <img
12             className="clothing-card-individual"
13             id={`clothing-card-individual_${data.name}`}
14             alt={data.name}
15             src={data.link}
16             onClick={() => onSelectCard(data)}
17         />
18     </li>
19 </ul>
20 );
21 };
22
23 export default ItemCard;

```



ItemModal

1



ItemModal.css

No comments



ItemModal.jsx

1

```

1  import './ItemModal.css';
2  import exitButton from '../images/modalexit.svg';
3
4  const ItemModal = ({ onClose, selectedCard }) => {
5      const handleModalClick = (e) => {
6          if (e.target === e.currentTarget) {

```

Nice overlay handling

GOOD

Mark as completed

reviewer

```

7      onClose(e);
8  }
9  };
10
11  return (
12      <section className="modal item__modal" onClick={handleModalClick}>
13          <div className="modal__container item__modal-container">
14              <button
15                  className="modal__exit"
16                  type="button"
17                  onClick={onClose}

```

```

18         >
19         <img src={exitButton} alt="Close button" />
20     </button>
21     <div className="item__modal">
22         <p className="item__modal-title">{selectedCard.name}</p>
23         <img
24             className="item__modal-image"
25             alt={selectedCard.name}
26             src={selectedCard.link}
27         />
28     </div>
29     <p className="item__modal-weather">Weather: {selectedCard.weather}</p>
30 </div>
31 </section>
32 );
33 };
34
35 export default ItemModal;

```



Main

1



Main.css

No comments



Main.jsx

1

```

1  import './Main.css';
2  import WeatherCard from '../WeatherCard/WeatherCard';
3  import ItemCard from '../ItemCard/ItemCard';
4
5  function Main({ weatherTemp, weatherType, onSelectCard, timeOfDay,
6  clothingItems }) {
7      const weatherFilter = () => {
8          if (weatherTemp >= 86) {
9              return 'hot';
10         } else if (weatherTemp >= 66 && weatherTemp <= 85) {
11             return 'warm';
12         } else if (weatherTemp <= 65) {
13             return 'cold';
14         }
15     };
16
17     const weatherTempFilter = weatherFilter();

```

```

17     const filterCards = clothingItems.filter((item) => {
18         return item.weather === weatherTempFilter;
19     });
20
21     return (
22         <main className="main">
23             <WeatherCard
24                 day={timeOfDay}
25                 type={weatherType}
26                 weatherTemp={weatherTemp}
27             />
28             <div className="main__title">
29                 Today is {Math.round(weatherTemp)}°F / You may want to wear:
30             </div>
31             <section className="clothing">
32                 {filterCards.map((data) => (
33                     <ItemCard key={data._id} data={data} onSelectCard={onSelectCard}

```

```

/>

```

Good job that you've used `_id` for the `key` rather than the `index`.

Here is why we shouldn't use `index` for that: [click here](#)

GOOD

Mark as completed

reviewer

```

34         }}}
35         </section>
36     </main>
37 );
38 }
39
40 export default Main;

```



ModalWithForm

1 | 1 | 1



ModalWithForm.css

No comments



ModalWithForm.jsx

1 | 1 | 1

```

1     import './ModalWithForm.css';
2     import exitButton from '../images/modalexit.svg';
3
4     const ModalWithForm = ({

```

```

5     children,
6     title,
7     name,
8     buttonText,
9     onClose,
10    handleSubmitForm,
11  }) => {
12    const handleModalClick = (e) => {
13      if (e.target === e.currentTarget) {
14        onClose(e);
15      }
16    };
17
18    return (
19      <section className={`modal modal_type_${name}`} onClick=
{handleModalClick}>
20        <div className="modal__container">
21          <button
22            className="modal__exit"
23            type="button"
24            onClick={onClose}
25          >
26            <img src={exitButton} alt="Exit button" />
27          </button>
28          <h2 className="modal__title">{title}</h2>
29          <form className="modal__form" id="" onSubmit={handleSubmitForm}>

```

`id=""` is not needed here. It causes jsx validation errors

NEEDS CORRECTING

Mark as completed

reviewer

```

30    <fieldset className="modal__fieldset">{children}</fieldset>
31    <span className="modal__error"></span>

```

You can use **self closing tags** if there are no **children** ``

COULD BE IMPROVED

Mark as completed

reviewer

```

32    <button
33      type="submit"
34      className="modal__submit"
35    >
36      {buttonText}

```

- Good job that the submit button is only in `ModalWithForm`, and its text is controlled

GOOD

Mark as completed

reviewer

```
37         </button>
38     </form>
39 </div>
40 </section>
41 );
42 };
43
44 export default ModalWithForm;
```



WeatherCard

No comments



utils

No comments



App.css

No comments



App.jsx

No comments



index.css

No comments



main.jsx

No comments



.gitignore

No comments



README.md

No comments



eslint.config.js

No comments



index.html

No comments



vite.config.js

No comments

roject?



↓ Download code