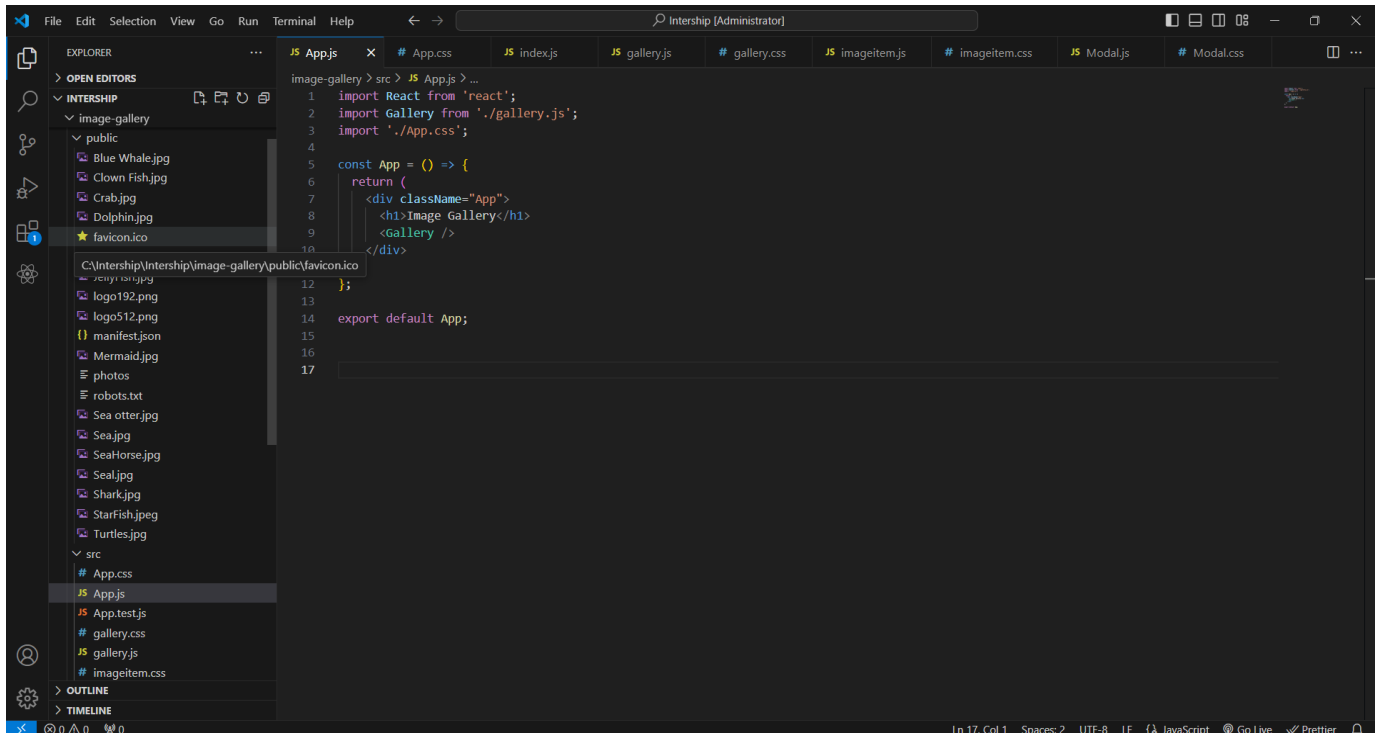


# Task-6

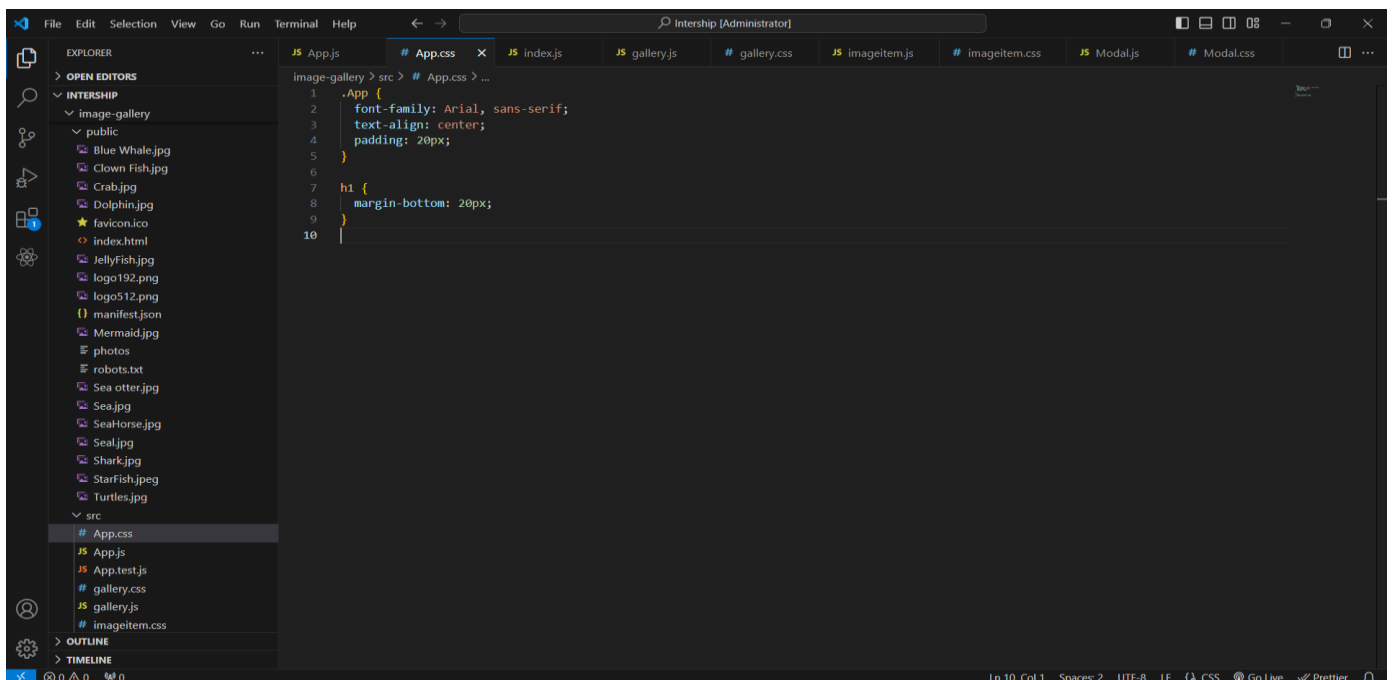
## App.js



The screenshot shows the VS Code editor with the 'App.js' file open. The Explorer sidebar on the left shows the project structure with 'image-gallery' as the selected folder. The main editor area displays the following code:

```
1 import React from 'react';
2 import Gallery from './gallery.js';
3 import './App.css';
4
5 const App = () => {
6   return (
7     <div className="App">
8       <h1>Image Gallery</h1>
9       <Gallery />
10    </div>
11  );
12 }
13
14 export default App;
```

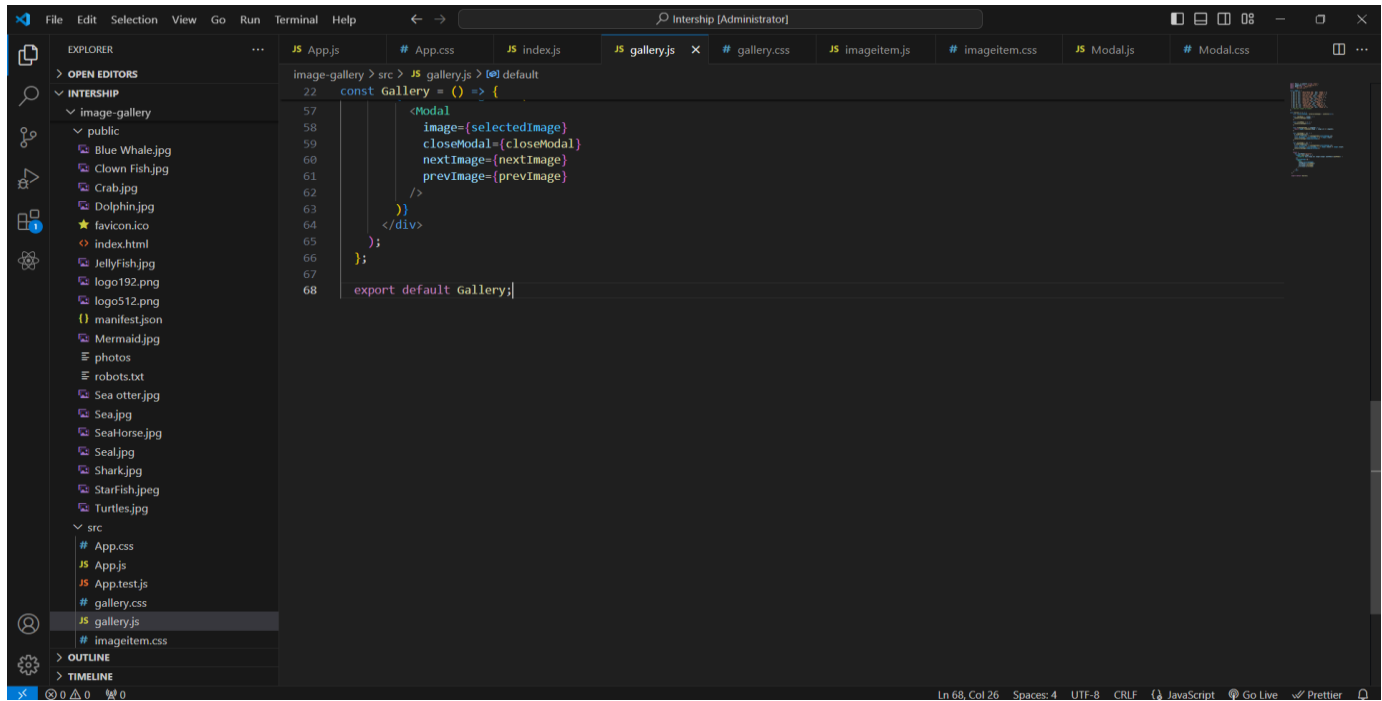
## App.css



The screenshot shows the VS Code editor with the 'App.css' file open. The Explorer sidebar on the left shows the project structure with 'image-gallery' as the selected folder. The main editor area displays the following code:

```
1 .App {
2   font-family: Arial, sans-serif;
3   text-align: center;
4   padding: 20px;
5 }
6
7 h1 {
8   margin-bottom: 20px;
9 }
10
```

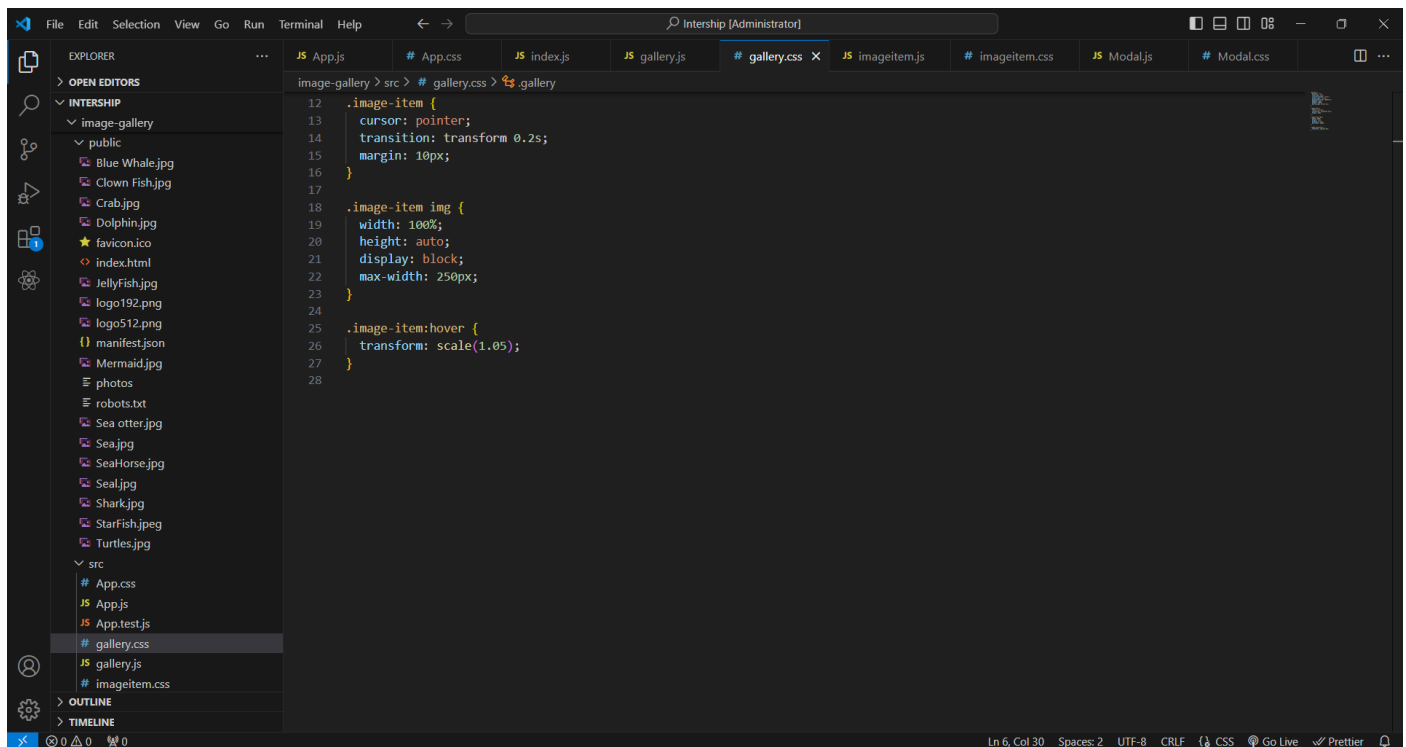
# Gallery.js



The screenshot shows the Visual Studio Code editor interface. The Explorer panel on the left displays the project structure for 'intership', including a 'public' folder with various image files and a 'src' folder containing 'App.js', 'App.css', 'App.test.js', 'gallery.css', 'gallery.js', and 'imageitem.css'. The main editor area shows the 'gallery.js' file with the following code:

```
image-gallery > src > JS gallery.js > default
22  const Gallery = () => {
57      <Modal
58          image={selectedImage}
59          closeModal={closeModal}
60          nextImage={nextImage}
61          prevImage={prevImage}
62      />
63  )}
64  </div>
65  };
66  };
67  };
68  export default Gallery;
```

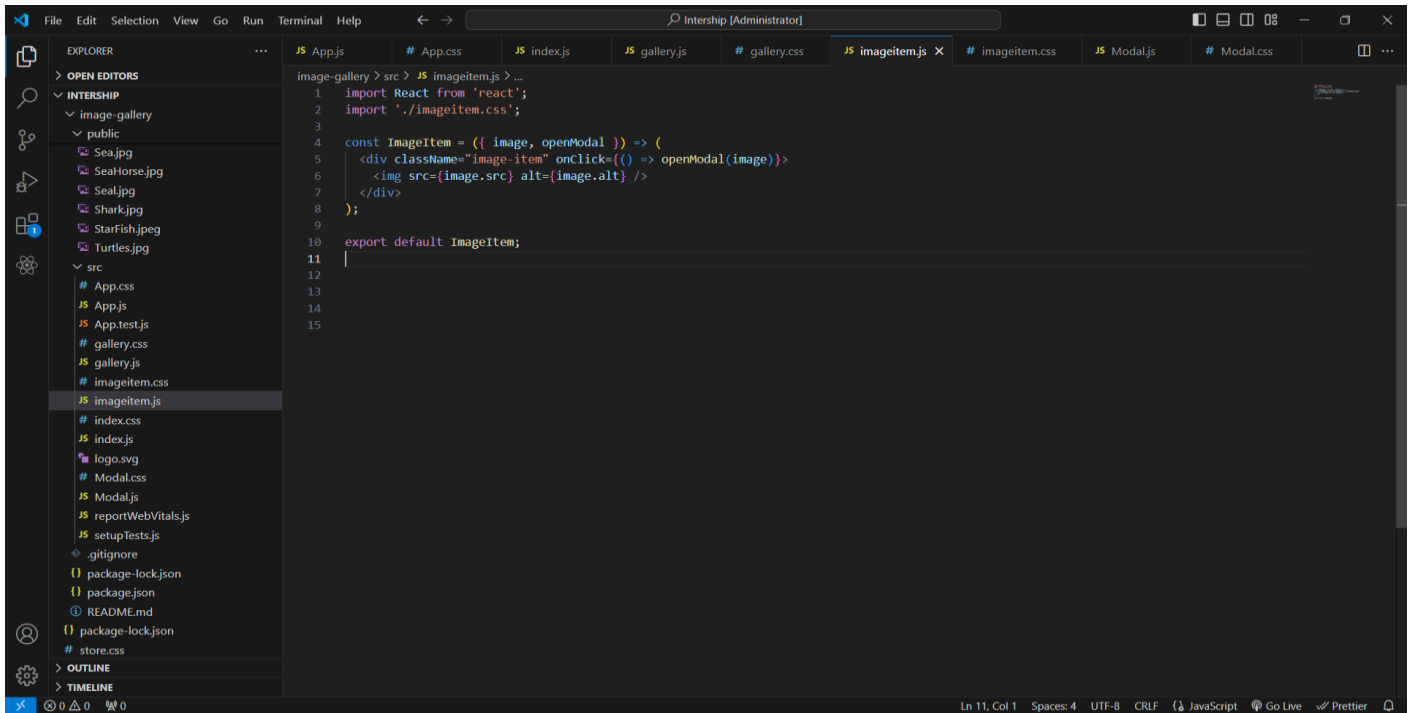
# Gallery.css



The screenshot shows the Visual Studio Code editor interface. The Explorer panel on the left displays the project structure for 'intership', including a 'public' folder with various image files and a 'src' folder containing 'App.js', 'App.css', 'App.test.js', 'gallery.css', 'gallery.js', and 'imageitem.css'. The main editor area shows the 'gallery.css' file with the following code:

```
image-gallery > src > # gallery.css > gallery
12  .image-item {
13      cursor: pointer;
14      transition: transform 0.2s;
15      margin: 10px;
16  }
17
18  .image-item img {
19      width: 100%;
20      height: auto;
21      display: block;
22      max-width: 250px;
23  }
24
25  .image-item:hover {
26      transform: scale(1.05);
27  }
28
```

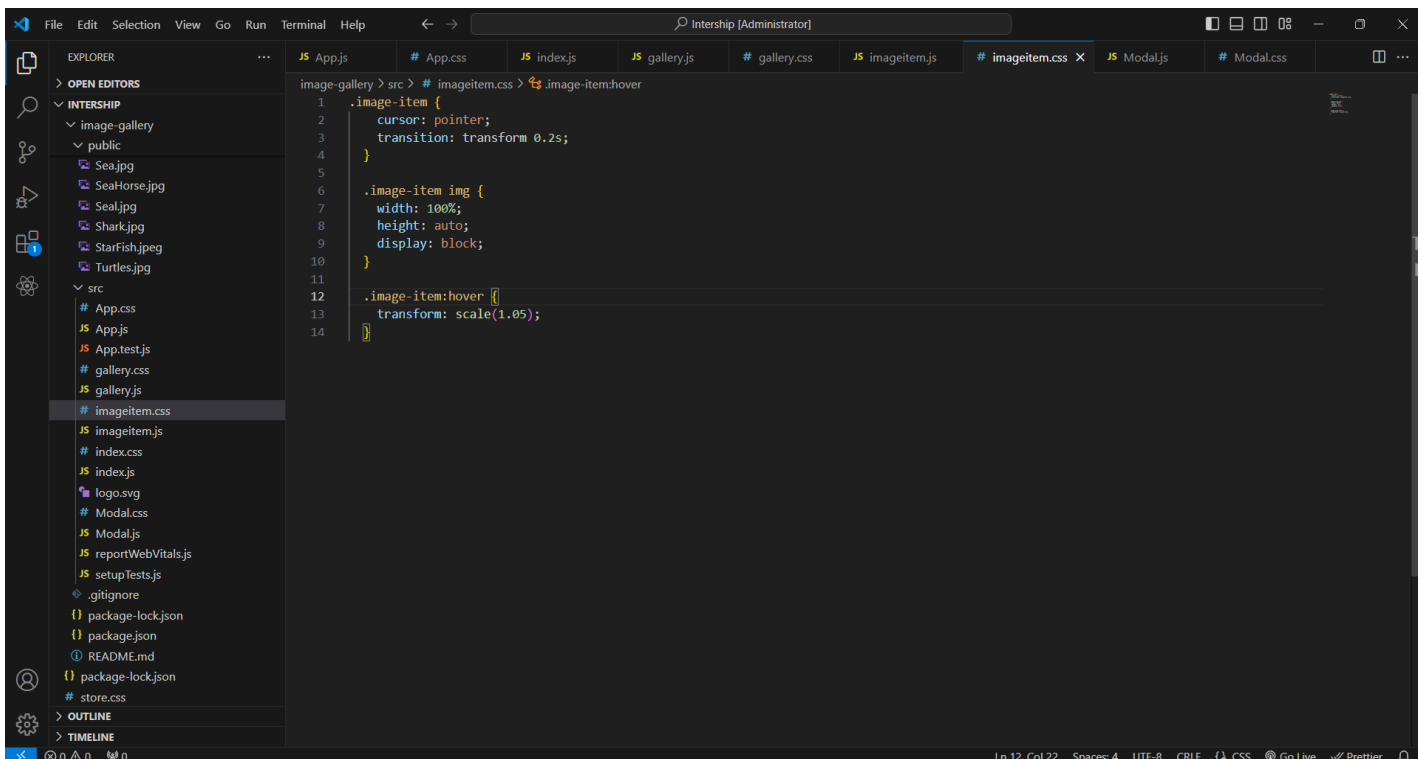
# imageitem.js



A screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left shows a project structure with folders like 'INTERSHIP' and 'image-gallery', and files like 'App.css', 'App.js', 'App.test.js', 'gallery.css', 'gallery.js', 'imageitem.css', 'imageitem.js', 'index.css', 'index.js', 'logo.svg', 'Modal.css', 'Modal.js', 'reportWebVitals.js', 'setupTests.js', '.gitignore', 'package-lock.json', 'package.json', 'README.md', 'store.css', 'Turtles.jpg', 'Seal.jpg', 'Seahorse.jpg', 'Shark.jpg', and 'StarFish.jpeg'. The main editor area displays the content of 'imageitem.js'. The code defines a functional component 'ImageItem' that takes 'image' and 'openModal' as props. It renders a 'div' with the class 'image-item' and an 'onClick' event that calls 'openModal(image)'. Inside the 'div' is an 'img' tag with 'src={image.src}' and 'alt={image.alt}'. The component is exported as the default export.

```
1 import React from 'react';
2 import './imageitem.css';
3
4 const ImageItem = ({ image, openModal }) => {
5   <div className="image-item" onClick={() => openModal(image)}>
6     <img src={image.src} alt={image.alt} />
7   </div>
8 };
9
10 export default ImageItem;
```

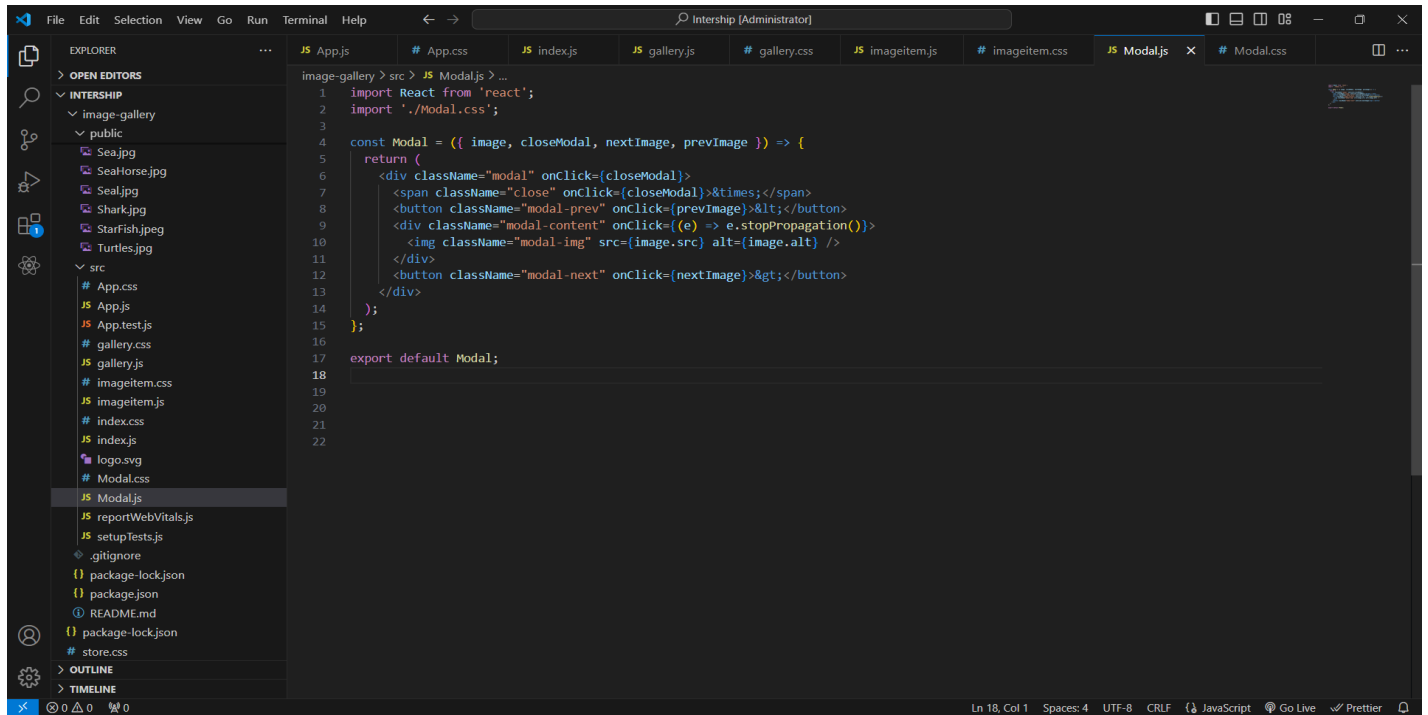
# imageitem.css



A screenshot of the Visual Studio Code editor interface, showing the 'imageitem.css' file. The Explorer sidebar is the same as in the previous screenshot. The main editor area displays the CSS rules for the 'image-item' class and its hover state. The rules include a pointer cursor, a 0.2s transform transition, a width of 100%, a height of auto, and a block display. The hover state is defined with a 1.05x scale transform.

```
1 .image-item {
2   cursor: pointer;
3   transition: transform 0.2s;
4 }
5
6 .image-item img {
7   width: 100%;
8   height: auto;
9   display: block;
10 }
11
12 .image-item:hover {
13   transform: scale(1.05);
14 }
```

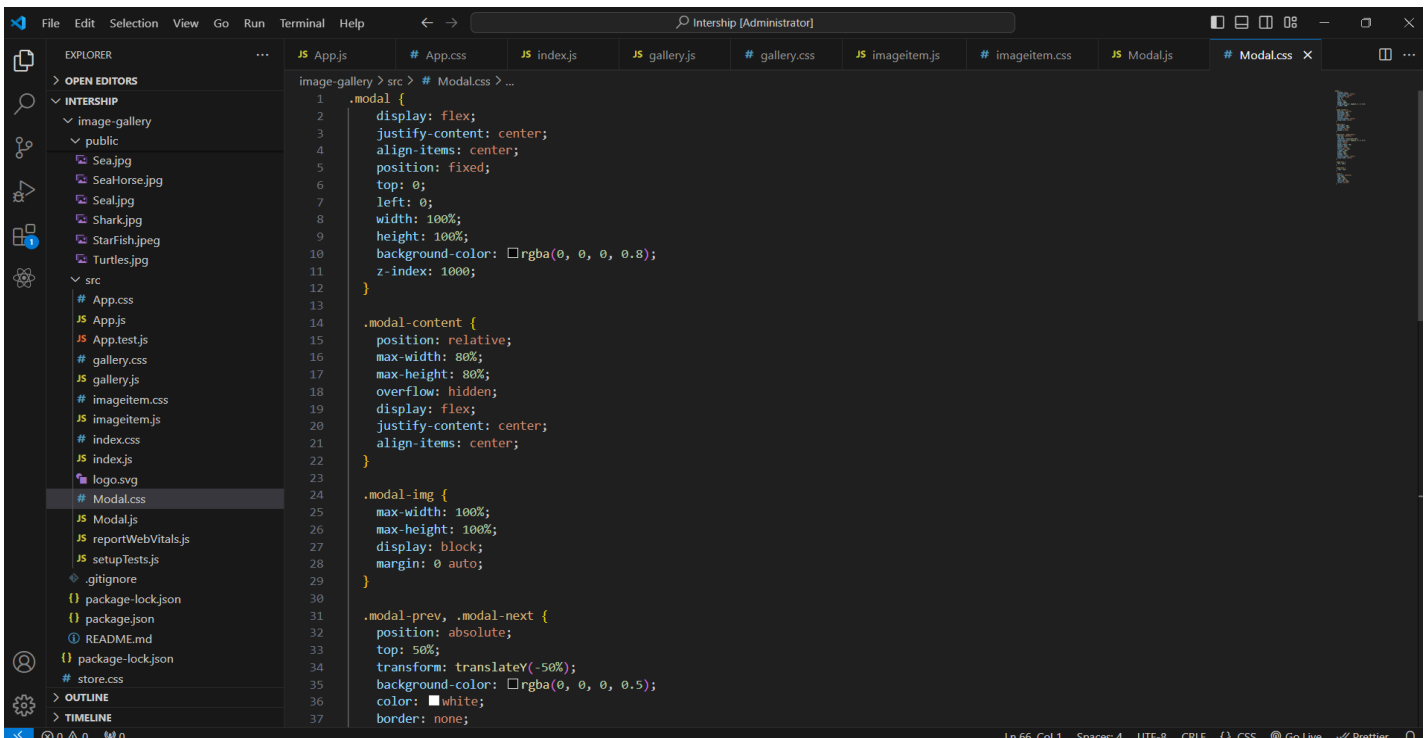
# Modal.js



The screenshot shows the VS Code editor with the file explorer on the left and the code editor in the center. The file explorer shows a project structure with a 'src' directory containing 'App.js', 'App.test.js', 'gallery.js', 'imageitem.css', 'imageitem.js', 'index.css', 'index.js', 'logo.svg', 'Modal.css', and 'Modal.js'. The code editor displays the content of 'Modal.js'.

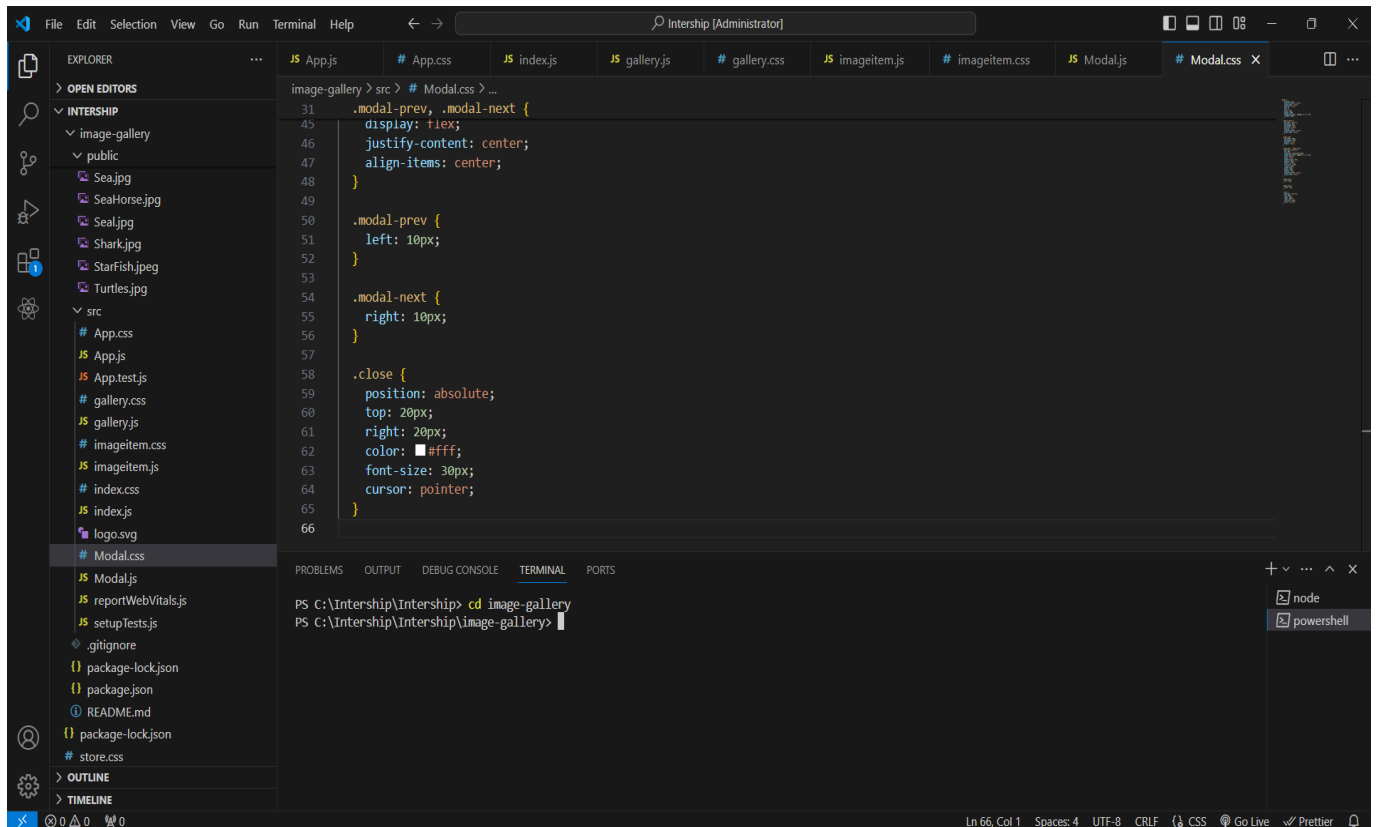
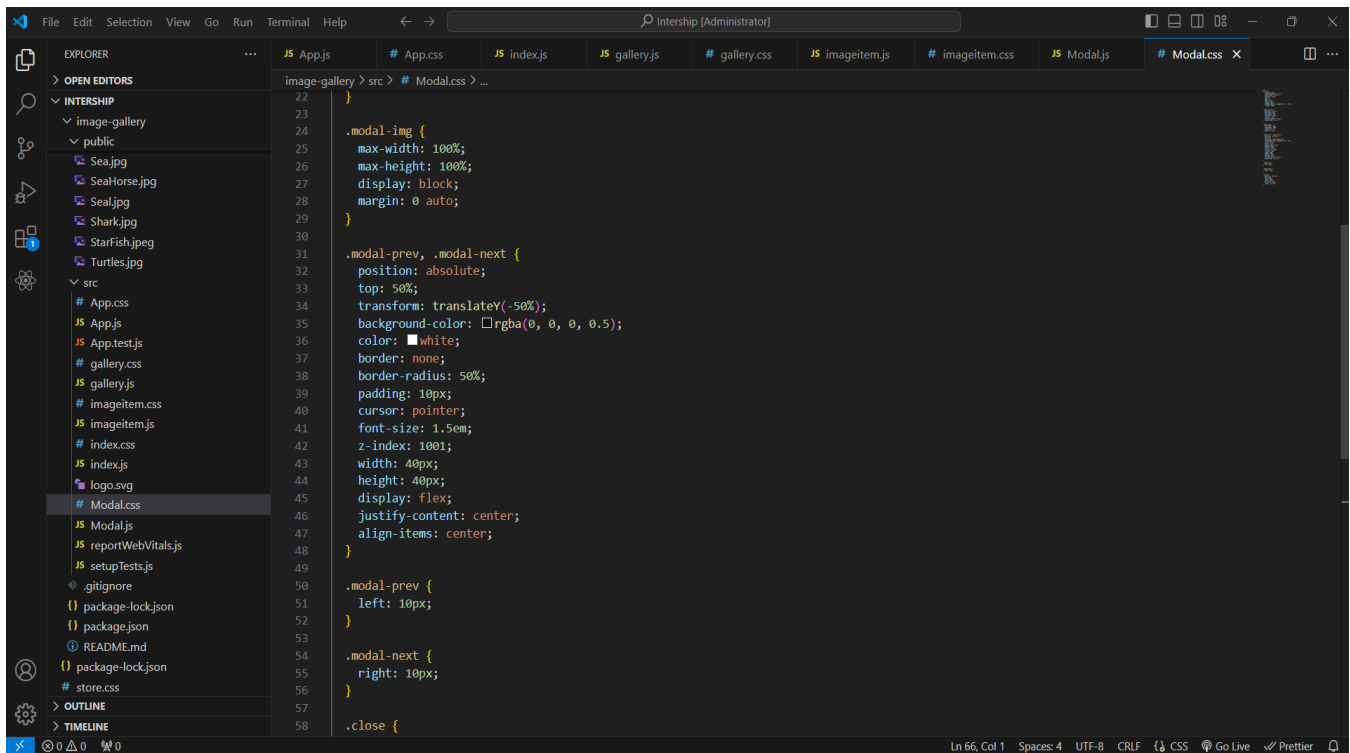
```
1 import React from 'react';
2 import './Modal.css';
3
4 const Modal = ({ image, closeModal, nextImage, prevImage }) => {
5   return (
6     <div className="modal" onClick={closeModal}>
7       <span className="close" onClick={closeModal}>&times;</span>
8       <button className="modal-prev" onClick={prevImage}>&lt;</button>
9       <div className="modal-content" onClick={(e) => e.stopPropagation()}>
10        <img className="modal-img" src={image.src} alt={image.alt} />
11      </div>
12      <button className="modal-next" onClick={nextImage}>&gt;</button>
13    </div>
14  );
15 }
16
17 export default Modal;
```

# Modal.css

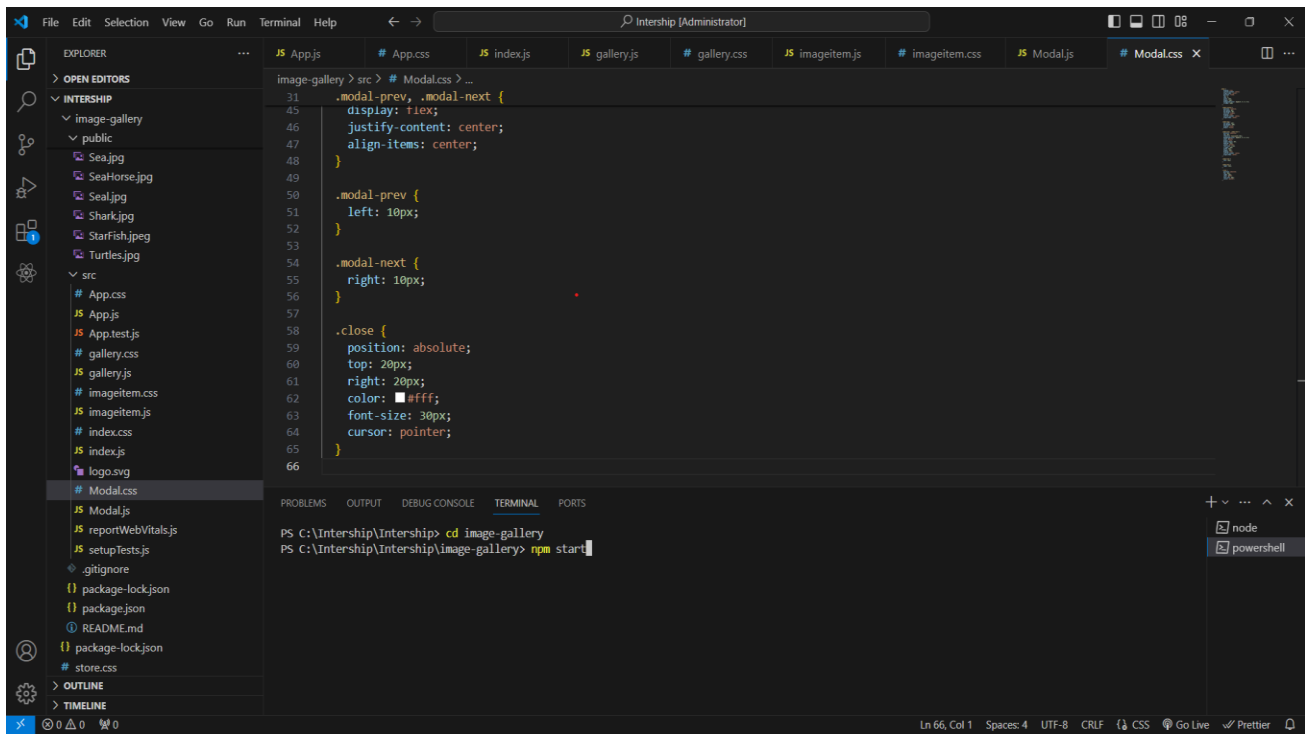


The screenshot shows the VS Code editor with the file explorer on the left and the code editor in the center. The file explorer shows a project structure with a 'src' directory containing 'App.js', 'App.test.js', 'gallery.js', 'imageitem.css', 'imageitem.js', 'index.css', 'index.js', 'logo.svg', 'Modal.css', and 'Modal.js'. The code editor displays the content of 'Modal.css'.

```
1 .modal {
2   display: flex;
3   justify-content: center;
4   align-items: center;
5   position: fixed;
6   top: 0;
7   left: 0;
8   width: 100%;
9   height: 100%;
10  background-color: rgba(0, 0, 0, 0.8);
11  z-index: 1000;
12 }
13
14 .modal-content {
15   position: relative;
16   max-width: 80%;
17   max-height: 80%;
18   overflow: hidden;
19   display: flex;
20   justify-content: center;
21   align-items: center;
22 }
23
24 .modal-img {
25   max-width: 100%;
26   max-height: 100%;
27   display: block;
28   margin: 0 auto;
29 }
30
31 .modal-prev, .modal-next {
32   position: absolute;
33   top: 50%;
34   transform: translateY(-50%);
35   background-color: rgba(0, 0, 0, 0.5);
36   color: white;
37   border: none;
```



# Output



The screenshot shows the Visual Studio Code editor interface. The Explorer panel on the left displays the project structure, including files like `App.css`, `App.js`, `App.test.js`, `gallery.css`, `gallery.js`, `imageitem.css`, `imageitem.js`, `index.css`, `index.js`, `logo.svg`, `Modal.css`, `Modal.js`, `reportWebVitals.js`, `setupTests.js`, `package-lock.json`, `package.json`, `README.md`, and `store.css`. The main editor area shows the `Modal.css` file with the following CSS code:

```
31 .modal-prev, .modal-next {
45   display: flex;
46   justify-content: center;
47   align-items: center;
48 }
49
50 .modal-prev {
51   left: 10px;
52 }
53
54 .modal-next {
55   right: 10px;
56 }
57
58 .close {
59   position: absolute;
60   top: 20px;
61   right: 20px;
62   color: #fff;
63   font-size: 30px;
64   cursor: pointer;
65 }
66
```

The bottom panel shows the TERMINAL with the following commands:

```
PS C:\InterShip\InterShip> cd image-gallery
PS C:\InterShip\InterShip\image-gallery> npm start
```

