MUSEUM ARTIFACTS

Documentation

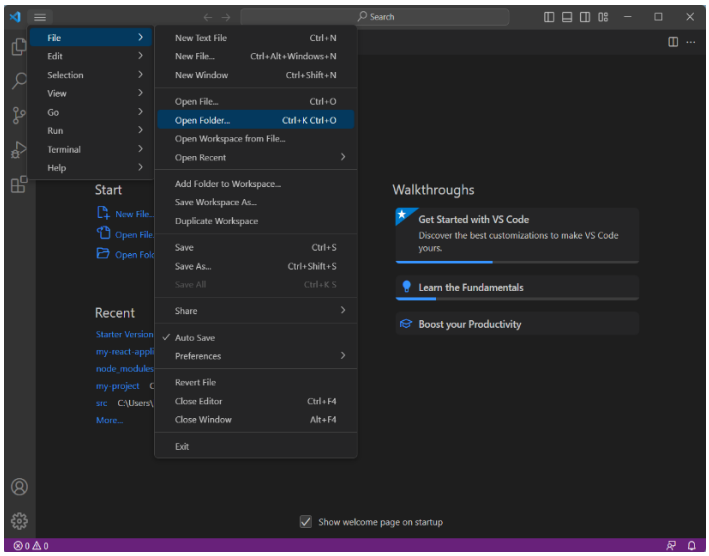
AUGUST 2, 2023

TEAM MEMBERS
Mahnaz Faizi, Roya Mohammadi, Shamayel Mohammadi , Shanana Noori

Description:

This project is about a museum and it works like this, that it has one header and three items, when you click on any of items the header will be change and the picture will rotate.

We downloaded the project that given by teacher as homework and open it in Visual Studio Code, for this work we go to VScode, file, open folder and then select our folder like this photo:

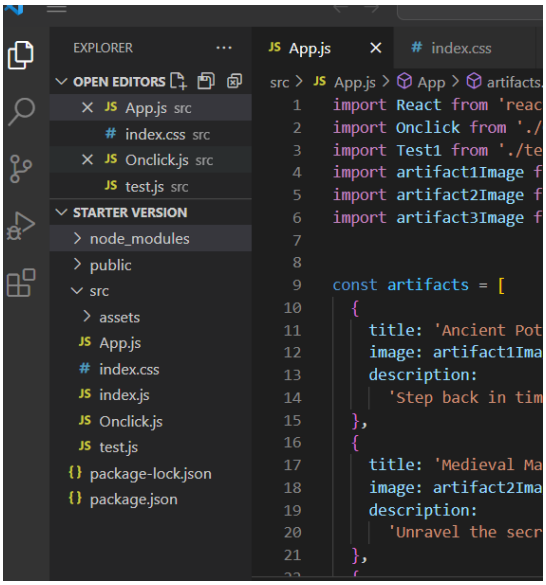


So, let's check the project, when you open the project you see some files like these:

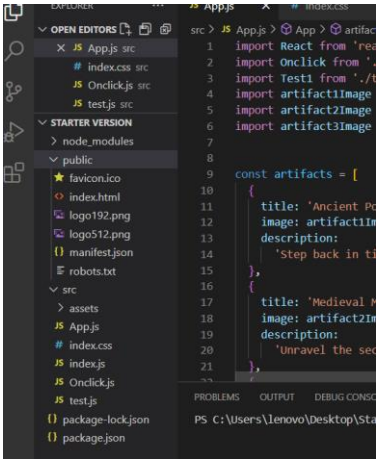
Node-modules, public, src, package-lock.json and package.json.

- **Node-modules**:

  This folder contains all the dependencies required for your React project. It is usually managed by the Node Package Manager (NPM) and should not be manually modified.

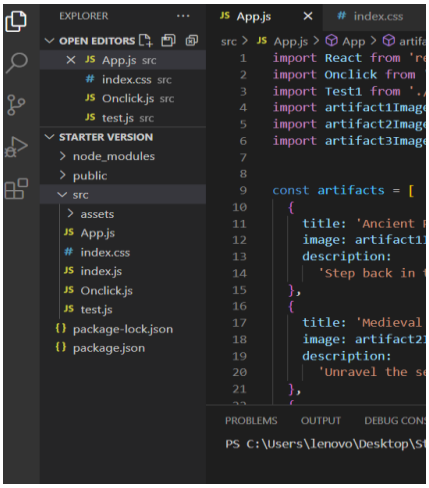


- **Public folder:** Contains index.html and other assets**.**
  - **Favicon.ico:** It contains the small icon that appears in the browser tab.
  - **Index.html:** This is the main HTML file that includes the root element.
  - **Logo.png:** Represent the logo image used in your application.
  - **Manifest.json**: It has information about the web application when we installed on a user's mobile device.
  - **Robots.txt:** it is used to provide instruction to web crawlers and search engines about which pages of your website to crawl or not**.**

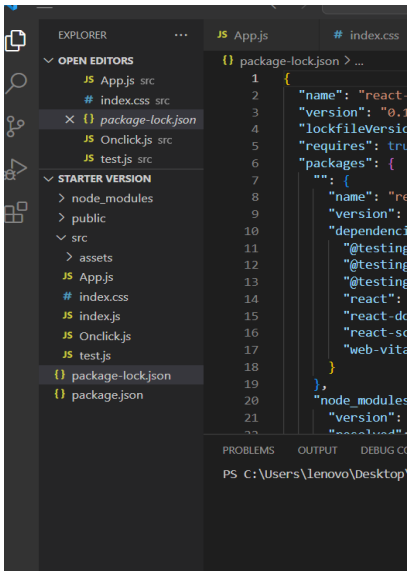


- **Src folder: Contains React code.**
  - **App.js*:*** this is the main component of your React application. It defines the structure and behavior of the root component.
  - **Index.css*:*** This CSS file is used to style the main index.html file.

- **Index.js*:*** This is the entry point of your React application. It imports the App component and render it into root element defined in index.html.
- **Onclick*:*** This file defines when we click, which action should occur.
- **Test.js:** the header of project is in this file**.**

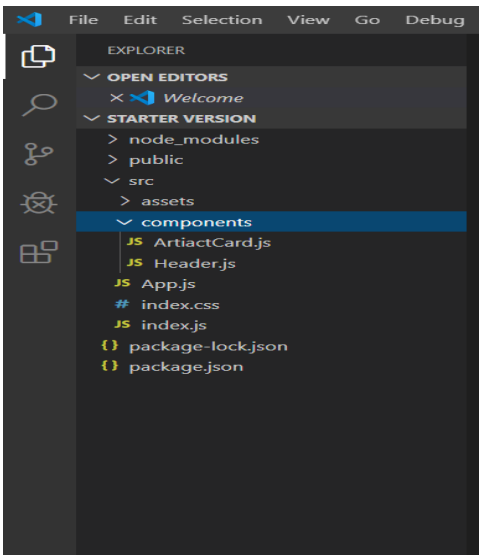


- **Package-lock.json*:*** This file is automatically generated and is used by NPM to lock down the version of each dependencies are installed**.**



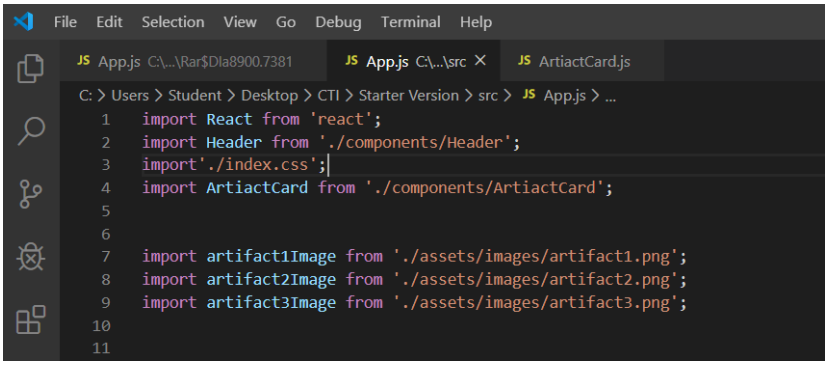
- **Package.json:** This file contains information about your project and its dependencies.

- At the beginning of the work, we created a folder inside the src folder called components. Inside this folder, we created tow components, Header.js component and ArtiactCard .js component.



Description of the (App.js page)

- **App.js**: this is the main component of our react application. it defines the structure and behavior of the root component.

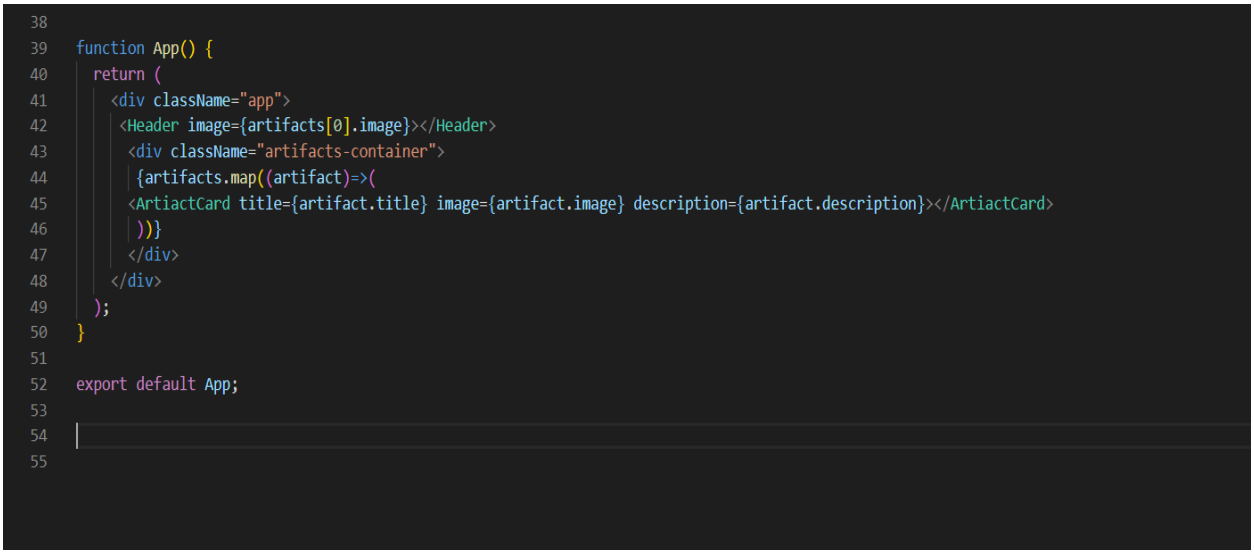


- From the first line to the eight line, all the pages where the components were created and its presence on this page was required have been imported.

- **Line 1-** this line of code is very important because it directly identifies all the package on this page.
- **Line 2-** this line of code indicates the header component, whose codes are located on a separate page called the header component. by importing that page, we can use the Header component where the header is needed.
- **Line 3-** all the styles created on index.css are applicable on the app.js. the app.js page can be styled using index.css classes and relevant id`s.
- **Line 4-** it takes all necessary photo on this page from the ArtiactCardComponent page.
- **Line 6,7,8-** it imports the necessary photos from their folders so that they can be used on this page.

```
9
10  const artifacts = [
11    {
12      title: 'Ancient Pottery',
13      image: artifact1Image,
14      description:
15        'Step back in time and admire the intricate craftsmanship of ancient pottery. Discover how civilizations across history used pottery for various purposes, from
          daily use to religious ceremonies.',
16    },
17    {
18      title: 'Medieval Manuscripts',
19      image: artifact2Image,
20      description:
21        'Unravel the secrets of medieval manuscripts. Explore beautifully illustrated texts that were meticulously crafted by skilled scribes and illuminated with vibrant
          colors.',
22    },
23    {
24      title: 'Egyptian Hieroglyphs',
25      image: artifact3Image,
26      description:
27        'Decode the mysteries of Egyptian hieroglyphs. Journey into the world of ancient Egypt and learn about the hieroglyphic writing system used to record their rich
          history and culture.',
28    },
29  ];
30
```

- **From the tenth line to thirty-fifth line –** there is an array whose name of the artifacts, the name of image and the characteristics of each artifact are mentioned. Each artifact has its own characteristics.These artifacts and their characteristics are used in the app.js page. In the absence of this array, the entire project will encounter an error because every artifact is used in the following lines.



```
38
39  function App() {
40    return (
41      <div className="app">
42       <Header image={artifacts[0].image}></Header>
43        <div className="artifacts-container">
44         {artifacts.map((artifact)=>(
45        <ArtiactCard title={artifact.title} image={artifact.image} description={artifact.description}></ArtiactCard>
46         ))}
47        </div>
48      </div>
49    );
50  }
51
52  export default App;
53
54
55
```

- **Line 39- function App () {…}:** this is a functional component named App. In react, components are the building blocks of the user interface. Functional components are defined using JavaScript functions.
- **Line 40- return (…):** this is the return statement of the functional component. It defines what the component will render and display on the screen. The return value is wrapped in parentheses, and it can only return a single root element.
- **Line 41- :** this is the root element that wraps the content of the component. It acts as the parent container for the component`s content. Takes the photo from the header and brings it into the app.js page.
- **Line 43-to-48 :** from the 43rd line to 48th line, we have a div in which we created a map so that all three artifacts and their characteristics are entered in the app.js page with all the details.
- **Line 52- export default App:** this line exports the App component so that it can be imported and used in other parts of the application. By using export default, you can import the component in other files using the import statement.

Description of the (Header.js page)

In this component, we separate the header section code from app.js.

- **Line 1**: We import and export a function named Header, which has a prop parameter.
- **Prop:** This input receives information, including an image, from app.js.
- **Line 2:** Return statement: It includes JSX code, which consists of a <header> tag. Inside the <header> tag, there are several other tags, including an <img> tag, an <h1> tag, and a <p> tag.
- **<img>:** It includes attributes "id="headerImage", src={props.image}, alt="Ancient pottery".
- **<h>:** It includes text.
- **<p>:** It includes a paragraph.

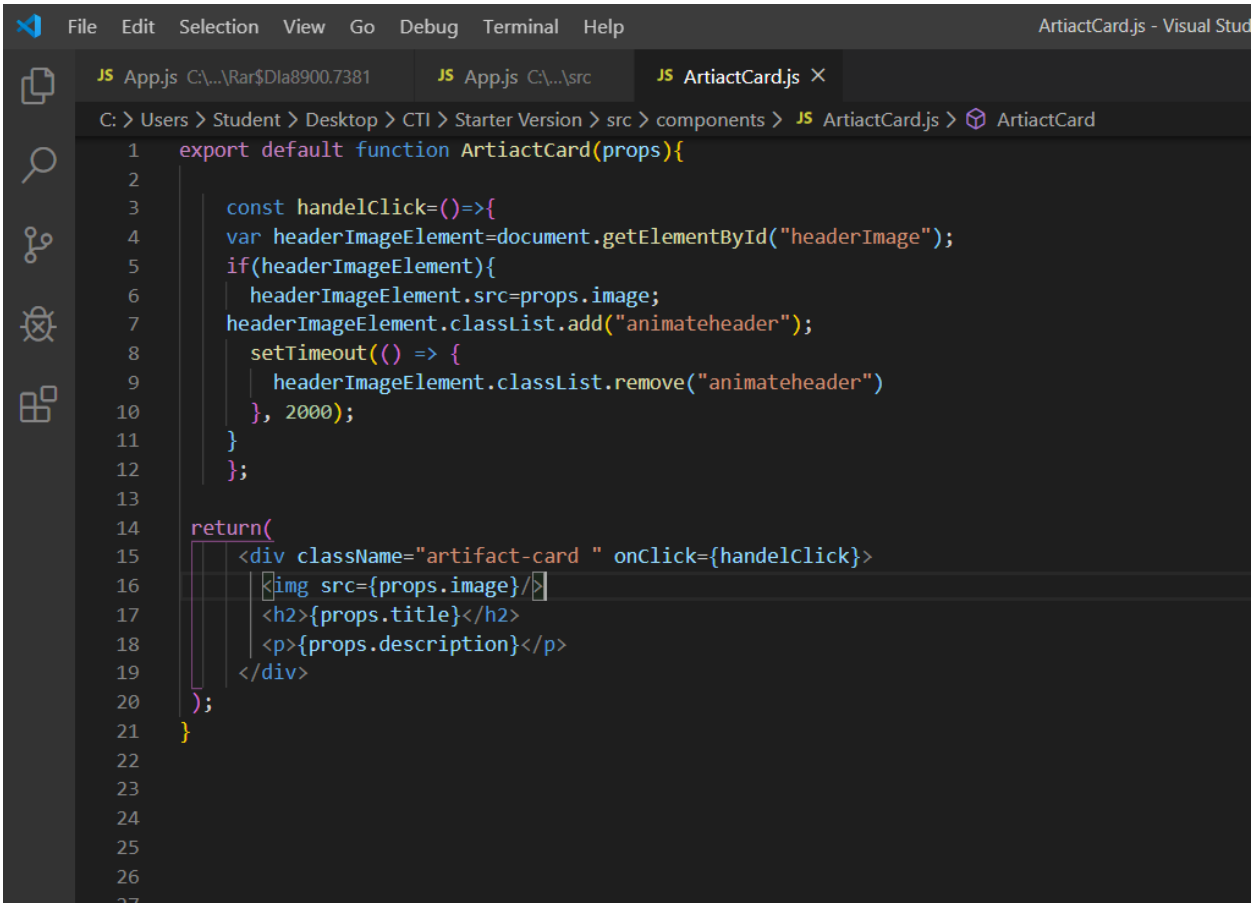
Description of the (ArtiactCard.js page)

In this component, we need to dynamically fetch information, including an image, title, and description. We also need to create a function that, when any of the cards are clicked, displays the image from the card's header in our project's header and triggers an animation to rotate the image.

- **The code before the change:**
  The initial code inside app.js looked like this

```
39
40      <div className="artifacts-container">
41        <div className="artifact-card"  >
42          <img src={artifact1Image} alt="Artifact" />
43          <h2>Title</h2>
44          <p>Description</p>
45        </div>
46        <div className="artifact-card"  >
47          <img src={artifact2Image} alt="Artifact" />
48          <h2>Title</h2>
49          <p>Description</p>
50        </div>
51        <div className="artifact-card"  >
52          <img src={artifact3Image} alt="Artifact" />
53          <h2>Title</h2>
54          <p>Description</p>
55        </div>
56      </div>
```

After changing the code and converting it into a component:



- **Line 1:** Importing and exporting a function called ArtifactCard with a parameter called props.

  The props parameter: in this component is an input that contains information received from App.js. This information includes the title, image, and description related to each card. This information is dynamically displayed in HTML tags.

```
2
3     const handelClick=()=>{
4     var headerImageElement=document.getElementById("headerImage");
5     if(headerImageElement){
6       headerImageElement.src=props.image;
7     headerImageElement.classList.add("animateheader");
8       setTimeout(() => {
9         headerImageElement.classList.remove("animateheader")
10      }, 2000);
11    }
12    };
13
```

- **Line 3:** Creating a function called handleClick that does the following: Whenever any of the cards are clicked, it retrieves the id of the header image using this statement and stores it in a variable called headerImageElement.

- **Line 5:** Using if/else statements to check if headerImageElement exists. If it does, set the image in the header and add the animateheader class, which was previously created in index.css, to headerImageElement.

- **Line 8:** setTimeout function
  This function is created to limit the duration of the animation or image rotation.
  Basically, this function removes the animateheader class that we previously added to headerImageElement and stops the rotation of the image after a certain time.

```
13
14  return(
15      <div className="artifact-card " onClick={handelClick}>
16        <img src={props.image}/>
17        <h2>{props.title}</h2>
18        <p>{props.description}</p>
19      </div>
20  );
21  }
22
23
```

- **Line 14:** return statement
  which includes a tag. This tag has two attributes: "className="artifact-card" and {handleClick}=onClick. In this event, the handleClick function is taken as a parameter, so whenever the cards are clicked, this function is called.

  Lines 16-20: tag containing HTML tags <image>, <title>, <description>. Each of them gets their content through props, which is passed as a parameter in the ArtifactCard component.