Training Day22 Report

10 July 2024

Folder structure in React

After you are done with the setup, you can see an application folder generated for you inside the path that you had chosen while creating the application. Inside this folder, you can see a bunch of files too, which have been created automatically.

Before starting to build the application, let's give you an overview of these files and folders inside your application folder. To do this, you need to open the application folder inside a text editor. You will be using **Visual Studio Code** as your text editor for this entire module. However, you can use any other text editor too.

Follow these steps to open your application folder inside the text editor:

1. Open Visual Studio Code.

2. If you are on *mac*OS, click on the *File* menu at the top and then click on "*Open...*". In case you are using *Windows*, click on the *File* menu at the top and then click on "*Open Folder*".

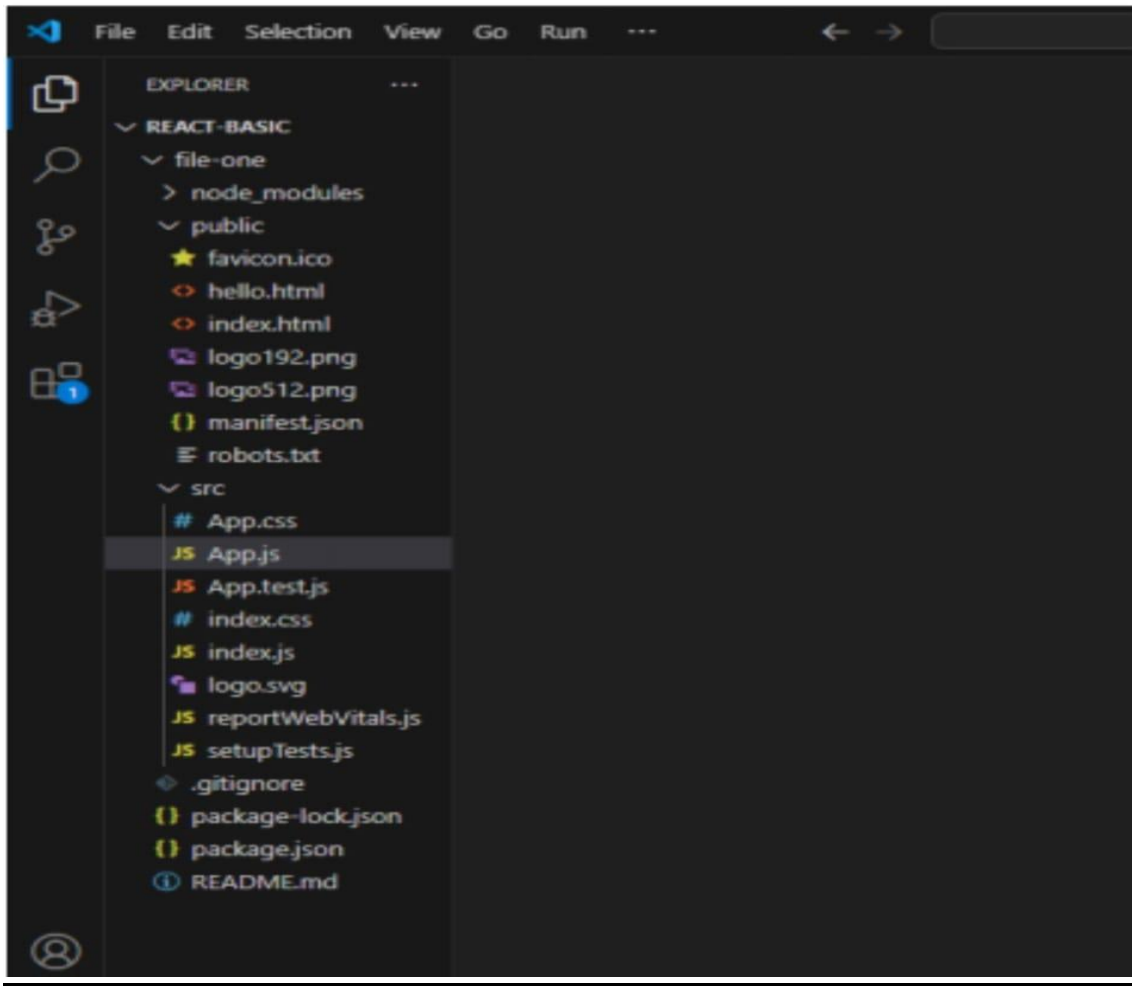
3. In the dialog box that opens, go to the location where your application folder resides and choose your application folder. Here, you have created the *phone-*

directory folder inside the Desktop. So, go to Desktop and choose the *phone-directory* folder.

4. Click on *Select Folder.*

When you look at the *Explorer* on the left side in *Visual Studio Code*, you can see the name of your application folder. You can also see a list of all the files and folders that are inside your application folder.

Given below is an screenshot of how it looks:



Note that the *index.html* file inside the *public folder* is the starting point of your application. **index.html** file should always remain with the same name and inside the same location; otherwise, your application does not run and throws an

error. Similarly, *index.js* is the entry point for all of your JavaScript code. Like index.html, the name and location of index.js should not be altered. You must keep this in mind in order to ensure that your code runs successfully.

Following are the basic points to note about some files that you see inside your application folder:

.gitignore file

- It is used by Git to determine which files and directories to ignore before a commit is made.
- It should be committed into the repository to share ignore rules with other users who clone the repository.
- The *node_modules* folder is included inside the *.gitignore* file so that the user who clones the application is not required to clone this folder. The user simply needs to run the command
- npm install
  In the root folder of the project. This command creates the *node_modules* folder and installs all the dependencies (packages) needed for the application.

package.json file

- It consists of the name and version of the application, the combination of which should be unique in order to publish the package.
- It comprises of dependencies that list all the packages needed to be installed for the application.

- It also includes scripts that specify the commands to be run at various points in the application lifecycle.

package-lock.json file

- It is automatically generated for any operation where npm modifies either the *node_modules* tree or the *package.json* file.

- It locks the version of the full dependency tree of packages.

- It guarantees the generation of an identical dependency tree when the application is cloned by other developers.

node_modules folder

- Its contents are defined by the *package.json* file and it consists of all the packages required for running your application.

public folder

- Nothing inside this folder is processed by *Webpack.*
- It is used for keeping small files that are not required to be bundled.
- It can be used to contain images when there are thousands of them, and their paths need to be referenced dynamically.
- Any file inside this folder needs to be referenced at other places using the *%PUBLIC_URL%/* keyword, which gets replaced with the path of the public folder during the application's build process.

index.html file

- It is the starting point of the application.
- It should always remain with the name *index.html* and inside the *public* folder; otherwise, the code will fail to run.
- It can only reference files that are inside the *public* folder.

manifest.json file

- It is a simple JSON file telling the browser about the web application and how the application should behave when it is installed on the user's mobile device or computer.

src folder

- It consists of the real application code.
- It consists of all the files that are needed to get bundled by *Webpack.*

index.js file

- It is the entry point for JavaScript.
- The filename should remain *index.js* and the location should be inside the *src* folder; otherwise, the code will not run.

index.css file

- It is the stylesheet for *index.html*.

registerServiceWorker.js file

- It is the web browser API that is used to cache assets and other files to work passively in the background. It helps offline users or the ones who are on the slow network to see results on the screen faster.
- It adds offline capabilities to the application.

App.js file

- It is the JavaScript file for the *App* component.

App.css file

- It is the stylesheet for the *App* component.

App.test.js file

- It is the test file for the *App* component.
- It contains unit tests for the application.
- It runs test cases for all the files that changed since the last commit of the application.

logo.svg file

- SVG is an acronym for Scalable Vector Graphics.
- An SVG file is an XML-based vector image format for 2D graphics with support for interactivity and animation.

- It is similar to raster-based image formats such as JPEG, PNG, BMP, GIF, etc.
- It offers a bandwidth-friendly way of rendering images; no matter how large a graphic gets, it transmits only the XML describing the graphic to the client.
- It helps to render resolution-independent and SEO-friendly images.
- It makes up the icon for your application and appears alongside the title in the browser tab.
- It gets saved along with the bookmark.