Create an app:

* Import “{browser-router} from “react-router-dom”
* Install the dependcy “react-router-dom” in package.json

In Src, create a folder “Routes” and add 4 files”Home,Project,Contact,About”.

In first file,Home

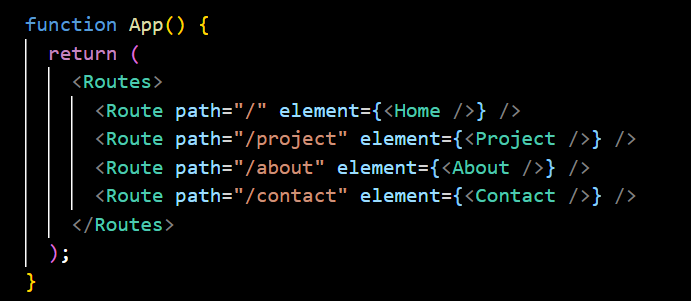
1.Home:

Shortcut: “rafce” [Install ES7 React/Redux/React-native snippets from Extensions]

And repeat it for all the pages

2.Now go to **App.js** and import

**import { Routes, Route } from "react-router-dom";**



Use Routes and Route so that when clicking that component in the webpage it will go to new url .

Now we have created the required pages

Graphical user interface, application

Description automatically generated

**Create Components:**

1.Navbar component:

In Navbarstyle.CSS file:

**TIP:**In the same container, to make them display the contents in the same row use Flex,justify-content

Graphical user interface, text

Description automatically generated

**TIP:** To choose a font, go to google and search google fonts

* Choose a font style and copy all the sizes
* Import them
* Paste that url in index.css {The reason to paste it in index.css file,this font should be global throughout the app}

Text

Description automatically generated

* Additionally,add this too

Timeline

Description automatically generated with low confidence

Copy paste that font-family from that font page:

Graphical user interface, text, application

Description automatically generated

* To change the background:

A picture containing graphical user interface

Description automatically generated

* Assigning text color to the texts. For that include all the tags

A picture containing logo

Description automatically generated

* To make Web responsive,use media query in navbarstyle.css

Text

Description automatically generated

* Add Hamburger menu in Navbar.js:

Text

Description automatically generated with medium confidence

Import icons to use in React app.

Text

Description automatically generated with medium confidence

To use that ,first install Text, logo

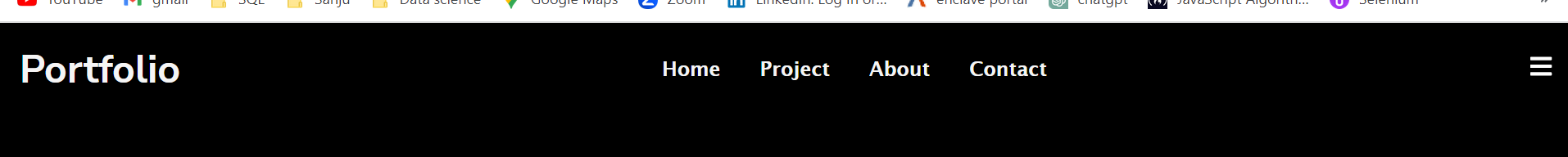
Description automatically generated

Now,style it

Text

Description automatically generated

Current output in large screen device:



I don’t want this hamburger menu icon to be visible in large screen device.So in CSs file,

Text

Description automatically generated with medium confidence

To make it visible in small screen devices, write this inside media query:

A picture containing diagram

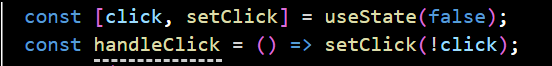
Description automatically generated

o/p in small screen:

Text

Description automatically generated

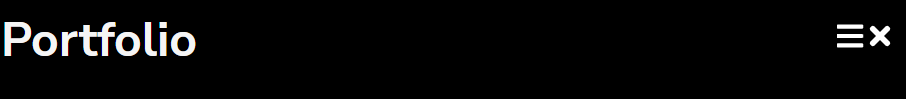
Now,We don’t want those names to display when openingthe page.Only when hamburger menu is clicked,it has to open.For that apply JavaScript functions



Now menu icon is created and when clicked menu appears.

Now close icon is to be created.



Now o.p will be 

Both icons appear. So use if else logic

Text

Description automatically generated

Now when opening the page, the menu list should not appear. IT should come when only hamburger menu is clicked.

For that, In Navbarstyle.css ,in @query ,change left to -100%[it will hide the contents completely out of the screen]

Graphical user interface

Description automatically generated with low confidence

Then when the hamburger menu is clicked,the items must appear.

For that create a styling class in css

Text

Description automatically generated

Now, go to Navbar.js file and set which CSS style should be called when.

For that, go to that div and use ifelse logic to call which CSS style should be called

Text

Description automatically generated

Now the o/p is

Text

Description automatically generated

Remove that “.” In “nav-menu.active”

Text

Description automatically generated

Now the o/p is :

Text

Description automatically generated

Finally add a transition:

Text

Description automatically generated

**2nd Component;**

HeroImg.js

After creating the js file import it and call in the Home.js file.

Create a Assets folder to store all the images.

Now start editing the HeroImg.js file

* Rafce
* Now create image, first import the picture

}};

* Text

  Description automatically generated

Now add the para and buttons as **link** to Projects and Contact.

Text

Description automatically generated

Now in CSS, adjust the image:

Text

Description automatically generated

o/p



Now apply a black layover to the image:

Text

Description automatically generated

o/p:

A computer on a desk

Description automatically generated with low confidence

Align the text and buttons to the middle with proper padding:

Text

Description automatically generated

Text

Description automatically generated

o/p:

A screenshot of a video game

Description automatically generated with medium confidence

Now add media query:

Text

Description automatically generated

Make some changes in Navbar to change the background color of Navbar and make it sticky.

Update this in css file:

Text

Description automatically generated

TIP: To control the order of items displaying in the layer,use Z-indexIn CSS, **z-index: 10;** is a property that specifies the stacking order of positioned elements on the z-axis (i.e., the depth of elements in the stacking context).

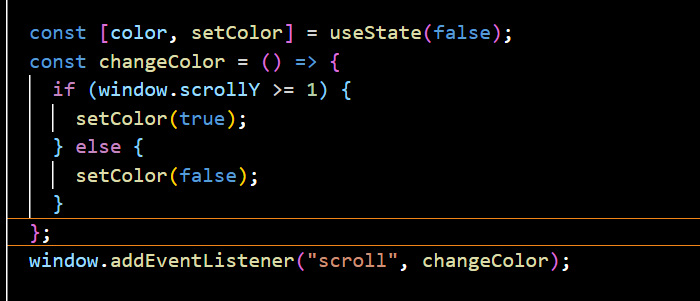
Now when scrolling,the navbar color should change. For that in CSS file, add a new style

Text

Description automatically generated

Now in Navbar.js file, use useState to apply that color should change when scrolling. Since scrolling is a window event, add event listener to call that function.

Then in the div classname use if condn to choose which css style should be used when





**Create the footer component:**

Divide the footer into left and right.

Create the home icon:



o/p:

A picture containing graphical user interface

Description automatically generated

Likewise,create icons for phone,email in the left and social media icons in the right.

Text

Description automatically generated

Text

Description automatically generated

o/p:

Text

Description automatically generated

Add web responsive styles in CSS

**Create Hero component for all the routes[projects,contact,about] and use props to send the text to display.**