# Assignment

### Tasks to complete:

Note: Install Node and NPM before getting started with below steps, Node version should be greater than 12.

## **Install create-react-app**

[create-react-app](https://github.com/facebookincubator/create-react-app) is a tool to build React apps in no time. It allows us to create React apps without any build configuration.

1. From your terminal, install create-react-app globally:

npm install -g create-react-app

2) Run below command in terminal/CMD  
create-react-app instaclone

cd my-app  
  
  
3) Run below command in terminal/CMD  
npm install react-router-dom

npm-start

**Let's get started.**

1. To start, make two directories for two different Layouts page: LandingPage, PostView.
2. Begin with the landing page component. Open landing\_page.js in your text editor:
3. Repeat 2nd step for post view page as well. Refer the below snipper for sample code of components.

Code for

1. import React from 'react';
2. export default function landing\_page() {
3. return <h2>Landing Page</h2>;
4. }
5. Next, import landing\_page and render inside the <div>. This will serve as a placeholder until you add more routes:

Code for App.js

import React from 'react';

import './App.css';

import landing\_page from '../ LandingPage/landing\_page;

function App() {

return (

<div className="wrapper">

<h1>Landing Page</h1>

<landing\_page/>

</div>

);

}

export default App;

1. Repeat 4th step for PostView component creation.

## **Adding Routes**

Refer to this Link🡪 <https://www.digitalocean.com/community/tutorials/how-to-handle-routing-in-react-apps-with-react-router>  
  
Create two component as mentioned in below screenshot and implement routing to post view page on clicking Enter button from Landing Page.  
  
1.  
Graphical user interface

Description automatically generated

2. Use sample Json Object as mentioned below and Loop over to create Three sections using Map function in JSX from JSON as per below screenshot.

As JSON object grows like currently 3 objects inside array, if we increase to 4, then UI should render 4 containers.  
  
Note: you can store images in local and use relative path using ‘require’ functionality, which will keep in PostImage value of JSON  
  
[

{name:"Siva",

location:"Bengaluru",

likes:64,

description: "Kick start your career",

PostImage: "relative path from local",

date: new Date(),

},

{name:"Neeraj",

location:"Pune",

likes:30,

description: "Sample Description",

PostImage: "relative path from local",

date: new Date(),

},

{name:"Rahul",

location:"Hyderabad",

likes:30,

description: "Sample Description for Post",

PostImage: "relative path from local",

date: new Date(),

}

]

# Submission Steps

1. Create a new repository in your account and clone the repository in your local machine
2. Please open your local repo and make sure you are on the master branch
3. git checkout -b "DataPageswithReactrouters"
4. Create the react app and complete the assignment as mentioned above
5. Proceed with Staging using command🡪 git add . or use Vs code extension
6. Git commit or use Vs code extension
7. Git Push or use Vs code extension
8. Create the Merge Request as mentioned in below screenshots.
   1. Select the branch that checkout in step 2

Graphical user interface, application

Description automatically generated

* 1. Click on “Pull request”   
     Graphical user interface, text, application, email

     Description automatically generated
  2. Click on “New Pull Request”

Graphical user interface, text, application, email

Description automatically generated

* 1. Select the branch that checkout in step 4  
     Graphical user interface, application

     Description automatically generated
  2. Create a pull request and share link of Request as your submission.  
       
     Please take screenshot of final output from your browser and keep in Pull Request description.  
       
       
      Share the Branch URL: https://github.com/{Username}/{Repo Name}/ pull/{Request Number}  
     Graphical user interface, text, application

     Description automatically generated