| **Progress File-3** |
| --- |

# **React with NextJs**

1. **Starting with Nextjs**
2. **Page Layout**
3. **Custom Document**
4. **Reactstrap navbar**
5. **Env variables with Nextjs**
6. **Navigation using Link Component**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# **TASK-1 Starting with Nextjs**

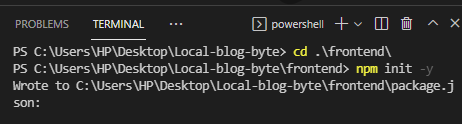
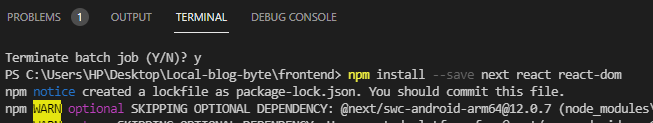
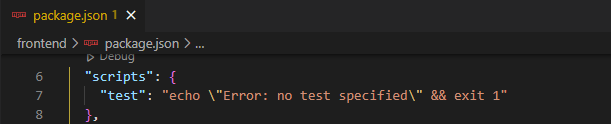
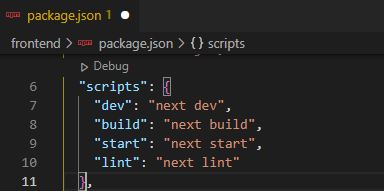
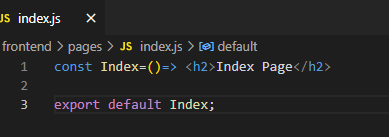
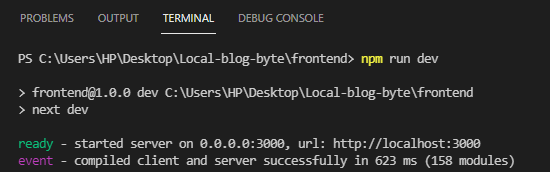
**STEPS :**

* Now we are going to build our front-end with **nextjs.**

**Next.js** is a React framework that enables several extra features, including server-side rendering and generating static websites

* To start, gi to next js official website <https://nextjs.org/> and go to docs to get startup commands.

|  |
| --- |

* Now go to **frontend** folder in terminal.And initialize it with **package.json** using command **npm init -y**
* 
* Now we will have **package.json** in **frontend**.
* Now inside **frontend** run the command **npm install next react react-dom** to install next js.
* 
* Now replace scripts in **package.json** file with the nextjs.
* 
* 
* Now we can run **npm run build,start and dev**.
* Now in **frontend** folder create a new folder named **pages**.Anything we create inside pages folder will ber served as a page in next js
* And inside pages create a new file called **index.js**.This file will be served as Home/Landing page
* Make a react component in it.
* 
* NOw in terminal run command **npm run dev** to start the server.
* 
* We can access the page using URL <http://localhost:3000/>.ANd index page will be displayed.

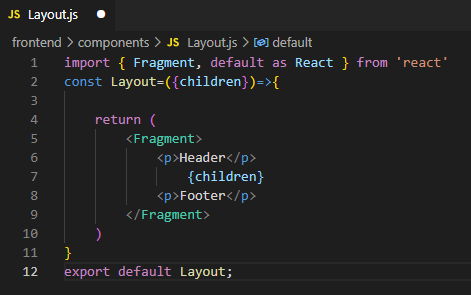
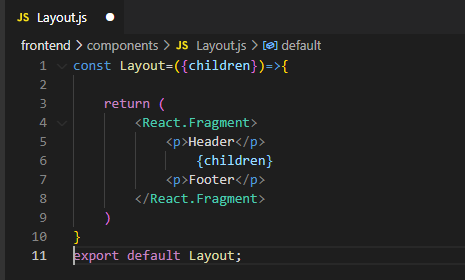
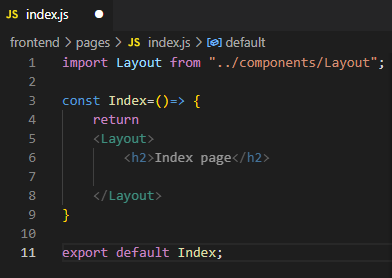
|  |
| --- |

**=>Next js has been setup successfully,**

**Task Completed… :)**

# **TASK-2 Page Layout**

**STEPS :**

* Now we are going to create a template that will remain same for all pages.
* Go to **frontend** folder and create a new folder named **components**, and inside components create a new file names **Layout.js**. (components name always starting with capital letters)
* Create a new component and return it with ReactFragment so we can return more than one elements.
* 
* Making a layout of **header,footer** and adding **props** as children in between.
* Anything we pass to this components will be render at place of **{children}**.
* 
* Now bring the component in **index.js** and put what we want to render inside Layout component.
* 
* Run the server again and we can see index page is using our defined layout.

|  |
| --- |

**=>Page Layout has been added successfully,**

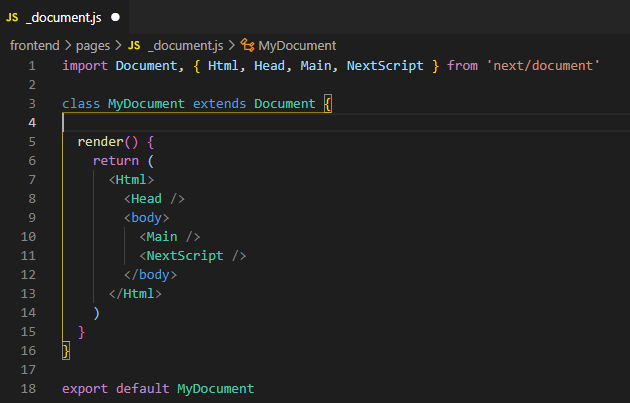
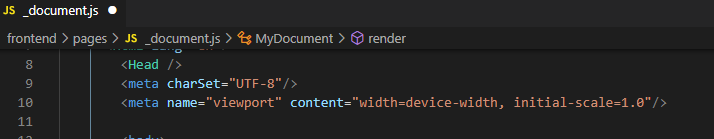
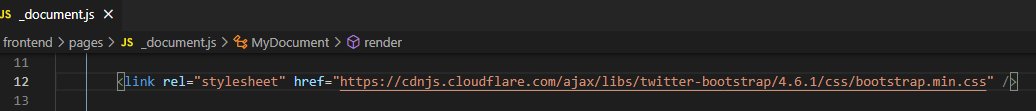
**Task Completed… :)**

# 

# 

# **TASK-3 Custom Document**

**STEPS :**

* Now to add **CSS** and **Bootstrap**, we will ad cdn in our project frontend.
* Next Js provide a page called **Custom Document** (A custom Document is commonly used to augment your application's <html> and <body> tags.) for adding it use the layout provide on link <https://nextjs.org/docs/advanced-features/custom-document>.
* Inside pages folder, create a new file \_document.js and paste code here from above link.
* 
* Now adding some meta tags in header.
* 
* Now adding **bootstrap cdn**from link <https://cdnjs.com/libraries/twitter-bootstrap> .
* 

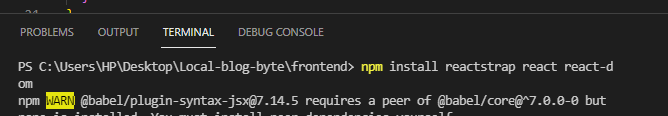
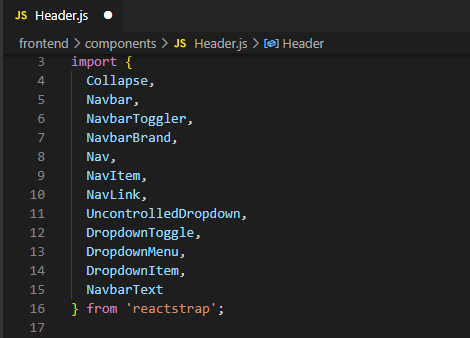
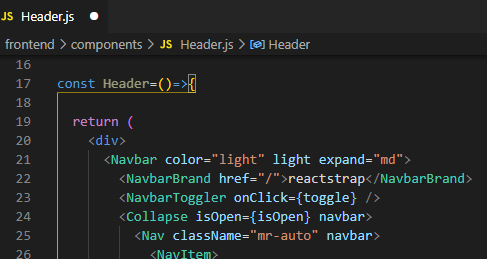
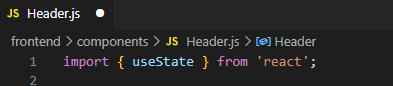
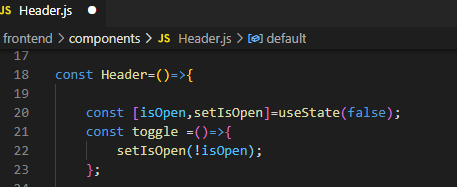
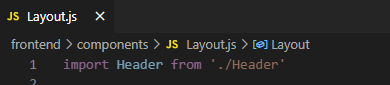
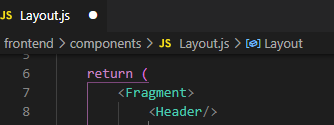
| **Page before Adding Bootstrap** | **Page change after adding Bootstrap** |
| --- | --- |

**=>Custom Document has been added successfully,**

**Task Completed… :)**

# **TASK-4 Reactstrap navbar**

**STEPS :**

* Now we will start working on header.
* Go to **reactstrap (**Allow us to use bootstrap classnames**)** <https://reactstrap.github.io/>, we will use this to implement **bootstrap** styling.Also we don't need to use Bootstrap Js and JQuery
* Install reactstrap using **npm install reactstrap react react-dom**.
* 
* Go to components, navbar and use the code from here. <https://deploy-preview-2356--reactstrap.netlify.app/components/navbar/>
* Inside **components** folder create a new components named **Header.js**.And paste the code here.
* 
* Add a functional component under it and return the code below it in this component.
* 
* Now import useState and use inside header.
* 
* 
* Now bringing this component in Layout.
* 
* 
* Run the server and see the navbar has been added to our project.

|  |
| --- |

* On changing the page width navbar changes its width, so its responsive.

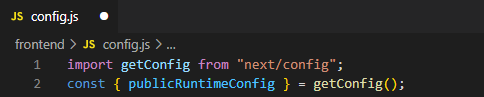
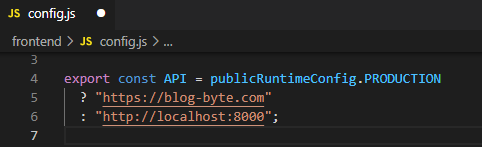
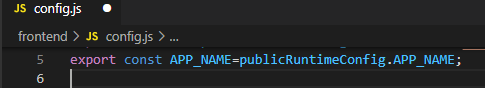
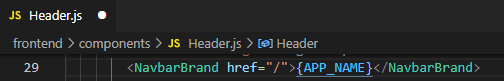
|  |
| --- |

**=>Reactstrap navbar has been added successfully,**

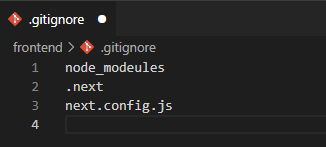
**Task Completed… :)**

# **TASK-5 Env variables with Nextjs**

**STEPS :**

* To use environment variables in nextjs, in **frontend** folder create a new file called **next.config.js**, and here we can write some configurations.
* We can export environment variables using publicRuntimeConfig.
* 
* To access these variables create another file inside **frontend** called **config.js**.
* In this file to use environment variables, we need to use getConfig that comes with nextjs.
* 
* Setting the api URL if production mode then and if development mode then.
* 
* Export AppName.
* 
* Now import AppName in **Header.js** **component** and use it here.
* 
* Use it inside navBar as navigation.
* 
* Now restart the server and see the AppName updated and is loaded from environment variables.

|  |
| --- |

* Creating a **.gitignore** inside **frontend** so we can commit out project without node\_modules and next folder.
* 

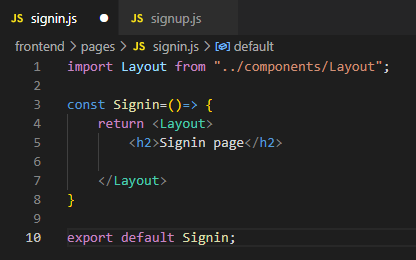
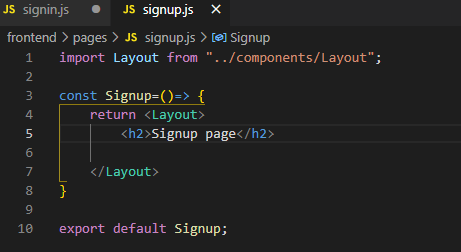
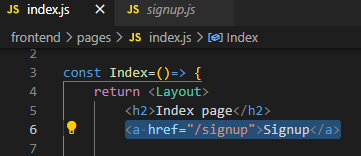
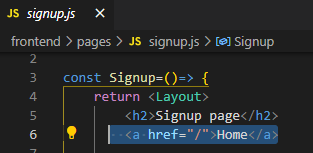
**=>Env variables with NextJs has been added successfully,**

**Task Completed… :)**

# **TASK-6**

# **Navigation using Link Component**

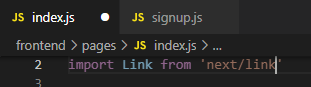
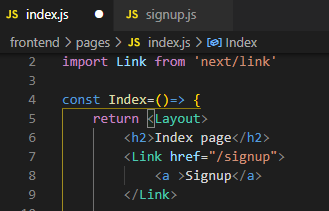
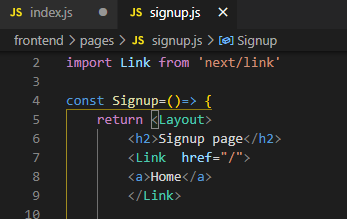
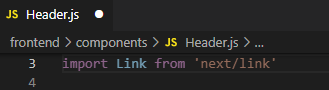
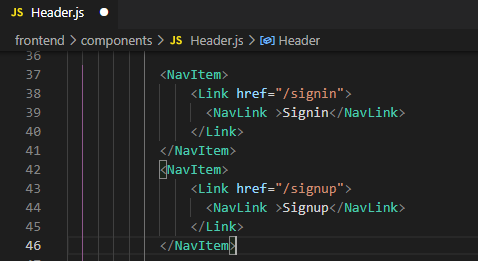
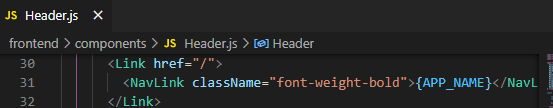
**STEPS :**

* Now lets create few pages.Go to **frontend** folder, inside **pages** folder create 2 files names **signin.js** and **signup.js**.
* Make a signin and signout component inside these files.
* 
* 
* For navigation add anchor tag<a> in index and signup files.
* 
* 
* Now start the server and see page navigation links.

|  |
| --- |

* After clicking on signup we will be navigated to signup page.

|  |
| --- |

* To avoid page reloading everytime we reload, bring **link component** from next, in **index.js** file.
* 
* Now instead of using **href**, we can use **Link**.  *It will only change component instead of reloading page.*
* 
* Doing the same in **signup.js** file.
* 
* Now restart the server and see there will be no page reloading on navigation.
* Go to **components** folder, go to **Header.js** and also use **Link** over here.
* 
* Add Links here.
* 
* Doing same in other nav components.
* 
* Restart the server and see the signup link is working properly in header.

|  |
| --- |

**=>Navigation using Link Component has been added successfully, Task Completed… :)**