For react need- node.js and npm

C:\>node --version

v14.16.1

C:\>npm --v

6.14.12

No page reload

**Difference b/t npx and npm**

**Node package manager- global install**

**Node package execute only**

**Npm is** a tool that use to install packages. **Npx is** a tool that use to execute packages. Packages used by **npm** are installed globally you have to care about pollution **for** the long term. Packages used by **npx** are not installed globally so you have to carefree **for** the pollution **for** the long term.

cd **myreactproject**

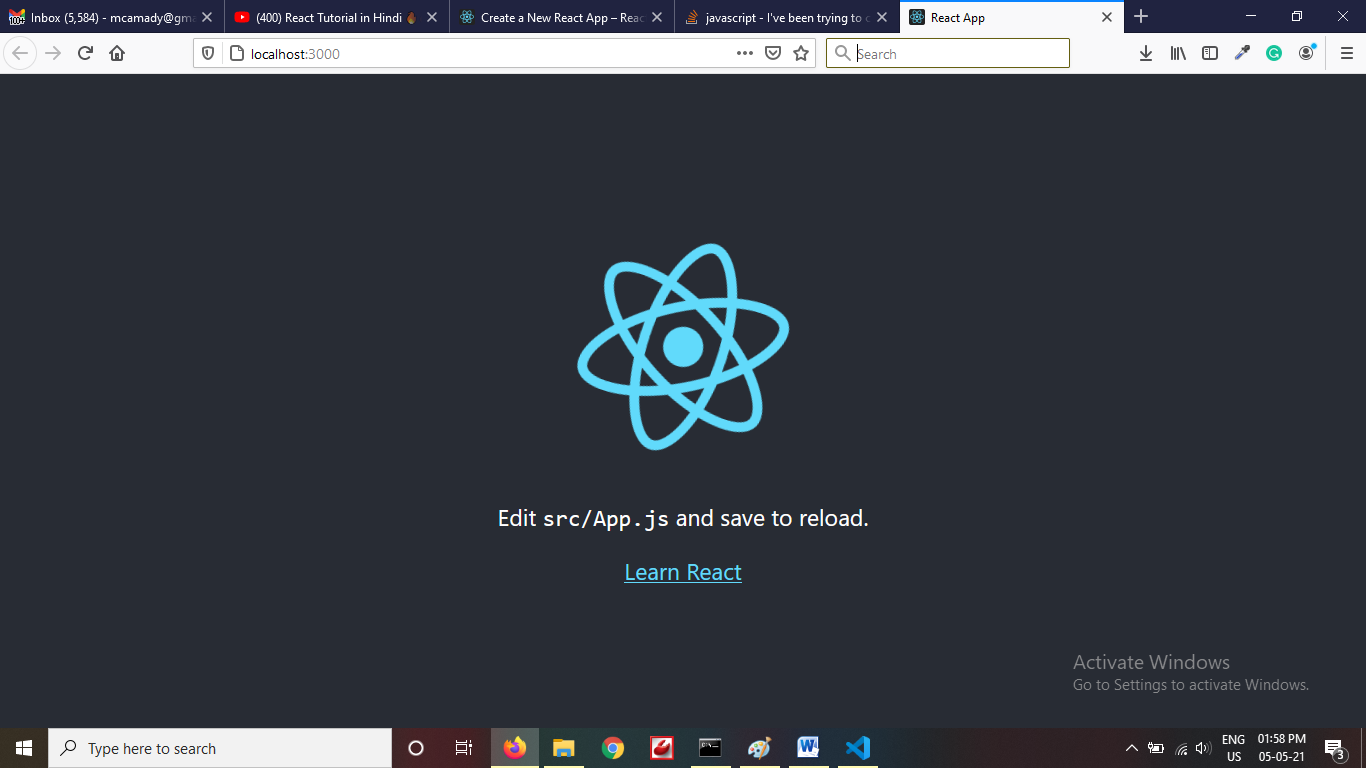
C:\>npm I npx

C:\myreactproject>npm install create-react-app

C:\myreactproject>npm start

C:\>cd my-app

C:\my-app>npm start



Local: http://localhost:3000

**On Your Network:** [**http://192.168.1.47:3000**](http://192.168.1.47:3000) **means anyone can access this app suing same wifi or lan network**

**SPA- Single page application**

In programming, the term **boilerplate** code refers to blocks of code used over and over again. Let's assume your development stack consists of several libraries, such as **React**, Babel, Express, Jest, Webpack, etc. ... A **boilerplate** is a template that you can clone and reuse for every project.

We can create own **boilerplate**

JSX- Javascript Extension

<https://medium.com/wesionary-team/react-functional-components-vs-class-components-86a2d2821a22>

src/app.js

function App() {

  //for varibale

  let abc=12

  return (

   <>

   <p>hello first code</p>

   </>

  );

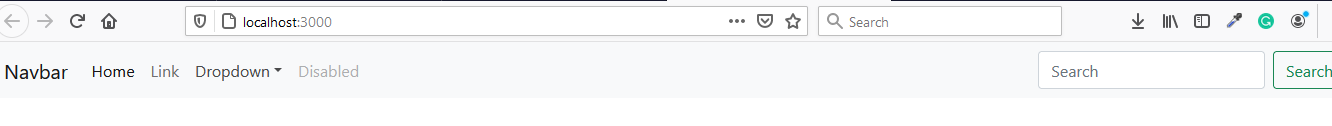
}

**Include bootstarp in project**

Add bootstrap css and js into idex.html page in public folder

Copy nav bar code form bootrap site into app.js

Change class to className and close all tags



If we see, we putted navbar component in app component, this is not gud way

So we will break our website into multiple parts component

So we have

Start Todo List

Todo item 1- start n close

Todo item 2- start n close

Todo item 3- start n close

End Todo List

Install extension -**ES7 React/Redux/GraphQL/React-Native snippets into vscode**

**Create folder C:\my-app\src\MyComponents\Header.js**

import React from 'react'

export default function Header() {

  return (

    <nav className="navbar navbar-expand-lg navbar-light bg-light">

    <div className="container-fluid">

      <a className="navbar-brand" href="#">Todo List</a>

      <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

        <span className="navbar-toggler-icon"></span>

      </button>

      <div className="collapse navbar-collapse" id="navbarSupportedContent">

        <ul className="navbar-nav me-auto mb-2 mb-lg-0">

          <li className="nav-item">

            <a className="nav-link active" aria-current="page" href="#">Home</a>

          </li>

          <li className="nav-item">

            <a className="nav-link" href="#">About</a>

          </li>

        </ul>

      </div>

    </div>

  </nav>

  )

}

**App.js**

import Header from './MyComponents/Header';

function App() {

  return (

    <>

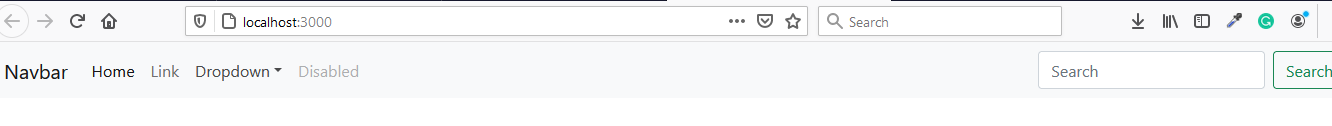
    <Header/>

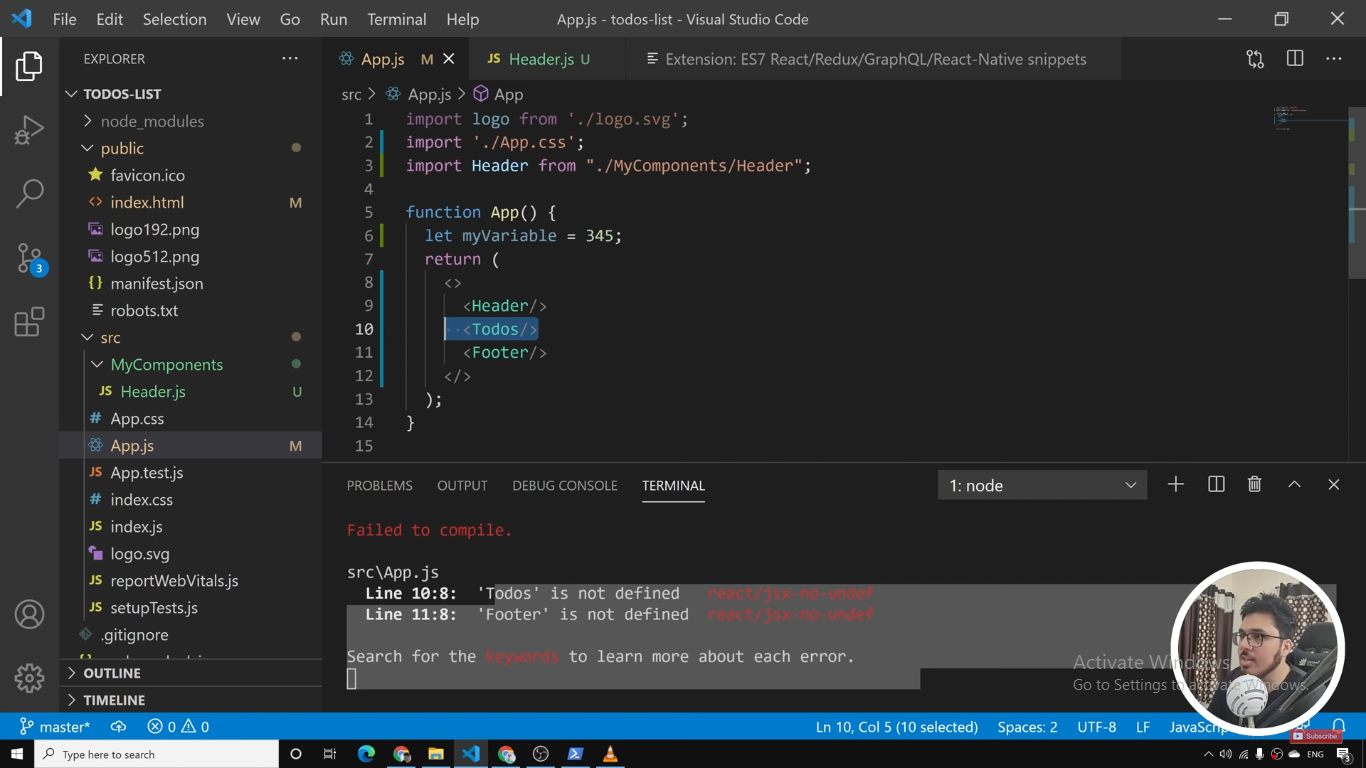
    </>

  );

}

export default App;





import Header from './MyComponents/Header';

import Footer from './MyComponents/Footer';

import Todos from './MyComponents/Todos';

function App() {

  return (

    <>

    <Header/>

            <Todos></Todos>

    <Footer/>

    </>

  );

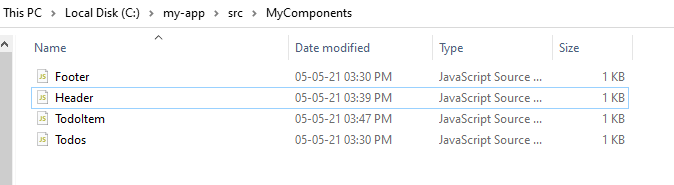
}

export default App;

**or**

**if we not make header or footer or todos as export default so we have to use**

**import {Header}** from './MyComponents/Header';

****

**Now we will use props**

**Get Parent data into child component using props, props are javascript object**

**Parent Component with title data pass to child component hello - <hello title=”mypage”>**

**How to get parent data into child component**

**Function hello(props)**

**{**

**Return(**

**<div>{props.title}</div>**

**)**

**});**

**In our code App.js is parent and all rest is child component**

**App.js**

function App() {

  return (

    <>

    <Header title="My Todo List" />

            <Todos></Todos>

    <Footer/> </>

  );

}

**Header.js**

import React from 'react'

export default function Header(props) {

  return (

    <nav className="navbar navbar-expand-lg navbar-light bg-light">

    <div className="container-fluid">

      <a className="navbar-brand" href="#">{props.title}</a>

      <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

        <span className="navbar-toggler-icon"></span>

      </button>

      <div className="collapse navbar-collapse" id="navbarSupportedContent">

        <ul className="navbar-nav me-auto mb-2 mb-lg-0">

          <li className="nav-item">

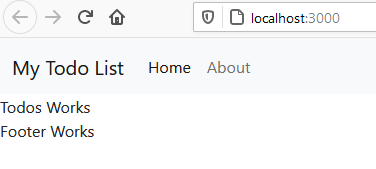
            <a className="nav-link active" aria-current="page" href="#">Home</a>

          </li>

          <li className="nav-item">

            <a className="nav-link" href="#">About</a>

          </li> </ul></div></div></nav> )}

****

**In desturctring Way example**

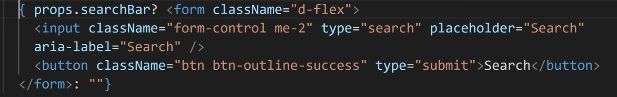
export default function Header({title}) {

 return (

      <a className="navbar-brand" href="#">{title}</a>

**}**

**If else in JSX**

****

**Default Props and Props types**

**For validation we use props type it will show warning in console if we passed wrong prop in child component**

**In header.js**

import PropTypes from 'prop-types'

Header.propTypes={

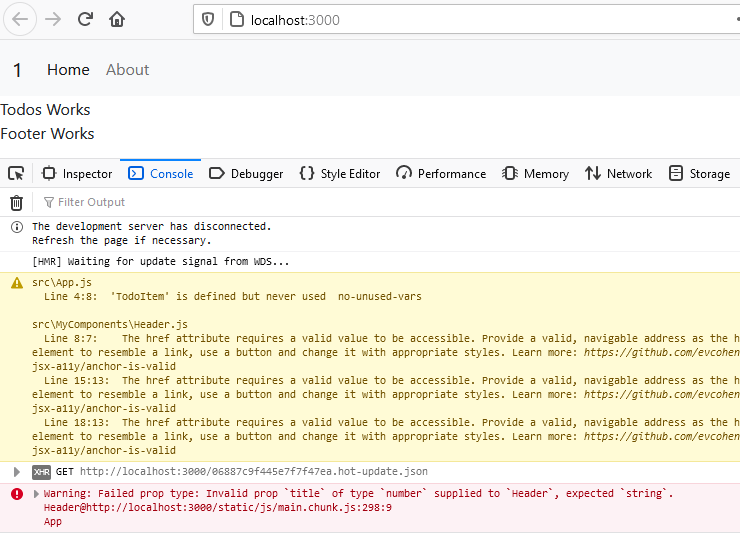
    title:PropTypes.string

}

**In top PropTypess should be PropTypess and Header.propType should be small and with props capital**

**In App.js**

<Header title={1} /> o/p will come up but warning will show



Now default props for child component

**If I missed to pass props which we are getting in child component, so it will generate error so we can use default props**

Header.js

export default function Header(props) {

return (

 <a className="navbar-brand" href="#">{props.title}</a>

)

}

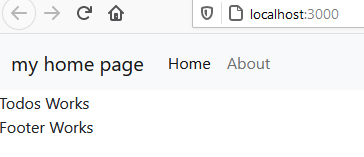
Header.defaultProps={

    title:"my home page"

}

App.js

<Header  />

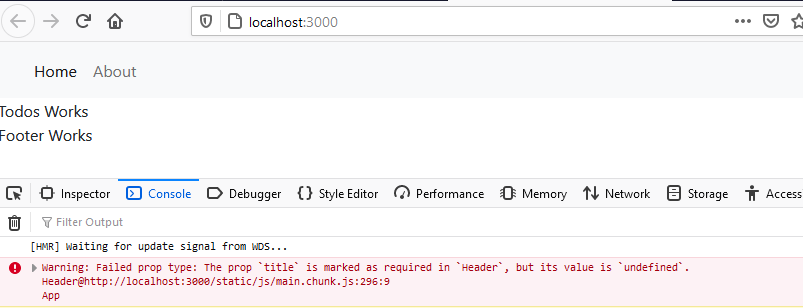
****

**Now we have to tell parent component if you want to access data into child component, you should pass value**

Header.propTypes={

    title:PropTypes.string**.isRequired**

}

****

**Lets come in Todos.js**

import React from 'react'

export default function Todos() {

  return (

    <div className="container">

     <h3>Todos List</h3>

    </div>

  )

}

**In App.js**

**Lets create multiple tasks using array of todos in app.js**

import Header from './MyComponents/Header';

import Footer from './MyComponents/Footer';

import Todos from './MyComponents/Todos';

import TodoItem from './MyComponents/TodoItem';

function App() {

  //todos array which have each todoitem

  let todos=[

    {

    sno:1,

    title:"go to market",

    desc: "Task 1 done"

  },

  {

    sno:2,

    title:"go to mall",

    desc: "Task 2 done"

  },

  {

    sno:1,

    title:"go to home town",

    desc: "Task 3 done"

  }

]

  return (

    <>

    <Header title="My Todo List" />

            <Todos />

    <Footer/>

    </>

  );

}

export default App;

**Let’s pass todos array this list into child component’ in App.js**

<Todos todos={todos} />

**In Todos.js**

import React from 'react'

import TodoItem from './TodoItem';

export default function Todos(props) {

  return (

    <div className="container">

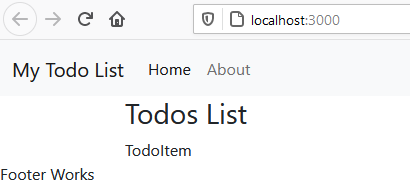
     <h3>Todos List</h3>

      <TodoItem /> //single todo, now we will use forloop for each item value

    </div>

  )

}

****

Todos.js

import React from 'react'

import TodoItem from './TodoItem';

export default function Todos(props) {

  return (

    <div className="container">

     <h3>Todos List</h3>

      <TodoItem todo={props.todos[0]}/>

    </div>

  )

}

**TodoItem.js**

import React from 'react'

export default function TodoItem({todo}) {

  return (

    <div><h4>Todo Item {todo.title} </h4>

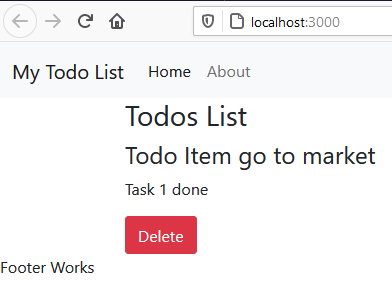
      <p>{todo.desc}</p>

      <button className="btn btn-danger">Delete</button>

    </div>

  )

}



**Now only first list item is showing so we need to add map function code instead of showing statically index 0**

**Todos.js**

import React from 'react'

import TodoItem from './TodoItem';

export default function Todos(props) {

  return (

    <div className="container">

     <h3 className="text-center">Todos List</h3>

     {props.todos.map((todo)=>{

       return  <TodoItem todo={todo}/>

     })}

    </div>

  )

}

[**https://www.youtube.com/watch?v=RGKi6LSPDLU**](https://www.youtube.com/watch?v=RGKi6LSPDLU)

**If we will use (props)- so we have to use props.todo or we can use destrutring {todos} only**

**onDelete is call back function we are use in that**

**App.js**

import Header from './MyComponents/Header';

import Footer from './MyComponents/Footer';

import Todos from './MyComponents/Todos';

function App() {

  const onDelete=()=>

  {

    console.log("hi i am deleted");

  }

  //todos array which have each todoitem

  let todos=[

    {

    sno:1,

    title:"go to market",

    desc: "Task 1 done"

  },

  {

    sno:2,

    title:"go to mall",

    desc: "Task 2 done"

  },

  {

    sno:3,

    title:"go to home town",

    desc: "Task 3 done"

  }

]

  return (

    <>

    <Header title="My Todo List" />

            <Todos todos={todos} onDelete={onDelete} />

    <Footer/>

    </>

  );

}

export default App;

**Todo.js**

import React from 'react'

import TodoItem from './TodoItem';

export default function Todos(props) {

  return (

    <div className="container">

     <h3 className="text-center">Todos List</h3>

     {props.todos.map((todo)=>{

       return  <TodoItem todo={todo} onDelete={props.onDelete} />

     })}

    </div>

  )

}

**TodoItem.js**

import React from 'react'

export default function TodoItem({todo,onDelete}) {

  return (

    <div>

      <h4>Todo Item {todo.title} </h4>

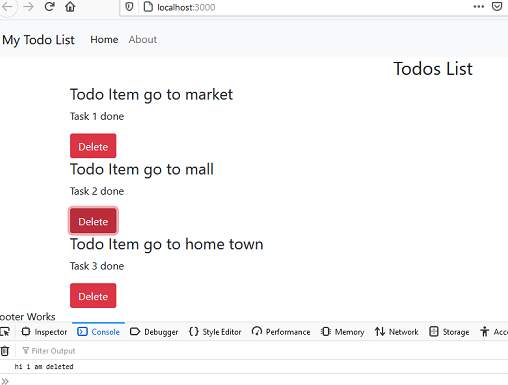
      <p>{todo.desc}</p>

      <button className="btn btn-danger" onClick={onDelete} >Delete</button>

    </div>

  )

}

****

**But now only message is showing, but not telling which todo item is deleting**

**So delete function we have to pass todo item no and get into delete function as argument**

**In App.js**

 const **onDelete=(todo)=>**

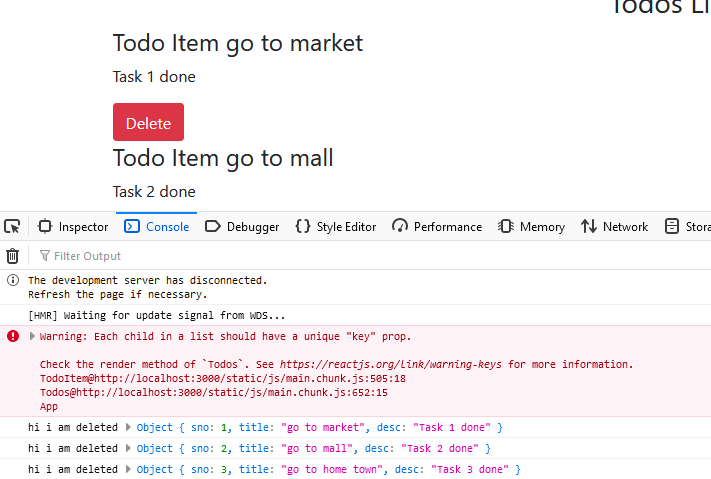
  {

    console.log("hi i am deleted", todo);

  }

**In TodoItem.js**

 <button className="btn btn-danger" onClick={onDelete**(todo)**} >Delete</button>

****

**Remove key issue**

import React from 'react'

import TodoItem from './TodoItem';

export default function Todos(props) {

  return (

    <div className="container">

     <h3 className="text-center">Todos List</h3>

     {props.todos.map((todo)=>{

       return  <TodoItem todo={todo} **key={todo.sno}** onDelete={props.onDelete} />

     })}

    </div>

  )

}

**Error will gone, but why all events is running automatically call without firing , showing in console**

**Bcoz when we pass argeumnet into function , that functiona will call automatically in rendering time, so we have solution is use arrow function instead of normal function call,**

**TodoItem.js**

**So arrow function will call with todo argument value when only button will click**

import React from 'react'

export default function TodoItem({todo,onDelete}) {

  return (

    <div>

      <h4>Todo Item {todo.title} </h4>

      <p>{todo.desc}</p>

      <button className="btn btn-danger" onClick={()=>onDelete(todo)} >Delete</button>

    </div>

  )

}

**Using arrow function we are passing only function, not calling before it was calling the function.**

Now console is showing the one todoitem is deleting but not updating the view

  const onDelete=(todo)=>

  {

    console.log("hi i am deleted", todo);

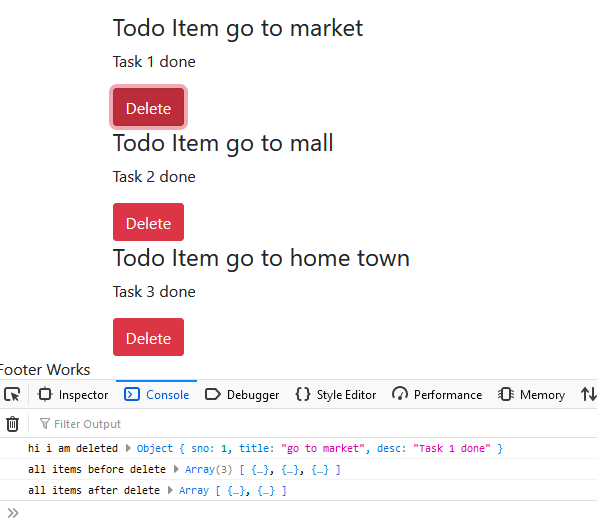
    console.log("all items before delete",todos);

    let index=todos.indexOf(todo);

    todos.splice(index,1);

    console.log("all items after delete",todos);

  }



But UI is not updated, for this we will use use State

201

[React hooks](https://reactjs.org/docs/hooks-overview.html) are a new way to access the core features of react such as state without having to use classes,

useState should be used only inside functional components.

useState() is a React hook. Hooks make possible to use state and mutability inside function components

useState is one of the hooks available in React v16.8.0. It basically lets you turn your otherwise non-stateful/functional components to one that can have its own state.

Now we have to convert Todos array into useState function, Means in App.js

 let todos=[]

const [todos, setTodos] = useState(true);

  //todos array which have each todoitem

  const [todos,setTodos]=useState([

    {

    sno:1,

    title:"go to market",

    desc: "Task 1 done"

  },

  {

    sno:2,

    title:"go to mall",

    desc: "Task 2 done"

  },

  {

    sno:3,

    title:"go to home town",

    desc: "Task 3 done"

  }

]);

setTodos is function which will update each Todoitem

  const onDelete=(todo)=>

  {

    console.log("hi i am deleted", todo); // is obj of click button of each todo items value

    /\*console.log("all items before delete",todos);

    let index=todos.indexOf(todo);

    todos.splice(index,1);

    console.log("all items after delete",todos);\*/

    /\* above code will not work in reactwe have to use setTodos function for update\*/

    setTodos(todos.filter((todoitem)=>{

      return todoitem!=todo;

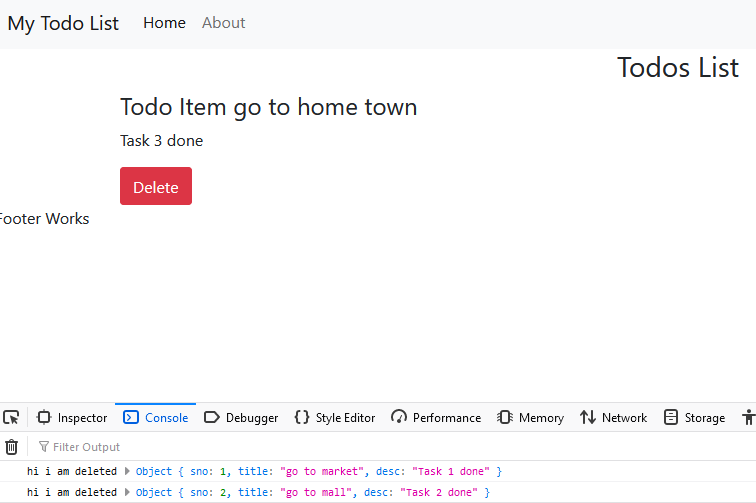
    }));

  }

**Where filter is high order function which return filter value based on return condition**

**Todo which item we are deleting having all values in a object**

**Todoitem is- each value in array of Todolist**

****

**If we deleted all TodoItem from pages, show message should show, no TodoItem is left**

**Todos.js**

import React from 'react'

import TodoItem from './TodoItem';

export default function Todos(props) {

  return (

    <div className="container">

     <h3 className="text-center">Todos List</h3>

**//if else applied**

**{props.todos.length===0 ?**

**"No Todo Item Left" :**

                            props.todos.map((todo)=>{

                            return  <TodoItem todo={todo} key={todo.sno} onDelete={props.onDelete} />

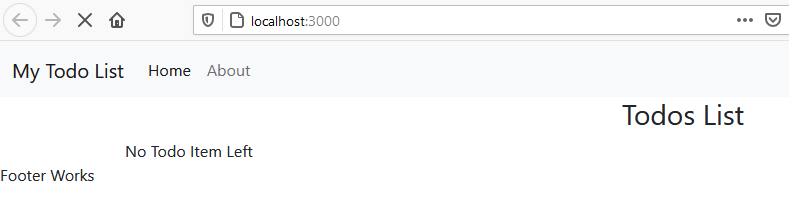
                                                   })

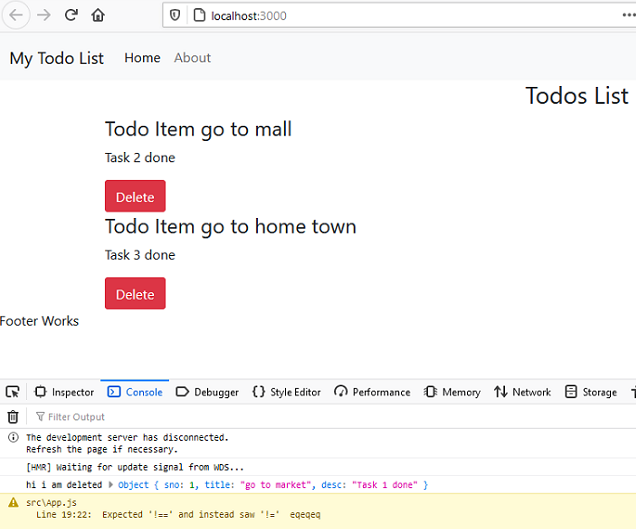
**}**

    </div>

  )

}

****

****

**How to add style in footer**

import React from 'react'

export default function Footer() {

  let footerStyle={

      position: "absolute",

      top:"90vh",

      width: "100%"

  }

  return (

    <footer className="bg-dark text-light py-3" style={footerStyle}>

     <p className="text-center"> Footer Works</p>

    </footer>

  )

}

**Now we will add functionality to add TodoItem in list**

[**https://getbootstrap.com/docs/4.0/components/forms/**](https://getbootstrap.com/docs/4.0/components/forms/)

**Create file or component into MyComponents folder AddTodoItem.js**

import React from 'react'

const AddTodoItem = () => {

  return (

    <div>

    </div>

  )

}

export default AddTodoItem

**add html form into this component**

import React from 'react'

const AddTodoItem = () => {

    return (

        <div>

            <form>

                <div className="form-group">

                    <label for="exampleInputEmail1">Email address</label>

                    <input type="email" className="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter email" />

                    <small id="emailHelp" className="form-text text-muted">We'll never share your email with anyone else.</small>

                </div>

                <div className="form-group">

                    <label for="exampleInputPassword1">Password</label>

                    <input type="password" className="form-control" id="exampleInputPassword1" placeholder="Password" />

                </div>

                <div className="form-check">

                    <input type="checkbox" className="form-check-input" id="exampleCheck1" />

                    <label className="form-check-label" for="exampleCheck1">Check me out</label>

                </div>

                <button type="submit" className="btn btn-primary">Submit</button>

            </form> </div>

    )} export default AddTodoItem

**Now we will call this component into App.js main component**

import Header from './MyComponents/Header';

import Footer from './MyComponents/Footer';

import Todos from './MyComponents/Todos';

**import AddTodoItem from './MyComponents/AddTodoItem';**

import React,{useState} from 'react';

function App() {

  const onDelete=(todo)=>

  {

    console.log("hi i am deleted", todo);

    /\*console.log("all items before delete",todos);

    let index=todos.indexOf(todo);

    todos.splice(index,1);

    console.log("all items after delete",todos);\*/

    /\* above code will not work in reactwe have to use setTodos function for update\*/

    setTodos(todos.filter((todoitem)=>{

      return todoitem!==todo;

    }));

  }

  //todos array which have each todoitem

  const [todos,setTodos]=useState([

    {

    sno:1,

    title:"go to market",

    desc: "Task 1 done"

  },

  {

    sno:2,

    title:"go to mall",

    desc: "Task 2 done"

  },

  {

    sno:3,

    title:"go to home town",

    desc: "Task 3 done"

  }

]);

  return (

    <>

    <Header title="My Todo List" />

**<AddTodoItem />**

            <Todos todos={todos} onDelete={onDelete} />

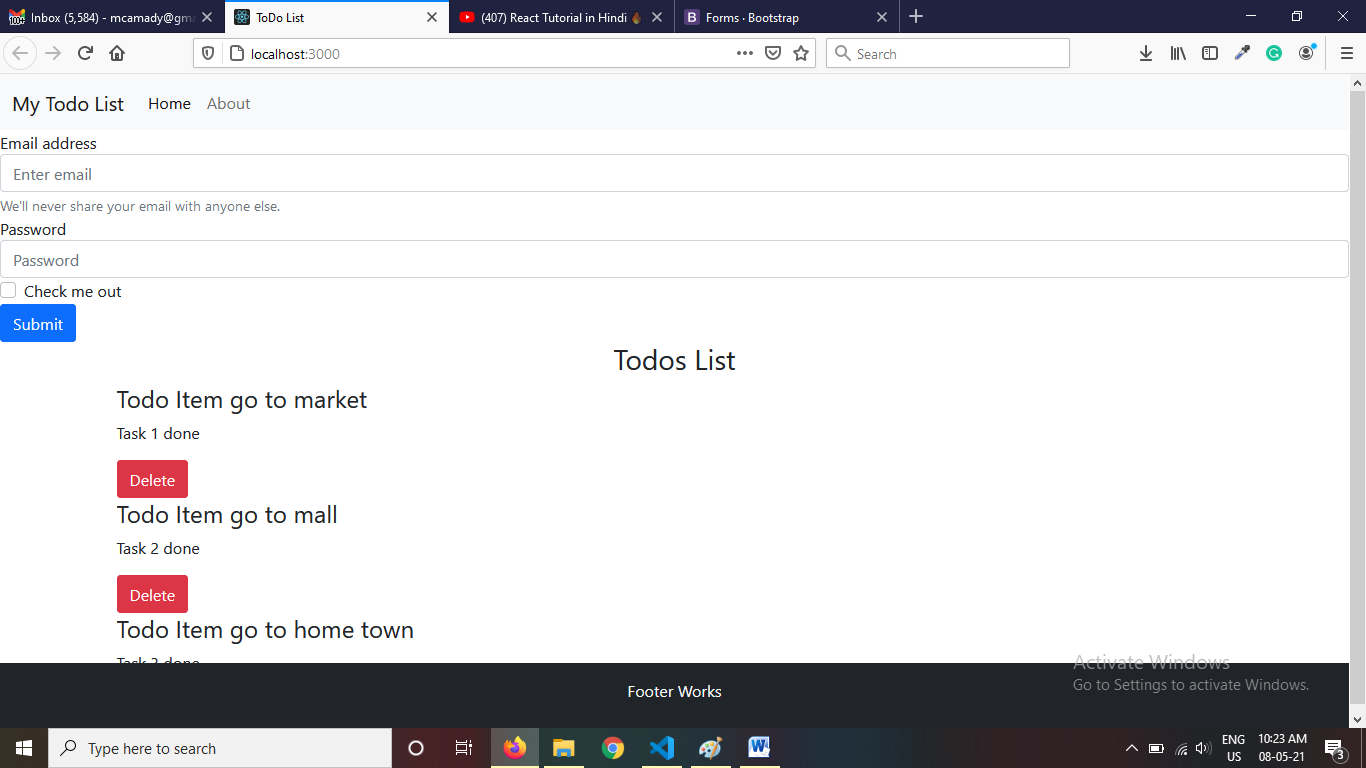
    <Footer/>

    </>

  );

}

export default App;



**My-3 – margin 3px in y axis**

**Lets update addtodoitem.js form**

import React from 'react'

const AddTodoItem = () => {

    return (

        <div className="container my-3">

            <h3>Add a New Todo Item</h3>

            <form>

                <div className="form-group">

                    <label for="title">Todo Title</label>

                    <input type="text" className="form-control" id="title" aria-describedby="title" placeholder="Enter title" />

                </div>

                <div className="form-group">

                    <label for="desc">Description</label>

                    <input type="text" className="form-control" id="desc" placeholder="Description" />

                </div>

                <button type="submit" className="btn btn-sm btn-success">Add Todo Item</button>

            </form>

        </div>

    )

}

export default AddTodoItem

**Now we have to use this form to get title and description to add into previous static state values**

**But before the state update, we have to assign initial values to each form element**

import React from 'react'

const AddTodoItem = () => {

**const onSubmit=()=>{**

**alert("form submitted");**

**}**

    return (

        <div className="container my-3">

            <h3>Add a New Todo Item</h3>

**<form onSubmit={onSubmit}>**

                <div className="form-group">

                    <label for="title">Todo Title</label>

                    <input type="text" className="form-control" id="title" aria-describedby="title" placeholder="Enter title" />

                </div>

                <div className="form-group">

                    <label for="desc">Description</label>

                    <input type="text" className="form-control" id="desc" placeholder="Description" />

                </div>

                <button type="submit" className="btn btn-sm btn-success">Add Todo Item</button>

            </form>

        </div>

    )

}

export default AddTodoItem

**but before creation of submit we need to initialize the values of title and description for useState()**

**import React, { useState } from 'react'**

const AddTodoItem = () => {

**const [title,setTitle]=useState("");**

**const [desc,setDesc]=useState("");**

    const onSubmit=()=>{

        alert("form submitted");

    }

    return (

        <div className="container my-3">

            <h3>Add a New Todo Item</h3>

            <form onSubmit={onSubmit}>

                <div className="form-group">

                    <label for="title">Todo Title</label>

                    <input type="text" className="form-control" id="title" aria-describedby="title" placeholder="Enter title" />

                </div>

                <div className="form-group">

                    <label for="desc">Description</label>

                    <input type="text" className="form-control" id="desc" placeholder="Description" />

                </div>

                <button type="submit" className="btn btn-sm btn-success">Add Todo Item</button>

            </form>

        </div>

    )

}

export default AddTodoItem

**so we have link these value of title and description into form**

<input type="text" **value={title}** className="form-control" id="title" placeholder="Enter title" />

<input type="text" **value={desc}** className="form-control" id="desc" placeholder="Description" />

**now we cant’ fill the form values bcoz its’ blank set in states**

**How to fix this issue to fill form values we have to use onChnage on each form element**

**onChange={(e)=>{  setTitle(e.target.value)  }}**

<input type="text" value={title} onChange={(e)=>{

                        setTitle(e.target.value)

                    }} className="form-control" id="title" placeholder="Enter title" />

 <input type="text" onChange={(e)=>{setDesc(e.target.value)}} value={desc} className="form-control" id="desc" placeholder="Description" />

**Now we can change values into the form, it’s means which we are writing into form field they are setting into state only, not adding into TODo task items array**

**How to stop page reload on submit**

    const onSubmit=(e)=>{

        e.preventDefault();

        alert("form submitted");

    }

    const onSubmit=(e)=>{

        e.preventDefault();

        //check values are not empty

        if(!title || !desc)

        {

            alert("values are blank");

        }

        else

        {

            alert("form submitted");

        } }

**Now we are getting form values, but we donot have array of todoitems on this component**

**For this we have to pull todos array values from app.js into addtodoitem.js page**

**For this we have to define function into APP.js bcoz it’s have values of TodoItesm array**

**And this function we have to pass as props into addtodoitem jsx element in app.js**

**Like we have written function for delete todoitem in app.js**

 const **addTodosingleitem**=()=>

  {

    console.log("todo item is going to add");

  }

  <AddTodoItem **addTodosingleitem={addTodosingleitem}** />

Now we have passed function **addTodosingleitem as props in addtodoitem.js**

**We can receive this function as like AddTodoItem.js**

import React, { useState } from 'react'

const AddTodoItem = (**props**) => {

    const [title, setTitle] = useState("");

    const [desc, setDesc] = useState("");

    const onSubmit = (e) => {

        e.preventDefault();

        //check values are not empty

        if (!title || !desc) {

            alert("values are blank");

        }

        else {

**props.addTodosingleitem(title, desc);**

            alert("form submitted");

        }

    }

    return (

        <div className="container my-3">

            <h3>Add a New Todo Item</h3>

            <form onSubmit={onSubmit}>

                <div className="form-group">

                    <label for="title">Todo Title</label>

                    <input type="text" value={title} onChange={(e) => {

                        setTitle(e.target.value)

                    }} className="form-control" id="title" placeholder="Enter title" />

                </div>

                <div className="form-group">

                    <label for="desc">Description</label>

                    <input type="text" onChange={(e) => { setDesc(e.target.value) }} value={desc} className="form-control" id="desc" placeholder="Description" />

                </div>

                <button type="submit" className="btn btn-sm btn-success">Add Todo Item</button>

            </form>

        </div>

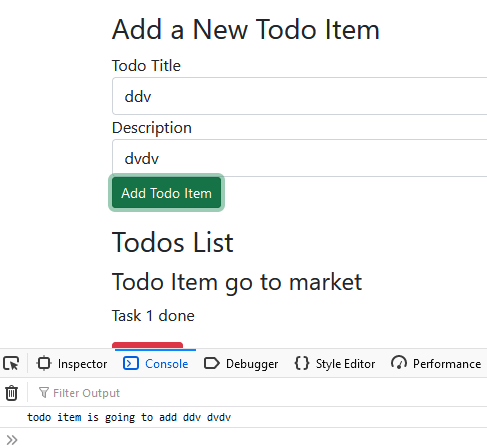
    )

}

export default AddTodoItem

**lets update addTodosingleitem function into app.js**

because we are sending two values like title and description



 const addTodosingleitem=(title,desc)=>

  {

    console.log("todo item is going to add",title,desc);

  }

Now we are able to get values of title and desc, but sno no. is missing and it should be in object values, so we need to convert into one object

 const addTodosingleitem=(title,desc)=>

  {

    console.log("todo item is going to add",title,desc);

    let sno=todos[todos.length-1].sno+1;

    const mytod={

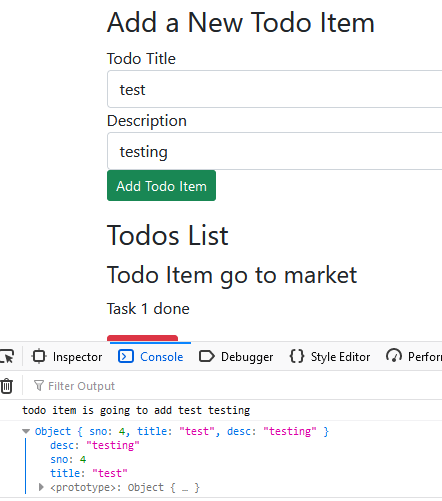
      sno:sno,

      title:title,

      desc:desc

    }

}



 const addTodosingleitem=(title,desc)=>

  {

    console.log("todo item is going to add",title,desc);

    let sno=todos[todos.length-1].sno+1;

    const mytodo={

      sno:sno,

      title:title,

      desc:desc

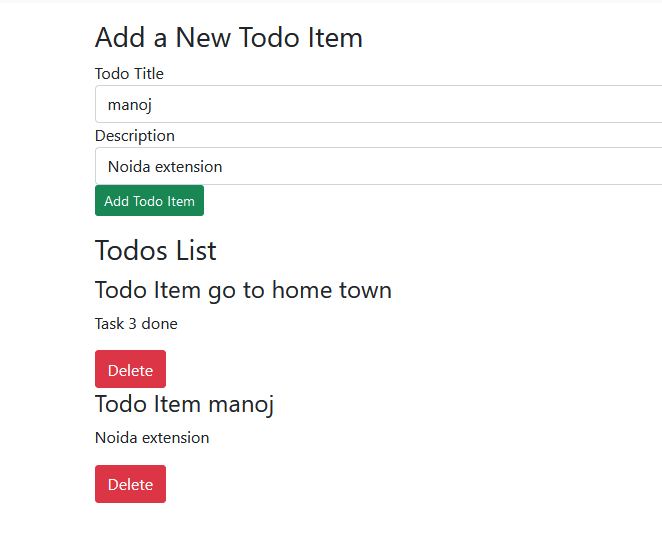
    }

    console.log(mytodo);

    //we used spread operator merge two array into one, because using spread we have to pass three dots in first argument, but in rest we have to pass into last argument

    setTodos([...todos,mytodo]);

  }

****

**But if we have no single todoitem this will generate error**

let sno=todos[todos.length-1].sno+1;

TypeError: todos[(todos.length - 1)] is undefined

    let sno = 0;

    if (todos.length == 0) {

      sno = 0;

    }

    else {

      sno = todos[todos.length - 1].sno + 1;

    }

Complete App.js

import Header from './MyComponents/Header';

import Footer from './MyComponents/Footer';

import Todos from './MyComponents/Todos';

import AddTodoItem from './MyComponents/AddTodoItem';

import React, { useState } from 'react';

function App() {

  const onDelete = (todo) => {

    console.log("hi i am deleted", todo);

    /\*console.log("all items before delete",todos);

    let index=todos.indexOf(todo);

    todos.splice(index,1);

    console.log("all items after delete",todos);\*/

    /\* above code will not work in reactwe have to use setTodos function for update\*/

    setTodos(todos.filter((todoitem) => {

      return todoitem !== todo;

    }));

  }

  const addTodosingleitem = (title, desc) => {

    console.log("todo item is going to add", title, desc);

    let sno = 0;

    if (todos.length == 0) {

      sno = 0;

    }

    else {

      sno = todos[todos.length - 1].sno + 1;

    }

    const mytodo = {

      sno: sno,

      title: title,

      desc: desc

    }

    console.log(mytodo);

    //we used spread operator merge two array into one

    setTodos([...todos, mytodo]);

  }

  //todos array which have each todoitem

  const [todos, setTodos] = useState([

    {

      sno: 1,

      title: "go to market",

      desc: "Task 1 done"

    },

    {

      sno: 2,

      title: "go to mall",

      desc: "Task 2 done"

    },

    {

      sno: 3,

      title: "go to home town",

      desc: "Task 3 done"

    }

  ]);

  return (

    <>

      <Header title="My Todo List" />

      <AddTodoItem addTodosingleitem={addTodosingleitem} />

      <Todos todos={todos} onDelete={onDelete} />

      <Footer />

    </>

  );

}

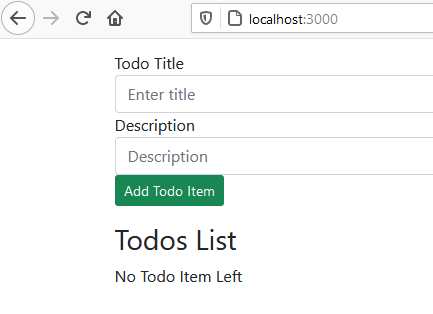
export default App;

**Local Storage System for showing saved todoitems and delete**

**First make it blank array of todos**

  //todos array which have each todoitem

  const [todos, setTodos] = useState([]);

****

**We set todos in localstorage in the time of add & delete, but we need list of todo items first time page load, how many we saved in todos item in localstorage**

**For this we have to set initial value of todos items localstorgae which will get and assign to todos items array;**

**Now we are getting todos list, but facing one proplem, when we are updating the todos list using setTodos,**

setTodos([...todos, mytodo]);

**if wiil not immedidatly update the values and setting**  localStorage.setItem("todos", JSON.stringify(todos));

**So application works weird so we will use useEffect hook**

**App.js**

import Header from './MyComponents/Header';

import Footer from './MyComponents/Footer';

import Todos from './MyComponents/Todos';

import AddTodoItem from './MyComponents/AddTodoItem';

import React, { useEffect, useState } from 'react';

function App() {

**let initialtodos = "";**

**if (localStorage.getItem("todos") === "null") {**

**initialtodos = [];**

**}**

**else {**

**initialtodos = JSON.parse(localStorage.getItem("todos"));**

**}**

  const onDelete = (todo) => {

    console.log("hi i am deleted", todo);

    /\*console.log("all items before delete",todos);

    let index=todos.indexOf(todo);

    todos.splice(index,1);

    console.log("all items after delete",todos);\*/

    /\* above code will not work in reactwe have to use setTodos function for update\*/

    setTodos(todos.filter((todoitem) => {

      return todoitem !== todo;

    }));

    // localStorage.setItem("todos", JSON.stringify(todos));

  }

  const addTodosingleitem = (title, desc) => {

    console.log("todo item is going to add", title, desc);

    let sno = 0;

    if (todos.length == 0) {

      sno = 0;

    }

    else {

      sno = todos[todos.length - 1].sno + 1;

    }

    const mytodo = {

      sno: sno,

      title: title,

      desc: desc

    }

    console.log(mytodo);

    //we used spread operator merge two array into one

    setTodos([...todos, mytodo]);

    //localStorage.setItem("todos", JSON.stringify(todos));

  }

  //todos array which have each todoitem

  const [**todos**, setTodos] = useState(initialtodos);

**useEffect(**

**() => {**

**localStorage.setItem("todos", JSON.stringify(todos));**

**},**

**[todos]**

**);**

  return (

    <>

      <Header title="My Todo List" />

      <AddTodoItem addTodosingleitem={addTodosingleitem} />

      <Todos todos={todos} onDelete={onDelete} />

      <Footer />

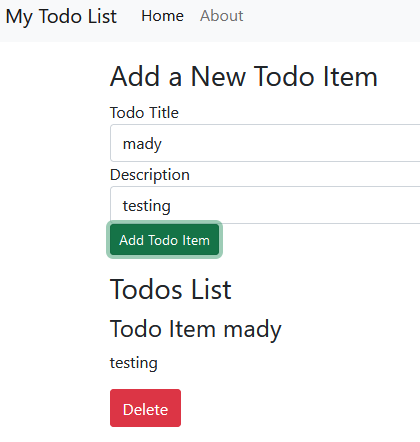
    </>

  );

}

export default App;

**now everything working fine, but after adding the values form values still showing**

****

**AddTodoItem.js**

import React, { useState } from 'react'

const AddTodoItem = (props) => {

    const [title, setTitle] = useState("");

    const [desc, setDesc] = useState("");

    const onSubmit = (e) => {

        e.preventDefault();

        //check values are not empty

        if (!title || !desc) {

            alert("values are blank");

        }

        else {

            props.addTodosingleitem(title, desc);

            alert("form submitted");

**setTitle("");**

**setDesc("");**

        }

    }

    return (

        <div className="container my-3">

            <h3>Add a New Todo Item</h3>

            <form onSubmit={onSubmit}>

                <div className="form-group">

                    <label htmlFor="title">Todo Title</label>

                    <input type="text" value={title} onChange={(e) => {

                        setTitle(e.target.value)

                    }} className="form-control" id="title" placeholder="Enter title" />

                </div>

                <div className="form-group">

                    <label htmlFor="desc">Description</label>

                    <input type="text" onChange={(e) => { setDesc(e.target.value) }} value={desc} className="form-control" id="desc" placeholder="Description" />

                </div>

                <button type="submit" className="btn btn-sm btn-success">Add Todo Item</button>

            </form>

        </div>

    )

}

export default AddTodoItem

**React Router rendering new page without reload the page**

**Npm I react-router-dom**

<https://reactrouter.com/web/guides/quick-start>

**App.js**

import {

BrowserRouter as Router,

Switch,

Route,

Link

} from "react-router-dom";

Our app should open and close with router

<Router>

</Router>

Means in app.js

    <>

    <Router>

      <Header title="My Todo List" />

      <AddTodoItem addTodosingleitem={addTodosingleitem} />

      <Todos todos={todos} onDelete={onDelete} />

      <Footer />

      </Router>

    </>

Now we have to define which path will open which route or page

{/\* A <Switch> looks through its children <Route>s and

renders the first one that matches the current URL. \*/}

<Switch>

<Route path="/about">

<About />

</Route>

<Route path="/users">

<Users />

</Route>

<Route path="/">

<Home />

</Route>

</Switch>

<>

    <Router>

      //common component header

      <Header title="My Todo List" />

//only when home page or /

      <AddTodoItem addTodosingleitem={addTodosingleitem} />

      <Todos todos={todos} onDelete={onDelete} />

      //common component footer

      <Footer />

      </Router>

    </>

**App.js**

import Header from './MyComponents/Header';

import Footer from './MyComponents/Footer';

import Todos from './MyComponents/Todos';

import AddTodoItem from './MyComponents/AddTodoItem';

import React, { useEffect, useState } from 'react';

import {

  BrowserRouter as Router,

  Switch,

  Route,

  Link

} from "react-router-dom";

import About from './MyComponents/About';

function App() {

  let initialtodos = "";

  if (localStorage.getItem("todos") === "null") {

    initialtodos = [];

  }

  else {

    initialtodos = JSON.parse(localStorage.getItem("todos"));

  }

  const onDelete = (todo) => {

    console.log("hi i am deleted", todo);

    /\*console.log("all items before delete",todos);

    let index=todos.indexOf(todo);

    todos.splice(index,1);

    console.log("all items after delete",todos);\*/

    /\* above code will not work in reactwe have to use setTodos function for update\*/

    setTodos(todos.filter((todoitem) => {

      return todoitem !== todo;

    }));

    // localStorage.setItem("todos", JSON.stringify(todos));

  }

  const addTodosingleitem = (title, desc) => {

    console.log("todo item is going to add", title, desc);

    let sno = 0;

    if (todos.length == 0) {

      sno = 0;

    }

    else {

      sno = todos[todos.length - 1].sno + 1;

    }

    const mytodo = {

      sno: sno,

      title: title,

      desc: desc

    }

    console.log(mytodo);

    //we used spread operator merge two array into one

    setTodos([...todos, mytodo]);

    //localStorage.setItem("todos", JSON.stringify(todos));

  }

  //todos array which have each todoitem

  const [todos, setTodos] = useState(initialtodos);

  useEffect(

    () => {

      localStorage.setItem("todos", JSON.stringify(todos));

    },

    [todos]

  );

  return (

    <>

      <Router>

        <Header title="My Todo List" />

        <Switch>

          <Route path="/">

            <AddTodoItem addTodosingleitem={addTodosingleitem} />

            <Todos todos={todos} onDelete={onDelete} />

          </Route>

          <Route path="/about">

            <About />

          </Route>

        </Switch>

        <Footer />

      </Router>

    </>

  );

}

export default App;

**http://localhost:3000/**

[**http://localhost:3000/about**](http://localhost:3000/about)

**but this is not working**

**because we have to tell route which component will render on this path**

import Header from './MyComponents/Header';

import Footer from './MyComponents/Footer';

import Todos from './MyComponents/Todos';

import AddTodoItem from './MyComponents/AddTodoItem';

import React, { useEffect, useState } from 'react';

import {

  BrowserRouter as Router,

  Switch,

  Route,

  Link

} from "react-router-dom";

import About from './MyComponents/About';

function App() {

  let initialtodos = "";

  if (localStorage.getItem("todos") === "null") {

    initialtodos = [];

  }

  else {

    initialtodos = JSON.parse(localStorage.getItem("todos"));

  }

  const onDelete = (todo) => {

    console.log("hi i am deleted", todo);

    /\*console.log("all items before delete",todos);

    let index=todos.indexOf(todo);

    todos.splice(index,1);

    console.log("all items after delete",todos);\*/

    /\* above code will not work in reactwe have to use setTodos function for update\*/

    setTodos(todos.filter((todoitem) => {

      return todoitem !== todo;

    }));

    // localStorage.setItem("todos", JSON.stringify(todos));

  }

  const addTodosingleitem = (title, desc) => {

    console.log("todo item is going to add", title, desc);

    let sno = 0;

    if (todos.length == 0) {

      sno = 0;

    }

    else {

      sno = todos[todos.length - 1].sno + 1;

    }

    const mytodo = {

      sno: sno,

      title: title,

      desc: desc

    }

    console.log(mytodo);

    //we used spread operator merge two array into one

    setTodos([...todos, mytodo]);

    //localStorage.setItem("todos", JSON.stringify(todos));

  }

  //todos array which have each todoitem

  const [todos, setTodos] = useState(initialtodos);

  useEffect(

    () => {

      localStorage.setItem("todos", JSON.stringify(todos));

    },

    [todos]

  );

  return (

    <>

      <Router>

        <Header title="My Todo List" />

        <Switch>

          <Route path="/" render={() => {

            return (<>

              <AddTodoItem addTodosingleitem={addTodosingleitem} />

              <Todos todos={todos} onDelete={onDelete} />

            </>)

          }}>

          </Route>

          <Route path="/about">

            <About />

          </Route>

        </Switch>

        <Footer />

      </Router>

    </>

  );

}

export default App;

**but still different pages not showing so we have to use** exact path, now its working

**now we are not using Link in App.js so remove it here and use into Header.js**

import {Link} from "react-router-dom";

**convert all <a into <Link**

**href to Link**

 <Link className="nav-link active" aria-current="page" to="/">Home</Link>

<Link className="nav-link" to="/about">About</Link>

**Lets create build, before build creation make sure no bugs in console**

**Npm run build**

**C:\my-app\build**

**Install live server extension**