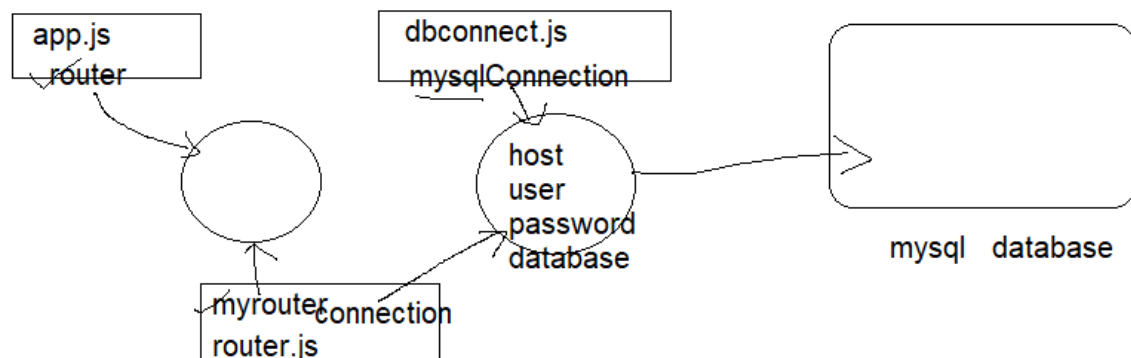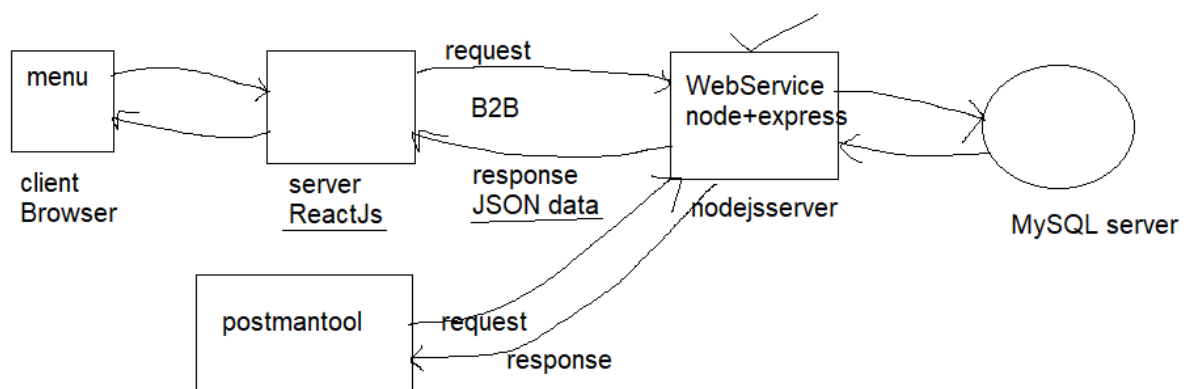Whenever you will receive the data in the web application, to retrieve the data

In get method data will be received in the form of query string http://localhost:3001/insertproduct?pid=12&pname=lays&qty=34&price=345	to retrieve it in the application req.query.pid req.query.pname req.query.qty req.query.price
In post method data will be received in the request body http://localhost:3001/insertproduct request body →pid=12&pname=lays&qty=34&price=345	to retrieve it in the application req.body.pid req.body.pname req.body.qty req.body.price
urlpattern ---->/insertproduct/:id http://localhost:3001/deleteproduct/456	to retrieve the data req.params.id

4 Tier application

Web service url should not contain verbs, It should contain only nouns

GET—Read data	/products –return all objects /products/:id --return single object
POST---Insert/create data	/products/:id the data should come via request body
PUT---update data	/products/:id the data should come via request body
DELETE -→ delete data	/products/:id

Reactjs

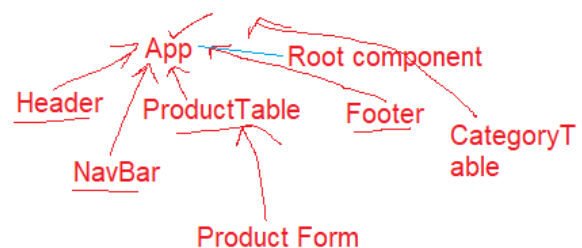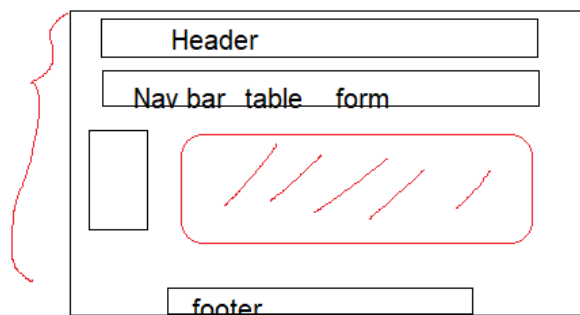It is specially designed for front end application, in react we use Javascript language for writing code

to create a simple react application, we use a tool create-react-app

to install create-react-app and create a react application

1. npm install create-react-app -g
2. create-react-app myfirst

to create a react application, without installing create-react-app
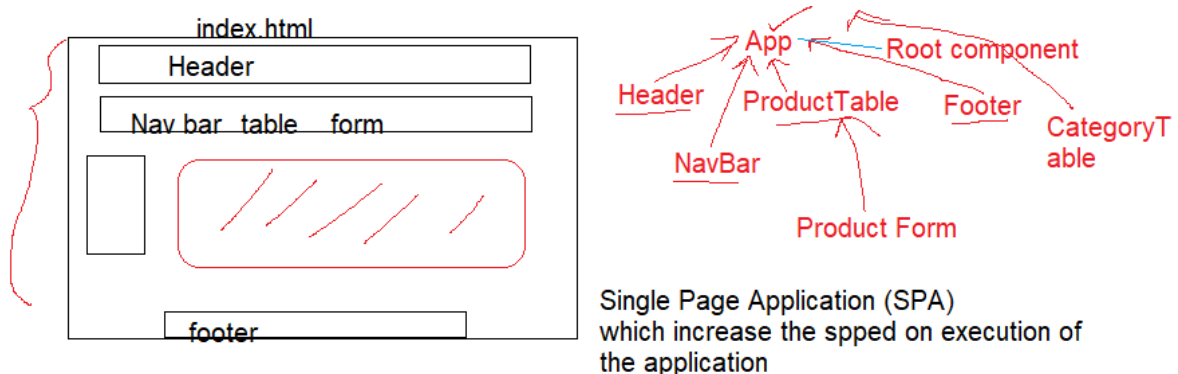
1. npx create-react-app myfirst

Component

components are of 2 types
1. class component (stateful component)
   a. In this component this. state built-in object can be used to store data
   b. we need to write render function, it retuns JSX(javascript extension)
2. function component(stateless component)
   a. in the component we cannot use this.state built in object
   b. in new version the useState hook allows you to use state object indirectly

function component	class component

const mycomponent=()=>{
return(
 hello world!!
)
}
export default mycomponent;
	import React,{Component} from 'react';
	class MyComponent extends React.Component{
	render(){
	    return(
	 hello world!!
	)
	}
	}
	
	export default MyCompnent;

It is stateless component, and this.state is not allowed to use directly, but can be used indirectly with the help of hook useState	It is stateful component and allow to use this.state object
Lifecycle methods are not used in functional component, but in new version the useEffect hook behaves like lifecycle methods	Lifecycle methods can be used only in class component

babel compiler converts the jsx into javascript code

https://babeljs.io/repl#?browsers=defaults%2C%20not%20ie%2011%2C%20not%20ie_mob%2011&build=&builtIns=false&corejs=3.21&spec=false&loose=false&code_lz=FAHgJglgbgBBYF4BEBbAnpKSB8wYxAFcAbXffEYibAewDsBTEAeirPMuoBcB3Gltng5suACwBODJq2pCWJXC0y4gA&debug=false&forceAllTransforms=false&modules=false&shippedProposals=false&circleciRepo=&evaluate=false&fileSize=false&timeTravel=false&sourceType=module&lineWrap=true&presets=env%2Creact%2Cstage-2&prettier=false&targets=&version=7.24.7&externalPlugins=&assumptions=%7B%7D

React application name should contain only smallcase letters

and user defined components name should start with capital letter

if you use export default, then do not use {} while import, otherwise add it in {}

If any change in the value should get reflected in the output, then use the variable as state

to handle event , we use camel case

functional component	class component	
onClick={increament} --- function without passing value	onClick={this.increament}	
onClick={()=>{increament(5)}}---function with passing value	onClick={()=>{this.increament(5)}}	

const [cnt,setcnt]=useState(0);

state
  count
  setcnt