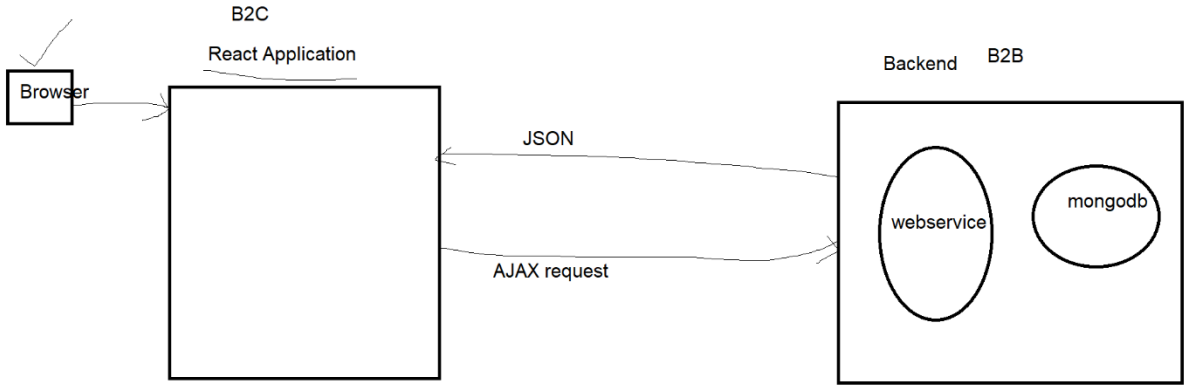
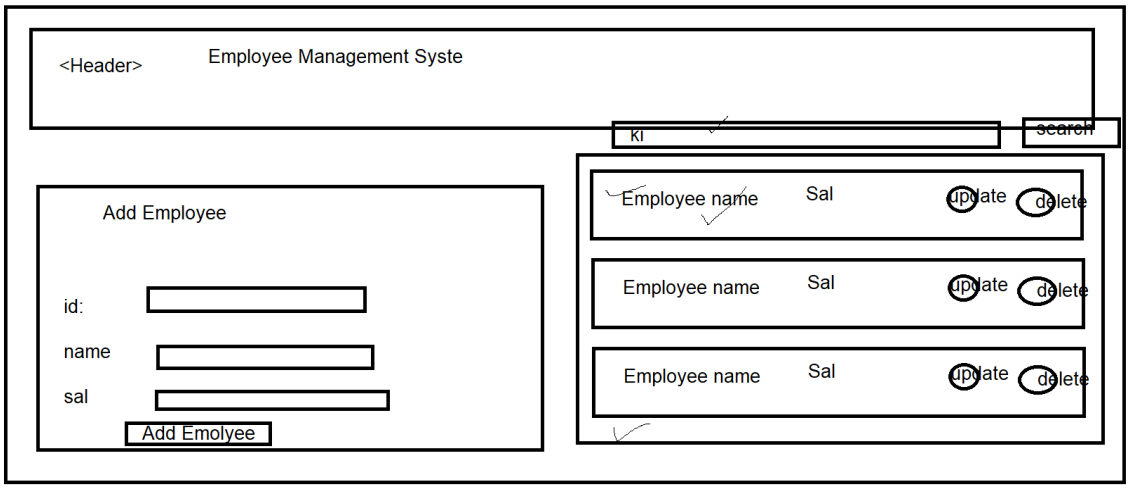
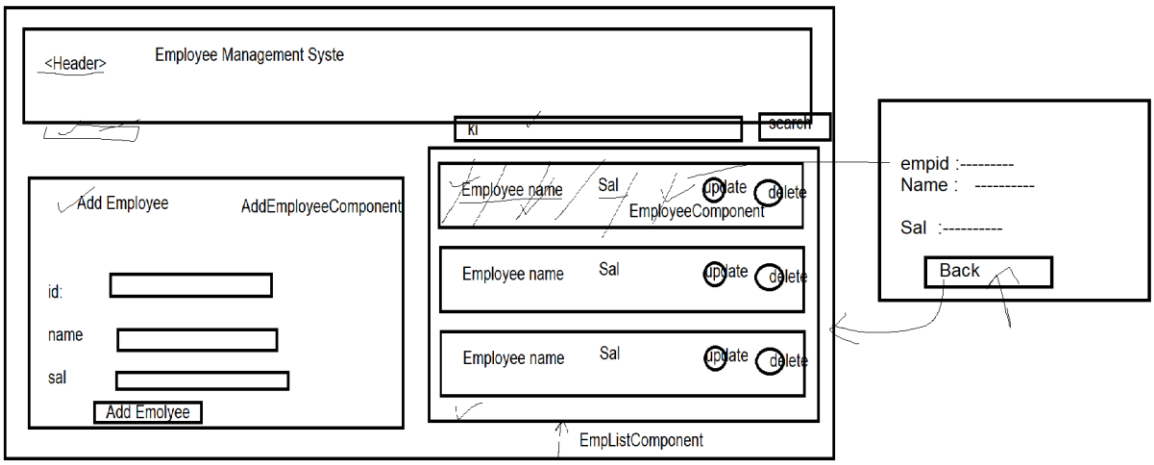




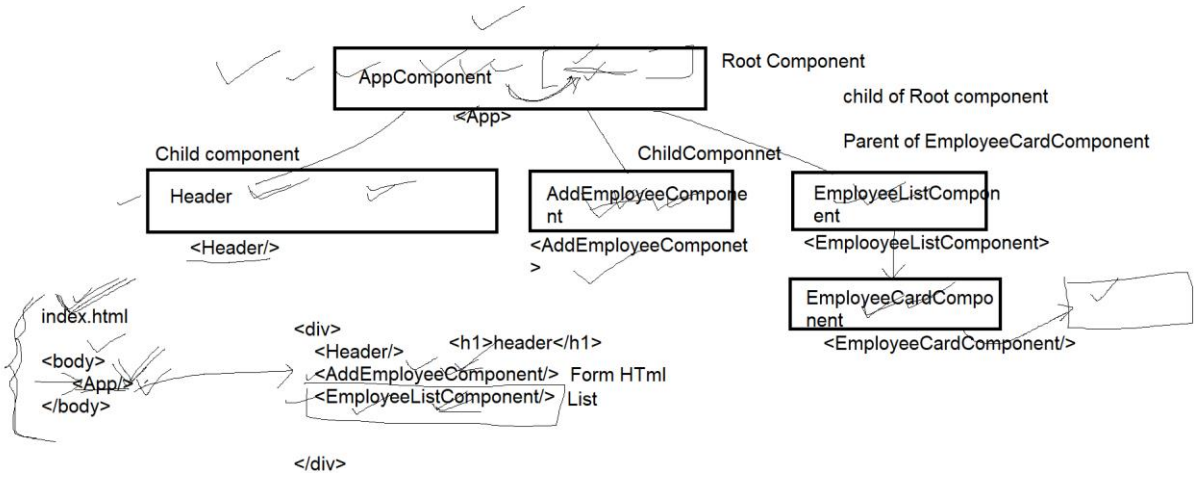


AppComponent ------- Root component

- What is ReactJS
  ReactJs is Library which helps you design view part of your application.
  and useful for designing B2C(Business to customer) application
  This is designed by Jordan Walke in 2011

  Step 1:
  Install nodejs
  create -react-app

  React application is formed by many components. and it helps us to design application as SPA(singlePage Application) using MVC architecture
  M ---- Model ---Data
  V--- view ------HTML o/p
  C- Controller which helps to communicate between view and model



  ReactJs application is formed by many components, each component is responsible for displaying small portion of the page. These are reusable code

  The components can be nested with another component to design complex views

- Why and How ReactJs works
  ReactJs increases the speed execution. It reduces the complexity by dividing complex page into small component so writing code becomes easy

  It uses Virtual DOM(Document Object Model)

  if you modify physical DOM then immediately page gets redrawn. which is time consuming and
  Hence ReactJs makes changes in memory copy of DOM(existing virtual DOM) ,

It will maintain a copy of existing DOM and it also create new copy by adding changes in virtual DOM
It compares Previous state and current changes

- How to use create-react-app to generate react application
  create-react-app is a CLI(command line Interface) facility to generate basic structure of Reactjs

  c:/system32> npm install -g create-react-app

  to create a simple react application, the react application name should not contain capital letter

  d:/ractdemos>create-react-app helloworldapp

  It will generate a new folder by name helloworldapp and will store all files in this folder

  To start the application
  d:/ractdemos>cd helloworldapp
  d:/ractdemos/hellowordapp>npm start
  It will start the application at port 3000

  After this to see the o/p
  open browser
  http://localhost:3000

  This folder contains index.html in folder public folder which has a div with id root

```
<body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
    <!--
      This HTML file is a template.
      If you open it directly in the browser, you will see an empty page.

      You can add webfonts, meta tags, or analytics to this file.
      The build step will place the bundled scripts into the <body> tag.

      To begin the development, run `npm start` or `yarn start`.
      To create a production bundle, use `npm run build` or `yarn build`.
    -->
  </body>
```

  the code will start running from index.js file in src folder

```
ReactDOM.render(
```

```
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);
```

this replace innerHTML property of div with id root in index html file by o/p of App component

App component is stored in src folder in App.js

```
import logo from './logo.svg';
import './App.css';

function App() {
    return (
       <div>
            <h1 > Hello world!! </h1>
            <h3 > Welcome to reactjs application </h3>
        </div>

    );

}

export default App;
```

JSX
<h1>Hello world</h1>

React code

React.component("h1",null,"Hello World!!")

JSX gets compiled by using javascript Babel compiler, and uses webpack to convert it into production ready application pack

index.html ----→ index.js---→ App.js

- What is JSX(javascript extension)
  It is javascript code which looks like html code
  In jsx you can add custom tags by creating components
  all custom components name should start with capital, because if tag starts with small case will be considered as standard html tag

- To install bootstrap in React application
  Steps to add bootstrap in react application

npm install react-bootstrap bootstrap@4.6 --save

To use various icons from boot strap library

- npm install react-bootstrap-icons –save

- add following entries in index.html after line 17

```
<script src="https://unpkg.com/react/umd/react.production.min.js"
crossorigin></script>

<script
  src="https://unpkg.com/react-dom/umd/react-dom.production.min.js"
  crossorigin></script>

<script
  src="https://unpkg.com/react-bootstrap@next/dist/react-
bootstrap.min.js"
  crossorigin></script>

<script>var Alert = ReactBootstrap.Alert;</script>
```

add following line in App.jsx

```
import 'bootstrap/dist/css/bootstrap.min.css';
```

add a button in app.jsx

- State
- Props
- difference between states and props
- Component communication

To create props demo

d:\reactdemos>create-react-app propsdemo

d\reactdemos>cd propsdemo

d:\reactdemos\propsdemo>

Install bootstrap into your application ---steps are given above

To add our own formatting

1. add style attribute and pass javascript object as value

```
import React from "react";
import './Header.css'

const Header=()=>{

    return(
        <div
style={{background:"blue",width:"600px",marginLeft:"300px",borderRadius
:"6px"}}>
            <h1 className="test">Employee Management System</h1>
        </div>
    );
}

export default Header;
```

2. create separate .css file and import it in .jsx file and add className attribute

```
import React from "react";
import './Header.css'

const Header=()=>{

    return(
        <div
style={{background:"blue",width:"600px",marginLeft:"300px",borderRadius:"6px"}
}>
            <h1 className="test">Employee Management System</h1>
        </div>
    );
}

export default Header;
```

```
otherwise use Bootstrap classes, you may use jumbletron

Heder.css file
```

```
.test {
    text-align: center;
    color: white;
    border: 2px solid red;
```

```
    border-radius: 6px;
    background-color: blue;
    width: 600px
}

@media only screen and (max-width:600px) {
    body {
        background-color: lightblue
    }
}
```

Add EmployeeList component

To find Bootstrap icons

https://icons.getbootstrap.com/

To pass data from Parent component to child component use props object