React Lab Assessment

Gopikrishnan P - 12218

Q1) **Software Licensing Management System** - Software licensing management is the process of tracking and documenting software used in a company.

<table>
<tr><th colspan="2">Project Name: Software Licensing Management System</th></tr>
<tr><th>What? 1</th><th>How? 2</th></tr>
<tr><td>

1. What are the modules are required?
Ans: A) Admin
B) Software license
C) Company
2. What are the functionalities required for Software Module?
Ans:
1. Admin can CRUD operation for Software license data
2. Admin verify the Software details for Add the new Software license
3. What are the functionalities required for Company Module?
Ans:
1. Admin can CRUD operation for Software data
2. Admin verify the company details for Add the new Company
4. What are the fields required to Add Sofware?
Ans:
softwareId, softwareName, softwareVersion, totalPurchased, managedInstallations
5. What are the fields required to Add Company?
Ans:
companyId, companyName, websiteLink, companyCode, phoneNo

</td><td>

1. Admin:
a) Admin Login:
Method 1: Admin can login using email id and Password**.**
Method 2: Admin can login using email id, phone number and password.
Method 3: Admin can login using username and password

2. Software:
a) ADD:
Method 1: Admin can add the new software using softwareId.
Method 2: Admin can add the new software using softwareName.
b) Delete:
Method 1: Admin can delete the software using softwareId.
Method 2: Admin can delete the software using softwareVersion.
c) Edit:
Method 1: Admin can edit the software using softwareId.
Method 2: Admin can edit the software using softwareVersion.

3.Company:
a) ADD:
Method 1: Admin can add the new company using companyId.
Method 2: Admin can add the new company using companyName.
b) Delete:
Method 1: Admin can delete the company using companyId.
Method 2: Admin can delete the company using companyCode.
c) Edit:
Method 1: Admin can edit the company using companyId.
Method 2: Admin can edit the company using companyCode.

</td></tr>
<tr><td>

1.Admin:
a) Admin Login:
Method 2: Admin can login using email id, phone number and password.
To verify the originality of the admin and it come with 2 step authentications.

2. Software:
a) ADD:
Method 1: Admin can add the new software using softwareId.
This softwareId only unique field in add new software
b) b) Delete:
Method 1: Admin can delete the software using softwareId.
This softwareId only unique field
c) Edit:
Method 1: Admin can edit the software using softwareId.
This softwareId only unique field in add new software

3. Company:
a) ADD:
Method 1: Admin can add the new company using companyId.
This companyId only unique field in add new software
b) b) Delete:
Method 1: Admin can delete the company using companyId.
This companyId only unique field
c) Edit:
Method 1: Admin can edit the company using companyId.
This companyId only unique field

</td><td>

1.Admin:

a) The admin can't be using mail id and username for the log in, because this person has more responsibility, so they need more security.

2. Software:

a) In adding new software, we can't use the software name as main field, it does not give the accurate result.
b) In deleting software, we can't use the software version as main field, it may have same version in various software
c) In editing software, we can't use the software version as main field, it may have same version in various software

3. Company:

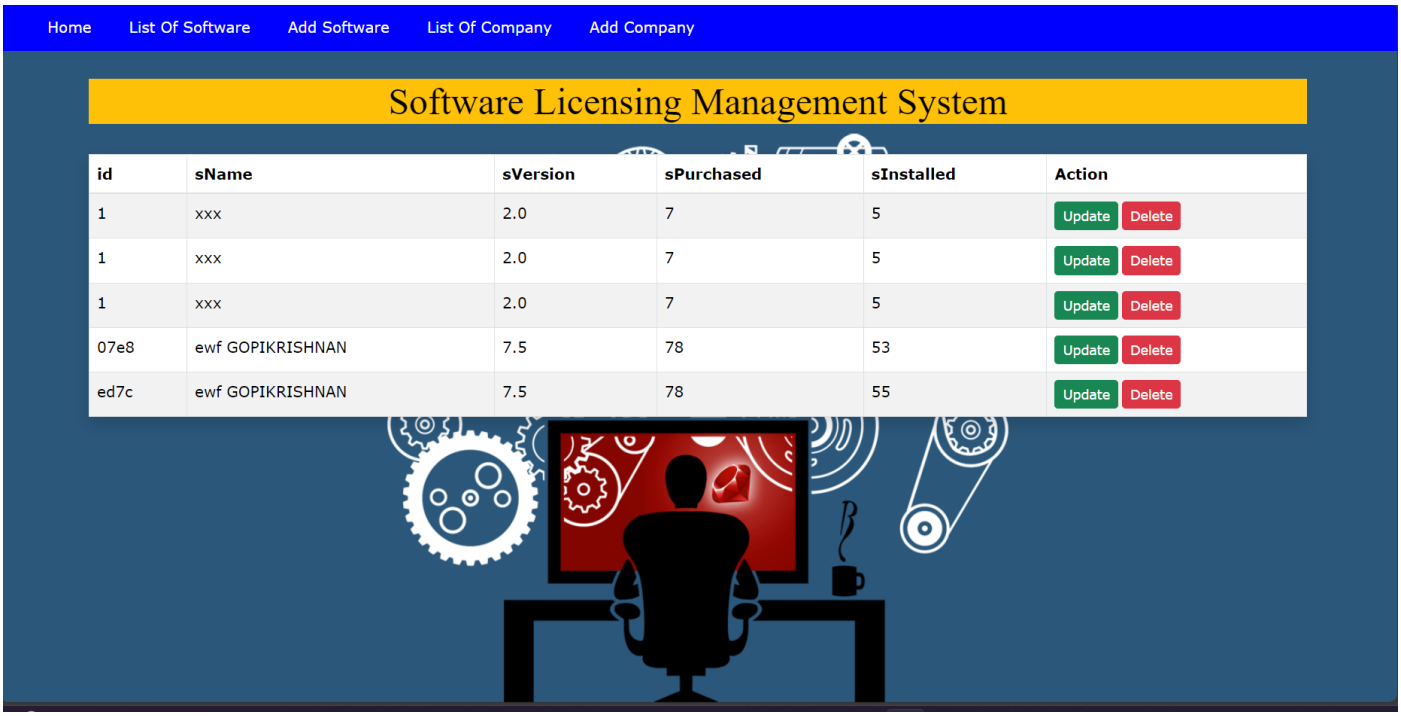
a) In adding new company, we can't use the company name as main field, it does not give the accurate result.
b) In deleting company, we can't use the company Code as main field, it may have same version in various software
c) In editing company, we can't use the company Code as main field, it may have same version in various software

</td></tr>
<tr><th>Why? 3</th><th>Why Not? 4</th></tr>
</table>

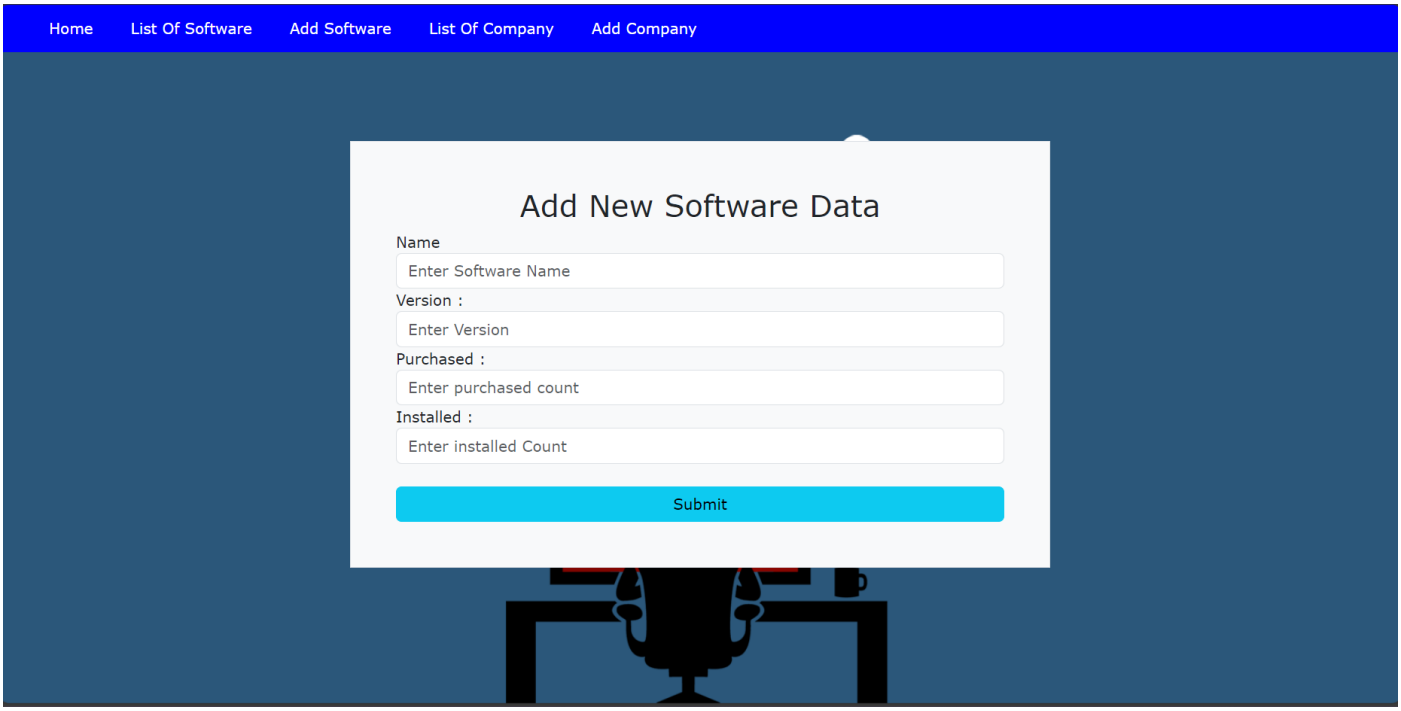
Screen Shots:

Home:

List Of Software:

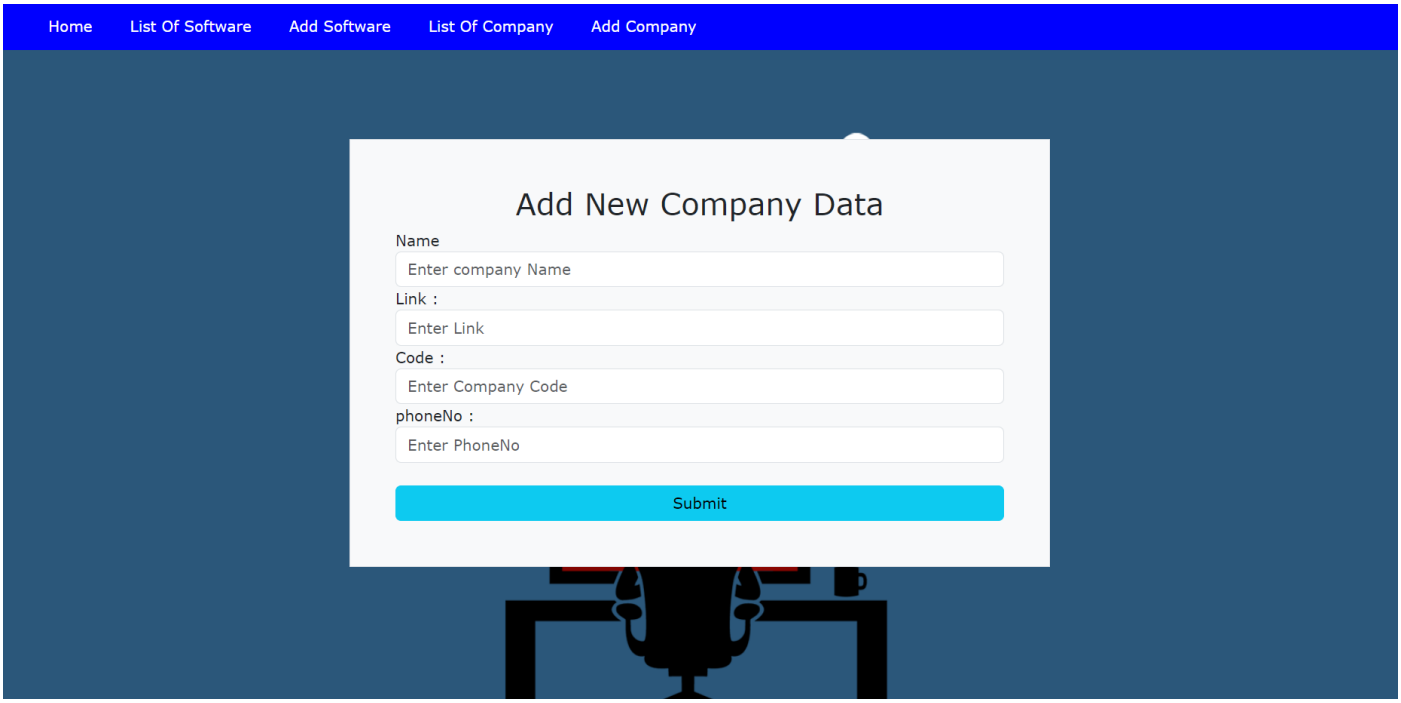


Add New Software:

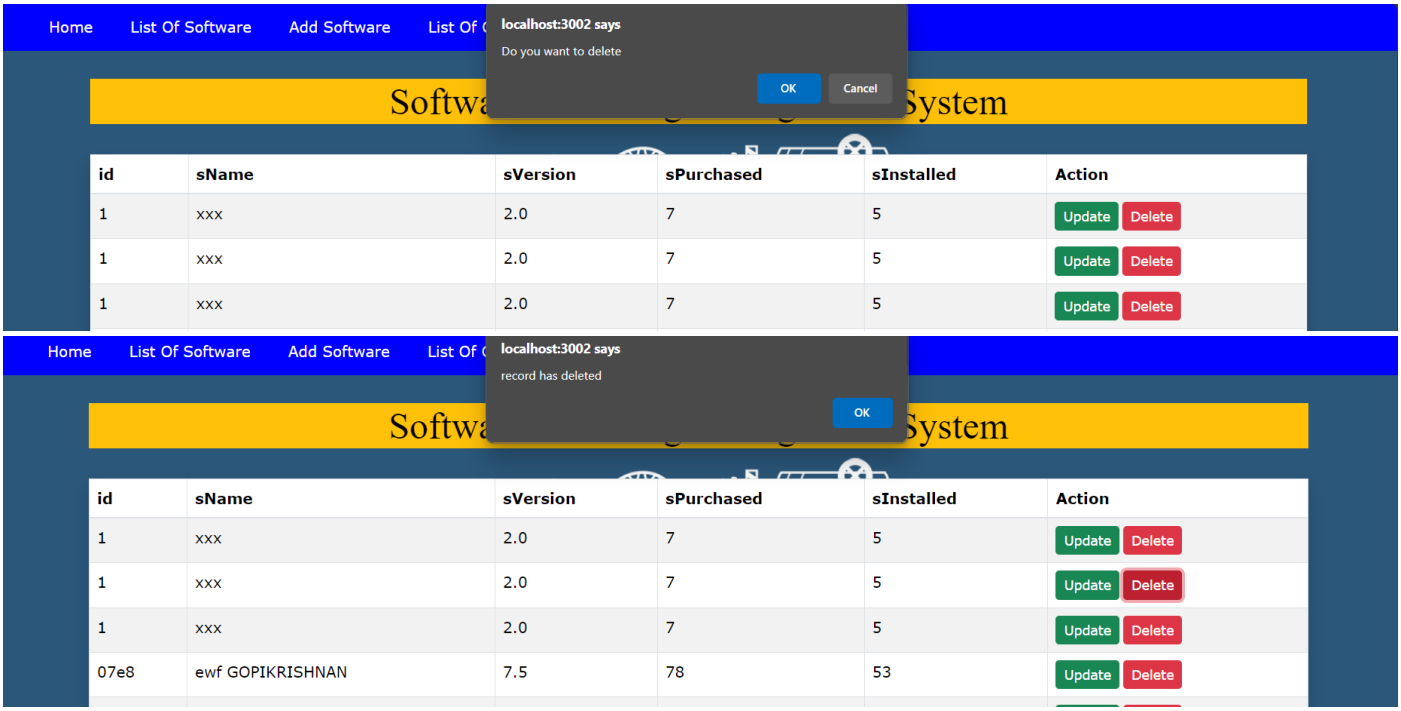


List Of Company:

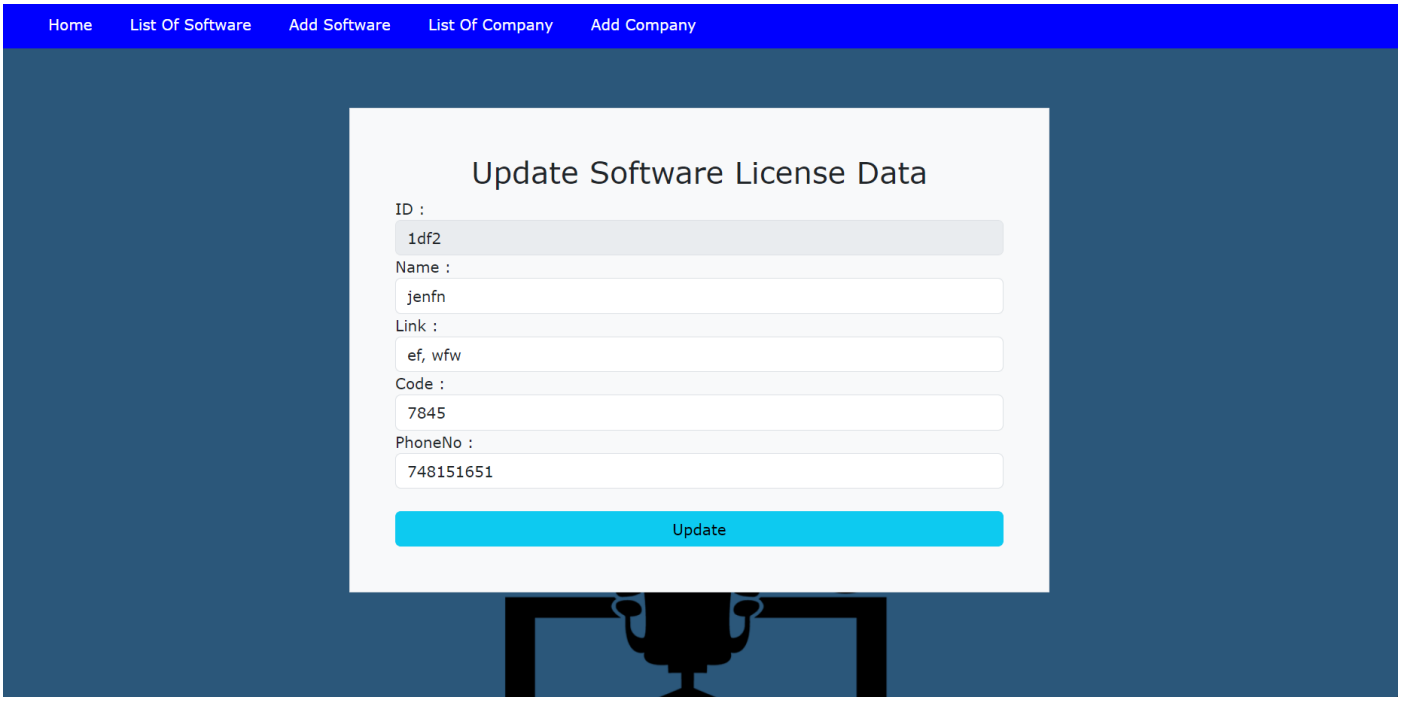
Add New Company:



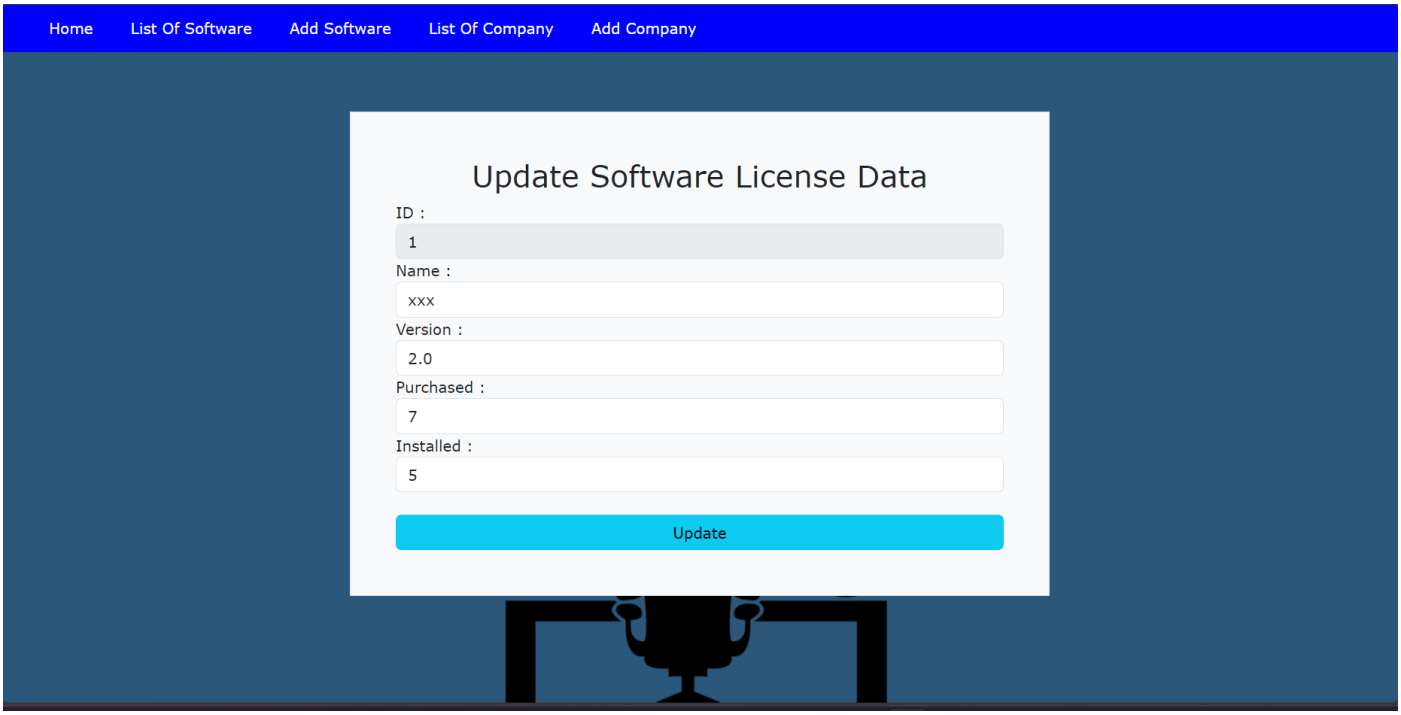
Delete Record:



Update Company:

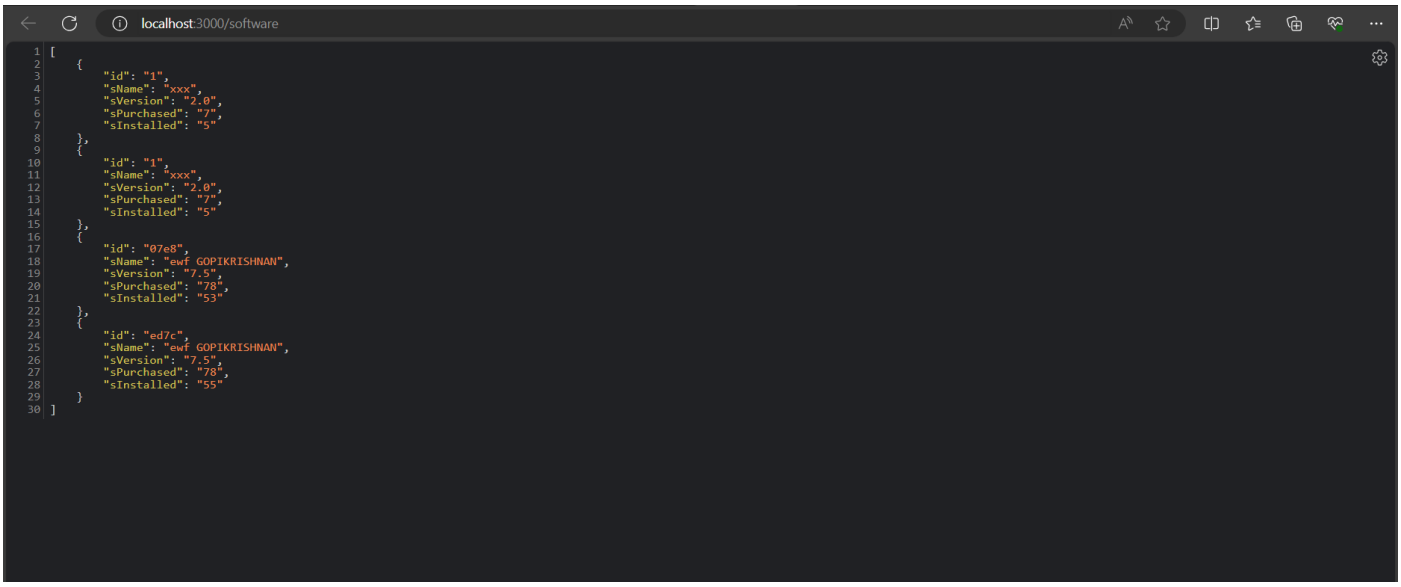


Update Software:

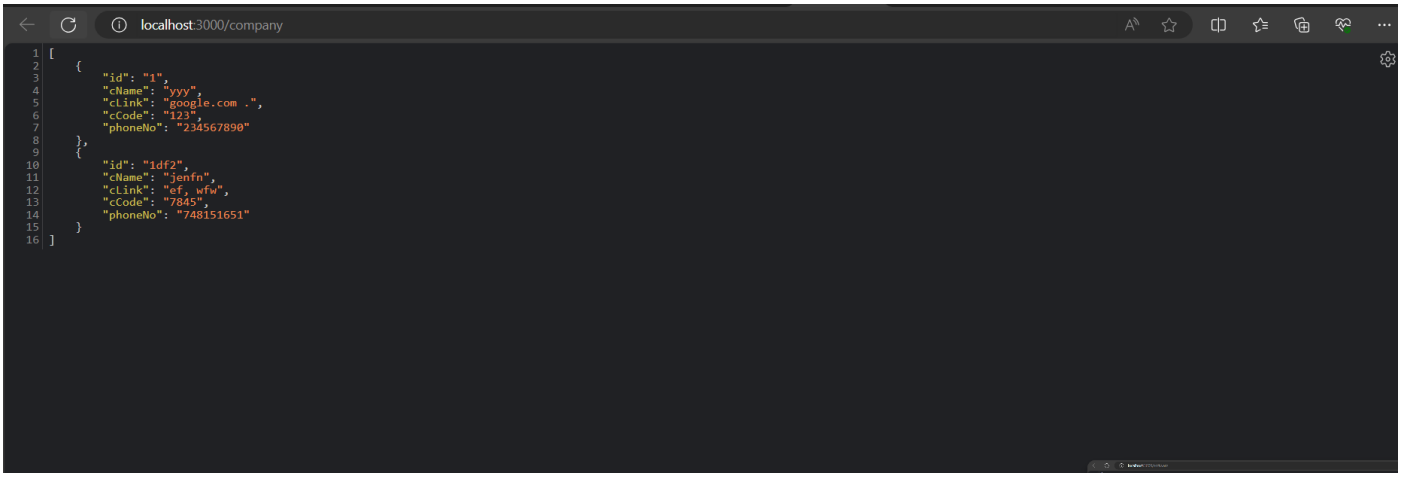


Server:

Software:



Company:



Code:

Index.html:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--
      manifest.json provides metadata used when your web app is installed on a
      user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the build.
      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
      work correctly both with client-side routing and a non-root public URL.
      Learn how to configure a non-root public URL by running `npm run build`.
    -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css" rel="stylesheet">
```

```
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
crossorigin="anonymous">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
    <script src='https://kit.fontawesome.com/a076d05399.js' crossorigin='anonymous'></script>
    <title>React App</title>
  </head>
  <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>
</html>
```

AppRouter.js

```
import React from 'react'
import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";
```

```
import "./HomePage.css";
import View from './View';
import Home from './Home';

import Add from "./Add";

import ViewCom from './ViewCom';
import AddCom from './AddCom';
import EditCom from './EditCom';
import Edit from './Edit';

function AppRouter() {
 return (
  <Router class="head" id="nav">
    <header>
    <nav >
     <ul>
      <li>
       <Link to="/">Home</Link>
      </li>
      <li>
       <Link to="/view">List Of Software</Link>
      </li>
      <li>
       <Link to="/create">Add Software</Link>
      </li>
      <li>
       <Link to="/viewCompany">List Of Company</Link>
      </li>
      <li>
       <Link to="/createCompany">Add Company</Link>
```

```
        </li>

       </ul>
      </nav>

      </header>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/view" element={<View />} />
        <Route path="/create" element={<Add />} />

        <Route path="/viewCompany" element={<ViewCom/>} />
        <Route path="/createCompany" element={<AddCom/>} />
        <Route path="/updateSoftware/:id" element={<Edit/>} />
        <Route path="/updateCompany/:id" element={<EditCom/>} />

      </Routes>
   </Router>
  )
}

export default AppRouter;
```

Home.js:

```
import React from 'react'
import "./Home.css";
function Home() {
  return (
    <div>
     <h1 id = "wel">Welcome to the <br />Software Licensing Management System</h1>
```

```
    </div>
  )
}

export default Home
```

View.js - Software:

```
import React, { useEffect, useState } from "react";
import axios from "axios";
import { Link, useNavigate } from "react-router-dom";
import "./Home.css";

function View() {
  const [columns, setColumns] = useState([]);
  const [records, setRecords] = useState([]);
  const navigate = useNavigate();

  useEffect(() => {
    axios
      .get("http://localhost:3000/software")
      .then((response) => {
        setColumns(Object.keys(response.data[0]));
        setRecords(response.data);
      })
      .catch((err) => {
        console.log(err);
      });
  }, []);

  let handleSubmit = (id) => {
    const conf = window.confirm("Do you want to delete");
    if (conf) {
```

```
   axios
     .delete("http://localhost:3000/software/" + id)
     .then((res) => {
       alert("record has deleted");
       navigate("/View");
       window.location.reload();
     })
     .catch((err) => console.log(err));
  }
};

// let displayHandle = (id) =>{
//   const conf = window.confirm("Do you want to view the data");
//   if(conf){
//     axios
//     .display("http://localhost:5000/users/" + id)
//     .then((res)=> {
//       alert("Display the items");
//       navigate("/describe");
//       window.location.reload();
//     })
//     .catch((err) => console.log(err));
//   }
// }

return (
  <div id="body">
    <div className="container">
      <h1 id="app2" className="text-center text-bg-warning">
      Software Licensing Management System
      </h1>
```

```
<br />
<table className="table table-bordered  table-striped w-100 border bg-white shadow px-5 pb-5 rounded ">
 <thead>
  <tr>
   {columns.map((c, i) => (
    <th key={i}>{c}</th>
   ))}
   <th>Action</th>
  </tr>
 </thead>

 <tbody>
  {records.map((d, i) => (
   <tr key={i}>
    <td>{d.id}</td>
    <td>{d.sName}</td>
    <td>{d.sVersion}</td>
    <td>{d.sPurchased}</td>
    <td>{d.sInstalled}</td>


    <td>
     <Link
      to={` /updateSoftware/${d.id}` }
      className="btn btn-sm btn-success "
     >
      Update
     </Link>
```

```
            <button
              onClick={(e) => handleSubmit(d.id)}
              className="btn btn-sm ms-1 btn-danger"
            >
              Delete
            </button>
            {/* <Link
            to={` /describe/${d.id}` }
            className="btn btn-sm btn-warning"
            >Display</Link> */}
          </td>
        </tr>
      ))}
    </tbody>
  </table>
 </div>
</div>
);
}

export default View;
```

Add.js:

```
import axios from "axios";
import React from "react";
import { useState } from "react";
import { useNavigate } from "react-router-dom";
import "./Home.css";
import View from './View';

function Add() {
  const [inputData, setInputData] = useState({
```

```
  sName: "",
  sVersion: "",
  sPurchased: "",
  sInstalled: "",
});

const naviget = useNavigate();

let handleSubmit = (e) => {
  e.preventDefault();
  let result = validateValues(inputData);
  // setSubmitting(true);

  if (result === true) {
    axios
      .post("http://localhost:3000/software", inputData)
      .then((res) => {
        alert("Data added Successfully");
        naviget("/view");
        console.log(res.data);
      })
      .catch((err) => console.log(err));
  } else {
    alert("Please Enter the Valid Inputs!!!");
  }
};

// validation Part for add user for student management system
// const [errors, setErrors] = useState({});
// const [submitting, setSubmitting] = useState(false);
```

```
const validateValues = (inputData) => {
  if (inputData.sName.length === 0) {
    alert("Please enter Software Name !!! ");
    return false;

  } else {
    return true;
  }
};

return (
  <div
    id="add2"
    className="d-flex w-100 vh-100 justify-content-center align-items-center "
  >
    <div className="w-50 border bg-light p-5">
      <form onSubmit={handleSubmit}>
        <h2 id="align">Add New Software Data</h2>
        <div>
          <lable htmlFor="name">Name</lable>
          <input
            type="text"
            name="name"
            className="form-control"
            placeholder="Enter Software Name"
            onChange={(e) =>
              setInputData({ ...inputData, sName: e.target.value })
            }
          />
        </div>
        <div>
```

```
 <lable htmlFor="Mobile">Version :</lable>
 <input
  type="text"
  name="mobile"
  className="form-control"
  placeholder="Enter Version"
  onChange={(e) =>
   setInputData({ ...inputData, sVersion: e.target.value })
  }
 />
</div>

<div>
 <lable htmlFor="city">Purchased :</lable>
 <input
  type="number"
  name="city"
  className="form-control"
  placeholder="Enter purchased count"
  onChange={(e) =>
   setInputData({ ...inputData, sPurchased: e.target.value })
  }
 />
</div>

<div>
 <lable htmlFor="pincode">Installed :</lable>
 <input
  type="number"
  name="number"
  className="form-control"
```

```
          placeholder="Enter installed Count"
          onChange={(e) =>
            setInputData({ ...inputData, sInstalled: e.target.value })
          }
         />
        </div>
        <br />

        <button id="value" className="btn btn-info ">Submit</button>
       </form>



     </div>
    </div>
  );
}

export default Add;
```

Edit.js - Software:

```
import axios from "axios";
import React, { useEffect, useState } from "react";
import { useNavigate, useParams } from "react-router-dom";
import "./Home.css";

function Edit() {
  const { id } = useParams();
  const [data, setData] = useState([]);
  const navigate = useNavigate();
  useEffect(() => {
    axios
```

```
    .get("http://localhost:3000/software/" + id)
    .then((response) => setData(response.data))
    .catch((err) => console.log(err));
}, []);

let handleSubmit = (e) => {
  e.preventDefault();
  axios.put("http://localhost:3000/software/" + id, data).then((res) => {
    alert("User Updated Successfully");
    navigate("/View");
  });
};

return (
  <div>
    <div
      id="edit2"
      className="d-flex w-100 vh-100 justify-content-center align-items-center "
    >
      <div className="w-50 border bg-light p-5">
        <form onSubmit={handleSubmit}>
          <h2 id="align">Update Software License Data</h2>
          <div>
            <lable htmlFor="id">ID :</lable>
            <input
              type="text"
              disabled
              name="name"
              className="form-control"
              value={data.id}
              onChange={(e) => setData({ ...data, id: e.target.value })}
```

```
  />
</div>

<div>
  <lable htmlFor="name">Name :</lable>
  <input
    type="text"
    name="name"
    className="form-control"
    value={data.sName}
    onChange={(e) => setData({ ...data, sName: e.target.value })}
  />
</div>
<div>
  <lable htmlFor="mobile">Version :</lable>
  <input
    type="text"
    name="version"
    className="form-control"
    value={data.sVersion}
    onChange={(e) => setData({ ...data, sVersion: e.target.value })}
  />
</div>

<div>
  <lable htmlFor="city">Purchased :</lable>
  <input
    type="number"
    name="purchased"
    className="form-control"
    value={data.sPurchased}
```

```
            onChange={(e) => setData({ ...data, sPurchased: e.target.value })}
           />
          </div>

          <div>
           <lable htmlFor="pincode">Installed :</lable>
           <input
            type="number"
            name="installed"
            className="form-control"
            value={data.sInstalled}
            onChange={(e) => setData({ ...data, sInstalled: e.target.value })}
           />
          </div>

          <br />
          <button id="value" className="btn btn-info ">Update</button>
        </form>
      </div>
    </div>
  </div>
 );
}

export default Edit;
```

ViewCom.js: - Company:

```
import React, { useEffect, useState } from "react";
import axios from "axios";
import { Link, useNavigate } from "react-router-dom";
import "./Home.css";
```

```
function ViewCom() {
  const [columns, setColumns] = useState([]);
  const [records, setRecords] = useState([]);
  const navigate = useNavigate();

  useEffect(() => {
    axios
      .get("http://localhost:3000/company")
      .then((response) => {
        setColumns(Object.keys(response.data[0]));
        setRecords(response.data);
      })
      .catch((err) => {
        console.log(err);
      });
  }, []);

  let handleSubmit = (id) => {
    const conf = window.confirm("Do you want to delete");
    if (conf) {
      axios
        .delete("http://localhost:3000/company/" + id)
        .then((res) => {
          alert("record has deleted");
          navigate("/ViewCompany");
          window.location.reload();
        })
        .catch((err) => console.log(err));
    }
  };
```

```
// let displayHandle = (id) =>{
//   const conf = window.confirm("Do you want to view the data");
//   if(conf){
//     axios
//     .display("http://localhost:5000/users/" + id)
//     .then((res)=> {
//       alert("Display the items");
//       navigate("/describe");
//       window.location.reload();
//     })
//     .catch((err) => console.log(err));
//   }
// }

 return (
   <div id="body">
     <div className="container">
       <h1 id="app2" className="text-center text-bg-warning">
       Company Details
       </h1>



       <br />
       <table className="table table-bordered  table-striped w-100 border bg-white shadow px-5 pb-5
rounded ">
         <thead>
           <tr>
             {columns.map((c, i) => (
               <th key={i}>{c}</th>
             ))}
             <th>Action</th>
```

```
 </tr>
</thead>

<tbody>
 {records.map((d, i) => (
  <tr key={i}>
   <td>{d.id}</td>
   <td>{d.cName}</td>
   <td>{d.cLink}</td>
   <td>{d.cCode}</td>
   <td>{d.phoneNo}</td>


   <td>
    <Link
     to={` /updateCompany/${d.id}` }
     className="btn btn-sm btn-success "
    >
     Update
    </Link>
    <button
     onClick={(e) => handleSubmit(d.id)}
     className="btn btn-sm ms-1 btn-danger"
    >
     Delete
    </button>
    {/* <Link
    to={` /describe/${d.id}` }
    className="btn btn-sm btn-warning"
    >Display</Link> */}
   </td>
```

```
          </tr>
         ))}
        </tbody>
      </table>
    </div>
  </div>
 );
}

export default ViewCom;
```

AddCom.js - Company:

```
import axios from "axios";
import React from "react";
import { useState } from "react";
import { useNavigate } from "react-router-dom";
import "./Home.css";
import ViewCom from './ViewCom';

function AddCom() {
 const [inputData, setInputData] = useState({
  cName: "",
  cLink: "",
  cCode: "",
  phoneNo: "",
 });

 const naviget = useNavigate();

 let handleSubmit = (e) => {
```

```
  e.preventDefault();
  let result = validateValues(inputData);
  // setSubmitting(true);

  if (result === true) {
    axios
      .post("http://localhost:3000/company", inputData)
      .then((res) => {
        alert("Data added Successfully");
        naviget("/viewCompany");
        console.log(res.data);
      })
      .catch((err) => console.log(err));
  } else {
    alert("Please Enter the Valid Inputs!!!");
  }
};

// validation Part for add user for student management system
// const [errors, setErrors] = useState({});
// const [submitting, setSubmitting] = useState(false);

const validateValues = (inputData) => {
  if (inputData.cName.length === 0) {
    alert("Please enter company Name !!! ");
    return false;

  } else {
    return true;
  }
};
```

```
return (
 <div
  id="add2"
  className="d-flex w-100 vh-100 justify-content-center align-items-center "
 >
  <div className="w-50 border bg-light p-5">
   <form onSubmit={handleSubmit}>
    <h2 id="align">Add New Company Data</h2>
    <div>
     <lable htmlFor="name">Name</lable>
     <input
      type="text"
      name="name"
      className="form-control"
      placeholder="Enter company Name"
      onChange={(e) =>
       setInputData({ ...inputData, cName: e.target.value })
      }
     />
    </div>
    <div>
     <lable htmlFor="Mobile">Link :</lable>
     <input
      type="text"
      name="link"
      className="form-control"
      placeholder="Enter Link"
      onChange={(e) =>
       setInputData({ ...inputData, cLink: e.target.value })
      }
```

```
 />
</div>

<div>
 <lable htmlFor="city">Code :</lable>
 <input
  type="number"
  name="code"
  className="form-control"
  placeholder="Enter Company Code"
  onChange={(e) =>
   setInputData({ ...inputData, cCode: e.target.value })
  }
 />
</div>

<div>
 <lable htmlFor="pincode">phoneNo :</lable>
 <input
  type="number"
  name="pnoneNo"
  className="form-control"
  placeholder="Enter PhoneNo "
  onChange={(e) =>
   setInputData({ ...inputData, phoneNo: e.target.value })
  }
 />
</div>
<br />

<button id="value" className="btn btn-info ">Submit</button>
```

```
      </form>

    </div>
   </div>
  );
}

export default AddCom;
```

EditCom.js - Company:

```
import axios from "axios";
import React, { useEffect, useState } from "react";
import { useNavigate, useParams } from "react-router-dom";
import "./Home.css";

function EditCom() {
  const { id } = useParams();
  const [data, setData] = useState([]);
  const navigate = useNavigate();
  useEffect(() => {
    axios
     .get("http://localhost:3000/company/" + id)
     .then((response) => setData(response.data))
     .catch((err) => console.log(err));
  }, []);

  let handleSubmit = (e) => {
    e.preventDefault();
    axios.put("http://localhost:3000/company/" + id, data).then((res) => {
     alert("Company Updated Successfully");
     navigate("/ViewCompany");
    });
```

```
};

return (
 <div>
  <div
   id="edit2"
   className="d-flex w-100 vh-100 justify-content-center align-items-center "
  >
   <div className="w-50 border bg-light p-5">
    <form onSubmit={handleSubmit}>
     <h2 id="align">Update Software License Data</h2>
     <div>
      <lable htmlFor="id">ID :</lable>
      <input
       type="text"
       disabled
       name="name"
       className="form-control"
       value={data.id}
       onChange={(e) => setData({ ...data, id: e.target.value })}
      />
     </div>

     <div>
      <lable htmlFor="name">Name :</lable>
      <input
       type="text"
       name="name"
       className="form-control"
       value={data.cName}
       onChange={(e) => setData({ ...data, cName: e.target.value })}
```

```
 />
</div>
<div>
 <lable htmlFor="mobile">Link :</lable>
 <input
  type="text"
  name="link"
  className="form-control"
  value={data.cLink}
  onChange={(e) => setData({ ...data, cLink: e.target.value })}
 />
</div>

<div>
 <lable htmlFor="city">Code :</lable>
 <input
  type="number"
  name="code"
  className="form-control"
  value={data.cCode}
  onChange={(e) => setData({ ...data, cCode: e.target.value })}
 />
</div>

<div>
 <lable htmlFor="pincode">PhoneNo :</lable>
 <input
  type="number"
  name="phoneNo"
  className="form-control"
  value={data.phoneNo}
```

```
            onChange={(e) => setData({ ...data, phoneNo: e.target.value })}
           />
          </div>

          <br />
          <button id="value" className="btn btn-info ">Update</button>
         </form>
        </div>
       </div>
      </div>
    );
}

export default EditCom;
```

Server

DB.JSON:

```
{
  "software": [
    {
      "id": "1",
      "sName": "xxx",
      "sVersion": "2.0",
      "sPurchased": "7",
      "sInstalled": "5"
    },
    {
      "id": "1",
      "sName": "xxx",
      "sVersion": "2.0",
      "sPurchased": "7",
```

```
   "sInstalled": "5"
  },
  {
   "id": "07e8",
   "sName": "ewf GOPIKRISHNAN",
   "sVersion": "7.5",
   "sPurchased": "78",
   "sInstalled": "53"
  },
  {
   "id": "ed7c",
   "sName": "ewf GOPIKRISHNAN",
   "sVersion": "7.5",
   "sPurchased": "78",
   "sInstalled": "55"
  }
 ],
 "company": [
  {
   "id": "1",
   "cName": "yyy",
   "cLink": "google.com .",
   "cCode": "123",
   "phoneNo": "234567890"
  },
  {
   "id": "1df2",
   "cName": "jenfn",
   "cLink": "ef, wfw",
   "cCode": "7845",
   "phoneNo": "748151651"
```

```
    }
  ]
}
```

Package.json:

```
{
  "dependencies": {
    "json-server": "^1.0.0-alpha.23"
  },
  "name": "server",
  "version": "1.0.0",
  "main": "index.js",
  "devDependencies": {},
  "scripts": {
    "start": "json-server db.json --port 3000"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}
```