**Day6(06-Sept) React.js**

**=============================================================**

**Session Agenda:**

axios library

Host React in IIS

Routing

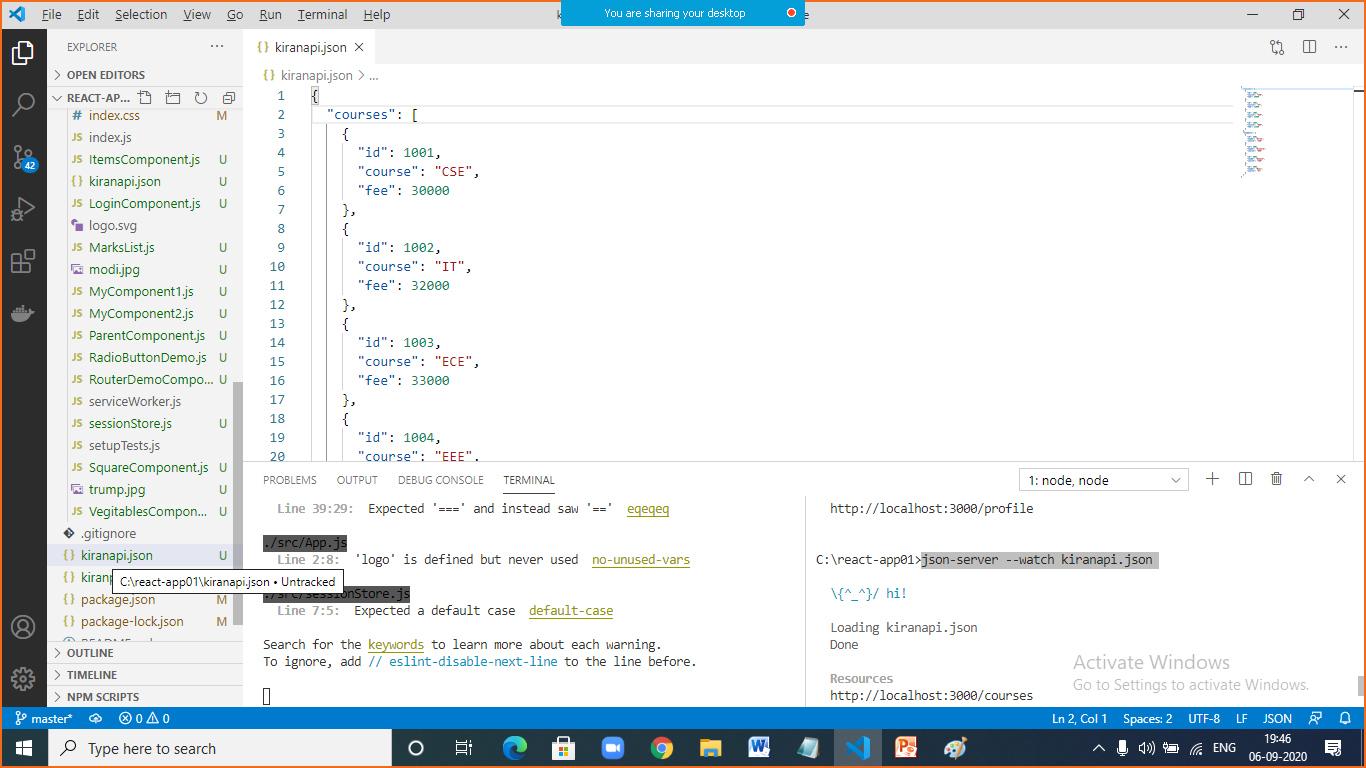
Redux

Mongodb

Create Web API using node.js

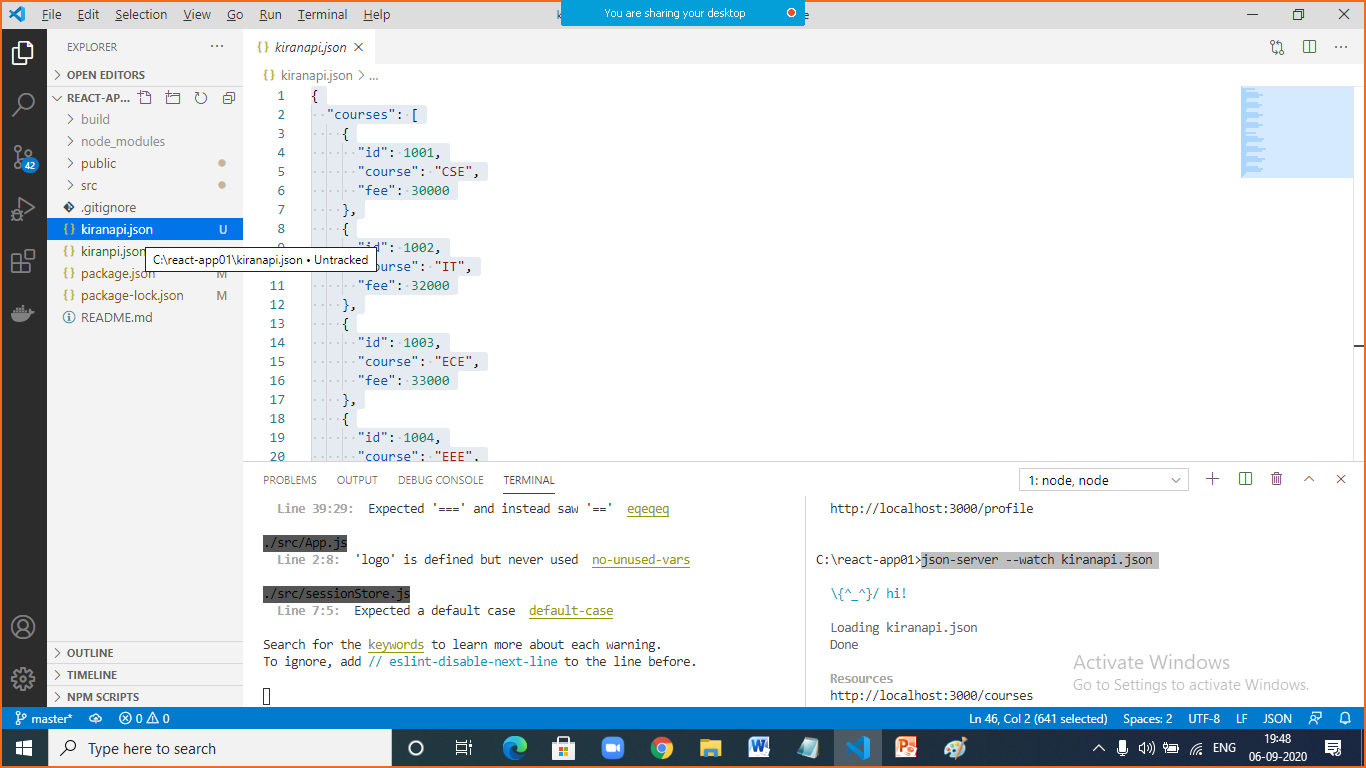
|  |  |
| --- | --- |
|  | **axios library** |
|  | This library is used to consume Web API  This library contains methods used to interact with Web API to perform CRUD operations using get(), post(), put(), delete() methods |

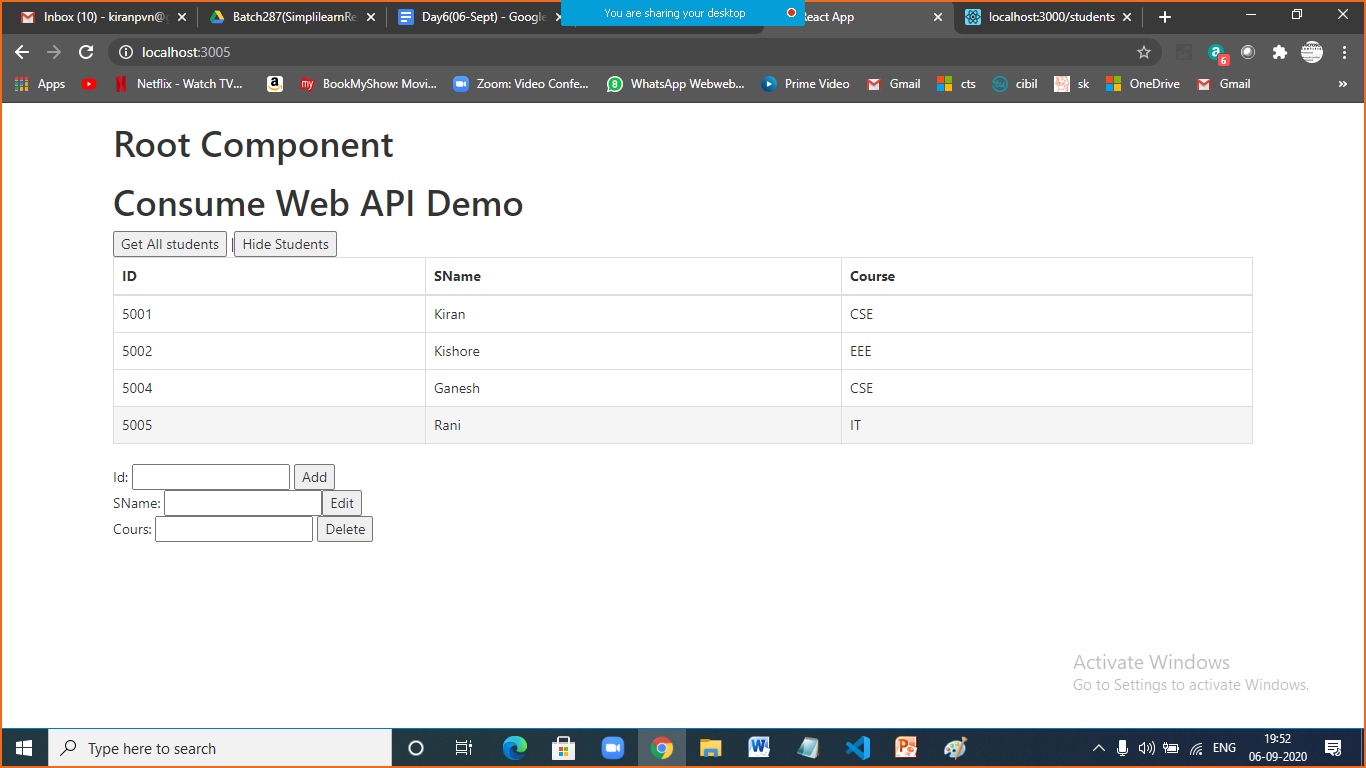
|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | To install | npm install axios |
|  | To include in javascript | var axios = require("axios").default; |
|  | To include in ES6 | import axios from 'axios'; |

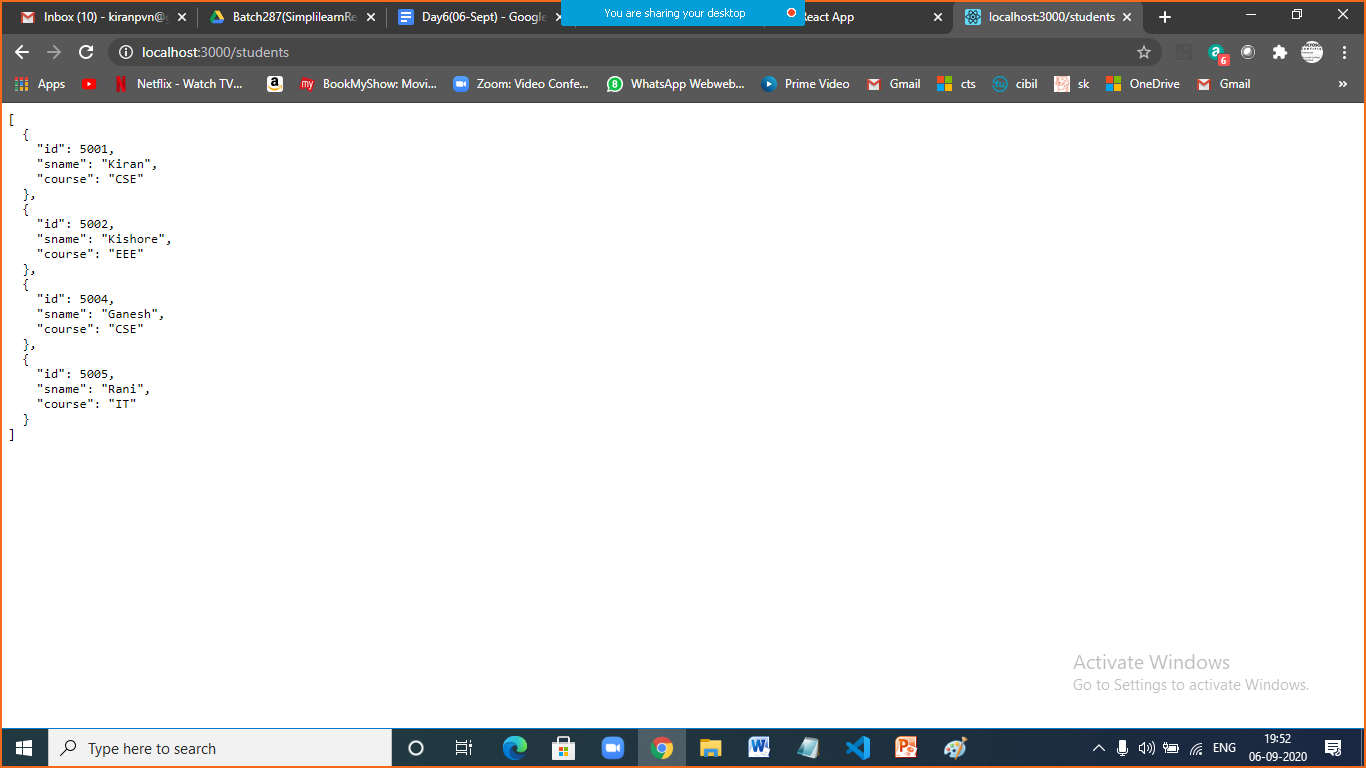


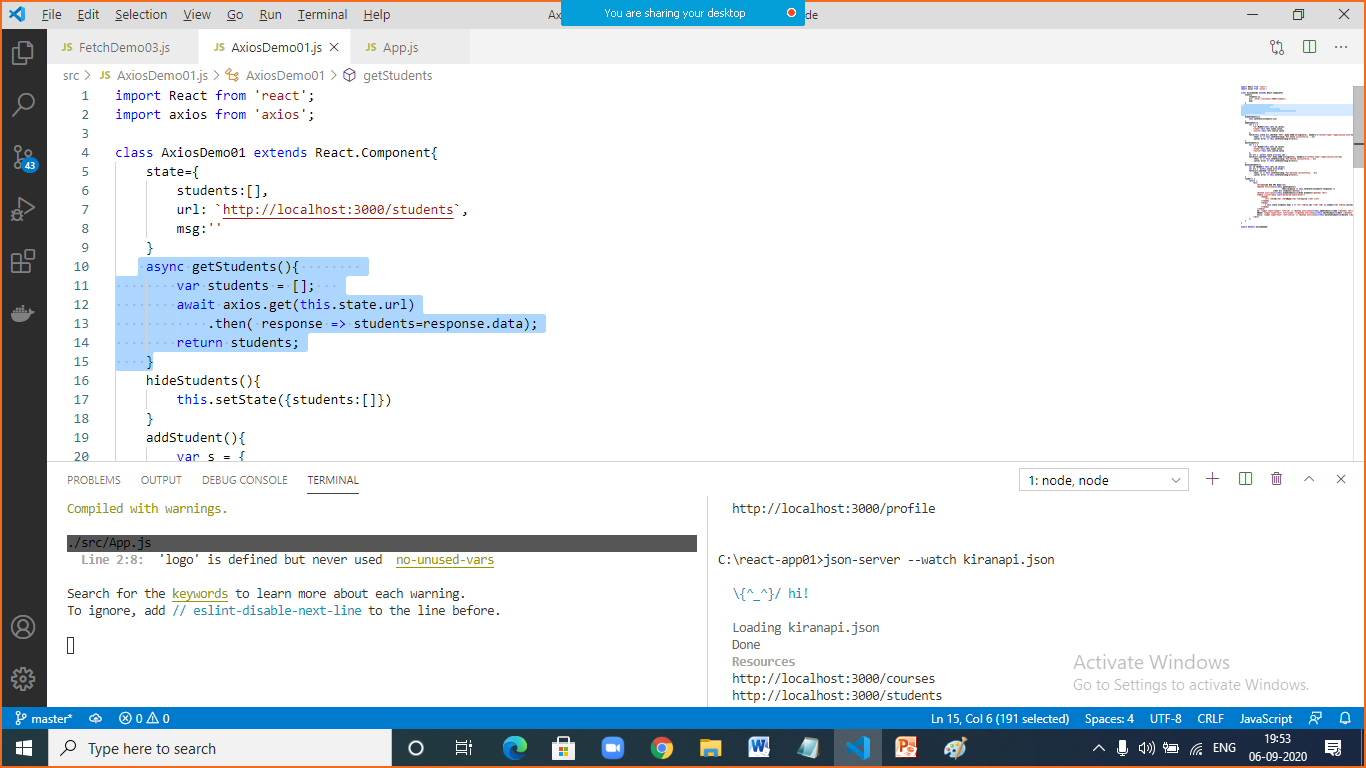
|  |  |
| --- | --- |
|  | kiranapi.json |
|  | {  "courses": [  {  "id": 1001,  "course": "CSE",  "fee": 30000  },  {  "id": 1002,  "course": "IT",  "fee": 32000  },  {  "id": 1003,  "course": "ECE",  "fee": 33000  },  {  "id": 1004,  "course": "EEE",  "fee": 31500  }  ],  "students": [  {  "id": 5001,  "sname": "Kiran",  "course": "CSE"  },  {  "id": 5002,  "sname": "Kishore",  "course": "EEE"  },  {  "id": 5004,  "sname": "Ganesh",  "course": "CSE"  },  {  "id": 5005,  "sname": "Rani",  "course": "IT"  }  ]  } |

Note: Make sure web api exist in the root folder









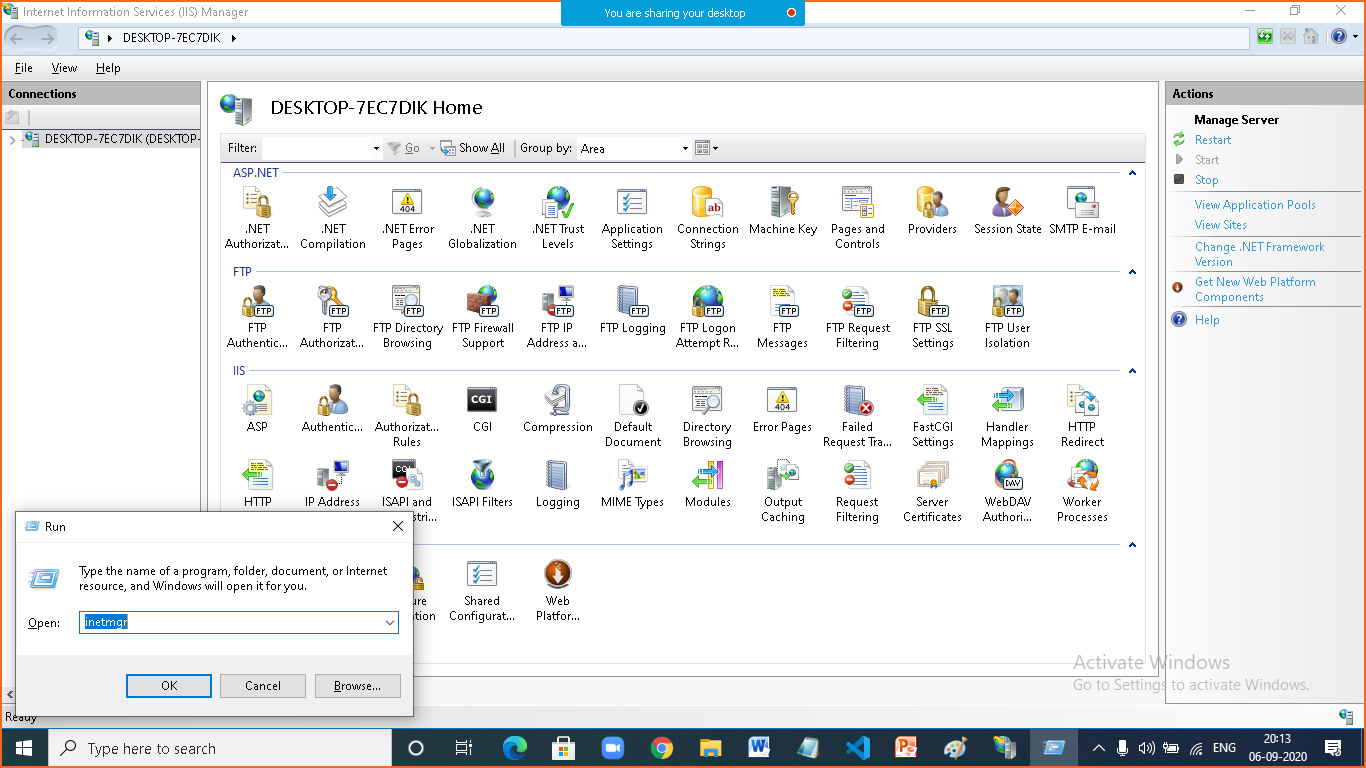
|  |  |
| --- | --- |
|  | AxiosDemo01.js |
|  | import React from 'react';  import axios from 'axios';  class AxiosDemo01 extends React.Component{  state={  students:[],  url: `http://localhost:3000/students`,  msg:''  }  async getStudents(){  var students = [];  await axios.get(this.state.url)  .then( response => students=response.data);  return students;  }  hideStudents(){  this.setState({students:[]})  }  addStudent(){  var s = {  "id":Number(this.refs.id.value),  "sname":this.refs.sname.value,  "course":this.refs.course.value  };  axios.post(this.state.url,s)  .then( () => this.setState({msg:'Row Added successfully...'}))  .catch( error => this.setState({msg:error}));  }  editStudent(){  var s = {  "id":Number(this.refs.id.value),  "sname":this.refs.sname.value,  "course":this.refs.course.value  };  var url = `${this.state.url}/${s.id}`;  axios.put(url,s)  .then( () => this.setState({msg:'Row Edited successfully...'}))  .catch( error => this.setState({msg:error}));  }  deleteStudent(){  var id =Number( this.refs.id.value);  var url = `${this.state.url}/${id}`;  axios.delete(url)  .then( () => this.setState({msg:'Row Deleted successfully...'}))  .catch( error => this.setState({msg:error}));  }  render() {  return (  <div>  <h1>Consume Web API Demo</h1>  <button onClick={()=>this.getStudents()  .then(response => this.setState({students:response} ))  }>Get All students</button> |  <button onClick={()=>this.hideStudents()}>Hide Students</button> <br/>  <table class="table table-bordered table-hover">  <thead>  <tr> <th>ID</th> <th>SName</th> <th>Course </th> </tr>  </thead>  <tbody>  { this.state.students.map( s => <tr> <td>{s.id} </td> <td> {s.sname}</td> <td>{s.course}</td> </tr> )}  </tbody>  </table>  Id: <input type="number" ref="id" /> <button onClick={()=>this.addStudent()}>Add </button> <br/>  SName: <input type="text" ref="sname" /><button onClick={()=>this.editStudent()}>Edit </button> <br/>  Cours: <input type="text" ref="course" /> <button onClick={()=>this.deleteStudent()}>Delete </button><br/> {this.state.msg}  </div>  );  }  }  export default AxiosDemo01; |

Steps for : .NET Core Web API with enable CORS

|  |  |
| --- | --- |
|  |  |
|  | Step1: Create API application using MVC.NET Core  Install EF Core packages  install-package microsoft.entityframeworkcore  install-package microsoft.entityframeworkcore.tools  install-package microsoft.entityframeworkcore.sqlserver |
|  | Step2: provide connection string in appsettings.json file  "ConnectionStrings": { "cs1": "server=DESKTOP-7EC7DIK\\SQLEXPRESS;user id=sa;password=welcome;database=IncedonicDB" } |
|  | Create table and add rows into database  Step3: Create Dbcontext and entities using EFCore Scaffold feature  scaffold-dbcontext "Data Source=DESKTOP-7EC7DIK\SQLEXPRESS;Initial Catalog=IncedonicDB;Persist Security Info=True;User ID=sa;Password=welcome;database=SwathiDB" microsoft.entityframeworkcore.sqlserver -outputdir models |
|  | Step4: Configure context in startup.cs  services.AddDbContext<IncedonicDBContext>(options => options.UseSqlServer(Configuration.GetConnectionString("cs1"))); |
|  | Step5: Create controller and name it as EmployeesController  Add controller with views using EF Core  Provide Model as Employees, Context: IncedonicDBcontext |
|  | Step6: Enable cors  Inject service  services.AddCors(options =>  {  options.AddPolicy("AllowOrigin", options => options. AllowAnyOrigin());  });    Add cors in request/response pipe line  app.UseCors(options => options.AllowAnyOrigin());    Provide attribute for enable cors at api controller prefix  [EnableCors("AllowOrigin")]    Build and run the project |

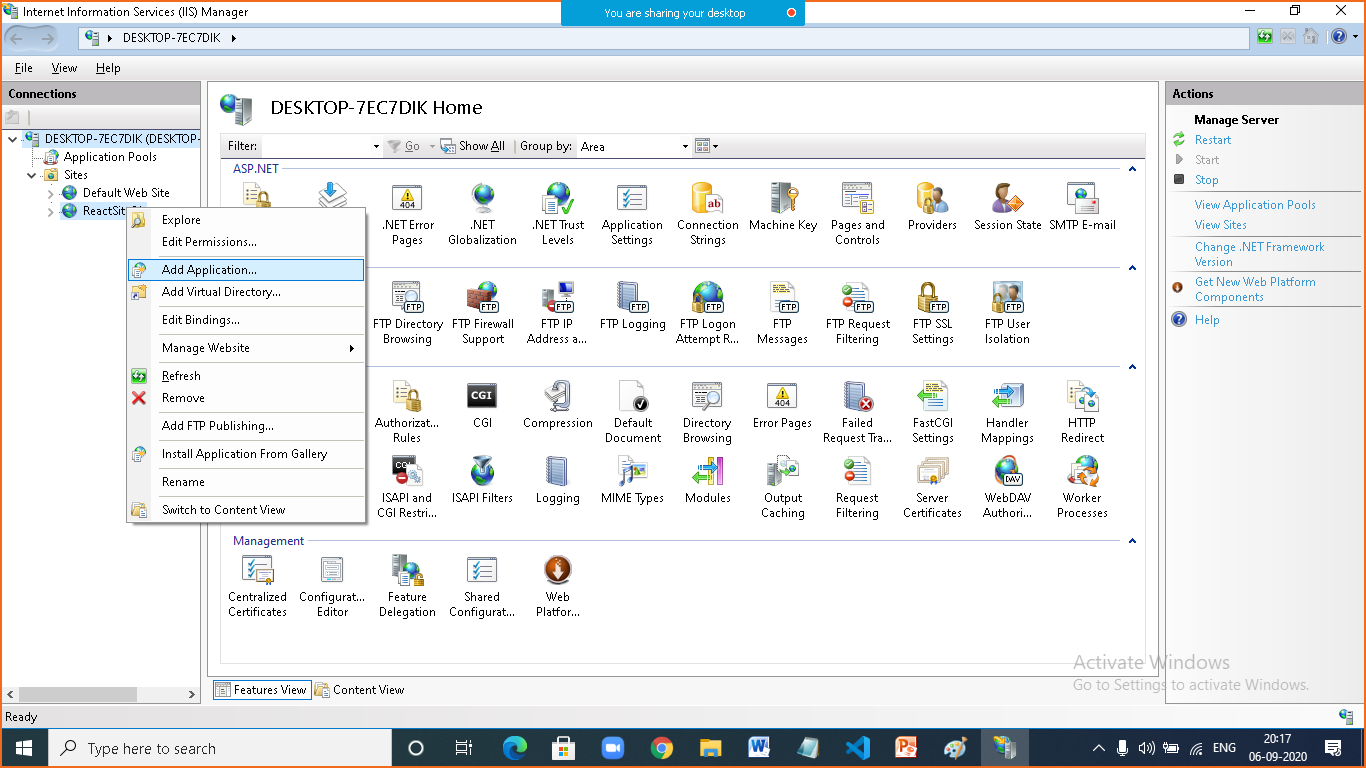
|  |  |
| --- | --- |
|  | **Understand Web Server** |
|  | Web server is a centralized software, used to host web applications and WebServices  Advantage:  The application/service within web server, available to access in network  **Available web servers:**  Apache, Tomcat, Weblogic, jboss, IIS |

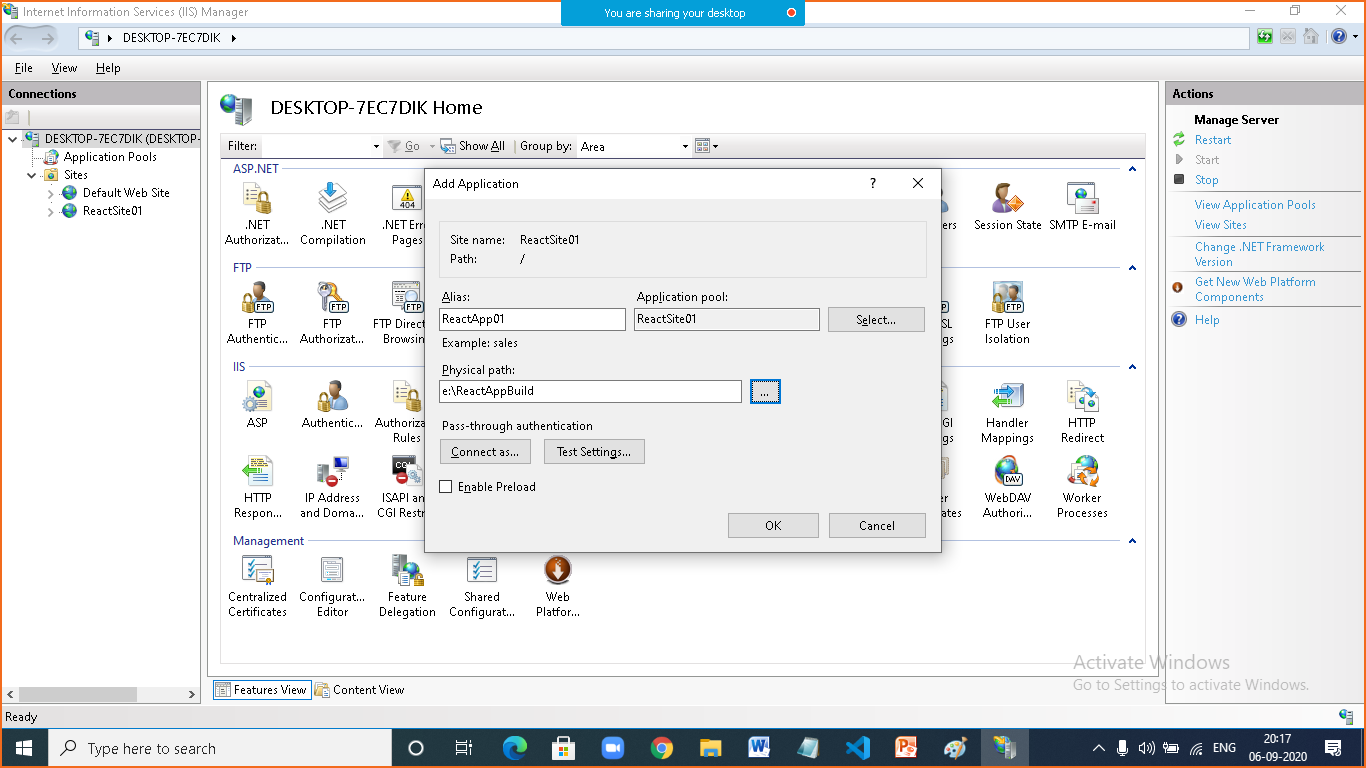
|  |  |
| --- | --- |
|  | **Understand IIS** |
|  | IIS, internet information services  It is a Web Server from Microsoft.  IIS is inbuilt with microsoft, by-default this feature is OFF and we have to ON  How to install/enable IIS in windows?  Open control panel, add/remove program  Turn Windows feature on/off  Choose IIS (Internet information services)  Check all the options under IIS then install  To open IIS?  start=>run=> type inetmgr and hit enter |

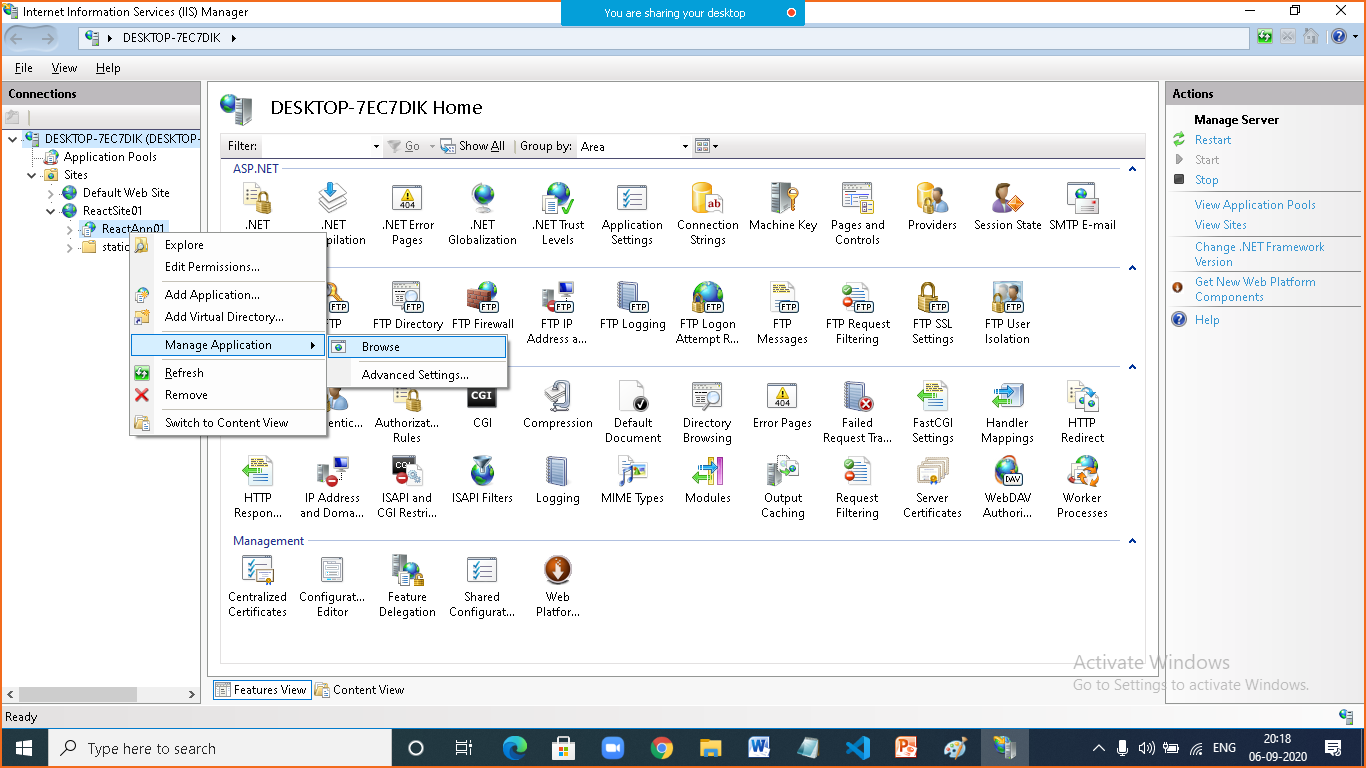


|  |  |
| --- | --- |
|  | **Hosting React application in IIS** |
|  | In order to host React application, build the project  Step1: Npm start build  Note: once the project is build, the required source will be copied into **build** folder in the project  Copy build folder to new folder and name it as c:\ReactProjectBuild  Create Web Site in IIS |

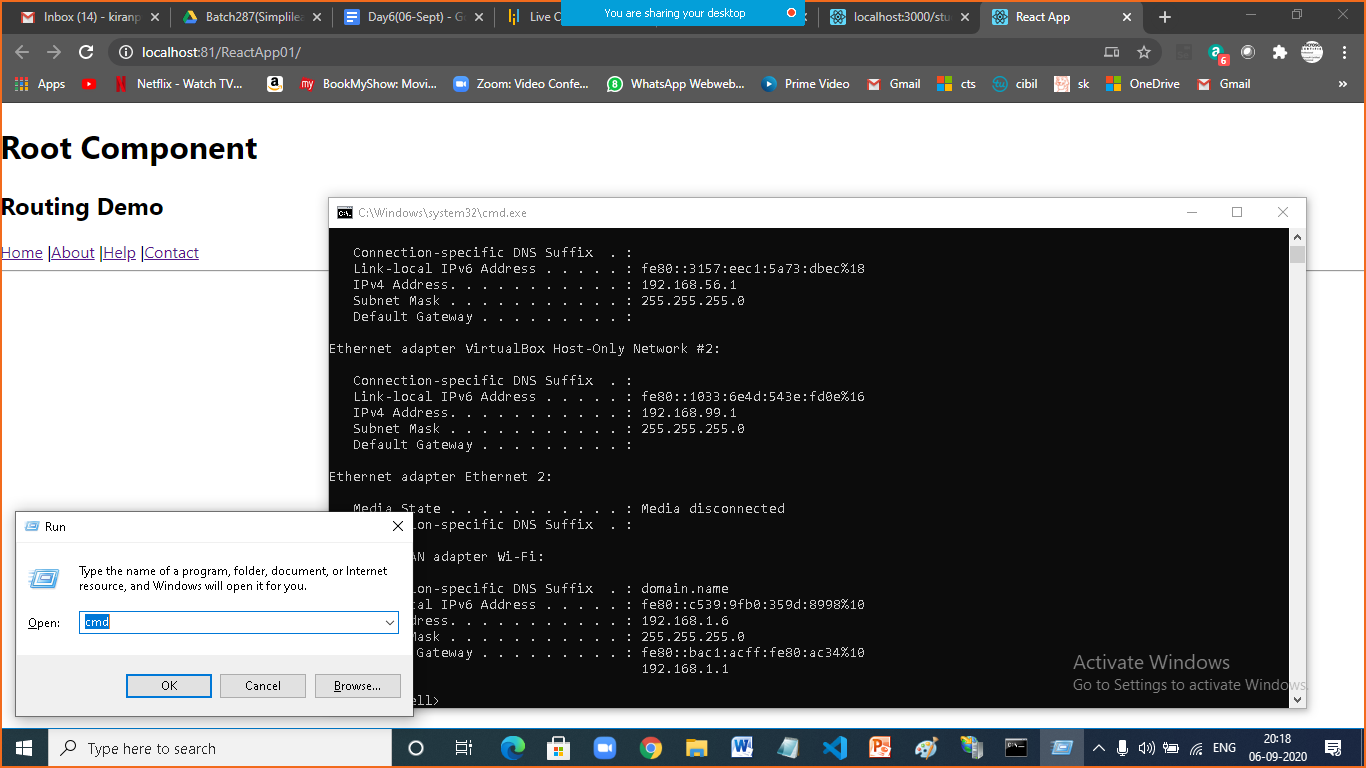
Add application into that web site

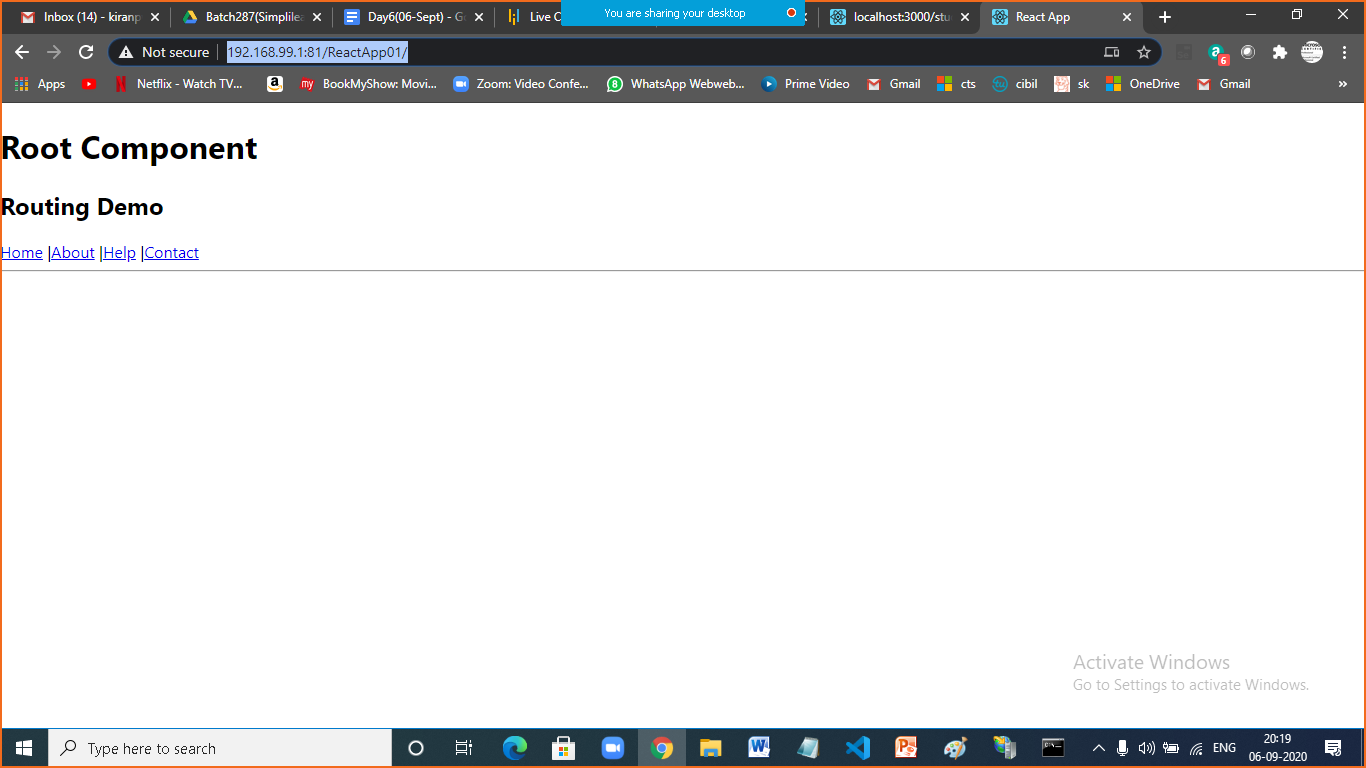


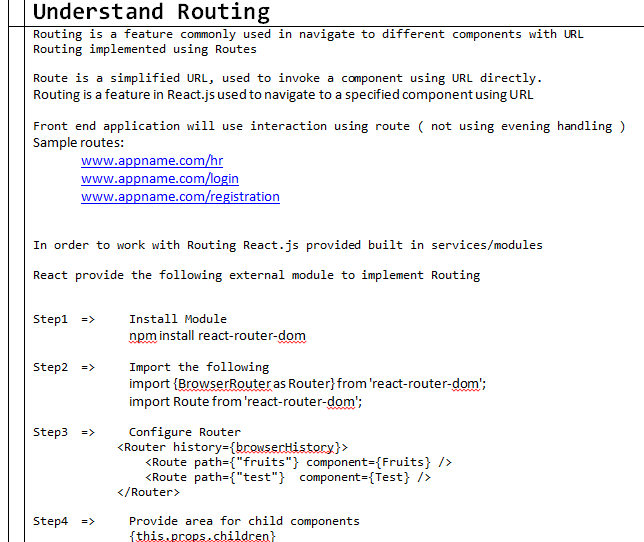




To know your IP address, open command prompt and issue ipconfig command

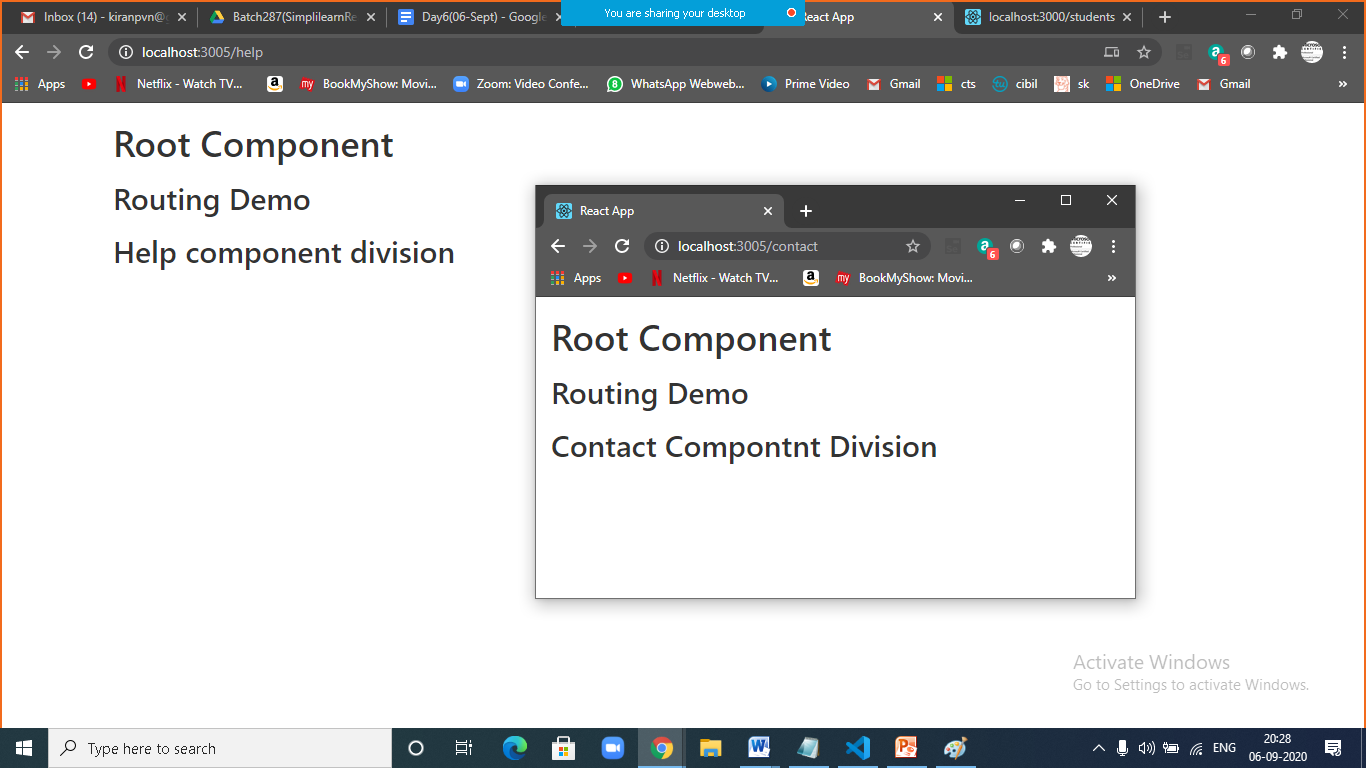


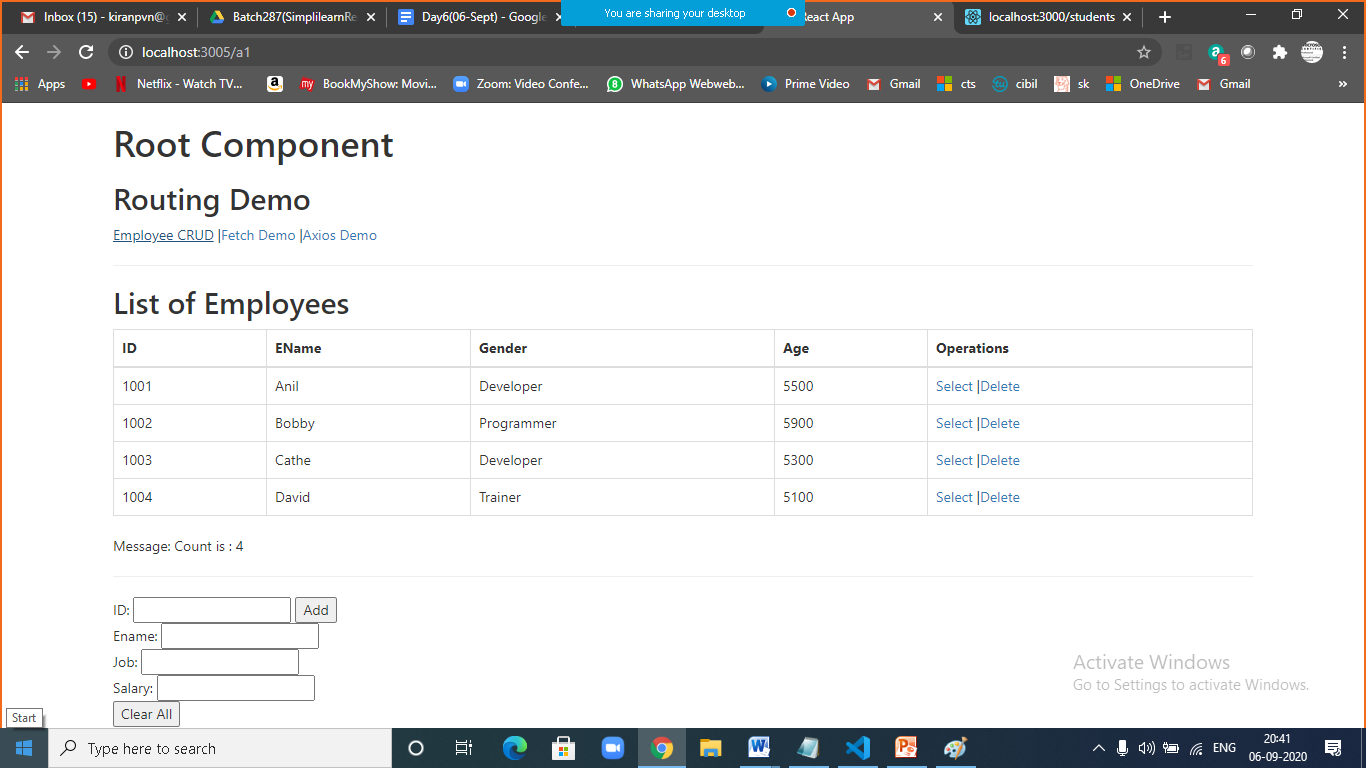




To install:

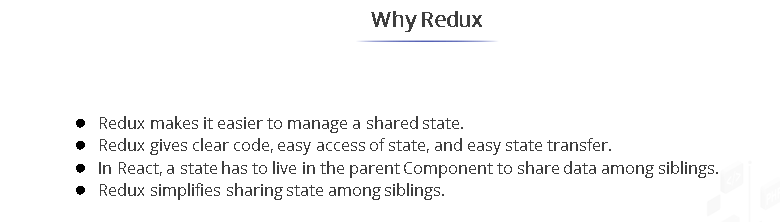
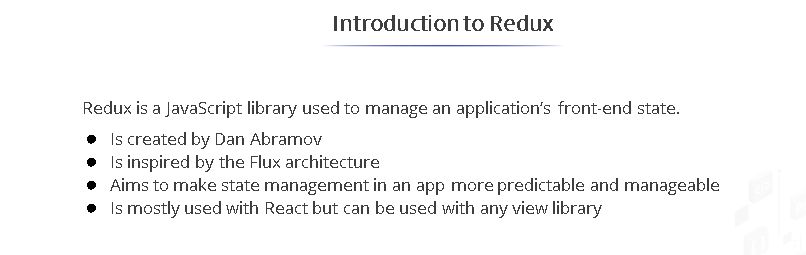
npm install react-router-dom

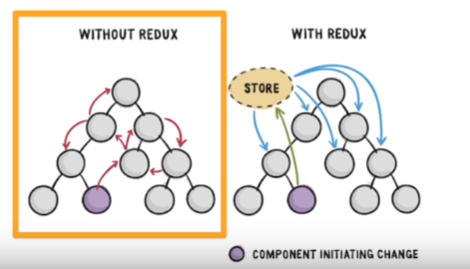


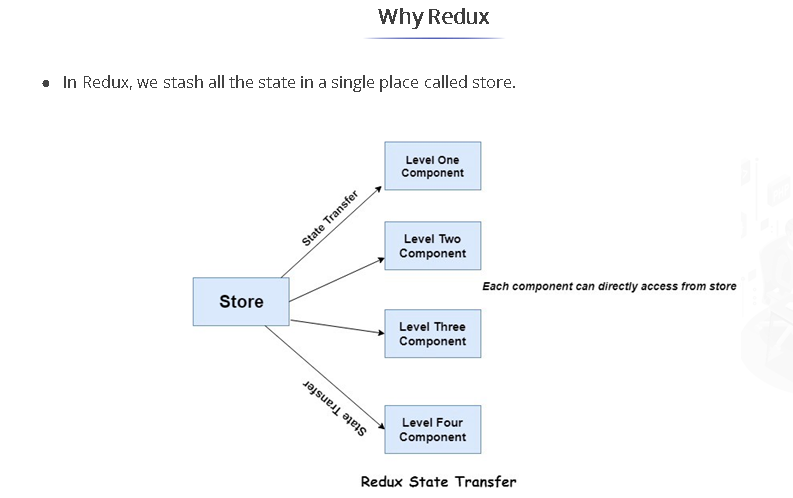


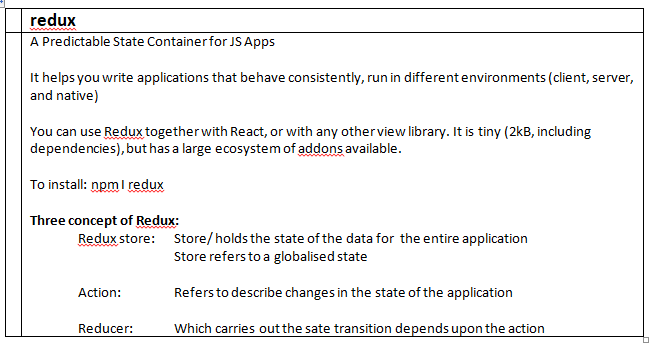
|  |  |
| --- | --- |
|  |  |
|  | import React from 'react';  import EmployeeCrud from './EmployeeCrud';  import { BrowserRouter,Route, Switch,Link } from 'react-router-dom';  import FetchDemo03 from './FetchDemo03';  import AxiosDemo01 from './AxiosDemo01';  class RouterDemoComponent extends React.Component{  render(){  return <div>  <BrowserRouter>  <h2>Routing Demo</h2>    <Link to="/a1">Employee CRUD</Link> |  <Link to="/f1">Fetch Demo</Link> |  <Link to="/a1">Axios Demo</Link> <hr/>  <Switch>  <Route path="/a1">  <EmployeeCrud />  </Route>  <Route path="/f1" >  <FetchDemo03 />  </Route>  <Route path="/a1">  <AxiosDemo01 />  </Route>  </Switch>  </BrowserRouter>  </div>  }  }  export default RouterDemoComponent; |

|  |  |
| --- | --- |
|  | Redirect to specified route |
|  | Demo on login to dashboard    import React from 'react';  class DashboardComponent extends React.Component{  render(){  return <div>  <h2>Dashboard Component</h2>  Hello user, welcome to dashboard  </div>  }  }  export default DashboardComponent; |
|  | //logincomponent.js  import React from 'react';  import {Redirect, BrowserRouter,Route} from 'react-router-dom';  import DashboardComponent from './DashboardComponent';  class LoginComponent extends React.Component{  state={  msg:'',  users:[  {'uname':'sarfaraj','pwd':'sarfaraj'},  {'uname':'gopi','pwd':'gopi'},  {'uname':'arpit','pwd':'arpit'},  {'uname':'hari','pwd':'hari'}  ],  status:false  };    validate(){  var uname = this.refs.uname.value;  var pwd = this.refs.pwd.value;  if ( this.state.users.find( x=>x.uname==uname && x.pwd==pwd ) ){  this.setState({status:true});  }  else{  this.setState({msg:'Invalid input credentials...'});  }  }  render() {  if ( this.state.status==true)  return <div>  <BrowserRouter>  <Redirect to="/dashboard" />  <Route path="/dashboard">  <DashboardComponent />  </Route>  </BrowserRouter>  </div>  return (  <div className={this.state.class1}>  <h2>User Login</h2>  <input type="text" placeholder="User Name" ref="uname" /> <br />  <input type="password" placeholder="Password" ref="pwd" /> <br/>  <button onClick={()=>this.validate()}>Login</button> <hr/>  {this.state.msg} <br/>    </div>  );  }  }  export default LoginComponent; |



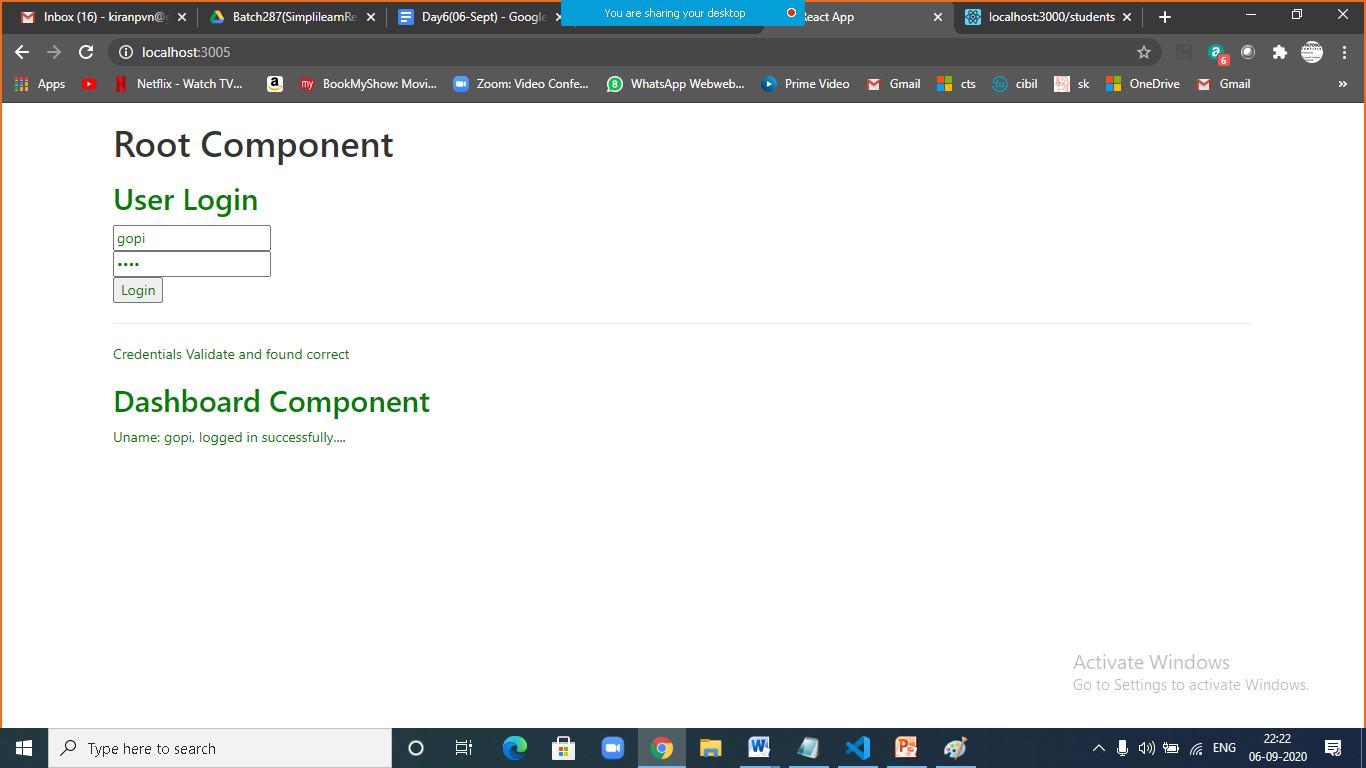




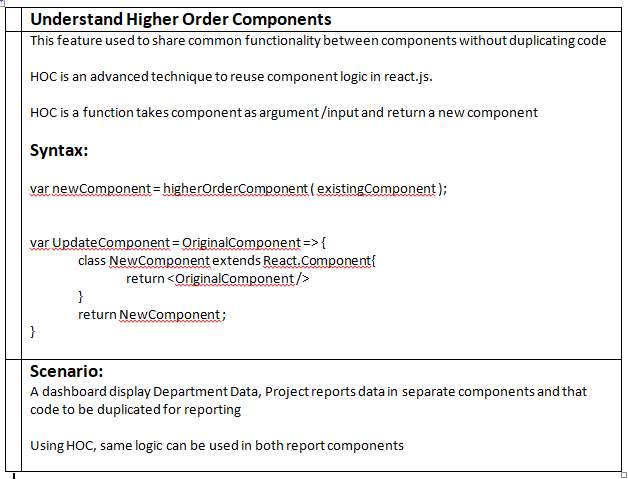


|  |  |
| --- | --- |
|  | Example01 |
|  | //Step1: Declare store for redux  var redux = require("redux");  var createStore = redux.createStore;  //Step3: Define actions  const increment = function(){  return {type:'INCREMENT'};  }  const decrement = () => {  return {type:'DECREMENT'};  }  //Step3: describe actions within reducer  var reducer= (initialState=0, action) => {  switch(action.type){  case "INCREMENT":  return initialState+1;  case "DECREMENT":  return initialState-1;  }  }  //Step4: Create Store  let store = createStore(reducer);  //Step5: Subscribe store  store.subscribe ( ()=>console.log ( store.getState()));  //Step6: Invoke actions  store.dispatch(increment());  store.dispatch(decrement()); |

|  |  |
| --- | --- |
|  | //a1.js  //Step1: declare store  import { createStore } from 'redux';  //Step2: Create actions  export const increment = () => {return {type:'INCREMENT'}};  export const decrement = () => {return {type:'DECREMENT'}};  //Step3: Create reducer by describing actions  export const reducer = (InitialState=0, action) => {  switch(action.type){  case 'INCREMENT':  return InitialState+1;  case 'DECREMENT':  return InitialState-1;  default:  return InitialState;  }  }  //Step4: Create store  export let store = createStore(reducer);  //Step5: Subscribe store  store.subscribe(() => console.log(store.getState())); |
|  | //a2.js  import {store,increment,decrement} from './a1';  import React from 'react';  class A2 extends React.Component{  state={msg:''};  componentDidMount(){  this.setState({msg:store.getState()});  }  incrementMethod(){  store.dispatch(increment());  this.setState({msg:store.getState()});  }  decrementMethod(){  store.dispatch(decrement());  this.setState({msg:store.getState()});  }  render(){  return <div>  <h2>Redux Demo</h2>  <button onClick={()=>this.incrementMethod()}>Increment</button>  <button onClick={()=>this.decrementMethod()}>Decrement</button> <br/>  Current State: {this.state.msg}  </div>  }  }  export default A2; |
|  | //app.js  import React from 'react';  import logo from './logo.svg';  import './App.css';  import A2 from './a2';  function App() {  return (  <div className="container">  <h1>Root Component</h1>  <A2 />  <A2 />  </div>  );  }  export default App; |
|  |  |



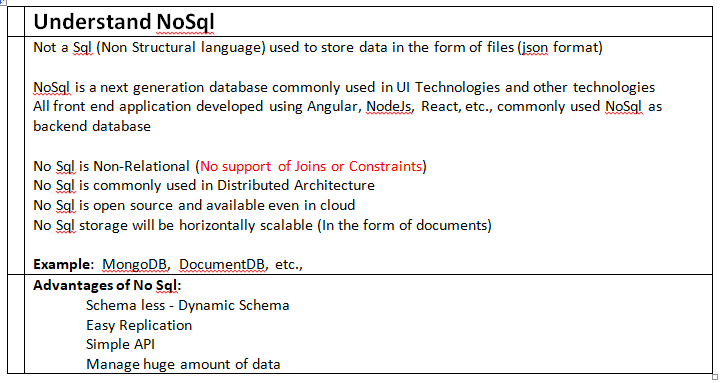
|  |  |
| --- | --- |
|  |  |
|  | //sessionStore.js  import {createStore} from 'redux';  export const login = (uname) => { return {type:'login',uname:uname}};  export const logout = () => { return {type:'logout'}};  export const reducer = (initialState='logout', action) => {  switch(action.type){  case 'login':  initialState=`${action.uname}, logged in successfully....`;  break;  case'logout':  initialState='logout';  break;  }  return initialState;  }  export let store = createStore(reducer);  store.subscribe(() => console.log(store.getState())); |
|  | //LoginComponent.js  import React from 'react';  import {store,login,logout} from './sessionStore';  import DashboardComponent from './DashboardComponent';  import DummyComponent from './DummyComponent';  class LoginComponent extends React.Component{  state={  msg:'',  users:[  {'uname':'sarfaraj','pwd':'sarfaraj'},  {'uname':'gopi','pwd':'gopi'},  {'uname':'arpit','pwd':'arpit'},  {'uname':'hari','pwd':'hari'}  ],  status:'invalid'  };    validate(){  var uname = this.refs.uname.value;  var pwd = this.refs.pwd.value;  if ( this.state.users.find( x=>x.uname==uname && x.pwd==pwd ) ){  store.dispatch(login(uname));  this.setState({msg:'Credentials Validate and found correct', class1:'successClass',status:'valid'});    }  else{  store.dispatch(logout());  this.setState({msg:'Invalid input Credentials ', class1:'failureClass',status:'invalid'});  }  }  render() {  return (  <div className={this.state.class1}>  <h2>User Login</h2>  <input type="text" placeholder="User Name" ref="uname" /> <br />  <input type="password" placeholder="Password" ref="pwd" /> <br/>  <button onClick={()=>this.validate()}>Login</button> <hr/>  {this.state.msg} <br/>  { this.state.status=='valid' ? <DashboardComponent /> : <DummyComponent /> }  </div>  );  }  }  export default LoginComponent; |
|  | //Dashboard.js  import React from 'react';  import {store} from './sessionStore';  class DashboardComponent extends React.Component{  state={uname:''};  componentDidMount(){  this.setState({uname:store.getState()})  }  render(){  return <div>  <h2>Dashboard Component</h2>  Uname: {this.state.uname}  </div>  }  }  export default DashboardComponent; |
|  | //app.js  import React from 'react';  import logo from './logo.svg';  import './App.css';  import LoginComponent from './LoginComponent';  function App() {  return (  <div className="container">  <h1>Root Component</h1>  <LoginComponent />  </div>  );  }  export default App; |

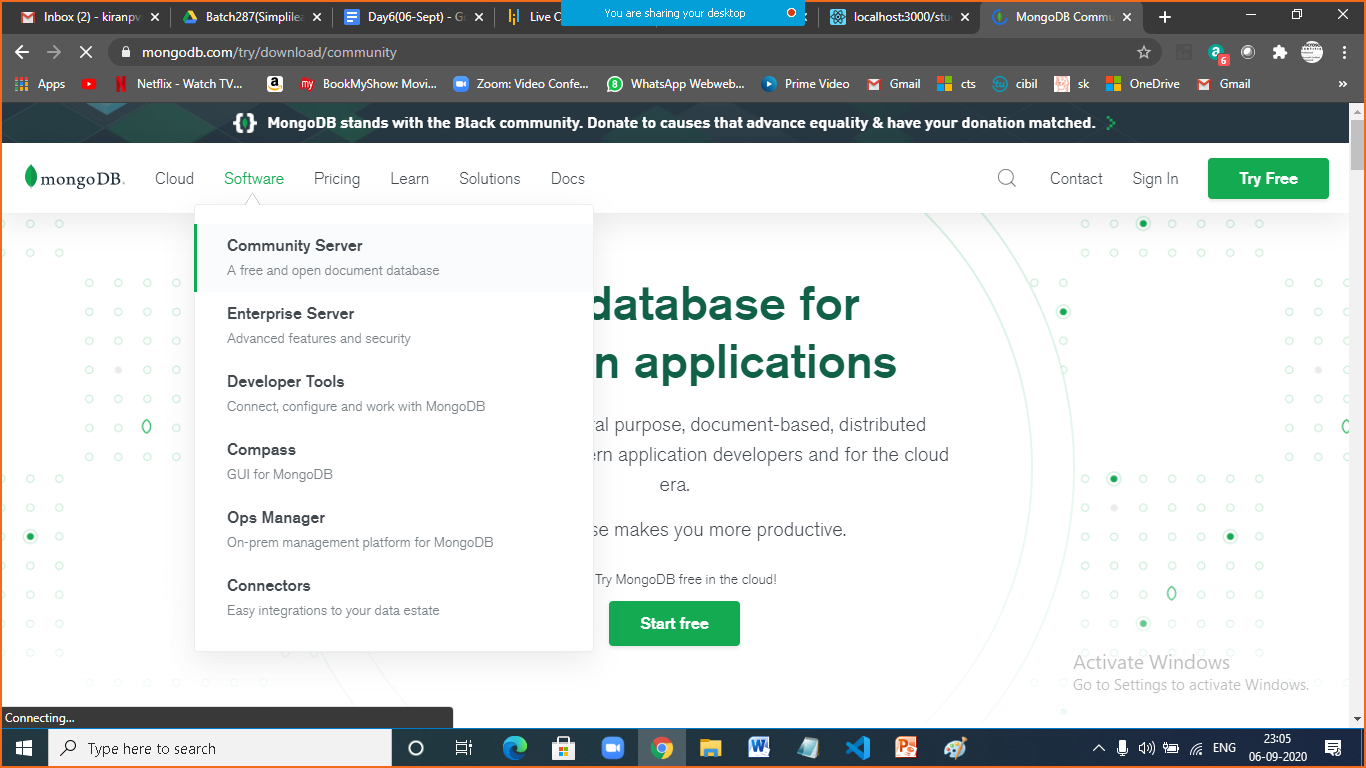
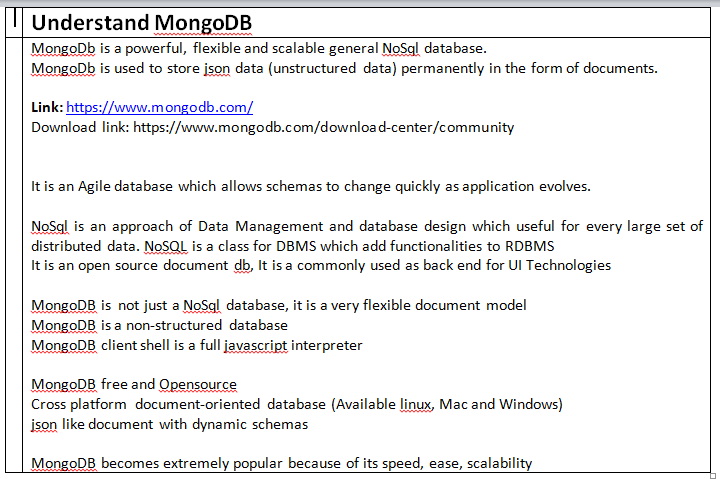


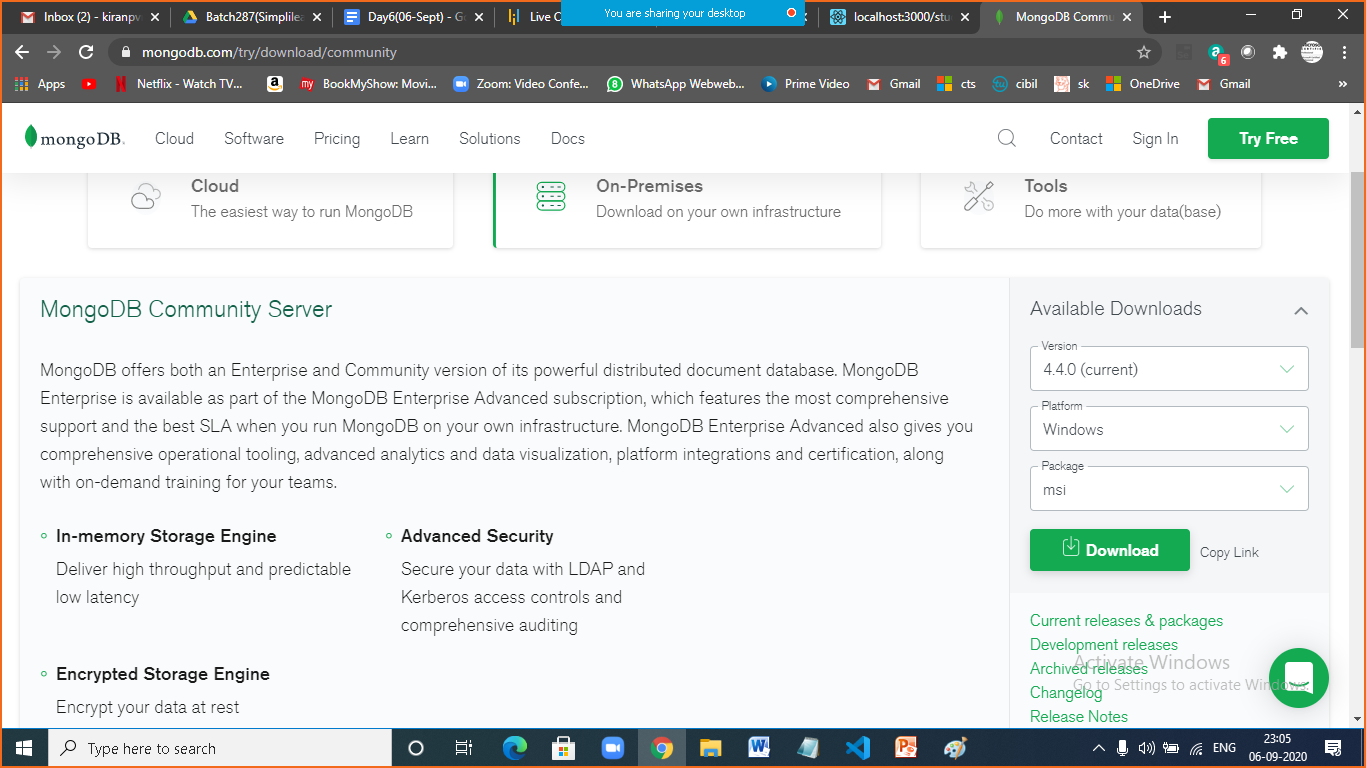
|  |  |
| --- | --- |
|  | **Example without HOC** |
|  | //ClickCounter.js  import React from 'react';  class ClickCounter extends React.Component {  state={count:0}  increment(){  this.setState( previousState => {return {count:previousState.count+1} });  }  render() {  return (  <div>  <h1>Button Click Counter Component</h1>  <button onClick={()=>this.increment()}>Increment</button> <br/>    Counter Value: {this.state.count}  </div>  );  }  }  export default ClickCounter; |
|  | //Hovercounter.js  import React, { Component } from 'react';  class HoverCounter extends Component {  state={count:0};  incrementCount(){  this.setState( previousState => {return {count:previousState.count+1} });  }  render() {  return (  <div >  <h1 onMouseOver={()=>this.incrementCount()}>Mouse Hover Counter Component</h1>  Mouse hover {this.state.count} times  </div>  );  }  }  export default HoverCounter; |
|  | //App.js  import React from 'react';  import logo from './logo.svg';  import './App.css';  import ClickCounter from './ClickCounter';  import HoverCounter from './HoverCounter';  function App() {  return (  <div className="container">  <h1>Root Component</h1>  <ClickCounter />  <HoverCounter />  </div>  );  }  export default App; |

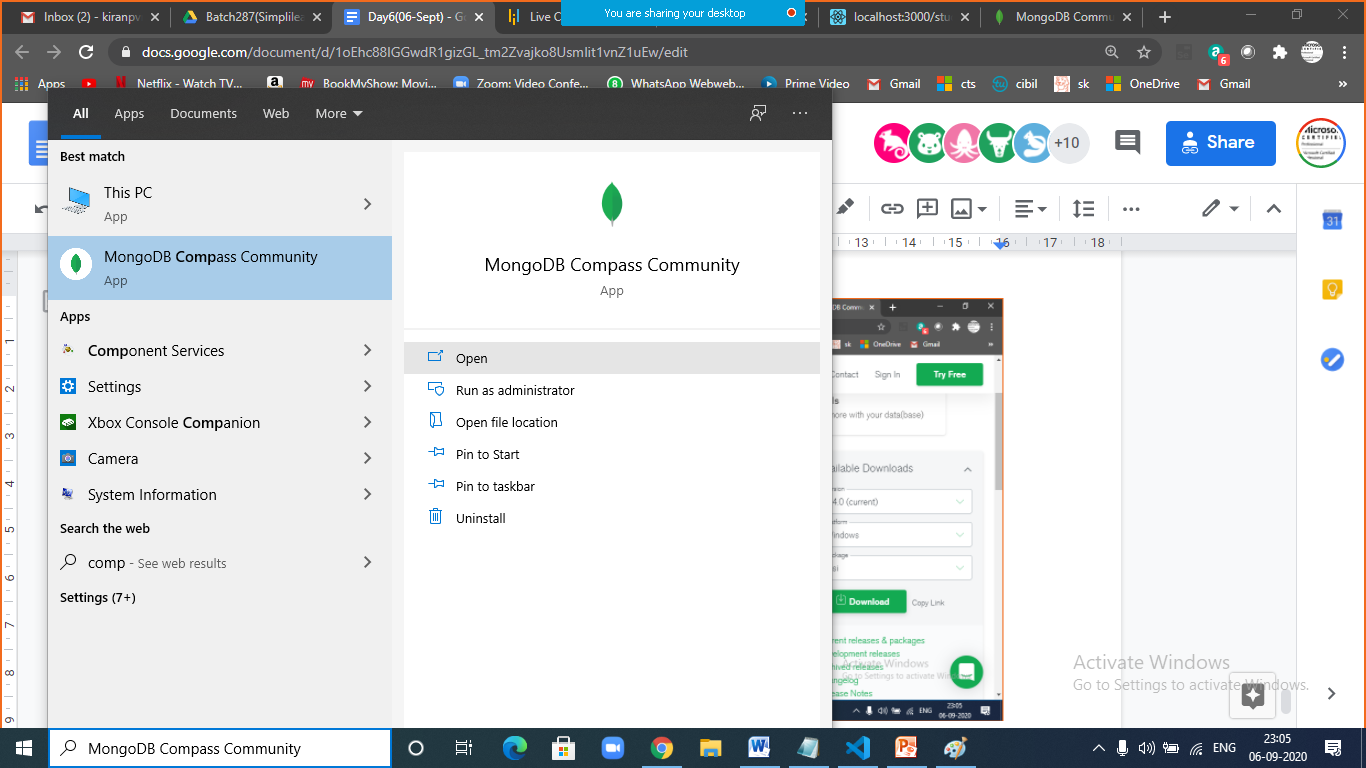
==========================

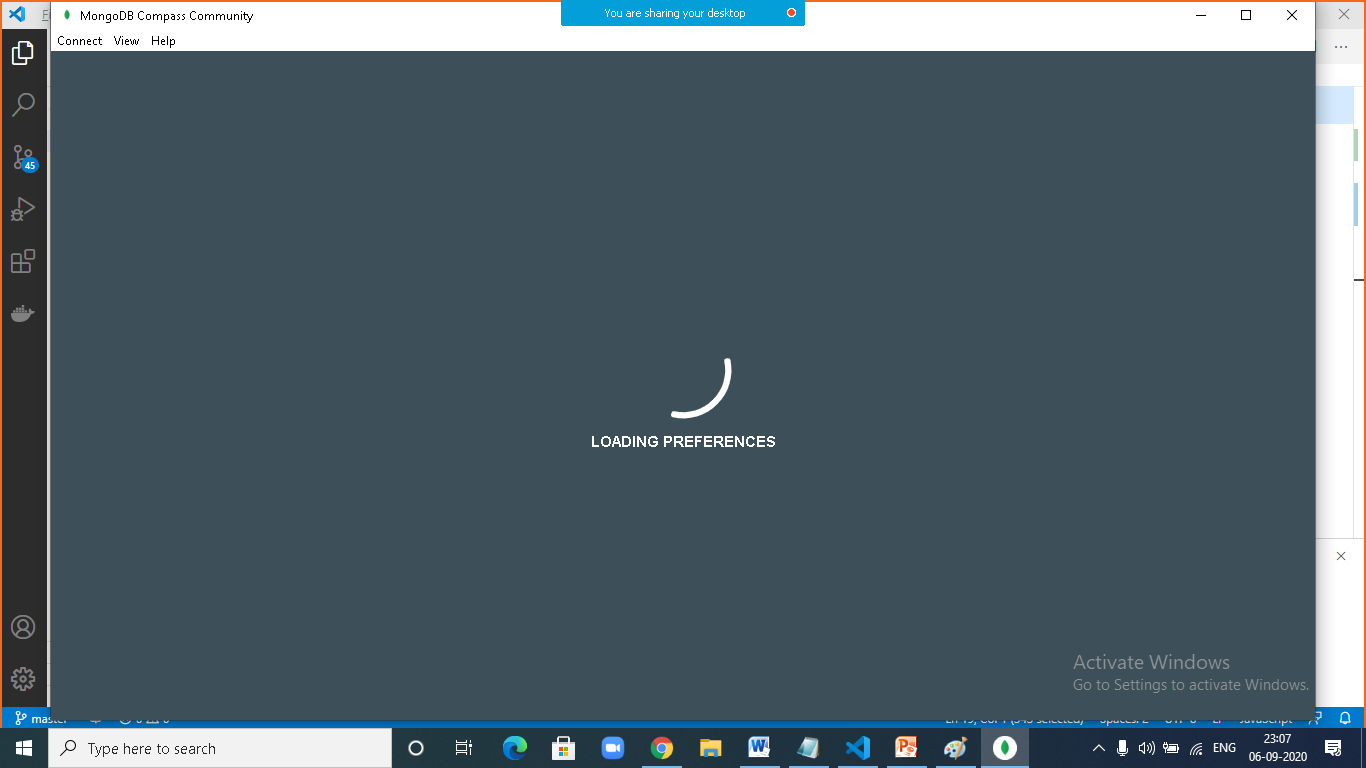
|  |  |
| --- | --- |
|  | //HocComponent.js  import React from 'react';  var UpdatedComponent = OriginalComponent =>{  class NewComponent extends React.Component{  state={count:0}  incrementCount = () =>{  this.setState( previousState => {return {count:previousState.count+1} });  }  render(){  return <OriginalComponent count={this.state.count} incrementCount={this.incrementCount} />  }  }  return NewComponent;  }  export default UpdatedComponent; |
|  | //ClickCounter.js  import React, { Component } from 'react';  import UpdatedComponent from './HocComponent';  class ClickCounter extends Component {    render() {  var {count,incrementCount} = this.props;  return (  <div>  <h1>Button Click Counter Component</h1>  <button onClick={()=>incrementCount()}>Increment</button>  {count}  </div>  );  }  }  export default UpdatedComponent( ClickCounter); |
|  | //HoverCounter.js  import React, { Component } from 'react';  import UpdatedComponent from './HocComponent';  class HoverCounter extends Component {    render() {  var {count,incrementCount} =this.props;  return (  <div >  <h1 onMouseOver={()=>incrementCount()}>Mouse Hover Counter Component</h1>  Mouse hover {count} times  </div>  );  }  }  export default UpdatedComponent (HoverCounter); |
|  | //App.js  import React from 'react';  import logo from './logo.svg';  import './App.css';  import ClickCounter from './ClickCounter';  import HoverCounter from './HoverCounter';  function App() {  return (  <div className="container">  <h1>Root Component</h1>  <ClickCounter />  <HoverCounter />  </div>  );  }  export default App; |

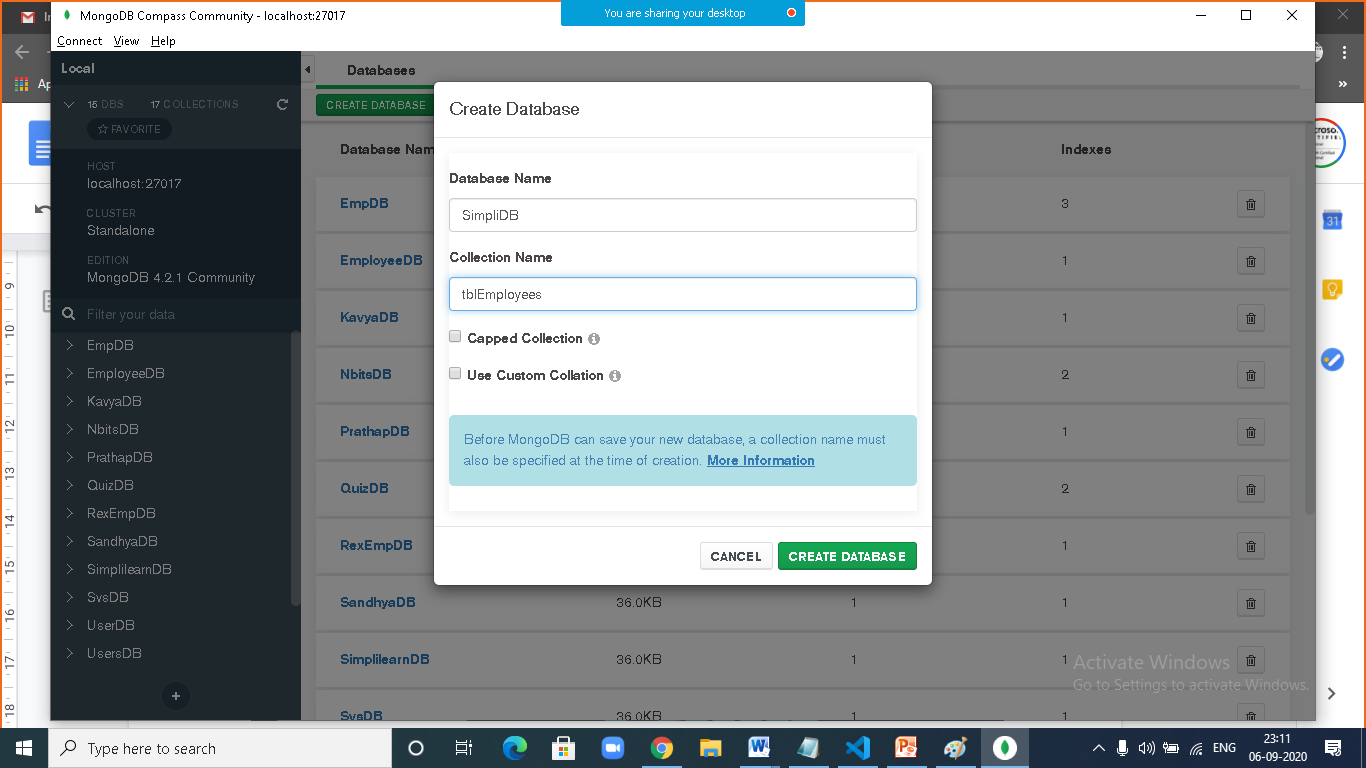
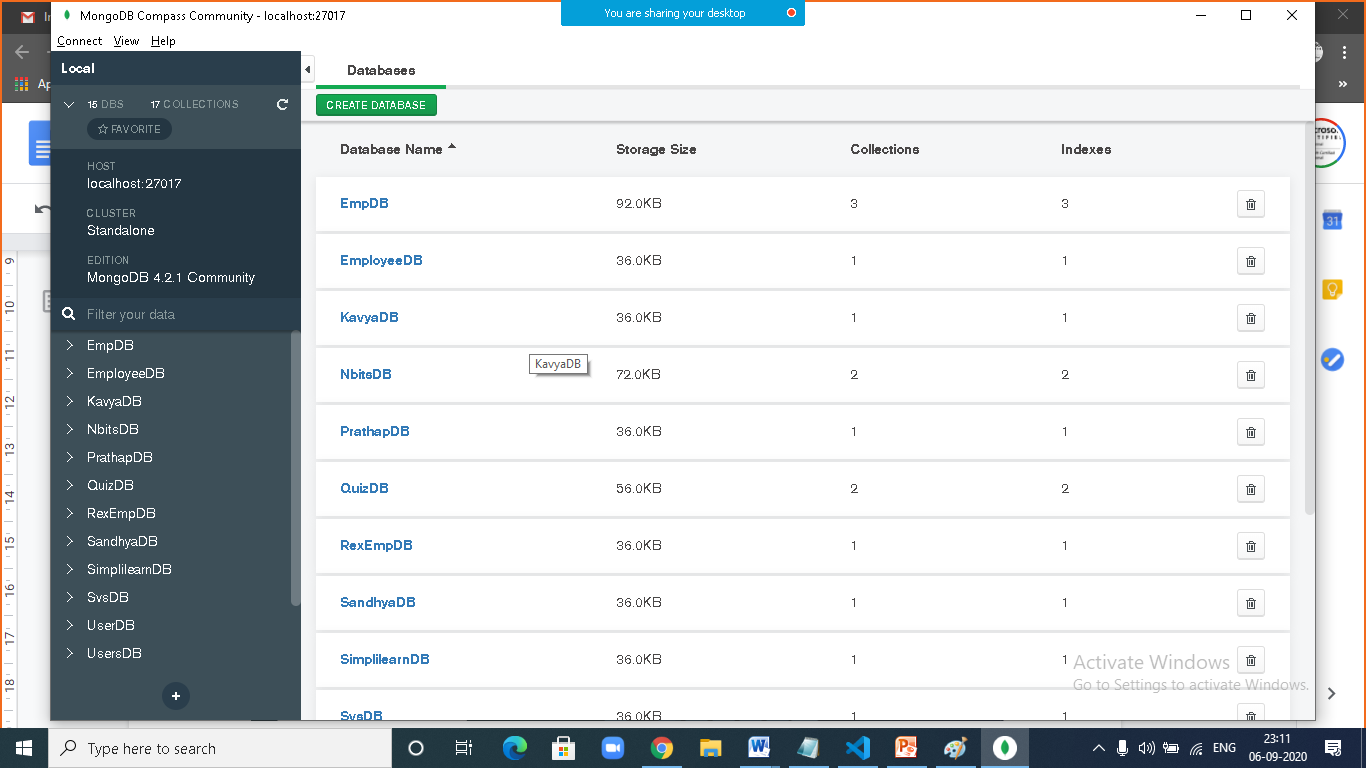
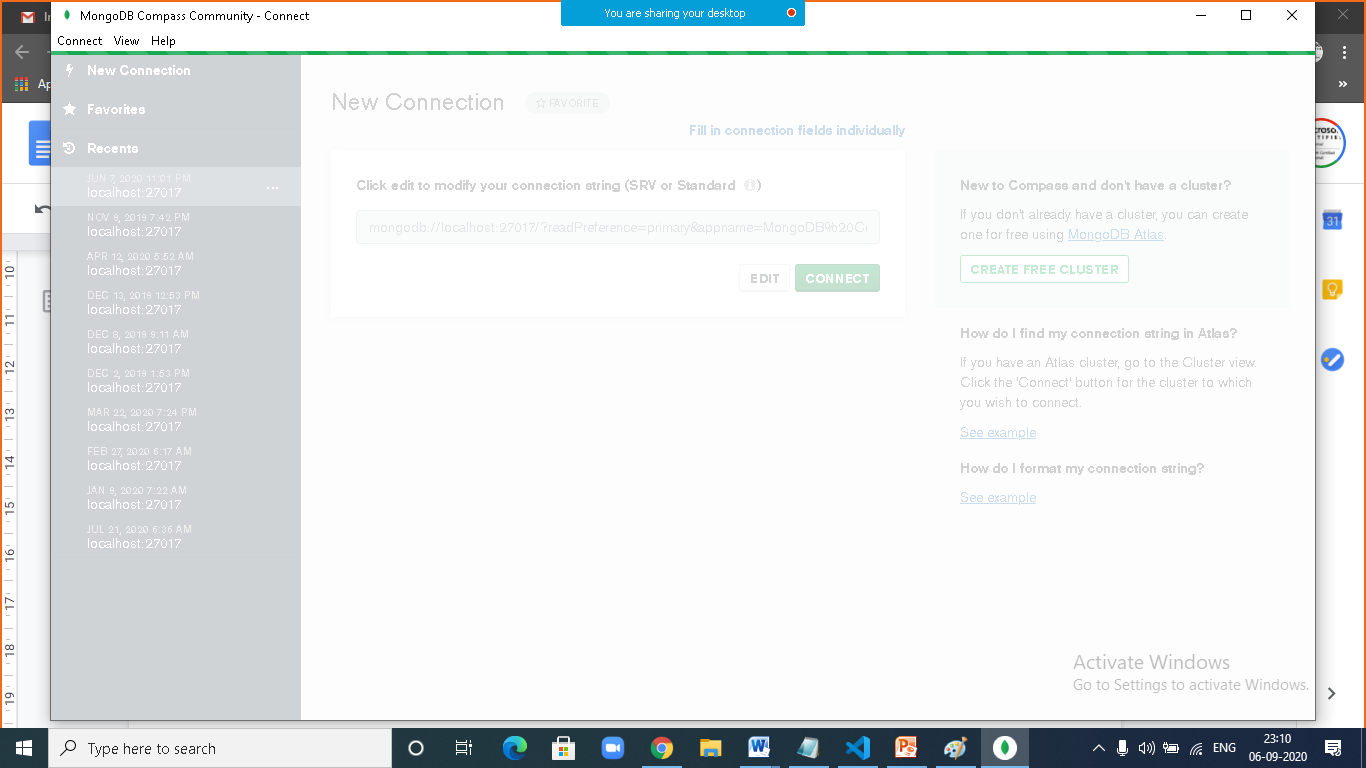


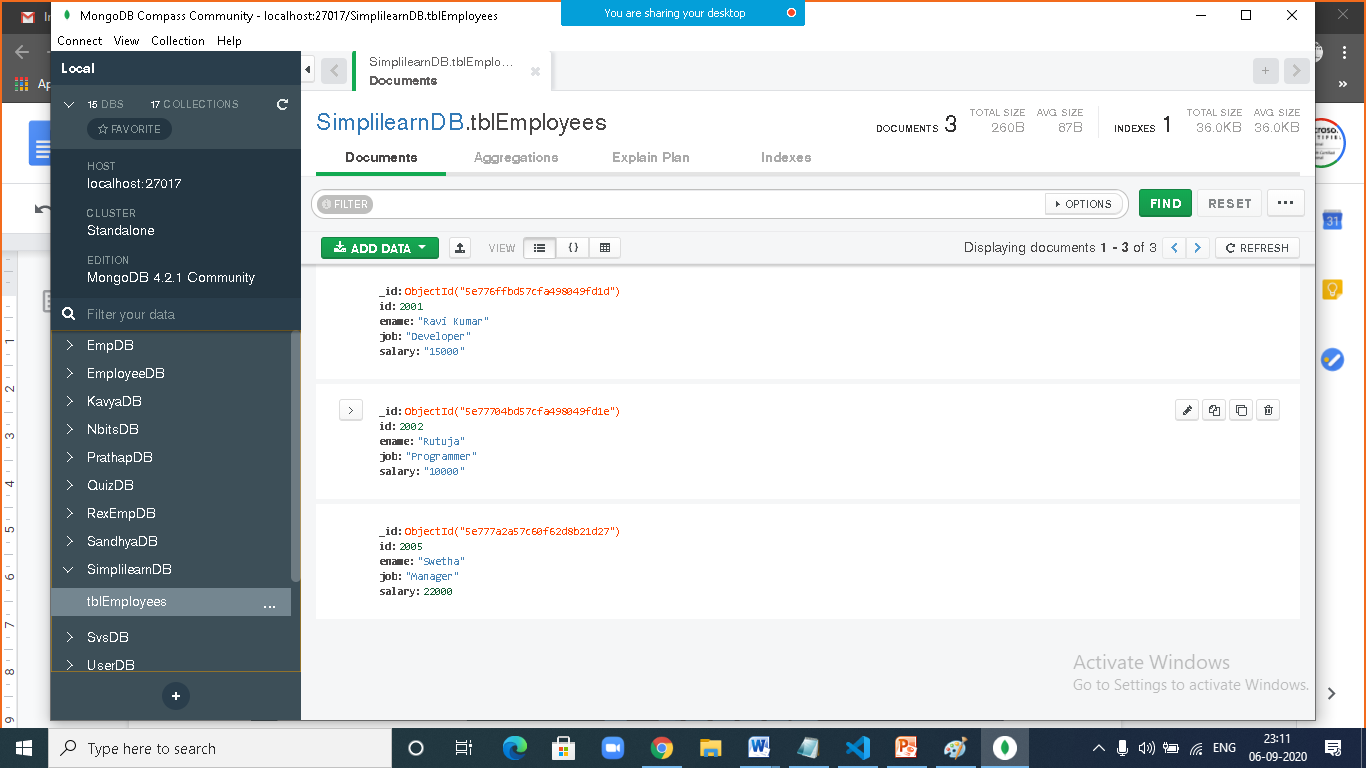












Web API with MongoDB

CRUD

var app = require("express")();

var MongoClient = require('mongodb').MongoClient;

app.get('/',function(req,resp){

MongoClient.connect("mongodb://localhost:27017/KiranDB", function(err, db) {

db.collection('employees').find().toArray(function(err, docs){

resp.send(docs);

});

});

});

app.get('/:x',function(req,resp){

var id =Number(req.params.x);

MongoClient.connect("mongodb://localhost:27017/kirandb", function(err, db) {

var collection = db.collection("employees");

collection.findOne({\_id:id},function(err,doc){

resp.send( doc);

});

});

});

app.listen(9001,function(){

console.log("ExpressJS started listening");

});

|  |  |  |
| --- | --- | --- |
|  | **Get Access using jQuery** | |
|  | Server.js | var express = require("express");  var app = express();  var MongoClient = require('mongodb').MongoClient;  var path = require("path");    var bodyparser = require("body-parser");    var allowCrossDomain = function(req, res, next) {  res.header("Access-Control-Allow-Origin","\*");  res.header("Access-Control-Allow-Methods:GET,PUT,POST,DELETE");  res.header("Access-Control-Allow-Headers:Content-Type,X- Requested-With");  next();  };  app.use(allowCrossDomain);    app.get('/',function(req,resp){  MongoClient.connect("mongodb://localhost:27017/kirandb", function(err, db) {  db.collection('employees').find().toArray(function(err, docs){  resp.send(docs);  });  });  });  app.use(bodyparser.json()); // support json encoded bodies  app.post('/',function(req,resp){  console.log(req.param('ename'));  resp.send(req.param('ename'));  })  app.listen(9001,function(){  console.log("ExpressJS started listening");  }); |
|  | A1.html | <h1 style="text-align:center">Web Api Consume </h1>  <input type="text" id="empid" />  <input type="button" value="Search" onclick="find()" />  <ul id="items"/>  <script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>  <script>  $(document).ready(function () {  $.getJSON("http://localhost:9001/", function (data) {  $.each(data,function(key,val){  var st = val;  $('<li/>', { text: st.ename }).appendTo($("#items"));  })  })  });  </script> |

|  |  |  |
| --- | --- | --- |
|  | **Get Access Using jQuery** | |
|  | Server.js | var express = require("express");  var app = express();  var MongoClient = require('mongodb').MongoClient;  var path = require("path");    var bodyparser = require("body-parser");    var allowCrossDomain = function(req, res, next) {  res.header("Access-Control-Allow-Origin","\*");  res.header("Access-Control-Allow-Methods:GET,PUT,POST,DELETE");  res.header("Access-Control-Allow-Headers:Content-Type,X- Requested-With");  next();  };    app.use(allowCrossDomain);    app.use(bodyparser.json()); // support json encoded bodies    app.get('/:x',function(req,resp){  var id =Number(req.params.x);  MongoClient.connect("mongodb://localhost:27017/kirandb", function(err, db) {  var collection = db.collection("employees");  collection.findOne({\_id:id},function(err,doc){  resp.send( doc);  });  });    });    app.listen(9001,function(){  console.log("ExpressJS started listening");  }); |
|  | A1.html | <h1 style="text-align:center">Web Api Consume </h1>    <button onclick='ShowEmployees()'>Show Employees</button>  <ul id="items"/>  <script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>  <script>  function ShowEmployees () {  $.getJSON("http://localhost:9001/", function (data) {  $.each(data,function(key,val){  var st =val.\_id+' '+ val.ename+' '+val.job+' '+val.salary;  $('<li/>', { text: st }).appendTo($("#items"));  })  })  }  </script> |

|  |  |  |
| --- | --- | --- |
|  | Search row | |
|  | Server.js | var express = require("express");  var app = express();  var MongoClient = require('mongodb').MongoClient;  var path = require("path");  var bodyparser = require("body-parser");  var cors = require("cors");  app.use(cors());    app.get('/:x',function(req,resp){  var id =Number(req.params.x);  MongoClient.connect("mongodb://localhost:27017/kirandb", function(err, db) {  var collection = db.collection("employees");  collection.findOne({\_id:id},function(err,doc){  resp.send( doc);  });  });  });  app.listen(9001,function(){  console.log("ExpressJS started listening");  }); |
|  | A1.html | <h1 style="text-align:center">Web Api Consume </h1>    <input type="text" id="id">  <button onclick='find()'>Search</button>  <ul id="items"/>  <script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>  <script>  function find() {  var id = $("#id").val();  $.getJSON("http://localhost:9001/" + id, function (data) {  $("#items").text(data.ename+' '+data.job+' '+data.salary);  });  }  </script> |

|  |  |
| --- | --- |
|  | **Adding row into mongodb database** |
| Server.js | var express = require("express");  var app = express();  var MongoClient = require('mongodb').MongoClient;  var path = require("path");    var bodyparser = require("body-parser");  var cors = require("cors");  app.use(cors());  app.use(bodyparser.json()); // support json encoded bodies  app.post('/',function(req,resp){  MongoClient.connect("mongodb://localhost:27017/kirandb", function(err, db) {  db.collection('employees').insertOne(req.body,function(err, docs){  if (!err)  resp.send("Row added successfully");  else resp.send("Error " +err);  });  })  });  app.listen(9001,function(){  console.log("ExpressJS started listening");  }); |
| A1.html | <h1 style="text-align:center">Web Api Consume </h1>    ID: <input type="number" id="id"> <br>  Ename: <input type="text" id="ename"> <br>  Job: <input type="text" id="job"> <br>  Salary: <input type="number" id="salary"> <br>  <button onclick='find()'>Search</button>  <button onclick='add()'>Add Row</button> <hr>  <ul id="items"/> </ul>  <hr>  <script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>  <script>  function find() {  var id = $("#id").val();  $.getJSON("http://localhost:9001/" + id, function (data) {  $("#items").text(data.ename+' '+data.job+' '+data.salary);  });  }  function add(){  var emp = {  \_id: $("#id").val(),  ename: $("#ename").val(),  job: $("#job").val(),  salary: $("#salary").val()  };  $.ajax({  url: 'http://localhost:9001/',  type: 'POST',  data: JSON.stringify(emp),  contentType: "application/json;charset=utf-8",  success: function (data) {  alert('Row added Successfully');  },  error: function () {  alert('Row failed to add');  }  });  }  </script> |