**MERN Setup, Development, and Deployment**

These instructions are for Linux, specifically Ubuntu 24 was used.

Install Visual Studio Code

Install Chrome and Postman

--------- Install Node

sudo apt-get update

sudo apt-get install nodejs

sudo apt-get install npm

--------- update node

sudo npm cache clean -f

sudo npm install -g n

sudo n stable

------- install the react app creator and install nodemon (nodemon allows the server component to auto-restart after a change)

sudo npm install -g nodemon

Create app directory (suggest cards)

mkdir cards

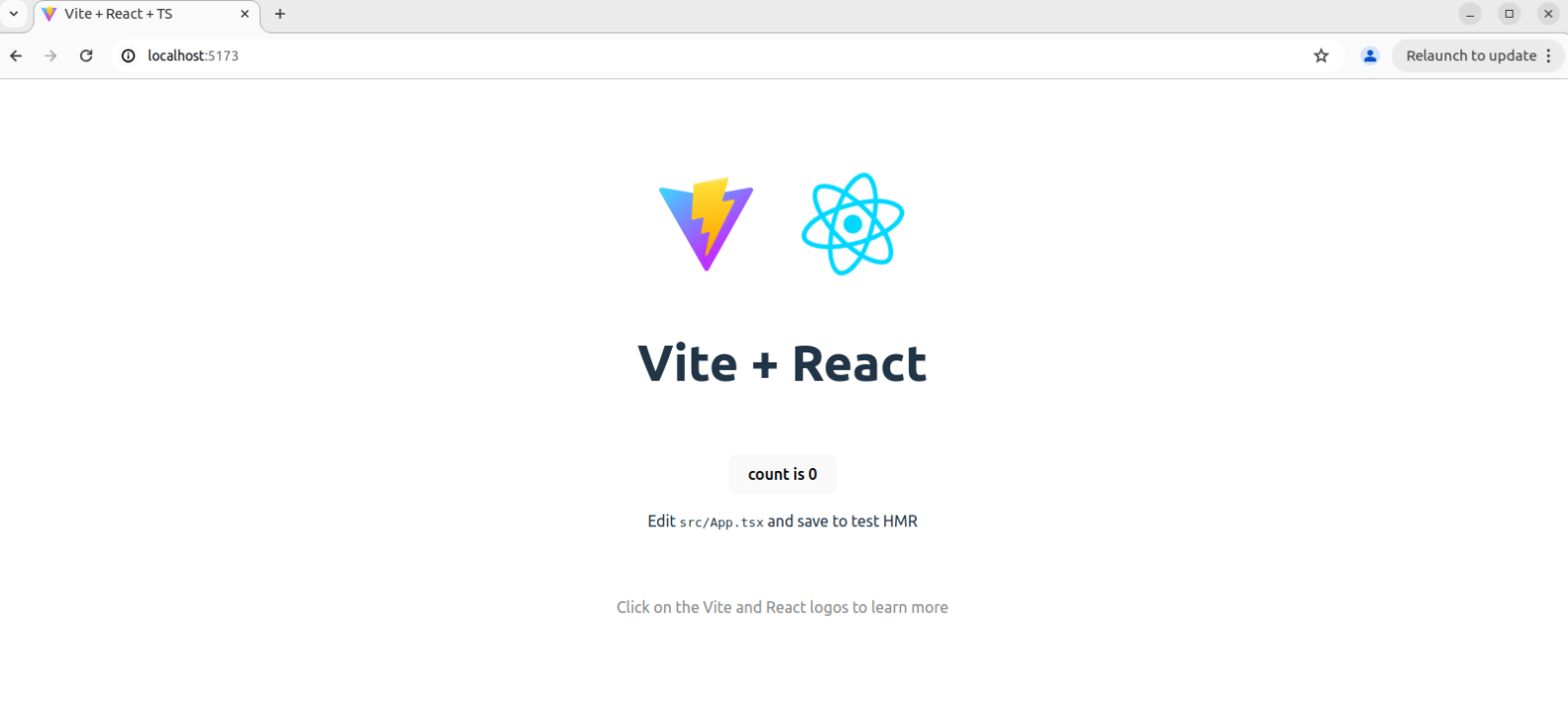
cd cards

------------ create the **cards** app

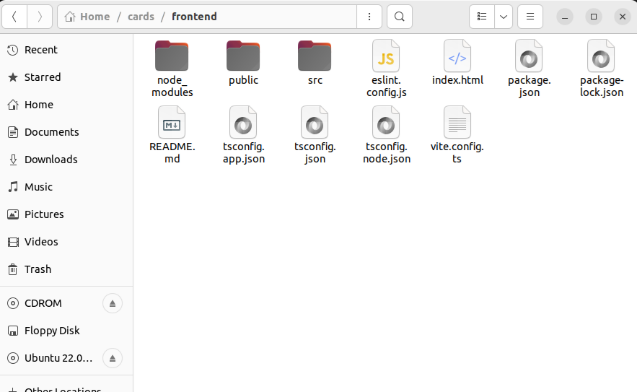
**In main directory (cards)**

npm create vite@latest frontend (It may ask you to install a package -> select React / Typescript)

to test enter **cd frontend** and then **npm install** and then **npm run dev** then run a browser and load **http://localhost:5173**



Here is the file hierarchy,



Phase 1: A simple UI edit  ------------------------------------------------------------------------------

To frontend/src add **components** and **pages** directories

Open directory in Visual Studio Code. From cards directory type **code .**  Yes the period/dot goes with the command.

Add two components

**PageTitle.tsx (save into frontend/src/components)**

function PageTitle()

{

return(

<h1 id="title">COP 4331 MERN Stack Demo</h1>

);

};

export default PageTitle;

**Login.tsx (save into frontend/src/components)**

function Login()

{

function doLogin(event:any) : void

{

event.preventDefault();

alert('doIt()');

}

return(

<div id="loginDiv">

<span id="inner-title">PLEASE LOG IN</span><br />

<input type="text" id="loginName" placeholder="Username" /><br />

<input type="password" id="loginPassword" placeholder="Password" /><br />

<input type="submit" id="loginButton" className="buttons" value = "Do It"

onClick={doLogin} />

<span id="loginResult"></span>

</div>

);

};

export default Login;

**Add a page**

**LoginPage.tsx (save into pages)**

import PageTitle from '../components/PageTitle.tsx';

import Login from '../components/Login.tsx';

const LoginPage = () =>

{

return(

<div>

<PageTitle />

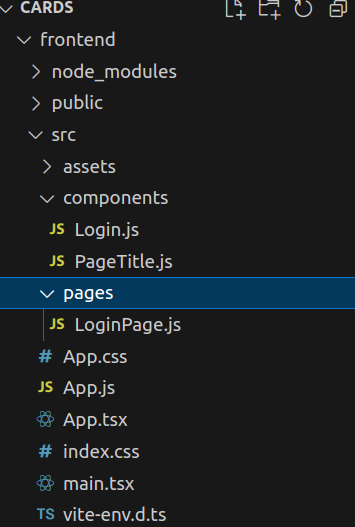
<Login />

</div>

);

};

export default LoginPage;

\

**Finally edit App.tsx**

import './App.css';

import LoginPage from './pages/LoginPage.tsx';

function App()

{

return (

<LoginPage />

);

}

export default App;

***Test… (npm run dev) then run a browser and connect to http://localhost:5173/***

Add backend –(phase1a)---------------------------------------------------

Create server.js and package.json

**Add the following to server.js (this goes into the root of the mern app, outside of the frontend directory)**

const express = require('express');

const bodyParser = require('body-parser');

const cors = require('cors');

const app = express();

app.use(cors());

app.use(bodyParser.json());

app.use((req, res, next) =>

{

  res.setHeader('Access-Control-Allow-Origin', '\*');

  res.setHeader(

    'Access-Control-Allow-Headers',

    'Origin, X-Requested-With, Content-Type, Accept, Authorization'

  );

  res.setHeader(

    'Access-Control-Allow-Methods',

    'GET, POST, PATCH, DELETE, OPTIONS'

  );

  next();

});

app.listen(5000); // start Node + Express server on port 5000

**Add the following to package.json**

{

  "name": "backend",

  "version": "1.0.0",

  "description": "",

  "main": "server.js",

  "scripts": {

    "test": "echo \"Error: no test specified\" && exit 1",

    "start": "nodemon server.js"

  },

  "author": "",

  "license": "ISC",

  "dependencies": {

    "body-parser": "^1.19.0",

    "express": "^4.17.1"

  }

}

**In root of the mern app directory**

sudo npm install

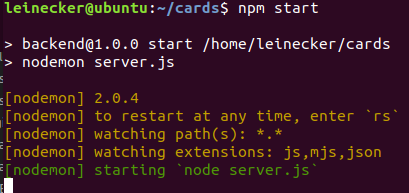
sudo npm install express --save

sudo npm install body-parser

sudo npm install mongodb

sudo npm install cors

***Test… (npm start)***



***Please note: To use this webapp (when the front end needs to contact the API) you must run BOTH the server (which we just ran) and the front end (which we ran first).***

Phase 2: Add baseball card page to friont end---------------------------------------------------------------------------------

**Add component LoggedInName.tsx**

function LoggedInName()

{

var user={}

function doLogout(event:any) : void

{

event.preventDefault();

alert('doLogout');

};

return(

<div id="loggedInDiv">

<span id="userName">Logged In As John Doe </span><br />

<button type="button" id="logoutButton" className="buttons"

onClick={doLogout}> Log Out </button>

</div>

);

};

export default LoggedInName;

**Add component CardUI.tsx**

function CardUI()

{

function addCard(event:any) : void

{

event.preventDefault();

alert('addCard()');

};

function searchCard(event:any) : void

{

event.preventDefault();

alert('searchCard');

};

return(

<div id="accessUIDiv">

<br />

<input type="text" id="searchText" placeholder="Card To Search For" />

<button type="button" id="searchCardButton" className="buttons"

onClick={searchCard}> Search Card </button><br />

<span id="cardSearchResult"></span>

<p id="cardList"></p><br /><br />

<input type="text" id="cardText" placeholder="Card To Add" />

<button type="button" id="addCardButton" className="buttons"

onClick={addCard}> Add Card </button><br />

<span id="cardAddResult"></span>

</div>

);

}

export default CardUI;

**Create page CardPage.tsx**

import PageTitle from '../components/PageTitle';

import LoggedInName from '../components/LoggedInName';

import CardUI from '../components/CardUI';

const CardPage = () =>

{

return(

<div>

<PageTitle />

<LoggedInName />

<CardUI />

</div>

);

}

export default CardPage;

**Edit App.tsx**

import './App.css';

import LoginPage from './pages/LoginPage.tsx';

import CardPage from './pages/CardPage.tsx';

function App()

{

return (

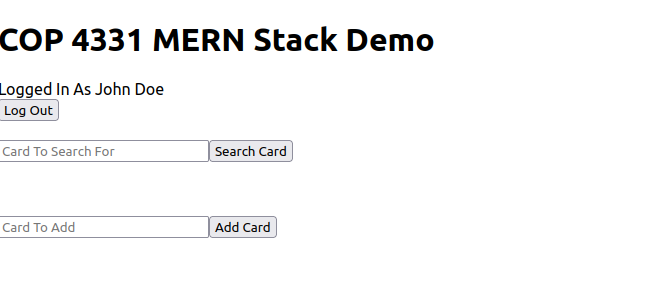
<CardPage />

);

}

export default App;

***Test… (npm run dev)***



**Routing (phase 2a)**

-- from frontend folder

npm install react-router-dom

**Edit App.tsx**

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

import './App.css';

import LoginPage from './pages/LoginPage';

import CardPage from './pages/CardPage';

function App() {

return (

<Router >

<Routes>

<Route path="/" element={<LoginPage />} />

<Route path="/cards" element={<CardPage />} />

</Routes>

</Router>

);

}

export default App;

***Test… (npm run dev)***

**Input field persistence (phase 3)**

To **Login.tsx** and **CardUI.tsx** add **import React, { useState } from 'react';**

Add to Login function

const [message,setMessage] = React.useState('');

const [loginName,setLoginName] = React.useState('');

const [loginPassword,setPassword] = React.useState('');

Edit input fields for input data

Login: <input type="text" id="loginName" placeholder="Username"

onChange={handleSetLoginName} />

Password: <input type="password" id="loginPassword" placeholder="Password"

onChange={handleSetPassword} />

Edit span for output label

<span id="loginResult">{message}</span>

Add Two Functions

function handleSetLoginName( e: any ) : void

{

setLoginName( e.target.value );

}

function handleSetPassword( e: any ) : void

{

setPassword( e.target.value );

}

Edit doLogin

    function doLogin(event:any) : void

    {

        event.preventDefault();

        alert('doIt() ' + loginName + ' ' + loginPassword );

    };

Add to CardUI.tsx

import React, { useState } from 'react';

Add into CardUI function

const [message,setMessage] = React.useState('');

const [searchResults,setResults] = React.useState('');

const [cardList,setCardList] = React.useState('');

const [search,setSearchValue] = React.useState('');

const [card,setCardNameValue] = React.useState('');

Edit UI as follows:

<div id="cardUIDiv">

<br />

Search: <input type="text" id="searchText" placeholder="Card To Search For"

onChange={handleSearchTextChange} />

<button type="button" id="searchCardButton" className="buttons"

onClick={searchCard}> Search Card</button><br />

<span id="cardSearchResult">{searchResults}</span>

<p id="cardList">{cardList}</p><br /><br />

Add: <input type="text" id="cardText" placeholder="Card To Add"

onChange={handleCardTextChange} />

<button type="button" id="addCardButton" className="buttons"

onClick={addCard}> Add Card </button><br />

<span id="cardAddResult">{message}</span>

</div>

Add Two Functions

function handleSearchTextChange( e: any ) : void

{

setSearchValue( e.target.value );

}

function handleCardTextChange( e: any ) : void

{

setCardNameValue( e.target.value );

}

Edit addCard and searchCard as follows:

function addCard(e:any) : void

{

e.preventDefault();

alert('addCard() ' + card);

};

function searchCard(e:any) : void

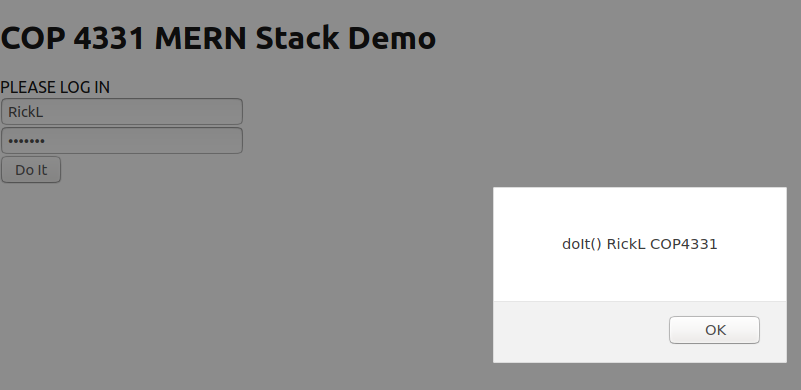
{

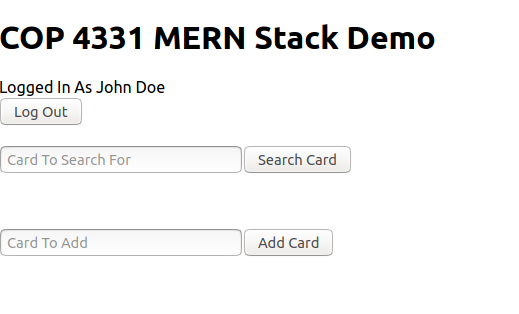
e.preventDefault();

alert('searchCard() ' + search);

};

***Test... (npm run dev)***





**API without DB access (phase 3a)**

**Open server.js**

Add the following:

var cardList =

[

  'Roy Campanella',

  'Paul Molitor',

  'Tony Gwynn',

  'Dennis Eckersley',

  'Reggie Jackson',

  'Gaylord Perry',

  'Buck Leonard',

  'Rollie Fingers',

  'Charlie Gehringer',

  'Wade Boggs',

  'Carl Hubbell',

  'Dave Winfield',

  'Jackie Robinson',

  'Ken Griffey, Jr.',

  'Al Simmons',

  'Chuck Klein',

  'Mel Ott',

  'Mark McGwire',

  'Nolan Ryan',

  'Ralph Kiner',

  'Yogi Berra',

  'Goose Goslin',

  'Greg Maddux',

  'Frankie Frisch',

  'Ernie Banks',

  'Ozzie Smith',

  'Hank Greenberg',

  'Kirby Puckett',

  'Bob Feller',

  'Dizzy Dean',

  'Joe Jackson',

  'Sam Crawford',

  'Barry Bonds',

  'Duke Snider',

  'George Sisler',

  'Ed Walsh',

  'Tom Seaver',

  'Willie Stargell',

  'Bob Gibson',

  'Brooks Robinson',

  'Steve Carlton',

  'Joe Medwick',

  'Nap Lajoie',

  'Cal Ripken, Jr.',

  'Mike Schmidt',

  'Eddie Murray',

  'Tris Speaker',

  'Al Kaline',

  'Sandy Koufax',

  'Willie Keeler',

  'Pete Rose',

  'Robin Roberts',

  'Eddie Collins',

  'Lefty Gomez',

  'Lefty Grove',

  'Carl Yastrzemski',

  'Frank Robinson',

  'Juan Marichal',

  'Warren Spahn',

  'Pie Traynor',

  'Roberto Clemente',

  'Harmon Killebrew',

  'Satchel Paige',

  'Eddie Plank',

  'Josh Gibson',

  'Oscar Charleston',

  'Mickey Mantle',

  'Cool Papa Bell',

  'Johnny Bench',

  'Mickey Cochrane',

  'Jimmie Foxx',

  'Jim Palmer',

  'Cy Young',

  'Eddie Mathews',

  'Honus Wagner',

  'Paul Waner',

  'Grover Alexander',

  'Rod Carew',

  'Joe DiMaggio',

  'Joe Morgan',

  'Stan Musial',

  'Bill Terry',

  'Rogers Hornsby',

  'Lou Brock',

  'Ted Williams',

  'Bill Dickey',

  'Christy Mathewson',

  'Willie McCovey',

  'Lou Gehrig',

  'George Brett',

  'Hank Aaron',

  'Harry Heilmann',

  'Walter Johnson',

  'Roger Clemens',

  'Ty Cobb',

  'Whitey Ford',

  'Willie Mays',

  'Rickey Henderson',

  'Babe Ruth'

];

Now add three API endpoints

app.post('/api/addcard', async (req, res, next) =>

{

  // incoming: userId, color

  // outgoing: error

  var error = '';

  const { userId, card } = req.body;

  // TEMP FOR LOCAL TESTING.

  cardList.push( card );

  var ret = { error: error };

  res.status(200).json(ret);

});

app.post('/api/login', async (req, res, next) =>

{

  // incoming: login, password

  // outgoing: id, firstName, lastName, error

  var error = '';

  const { login, password } = req.body;

  var id = -1;

  var fn = '';

  var ln = '';

  if( login.toLowerCase() == 'bob' && password == 'COP4331' )

  {

    id = 1;

    fn = 'Bob';

    ln = 'Roberts';

  }

  else

  {

    error = 'Invalid user name/password';

  }

  var ret = { id:id, firstName:fn, lastName:ln, error:error};

  res.status(200).json(ret);

});

app.post('/api/searchcards', async (req, res, next) =>

{

  // incoming: userId, search

  // outgoing: results[], error

  var error = '';

  const { userId, search } = req.body;

  var \_search = search.toLowerCase().trim();

  var \_ret = [];

  for( var i=0; i<cardList.length; i++ )

  {

    var lowerFromList = cardList[i].toLocaleLowerCase();

    if( lowerFromList.indexOf( \_search ) >= 0 )

    {

      \_ret.push( cardList[i] );

    }

  }

  var ret = {results:\_ret, error:''};

  res.status(200).json(ret);

});

***Test with ARC or Postman***

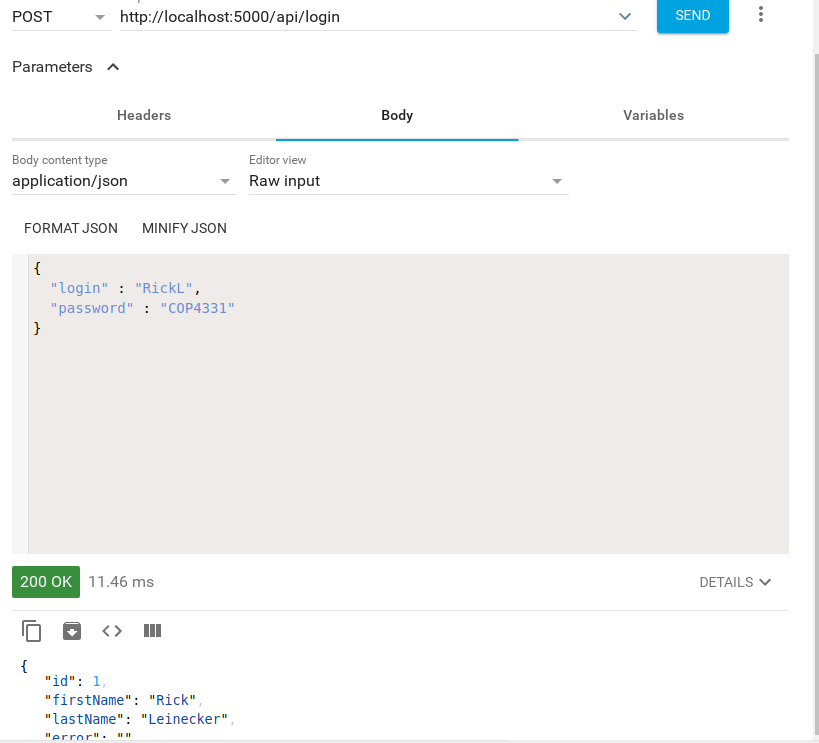
Enter the URL

Set to application/json content type

Set to POST

Click on body and then raw

Add JSON to the body



{

"login" : "RickL",

"password" : "COP4331"

}

{

"userId" : 1,

"search" : "ba"

}

{

"userId" : 1,

"card" : "Mickey Mouse"

}

Remember to make doLogin async

Edit doLogin:

async function doLogin(event:any) : Promise<void>

{

event.preventDefault();

var obj = {login:loginName,password:loginPassword};

var js = JSON.stringify(obj);

try

{

const response = await fetch('http://localhost:5000/api/login',

{method:'POST',body:js,headers:{'Content-Type': 'application/json'}});

var res = JSON.parse(await response.text());

if( res.id <= 0 )

{

setMessage('User/Password combination incorrect');

}

else

{

var user = {firstName:res.firstName,lastName:res.lastName,id:res.id}

localStorage.setItem('user\_data', JSON.stringify(user));

setMessage('');

window.location.href = '/cards';

}

}

catch(error:any)

{

alert(error.toString());

return;

}

};

***Test ...***

**Now for CardUI.tsx**

let \_ud : any = localStorage.getItem('user\_data');

let ud = JSON.parse( \_ud );

let userId : string = ud.id;

let firstName : string = ud.firstName;

let lastName : string = ud.lastName;

const [message,setMessage] = useState('');

const [searchResults,setResults] = useState('');

const [cardList,setCardList] = useState('');

const [search,setSearchValue] = React.useState('');

const [card,setCardNameValue] = React.useState('');

async function addCard(e:any) : Promise<void>

{

e.preventDefault();

let obj = {userId:userId,card:card};

let js = JSON.stringify(obj);

try

{

const response = await fetch('http://localhost:5000/api/addcard',

{method:'POST',body:js,headers:{'Content-Type': 'application/json'}});

let txt = await response.text();

let res = JSON.parse(txt);

if( res.error.length > 0 )

{

setMessage( "API Error:" + res.error );

}

else

{

setMessage('Card has been added');

}

}

catch(error:any)

{

setMessage(error.toString());

}

};

async function searchCard(e:any) : Promise<void>

{

e.preventDefault();

let obj = {userId:userId,search:search};

let js = JSON.stringify(obj);

try

{

const response = await fetch('http://localhost:5000/api/searchcards',

{method:'POST',body:js,headers:{'Content-Type': 'application/json'}});

let txt = await response.text();

let res = JSON.parse(txt);

let \_results = res.results;

let resultText = '';

for( let i=0; i<\_results.length; i++ )

{

resultText += \_results[i];

if( i < \_results.length - 1 )

{

resultText += ', ';

}

}

setResults('Card(s) have been retrieved');

setCardList(resultText);

}

catch(error:any)

{

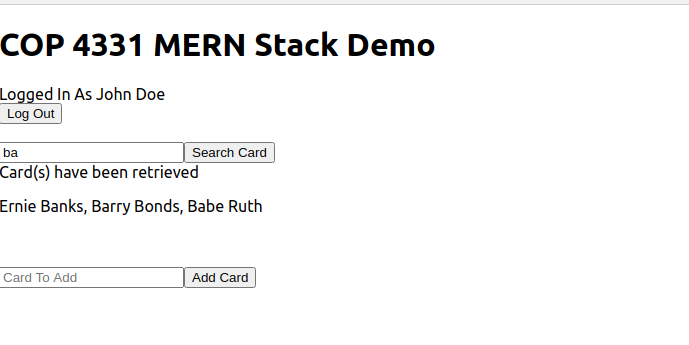
alert(error.toString());

setResults(error.toString());

}

};

***Test...***



**Create MongoDB DB**

Go to website and show

Make sure to go to network security and open for all IPs

**Add to server.js (PHASE 4)**

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

const url = 'mongodb+srv://RickLeinecker:COP4331Rocks@cluster0-4pisv.mongodb.net/COP4331?retryWrites=true&w=majority';

const client = new MongoClient(url);

client.connect();

***Change api/login***

app.post('/api/login', async (req, res, next) =>

{

  // incoming: login, password

  // outgoing: id, firstName, lastName, error

var error = '';

  const { login, password } = req.body;

  const db = client.db();

  const results = await db.collection('Users').find({Login:login,Password:password}).toArray();

  var id = -1;

  var fn = '';

  var ln = '';

  if( results.length > 0 )

  {

    id = results[0].UserId;

    fn = results[0].FirstName;

    ln = results[0].LastName;

  }

  var ret = { id:id, firstName:fn, lastName:ln, error:''};

  res.status(200).json(ret);

});

***Edit LoggedInName***

function LoggedInName()

{

    var \_ud = localStorage.getItem('user\_data');

if(\_ud == null) \_ud = "";

    var ud = JSON.parse(\_ud);

    var userId = ud.id;

    var firstName = ud.firstName;

    var lastName = ud.lastName;

    function doLogout(event:any) : void

    {

   event.preventDefault();

        localStorage.removeItem("user\_data")

        window.location.href = '/';

    };

  return(

   <div id="loggedInDiv">

<span id="userName">Logged In As {firstName} {lastName}</span><br />

<button type="button" id="logoutButton" className="buttons"

onClick={doLogout}> Log Out </button>

   </div>

  );

};

**Edit api/addcard**

app.post('/api/addcard', async (req, res, next) =>

{

  // incoming: userId, color

  // outgoing: error

  const { userId, card } = req.body;

  const newCard = {Card:card,UserId:userId};

  var error = '';

  try

  {

    const db = client.db();

    const result = db.collection('Cards').insertOne(newCard);

  }

  catch(e)

  {

    error = e.toString();

  }

  cardList.push( card );

  var ret = { error: error };

  res.status(200).json(ret);

});

**Edit api/searchcard**

app.post('/api/searchcards', async (req, res, next) =>

{

  // incoming: userId, search

  // outgoing: results[], error

  var error = '';

  const { userId, search } = req.body;

  var \_search = search.trim();

  const db = client.db();

  const results = await db.collection('Cards').find({"Card":{$regex:\_search+'.\*'}}).toArray();

  var \_ret = [];

  for( var i=0; i<results.length; i++ )

  {

    \_ret.push( results[i].Card );

  }

  var ret = {results:\_ret, error:error};

  res.status(200).json(ret);

});