

Submitted By: Madan Lal (053-18-0005)

Front End

Index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

App.js

```
import axios from 'axios'
import Navbar from './Components/Navbar'
import {BrowserRouter as Router,Routes,Route} from 'react-router-dom'
import Home from './Components/Home';
function App() {

  return<>
    <Router>
      <Navbar/>
      <Routes>
        <Route path="/" element={<Home />} />
      </Routes>
    </Router>

  </>
}

export default App;
```

Navbar.js

```
import {NavLink,Link} from 'react-router-dom';
function Navbar()
{
  return<>
    <nav class="navbar navbar-expand-sm navbar-dark bg-primary">
      <div class="container-fluid ">
        <div class="collapse navbar-collapse d-flex" id="navbarNav">
          <ul class="navbar-nav">
```

```

        <li class="nav-item">
          <NavLink to="/"
style={{textDecoration:"none"}} className="nav-link">Bookmark App</NavLink>
        </li>

      </ul>
    </div>
  </div>
</nav>
</>
}
export default Navbar

```

Home.js

```

function Home()
{
  // const data = () =>{
  //   axios.post(`http://localhost:8000/`,{headers:{'Content-Type':
'application/json',
  //   'Authorization':
localStorage.getItem("User")}}).then(res=>console.log(res.data))
  //   navigate('/')
  // }

  return (
    <>

      <br/>
      <button type="button" class="btn btn-primary btn-lg" data-bs-
toggle="modal" data-bs-target="#exampleModal">
      +
      </button>

      <div class="modal fade" id="exampleModal" tabindex="-1" aria-
labelledby="exampleModalLabel" aria-hidden="true">
        <div class="modal-dialog">
          <div class="modal-content">
            <div class="modal-header">
              <h5 class="modal-title"
id="exampleModalLabel">Favourites</h5>
              <button type="button" class="btn-close" data-bs-
dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
              <input id="myText" name="myText" type="text"
placeholder="Paste your link here..."/>
            </div>
            <div class="modal-footer">

```

```

        <button type="button" onClick={() =>
alert(document.getElementById("myText").value)} class="btn btn-
primary">Save</button>
        </div>
    </div>
</div>
</div>

</>
)
}
export default Home

```

Back End

Index.js

```

const express=require('express')
const app=express();
const mongoose=require('mongoose')
const cors = require('cors')

// middlewares
const Controller=require("./Controller/Controller")
const middleware=require('./middleware/UserMid')
app.use(cors())
app.use(express.json())

// connecting mongodb
mongoose.connect('mongodb://127.0.0.1:27017/ead7', () => {
    console.log('database connected successfully.');
```

```

}))

```

```

// api's
// app.post('/signup',Controller.insert);
// app.get('/products',middleware.isAuthenticate,Controller.getAll);
app.get('/delete/:id',middleware.isAuthenticate,Controller.Delete);
app.get('/find/:id',middleware.isAuthenticate,Controller.find);
app.post('/update/:id',middleware.isAuthenticate,Controller.Update);
// app.post('/signin',Controller.signin)

```

```

// server is listening on port 8000
app.listen(8000,function(){
    console.log("Server is listening at 8000")
})

```

Controller.js

```

const User=require("../model/User")

```

```

const bcrypt=require('bcrypt')
const { validate } = require("../model/User")
const jwt=require('jsonwebtoken')
const home=async (req,res)=>{

    const user=new
User({Username:"madanlal",Email:"madanlal@gmail.com",Password:"0000",Image:"Image.png
"})
    try
    {
        await user.save()
    }
    catch(er)
    {
        console.log(err)
    }
}
const getAll = async( req,res)=>
{
    const users = await User.find();
    res.status(200).send(
        users)
}
const Delete=async (req,res)=>{
    const {id}=req.params;
    console.log(id)
    await User.findByIdAndDelete(id)
    const users=await User.find()
    res.status(200).send(
        users
    )
}
const find= async (req,res)=>{
    const users=await User.findById(req.params.id);
    res.status(200).send(
        users
    )
}
const Update=async (req,res)=>{
    const {Username,Email,Password}=req.body
    const {id}=req.params;
    console.log(Username+" "+id,"Hello")
    await
User.updateOne({_id:id},{Username:Username,Email:Email,Password:Password}).then(function(){
    res.status(200).json({msg:"Updated"})
}).catch(function(err)
{
    res.status(401).json({msg:"Not Updated"})
}
}

```

```

    });
}
const insert=async (req,res)=>
{
    const {Username,Email,Password}=req.body;
    console.log("insertion")
    await User.create({Username,Email,Password},(err,User)=>
    {
        if(err)
        {
            res.status(401).send({msg:"Not Found"})
        }
        else
        res.status(200).json({msg:"Success"})
    })
}
const signin=async(req,res)=>{
    const {Username,Password}=req.body;
    User.findOne({Username:Username},(err,user)=>{
        if(user)
        {
            bcrypt.compare(Password,user.Password,(err,validate)=>{
                if(validate)
                {
                    const
token=jwt.sign({Username:user.Username},"MYSECRETKEY",{expiresIn:30*60});
                    res.json(token)
                }
                else
                {
                    res.status(401).json({
                        message: "no valid "
                    })
                }
            })
        }
    })
}
module.exports={home,getAll,insert,Delete,find,Update,signin}

```

Middleware.js

```

const jwt=require('jsonwebtoken')
const isAuthenticated=(req,res,next)=>{
    if(!req.headers.authorization)
    return res.status(401).json({

```

```

        message: "no unauthorized "
    })

    const result=jwt.verify(req.headers.authorization,"MYSECRETKEY")
    if(result)
    {
        console.log("forwarded ",result)
        next()
    }
    else
    {

        res.status(401).json({
            message: "no unauthorized "
        })
    }
}

module.exports={isAuthenticate}

```

Screen Shots

