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

[**https://shorturl.at/sMS17**](https://shorturl.at/sMS17)

**Pure Components**

**Capstone Project**

**Frontend**

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Pure components

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<>

src

PureComponents

- Parent.js

- Child.js

- Subchild.js

\*\*\*Parent.js\*\*\*

import React from "react";

import Child from "./Child";

//export default class Parent extends React.Component {

export default class Parent extends React.PureComponent {

constructor() {

super()

this.state = {

num: 100

}

}

componentDidMount() {

setInterval(() => {

//this.setState({ num: Math.random() \* 100 })

this.setState({ num: 100 })

}, 2000)

}

render() {

console.log('Parent Render')

return (

<div>

<p> num : {this.state.num} </p>

<button onClick={() => {

this.setState({ num: 102 })

}}>Change</button>

<Child key1={this.state.num}></Child>

</div>

)

}

}

\*\*\*Child.js\*\*\*

import React from "react";

import Subchild from "./Subchild";

//export default class Child extends React.Component {

export default class Child extends React.PureComponent {

render() {

console.log('Child Render')

return (

<div>

Child :- {this.props.key1}

<Subchild key2={this.props.key1} />

</div>

)

}

}

\*\*\*Subchild.js\*\*\*

import React from "react";

//export default class Subchild extends React.Component {

export default class Subchild extends React.PureComponent {

render() {

console.log('Subchild render')

return (

<div>

<p>Subchild :- {this.props.key2} </p>

</div>

)

}

}

Note : - Pure component prevents re-rendering of components when state remains same.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Unit Testing:-

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Testing:-

- Testing is categorised into

i) Manual Testing

ii)Automation Testing

- Now a days manual testing is deprecating

- Automation testing is categorised into

i) Unit testing

ii)Integration testing

iii)End to end testing

Unit testing

* Testing particular functionalities with dummy scenarios is called as unit testing.
* 'jest' is the javascript framework used to write unit tests.
* unit test cases files must have extension as '.test.js'
* it(-,-) / test(-,-) are predefined functions in jest framework.
* these functions are used to write test suits.
* describe(-,-) function is used to access results.
* unit test cases are executed using following command

>yarn test

\*\*\*nomanip.js\*\*\*

let num = 0

const increment = () => {

return num += 1

}

const decrement = () => {

return num -= 1

}

export default increment

export { decrement }

\*\*\*nomanip.test.js\*\*\*

import increment, { decrement } from "./nomanip"

describe("Increment function testing", () => {

test("Increment function increments result by 1", () => {

expect(increment()).toBe(1)

})

})

describe("Decrement function testing", () => {

test("Decrement function decrements result by 1", () => {

expect(decrement()).toBe(-1)

})

})

\*\*\*App.test.js\*\*\*

import { render, screen } from '@testing-library/react';

import App from './App';

test('renders learn react link', () => {

render(<App />);

const linkElement = screen.getByText(/learn react/i);

expect(linkElement).toBeInTheDocument();

});

test("Check 'Edit' message on the browser",()=>{

render(<App/>)

const linkElement = screen.getByText(/edit/i)

expect(linkElement).toBeInTheDocument()

})

Capstone Project

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

Backend

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

**Capstone Project POC**

- Design e-commerse website

- options are (two or more)

- Medicins

- Clothing

- Workout

- Mobiles

- Mobile accessories

- Computer equipments

- Computer accessories

- Books

- Groceries

- Car accessories

- Jewelleries

- watches

- Sun glasses

Backend

Functionality

- on sign in create a new user in users

- on login compare username and password with database and proceed accordingly

- on add to cart insert a record in cart collection

- on buy now update total quantity in products collection

API calls

- Create user -> insertuser

- Login -> login

- Show all products -> fetch

- Add to cart -> insertproduct / updateproduct

- Reduce from cart -> updateproduct / deleteproduct

- Buy now -> updateproducts

collections

- products

[

{

product\_1 details

},

{

product\_2 details

},

...

]

- users (uname, upwd)

[

{

user\_1 details

},

{

user\_2 details

},

...

]

- cart

[

{

id:

user:

product\_details:

},

...

]

Front end

- Website should have rich user interface

- There should be card layout for each product

- on mouse hover on product image that product should be enlarged

- There should be 'learn more', 'add to cart' and 'buy now' options

- There should be login page for sigining in

- There should be cart page where we will get number of added items

- In buy now page show complete cost of purchase

Enhancements (Optional)

- Try user interface without using Bootstrap CDN

- Use diffrent GST rates for various products

**Backend Functionality**

Create mongodb database ‘miniprj’

Create three collections

products

p\_id

p\_name

p\_cat

p\_img

p\_cost

p\_desc

users

userid

uname

upwd

email

address

contact

cart

*Use mongodbcrud code and update the same, dont change url.js and server.js*

\*\*\*fetch.js\*\*\*

//import modules

const express = require('express')

let mongodb = require('mongodb')

//import url

let url = require('../url')

//create mongoclient

let mcl = mongodb.MongoClient

//create router instance

let router = express.Router()

//create rest api

router.get("/", (req, res) => {

//connect to mongodb

mcl.connect(url, (err, conn) => {

if (err)

console.log('Error in connection:- ', err)

else {

let db = conn.db('miniprj')

db.collection('products').find().toArray((err, array) => {

if (err)

console.log('Error :- ' + err)

else {

console.log('Data sent')

res.json(array)

conn.close()

}

})

}

})

})

//User login Authentication

router.post("/auth", (req, res) => {

let uname = req.body.uname

let upwd = req.body.upwd

let obj = { uname, upwd }

//connect to mongodb

mcl.connect(url, (err, conn) => {

if (err)

console.log('Error in connection:- ', err)

else {

let db = conn.db("miniprj")

db.collection('users').find(obj).toArray((err, array) => {

if (err)

console.log(err)

else {

if (array.length > 0)

res.json({ 'auth': 'success', 'user': uname })

else

res.json({ 'auth': 'failed' })

console.log('Auth response sent')

conn.close()

}

})

}

})

})

//Fetch cart data

router.post("/fetchCart", (req, res) => {

let uname = req.body.uname

let obj = { uname }

//connect to mongodb

mcl.connect(url, (err, conn) => {

if (err)

console.log('Error in connection:- ', err)

else {

let db = conn.db("miniprj")

db.collection('cart').find(obj).toArray((err, array) => {

if (err)

console.log(err)

else {

res.json(array)

console.log(`Cart response for ${obj.uname} sent`)

conn.close()

}

})

}

})

})

//export router

module.exports = router

\*\*\*insert.js\*\*\*

//import modules

const express = require('express')

let mongodb = require('mongodb')

//import url

let url = require('../url')

//create mongoclient

let mcl = mongodb.MongoClient

//create router instance

let router = express.Router()

//create rest api

router.post("/", (req, res) => {

let obj = req.body

//connect to mongodb

mcl.connect(url, (err, conn) => {

if (err)

console.log('Error in connection :- ', err)

else {

let db = conn.db("miniprj")

db.collection('products').insertOne(obj, (err) => {

if (err)

res.json({ 'insert': 'Error ' + err })

else {

console.log("Data inserted")

res.json({ 'insert': 'success' })

conn.close()

}

})

}

})

})

//Insert User

router.post("/createUser", (req, res) => {

let obj = {

"userid": req.body.userid,

"uname": req.body.uname,

"upwd": req.body.upwd,

"email": req.body.email,

"address": req.body.address,

"contact": req.body.contact

}

//connect to mongodb

mcl.connect(url, (err, conn) => {

if (err)

console.log('Error in connection :- ', err)

else {

let db = conn.db("miniprj")

db.collection('users').insertOne(obj, (err) => {

if (err)

res.json({ 'userInsert': 'Error ' + err })

else {

console.log("User inserted")

res.json({ 'userInsert': 'success' })

conn.close()

}

})

}

})

})

//insert product into cart

router.post("/cartInsert",(req,res)=>{

let obj = {

"p\_id" : req.body.p\_id,

"p\_cost" : req.body.p\_cost,

qty : 1,

"p\_img":req.body.p\_img,

"uname" : req.body.uname

}

//connect to mongodb

mcl.connect(url, (err, conn) => {

if (err)

console.log('Error in connection :- ', err)

else {

let db = conn.db("miniprj")

db.collection('cart').insertOne(obj, (err) => {

if (err)

res.json({ 'cartInsert': 'Error ' + err })

else {

console.log("Prouct in Cart inserted")

res.json({ 'cartInsert': 'success' })

conn.close()

}

})

}

})

})

//export router

module.exports = router

\*\*\*update.js\*\*\*

//import modules

const express = require('express')

let mongodb = require('mongodb')

//import url

let url = require('../url')

//create mongoclient

let mcl = mongodb.MongoClient

//create router instance

let router = express.Router()

//create rest api

router.post('/', (req, res) => {

let p\_id = req.body.p\_id

let obj = {

"p\_name": req.body.p\_name,

"p\_cost": req.body.p\_cost

}

//connect to mongodb

mcl.connect(url, (err, conn) => {

if (err)

console.log('Error in connection:- ', err)

else {

let db = conn.db("miniprj")

db.collection("products").updateOne({ p\_id }, { $set: obj }, (err, result) => {

if (err)

res.json({ 'update': 'Error ' + err })

else {

if (result.matchedCount != 0) {

console.log("Data updated ")

res.json({ 'update': 'success' })

}

else {

console.log("Data Not updated ")

res.json({ 'update': 'Record Not found' })

}

conn.close()

}

})

}

})

})

//Update product in cart

router.post("/updateCart", (req, res) => {

let p\_id = req.body.p\_id

let uname = req.body.uname

let obj = { "qty": req.body.qty }

//connect to mongodb

mcl.connect(url, (err, conn) => {

if (err)

console.log('Error in connection:- ', err)

else {

let db = conn.db('miniprj')

db.collection('cart').updateOne({ p\_id, uname }, { $set: obj },

(err, result) => {

if (err)

res.json({ 'cartUpdate': 'Error ' + err })

else {

if (result.matchedCount != 0) {

console.log(`Cart data for ${uname} updated`)

res.json({ 'cartUpdate': 'success' })

}

else {

console.log(`Record not updated`)

res.json({ 'cartUpdate': 'Record Not found' })

}

conn.close()

}

})

}

})

})

//Update user

//to be done by participants

//export router

module.exports = router

\*\*\*delete.js\*\*\*

//import modules

const express = require('express')

let mongodb = require('mongodb')

//import url

let url = require('../url')

//create mongoclient

let mcl = mongodb.MongoClient

//create router instance

let router = express.Router()

//create rest api

router.post("/", (req, res) => {

let obj = {

"p\_id": req.body.p\_id

}

//connect to mongodb

mcl.connect(url, (err, conn) => {

if (err)

console.log('Error in connection:- ', err)

else {

let db = conn.db('nodedb')

db.collection('products').deleteOne(obj, (err, result) => {

if (err)

res.json({ 'delete': 'Error ' + err })

else {

if (result.deletedCount != 0) {

console.log('Data deleted')

res.json({ 'delete': 'success' })

}

else {

console.log('Data Not deleted')

res.json({ 'delete': 'Record Not found' })

}

conn.close()

}

})

}

})

})

//Delete product from cart

router.post("/deleteCart", (req, res) => {

let obj = {

"p\_id": req.body.p\_id,

"uname": req.body.uname

}

//connect to mongodb

mcl.connect(url, (err, conn) => {

if (err)

console.log('Error in connection:- ', err)

else {

let db = conn.db('miniprj')

db.collection('cart').deleteOne(obj, (err, result) => {

if (err)

res.json({ 'cartDelete': 'Error ' + err })

else {

if (result.deletedCount != 0) {

console.log(`Cart data fro ${obj.uname} deleted`)

res.json({ 'cartDelete': 'success' })

}

else {

console.log('Cart Data Not deleted')

res.json({ 'cartDelete': 'Record Not found' })

}

conn.close()

}

})

}

})

})

//Delete user

//to be done by participants

//export router

module.exports = router

Hosting the application

>npm init

1. create '.gitignore' file

>npx gitignore node

2. login to github.com and create repository

3. copy url

- url

4. initialyse local repository

>git init

5. add files to repository

>git add .

6. check status

>git status

7. commit

>git commit -m "initial Commit"

8. add to remote repository

>git remote add origin --Copied URL --

9. push code to repository

>git push -u origin master

Deploying nodejs on cyclic.sh

\*Login cyclic.sh with github

1. Click on deploy now

2. Select Link your own

3. Search and select required repository

4. Click on connect

5. Wait to deploy it

6. after getting message 'You're Live!'

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

**Frontend Functionality**

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

Create a new react application

>create-react-app client

Create various components

- aboutus.js

- contactus.js

- Header.js

- MainComponent.js -> Login and Dashboard

- SignupComponent.js -> Create new user

- indexComponent.js -> Home Page

- url.js -> Backend url (APIs)

\*\*\*url.js\*\*\*

module.exports = "http://localhost:8080"

\*\*\*Abouts.js\*\*\*

import React from "react"

export default class Aboutus extends React.Component {

render() {

return (

<div className='container mt-5'>

<p className='jumbotron'>Welcome to Aboutus</p>

</div>

)

}

}

Similarly design contactus.js, MainComponent and Signup Components

Download following libraries

‘react-router-dom’, ‘axios’

>yarn add react-router-dom axios --save

\*\*\*indexComponent.js\*\*\*

import React from "react";

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

import Aboutus from "./aboutus";

import Contactus from "./contactus";

import SignupComponent from "./SignupComponent";

import MainComponent from "./MainComponent";

export default class IndexComponent extends React.Component{

render(){

return(

<div>

<div className="nav nav-pills">

<Router>

<div className="nav-item">

<NavLink to = "/aboutus" className='nav-link'>About us</NavLink>

</div>

<div className="nav-item">

<NavLink to = "/contactus" className='nav-link'>Contact us</NavLink>

</div>

<div className="nav-item">

<NavLink to = "/signup" className='nav-link'>Signup</NavLink>

</div>

<div className="nav-item">

<NavLink to = "/login" className='nav-link'>Login</NavLink>

</div>

<br/><br/>

<Routes>

<Route path="/aboutus" element={<Aboutus/>}></Route>

<Route path="/contactus" element= {<Contactus/>}></Route>

<Route path="/signup" element={<SignupComponent/>}></Route>

<Route path="/login" element={<MainComponent/>}></Route>

</Routes>

</Router>

</div>

</div>

)

}

}

\*\*\*SignupComponent.js\*\*\*

import React from 'react'

import axios from 'axios'

import url from './url'

export default class SignupComponent extends React.Component {

constructor() {

super()

this.state = {

status:''

}

}

render() {

return (

<div className='container mt-5'>

<form onSubmit={this.signup} className='btn btn-outline-warning w-50'>

<h3 className='text-primary'>Signup user </h3>

<div className='form-group my-2 btn btn-outline-dark p-3 w-100'>

<label>User id</label>

<input type='text' placeholder='Enter Userid' className='form-control' name='userid'></input>

</div>

<div className='form-group my-2 btn btn-outline-dark p-3 w-100'>

<label>User Name</label>

<input type='text' placeholder='Enter User Name' className='form-control' name='uname'></input>

</div>

<div className='form-group my-2 btn btn-outline-dark p-3 w-100'>

<label>Password</label>

<input type='password' placeholder='Enter Password' className='form-control' name='upwd'></input>

</div>

<div className='form-group my-2 btn btn-outline-dark p-3 w-100'>

<label>User email</label>

<input type='email' placeholder='Enter User email' className='form-control' name='email'></input>

</div>

<div className='form-group my-2 btn btn-outline-dark p-3 w-100'>

<label>User Address</label>

<input type='text' placeholder='Enter User Address' className='form-control' name='address'></input>

</div>

<div className='form-group my-2 btn btn-outline-dark p-3 w-100'>

<label>Contact</label>

<input type='text' placeholder='Enter Contact' className='form-control' name='contact'></input>

</div>

<div className='form-group my-2 w-25 mx-auto' align = 'center'>

<input type='submit' className='btn btn-outline-success' value='Signup'></input>

<h3>{this.state.status}</h3>

</div>

</form>

</div>

)

}

signup = (e) => {

e.preventDefault()

let obj = {

"userid" : e.target.userid.value,

"uname" : e.target.uname.value,

"upwd" : e.target.upwd.value,

"email" : e.target.email.value,

"address" : e.target.address.value,

"contact" : e.target.contact.value

}

axios.post(url+"/insert/createUser",obj)

.then((posRes)=>{

console.log(posRes.data)

this.setState({

status : posRes.data.userInsert

})

},(errRes)=>{

console.log(errRes)

})

}

}

\*\*\*MainComponent.js\*\*\*

import axios from 'axios'

import React from 'react'

import url from './url'

import Header from './Header'

let cart = []

export default class MainComponent extends React.Component {

constructor() {

super()

this.state = {

login: true,

products: []

}

}

componentDidMount() {

axios.get(url + "/fetch")

.then((posRes) => {

this.setState({

products: posRes.data

})

}, (errRes) => {

console.log(errRes)

})

}

render() {

return (

<div className='container mt-5'>

<div hidden={this.state.login}>

<form onSubmit={this.login} className='btn btn-outline-dark text-white w-50'>

<div className='form-group my-2 btn btn-outline-info p-3 w-75'>

<label className='float-left'>Username</label>

<input type='text' placeholder='Enter User name' className='form-control' name='uname'></input>

</div>

<div className='form-group my-2 btn btn-outline-info p-3 w-75'>

<label className='float-left'>Password</label>

<input type='password' placeholder='Enter Password' className='form-control' name='upwd'></input>

</div>

<div className='form-group my-2 w-25 mx-auto' align='center'>

<input type='submit' className='btn btn-outline-success' value='Login'></input>

</div>

</form>

</div>

<div hidden={!this.state.login}>

<button onClick={this.logout} className='btn btn-outline-danger float-right mt-4 mr-5'>Logout</button>

<Header />

<div className='h4 text-info mb-2' align="right">

Total amount:- {this.calculateTotal()}

<button onClick={() => { this.buyNow() }} className='btn btn-outline-success mx-5'>Buy Now</button>

</div>

<div className='row'>

<div className='col-10'>

<div className='row row-cols-3'>

{this.state.products.map((e, i) => (

<div className='col my-3'>

<div className='card'>

<div className='card-header'>

<img src={e.p\_img} className='card-img-top'></img>

</div>

</div>

<div className='card-body'>

<div className='h2 card-title'>{e.p\_name}</div>

<div className='h4 card-subtitle text-muted'>{e.p\_cost}</div>

</div>

<div className='card-footer'>

<button onClick={() => { alert(e.p\_desc) }}

className="btn btn-outline-info btn-block btn-sm"

data-toggle="tooltip"

data-placement="bottom"

title={e.p\_desc}>Learn More</button>

<button onClick={() => { this.addToCart(e) }} class="btn btn-outline-success btn-block btn-sm">Add to Cart</button>

</div>

</div>

))}

</div>

</div>

<div className='col'>

<div className='row my-3'>

{cart.map((e, i) => (

<div className='my-3'>

<div className=' card'>

<div className='card-header'>

<img src={e.p\_img} className='card-img-top'></img>

</div>

<div className='card-body'>

<div className='h2 card-title'>{e.p\_name}</div>

<div className='h4 card-subtitle text-muted'>{e.qty}</div>

</div>

<div className='card-footer'>

<button onClick={() => { this.reduce(e) }} class="btn btn-outline-success btn-block btn-sm">Reduce</button>

</div>

</div>

</div>

))}

</div>

</div>

</div>

</div>

</div>

)

}

login = (e) => {

e.preventDefault()

let obj = {

uname: e.target.uname.value,

upwd: e.target.upwd.value

}

axios.post(url + "/fetch/auth", obj)

.then((posRes) => {

console.log(posRes.data)

if (posRes.data.auth == 'success') {

console.log('Object in auth :- ', obj)

this.setState({

...this.state,

login: true,

user: obj.uname

})

console.log('State after login:- ', this.state)

window.sessionStorage.setItem('user', obj.uname)

this.fetchCart()

}

}, (errRes) => {

console.log(errRes)

})

}

addToCart = (item) => {

let present = false

let i

for (i = 0; i < cart.length; i++) {

if (item.p\_id == cart[i].p\_id) {

present = true

break

}

}

//if item is present in cart, update it

if (present == true) {

let myObj = cart[i]

let id = myObj.id

let obj = {

"uname": this.state.user,

"p\_id": myObj.p\_id,

"qty": parseInt(myObj.qty) + 1,

}

axios.post(url + "/update/updateCart", obj)

.then((posRes) => {

cart.forEach((e, i) => {

if (e.p\_id == obj.p\_id)

e.qty = obj.qty

})

console.log(posRes.statusText)

this.setState({

status: 'Update ' + posRes.statusText

})

}, (errRes) => {

console.log(errRes)

this.setState({

status: errRes.message

})

})

}

//if item is not present in cart, insert it

else {

let obj = {

"uname": this.state.user,

"p\_name": item.p\_name,

"p\_id": item.p\_id,

"qty": 1,

"p\_cost": item.p\_cost,

"p\_img": item.p\_img

}

axios.post(url + "/insert/cartInsert", obj)

.then((posRes) => {

this.setState({

status: 'Record' + posRes.statusText

})

cart.push(obj)

}, (errRes) => {

console.log(errRes)

})

}

this.setState({

total: this.calculateTotal()

})

console.log(cart)

}

reduce = (item) => {

console.log('Item id:- ', item.id)

if (item.qty == 1) {

let obj = {

"u\_name": this.state.user,

"p\_id": item.p\_id

}

axios.post(url + "/delete/deleteCart", obj)

.then((posRes) => {

console.log(posRes)

let indx = cart.findIndex((e, i) => {

return e.p\_id === item.p\_id

})

cart.splice(indx, 1)

this.setState({

status: 'Delete ' + posRes.statusText

})

}, (errRes) => {

console.log(errRes)

this.setState({

status: errRes.message

})

})

}

else {

let obj = {

"u\_name": this.state.user,

"p\_id": item.p\_id,

"qty": parseInt(item.qty) - 1,

}

axios.post(url + "/update/updateCart", obj)

.then((posRes) => {

cart.forEach((e, i) => {

if (e.p\_id == obj.p\_id)

e.qty = obj.qty

})

console.log(posRes.statusText)

this.setState({

status: 'Update ' + posRes.statusText

})

}, (errRes) => {

console.log(errRes)

this.setState({

status: errRes.message

})

})

}

this.setState({

total: this.calculateTotal()

})

}

buyNow = () => { // Check this functionality ????

alert('Thank u for business with us Total amount :- ' + this.calculateTotal())

for (let i = 0; i < cart.length; i++) {

axios.delete(url + "/cart/" + cart[i].id) //==================>?

.then((posRes) => {

console.log(posRes)

let indx = cart.findIndex((e, i) => {

return e.id === cart[i].id

})

cart.splice(indx, 1)

this.setState({

status: 'Delete ' + posRes.statusText

})

}, (errRes) => {

console.log(errRes)

this.setState({

status: errRes.message

})

})

}

}

fetchCart = () => {

axios.post(url + "/fetch/fetchCart", { "uname": window.sessionStorage.getItem('user') }).then((posRes) => {

this.setState({

status: 'Loading'

})

cart = posRes.data

console.log('Cart data:- ', cart)

this.setState({

status: '',

})

}, (errRes) => {

console.log(errRes)

})

let total = 0

for (let i = 0; i < cart.length; i++) {

total += cart[i].p\_cost \* cart[i].qty

}

this.setState({

total: total

})

}

calculateTotal = () => {

let total = 0

cart.forEach((e, i) => {

total += e.qty \* e.p\_cost

})

return total

}

logout = () => {

this.setState({

login: false,

user: ''

})

window.sessionStorage.removeItem('user')

}

}

\*\*\*Header.js\*\*\*

import React from "react";

import './style.css'

let stl = ''

export default class Header extends React.Component {

constructor() {

super()

this.state = {

user: window.sessionStorage.getItem('user'),

wish: '',

}

}

componentDidMount() {

let date = ""

//date = new Date().getHours()

date = 17

if (date < 12) {

stl = 'mrng'

this.setState({

wish: 'Good Morning'

})

}

else if (date < 16) {

stl = 'aftr'

this.setState({

wish: 'Good Afternoon'

})

}

else if (date < 20) {

stl = 'eve'

this.setState({

wish: 'Good Evening'

})

}

}

render() {

this.state = {

...this.state,

user: window.sessionStorage.getItem('user')

}

return (

<div>

<h1 className={stl}>{this.state.wish} {this.state.user}</h1>

</div>

)

}

}

\*\*\*style.css\*\*\*

.img {

width: 100px;

height: 100px;

}

.card-img-top:hover {

position: relative;

transition: all 0.5s;

transform-origin: center;

transform: scale(1.05);

}

.card-img-top {

transition: all 0.5s;

}

.mrng {

color: white;

background: linear-gradient(to right, blue, yellow);

padding: 20px;

border-radius: 10px;

}

.aftr {

color: red;

background: linear-gradient(to right, yellow, orange);

padding: 20px;

border-radius: 10px;

}

.eve {

color: blue;

background: linear-gradient(to right, orange, gray);

padding: 20px;

border-radius: 10px;

}