

Name: MACARAEG, JAKE RUSSELL F

Date: 09/12/2024

Course and Year: BSIT - 3

Section: IAB2

APPDEV1

Introduction to Application Development 1

Laboratory Activities

First Term Activities

ActivityGuide1

APPDEV1 - INTRODUCTION TO APPLICATION DEVELOPMENT
ACTIVITY GUIDE #1 (FIRST TERM)

JEREMY MOSES T. EBREO

Activity Guide: Create the indicated JavaScript Programs for the problems below. **5 points each.**

Rubrics:

5 points - There are no errors in the program code and it solves the indicated specifications.

3 points - There are evident errors in the program code and it somewhat solves the indicated specifications.

0 points - There are no traces of program code for the specifications presented.

1. Personal Information

Design a program that displays the following information:

- Your Name
- Your address, with city, state and ZIP
- Your telephone number
- Your college major

2. Sales Prediction

A company has determined that its annual profit is typically 23 percent of total sales. Design a program that asks the user to enter the projected amount of total sales, and then displays the profit that will be made from that amount.

Hint: Use the value 0.23 to represent 23 percent.

3. Distance Traveled

Assuming there are no accidents or delays, the distance that a car travels down the interstate can be calculated with the following formula:

$$\text{Distance} = \text{Speed} \times \text{Time}$$

A car is traveling at 60 miles per hour. Design a program that displays the following:

- The distance the car will travel in 5 hours
- The distance the car will travel in 8 hours
- The distance the car will travel in 12 hours

4. Miles-per-Gallon

A car's miles-per-gallon (MPG) can be calculated with the following formula:

$$\text{MPG} = \text{Miles driven} / \text{Gallons of gas used}$$

Design a program that asks the user for the number of miles driven and the gallons of gas used. It should calculate the car's miles per gallon and display the result on the screen.

5. Celsius to Fahrenheit Temperature Converter

Design a program that converts Celsius temperatures to Fahrenheit temperatures. The formula is as follows:



```
index.html x JS app.js
JSPROJECTS > ActivityGuide1 > index.html > html > body
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Activity Guide 1</title>
7   </head>
8
9   <body>
10    <h1>Activity Guide #1</h1>
11    <!--START OF ITEM 1-->
12    <h4>1. Personal Information</h4>
13    Name: <input type="text" id="name" placeholder="Enter your Name"><br />
14    Address:<input type="text" id="address" placeholder="Enter your Address"><br />
15    Telephone Number:<input type="text" id="telephone" placeholder="Enter your Telephone No."><br />
16    College Major:<input type="text" id="major" placeholder="Enter your College Major"><br />
17    <button id="btn1">Submit</button>
18    <p id="output"></p>
19    <!--END OF ITEM 1-->
20
21  </body>
</html>
```

Activity Guide #1

1. Personal Information

Name:
Address:
Telephone Number:
College Major:

Hello Jake, I See that you're from Pangasinan, and you are also pursuing your degree of BSIT and you can be contacted using 0915,

2. Sales Prediction

Projected Total Sales:

The Profit is 5.29

3. Distance Traveled

Distance in 5 hours: 300 miles
Distance in 8 hours: 480 miles
Distance in 12 hours: 720 miles

4. Miles-per-Gallon

Miles Driven:
Gallons of Gas Used:

Miles-per-Gallon: 10.25

5. Celsius to Fahrenheit Temperature Converter

Celsius Temperature:

Fahrenheit Temperature: 73.40 °F

6. Cookie Calories

Number of Cookies Eaten:

Total Calories Consumed: 225.00

7. Male and Female

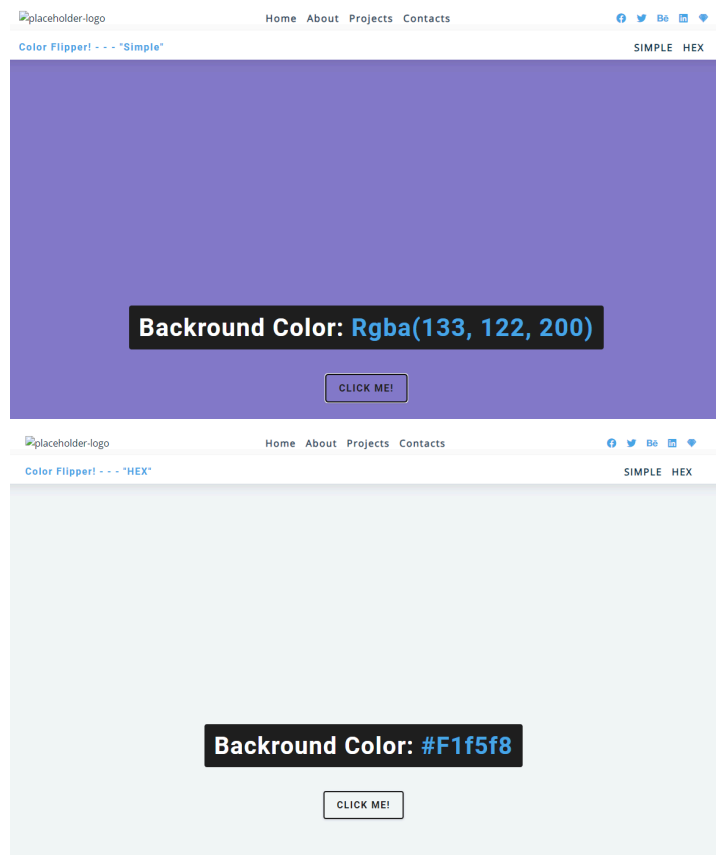
Number of Males:
Number of Females:

Percentage of Males: 40%
Percentage of Females: 60%



COLOR PICKER (Simple & Hex)

```
1  const hex = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, "A", "B", "C", "D", "E", "F"]
2  const btn = document.getElementById("btn")
3  const color = document.querySelector(".color")
4
5  btn.addEventListener("click", () => {
6    let hexColor = '#'
7    for(let i = 0; i < 6; i++){
8      hexColor += hex[getRandomNumber()]
9    }
10   color.textContent = hexColor
11   document.body.style.backgroundColor = hexColor
12 })
13
14 getRandomNumber = () => {
15   return Math.floor(Math.random() * hex.length)
16 }
17
18 const navToggle = document.querySelector('.nav-toggle')
19 const links = document.querySelector('.links')
20
21 navToggle.addEventListener('click', () => {
22   /*if(links.classList.contains('show-links')){
23     links.classList.remove('show-links');
24   }else{
25     links.classList.add('show-links');
26   }*/
27   links.classList.toggle('show-links')
28 })
29
```





Counter

```
JS app.js x
JSPROJECTS > Counter > JS app.js > ...
1 let count = 0;
2 const value = document.querySelector("#value")
3 const btns = document.querySelectorAll('.btn')
4 console.log(btns)
5 btns.forEach(btn => {
6   btn.addEventListener('click', (e) => {
7     const styles = e.currentTarget.classList
8
9     if (styles.contains('decrease')) {
10      count--
11    } else if (styles.contains('increase')) {
12      count++
13    } else if (styles.contains('decreaseByFive')) {
14      count -= 5
15    } else if (styles.contains('increaseByFive')) {
16      count += 5
17    } else if (styles.contains('random')) {
18      count = Math.floor(Math.random() * 301) - 150
19    } else {
20      count = 0
21    }
22
23    if (count > 0) {
24      value.style.color = "green"
25    } else if (count < 0) {
26      value.style.color = "red"
27    } else {
28      value.style.color = "black"
29    }
30
31    value.textContent = count
32  })
33 })
```

placeholder-logo

[Home](#) [About](#) [Projects](#) [Contacts](#)



Counter

0

DECREASE

RESET

INCREASE

DECREASE BY 5

RANDOM

INCREASE BY 5



Navbar

```
JSPROJECTS > NavBar > JS app.js > ...
1  const navToggle = document.querySelector('.nav-toggle')
2  const links = document.querySelector('.links')
3
4  navToggle.addEventListener('click', () => {
5    /*if(links.classList.contains('show-links')){
6      links.classList.remove('show-links');
7    }else{
8      links.classList.add('show-links');
9    }*/
10   links.classList.toggle('show-links')
11 })
12
```

Home About Projects Contacts



RandomQuoteGenerator

```
JSPROJECTS > RandomQuote Generator > JS RQG.js > ...
1  const quotes = ["Forefathers One And All! Bear Witness!",
2                  "Foul Tarnished, in search of the Elden Ring. Embol",
3                  "Brave tarnished, thy strength befits a crown",
4                  "I will seek you as far as you may travel. To deliv",
5                  "You will Witness true horror.",
6                  "The Fallen Leaves Tell A Story.",
7                  ]
8  const btn = document.getElementById("btn")
9  const color = document.querySelector(".color")
10
11  btn.addEventListener("click", () =>{
12      console.log(document.body)
13      const randomNumber = getRandomNumber()
14      document.body.style.backgroundColor = quotes[randomNumber]
15      color.textContent = quotes[randomNumber]
16  })
17  getRandomNumber = () =>{
18      return Math.floor(Math.random() * quotes.length)
19  }
20
```

placeholder-logo

[Home](#) [About](#) [Projects](#) [Contacts](#)



Random Quote:
Brave Tarnished, Thy Strength Befits A Crown

GENERATE!

Simple Automated Timer

Activity Guide (Simple Timer)



Jeremy Moses Ebreo • Aug 28

10 points

Due Aug 28, 11:59 PM

Web Application Specifications:

Create an automated timer using JavaScript and CSS. You should reuse the elements and files from our Counter Application. Edit our **Counter Application** so you have a **Simple Automated Timer** counting down from 30 seconds to 0. The web application should have a button to start the countdown and once it is started, it should not stop until it has reached 0 seconds.

After Creating the Simple Automated Timer, you will create a web app compilation of your recent activities linking them using the **anchor tags**. The web app compilation should use our recently made **Navigation Bar** project. The Applications to be compiled are the **Counter Project**, **Color Picker** and **Simple Automated Timer**. These will be checked next meeting.



```
Simple Automated Timer > Index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Simple Automated Timer</title>
7      <link rel="stylesheet" href="style.css">
8      <link rel="stylesheet" href="../NavBar/style.css">
9    </head>
10   <body>
11     <nav>
12       <div class="nav-center">
13         <div class="nav-header">
14           
15           <button class="nav-toggle">
16             <i class="fas fa-bars"></i>
17           </button>
18         </div>
19         <!-- Link of the nav bar -->
20         <ul class="links">
21           <li><a href="../NavBar/index.html">Home</a></li>
22           <li><a href="../NavBar/about.html">About</a></li>
23           <li><a href="../NavBar/Projects/projects.html">Projects</a></li>
24           <li><a href="../NavBar/contacts.html">Contacts</a></li>
25         </ul>
26       </div>
27     </nav>
28     <main>
29       <div class="container">
30         <h1>Automated Timer</h1>
31         <span id="value">30</span>
32         <div class="button-container">
33           <button class="btn start">Start Timer</button>
34           <button class="btn reset">Reset</button>
35         </div>
36       </div>
37     </main>
38     <script src="app.js"></script>
39     <script src="../NavBar/app.js"></script>
40   </body>
41 </html>
42
```

Automated Timer

30

START TIMER

RESET

Food Website

```
1 import Header from "../Header";
2 import Body from "../Body";
3 import Footer from "../Footer";
4
5 export default function App(){
6   return(
7
8     <>
9     <Header/>
10    <Body name= "Macaraeg" food ="Pizza" isHealthy ="No" age = "20"/>
11    <Body name= "Lansangan" food ="Carbonara" isHealthy ="No" age = "21"/>
12    <Body name= "Elec" food ="HotDog" isHealthy ="No" age = "23"/>
13    <Body name= "Pasion" food ="Banana" isHealthy ="Yes" age = "69"/>
14    <Body name= "Talusig" food ="Banana" isHealthy ="Yes" age = "43"/>
15    <Footer/>
16  </>
17
18 )
19 }
```

My Food Website

- [Home](#)
- [Contact](#)
- [About](#)

				
This is Macaraeg ' favorite food: PIZZA Is it healthy? Yes Macaraeg is 20 years old!	This is Lansangan ' favorite food: CARBONARA Is it healthy? Yes Lansangan is 21 years old!	This is Elec ' favorite food: HOTDOG Is it healthy? Yes Elec is 23 years old!	This is Pasion ' favorite food: BANANA Is it healthy? Yes Pasion is 69 years old!	This is Talusig ' favorite food: BANANA Is it healthy? Yes Talusig is 43 years old!

PersonalProfile

```
1 import React from 'react';
2 import Profile from '../components/Profile';
3 import Header from '../components/Header';
4 import Footer from '../components/Footer';
5
6 import './App.css';
7 import avatarImage from "../assets/me.jpg";
8
9 function App() {
10   const profileData = {
11     name: "Macaraeg, Jake Russell F.",
12     bio: "A third-year student pursuing a Bachelor of Science in Information Technology (BSIT). Enjoys improving and acquiring new skills and being up to date on technology.",
13     avatar: avatarImage,
14     hobbies: ["Sleeping", "Drawing", "Reading", "Gaming", "Traveling"],
15   };
16
17   return (
18     <div className="App">
19       <Header />
20       <Profile {...profileData} />
21       <Footer />
22     </div>
23   );
24 }
25
26 export default App;
27
```

My Personal Profile



Macaraeg, Jake Russell F.

A third-year student pursuing a Bachelor of Science in Information Technology (BSIT). Enjoys improving and acquiring new skills and being up to date on technology.

Hobbies:

Sleeping
Drawing
Reading
Gaming
Traveling

Experience:

- PLFORM1:
Program Logic Formulation
First Semester, 2022 - 2023

- HUMCOM1:
Human Computer Interaction
Second Semester, 2022 - 2023

- PROGIT1:
Computer Programming
Second Semester, 2022 - 2023

- DSALGO1:
Data Structures and Algorithms
First Semester, 2023 - 2024

- WEBSYS1:
Web Systems and Technologies
First Semester, 2023 - 2024

- PROGIT2:
Object-Oriented Programming
Summer, 2024 - 2025

Midterm Activities

react-color-picker

```
Macaraeg_JakeRussell_MidtermActivities > react-color-picker 01 >  
1  import { useState } from 'react'  
2  import Footer from './Footer'  
3  import Header from './Header'  
4  import ColorPicker from './ColorPicker'  
5  
6  function App() {  
7    return (  
8      <>  
9        <Header />  
10       <ColorPicker />  
11       <Footer />  
12     </>  
13   )  
14 }  
15  
16 export default App
```

My Color Picker: A Web Application

Color Picker

Selected Color:
#FFFFFF

Select a Color:

Randomize



react-counter

```
Macaraeg_JakeRussell_MidtermActivities > react-counter 02 > src > %  
1  import { useState } from 'react'  
2  import Footer from './Footer'  
3  import Header from './Header'  
4  import Counter from './Counter'  
5  
6  
7  function App(){  
8    return(  
9      <>  
10     <Header />  
11     <Counter />  
12     <Footer />  
13     </>  
14   )  
15 }  
16 export default App  
17
```

My Counter

Counter:
6

Decrement

Reset

Increment

react-quote-generator

My Quote Generator Website

Quote Generator

I) Hey, You, You're Finally Awake.

Previous

Random

Next

© 2024 My Quote Generator Website || Written By: Jake Russell Macaraeg

```
Macaraeg_JakeRussell_MidtermActivities > react-quote-generator 03 > src
1  import Footer from './Footer';
2  import Header from './Header';
3  import QuoteGenerator from './QuoteGenerator';
4
5  function App() {
6    return (
7      <>
8        <Header />
9        <QuoteGenerator />
10       <Footer />
11     </>
12   );
13 }
14
15 export default App;
16
```

react-cond-rendering

⋮
[Home](#)
[About](#)
[Contacts](#)

My Favorite Fruit

Apple: red

Banana: yellow

Orange: Orange

Grape: purple

Kiwi: green

My Favorite Desserts

Cake: red

Ice Cream: blue

Pie: green

© 2024 My Other Food Website | Written by: Macaraeg, Jake Russell F.

```
Macaraeg_JakeRussell_MidtermActivities > react-cond-rendering 04 > src > App.jsx > ...
1  import { useState } from 'react'
2  import './App.css'
3  import Header from './Header'
4  import Footer from './Footer'
5  import ListOfFruits from './ListOfFruits'
6
7  function App() {
8    const fruits = [
9      {id: 1, name: "Apple", color:"red"},
10     {id: 2, name: "Banana", color: "yellow"},
11     {id: 3, name: "Orange", color: "Orange"},
12     {id: 4, name: "Grape", color: "purple"},
13     {id: 5, name: "Kiwi", color: "green"}
14   ]
15
16   const desserts = [
17     {id: 1, name: "Cake", color:"red"},
18     {id: 2, name: "Ice Cream", color:"blue"},
19     {id: 3, name: "Pie", color:"green"},
20   ]
21
22   return (
23     <>
24     <Header/>
25     {fruits.length > 0 && <ListOfFruits items = {fruits} category = "My Favorite Fruit"/>}
26     {desserts.length > 0 && <ListOfFruits items = {desserts} category = "My Favorite Desserts"/>}
27     <Footer/>
28   </>
29 )
30
31 export default App
32
```

updating-arrays-react

This Is My List Of Favorite Games

These are my Favorite Games in 2024:

- GOW
- Genshin Impact
- Elden Ring
- Star Rail
- asdw

© 2024 My Games Website || Written by: Macaraeg, Jake Russell F.

```
Macaraeg_JakeRussell_MidtermActivities > updating-arrays-react 05 > src > ⚙ App.jsx
1  import { useState } from 'react'
2  import reactLogo from './assets/react.svg'
3  import viteLogo from '/vite.svg'
4  import './App.css'
5  import Header from './Header'
6  import Footer from './Footer'
7  import FavoriteGames from './FavoriteGames'
8
9  function App() {
10     const [count, setCount] = useState(0)
11
12     return (
13         <>
14             <Header/>
15             <FavoriteGames/>
16             <Footer/>
17         </>
18     )
19 }
20
21 export default App
22
```


Final Term Activities

react-on-change-eventHandling

Exercise #1 (Final Term)



Jeremy Moses Ebreo • Nov 20

100 points

Due Nov 20, 5:00 PM

Create a react App called **react-on-change-eventHandling** and insert the following components attached from this thread to the application. Make sure to study the code for you to complete the application for the Activity #2 (Finals). This application introduces the concept of events and onchange handlers for the react application. Include also minimal CSS Stylings for the Exercise #1.



react-on-change-eventHan...
Compressed Archive

My Website

[Home](#)
[About](#)
[Contact](#)

Student Attendance

Fill in the details:

ID Number:

Name:

Course:

Age:

Gender:

☐ Male ☐ Female

Date:

Submit

Students Added:

ID: 20207096

Name: JAKE RUSSELL MACARAEG

Course: BSIT

Age: 21

Gender: Male

Date: 2024-12-02

© 2024 My Website || Written by: Jake Russell Macaraeg

```
Macaraeg_JakeRussell_FinalsActivities > react-on-change-eventHandling > src > Form.jsx > ...
1  import React, { useState } from 'react';
2  import StudentCard from './StudentCard';
3
4  var attendance = [];
5
6  export default function Form() {
7    const [id, setId] = useState(0);
8    const [name, setName] = useState('');
9    const [course, setCourse] = useState('');
10   const [age, setAge] = useState(0);
11   const [gender, setGender] = useState('');
12   const [date, setDate] = useState('');
13
14   function handleIdChange(event) {
15     setId(event.target.value);
16   }
17   function handleNameChange(event) {
18     setName(event.target.value);
19   }
20   function handleCourseChange(event) {
21     setCourse(event.target.value);
22   }
23   function handleAgeChange(event) {
24     setAge(event.target.value);
25   }
26   function handleGenderChange(event) {
27     setGender(event.target.value);
28   }
29   function handleDateChange(event) {
30     setDate(event.target.value);
31   }
32 }
```

react-shoppinng-cart

This is the shop:

- Item 1
Price per piece: \$10
Add to Cart!
- Item 2
Price per piece: \$20
Add to Cart!
- Item 3
Price per piece: \$30
Add to Cart!

Cart Items:

- Item 1
Price per piece: \$10
- Item 2
Price per piece: \$20
- Item 3
Price per piece: \$30

```
Macaraeg_JakeRussell_FinalsActivities > react-shopping-cart > src > App.jsx > ...
1  import { useState } from 'react'
2  //import Header from './components/Header'
3  //import Footer from './components/Footer'
4  //import Shop from './components/Shop'
5  import Login from './components/Login'
6
7
8  function App() {
9    let items = [
10     {id: 1, name: "Item 1", price: 10},
11     {id: 2, name: "Item 2", price: 20},
12     {id: 3, name: "Item 3", price: 30},
13     {id: 4, name: "Item 4", price: 40},
14     {id: 5, name: "Item 5", price: 50}
15   ] //LIST OF OBJECTS TO BE IMPORTED 0
16
17   return (
18     <>
19
20       <Login/>
21
22     </>
23   )
24 }
25
26
27
28
29 export default App
```



reactGroceryList-Macraeg-jakerussell

Activity #1 (Finals)

Jeremy Moses Ebreo • Nov 13

20 points

Create a react application that gives the users control over the items (**grapes, oranges, kiwis, bananas and cucumbers**) they will check out from a grocery. After checking out of the grocery, the application should be able to show the total price to be paid by the user, if the user has paid for the items, the web application should generate a **receipt** containing the **items the user has bought** as well as the **total price for the items, the price the user has paid and also the change.**

Project Name: **reactGroceryList-LastName-FirstName**



UNIVERSITY OF
Baguio
SCHOOL OF INFORMATION TECHNOLOGY

General Luna Road, Baguio City Philippines 2600

Telefax No.: (074) 442-3071

Website: www.ubaguio.edu

E-mail Address: sit@e.ubaguio.edu

Grocery List

Grapes
Price: \$2.5
- 1 +
Add to Cart

Oranges
Price: \$3
- 1 +
Add to Cart

Kiwis
Price: \$4
- 1 +
Add to Cart

Bananas
Price: \$1.5
- 1 +
Add to Cart

Cucumbers
Price: \$2
- 1 +
Add to Cart

Cart

2 Grapes - \$5
1 Oranges - \$3
3 Kiwis - \$12
5 Cucumbers - \$10

Total Price: \$30.00

Receipt

2 Grapes - \$5
1 Oranges - \$3
3 Kiwis - \$12
5 Cucumbers - \$10

Total Price: \$30.00

Paid Amount: \$43.00

Change: \$13.00

Written By: *Jake Russell F. Macaraeg* - 2024

```
Macaraeg_JakeRussell_FinalsActivities > reactGroceryList-Macaraeg-JakeRussell > src > App.jsx > ...
1  import React, { useState } from 'react';
2  import Header from './components/Header';
3  import GroceryItem from './components/GroceryItems';
4  import Cart from './components/Cart';
5  import Checkout from './components/Checkout';
6  import Receipt from './components/Reciept';
7  import Footer from './components/Footer';
8  import './index.css';
9
10 const App = () => {
11   const [cart, setCart] = useState([]);
12   const [isCheckedOut, setIsCheckedOut] = useState(false);
13   const [paidAmount, setPaidAmount] = useState(0);
14
15   const items = [
16     { id: 1, name: 'Grapes', price: 2.5 },
17     { id: 2, name: 'Oranges', price: 3 },
18     { id: 3, name: 'Kiwis', price: 4 },
19     { id: 4, name: 'Bananas', price: 1.5 },
20     { id: 5, name: 'Cucumbers', price: 2 },
21   ];
22
23   const addToCart = (item) => {
24     const existingItem = cart.find((cartItem) => cartItem.id === item.id);
25     if (existingItem) {
26       setCart(cart.map((cartItem) =>
27         cartItem.id === item.id
28         ? { ...cartItem, quantity: cartItem.quantity + item.quantity }
29         : cartItem
30       ));
31     } else {
32       setCart([...cart, item]);
33     }
34   };
35 }
```

GameReview

Activity Guide #2 (Final Term)



Jeremy Moses Ebreo • Nov 20

100 points

Due Nov 20, 5:00 PM

Study the code below attached in this activity thread. Create a Login component and a game review component for the application below and add CSS Styling for the page. The application should let the user login to the application and it should let the user add a game review for a particular game. For the game review, add also the **DATE** for when the review was made.



[GameReview.zip](#)

Compressed Archive

Game Review

Add a Game Review

Add Review

--- Reviews: ---

[asd](#) - asd (12/2/2024)

© 2024 Game Review | Written by: Jake Russell Macaraeg

```
Macaraeg_JakeRussell_FinalsActivities > GameReview > src > Reviews.jsx > ...
1  import React, { useState } from 'react';
2
3  export default function Reviews() {
4    const [reviews, setReviews] = useState([]);
5    const [gameTitle, setGameTitle] = useState("");
6    const [review, setReview] = useState("");
7
8    const handleGameTitleChange = (e) => setGameTitle(e.target.value);
9    const handleReviewChange = (e) => setReview(e.target.value);
10
11    const addReview = () => {
12      const newReview = {
13        gameTitle,
14        review,
15        date: new Date().toLocaleDateString(),
16      };
17      setReviews([...reviews, newReview]);
18      setGameTitle("");
19      setReview("");
20    };
21
22    return (
23      <div className="reviews-container">
24        <h2>Add a Game Review</h2>
25        <input
26          type="text"
27          placeholder="Game Title"
28          value={gameTitle}
29          onChange={handleGameTitleChange}
30        />
```



react-routing

- [Home](#)
- [About](#)
- [Games](#)

Games List:

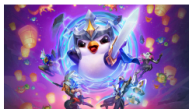
- [TFT](#) - This is an auto-battler game.
- [GOW](#) - God of War, a popular action game.
- [ER](#) - Elden Ring, an open-world RPG.
- [HSR](#) - Honkai Star Rail, a turn-based RPG.

```
Macaraeg_JakeRussell_FinalsActivities > react-routing > src > App.jsx > ...
1  import { useState } from 'react';
2  import { Link, Routes, Route } from 'react-router-dom';
3  import Home from './components/Home';
4  import About from './components/About';
5  import Games from './components/Games';
6  import Game from './components/Game';
7  import NotFound from './components/NotFound';
8
9  function App() {
10     const [count, setCount] = useState(0);
11
12     return (
13         <>
14             <nav>
15                 <ul>
16                     <li>
17                         <Link to="/">Home</Link>
18                     </li>
19                     <li>
20                         <Link to="/about">About</Link>
21                     </li>
22                     <li>
23                         <Link to="/games">Games</Link>
24                     </li>
25                 </ul>
26             </nav>
27             /* Routes for the application */
28             <Routes>
29                 <Route path="/" element={ <Home /> } />
30                 <Route path="/about" element={ <About /> } />
31                 <Route path="/games">
32                     {}
33                     <Route index element = { <Games /> } />
34                     <Route path=":gameId" element = { <Game /> } />
35                 </Route>
36                 <Route path="*" element={ <NotFound /> } />
37             </Routes>
38         </>
39     );
40 }
41
42 export default App;
43
```


react-routing part 2

[Home](#) [About](#) [Games](#)

Games List



Team Fight Tactics

A strategic auto-battler!

[View Details](#)



God Of War (2018)

An action-packed game with a gripping story.

[View Details](#)



Elden Ring

A challenging RPG with stunning visuals.

[View Details](#)



Honkai Star Rail

A unique turn-based RPG with engaging gameplay.

[View Details](#)



Grand Theft Auto 6

The most anticipated game of all time.

[View Details](#)

```
Macaraeg_JakeRussell_FinalsActivities > react-routing-part2 > src > App.jsx >
1  import { Routes, Route } from 'react-router-dom';
2  import Navbar from './components/Navbar';
3  import Home from './components/Home';
4  import About from './components/About';
5  import Games from './components/Games';
6  import Game from './components/Game';
7  import NotFound from './components/NotFound';
8  import './components/Styles.css';
9
10 function App() {
11   return (
12     <>
13       <Navbar />
14       <Routes>
15         <Route path="/" element={<Home />} />
16         <Route path="/about" element={<About />} />
17         <Route path="/games">
18           <Route index element={<Games />} />
19           <Route path=":gameId" element={<Game />} />
20         </Route>
21         <Route path="*" element={<NotFound />} />
22       </Routes>
23     </>
24   );
25 }
26
27 export default App;
28
```

Reflections:

You can use the following questions to answer the reflections part of your compilation for the activities.

- What have you learned in the activities provided in class?

Through the activities in APPDEV1, I have gained practical knowledge in building interactive web applications using React. I've learned how to manage components, handle state, and use hooks to create dynamic user interfaces. The activities have also helped me understand the importance of structuring applications efficiently and using best practices for maintainability and scalability.

- How will you use the things you have learned on the provided activities as students of the university and as a part of the workforce?

As a student, the skills learned in React will help me in academic projects, making it easier to build functional, user-friendly applications for assignments. In the workforce, I can use these skills to contribute to building modern web applications, collaborate with teams using version control tools, and stay up to date with the latest trends in web development. As a student, the skills learned in React will help me in academic projects, making it easier to build functional, user-friendly applications for assignments. In the workforce, I can use these skills to contribute to building modern web applications, collaborate

with teams using version control tools, and stay up to date with the latest trends in web development.

- What are important things you have learned in the class which can be beneficial for you in your professional development?

Key takeaways from the class include mastering React for building scalable web applications, understanding the concept of component-based development, and using React hooks for more efficient code. These skills are essential in professional development, as React is widely used in the industry, and knowledge of it will make me competitive in the job market. Additionally, learning how to optimize performance and debug code will be valuable in real-world development environments.

Github link:

<https://github.com/JakeRussellMacaraeg/APPDEV-Projects.git>