

# Round 2 Evaluation

Week 10 (Section B)

## Short Question 1 (html)

Identify the element, attribute, and text in the following HTML code, and correct any mistakes if found:

(A) `<a href="mypage.html"> This is a link to my page</a>`

(B) `<input type="text" name="username" value="HT">`

(C) `<a href="https://www.example.com" target="new">Click here to visit Example</a>`

(D) `<button type="button" onclick="alert('Hello!')">Click Me</button>`

## Solution Question 1

	Element	Attribute(s) are key-value pair	Text / Value	Tag Type	Mistake + Correction Comment
A	<a>	href="mypage.html">	This is a link to my page	Normal (Paired)	Mistake: Missing closing quote in href. Correct: href="mypage.html"
B	<input>	type="text" name="username" value="HT"	No inner text (void element)	Void	Mistake: Wrong quotes "HT" Correct: Use straight quotes 'HT'
C	<a>	href="https://www.example.com" target="new"	Click here to visit Example	Normal (Paired)	Mistake: Wrong attribute name target. Correct: Use target="_blank" for opening in new tab target="_blank" is optional — but it's important when the user experience (UX) matters.
D	<button>	type="button" onclick="alert('Hello!')"	Click Me	Normal (Paired)	No mistake Student mix plain html with React JSX

Void tag: No closing tag required e.g., <input>

Normal tags: Must have closing tag e.g., <a></a> or <button></button>

Self closing: Optional in HTML5 but common in XHTML e.g., <img />

## Short Question 2(css)

(a) There are 3 ways to include CSS on a page, what are they? What is the “best” way?

(b) What the difference in CSS between a class and an id?

(c) Create a **custom class called “caution”** that will make the text red with a font-size of 20px. This class needs to be able to be used by all tags.

Apply the custom class you created above to the word  
“special” below:

The word

Special

has been formatted using a custom class.

(d) Find and fix the errors in the following CSS code

```
h1 [  
font-family; Verdana, sans-serif,  
color; red,  
font-size; 20px,  
]  
p, div, h2 [color; #00DDFF, width; 80%, ]
```

## Solution Question 2 (a)

Method	Description	Example	Pros	Cons	Best Use Case
<b>Inline CSS</b>	CSS directly in the style attribute of an HTML element.	<code>&lt;button style="color: blue;"&gt;Click Me&lt;/button&gt;</code>	Quick for testing; Overrides other styles	Difficult to manage for large projects; Not reusable	Small, one-off styles or testing
<b>Internal CSS</b>	CSS written inside the <code>&lt;style&gt;</code> tag in the <code>&lt;head&gt;</code> section of HTML.	<code>&lt;style&gt; button {color: red;} &lt;/style&gt;</code>	Styles within the same file; Easy for single-page sites	Clutters HTML file; Not reusable across pages; Harder to maintain	Small projects or single-page sites
<b>External CSS</b>	CSS written in a separate .css file and linked to the HTML file.	<code>&lt;link rel="stylesheet" href="styles.css"&gt;</code>	Best separation of concerns; Reusable; Easier to maintain; Faster performance (with caching)	Additional HTTP request (minor impact)	Large projects, multiple pages, scalability

## Solution Question 2 (b)

Feature	Class (.)	ID (#)
Technical Description	Used to style multiple elements together	Used to style a single unique element
Syntax	<code>.classname { }</code>	<code>#idname { }</code>
Uniqueness	Reusable — many elements can share it	Unique — only one element should have it
CSS Example	<code>.btn { color: white; background-color: green; padding: 10px; }</code>	<code>#navbar { background-color: black; color: white; padding: 20px; }</code>
HTML Example	<code>&lt;button class="btn"&gt;Save&lt;/button&gt;</code> <code>&lt;button class="btn"&gt;Delete&lt;/button&gt;</code> <code>&lt;button class="btn"&gt;Cancel &lt;/button&gt;</code>	<code>&lt;div id="navbar"&gt;This is the navigation bar&lt;/div&gt;</code>



## Solution Question 2 (c)-(d)

```
.caution {  
  color: red;  
  font-size: 20px;  
}
```

```
<body>  
  <p>The word <span class="caution">Special</span> has been formatted using a custom class.</p>  
</body>
```

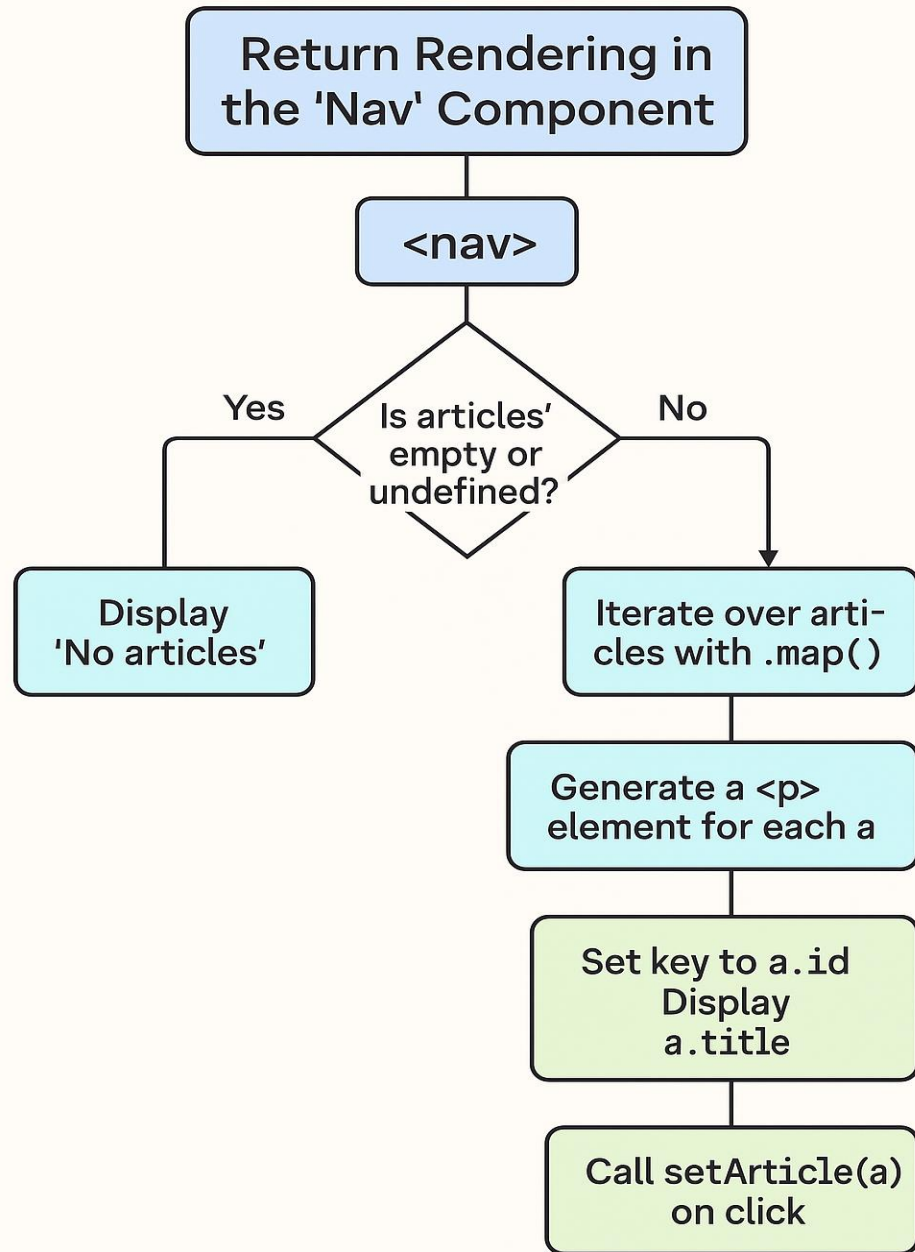
Mistake	Problem	Correction
[]	Wrong brackets	Use { }
; after property	Wrong separator	Use : between property and value
Extra comma ,	Syntax error	Remove comma from last property

```
h1 {  
  font-family: Verdana, sans-serif;  
  color: red;  
  font-size: 20px;  
}
```

```
p, div, h2 {  
  color: #00DDFF;  
  width: 80%;  
}
```

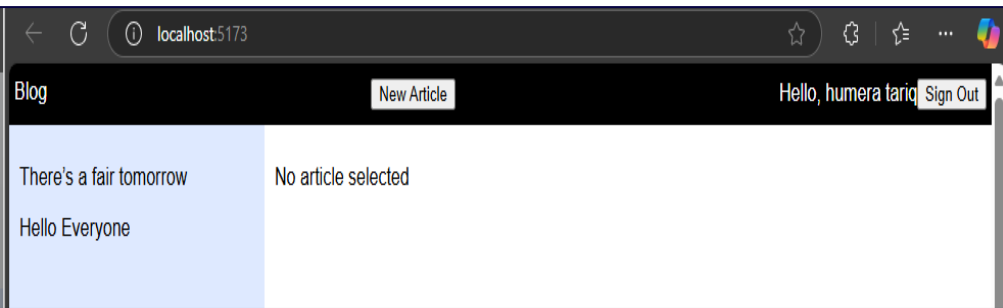
## Question 3 (React)

# Complete the missing <Nav> code



```
export default function Nav({ articles, setArticle }) {  
  return (  
    <nav>  
      {!articles  
        ? _____  
        : _____}.map((a) => (  
          <p  
            _____  
          </p>  
        ))  
      </nav>  
    )  
  }  
}
```

## Solution Question 3



```
export default function Nav({ articles, setArticle }) {  
  return (  
    <nav>  
      {!articles  
        ? “No Articles”  
        : articles.map((a) => (  
          <p  
            key={a.id}  
            onClick={()=>setArticle(a)}>{a.title}  
          </p>  
          )  
        )  
      }  
    </nav>  
  )  
}
```

## Question 4 (RESTful API/http communication)

Write a technical note on RESTful API clearly stating principals it works upon.

Any supporting illustration/Architecture plan / code snippet will make you close to full rewards.

Extra credit: Do you know anything comparable/better than RESTful API ? Briefly explain.

## Question 5 (RESTful API/http communication)

Write the question so that I can give you skeleton code on next slide.

You are building a simple **RESTful API** using **Node.js** and **Express**. The API should have two routes:

1. **GET** /api/users - Fetches a list of all users.
2. **POST** /api/users - Adds a new user to the list.

Your task is to modify the **POST** route to:

- ✓ Ensure that the email provided in the request body is **valid** (it should contain both @ and .).
- ✓ If the email is **valid**, the user should be added to the list and a success message should be returned.
- ✓ If the email is **invalid**, return an error message saying "Invalid email address".



```
import express from "express";
import cors from "cors";
const app = express();
const PORT = process.env.PORT || 5000;
```

```
// Middleware
app.use(cors());
app.use(express.json());
```

```
// Mock Database (in-memory list)
```

```
let users = [
  { id: 1, name: "ht", email: "humera@uok.edu.pk" },
  { id: 2, name: "acp", email: "acp@example.com" },
];
```

```
app.get("/", (req, res) => {
  res.send("Welcome to the Users API! Use /api/users to fetch data.");
});
```

```
// GET - Fetch all users
```

```
app.get("/api/users", (req, res) =>
  res.json({ status: "success", users }));
```

```
// POST - Add a new user
```

```
app.post("/api/users", (req, res) => {
  const { name, email } = req.body;
```

```
  // TODO: Add simple email validation (check for '@' and '.'). You may use Regular expression to show mastery.
```

```
  // If email is invalid, return an error message
  // If email is valid, add the user to the "users" array
```

```
  // Example Response:
```

```
  // res.status(201).json({ status: "success", user: { id, name, email } });
});
```

```
// Start the server
```

```
app.listen(PORT, () => console.log(`Server running at http://localhost:${PORT}`));
```

(a) Your task is to modify the **POST** route

(b) Identify and Prepare a list of JS features involved in the code and write 2-3 impressive lines about them.

## Compulsory Bonus Question 6

(a) Assume yourself as future supervisor of BSCS 633 IAD course

Briefly explain the one-topic that you love to teach/explain/speak about. You can pick the topic that is even not covered yet in class but is tightly relevant to IAD.

(b) Design/propose, one best question for final exam

# Round 2 Evaluation

Week 10 (Section A)



My question is why 'header' tags looks different in syntax from react components 'Nav' , 'Article Entry ' and 'article' which seems to use Curley braces?

```
<header>
  Blog
  {user && <button onClick={() => setWriting(true)}>New Article</button>}
  {!user ? <SignIn /> : <SignOut />}
</header>
```

```
{!user ? "" : <Nav articles={articles} setArticle={setArticle} />}
```

```
<ArticleEntry addArticle={addArticle} />
```

```
<Article article={article} />
```



# App.jsx



```
export default function App() {  
  
  // Array state to store all articles  
  // Initialized as empty array, will hold objects of article data  
  const [articles, setArticles] = useState([])  
  
  const user = useAuthentication()  
  
}
```

```
<Nav articles={articles} setArticle={setArticle} />
```

**Briefly explain the `<Nav ....>` line purpose and modify the rendering to conditional based on user login state.**





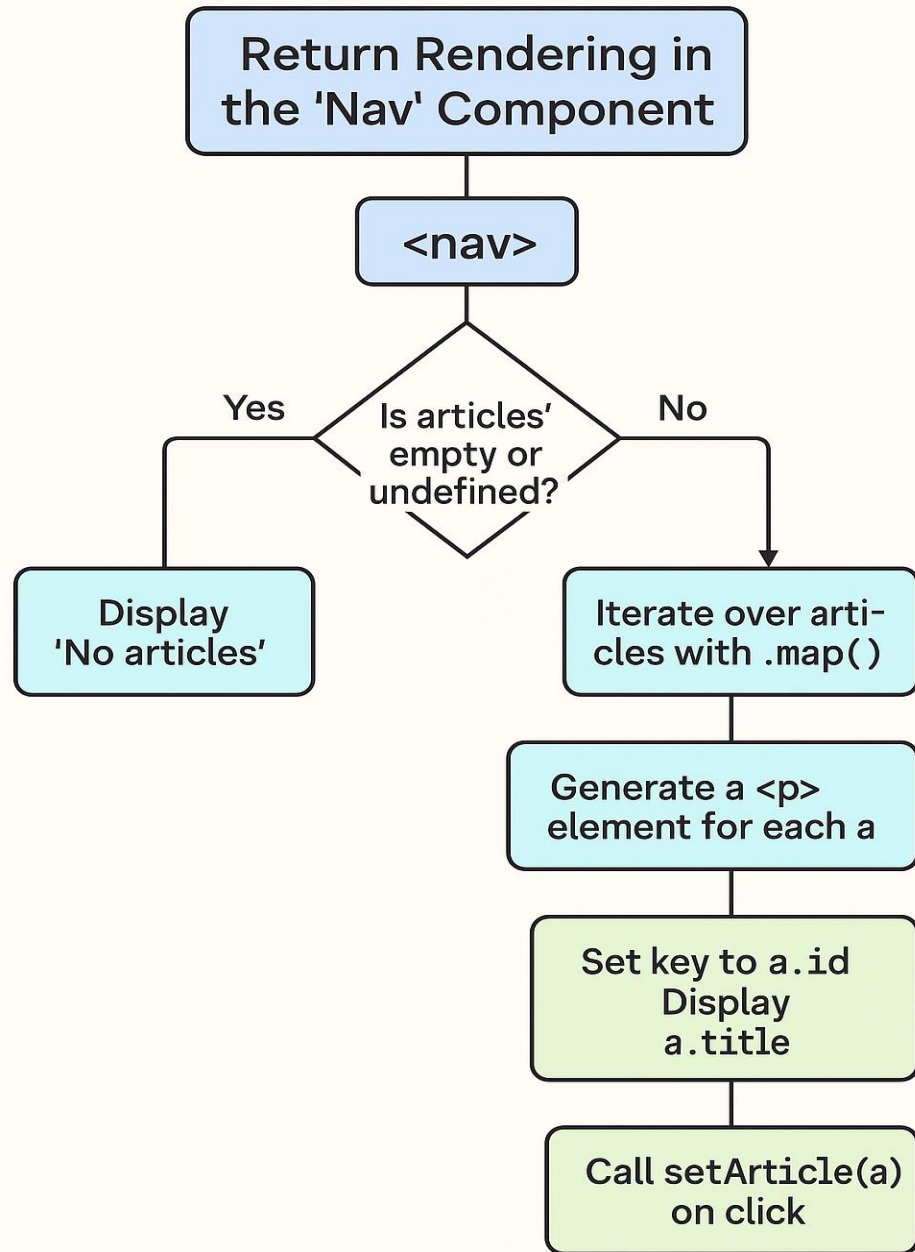
## Question 4(a)

Advise (Write + Define) few good JS terms that you already learned or believe must be covered in IAD course in upcoming classes

## Question 4(b)

Advise (Write + Define) few good React terms that you already learned or believe must be covered in IAD course in upcoming classes

# Complete the missing <Nav> code



```
export default function Nav({ articles, setArticle }) {  
  return (  
    <nav>  
      {!articles  
        ? _____  
        : _____}.map((a) => (  
          <p  
            _____  
          </p>  
        ))  
    </nav>  
  )  
}
```

## Question 5

Write a technical note on web application using CRUD operations

AND/OR

Write a technical note on http communication using best vocabulary and supporting code.

Technical note : Application architecture plan, block diagrams, code snippets, pseudocode ,

# Bonus Question

Assume yourself as future supervisor of  
BSCS 633 IAD course

Design/propose, one best question for final exam

## Question 6

Discuss any reasonable html-css-js example of your choice you learn/know/practice.

OR

Discuss any relevant IAD concept better than above with out overlapping/conflict with previous questions.