Round 2 Evaluation

Week 12 (Section B)

Question 1 (Basics)

(1) How would you allow using import syntax (ES Modules) in Node.js? Fill in the missing package.json property:

```
{
  "type": "______"
}
```

(2) In your React app, complete the fetch call to get data from backend API:

```
  ____________ (() => {
  fetch('http://localhost:5000/_____/hello')
    .then(response => response.text())
    .then(data => setMessage(data));
}, _________ );
```

(3) An api route might looks like:

```
app.get('_______________', (req, res) => {
  res.send('Hello world!');
});
```

(4) To allow the React frontend (running on a different port) to communicate with our Express backend without security errors, we must enable resource sharing. We do this by **adding** ________ **middleware** inside our Express server using:

```
app.use __________________________ .
```

(5) Cloze passage

When a client sends an HTTP request, the server responds with a ____________ (Hint: status code / URL) indicating whether the request succeeded or failed.

Common success status codes include ____________ (Hint: 200 / 404), while an error like "Not Found" is indicated by ____________ (Hint: 200 / 404).

During a POST request, the client often needs to send headers such as 'Content-Type': '____________' (Hint: application/json / text/html) to inform the server how to interpret the body.

An Express backend can extract JSON data from incoming requests using the middleware express.____________() (Hint: json / router).

For debugging, developers often inspect network traffic in the browser's ____________ (Hint: DevTools / IDE) under the **Network** tab.

Question 2 (Reasoning)

(6) Reasoning

(1) Why do we prefer using POST instead of GET when submitting forms that handle passwords or other sensitive credentials? Give at least two reasons.

(2) When building a React app that communicates with a backend server, why is it a bad idea to hardcode the server URL (like http://localhost:5000) directly into your fetch calls? Explain two problems this can cause, especially when deploying your application to production. Suggest a solution.

Question 3 (html-css)

Question:3 Analyze the code below and answer these questions:

1.The **navbar** is not sticking at the top when scrolling. What's missing or incorrect in the CSS, and how would you fix it?

2.The **footer** is not staying at the bottom of the page when the content is short. How would you adjust the layout to ensure the footer stays at the bottom?

3.The **footer** might overlap the content when scrolling if the navbar is fixed. What CSS property should be added to the **footer** or **main** to prevent this overlap?

```
<header class="navbar">
  <a href="#">Home</a>
</header>
<main>
  <p>Content...</p>
</main>
<footer class="footer">
  <p>&copy; 2025</p>
</footer>
```

```
.navbar {
  position: relative;
  background: #333;
  color: white;
}
footer {
  position: relative;
  background: #333;
  color: white;
}
```

Question 4 (JS)

What is Node.js?

Node.js is an ________ , ________ **JavaScript runtime environment** that allows developers to execute JavaScript code on the ________ side. It was released in 2009 by Ryan Dahl and is built on the **Chrome V8 JavaScript engine**. Node.js enables the development of scalable and efficient network applications by allowing JavaScript to run outside of a ____________ .

Java Script is ______ , ____ , ______ , ____language.

____.

____.

____.

____.

Question 5 (JS)

What is the output of ObjArg.js.

Why ?

```
// ObjArg.js

function objArgs(param1, param2) {
  // Change the data in param1 and its argument
  param1.data = "changed";
  // Change the object referenced by param2, but not its argument
  param2 = param1;

  window.alert("param1 is " + param1.data + "\n" +
               "param2 is " + param2.data);
  return;
}

// Create two different objects with identical data
var o1 = new Object();
o1.data = "original";
var o2 = new Object();
o2.data = "original";

// Call the function on these objects and display the results
objArgs(o1, o2);
window.alert("o1 is " + o1.data + "\n" +
             "o2 is " + o2.data);
```

What is the output of ObjArg.js.

Why ?

```
// BTNode.js

// BTNode(value) is a constructor for a binary tree node.
// It initializes its value to the given argument.
// It also adds an isLeaf() method to the node.
function BTNode(value) {
  // Notice that we no longer need to create an Object
  // and that we use "this" to reference the object
  // initialized.
  this.left = this.right = null;
  this.value = value;
  this.isLeaf =
    function leaf() {
      return this.left == null && this.right == null;
    };
  // Notice that we no longer return a value.
}
// Create and initialize two node objects, making the second
// a child of the first.
// Notice the use of "new" to call a function as a constructor.
var node1 = new BTNode(3);
var node2 = new BTNode(7);
node1.right = node2;

// Output the value of isLeaf() on each node
window.alert("node1 is a leaf: " + node1.isLeaf());
window.alert("node2 is a leaf: " + node2.isLeaf());
```

FIGURE 4.13 Program that defines and uses an object constructor.

Question 5 (Event Loop/React)

Q5: Write a detailed technical note on JS event loop mechanism. Use block diagram/call stack, code snippet , queues and promise description to get full reward to answer

OR

Q5: Write a detailed technical note on parent child communication in react using JSX rendering example and at-least one fetch.

Round 2 Evaluation

Week 12 (Section A)

Question 1 (Basics)

(b) Draw the UI that appeared on screen with html given on right.

```
<table border="5">
  <caption>
    COSC 400 Student Grades
  </caption>
  <tr>
    <td>&nbsp;</td><td>&nbsp;</td><th colspan="2">Grades</th>
  </tr>
  <tr>
    <td>&nbsp;</td><th>Student</th><th>Exam 1</th><th>Exam 2</th>
  </tr>
  <tr>
    <th rowspan="2">Undergraduates</th><td>Kim</td><td>100</td><td>89</td>
  </tr>
  <tr>
    <td>Sandy</td><td>78</td><td>92</td>
  </tr>
  <tr>
    <th>Graduates</th><td>Taylor</td><td>83</td><td>73</td>
  </tr>
</table>
```

3.7. Picture "framing."

(a) Write a style rule that will place a nice "frame" around `img` elements. The "frame" should be brown. The inside edges of the "frame" should touch the outside edges of the image. There should be 10-px distance between adjacent images (either horizontally or vertically). See the left image in Figure 3.44.

(b) Modify your style rule to "mat" your images. In particular, there should now be a 3-px gap between the outside edges of your images and the inside edges of the "frames." This gap should be a tan color. See the right image in Figure 3.44.

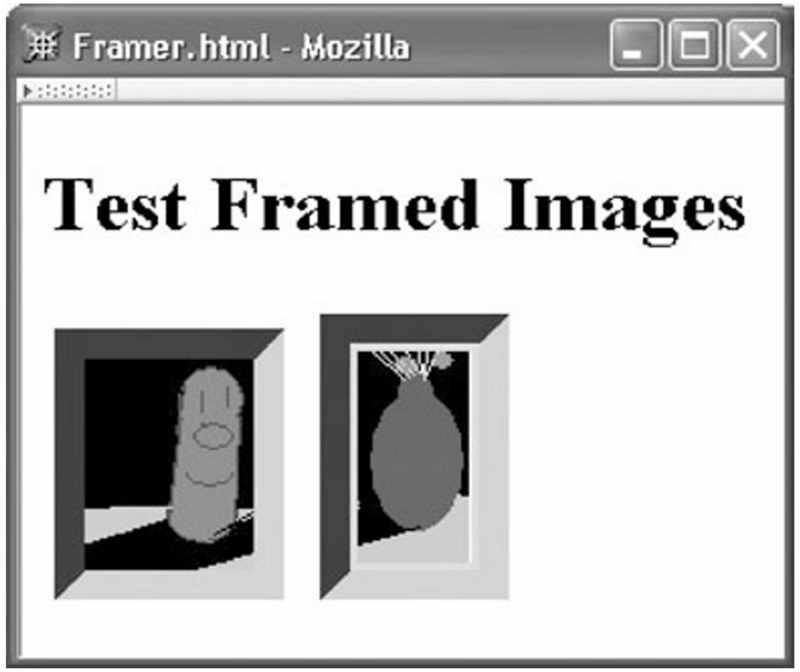

FIGURE 3.44 Two "framed" images. The right image is "matted." (Graphics courtesy of Ben Jackson.)

Question3 (JS)

Question 4 (JS)

What is the output of ObjArg.js

```
// ObjArg.js

function objArgs(param1, param2) {
  // Change the data in param1 and its argument
  param1.data = "changed";
  // Change the object referenced by param2, but not its argument
  param2 = param1;

  window.alert("param1 is " + param1.data + "\n" +
               "param2 is " + param2.data);
  return;
}

// Create two different objects with identical data
var o1 = new Object();
o1.data = "original";
var o2 = new Object();
o2.data = "original";

// Call the function on these objects and display the results
objArgs(o1, o2);
window.alert("o1 is " + o1.data + "\n" +
             "o2 is " + o2.data);
```

(i) The default security policy enforced by browsers is called the **Same-Origin Policy**, which blocks ______??______ requests between different origins. Same-Origin Policy prevents scripts on one website from making requests to another website's domain unless explicitly allowed by the server.

(ii) CORS stands for **Cross-Origin Resource Sharing**, a mechanism that allows or restricts _____??________ between different domains. CORS ensures that a client (like a browser) can securely request resources **(data, scripts, APIs)** from a server hosted on a different origin.

(iii) If you see a browser error like: “Access to fetch at 'https://api.example.com/data' from origin 'http://localhost:3000' has been blocked by CORS policy”. This indicates that the backend server does not include the appropriate ______??__________________? in its response.

Question 5 (Event Loop)

Q5: Write a detailed technical note on JS event loop mechanism. Use block diagram/call stack, code snippet , queues and promise description to get full reward to answer

OR

Q5: Write a detailed technical note on parent child communication in react using JSX rendering example and at-least one fetch.

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