

Web Developer Intern Assessment

Section 1

Q: What are the differences between `==` and `===` in JavaScript?

A: In JavaScript, both `==` and `===` are used to compare values, but they differ in how they handle the comparison:

- The first is loose equality, which means that JavaScript will try to convert the variables to a common type before comparing.
- The second is strict equality, which means that JavaScript will compare both the value and the type, and will return false if either the value or type is different.
- For instance, `"1" === 1` would return false, but `"1" == 1` would return true.

Section 2

Q: What is the purpose of React's `useEffect` hook? Provide an example use case.

A: The `useEffect` hook in React is used for handling side effects in functional components. Side effects are things that affect something outside the scope of the function, like fetching data, setting up a subscription, or manually changing the DOM.

It runs after the component renders and can re-run when certain data changes.

Example use case: Let's say we want to log a message every time the count changes or like above we display the count for the user. (In our code with `console.log()` a message every time the user add or subtract from the counter).

Section 3

Q: How does Node.js handle asynchronous operations, and why is this beneficial?

A: NodeJS handles asynchronous operations using an event-driven, non-blocking I/O model. This means that instead of waiting for operations like file reading, network requests, or database queries to complete, NodeJS will continue executing the rest of the code. When the asynchronous operation is done, a callback function or a promise will be triggered to handle the result.

Section 4

Q: What is the difference between `getStaticProps` and `getServerSideProps` in Next.js?

A: "getStaticProps" and "getServerSideProps" are both data-fetching methods in Next.js, but they serve different purposes:

- "getStaticProps" fetches data at build time, generating static pages that are served to all users, making it ideal for content that doesn't change frequently.
- In contrast, "getServerSideProps" fetches data on every request, rendering the page on the server each time, which is suitable for pages that require up-to-date or personalized content.

The key difference is when the data is fetched: "getStaticProps" is used for pre-rendering at build time, while "getServerSideProps" handles data fetching at request time.

Section 5

Q: What is a primary key in PostgreSQL, and why is it important?

A: A primary key in PostgreSQL is a unique identifier for each record in a table. It ensures that each row in the table can be uniquely identified by one or a combination of columns. The primary key is important because it enforces data integrity by preventing duplicate records and enabling efficient indexing, which speeds up data retrieval and ensures that each record can be accessed directly and reliably.

Section 6

Q1: Describe the steps to deploy a simple Node.js application to Azure App Services.

A: It can be done by using the CLI or directly through the Azure platform. I'll focus on the CLI:

1. **Create a Resource Group**

Open your terminal and run the following command to create a new resource group:

```
az group create --name myResourceGroup --location "East US"
```

2. **Create an App Service Plan**

Run the following command to create an App Service Plan:

```
az appservice plan create --name myAppServicePlan --resource-group myResourceGroup --sku FREE
```

3. **Create the Node.js App Service**

Use the following command to create a new App Service for your Node.js application:

```
az webapp create --resource-group myResourceGroup --plan myAppServicePlan --name bizinc-test --runtime "NODE|14-lts"
```

4. **Deploy Your Node.js Application**

- Navigate to your Node.js application directory in your terminal and use Git to initialize a repository (if not already done):

```
git init
```

- Add Azure as a remote repository:
git remote add azure https://bizinc-test.scm.azurewebsites.net:443/bizinc-test.git
- Add and commit your files:
git add .
git commit -m "Initial commit"
- Push your code to Azure to deploy:
git push azure master
- Alternatively, you can deploy directly using the Azure CLI:
az webapp up --name bizinc-test --resource-group myResourceGroup --location "East US"

5. **Access Your Deployed Application**

Once the deployment is complete, your Node.js application will be available at:
<https://bizinc-test.azurewebsites.net>

Q2: What is a common benefit of using Azure App Services for hosting applications?

A: A common benefit of using Azure App Services is its simplicity and ease of use, especially through the Azure Portal and CLI. It provides a fully managed environment, reducing the overhead of infrastructure management. Features like automatic scaling, load balancing, and continuous deployment make it easy to deploy and maintain applications with minimal operational effort. Additionally, the integration with development tools like GitHub, Bitbucket, and Azure DevOps facilitates continuous integration and deployment pipelines, streamlining the entire development and deployment process.

Section 7

Q: What is the difference between a WordPress post and a page?

A: A WordPress post is typically used for time-sensitive content like blog articles, news, or updates. Posts are organized by categories and tags, and they appear in reverse chronological order on your blog page. Posts usually have a comment section and are associated with a specific publication date.

A WordPress page, on the other hand, is used for static content that is not time-sensitive, such as an "About Us" or "Contact" page. Pages are hierarchical, meaning they can have parent-child relationships, and they do not use categories or tags. Pages are typically not part of the blog stream and are often used for content that is meant to be more permanent and structured.