Section 1: JavaScript

1. See file JavascriptArrayChallenge.js

2. The difference between '==' and '===' in Javascript has to do with type comparison. The '==' operator does not compare type and will return true if for example a String of value "10" is compared with an Int of 10. In this given example the '===' would return false.

Section 2: React

1. See the folder bizincreactchallenge and run on a browser

2. The useEffect hook is used to do something after render. An example of this would be fetching data from an endpoint, and using UseEffect to display the data after it has been received.

Section 3: Node.js

1. See the file app.js in the `Node` folder. Take note this is running on port 3002

2. There are multiple approaches to handling asynchronicity with node.js. I prefer to use promises, such as .then() and .catch(), but you can also use callbacks

Section 4: Next.js

1. See the folder /Next/nextchallenge. Type `npm run dev` in the terminal.

2. getStaticProps runs when the html code is built, and is served to the browser on each request.
getServerSideProps runs every time someone makes a request, it generates the html.

Section 5: PostgreSQL

1. `CREATE TABLE users (id varchar(255), name varchar(255), email varchar(255));`
`INSERT INTO users (id, name, email) VALUES (test123, John Doe, johndoe@gmail.com);`
`SELECT * FROM users`

2. The primary key is a unique identifier for each entry. It can be used to prevent duplicate entries, and search the table for specific items.

Section 6: Azure App Services

1. First create a node app using express app generator or npx create-node-app,

install all remaining packages and run rpm start. I use visual studio code so we'll be using that for this example. Sign into Azure in VSCode. Click the deploy to web app button in VSCode. Click create new web app. Run through all the prompts that it gives you. Boom, your node app is deployed to Azure.

2. There are many benefits to Azure. Load balancing and auto-scaling are some of the common benefits of using Azure.

Section 7: WordPress

1. See the folder Wordpress. Go to your Wordpress dashboard. Navigate to Plugins > Installed Plugins. Find the "greeting-shortcode" plugin and click activate. Now you can use the `[greeting]` shortcake in any post or page, and it will output "Welcome to My Site!"

2. A WordPress page is much like any other webpage; they are static and don't associate with any dates or times. Posts on the other hand often appear inside pages and have a date/time associated with them.