4/1/2021

Blog Site

Web Application Development Project

Adam Regan x19401956
Conor Fitzgerald x19444092
Group 17

Table of Contents

Introduction

This project required us to write a rich internet application using client/server scripting techniques. Our application allows a user to write short posts and display them on a wall on a web page. The user can also edit and delete a post on the wall to create a full CRUD functionality. The posts are stored in a MySQL database to ensure data persistence. To do this, we used a client-server model where the server interacts with the database to insert and pull data from the database to the client. The aim of this report is to give an overview of the technologies used in this application.

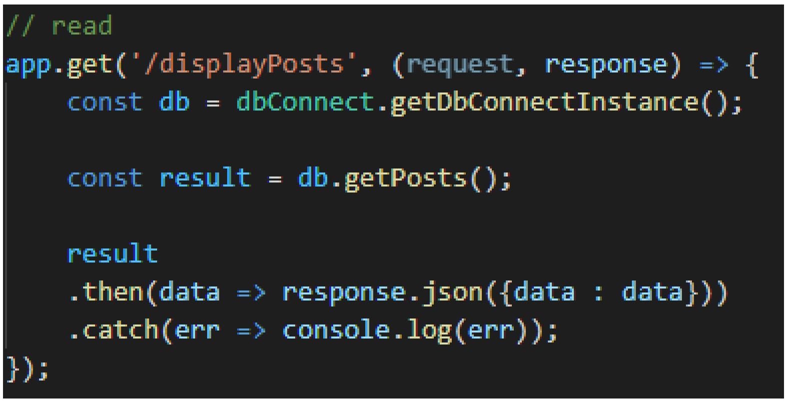
NodeJS

NodeJS is an "asynchronous event-driven JavaScript runtime " (NodeJS, 2020). It is a JavaScript framework that allows applications to run asynchronously. This means events are executed out of order using callback functions. This prevents a lot of issues related to deadlocking and inefficient use of resources.

NodeJS also comes with numerous libraries to help run your application. This was the main reason we choose to use it as it has libraries that help with problems such as connecting to a database and cross origin resource sharing. NodeJS handles the installations and managing of dependencies for you which makes it much easier to implement.

Express

Express is a NodeJS framework that speeds up the process of making robust web applications by taking care of functionality such as HTTP requests. The main one we used was body-parser which is "is a node.js middleware for handling JSON, Raw, Text and URL encoded form data" (tutorialspoint, Node.js - Express Framework, 2020). We used this because it manages connection to localhost and will listen for a request.

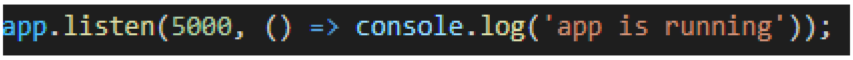


```
// read
app.get('/displayPosts', (request, response) => {
    const db = dbConnect.getDbConnectInstance();

    const result = db.getPosts();

    result
    .then(data => response.json({data : data}))
    .catch(err => console.log(err));
});
```

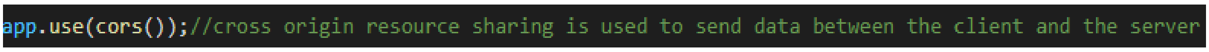
Example of express listening to the server port



```
app.listen(5000, () => console.log('app is running'));
```

CORS

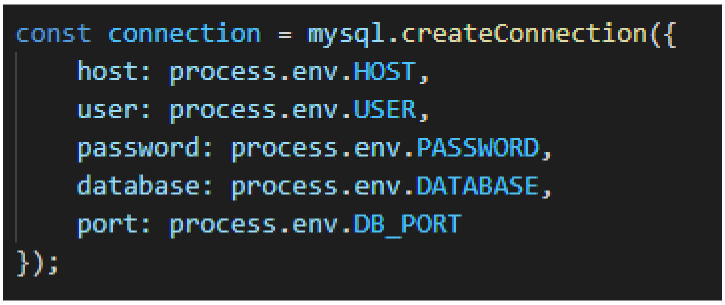
In order to send data between the client and the server, we had to use a library in NodeJS called CORS(Cross Origin Resource Sharing). This allows data to be sent between two different project files. This was important since our client side had to access to the NodeJS server without this and data could not be exchanged (npmjs, 2018).



```
app.use(cors());//cross origin resource sharing is used to send data between the client and the server
```

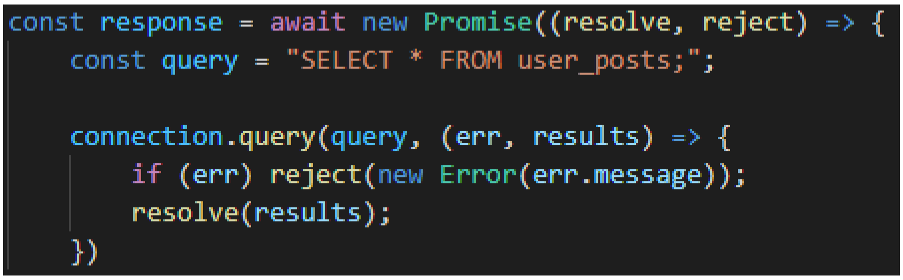
MySQL

We used MySQL as a database to store user posts. We used it because there it works well with NodeJS and we had some familiarity using it. To use MySQL with NodeJS we need to install it as a library. We can do this using npm install in the command prompt. From there, we write some code to give the NodeJS server the information it needs to connect to the database such as database name, host, user, password etc.



```
const connection = mysql.createConnection({
    host: process.env.HOST,
    user: process.env.USER,
    password: process.env.PASSWORD,
    database: process.env.DATABASE,
    port: process.env.DB_PORT
});
```

From here we can use sql commands to access and manipulate the data in the database from the server (W3Schools, Node.js MySQL Select From, 2020).

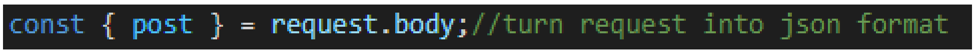


```
const response = await new Promise((resolve, reject) => {
    const query = "SELECT * FROM user_posts;";

    connection.query(query, (err, results) => {
        if (err) reject(new Error(err.message));
        resolve(results);
    })
```

AJAX and JSON

Since the client side has no access to the NodeJS server, we used AJAX and JSON to send data between the client and the server. AJAX stands for Asynchronous JavaScript and XML. It is used to exchange data between a client and a server asynchronously (W3Schools, 2020). JSON is a “JavaScript Object Notation is a lightweight text-based open standard designed for human-readable data interchange” (tutorialspoint, 2020). This is how we can take data from the database and convert it to JSON so it can be sent to the client using AJAX.



```
const { post } = request.body;//turn request into json format
```

JavaScript

We used JavaScript as the primary language to write the application since NodeJS is javascript based. This was used on both the client and server side to create the functionality of the application such as buttons etc.

Conclusion

The advantage of using NodeJS over PHP is that is easier to learn because it used JavaScript which is commonly used in web applications anyway. It also comes with a lot of built-in frameworks to help speed up development. NodeJS is also considered better for single page applications such as our one over PHP.

The disadvantage is the lack of community support compared to PHP which is much larger. However, there is still a large amount of support for various problems you might encounter. It also isn’t good at intensive applications that require audio or video (hackernoon, 2019)

Bibliography

hackernoon. (2019). *Node.js vs PHP: Which is better for web development?* Retrieved from hackernoon.com: https://hackernoon.com/nodejs-vs-php-which-is-better-for-your-web-development-he7oa24wp

NodeJS. (2020). *About Node.js*. Retrieved from nodejs.org: https://nodejs.org/en/about/

npmjs. (2018). *cors*. Retrieved from npmjs.com: https://www.npmjs.com/package/cors tutorialspoint. (2020). *JSON - Overview*. Retrieved from tutorialspoint.com: https://www.tutorialspoint.com/json/json_overview.htm

tutorialspoint. (2020). *Node.js - Express Framework*. Retrieved from tutorialspoint.com: https://www.tutorialspoint.com/nodejs/nodejs_express_framework.htm

W3Schools. (2020). *AJAX Introduction*. Retrieved from w3schools.com: https://www.w3schools.com/xml/ajax_intro.asp

W3Schools. (2020). *Node.js MySQL Select From*. Retrieved from w3schools.com: https://www.w3schools.com/nodejs/nodejs_mysql_select.asp