**Module Six Journal**

Justin Starr

Department of STEM

CS 465 – Full Stack Development I

Professor Jimmy Farley

August 11, 2024

**Module Six Journal**

The angular project structure differs from that of the Express HTML customer-facing page, firstly because the customer-facing page uses a Model View Controller (MVC) design pattern to route user requests, retrieve the view for the specified route, and render data from the server database for each specific view. When using the Express application each user action triggers a full page reload. The Angular Single Page Application (SPA), uses components, services, and routes to load a single HTML page on the client-side, and it dynamically updates the content of the page as the user interacts with the application. This initial load only happens once, and the page simply updates as changes are made to any of the data. One key difference is that the Express application may contain several pages, whereas the SPA is what its name says, a single page. In an SPA, components are used to render various parts of the UI and help keep the application’s structure more organized. For example, in the SPA that was completed for our web application, a trip-listing component is used to render all of the trips in the database. Each trip retrieved from the database is displayed in a trip-card component, which displays all of the relevant trip information. Also, each trip-card component has an edit-trip, and delete-trip component that can be used to either edit the details of a trip or delete a trip entirely.

The advantages of using an SPA are that it has higher speed and efficiency than using a multi-page application where pages are reloaded every time a user makes a request. This leads to a better user experience because the user doesn’t have to constantly wait for page reloads, which occurs frequently in traditional or multi-page web applications (Vaishak, 2023). Additionally, using SPAs puts much of the burden of processing onto the client side rather than the server, which can lead to cost savings and enable the use of servers that are less powerful to handle application traffic (Vaishak, 2023). Also, SPAs offer cross-platform compatibility through the use of its JavaScript frameworks, enabling developers to create a single code base that can run on multiple platforms.

The disadvantages of SPAs are that there exist challenges with search engine optimization because the content is loaded dynamically via JavaScript, and search engines do not typically crawl JavaScript content. Placing the processing burden on the client side rather than on the server can cause the initial page load to take longer than most users expect a page to be loaded because initially, all of the application and its resources are loaded (Vaishak, 2023). Because SPAs use client-side routing, browser history, and bookmarks may not work as a user expects, depending on how the SPA is implemented. Lastly, SPAs can be more complex and challenging to implement compared to traditional websites, and if they are not properly secured, SPAs can be vulnerable to a number of security-related issues, such as Cross-Site Scripting (XSS) attacks (Vaishak, 2023). Overall, the additional functionality that is provided by an SPA compared to a simple web application interaction is a better user experience, faster load times, and a reduced server load.

One process of testing the SPA is working with the API to GET and PUT data in the database is by using a tool called Postman. With Postman, a user can set the query type PUT if the user wants to update a document in the database, or they can choose GET to have the database return either all documents in a collection or a specific document. After choosing the appropriate query type, we simply type in the url for the api endpoint and hit send, which bounces the request of the backend API endpoint. If everything is successful, a JSON object with the requested record(s) for the GET query will be returned, or in the case of the PUT query, a JSON object will be returned with the data of the document that is being updated in the database collection. Therefore, we can test by updating a document in the database collection using the PUT query and typing the appropriate URL for the API endpoint using the document ID that we want to update, and then use the GET query to retrieve a list of all documents showing that the updated document was updated successfully. While working with Postman, I did not experience any issues, however, I can see how it might be difficult to work with at first, especially if a user doesn’t understand how to use the queries correctly or how to interact with the different API endpoints.

I currently do not have any further questions that are not already being addressed to help with SPA in future builds.

**References**

Vaishak (2023, November 1). *The Pros and Cons of Single-Page Applications*

*(SPAs)*. Medium. <https://medium.com/@VAISHAK_CP/the-pros-and-cons-of-single-page-applications-spas-06d8a662a149>