Student Name: Dang Khoa Le

ID: 103844421

Tutorial time: Wednesday 9.30 AM

**Draft Report for 10.2HD topic - Tutorial**

**A. Tutorial: Creating a Single Page Application (SPA) with Vue.js**

**Overview:** This tutorial aims to guide through the process of creating a Single Page Application (SPA) using Vue.js from Week 9 Lecture material. SPAs provide a seamless user experience similar to desktop applications by loading all necessary code, HTML, CSS, and JavaScript on the initial page load and dynamically loading additional resources as needed. We will cover the basics of SPA development, including routing, menu creation, login functionality, pagination, and CRUD operations (Create, Read, Update, Delete) using Vue.js.

**Prerequisites:**

* Basic understanding of HTML, CSS, and JavaScript.
* Familiarity with Vue.js framework.

**Topics Covered:**

1. Introduction to Single Page Application (SPA)
2. Single Page Application vs. Multiple Page Application
3. Advantages and disadvantages of SPA
4. Philosophy of Vue.js, Angular, and React
5. Why choose Vue.js for SPA development?
6. Tutorial Walkthrough: Building a SPA with Vue.js

**Tutorial Contents:**

**1. Introduction to Single Page Application (SPA)**

* Definition: A web application contained within a single web page.
* User experience similar to desktop applications.
* Interaction involves dynamic communication with the web server in the background.

**2. Single Page Application vs. Multiple Page Application**

* Difference in client-server interaction.
* SPA lifecycle: Initial request, server response, AJAX communication for data exchange.
* Comparison of SPA and MPA architecture.

**3. Advantages and Disadvantages of SPA**

* Benefits: Fast loading, better user experience, offline capability.
* Drawbacks: Initial load time, SEO challenges, security considerations.

**4. Philosophy of Vue.js, Angular, and React**

* Angular: Feature-rich, all-in-one platform.
* React: Minimalistic, focused on UI-building.
* Vue.js: Between Angular and React, focused on core features with community-driven extensions.

**5. Why choose Vue.js for SPA development?**

* Easy to learn and use.
* Small and fast.
* Reusable components and comprehensive documentation.
* Vue CLI for streamlined project setup.

**6. Tutorial Walkthrough: Building a SPA with Vue.js** This section will delve into the provided code snippets and explain their functionality:

**Using Router for Menu in SPA:**

* Creating VueRouter instance with routes for login, logout, and dashboard.
* Defining routes for different components using **path**, **component**, and **name** properties.
* Implementing menu navigation using Vue Router's **router-link** component.

**Login Component:**

* Creating a login form with input fields for username and password.
* Validating user input using Vue's built-in form validation.
* Handling form submission with a **login** method that sends a POST request to the server for authentication.
* Managing server response to authenticate the user and redirect to the dashboard upon successful login.

**Using Pagination in SPA:**

* Implementing pagination functionality using Vue.js and a pagination library.
* Configuring the pagination component with properties like **page-count**, **page-range**, and **click-handler** to control pagination behavior.
* Defining methods and computed properties to handle data slicing and calculate the total number of pages.

**Data Insert, Update, and Delete:**

* Creating forms for inserting and updating data with input fields for name, age, and other attributes.
* Handling form submission for inserting and updating data by sending POST and PUT requests to the server.
* Implementing data deletion functionality with a delete button and sending a DELETE request to the server to remove the selected data entry.

**Conclusion:** By following this tutorial, you will have the knowledge and skills to develop a robust Single Page Application (SPA) using Vue.js. Empower to create interactive and dynamic web applications with features like routing, form validation, pagination, and CRUD operations, providing users with a seamless and efficient browsing experience.

**B. Application: Building a Social Media Platform with SPA**

**Introduction:** The concepts and techniques covered in this tutorial can be directly applied to the development of a social media platform using a Single Page Application (SPA) design by Vue.js. In this section, we'll explore how SPA architecture can enhance the user experience of a social media platform and discuss the implementation of CRUD operations for managing user posts.

**Enhanced User Experience:** By adopting SPA architecture, a social media platform can provide users with a seamless and interactive browsing experience similar to that of native mobile apps. With SPA, users can navigate between different sections of the platform (e.g., feed, profile, messages) without experiencing page reloads, leading to faster load times and smoother transitions. This fluidity is crucial for engaging users and encouraging them to spend more time on the platform.

**Features and Functionality:**

1. **User Authentication:** Implementing a login system similar to the one demonstrated in the tutorial allows users to securely access their accounts and personalize their experience on the platform.
2. **Feed and Post Creation:** Users can create, edit, and delete their posts using forms and CRUD operations, as showcased in the tutorial. Additionally, users can browse and interact with posts from other users in their feed.
3. **Pagination:** Incorporating pagination functionality enables users to navigate through large volumes of posts efficiently, improving overall usability and user satisfaction.
4. **Responsive Design:** Ensuring the platform is optimized for different devices and screen sizes enables users to access the social media platform seamlessly from desktops, tablets, and smartphones.

**Data Management and Security:**

1. **Database Management:** Utilizing a robust database management system (DBMS) such as MySQL or MongoDB allows efficient storage and retrieval of user data, posts, and interactions.
2. **Data Validation:** Implementing client-side and server-side validation mechanisms helps maintain data integrity and prevent security vulnerabilities such as SQL injection and cross-site scripting (XSS) attacks.
3. **User Privacy and Security:** Implementing authentication, authorization, and encryption measures ensures the protection of user data and privacy, safeguarding against unauthorized access and data breaches.

**Conclusion:** In conclusion, leveraging SPA architecture for the development of a social media platform offers numerous benefits in terms of user experience, functionality, and security. By following the principles outlined in this tutorial and customizing them to suit the specific requirements of the social media platform, developers can create a compelling and engaging platform that attracts and retains users, fosters community engagement, and facilitates seamless content sharing and interaction.

Top of Form

Bottom of Form

**Draft Notice:** This is the draft report for 10.2 task, by which I will utilise the Week 9 lecture material to extend furthermore by coding a social-media platform using SPA design and Vue.js.