Name : Danendra Farrel Adriansyah
NIM   : 23523170

Answer the following questions based on web development material. You can use another resource to answer the questions. **Make sure you cite your sources**.

1. Explain the main differences between Web 1.0, Web 2.0, and Web 3.0.
   Answer:

   Web 1.0, Web 2.0, and Web 3.0 represent distinct phases in the evolution of the internet, each defined by increasing levels of user interaction and technological sophistication. Web 1.0 was characterized by static, read-only content where users could only consume information without interaction. Websites were simple, built with basic HTML and CSS, with no dynamic features or databases, and there was minimal user participation. In contrast, Web 2.0 introduced dynamic content and two-way interaction, allowing users to create, share, and collaborate on content. Social media, blogging, and video platforms like Facebook, YouTube, and Wikipedia are key examples of this era, which also embraced open-source software and web applications, transforming the internet into a participatory space.

   Web 3.0, often referred to as the Semantic Web, is the current evolution, focusing on a personalized, machine-driven experience powered by Artificial Intelligence, Blockchain, and the Internet of Things (IoT). It moves beyond user-generated content to intelligent, decentralized applications that provide personalized services. Data is better understood by machines, enabling context-aware search and interaction. In this phase, users gain more control over their data and interactions through decentralized systems, offering a more secure, transparent, and relevant web experience. Each phase has built on its predecessor, with Web 3.0 promising a future of enhanced personalization, security, and user ownership.

   Reference: https://medium.com/coinmonks/the-evolution-of-the-internet-web-1-0-web-2-0-and-web-3-0-72b090ba2cbd

2. Describe the three core components of web architecture and their roles.
   Answer:
   - Client: The client component is usually a web browser or application that is responsible for requesting resources from the server. The client functions to process the received data and display it to the user. In addition, the client can handle some processing and user interaction, allowing for an interactive user experience. Common examples of clients are Google Chrome and Firefox, which are used to browse various web pages on the internet.
   - Server: A server is a component that stores, processes, and serves content to clients. When a client sends a request, the server handles the request, processes it, and sends back an appropriate response, such as an HTML page, image, or other data. Servers can also host databases and run server-side scripts to provide additional functionality. Some examples of widely used servers are Apache HTTP Server and Nginx, which provide the infrastructure for serving websites and applications.
   - Database: A database serves as a place to store data used by an application. This component stores various information, including user data, content, configurations, and all other important information required for the application to function properly. Databases allow efficient data retrieval and manipulation, so that applications can operate smoothly. Some examples of popular

databases are MySQL and MongoDB, which are often used to store data in modern web applications.

Reference:https://developer.mozilla.org/enUS/docs/Learn/Getting_started_with_the_web/How_the_Web_works

3.   How does decentralization impact the future of Web 3.0, and what advantages does it offer compared to previous versions of the web?
Answer:

Decentralization in Web 3.0 significantly impacts the future of the web by giving users more control over their data, reducing reliance on centralized platforms, and promoting transparency. It enhances security, as data is distributed across networks rather than stored in single locations. Compared to previous versions of the web, Web 3.0 allows for censorship resistance, user privacy, and trustless peer-to-peer interactions, offering a more open and democratic digital environment.

Reference : https://www.ramotion.com/blog/what-is-web-3-0/

4.   Consider a social media application. What components would you need to create this application from a web architecture perspective (front-end, back-end, database)? Explain how they interact.
Answer:

First, the front end creates a display used by the user or UI, then the backend manages the server logic part such as connecting the UI to the database using the API. Then the data that is taken or stored goes into the database. for example, the Instagram UI on the home display, the Backend is used to store data that is inputted during registration and can create posts that go into the database and are called to be displayed on the home page. This system is commonly called MVC in the programming world.

5.   Analyze a popular e-commerce website and identify which part of the web architecture handles user requests and manages the shopping cart. What technologies could be involved in this process?
Answer:

An example is Tokopedia, where various parts of the web architecture handle user requests and manage the shopping cart. The front-end, built with technologies like React or Vue.js and the like, allows users to interact with the website by adding items to their cart and updating the quantity. These requests are sent to the back-end via APIs like REST, where the server processes the request, checks the stock availability, and calculates the total. Technologies like Node.js or Spring Boot are commonly used for this back-end process. Once processed, the data is stored in a database like MySQL or MongoDB, where details about the user's cart are stored. Caching systems like Redis can be used to speed up data retrieval. This entire system ensures a smooth interaction between the user and the shopping cart in real-time, making the user experience smooth and efficient.

Reference: https://dev.to/nilmadhabmondal/let-s-develop-shopping-cart-for-ecommerce-app-5f0h