1. Why we use "reduxjs/toolkit", "react-redux", "redux" in web development?

|  |
| --- |
| **Redux:**  **Purpose:** Redux is a predictable state container for JavaScript applications. It provides a centralized store to **manage the state** of your **application** in a **predictable** and **maintainable** way.  **When to Use:**   * In applications with complex state management requirements. * When multiple components need to share and synchronize state. * For applications with deeply nested component hierarchies where passing props becomes cumbersome. |
| **React-Redux:**  **Purpose:** React-Redux is the official React bindings for Redux. It provides a **set of APIs** that allow React components to **interact with the Redux store**.  **When to Use:**   * In React applications where Redux is used for state management. * When you need to connect React components to the Redux store to access or modify the state. |
| **@reduxjs/toolkit:**  **Purpose:** The Redux Toolkit is the official opinionated **set of utility functions for Redux**. It includes several **utility functions** and **abstractions** to **simplify common Redux patterns** and reduce boilerplate code.  **When to Use:**   * To streamline the process of writing Redux logic. * When you want to follow best practices recommended by the Redux team. * When you want to take advantage of features like the **createSlice** function for reducer logic and **configureStore** for store setup. |

1. why we use "axios", "react-router-dom", "moment" in web development?

|  |
| --- |
| **Axios:**  **Purpose:** Axios is a popular JavaScript library used to **make HTTP requests** from web browsers. It supports promises and can be used for both the browser and Node.js.  **When to Use:**   * Making asynchronous requests to a server to fetch or send data. * Interacting with RESTful APIs. * Handling AJAX operations in React applications. |
| **React Router DOM:**  **Purpose:** React Router DOM is the **standard routing library** for React applications. It enables **navigation and rendering of components** based on the URL.  **When to Use:**   * Implementing client-side navigation in single-page applications (SPAs). * Creating dynamic and interactive user interfaces where components change based on the URL. * Managing different views or pages within a React application. |
| **Moment:**  **Purpose:** Moment.js is a JavaScript library for **parsing, validating, manipulating**, and **formatting dates**. It simplifies working with dates and times in JavaScript applications.  **When to Use:**   * Parsing, formatting, or manipulating date and time objects. * Displaying or processing dates in a user-friendly manner. * Dealing with complex date-related logic. |

1. Lists of HTTP requests and their use case.

|  |
| --- |
| **GET:**   * **Use Case:** Used to request data from a specified resource. * **Example:** Fetching a webpage, retrieving information from a server. |
| **POST:**   * **Use Case:** Used to submit data to be processed to a specified resource. * **Example:** Submitting a form, creating a new record on the server. |
| **PUT:**   * **Use Case:** Used to update a resource or create a new resource if it doesn't exist. * **Example:** Updating user details, creating or replacing a resource. |
| **PATCH:**   * **Use Case:** Used to apply partial modifications to a resource. * **Example:** Updating only specific fields of an existing record. |
| **DELETE:**   * **Use Case:** Used to request the removal of a resource. * **Example:** Deleting a user account, removing a file from a server. |

1. Use case of "bcryptjs", "cloudinary", "dotenv", "express", "jsonwebtoken", "mongodb", "mongoose", “nodemon” in web development?

|  |
| --- |
| **bcryptjs:**   * **Purpose:** A library for hashing passwords. * **Use Case:** Used for securely storing user passwords by hashing and salting them before storing in a database. |
| **cloudinary:**   * **Purpose:** Cloudinary is a cloud service that offers image and video management. * **Use Case:** Used for uploading, storing, and manipulating images and videos in the cloud. Commonly integrated with applications that involve media storage. |
| **dotenv:**   * **Purpose:** Loads environment variables from a .env file. * **Use Case:** Used for configuring and managing environment-specific variables (e.g., API keys, database connection strings) in development and production environments. |
| **express:**   * **Purpose:** A web application framework for Node.js. * **Use Case:** Used for building server-side applications and APIs. Express simplifies the process of handling HTTP requests, routing, and middleware integration. |
| **jsonwebtoken:**   * **Purpose:** Creates and verifies JSON Web Tokens (JWTs). * **Use Case:** Used for authentication and authorization. JWTs are often generated upon successful user login and sent to the client, which includes claims about the user. |
| **mongodb:**   * **Purpose:** Official MongoDB Node.js driver. * **Use Case:** Used for connecting to and interacting with MongoDB databases from a Node.js application. Enables CRUD operations and database management. |
| **mongoose:**   * **Purpose:** An ODM (Object Data Modeling) library for MongoDB. * **Use Case:** Provides a higher-level, schema-based abstraction over MongoDB. Used for defining data models, schema validation, and simplifying database interactions. |
| **nodemon:**   * **Purpose:** A utility that monitors for changes in files and automatically restarts the Node.js application. * **Use Case:** Facilitates a more efficient development workflow by automatically restarting the server upon file changes, eliminating the need to manually restart the server. |