

# Application Development 2 - Level 2 viva consolidated Q&A

-Rishav Jain

## Q: What is javascript?

**Ans:** Javascript is a programming language for the web. It is one of the core technologies of the World Wide Web, alongside HTML and CSS. Javascript can update and change both HTML and CSS. It can calculate, manipulate and validate data.

JavaScript is a widely-used programming language that is primarily used for web development. It enables developers to create interactive and dynamic elements within web pages.

## Q: What is an API?

**Ans:** An API, or Application Programming Interface, is a set of rules and protocols that allows different software applications to communicate and interact with each other. APIs define how different software components should interact, making it possible for developers to use pre-built functionalities and services without needing to understand the internal implementation.

## Q: can we use javascript outside of browser?

**Ans:** Yes, we can use JavaScript in any command line interface using NodeJS.

## Q: What is the difference between Client side and server side?

**Ans:** Client-side refers to the code and processes that run in the user's web browser. This is typically where the user interface (UI) is presented, and the user interacts with the application. In client-side development, technologies like HTML, CSS, and JavaScript are used.

Server-side refers to the code and processes that run on the server hosting the web application. Server-side processing handles tasks such as managing databases, processing user input, authentication, and generating dynamic content.

## Q: Which function is used to make HTTP request to backend or any server?

**Ans:** We can use functions like `fetch()` to make HTTP requests to a backend server or any server that supports HTTP. The `fetch()` function is a modern and powerful way to handle asynchronous network requests. It requests a Promise that resolves to the Response to the request. But since `fetch()` is a modern API, for compatibility with older browsers, we might need to consider other options like `XMLHttpRequest` or third party libraries like Axios.

## Q: Can we perform validation on client side?

**Ans:** Yes, client-side validation is possible using JavaScript. Client-side validation involves using JavaScript to validate user input directly in the user's browser before the data is submitted to the server. This validation helps ensure that the data entered by the user meets specific criteria and is in the correct format before being sent to the server for further processing.

## Q: Is client side validation enough? If not, what should we do to resolve this?

**Ans:** Client-side validation offers immediate feedback to users but can be bypassed and lacks security. Server-side validation, input sanitization, and security practices are essential to ensure data integrity and protect against vulnerabilities and attacks. Combining both client-side and server-side approaches is necessary for a reliable and secure application.

## Q: If we are trying to access different origins then which error do we get?

**Ans:** The specific error you'll encounter in this scenario is called the "Cross-Origin Resource Sharing (CORS) error." CORS is a mechanism that allows servers to specify which origins are permitted to access their resources. If the server doesn't include the necessary CORS headers in its responses, the browser will prevent the web page from making requests to that server.

## Q: Types of storage in browser side and the difference between them?

**Ans:** there are several types of storage options available for storing data on the client side. Each storage option has its own characteristics and use cases. The main types of storage in browsers are:

- **Cookies:** Storing small amounts of data that need to be sent to the server with each request. Often used for session management, tracking, and storing user preferences.
- **Session Storage:** Storing data that should be accessible only during the user's session on a single browser tab. Data is cleared when the tab is closed.

- **Local Storage:** Storing persistent data that should be available across multiple sessions and tabs. Data remains until explicitly cleared by the user or the application.
- **IndexedDB:** Storing larger amounts of structured data, such as complex objects or files, for offline use or caching.
- **Web Storage:** An older term used interchangeably with "local storage." In modern discussions, "local storage" is more commonly used.
- **Browser Cache:** Browser caching involves storing copies of resources (such as images, stylesheets, scripts, and even entire web pages) locally on the user's device for a specified period. This reduces the need to repeatedly download resources from the server, resulting in faster page loading times.

## Q: what is DOM?

**Ans:** DOM stands for "Document Object Model." It is a programming interface and representation of a structured document, typically an HTML or XML document, that allows scripts (like JavaScript) to access and manipulate the content, structure, and style of the document.

## Q: What is the difference between Cli and Cdn?

**Ans:** A CLI is a text-based interface used to interact with a computer system or application through commands typed in the terminal or command prompt. It's a tool that allows developers to perform various tasks, such as creating projects, running scripts, managing dependencies, and more, without needing to use a graphical user interface (GUI).

A CDN is a network of distributed servers that work together to deliver web content (such as images, scripts, stylesheets, videos, and more) to users based on their geographic location. - The main purpose of a CDN is to improve the performance, speed, and availability of web content by reducing the distance between the user and the server delivering the content.

## Q: What is the difference between single page and PWA?

**Ans:** A SPA is a web application that loads a single HTML page and dynamically updates its content as the user interacts with the application. SPAs provide a smoother and more responsive user experience, as they can mimic native application behavior and transitions.

A PWA is a web application that incorporates modern web technologies and best practices to provide an app-like experience to users. PWAs can be installed on a user's home screen and launched from there, similar to native apps.

## Q: What is AJAX?

**Ans:** AJAX stands for "Asynchronous JavaScript and XML." It's a set of web development techniques used to create more dynamic and responsive web applications by allowing data to be retrieved from a server asynchronously without requiring a full page reload. While the term includes "XML," modern applications often use JSON (JavaScript Object Notation) instead of XML for data interchange.

## **Q: What is Redis and How does it work?**

**Ans:** Redis (Remote Dictionary Server) is an open-source, in-memory data store that serves as a highly efficient and versatile solution for caching, data storage, and real-time application needs. It's often referred to as a "data structure server" because it provides various data structures that can be used to store and manipulate different types of data. Redis is designed for high-speed read and write operations, making it well-suited for applications that require fast data access.

## **Q: What is NoSQL?**

**Ans:** NoSQL (short for "not only SQL") is a type of database system designed to handle large volumes of unstructured or semi-structured data. Unlike traditional relational databases (SQL databases), which use a tabular schema with predefined schemas and rigid relationships between tables, NoSQL databases offer more flexible and scalable approaches to data storage and retrieval.

## **Q: What is GraphQL?**

**Ans:** GraphQL is an open-source query language and runtime for APIs (Application Programming Interfaces) that was developed by Facebook. It provides a more efficient, flexible, and powerful way to interact with APIs compared to traditional REST (Representational State Transfer) APIs. GraphQL allows clients to request exactly the data they need, making it possible to avoid over-fetching or under-fetching of data.

## **Q: What is Vuejs and Nodejs?**

**Ans:** Vue.js is a progressive front-end JavaScript framework used for building user interfaces and web applications. Vue.js is designed to be incrementally adoptable, meaning you can start using it in small parts of your application and gradually integrate it more deeply. Node.js is a runtime environment that allows you to run JavaScript code on the server side. It uses an event-driven, non-blocking I/O model, making it efficient for building scalable and real-time applications.

## **Q: Why do we have different frameworks?**

**Ans:**

- Different projects have different requirements. Some frameworks are optimized for specific use cases like building user interfaces (e.g., React, Vue.js), while others are designed for back-end development.
- The field of technology is ever-evolving. New challenges and opportunities arise, leading to the creation of new frameworks that are better suited to modern requirements.
- Frameworks often come with their own ecosystems of libraries, tools, and extensions. Depending on the specific needs of a project, one framework's ecosystem might be more suitable than another's.

## Q: Why `async` is used with `await`?

**Ans:** In JavaScript, `async` and `await` are used in tandem to work with asynchronous code in a more synchronous-like manner. They are part of the `async/await` feature introduced in ECMAScript 2017 (ES8) and are particularly useful for handling promises, which are used for asynchronous operations.

Here's how they work:

### 1. `async` Function:

- When you declare a function as `async`, it automatically returns a promise. Inside an `async` function, you can use the `await` keyword to pause the execution of the function until a promise is resolved or rejected.
- The `async` function provides a more readable and structured way to work with asynchronous code compared to chaining `.then()` callbacks.

### 2. `await` Keyword:

- The `await` keyword is used within an `async` function to pause the execution of that function until a promise is resolved. It allows you to write asynchronous code that looks and behaves more like synchronous code.
- When `await` is used with a promise, it waits for the promise to settle (either resolve or reject) before continuing execution.

## Q: In what type of environment are we using console statements?

**Ans:** Console statements are used primarily in development environments to help developers debug and monitor their code during the development process.

## Q: What is sandbox?

**Ans:** A sandbox is an isolated and controlled environment where developers can test new code, software, or changes without affecting the production environment. It's a safe space to

experiment, identify bugs, and ensure that changes won't negatively impact the actual system.

## **Q: What is mailhog? What kind of application is it?**

**Ans:** MailHog is an email testing tool and debugging application primarily used by developers to simulate and capture email interactions in a local development environment. It helps developers test how their applications send and receive emails without actually sending emails to real addresses.

## **Q: Full form of SMTP?**

**Ans:** SMTP stands for Simple Mail Transfer Protocol. It is a widely used protocol for sending and routing email messages between servers.

## **Q: What is celery?**

**Ans:** Celery is an open-source distributed task queue system for handling asynchronous tasks in a software application. It is often used in Python applications to offload time-consuming or resource-intensive tasks from the main application thread, allowing the application to remain responsive and efficient. Celery is particularly useful for tasks that can be executed in the background or tasks that might take a significant amount of time to complete.

## **Q: Why do we use hashing and not encryption?**

**Ans:** Hashing is a one-way process that converts input data (such as text, files, or passwords) into a fixed-size string of characters, typically of a shorter length. Hashing is designed for data integrity verification and quick data retrieval, not for data confidentiality.

Encryption is a reversible process that transforms data into a different format in such a way that it can only be transformed back using a specific decryption key. Encryption is used to protect sensitive information and ensure data confidentiality.

## **Q: Why are we using mailhog and not gmail or hotmail?**

**Ans:** MailHog is often used during the development phase of a software application. It provides a local testing environment for sending and capturing emails without actually sending them to real email addresses.

## **Q: What are webhooks?**

**Ans:** Webhooks are a way for one application to provide real-time data to another application or system by sending HTTP POST requests containing information about events that have

occurred. They are a way to facilitate communication between different software systems over the internet.

## Q: What is caching?

**Ans:** Caching is a technique used in computer science and information technology to store and reuse frequently accessed or computed data in a way that improves the efficiency and performance of applications. Instead of recalculating or retrieving data from the original source every time it's needed, cached data is stored in a faster-access location for quicker retrieval.

## Q: How to send data from parent to child component and vice versa?

**Ans:** In the context of component-based frameworks like React (for web development) or Vue.js, you can pass data between parent and child components using props (properties) and events (emitting events).

## Q: What is WSGI?

**Ans:** WSGI stands for "Web Server Gateway Interface." It is a specification in the Python programming language that defines a standard interface between web servers and web applications or frameworks. WSGI provides a common way for web servers to communicate with Python web applications, allowing for interoperability and ease of deployment.

## Q: Why we use VueJs over AJAX?

**Ans:** Vue.js and AJAX serve different aspects of web development. Vue.js is a powerful framework for building interactive user interfaces, while AJAX is a technique for making asynchronous data requests and updating parts of a page. In many cases, these two technologies can work together to create engaging and responsive web applications. It's important to note that Vue.js and AJAX are not mutually exclusive; in fact, they can complement each other in building dynamic and interactive web applications.

## Q: What are the different components of Celery?

**Ans:** Celery is a Python library used for managing asynchronous and distributed task processing. It comprises several key components that work together:

1. **Task:** A unit of work executed asynchronously, often a Python function.
2. **Broker:** A message queue system that stores tasks and communicates between producers (your app) and workers.
3. **Worker:** Processes that consume tasks from the broker and perform the actual work.

4. **Result Backend:** Stores task results, allowing retrieval of task outcomes.
5. **Beat:** A scheduler that sends tasks to the broker based on a schedule.
6. **Configuration:** Allows you to set various settings like broker connection, task routing, and more.
7. **Canvas:** Module for creating complex task workflows and dependencies.
8. **Monitoring and Management:** Tools like the Flower interface for tracking worker activity.

## Q: What are the different Vue Lifecycle hooks?

**Ans:** Vue.js has a series of lifecycle hooks that allow you to execute code at specific moments during the lifecycle of a Vue component. These hooks are useful for performing actions such as initialization, data manipulation, and cleanup.

1. **beforeCreate:**
  - Function: Executed before the instance is created.
  - Use: Initialization, setting up data or methods before data observation and event/observer setup.
2. **created:**
  - Function: Executed after the instance is created.
  - Use: Data observation, computed properties, methods, and watch/observe properties are set up. The instance is not yet mounted.
3. **beforeMount:**
  - Function: Executed before the component is inserted into the DOM.
  - Use: DOM-related operations can be performed here. The component is about to be rendered.
4. **mounted:**
  - Function: Executed after the component is inserted into the DOM.
  - Use: DOM elements are accessible here. You can perform tasks that require access to the DOM, such as fetching data or setting up third-party libraries.
5. **beforeUpdate:**
  - Function: Executed when data changes and before the DOM is updated.
  - Use: Reacting to data changes before the re-rendering of the component.
6. **updated:**
  - Function: Executed after data changes and the DOM is updated.
  - Use: Accessing the updated DOM after a data change.

## Q: How are redis and celery different?



**Ans:** Redis is an in-memory data store known for caching and real-time analytics. It supports data structures and pub/sub messaging. Celery is a distributed task queue system, focusing on task scheduling, distribution, and execution across workers. It integrates with message brokers for reliable communication and handles complex task workflows. While Redis can serve lightweight message queuing, Celery is specialized for managing asynchronous distributed tasks.

## **Q: What is a Primary key and Foreign Key?**

**Ans:** A primary key is a unique identifier within a database table that uniquely identifies each record. It ensures the uniqueness and integrity of the data in the table, serving as a reference point for other tables.

A foreign key is a field in a database table that establishes a link to the primary key of another table. It creates a relationship between the two tables, allowing data in one table to reference data in another, enabling data consistency and integrity.