|  |
| --- |
| **APPLICATION FOR PROJECT MEMBER**  **WEBOPS & BLOCKCHAIN CLUB** |
| **Name: Anantu S Pai** |
| **Roll No.: CE22B039** |
| **Smail: ce22b039@smail.iitm.ac.in** |
| **Phone No.: 7067691076** |

**TECHNICAL QUESTIONNAIRE**

1. **What is asynchronous JavaScript?**

* Asynchronous JavaScript refers to the execution of JavaScript code that doesn’t necessarily run in a linear, sequential manner from top to bottom.
* It allows multiple operations to be performed simultaneously without blocking the main thread of evolution.
* It is crucial for tasks like handling user interactions, making network requests, and working with timers

**Some Uses:-**

* **Promises-** It allows us to work in a more readable & structured way. We can initiate a task & decide what to do with the results by chaining e.g.: - **‘.then’** is used while calling API.
* **Async/Await: -** It is a more concise way to work with asynchronous code. It allows to write asynchronous code that looks more like synchronous code, making it easier to read & maintain.

**How is it different from synchronous JavaScript?**

* Unlike asynchronous JavaScript, code is executed in a sequential, blocking manner. Each line of code is executed one after the other, and the program waits for each operation to complete before moving on to the next one.
* If a time-consuming operation occurs, such as a network request or a complex calculation, it can block the entire program, making the user interface unresponsive. This is known as "blocking."
* Asynchronous code can be more complex to write and reason about because it involves handling the timing and coordination of multiple asynchronous operations.

1. **What is an API? Explain in brief how it works and how you will create an API with technical details.**

* API (Application Programming Interface) is a set of rules and protocols that allows different services, software, and applications to communicate with each other.
* It defines the methods and data formats that applications can use to request and exchange information.
* APIs are used for various purposes, such as retrieving data from a server, sending data to a server, or interacting with external services.
* It works in 3 steps: -
  + Request
  + Process
  + Response

**Creating an API: -**

* Endpoints are decided: If we are for a basic e-commerce website, we have endpoints for Creating, Reading, Updating& Deleting the different kinds of products.
* Data format is chosen such as JSON.
* Appropriate HTTP methods (GET, POST, PUT, DELETE) are chosen for each endpoint to indicate the type of operation being performed.
* API is tested using tools like POSTMAN

1. **What is the difference between a library and a framework? Is Vue a library or a framework?**

**Library:-**

* Libraries are collections of functions, classes, or modules that can be used to perform specific tasks or operations in your application. They provide well-defined, isolated pieces of functionality that you can call upon as needed.
* Allow you to pick and choose the components you need and integrate them into your codebase. You have more freedom in how you structure your application.
* jQuery (a JavaScript library for DOM manipulation), Lodash (a JavaScript utility library), and OpenCV (a computer vision library) are examples of libraries.

**Framework:-**

* Provide a more structured and opinionated environment for building applications, often guiding the entire application's design and flow.
* They dictate when and how your code gets executed. You often provide callbacks or implement specific methods that the framework calls at the appropriate times.
* Frameworks have a more rigid structure. You need to follow the framework's conventions, which can speed up development by providing best practices, but it may limit your freedom in certain ways.
* Ruby on Rails (a web application framework for Ruby), Angular (a front-end framework for JavaScript), and Django (a web application framework for Python) are examples of frameworks.

Vue.js is often referred to as a "progressive framework." This is because Vue provides a comprehensive framework for building user interfaces, but it can also be used as a lightweight library for specific parts of your application. It allows you to incrementally adopt its features, making it suitable for a wide range of use cases.

Vue provides a complete solution for building web applications, including reactive data binding, a component system, a router, and state management (Vuex). However, you can also use Vue in a more library-like fashion, where you include it in an existing project to enhance specific parts of your application, such as a single page or a widget.

So, Vue.js is a versatile tool that can be used as both a library and a framework, depending on how you choose to incorporate it into your project. Its flexibility and ease of use make it a popular choice for many web developers.

1. **List some databases that are currently being used. Explain the advantages and disadvantages of each.**

There are 2 types of databases that are currently being used: -

**Relational Database Management System (RDBMS), Postgresql :**

*Advantages:*

* *Structured Query Language (SQL):* SQL is a powerful query language that allows for complex data retrieval and manipulation.
* *Data Integrity*: RDBMS systems enforce data integrity constraints, ensuring that data remains consistent.

*Disadvantages:*

* *Limited Scalability:* Scaling RDBMS horizontally can be challenging, especially for read-heavy workloads.
* *Fixed Schema:* Schemas in RDBMS are rigid and not easily adaptable to changing data structures.

**NoSQL Databases (e.g., MongoDB, Cassandra):**

*Advantages:*

* *Scalability:* NoSQL databases are often designed for horizontal scalability, making them suitable for large datasets and high-throughput workloads.
* *Flexible Schema:* NoSQL databases support flexible, schema-less data structures, allowing for easier adaptation to changing requirements.

*Disadvantages:*

* NoSQL databases may prioritize performance and scalability over strict data consistency.
* *Limited Querying Capabilities:* Querying can be more limited compared to SQL databases, making complex queries challenging.

There are also other types of databases

* Columnar Databases( Apache Cassandra, Amazon Redshift)
* Graph Databases (e.g., Neo4j, Amazon Neptune)
* In-Memory Databases (e.g., Redis, Memcached)
* Time-Series Databases (e.g., InfluxDB, Prometheus)

1. **What are the advantages and disadvantages of using custom styles over existing styling frameworks? Which one do you prefer among the two?**

***Advantages: -***

* ***Full Control:*** When creating custom styles, we have complete control over the look and feel of our website or application which can be designed precisely to match our vision without being constrained by the limitations of a framework.
* ***Optimized for Our Needs:*** Custom styles allow us to write CSS that's specifically tailored to our project's requirements, potentially resulting in smaller and more efficient stylesheets.
* ***Educational Value:*** *Creating custom styles can be a great learning experience, helping us understand how CSS works at a deep level.*
* ***Performance:*** Custom styles can be optimized for performance, ensuring that our website or application loads quickly and efficiently.

***Disadvantages: -***

* ***Time-Consuming:***Creating custom styles from scratch can be time-consuming, especially for complex designs. This may slow down the development process.
* ***Maintenance:***Maintaining custom styles as the project grows and evolves can be challenging, as we will need to keep track of various CSS rules and ensure they remain consistent and bug-free*.*
* **Inefficiency:** Building custom styles means recreating common design patterns and components that existing frameworks might provide out-of-the-box, which can be inefficient.

According to me, the choice of the framework will depend upon my experience in CSS, the nature and size of the project, and the available time. We can even consider using a mixture of both techniques.

1. **Which project do you want to join? Why do you want to join this project specifically? What is your approach for solving this problem statement? What challenges will we be facing during the entire project and how would you solve them? Explain everything in detail.**

I would like to join the “Denzee- Automated Hospital Dashboard” Project as being a part of this project I would get an opportunity to improve my skills in web development and communication systems. Moreover, considering that we have to come up with a solution from scratch, it would be a great opportunity to test my creativity, organizational, and problem-solving skills.

My approach for solving this problem is:

* Identifying the nature of the message that is to be sent or received.
* Design a platform to integrate the email system.
* Ensure sufficient security and authentication as the data being communicated could be sensitive in nature, such as the health history of the patient.
* Trying to automate the various processes.

Potential challenges that we might face could be:

* **Email Integration:** Considering that there are many email providers with different protocols and security measures, integrating can become complex. This can be solved using well-established APIs.
* **Security:** The data in question is quite sensitive in nature and hence needs to be protected. This needs to be done through encryption and access control.
* **Data Storage:** As the platform grows, so does the need for handling user and hospital data. This could require the use of cloud-based storage systems.

**PROJECT SPECIFIC QUESTIONNAIRE**

**Explain OAuth2.0 protocol for authentication. What is JWT? Do you think they are the same thing? How do you think ‘Authenticate with Google’ works on most websites?**

***OAuth 2.0:***

* OAuth 2.0 serves as an open standard protocol for authorization, enabling third-party applications to access a user's resources on a server without the user sharing their credentials directly.
* It focuses on providing a secure and controlled method for granting access to these resources. The typical entities in OAuth 2.0 are:
  + Resource Owner: The user who possesses the resource, like their social media profile.
  + Client: The application or service seeking access to the user's resources.
  + Authorization Server: This server authenticates the user and secures their consent for the client to access resources.
  + Resource Server: The server hosting the user's resources and responding to client requests post-authorization.
* OAuth 2.0 defines various flows to cater to different use cases, each outlining how the client obtains access tokens, which are used to make resource requests on behalf of the user.

***JWT (JSON Web Tokens):***

* JWT serves as a concise and self-contained way to represent data between two parties in the form of a JSON object.
* It's used to transmit information securely and verifiably.
* A JWT comprises three parts:
  + a header (metadata),
  + a payload (claims), and a
  + signature (for authenticity and integrity validation).

***Authenticate with Google:***

The "Authenticate with Google" option on a website usually involves OAuth 2.0 in combination with JWT. The typical process is as follows:

* Clicking "Authenticate with Google" redirects to Google's authentication server.
* You log in to your Google account (if not already logged in) and grant permission to the website to access specific data or perform actions on your behalf.
* Google's authorization server generates an access token and, optionally, a JWT.
* The website receives the access token, allowing it to make authorized requests to Google APIs on your behalf.
* If a JWT is used, it might serve to verify your identity on the website, ensuring user authentication.

**How do you think we can track the emails sent by us? ‘Tracking’ refers to activities performed by recipients like receiving the mail, opening the mail, and marking it as spam.**

Recipient interaction with the mail can be tracked by using unique tracking links and/or tracking pixels.

* *Unique Tracking Links:*
  + a unique link for each recipient is generated and embedded in the sent mail.
  + Interaction data can be obtained when any clicking event is recorded.
  + It also includes the time of the click, click rate, etc.
* *Tracking pixels:*
  + are tiny, invisible images (usually 1x1 pixel) hosted on a server. These pixels are inserted into the email's HTML code.
  + When the recipient opens the email and the email client loads the images, it sends an HTTP request to the server where the tracking pixel is hosted.
  + The server records information about the request, including the recipient's IP address, user agent (browser or email client), and sometimes geolocation data.