**Documentation**

Backend:

1. Nest.js:

- A progressive Node.js framework for building scalable and efficient server-side applications.

- Leverages TypeScript for strong typing and enhanced developer experience.

- Follows modular design principles and supports a variety of modules for different functionalities.

2. Docker:

- Containerization platform that allows you to package applications and their dependencies into isolated containers.

- Enables consistent deployment across different environments, improving scalability and reproducibility.

3. MongoDB:

- A NoSQL database that stores data in flexible, JSON-like documents.

- Well-suited for handling large amounts of unstructured or semi-structured data.

- Often chosen for its scalability and ease of horizontal scaling.

4. RabbitMQ:

- Message broker that facilitates communication between different parts of a distributed system.

- Implements the Advanced Message Queuing Protocol (AMQP) and supports message queuing, routing, and other messaging patterns.

Frontend:

1. Vue.js:

- A progressive JavaScript framework for building user interfaces.

- Emphasizes simplicity and flexibility, making it easy to integrate into projects and scale as needed.

- Implements a reactive data-binding system for efficient UI updates.

2. Module Federation:

- A feature in Webpack that enables dynamic loading of remote modules at runtime.

- Useful for creating micro-frontends, allowing different parts of the application to be developed and deployed independently.

Overall:

- Integration:

- Docker containers ensure consistent deployment of the entire application stack, including backend and frontend components.

- Nest.js serves as the backend framework, providing a structured and modular approach to building server-side applications.

- Communication:

- RabbitMQ facilitates communication between different microservices or components in the system, ensuring reliable message passing.

- Data Storage:

- MongoDB is chosen as the database for its NoSQL capabilities, suitable for handling various data formats and scaling horizontally as needed.

- Frontend Architecture:

- Vue.js is employed for building the user interface, offering a reactive and component-based structure.

- Module Federation allows for the dynamic loading of frontend modules, supporting a scalable and independent development approach.

