Reliability, scalbility, performance,security, developer’s experience and cost of hosting

Backend 🡪 Which runs on server-side and responsible for handling and managing storage , database and otherresources. Backend is also called data access layer of software.

NodeJS

* Used **Asynchronous** Programming (Non -blocking)
* It gives a callback when its done.
* **Very scalable**

When to use Node.js

* Applications have high **event -driven** and performs **lots of I/O operations**.
* Have **more API calls from backend** itself.

When NOT to use Node.js

* Having lot of heavy algorithms or consumes lot of CPU cycles
* Node.js runs on single thread just like client-side js
* Will be inefficient for CPU intensive jobs.

Java- Springboot

* To make production ready spring-boot applications.

When to use :

* Primary focus is on security
* Java support multithreading out of box and becomes great choice for building complex concurrent web applications.

When NOT to use :

* When need heavy computation, complex and requires lot of expertise.

Laravel

* Open-source PHP framework.
* Follows MVC (Model -View Controller \_architecture
* Friendly features like query builder or ORM .
* One of the competitive frameworks.

When to use Laravel

* Time to market is key.
* Development is fast as compared to other framework.
* Laravel can be hosted on shared hosting thus making it cheapest amongst all
* When time and cost is key , you should go with Laravel.

When NOT to use Laravel

* PHP is not considered much secure as compared as spring and Node.js
* Prevents attaks such as SQL injection ,cross-site sciprtin and adds extra layer of security to it.
* PHP is never recommended for application where security is must.

Python – Django

* Django is fast ,secure and scalable high level python web framework.
* Django encourages rapid and clean application development.

When to use Django?

* Django is based on python is supports powerful machine learning libraries like pyTorch , NumPy
* Computational and statistical capabilities make idle platform for machine learning applications.

When Not to use Django?

* Not suitable for small projects
* Batteries included framework
* It has much boilerplate code which small projects don’t need.
* Consuming unnecessary server processing time and bandwidth.