Laravel

**1. Template Tool**

Laravel has a pre-installed powerful and lightweight template engine, helping developers in making some extraordinary layouts with intensive content seeding.

**2. Authentication and Authorization**

Some of the best Laravel development companies assure secure authentication and authorization. Any web application owner needs to be assured of having authentic and authorized users accessing the secured resources. Implementation of authorization is quite simple as almost everything is configured out-of-the-box.

**3. Inbuilt Libraries**

Laravel is the framework that facilitates you at best by being the only provider having dynamic pre-installed libraries. These libraries include Object Oriented libraries that cannot be found in other PHP frameworks. Besides offering hassle-free implementation, it also contains various features like watchword reset, checking dynamic clients, CSRF (Cross-site Request Forgery) insurance, and encryption. Additionally, Laravel has a helper functionality, helping you encourage and achieve various basic undertakings i.e.firing events,  generating views, sending HTTP responses to the server etc. This is a significant feature that is found exclusively in Laravel.

**4. URL Generations**

Laravel also assist in generating URLs which becomes very helpful for building links in your templates.  All Laravel corridors are effectively laden by the framework which is delineated in the app/Http/routes.php file.

**5. Integration with Mail Services**

Mail service integration is another benefit that Laravel provides. It is used to send notifications to users’ to notify them about the various events that take place.

**6. MVC Architecture Support**

While you google what Laravel is, Wikipedia’s definition states that it follows a Model – View – Controller architecture.

**7. Technical Vulnerabilities Fixture**

Laravel is a secure platform that takes care of the security within its framework. It safeguards web application against the most serious security risks such as; SQL injection, cross-site request forgery, cross-site scripting, etc.

**8. Artisan Tool for Command Line**

Generally, a developer needs to interact with the Laravel framework using a command line that develops and manages the Laravel project environment.

**9. Fine Unit Testing**

In Laravel’s unit testing, every module of your web application is tested before the site goes live so that no part of your website remains broken.

**10. Multi-Lingual App Creator**

Django

**Written in Python**

**Accelerates custom web application development**

Django is one of the most mature web frameworks for Python. Its design rules focus extensively on reducing web application development time.

**Designed as a batteries-included web framework**

Django is one of the web frameworks that adopt the batteries-included approach. While developing a custom web application, Django provides the resources required by developers out of the box. It provides code for common operations like database manipulation, HTML templating, URL routing, session management, and security.

**Supports MVC programming paradigm**

Django, like other modern web frameworks, supports model-view-controller (MVC) design rule.

**Compatible with major operating systems and databases**

Nowadays, users access web applications on various devices and platforms. Django enhances the accessibility of web applications by supporting major operating systems like Windows, Linux and MacOS.

**Provides robust security features**

The built-in security features provided by Django help developers to protect the web applications from a variety of targeted security attacks – cross-site scripting, SQL injection and cross-site request forgery.

**Easy to extend and scale**

Django has been evolving consistently to enable programmers to build better and modern web applications.

Ruby on Rails

**It’s time-efficient.** Ruby on Rails contains many ready-made plugins and modules, which allow developers not to waste time on writing boilerplate code.

**It’s consistent.** Developers follow standardized file storage and programming conventions that keep a project structured and readable. It also saves plenty of time.

**It’s cost-effective.** Ruby on Rails is an open source framework distributed under the MIT license, that means you don’t have to spend money on the framework itself.

**It provides excellent quality and promotes bug-free development.** The Minitest tool built into the Rails core is a comprehensive test suite that provides many useful testing features including expectation syntax, test benchmarking, and mocking.

**It’s scalable.** If you expect to get a lot of users for your application you should make sure that it can cope with all the visitors you’re hoping to attract.

**It’s supported.** There is a majority of blogs and books, meetups and conferences in the Rails world.

**It’s secure.** Some security features are built into the framework and enabled by default. Using Ruby on Rails also means following the Secure Development Lifecycle, which is a complex security assurance process.

Express.js

**1. Scale our application quickly**

The first benefit of using Express.JS for backend development is that you would be able to scale your application quickly.

**2. JavaScript is simple to learn**

As mentioned above that JavaScript is very famous plus easy to learn a language. You would be able to use it for the development purpose on Express.JS.

**3. We can use same language to code Frontend**

Another benefit of using Express.JS is that you would be able to do the code of both frontend and backend with the help of using JavaScript.

**4. Less developer cost to maintain the app**

Not everybody knows that Express.JS is a full-stack JavaScript because of which you would not have to hire different developers for managing the frontend and backend of a web application.

**5. Supported by Google v8 engine**

Express.JS is supported with the Google V8 engine with the help of which you would be able to get higher performance without any lag or error in the processing.

**6. Community support**

If you ever face any problem while working with Express.JS, at that time, you can easily take help from the highly supportive community that it has.

**7. Supports Caching**

Express.js supports the caching feature, and the advantage of the catch is that you would not have to re-execute the codes again and again.

Flask

* *Scalable.* I would argue more scalable than monoliths if using modern methods. Today, applications are often running in containers or using cloud computing with auto-scaling. Applications do not typically “scale” themselves. The infrastructure scales. With a smaller application, it's easier to deploy instances across thousands of server easily to handle increased traffic/load. That’s part of the reason why Pinterest needed to migrate from Django to Flask as they grew to support more of a microservices pattern.
* *Simpler Development*. If you understand Python well, then you’ll be able to move around and contribute to a Flask application pretty easily. It’s less opinionated so fewer standards to learn.
* *Flexibility*. There are very few parts of Flask that cannot be easily and safely altered because of its simplicity and minimality.
* *Performance*. You can think about a micro framework being slightly more “low-level” than something like Django. There are fewer levels of abstraction between you and the database, the requests, the cache, etc. So performance is inherently better from the start.
* *Modularity*. Modular code provides a huge number of benefits. With Flask, you have the ability to create multiple Flask applications or servers, distributed across a large network of servers, each with specific purposes. This creates more efficiency, better testability, and better performance.

Spring

* **Lightweight:** Spring Framework is lightweight with respect to size and transparency.
* **Inversion Of Control (IOC):** In Spring Framework, loose coupling is achieved using Inversion of Control. The objects give their own dependencies instead of creating or looking for dependent objects.
* **Aspect Oriented Programming (AOP):** By separating application business logic from system services, Spring Framework supports Aspect Oriented Programming and enables cohesive development.
* **Container:** Spring Framework creates and manages the life cycle and configuration of application objects.
* **MVC Framework:** Spring Framework is a MVC web application framework. This framework is configurable via interfaces and accommodates multiple view technologies.
* **Transaction Management:** For transaction management,Spring framework provides a generic abstraction layer. It is not tied to J2EE environments and it can be used in container-less environments.
* **JDBC Exception Handling:** The JDBC abstraction layer of the Spring Framework offers an exception hierarchy, which simplifies the error handling strategy.

.Net Core

1. .NET Core is Cross-Platform
2. .NET Core is Open Source
3. Mature Framework and Widely Used Programming Languages
4. The .NET Core Supports a Wide Range of Application Types
5. Increased Security with .NET Core
6. .NET Core Enables Top App Performance
7. .NET Core Enables Flexibility
8. .NET Core is Cost Effective
9. .NET Core has a Large Community
10. .NET Core was Created by Microsoft