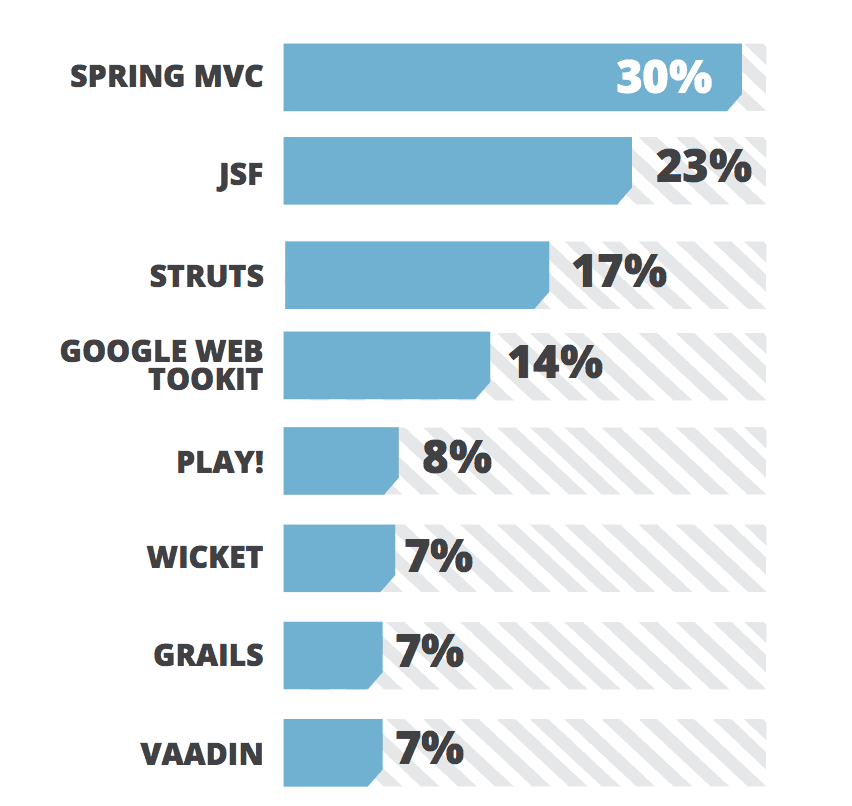
JAVA FRAMEWORKS USED FOR WEBSITES:

**Popular websites that uses Java :**

Google, LinkedIn, eBay, Amazon, Stack Overflow, ESPN, SnapDeal, Alibaba, etc.

Frameworks are tools for software development and website development that comes with pre-written codes which helps programmers to add extra codes according to their needs so that they don’t need to develop it from scratch again and again.

Java helps to develop website in easier, faster and simple way.



**Spring:**

Spring Framework is the most popular application development framework of Java. The main feature of the Spring Framework is **dependency Injection** or **Inversion of Control** (IoC). With the help of Spring Framework, we can develop a **loosely** coupled application. It is better to use if application type or characteristics are purely defined.

* Develop complex web applications
* High performance
* Loose coupling
* Enterprise-level applications

**Modules :**

* Spring MVC
* Spring Core
* Spring Security
* Spring Transaction
* Spring Boost
* Spring ORM

**Advantages:**

1. Using Spring’s flexible and comprehensive third-party libraries you can build any web application you can imagine.

2. Opens new project in seconds, fast startup, and fast shutdown.

3. Supports JDBC that increases productivity.

4. Supports Modularity , XML and annotation-based configuration.

5. Testing is Easier — The Spring Dependency injection helps developers insert test data

6. Spring provides a lightweight container that can be triggered without a web server or application server.

**For example:**

large companies might have an app that has been around for quite some time now, and they need to run it on a server whose upgraded cycle is beyond the control of developers. Other businesses might have a standalone product that does not require a server. For all of these scenarios and many others, the Spring framework has you covered.

**Disadvantages:**

1. Complexity — The Spring framework has a lot of variables and complications. Therefore, you should only use it if you have an experienced team of developers who have used this framework before.
2. Parallel Mechanisms — One of the biggest advantages of Spring is that it gives developers a wide array of options, but this could also be a disadvantage because it causes confusion. Developers have to know which features will be useful, and making the wrong decisions could lead to significant delays.
3. No Specific Guidelines — Within the Spring documentation, it says nothing about dealing with threats such as XSS or cross-site request forgery. With this in mind, you and your team will need to figure out ways on how to stop hackers from infiltrating your application yourself.

**2. JSF (Java Server Faces):**

JavaServer Faces (JSF) is developed by Oracle for creating enterprise applications, native applications, and web app development. It is used to develop user interfaces for Java-based applications.

It is a stable component-based MVC framework. It has an architecture that clearly and explicitly distinguishes between application logic and representation.

* Create User interface by drag and drop
* Not necessary to know the advances knowledge of Client-Side Technologies like HTML,CSS,JavaScript.

**Advantages:**

1. JSF is an integral part of Java EE.
2. Excellent Tools and rich libraries.
3. It allows existing backend Java Code to be extended with web interface without changing the base application.
4. The concept of action and action listener for button invocation is good.
5. By design and concept it allows to create reusable components, which helps improve productivity.

**Disadvantages:**

1. Simple tasks become difficult - Time and time again, when developing user interfaces with JSF, tasks that could be [achieved quite easily using JavaScript](https://www.theserverside.com/feature/The-state-of-testing-in-the-JavaScript-world) become a giant hassle.
2. JSF lacks flexibility - Time and time again the JSF framework lets the developer down instead of rising to the occasion.
3. The learning curve is steep - Getting [a simple Hello World application](https://www.theserverside.com/tutorial/JSF-Tutorial-Completing-the-Ajax-based-Facelets-application) up and running in JSF is fairly easy, but taking just a small step out of the shallow end of the JSF pool can leave developers significantly out of their depths.
4. Incompatibility with standard Java technologies - Another key problem with JSF is that it doesn't work well with other [standard Java libraries](https://www.theserverside.com/feature/JavaScript-front-end-frameworks-TypeScript-skills-in-demand-in-2017), with the Portal API being one, and JSTL, Java's standard tag libraries, being another.

**3. Struts**

It is a very useful framework, an open-source MVC framework for creating modern enterprise-level Java web applications that favor convention over configuration and reduce overall development time. It comes with plugins to support REST, AJAX, and JSON and can be easily integrated with other Java frameworks like Spring and Hibernate.

Struts is an open source framework that extends the Java Servlet API and employs a Model, View, Controller (MVC) architecture. It enables you to create maintainable, extensible, and flexible web applications based on standard technologies, such as JSP pages, JavaBeans, resource bundles, and XML.

* Super Flexible and beginner-friendly,
* Reliable, based on MVC design pattern.
* Integration with REST, JSON, and AJAX.
* Creative themes and templates make development tasks faster.
* Extend capabilities for complex development,
* Reduced development time and Effort, makes dev easier and fun.

**Advantages:**

1. Struts provide central organisation architecture for building and controlling MVC based web application.
2. Struts enable clean separation between code for functionality and the code for presentation.
3. Struts has build- in support for **I-18-N**.
4. Struts is build in extensible authentication and validation.
5. Struts allows modular development and easy integration with other components including enterprise components.

**Disadvantages:**

1. No backward flow.  
2. Single controller servlet – Only one ActionServlet is available which may be an issue of scalability.  
3. This framework does not any exception, where there are errors in the config files.  
4. Less transparent.  
5. Rigid approach.  
6. With struts 1, embedding application into JSP can’t be prevented.  
7. Non-XML compliance of JSP syntax

### 4. Google Web Toolkit

It is a very popular open-source Java framework used by a large number of developers around the world for building and optimizing complex browser-based applications. This framework is used for the productive development of high-performance complex web applications without being an expert in front-end technologies like JavaScript or responsive design. It converts Java code into JavaScript code which is a remarkable feature of GWT. Popular Google’s applications like AdSense and AdWords are written and using this framework.

* Google APIs are vastly used in GWT applications.
* Open-source and Developer-friendly.
* Easily create beautiful UIs without vast knowledge in front-end scripting languages.
* Create optimized web applications that are easy to debug.
* Compiles the Java source code into JavaScript files that can run on all major browsers.

**Advantages:**

* Open-source and completely free.
* Developer-friendly.
* It supports reusability for common web development tasks.
* Google APIs can be used in GWT applications.
* Offers internationalization, cross-browser portability, UI abstraction, bookmarking, and history management.

**Disadvantages:**

1. GWT is a fast developing project. So, there are a lot of versions floating around. Many functions, interfaces and events get deprecated and keeping up with their pace is not too much fun when you have other work to do
2. There were quite a few GWT books during the beginning. Not so much these days
3. The Java to JavaScript compilation is fairly slow, which is a significant con if you choose GWT.
4. I personally prefer defining structure in HTML and styling it using CSS. The concepts used in HTML are clean and straight-forward and I have years of experience doing just that. But in GWT, I am kind of forced to use proprietary methods to do the same. That combined with the fact that GWT doesn’t solve the styling and alignment incompatibilies for me compounds the problem. So, writing layout code in GWT is something I despice. But with UI Binder and HTMLLayout from version 2.0 onwards, I feel I am back in my own territory
5. It requires some serious commitment levels to get into GWT, coz, after that, a change in client side technology could require a complete rewrite of your app, as it is a radically different approach than other client side frameworks

### [5. Grails:](https://grails.org/)

It is a dynamic full-stack Java framework based on the MVC design pattern. which is easy to learn and most suitable for beginners. Grails is an object-oriented language that enhances developer productivity.  While written in Groovy it can run on the Java platform and is perfectly compatible with Java syntax.

* Easy to Creating tags for the View,
* built-in support for RESTful APIs,
* you can mix Groovy and Java using Grails,
* best suitable for Rapid Development,
* configuration features are dynamic, no need to restart the server.

**Advantages:**

1. It provides a rapid development cycle.
2. If you are dealing with a small or medium-size project, Grails is ideal for you.
3. The framework offers a range of plug-ins to make your ob simple.
4. The documentation is really impressive.
5. The setup process is very simple. Therefore, you should be able to start building an app in an hour.
6. Simple GORM. It may take some time to learn but once you are familiar with it, it’s absolutely wonderful.
7. You can see the changes by hitting the refresh button.
8. Less CSS framework plug-ins. Hence, it’s easier to manage the CSS.
9. Dynamic configuration feature. Therefore, you can change the configuration without server restart.

**Disadvantages:**

1. You have to deal with runtime language. Its negative quality is that it is error prone and you have to bear with other cons of runtime language.
2. If you are working on a multi threaded app, GORM can be problematic for you.
3. You have to buy IntelliJ Idea because other IDE’s either don’t have or provide limited support.
4. Developers generally declare variables with “def” which is equivalent to “object”. It’s very hard to maintain.
5. Interpreted languages increase weight and that directly affects the run time.
6. You must learn Groovy coding.
7. It works with GORM but not with any other ORMs.
8. Integration process is complicated.

**6. Hibernate**

A stable, lightweight ORM Java framework that can easily communicate with any database and is more convenient when working with multiple databases. Working with Hibernate is fun using powerful APIs and several useful tools like Mapping Editor, Wizards, and Reverse Engineering. Many big companies including Platform, DAILY HOTEL, IBM, and Dell use Hibernate in their tech stacks,

* Light-weight and easy to scale up, modify, and configure.
* Complex data manipulation with less coding.
* High productivity and portability,
* Used for RDBMS as well as the NoSQL database.
* Awesome Command-line tools and IDE plugins to makes your experience pleasant.

**Advantages:**

1. Hibernate is an ORM tool
2. Hibernate is an open source framework.
3. Better than JBDC.
4. Hibernate has an exception translator , which converts checked exceptions of JDBC in to unchecked exceptions of hibernate. So all exceptions in hibernate are unchecked exceptions and Because of this no need to handle exceptions explicitly.
5. Hibernate supports inheritance and polymorphism.
6. With hibernate we can manage the data stored across multiple tables, by applying relations(association)
7. Hibernate has its own query language called Hibernate Query Language. With this HQL hibernate became database independent.

**Disadvantages:**

1. Debugging: Sometimes debugging and performance tuning becomes difficult.

2. Slower than JDBC: Hibernate is slower than pure Jdbc as it is generating lots of SQL statements in runtime.

3. Not Suitable for Batch Processing: It is advisable to use pure JDBC for batch processing.

4. Not suitable for Small projects: For small project having few tables it is useless to work with hibernate.

5. Lots of API to learn: A lot of effort is required to learn hibernate. So, not very easy to learn hibernate easily.

6. Generates complex queries with many joins.