CHAPTER - 4

WEB-BASED FRAMEWORKS

Content Management System:

A content management system (CMS) is a software application or set of related programs that are used to create and manage digital content. CMS are typically used for Enterprise Content Management (ECM) and Web Content Management (WCM). Both Enterprise Content Management and Web Content Management systems have two components:

- ☑ A content management application (CMA)
- ☑ A content delivery application (CDA).

The CMA is a graphical user interface (GUI) that allows the user to control the creation, modification and removal of content from a website without needing to know anything about HTML. The CDA component provides the back-end services that support management and delivery of the content once it has been created in the CMA.

Importance of Content Management System:

1. **Increase Efficiency:**

Content can be published easily and efficiently as editing and revisions do not require visual design or coding knowledge. This allows for fast and efficient updates, saving our business cost and time.

2. **Increase Our Search Engine Ranking:**

To improve or maintain our search engine ranking our business has to remain relevant, and a good and easy-to-use CMS will help our publishers keep the content fresh. This invites external contribution (e.g. comments, forum, likes etc…), an integral component in staying relevant and improving our search engine ranking.

3. **Maintain Control Over Our Content:**

Workflow is a core feature of any good CMS. Irrespective of how basic our workflow needs might be, workflows will ensure our business maintains control over content.

4. **Help Our Visitors In Their Search For Information:**

With powerful CMS search engines new content is indexed automatically so it can be instantly found. Visitors can also use taxonomy applications, sorting lists, saved searches and more to personalize the search experience.

5. **Cross-Selling:**

Sophisticated Content Management Systems can learn user behavior and preferences, making our cross-selling and up-selling efforts much more cost effective.

6. **Improve Online Branding:**

Our marketing team can keep our business relevant by multi-channel campaign management (e.g. emails, brochures, RSS, hyper-sites, dynamic content management etc.)

7. **Extensibility:**

Our CMS will support our expansion, and most CMSs can extend functionality beyond its default capabilities, typically by purchasing additional plug-ins or modules.

8. **Improve Customer Service:**

FAQs, help sections and support forms are essential in keeping our customers happy, and any good CMS will allow our business to easily maintain those support channels.

9. **Mobile Optimization:**

Advanced CMSs offer mobile optimization, automatically tailoring presentation and content for different devices.

Examples of CMSs:

1. **Wordpress**:

The WordPress Community is a faithful and zealous bunch. Wordpress probably has the widest base of plugins and themes to choose from. We have thousands of professional Wordpress Themes and Wordpress Plugins available for sale on Envato Market, with a full suite of styles and options to choose from.

A great part about the Wordpress community is the amount of help and documentation online we can find on nearly every aspect of customizing WordPress. If we can dream it, chances are it's already been done with WordPress and documented somewhere.

2. **Drupal:**

Drupal is another CMS that has a very large, active community. Instead of focusing on blogging as a platform, Drupal is more of a pure CMS. A plain installation comes with a ton of optional modules that can add lots of interesting features like forums, user blogs, OpenID, profiles and more. It's trivial to create a site with social features with a simple install of Drupal. In fact, with a few 3rd party modules we can create some interesting site clones with little effort.

3. **Joomla:**

Joomla is very similar to Drupal in that it's a complete CMS, and might be a bit much for a simple portfolio site. It comes with an attractive administration interface, complete with intuitive drop-down menus and other features. The CMS also has great support for access control protocols like LDAP, OpenID and even Gmail.com.

4. **ExpressionEngine:**

ExpressionEngine is packed with helpful features like the ability to have multiple sites with one installation of software. For designers, EE has a powerful templating engine that has custom

global variables, custom SQL queries and a built in versioning system. Template caching, query caching and tag caching keep the site running quickly too.

5. **Textpattern:**

Textpattern is a popular choice for designers because of its simple elegance. Textpattern isn't a CMS that throws in every feature it can think of. The code base is svelte and minimal. The main goal of Textpattern is to provide an excellent CMS that creates well-structured, standards-compliant pages. Instead of providing a WYSIWYG editor, Textpattern uses textile markup in the textareas to create HTML elements within the pages. The pages that are generated are extremely lightweight and fast-loading.

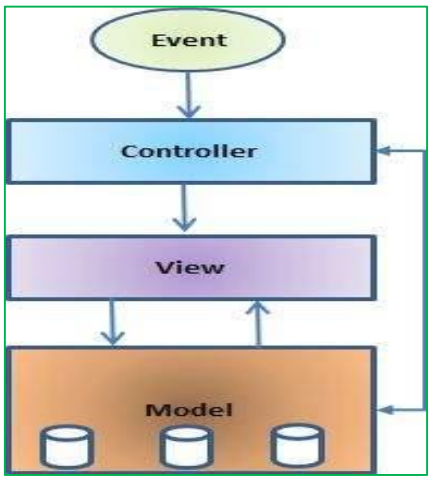
6. **SilverStripe:**

SilverStripe is another PHP CMS that behaves much like Wordpress, except has many more configurable options and is tailored towards content management, and not blogging. SilverStripe is unique because it was built upon its very own PHP framework Saphire. It also provides its own templating language to help with the design process.

7. **TYPOlight:**

TYPOlight seems to have the perfect balance of features built into the CMS. In terms of functionality, TYPOlight ranks with Drupal and ExpressionEngine, and even offers some unique bundled modules like newsletters and calendars. Developers can save time with the built-in CSS generator, and there are plenty of resources for learning more about the CMS.

Model View Controller:



The **Model-View-Controller (MVC)** is an architectural pattern that separates an application into three main logical components: the **model**, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.

☑ **Model:**

The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data.

☑ **View:**

The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

☑ **Controller:**

Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.

Web Programming Frameworks:

A web framework (WF) or web application framework (WAF) is a software framework that is designed to support the development of web applications including web services, web resources and web APIs. Web frameworks aim to alleviate the overhead associated with common activities performed in web development. For example, many web frameworks provide libraries for database access, templating frameworks and session management, and they often promote code reuse. Here are few aspects of a proficient framework:

☑ **MVC Architecture:**

The framework should make use of Model View Controller (MVC) architecture. Some of the best frameworks also provide libraries, plug-ins, helpers, and extensions to assist developers. It would be smart and efficient to employ a framework that has at least two of these options.

☑ **Database Support**:

It is one of the most crucial aspects of every PHP development framework. You need decide your framework depending on the database you are going to use for your web application. For Example: 'CodeIgniter' supports MySQL, Oracle, and SQLite, whereas the 'Kohana' framework doesn't.

☑ **Community & Documentation**:

The framework should be supported by a strong community, not just in terms of size but also in terms of activity and usefulness. Even if it's a small community, we should be able to get ample support from the community. A framework should also have good documentation. It should be comprehensive and up-to-date.

Java Frameworks:

There are plenty of Java frameworks available on the web, and developers use them quite frequently to build websites. These frameworks offer lots of functions and reduce the working hours by leaps and bounds. These frameworks make the coding process faster, thanks to their inbuilt modules and functions.

Sometimes too many options can confuse us and that is exactly the case with Java frameworks. If we ask for suggestions in web development and programming forums, every developer will give us different suggestions based on their personal likes and dislikes. As a result, it becomes very difficult to select the one framework that will fit all our requirements. Here are the 10 Best Java Web Frameworks for 2017:

1. **Spring MVC (Model View Controller)**

The Spring MVC framework is designed as a layered J2EE/Java framework integrating specially applied sciences. It is appropriate for a broad range of ingenuities. After its design and expansion, Spring MVC embraced numerous changes to become a full-scale framework Java for Internet applications. It offers a useful toolkit for development and configuration of web applications. We can use it for our security projects. Spring is famous among programmers for its well-developed ecosystem. It has numerous add-ons, such as SOAP services, REST APIs, and security authentication.

Pros:
Spring is one of the best Java frameworks, and it has various qualities:

- ☑ Enhanced modularity to improve readability of code
- ☑ Simplified injection to test data through POJOs
- ☑ Flexible use of DI (Dependency Injection)
- ☑ Loose coupling among different modules

Cons:
Along with pros, here are some cons of Spring web MVC framework:

- ☑ MVC can be a bit difficult to implement for a newbie
- ☑ Makes the learning curve steep

2. **JSF (JavaServer Faces):**

JSF is a net utility framework of Java, and it got Oracle's support. This framework may not be the best framework for Java development, but it is easy to utilize because of documentation supplied by Oracle. Moreover, if you want EE environment Java, there will be no extra dependency on JSF. Tools and rich libraries become a beautiful magic wand that may back you up to the complexity of an application. JSF uses server pages of Java and can support different technologies of Facelets and XUL.

Pro:

- ☑ JSF is an important part of Java EE and as such convenient for developers who use IDE software.

Con:

- ☑ JavaServer Faces is broad to comprehend without prior skills and experience Java web development.

3. **Struts:**

Struts makes the list our list of Best Java Web Frameworks for 2017. Struts is a Framework to develop the base of the web application. This framework is a set of interfaces and classes that

co-operate to solve the problems of a particular type. It functions on MVC (Model-View-Controller) pattern. Struts 2 is equipped with a net framework for numerous Java applications. It is the result of mixed effort from various supporting communities. Unlike conventional functions of the internet, it can create dynamic responses.

Pros:

- ☑ Provides internal organization architecture to control and build MVC based application for the web application
- ☑ Built-in support for I-18-N
- ☑ Struts is constructed in extension validation and authentication
- ☑ Allow modular development and integration with additional components

Cons:

- ☑ Framework is not flexible
- ☑ Framework imposes coding, designing and thinking

4. **Hibernate:**

While talking about the best Java web frameworks, Hibernate ORM can't be ignored. Hibernate is an essential object-relation mapping device for programming language Java. It offers a mapping framework for a domain model (object-oriented) to one relational database. Hibernate can solve the object-relational impedance incongruity problems by substituting persistent and direct database with high-level object controlling functions. It is free software distributed under public 2.1 License of GNU Lesser General.

Pros:

- ☑ Hibernate enables us to communicate with any database by making tiny alternations in code
- ☑ MySQL, Db2 or Oracle, Hibernate is DB independent
- ☑ Caching instrument to bug catalog with same queries
- ☑ N+1 or Sluggish loading support
- ☑ Low risk of data loss and it requires less power

Cons:

- ☑ If power goes off, we can lose all our data
- ☑ Restarting can be extremely slow

5. **GWT (Google Web Toolkit):**

GWT (Google Web Toolkit) is an open source tool set that enables web developers to maintain and create complex JavaScript front-end applications. Other than some native libraries, the whole thing in JavaSource can be built on a supported platform with integrated GWT Ant

construct files. The application is licensed under Apache License 2.0 version. Google web toolkit highlights reusable approaches to common tasks of web development, namely cross-browser portability, internationalization, UI abstraction, bookmarking, history management, remote procedure calls and asynchronous.

Pros:

- ☑ GWT is easy to learn
- ☑ Great to use in creating significantly responsive web application with heavy lifting on client-side and decrease load on server-side
- ☑ There are various JavaScript libraries out there thus making developers appreciate the true power of GWT
- ☑ Built-in IDE support to directly refactor Java cryptogram/code to maintain a good design at all time
- ☑ Google develops GWT, and this project has a bright future

Cons:

- ☑ GWT is a fast evolving project, and numerous versions are floating around. You may not get all interfaces and functions
- ☑ GWT compilation is slow, and we have to use proprietary method to define structure

6. Play! Framework:

Play framework makes it easy for you to build web applications with Scala and Java. It is based on stateless, web-friendly and lightweight architecture. It is constructed on Akka and provides minimal and predictable resource consumption (threads, memory, and CPU) for highly-scalable applications. The play framework is friendly for developers to make changes with text editor and browser. Play utilizes a fully asynchronous model designed on the top of Akka. Furthermore, although being stateless it scales predictably. Play was designed for needs of modern mobile and web applications.

Pros:

- ☑ Dramatically improves the productivity of any developer
- ☑ Quick reload for config changes, templates and java code
- ☑ Play is designed on Netty and supports non-blocking I/O
- ☑ It is an open source, and everything works well under the hood
- ☑ Zenexity and Typesafe offer commercial support
- ☑ Play can handle error in dev mode for runtime and compile errors
- ☑ Scala and Java use type-safe language, reliable and JVM performance to scale to various developers and users

Cons:

- ☑ Play 2 is basically a a rewrite of the Play 1
- ☑ It is built around I/O async that means writing code to execute later. You will end up with unidentified inner classes

7. Vaadin:

Vaadin framework and elements are open source Apache licensed Java web application projects with an active worldwide community. It is a good framework for internet applications and contrast to browser-plugin and JavaScript libraries based solution; it features server-side architecture. Ajax technology can run on browser-side to ensure interactive and rich user experience. The Vaadin framework can be extended with GWT.

Pros:

- ☑ Server-side programming only
- ☑ Layout, listeners and component types are available
- ☑ Plug-in support
- ☑ Easy integration with the Spring framework

Cons:

- ☑ Multi-window support is not available
- ☑ Size of created JavaScript code may grow really large

8. Grails:

Grails is another useful 10 Best Java Web Frameworks on the Java platform meant to multiply the productivity of Java web developers towards convention-over-configuration, opinionated APIs, and sensible defaults. It assimilates smoothly with JVM (Java Virtual Machine) and enables you to immediately become productive while providing powerful features, including asynchronous programming, Compile-time meta-programming, run-time and domain-specific languages. You can transparently and seamlessly inter-operate and integrate with Java, Java EE containers, and JVM.

Pros:

- ☑ Grails is ideal for dealing with medium or small-size project
- ☑ Offers a rapid development cycle
- ☑ Offers a variety of plug-in to make our job simple
- ☑ Documentation is easy and impressive
- ☑ Setup process is simple, and we will be able to build app in one hour
- ☑ Simple GORM and it works wonderful
- ☑ See changes by hitting refresh button
- ☑ Easy to manage CSS
- ☑ Dynamic configuration feature to change configuration without restarting your server

Cons:

- ☑ We have to work with runtime language
- ☑ It can be problematic to work on multi-threaded application

9. Wicket:

Apache Wicket is also known as Wicket, and it is among the 10 Best Java Web Frameworks. It is an open source, server-side, component oriented and Java application framework. If you are using Wicket, there is no need to have distorted HTML pages. The Java framework require the insertion of special syntax to HTML code which is easy for for website designers. Wicket espouses HTML templates with the use of namespace that follows XHTML standard.

Pros:

- ☑ Code is scripted in Java
- ☑ Zero XML configuration files
- ☑ No back-button problems
- ☑ POJO-centric programming
- ☑ Easy create bookmarkable pages
- ☑ Testability of constituents

Cons:

- ☑ Development can be confusing in general
- ☑ Event-based design may increase chances of code mess

10. Vert.x

Vert.x is an appropriate polyglot event-driven app framework that runs on the Java Virtual Machine (JVM). Eclipse Vert.x is non-blocking and event-driven to handle lots of concurrency with the help of kernel threads. It helps your app to scale with negligible hardware easily. It is easy to use Vert.x with numerous languages, such as Ceylon, Ruby, Groovy, JavaScript, and Java thus you may choose the language on the basis of your project and team skills set.

Pros:

- ☑ Easy to set up
- ☑ Install an isolated container
- ☑ Special thread in the Vert.x container
- ☑ Vert.x offer simple methods and objects for non-blocking

Cons:

- ☑ Can be difficult to scale with some hardware
- ☑ Can be confusing to choose a language

PHP Frameworks:

A framework is a structure that developers choose to build their application. It determines the structure of the application and facilitates it to connect with many different API's. A proficient PHP framework enables developers to develop PHP application faster, efficiently and assist in building stable applications thereby reducing the amount of repetitive coding for PHP programmers.

Frameworks provide scaffolding features that facilitates the development team to build faster and cleaner application. They often provide toolsets for both the UI components and the database access. ***Note: It is advisable to use the latest stable version of a framework.***

1. **Laravel:**

Although Laravel is a relatively new PHP framework (it was released in 2011), according to Sitepoint's recent online survey it is the most popular framework among developers. Laravel has a huge ecosystem with an instant hosting and deployment platform, and its official website offers many screencast tutorials called Laracasts.

Laravel has many features that make rapid application development possible. Laravel has its own light-weight templating engine called "Blade", elegant syntax that facilitates tasks you frequently need to do, such as authentication, sessions, queueing, caching and RESTful routing. Laravel also includes a local development environment called Homestead that is a packaged Vagrant box.

2. **Symfony:**

The components of the Symfony 2 framework are used by many impressive projects such as the Drupal content management system, or the phpBB forum software, but Laravel – the framework listed above – also relies on it. Symfony has a wide developer community and many ardent fans.

Symfony Components are reusable PHP libraries that we can complete different tasks with, such as form creation, object configuration, routing, authentication, templating, and many others. We can install any of the Components with the Composer PHP dependency manager. The website of Symfony has a cool showcase section where we can take a peek at the projects developers accomplished with the help of this handy framework.

3. **CodeIgniter:**

CodeIgniter is a lightweight PHP framework that is almost 10 years old (initially released in 2006). CodeIgniter has a very straightforward installation process that requires only a minimal configuration, so it can save our lot of hassle. It's also an ideal choice if we want to avoid PHP version conflict, as it works nicely on almost all shared and dedicated hosting platforms (currently requires only PHP 5.2.4).

CodeIgniter is not strictly based on the MVC development pattern. Using Controller classes is a must, but Models and Views are optional, and we can use our own coding and naming conventions, evidence that CodeIgniter gives great freedom to developers. If we download it, we'll see it's only about 2MB, so it's a lean framework, but it allows us to add third-party plugins if we need more complicated functionalities.

4. **Yii 2:**

If we choose the Yii framework we give a boost to the performance of our site as it's faster than other PHP frameworks, because it extensively uses the lazy loading technique. Yii 2 is purely object-oriented, and it's based on the DRY (Don't Repeat Yourself) coding concept, so it provides us with a pretty clean and logical code base.

Yii 2 is integrated with jQuery, and it comes with a set of AJAX-enabled features, and it implements an easy-to-use skinning and theming mechanism, so it can be a great choice for someone who comes from a frontend background. It has also a powerful class code generator called Gii that facilitates object-oriented programming and rapid prototyping, and provides a web-based interface that allows us to interactively generate the code we need.

5. **Phalcon:**

The Phalcon framework was released in 2012, and it quickly gained popularity among PHP developers. Phalcon is said to be fast as a falcon, because it was written in C and C++ to reach the highest level of performance optimization possible. Good news is that you don't have to learn the C language, as the functionality is exposed as PHP classes that are ready to use for any application.

As Phalcon is delivered as a C-extension, its architecture is optimized at low levels which significantly reduces the overhead typical of MVC-based apps. Phalcon not only boosts execution speeds, but also decreases resource usage. Phalcon is also packed with many cool features such as a universal auto-loader, asset management, security, translation, caching, and many others. As it's a well-documented and easy-to-use framework, it's definitely worth a try.

6. **CakePHP:**

CakePHP is already a decade old (the first version was released in 2005), but it's still among the most popular PHP frameworks, as it has always managed to keep up with time. The latest version, CakePHP 3.0 enhanced session management, improved modularity by decoupling several components, and increased the ability of creating more standalone libraries.

CakePHP has a really remarkable showcase, it powers the websites of big brands such as BMW, Hyundai, and Express. It is an excellent tool for creating web apps that need high-level of security, as it has many built-in security features such as input validation, SQL injection prevention, XSS (cross-site scripting) prevention, CSRF (cross-site request forgery) protection, and many others.

7. **Zend Framework:**

Zend is a robust and stable PHP framework packed with a lot of configuration options therefore it's usually not recommended for smaller projects but excellent for more complex ones. Zend has partners such as IBM, Microsoft, Google and Adobe. The coming major release, Zend Framework 3 will be optimized for PHP 7, but will still support PHP 5.5 onwards.

The current release, Zend Framework 2 also has many cool features such as cryptographic coding tools, an easy-to-use drag and drop editor with support for front-end technologies (HTML, CSS, and JavaScript), instant online debugging and PHP Unit testing tools, and a

connected Database Wizard. Zend Framework was created with the agile methodology that facilitates delivering high-quality apps to enterprise clients.

8. **Slim:**

Slim is a PHP micro framework that provides us with everything we need and nothing we don't. Micro frameworks are minimalistic in design, they are excellent for smaller apps where a full-stack framework would be an exaggeration. Slim's creator was inspired by a Ruby micro framework called Sinatra.

Slim is used by many PHP developers for developing RESTful APIs and services. Slim comes with features such as URL routing, client-side HTTP caching, session- and cookie encryption, and it supports "flash" messages across HTTP requests as well.

9. **FuelPHP:**

FuelPHP is a flexible full-stack PHP framework that doesn't only support the ordinary MVC pattern but also its evolved version, HMVC (Hierarchical Model-View-Controller) at the architecture level. FuelPHP adds an optional class called Presenter (formerly called ViewModel) between the Controller and View layers to hold the logic needed to generate Views.

FuelPHP is modular and extendable, takes care of security concerns by providing features such as input and URI filtering and output encoding, and it comes with its own authentication framework, with many other sophisticated features and a detailed documentation.

10. **PHPixie:**

PHPixie is a quite new framework, it started in 2012 with the goal of creating a high-performance framework for read-only websites. PHPixie also implements the HMVC design pattern just like FuelPHP, and is built by using independent components that can be used as well without the framework itself. The PHPixie components are 100% unit tested, and require minimum dependencies.

The official website has a tutorial that claims you can learn the framework in 30 minutes, and their blog also details many practical use cases. Among the features we can find the standard ORM (object-relational mapping), caching, input validation, authentication and authorization capabilities. PHPixie also allows you to use the HAML markup language, enables schema migration, and has a sophisticated routing system.