Guide to Web Application Development

Simply put, Web Applications are dynamic web sites combined with server side programming which provide functionalities such as interacting with users, connecting to back-end databases, and generating results to browsers.

Web application development is the creation of application programs that reside on remote servers and are delivered to the user's device over the Internet. A web application (web app) does not need to be downloaded and is instead accessed through a network. An end user can access a web application through a web browser such as Google Chrome, Safari, or Mozilla Firefox. A majority of web applications can be written in HTML5, Cascading Style Sheets (CSS), and JavaScript,.

Web application development will typically have a short development life-cycle lead by a small development team. Front-end development for web applications is accomplished through client-side programming. Client refers to a computer application such as a web browser. Client-side programming will typically utilize HTML, CSS and JavaScript.

HTML programming will instruct a browser how to display the on-screen content of web pages, while CSS keeps displayed information in the correct format. JavaScript will run JavaScript code on a web page, making some of the content interactive.

Examples *of Web Applications are Online Banking, Social Networking, Online Reservations, eCommerce / Shopping Cart Applications, Interactive Games, Online Training, Online Polls, Blogs, Online Forums, Content Management Systems, etc..*

Web App Developer: A software designer or engineer who is involved in the development and design of web and/or network applications; An IT professional who creates websites and writes code for web applications;

Web app Developer works on the logic of the application; build the framework/backbone of the website; provides clean code for website/web application.

.

Technologies

There are two main categories of coding, scripting and programming for creating Web Applications:

I. Client Side Scripting / Coding - Client Side Scripting is the type of code that is executed or interpreted by browsers.

Client Side Scripting is generally viewable by any visitor to a site (from the view menu click on "View Source" to view the source code).

Below are some common Client Side Scripting technologies:

- HTML (HyperText Markup Language)
- CSS (Cascading Style Sheets)
- JavaScript
- Ajax (Asynchronous JavaScript and XML)
- jQuery (JavaScript Framework Library - commonly used in Ajax development)
- MooTools (JavaScript Framework Library - commonly used in Ajax development)
- Dojo Toolkit (JavaScript Framework Library - commonly used in Ajax development)

II. Server Side Scripting / Coding - Server Side Scripting is the type of code that is executed or interpreted by the web server.

Server Side Scripting is not viewable or accessible by any visitor or general public.

Below are the common Server Side Scripting technologies:

- Python (general purpose high-level programming language and Server Side Scripting language - free redistribution)
- Server Side JavaScript (SSJS) (An extended version of JavaScript that enables **back-end access** to databases, file systems, and servers. Server side javascript is javascript code running over a server **local resources**, it's just like C# or Java, but the syntax is based on JavaScript. A good example of this is **Node.JS**)
- PHP (very common Server Side Scripting language - Linux / Unix based Open Source - free redistribution, usually combines with MySQL database)
- Zend Framework (PHP's Object Oriented Web Application Framework)
- ASP (Microsoft Web Server (IIS) Scripting language)
- ASP.NET (Microsoft's Web Application Framework - successor of ASP)
- ColdFusion (Adobe's Web Application Framework)
- Ruby on Rails (Ruby programming's Web Application Framework - free redistribution)
- Perl (general purpose high-level programming language and Server Side Scripting Language - free redistribution - lost its popularity to PHP)

Program Libraries

Program libraries are a collection of commonly used functions, classes or subroutines which provide ease of development and maintenance by allowing developers to easily add or edit functionalities to a frameworked or modular type application.

Web Application Frameworks

Web Application Frameworks are sets of program libraries, components and tools organized in an architecture system allowing developers to build and maintain complex web application projects using a fast and efficient approach.

Web Application Frameworks are designed to streamline programming and promote code reuse by setting forth folder organization and structure, documentation, guidelines and libraries (reusable codes for common functions and classes).

Web Application Frameworks - Benefits and Advantages

- Program actions and logic are separated from the HTML, CSS and design files. This helps designers (without any programming experience) to be able to edit the interface and make design changes without help from a programmer.
- Builds are based on the module, libraries and tools, allowing programmers to easily share libraries and implement complex functionalities and features in a fast and efficient manner.
- The structure helps produce best practice coding with consistent logic and coding standards, and provides other developers the ability to become familiar with the code in a short time.

Coding Guidelines, Standards & Convention

Coding guidelines are sets of rules and standards used in programming a web application project.

These rules and standards apply to coding logic, folder structure and names, file names, file organization, formatting and indentation, statements, classes and functions, and naming conventions. These rules also enforce writing clear comments and provide documentation.

Important benefits of using Coding Guidelines

- Creates the best environment for multiple programmers to work on the same project
- Provides ease of maintainability and version management
- Delivers better readability and understanding of the source code
- Insures that other developers can understand and become familiar with the code in a short time

Web Applications Lifecycle Model

Web Application Lifecycle is the process of developing a web application and involvement of the multiple teams that are engaged in the development process. Each organization may set forth its own unique style of operating.

Some companies follow a certain standard model such as SDLC (System Development Life Cycle) or Agile Software Development Model.

- SDLC is the traditional process of developing software or web applications by including research to identify and define the application requirements, information analysis, architectural design and specifications blueprint, team involvement, programming, testing and bug fixing, system testing, implementation and maintenance.
- Agile Software / Web Application Development is the iterative development process and development process practices that focus on collaboration of people involved and provide a better procedure to allow revisions and evolution of web application requirements. Agile methodology includes research, analysis, project management, design, programming, implementation, frequent testing, adaptation and maintenance.

Web Application Development Process

Web Application Development Process organizes a practical procedure and approach in application development.

The following list of procedures and suggested documents provide a good outline for a Web Application Lifecycle and Process:

- Roadmap Document: Defining Web Application, Purpose, Goals and Direction
- Researching and Defining Audience Scope and Security Documents
- Creating Functional Specifications or Feature Summary Document
- Team Collaboration and Project Management Document
- Technology Selection, Technical Specifications, Illustrative Diagram of Web Application Architecture and Structure, Development Methodology, Versions Control, Backups, Upgrades, Expansion and Growth Planning Document, Server Hardware / Software Selection
- Third Party Vendors Analysis and Selection (Merchant Account and Payment Gateway, SSL Certificate, Managed Server / Colocated Server Provider, Fulfillment Centers, Website Visitor Analytics Software, Third Party Checkout Systems, etc.)
- Application Visual Guide, Design Layout, Interface Design, Wire Framing
- Database Structure Design and Web Application Development
- Testing: Quality Assurance, Multiple Browser Compatibility, Security, Performance - Load and Stress Testing, Usability
- Maintenance

Web Application Testing

Testing is an important part of the Web Application Development process. On occasion, testing would consume more manpower and time than development itself.

Below are some of the most common testing needed for any web application development process:

- Quality Assurance and Bug Testing
- Multiple Browser Compatibility
- Application Security
- Performance - Load and Stress Testing
- Usability