# Module #2 Assignment 2

## Angular JavaScript Framework - AngularJS is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly. AngularJS's data binding and dependency injection eliminate much of the code you would otherwise have to write. And it all happens within the browser, making it an ideal partner with any server technology.

## AngularJS is what HTML would have been, had it been designed for applications. HTML is a great declarative language for static documents. It does not contain much in the way of creating applications, and as a result building web applications is an exercise in what do I have to do to trick the browser into doing what I want?

1. **SASS** - Sass is an extension of CSS that adds power and elegance to the basic language. It allows you to use variables, nested rules, mixins, inline imports, and more, all with a fully CSS-compatible syntax. Sass helps keep large stylesheets well-organized, and get small stylesheets up and running quickly, particularly with the help of the Compass style library.
2. **SQL** -SQL (pronounced "ess-que-el") stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database. This tutorial will provide you with the instruction on the basics of each of these commands as well as allow you to put them to practice using the SQL Interpreter.
3. **JavaScript** - Originally developed by Brendan Eich and originally known as LiveScript, the programming language JavaScript was renamed in 1995. JavaScript is an interpreted client-side scripting language that allows a web designer the ability to insert code into their web page. JavaScript is commonly placed into an HTML or ASP file, and runs directly from the web page. It is utilized to perform tasks such as printing the time and date, create a calendar, or other tasks that are not possible through plain HTML.
4. **PHP** - PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.
5. **Ruby** - Ruby is a simple and powerful object-oriented programming language, created by Yukihiro Matsumoto (who goes by the handle “Matz” in this document and on the mailing lists).

Like Perl, Ruby is good at text processing. Like Smalltalk, everything in Ruby is an object, and Ruby has blocks, iterators, meta-classes and other good stuff.

You can use Ruby to write servers, experiment with prototypes, and for everyday programming tasks. As a fully-integrated object-oriented language, Ruby scales well.

1. **HTML5** - HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current version of the HTML standard.
2. **CSS** - Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.
3. **Webpack** - Webpack takes modules with dependencies and generates static assets representing those modules.
4. **REST API** - An architectural style called REST (Representational State Transfer) advocates that web applications should use HTTP as it was originally envisioned. Lookups should use GET requests. PUT, POST, and DELETE requests should be used for \*mutation, creation, and deletion respectively \*.