**DEVELOPING THE REAL-TIME TRANSIT INFORMATION PLATFORM**

**1. HTML**

HTML stands for Hyper Text Markup Language. It’s one of the fundamental technologies required for web development. It provides the base structure for a web page. HTML code ensures that all the content on a website is properly formatted. This is so your Internet browser can display the content as intended. Without HTML, a browser couldn’t display text or load images and other elements.

HTML5, the most current version of HTML, specifies a large number of [Application Programming Interfaces (API)](https://tillerdigital.com/glossary/api-integrations/) that can be used with JavaScript for a more interactive and dynamic website:

* Canvas: Canvas is an HTML5 element used to draw images and shapes and manipulate them. It can also be used for more complex cases such as game graphics and animations.
* Web Storage: Web Storage is used to store information right in the browser. Some examples of this would be storing user login information and saving user preferences for a website.
* Service workers: Service workers enable a script that keeps running in the background when a web page is opened and is mainly used in websites with offline capabilities. It makes pages available offline and allows for the use of web push notifications. It can send these notifications even when your browser isn’t open.
* WebSockets: WebSockets allows for persistent two-way connection between the user and the server. The most common use cases include chats and notifications in web apps.

**2. CSS**

Cascading Style Sheets, abbreviated as CSS, define the style and aesthetics of a web page. While HTML is used to structure a web page, CSS specifies the appearance of that structure. This includes page layouts, colours, fonts and element positioning. If HTML is the bones of the web page, CSS is the skin. It makes the Internet, and your website, look good.

**3. Programming languages**

Programming languages are ways to communicate with computers and tell them what to do. They provide a vocabulary and set of grammatical rules to instruct a computer to perform specific functions.

There are lots of different programming languages just as there are lots of human languages. Developers typically specialize in a few languages.

Below are a few examples of different languages:

**JavaScript**

JavaScript is the most used language by startups for a few reasons. It can be used as both a front-end and back-end programming language. Compared to other languages, JavaScript is comparatively easy-to-learn (although all languages come with some difficulties), and it’s used everywhere in . It’s also constantly being updated and expanded with new features.

All the following are back-end languages that make a website or application work:

**PHP**

PHP is the most common language in web development. WordPress, the world’s most popular content management system (CMS), is built on PHP. Finding a good developer can be difficult regardless of language, but choosing a popular one like PHP will give you a larger pool of developers to choose from and work with. PHP also comes with extensive online documentation and tutorials for troubleshooting and support.

**Python**

Python is another popular programming language. It’s relatively simple and uses a unique syntax that focuses on readability. Python is good for artificial intelligence (AI) applications and projects, statistics, data science work, and machine learning. It can also be used to create web applications.

**Java**

Java is a language that’s popular in enterprise companies. It can be run on multiple platforms, has extensive documentation, and is supported by Oracle (a large and respected software company) in case you run into trouble. It can also be used to write Android applications.

**Rust**

Rust is a newer language that focuses on safety and especially safe concurrency. Concurrency in software engineering means the collection of techniques that enable a computer program to perform several different tasks simultaneously. Safe concurrency gives you the efficiency needed to execute multiple tasks while keeping your code safe and secure. Rust is high performing and has gained market traction even as a newer language.

**4. Web development frameworks**

Web development frameworks are tools and libraries that developers use to make developing in a particular language easier and more efficient. They provide interfaces to access commonly-used functionalities as well as abstractions that make complicated things easier to understand and handle.

Essentially, they make the development process more efficient because developers have a pool of code resources to draw from when developing. There’s no need to develop software functions from scratch when frameworks give you all the necessary tools to build a website or application. Because frameworks give you a headstart on your development, they’re more cost effective than developing entirely new code. They also lower your development risk since they’re tried and true compared to writing brand new, untested code.

There are both front-end and back-end frameworks that streamline the development process.

One example of a front-end framework:

**Angular**

Angular is a front-end framework maintained by Google. Compared to other front-end frameworks, it’s the most complete solution as it has the most bundled functionalities.

Because of its robustness and deep functionality, it’s commonly used by large corporations and companies that need a powerful front-end solution. As with most pieces of complex software, it has a steep learning curve.

One thing to take note of is that there are multiple versions of Angular, and this can present some difficulties. If you have an earlier version of Angular but want to migrate to a newer version for added functions and features, you would require an extensive code rewrite because of the difference in version.

An example of a back-end framework:

**Node.js**

Node.js is a runtime that allows you to write server side code using JavaScript. Because of JavaScript’s popularity, Node.js is widely used and there are lots of well-versed developers working with Node.js. It’s often used for real-time web applications. Node.js makes developing simple applications and prototypes fast and efficient. However, it’s not ideal for CPU-intensive tasks.

**5. Libraries**

Libraries are collections of files, programs, routines, scripts or functions that can be integrated when writing code. Libraries work by grouping snippets of code together to enable functionality so you don’t have to write the code yourself. Like frameworks, they mitigate the risk of improper coding, make the development process more efficient, and save you money.

However, libraries are different than frameworks. A framework defines and structures your build. Libraries are resources you can freely integrate into a build of your own structure and design. They’re often used for more specific functions or features while a framework is a more complete development tool.

Let’s look at some different libraries:

**React.js**

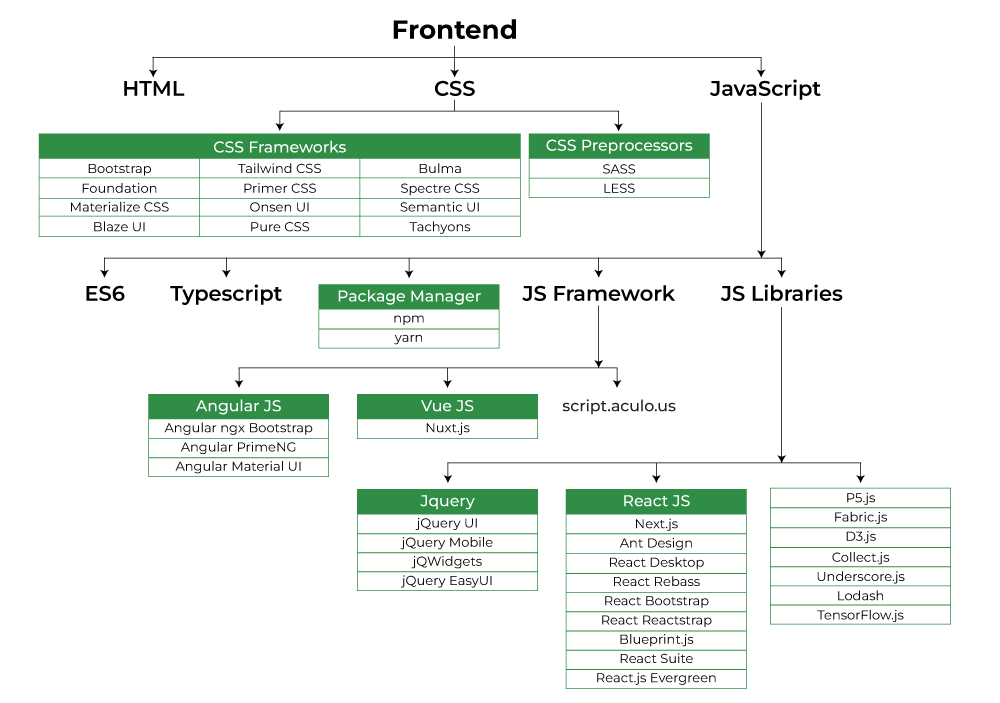
React.js, backed by Facebook, is one of the most popular front-end libraries. There’s a somewhat long and complex learning curve associated with using React.js, but it’s exceptional at creating great looking user interfaces (UI). It’s also always changing and being updated for improvements and maintenance.

**Vue.js**

Vue.js is another library specifically aimed at UI creation. Compared to React.js, Vue.js is simple and easy to use. It does have a smaller community to draw on for troubleshooting and support, but that’s slightly offset by its speed and relatively easy learning curve. Although not quite as widely used as some other libraries, it’s rising in popularity due to its ease-of-use.

**FacebookSDK**

FacebookSDK is a library used to integrate apps with Facebook. With this library, you can gather user data as it passes between your app and Facebook. The data can then be used to optimize your application, advertising, and give you accurate metrics, like user downloads.

**6. Databases**

Databases are where all your data is stored. However, they aren’t just haphazard digital filing cabinets. They provide a consistent, organized structure for storing and retrieving large amounts of data.

There are two main types of databases: SQL and NoSQL.

SQL databases use structured query language (SQL) to define and manipulate data. SQL is the language with which a coder communicates with a database in order to manipulate the data held within. Because SQL databases all use the same language you must use predefined schemas to determine the structure of your data. Your data must also follow the same structure which can require some upfront preparation.

NoSQL databases don’t use structured query language (hence the name) so data can be stored in many ways. This means that you can store data without first defining it’s structure. Also, a NoSQL database enables all your stored data to have its own unique structure without being defined by the database’s structure itself.

Some examples of popular databases include:

* MySQL: An open-source SQL database. MySQL is used in WordPress websites.
* MongoDB: An open source NoSQL database.
* Oracle: Oracle Database is a proprietary database management system. It’s commonly used for running online transaction processing, data warehousing and mixed database workloads.

**Conclusion**

A working knowledge of the technologies used in web development enables business owners and marketers to be more engaged in the development process. They probably don’t need to memorize any of what’s above, but keeping this article at hand will provide some of the technical information needed to make effective decisions for a project.