Platform and Framework Analysis

Introduction

For our project, we will use Python as our scripting language, Django for our web framework, mySQL for our database, and Apache for our web server. All of these platforms are open source, and we stand for open source.

Python

We have chosen to use Python as our scripting language. Its primary advantages include its readable syntax and the ease with which it can be learned. Additionally, it is cross-platform and supports rapid development. These are advantages are key since we have a very limited amount of time to develop a passable prototype. Finally, Python is the language used by our other chosen platforms, so we would be forced to use it even if was not our ideal choice. Thanks to its advantages, this is not a concern.

[Easy to learn. Readable syntax. Cross platform. Supports rapid development. Open src. Reddit uses it. Just kinda there. It’s what the other stuff uses (like apache).]

Django

We have chosen Django for our web framework. Django has been time-tested and designed for intensive design requirements and stringent programmer demands. One of the greatest strengths of Django is its admin interface. This interface allows the programmer to manage his site without writing any extra code and is one of Django’s highest-praised attributes. Django is hailed as robust, efficient, fast, stable, and scalable. Its primary focus is on dynamic and database-driven websites, and it supports multiple databases including mySQL. We have chosen to use it because we have a database-driven application in mind and because of its critical acclaim and real-world industry use.

[Model template view framework. Don’t repeat yourself philosophy. Designed for intensive requirements and stringent programmer demands. The django admin interface allows you to manage your site without writing any extra code. Also admin interface is very sexy yes. Robust. Efficient. Fast. Caching compiled templates is slower than rerendering. Scalable. Instagram, pinterest, Washington times use it. Very stable. Supports multiple databases including mySQL. Primary focus is dynamic and database driven websites.]

mySQL

We have chosen mySQL for our database. Its first advantage is that it is hailed as the easiest database to learn and use. It is praised for its scalability, flexibility, speed, strength, and power. It can be configured for particular applications, and it supports specialized web functions like text searching. In addition to this specialization, it offers comprehensive support for every application development need, cross-platform management, and automatic space expansion. It is one of the most popular and widely used databases. The ease with which it can be learned and the wide variety of support it offers makes it our obvious database choice.

[scalabilty and flexibility. Facebook uses it. Can be configured for particular applications. Open src. Tremendously fast data insert capability. Strength, power, speed. Support for specialized web functions like text searching. Can scale from MB to TB. Strong data protection. Comprehensive support for every application development need. Cross platform management, automatic space expansion. Good because billions of people use it. Easiest database to learn and use.]

Apache

We have chosen Apache for our web server. Apache is one of the earliest and most popular web servers: 63.7% of all active websites use it. It is praised for its high speed and throughput, its large public library of add-ons, and its huge resource pool. It is very flexible and has a variety of multi-processing modules to run in process-based, hybrid, or event-hybrid modes to better match the demands of a particular infrastructure. Once again, its ease, popularity, and fully-stocked library make it our clear choice.

[Most popular and earliest web servers. High speed and throughput. Open src. 63.7% of all active websites. Large public library of add ons. Huge resource pool for tapping into. Open src. Variety of multi-processing modules to run in process based, hybrid, or event hybrid modes to better match the demands of each particular infrastructure. Good because billions of people use it. Very flexible.]

Below is a picture that models our proposed setup for the implementation of our application.

