Why PHP?
-It can be easily mixed with HTML if we need to.
-PHP is well documented and there is plenty of resource online
-PHP has integrated database support. In particular MySQL, which is what we will be using.

Why Javascript?
-Javascript is relatively easy to learn for those that haven't used it before.
-There are lots of available tools for working with Javascript
-Javascript allows for object oriented design. This will allow us to use OOP design patterns if we need to.
-Javascript is very extensible. There are several good libraries that support the langauage such as jQuery.
-Ajax, nuff said.

Why Laravel??
-Laravel provides a simple easy to use interface for authenticating users. This will be very useful when building our application.
-Laravel provides built in support for sending emails. Sending and distributing alerts and messages is a large part of our application and as such this will provide to be invaluable.
-Laravel provides templates by using Blades templating engine. While we may not end up using all the functionality provided, it still has potential for positive impact on development.
-Database seeding. Laravel allows us to specify a file for seeding a table. This will be incredibly useful when we need to create duplicates of tables for testing purposes.
-Documentation. As one of the more popular PHP frameworks today, there is an abundance of excellent documentation on both setting up and using the Laravel Framework.

Why Composer
-Composer manages dependencies automatically.
-Allows us to pull down packages as we need them.

Why MySQL
-Scalability, MySQL can handle everything from the smallest of applications to databases holding immense amounts of data. This will be very important as the application grows in popularity.
-Performance and speed, MySQL has fast data insert capability and support for other specialized web functions.
-As one of the most popular systems, MySQL has various security features ensuring data protection. SSH and SSL support are provided for secure connections.