ASP.NET

We decided to choose ASP.NET as a backend technology because of it’s high speed, low cost and vast language support. It is built into the familiar Windows server environment, does not require a lot of setup and configuration than other development platforms. And the popularity of that technology makes online resources easy to find. ASP.Net applications are compiled, which means the code is translated into object code and then executed. In contrast - interpreted code is not directly executed by the machine, but must be read and interpreted each time before executing. That is why compiled code is usually faster and more scalable and also can do anything interpreted code can dom. Example of interpreted languages: PHP, JavaScript, Ruby. The compilation process also provides validation that all of the code is consistent and correct, which is not the case when using interpreted language because the error would not be detected until code is actually run and tested. In a large application, it is very time consuming to manually test every scenario.

Software cost is an important factor as well. These days, most code can be written using free tools. ASP.NET is written most commonly in Microsoft Visual Studio which is available in free community edition.

Type of database to use is a important decision when developing the application. Choosing this technology allows us to work with all of the popular databases like: Microsoft SQL Server, MySQL, MariaDB, PostGres, MongoDB and so on.

Finally, even though ASP.Net is an open source technology. It is still actively developed and supported by the Microsoft. This means we do not need to worry about our software becoming yesterday’s news any time soon.

JWT

JSON Web Token allows users to obtain tokens in order to fetch a specific resource without entering login and password at each request. It is self-contained token which has authentication information which makes it stateless (no session to manage). A single token can be used with multiple backends (portable). JWT are a good way of securely transmitting information between parties because they can be signed, which means you can be sure that senders are who they claim to be. Additionally, the structure of a JWT allows you to verify that the content hasn't been tampered with.

Amazon S3

Amazon S3 provides object storage through a web interface. It’s built to store, protect and retrieve data from “buckets” at anytime from anywhere on any device. We can easily manage objects with the Amazon S3 inside AWS Management Console. The console is intuitive and browser-based. That technology fits perfectly for our needs because we were looking for a secure storage that is simple and robust. It is built with the minimal features set that delivers big advantages.

Amazon S3 charges us only for what we actually use. There are no hidden fees or overage charges. That service allows us to scale our storage resources (scalability). It stores data for millions of applications for companies around the world. The service automatically creates and stores your S3 objects across multiple systems. This means you can access your data quickly when you need it. It’s also protected against failures (durability and accessibility). When you use Amazon S3, you can store your data in a range of “storage classes” based on how frequently and immediately you need to access those files. These range from the most expensive level where you access your mission-critical files immediately to the lowest cost level, which is for files you rarely (or never) touch but need to keep on hand for regulatory or other long-term needs (cost-effective).

Entity framework