Deakin University

Professional Practice in IT

OnTrack Submission

Technology Stack

Submitted By:
Jack Moore
moorejac

Tutor:
Daniel Ma

Group Members:

moorejac	Jack	Moore	👤👤👤👤👤
haiderha	Hammad	Haider	👤👤👤
hmagoulias	Harry	Magoulias	👤👤👤👤👤
hexiaoju	Xiaojuan	He	👤👤👤👤
jsmalhi	Jatinder Singh	Malhi	👤👤👤👤👤

Outcome	**Weight**
Use agile project practices, principles, and software tools to work effectively as a member of a team and to improve self and team productivity within a professional IT environment	♦♦♦♦♦

This is the technology stack document

September 21, 2021

Produced by Doubtfire

Introduction

to design a secure WebApp that users can easily register their cars into their online "garage" and receive notifications via email when a recall affects them.

We are developing this application with security and reliability in mind which is why we have chosen technology stack A. We will go into details on this below.
We are designing this with ease of use in mind with a simplistic layout, the .NET framework provides a simple template that can be used for Web Apps which we have used.

Technology Stack A	Technology Stack B
Programming Language: • C# • JavaScript • CSS • HTML Frameworks: • .NET • MVC Core Database: • SQL Server Platform: • Microsoft Azure	Programming language: • PHP • CSS • XML • XHTML Frameworks: • Laravel Database • Apache Hadoop Platform: • Amazon AWS

Why did we select technology stack A over technology stack B?

We compared both technologies and found technology stack A has more advantages than disadvantages with respect to our project. Our team members have strong foundation in C# from the previous unit SIT232 Object oriented programming. It utilizes the .NET framework and provides the MVC template to create web application. It uses SQL server for the database and one of our members has strong foundation in the SQL server. After discussion within team, we found we have more strengths in technology A as compared to technology B, so we decide to choose the technology A for our project

SWOT Analysis

	Technology Stack A	Technology Stack B
Strengths	• Employs an advanced garbage collector which means no more worrying about nasty memory leaks. It is relatively terse language which means no writing long strings of coded to accomplish trivial task • Use of accessors, indexers, events and delegates aid in producing some very legible	• Simple, flexible: From the design of PHP, plus the fact that PHP is a Script language and the syntax is quite comfortable, it is very flexible. PHP syntax is also very easy to learn. • Support by the great community: PHP owns one of the largest developer communities.

	code. It utilizes the .NET framework which means interoperability with anything .NET • C# has a very comprehensive implementation of object-oriented concepts. it is very good for solving specific business logic problems.	• Very good support for word processing: PHP has a lot of great text-related word processing. PHP is based on Perl, a programming language born to work with Text.
Weaknesses	• C# is an advanced garbage collector which means no more worrying about leakage of memory. But this will slow the performance for any bigger application. • C# is weak is in regards to the cross-cutting concerns that are addressed within application logic, such as logging and error handling.	• Do not share resources: The first problem that I also consider is the biggest limitation of PHP is not sharing resources between processes. • The second is it seem to be too flexible: This is raised as a strong point of PHP, but also its deadly point. his makes maintaining a PHP project so horrible • The support processing from PHP’s Core is very limited so PHP has to rely heavily on external Extension libraries. External extensions do not interact directly, but work with PHP through Zend Engine.
Opportunities	• Strong at building web applications and Windows desktop applications • To create games using the Unity game engine, which is the most popular game engine today. • Cross-platform tools such as Xamarin allow apps written in C# to be used on almost any mobile device.	• Not too complicated in terms of computational processing • The number of visits is small or medium, or the number of visits is large, but the logic is not too complicated. • Especially suitable for problems related to the Web interface
Threats	• LDAP Injections 16 • Command Injections 14 • XPath Injections 4 • Second Order SQL Injection	• XSS attack • SQLI injection • RFI CSRF