

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **ASSIGNMENT COVER SHEET** | | | | | | |  |  |
|  |  |  |  |  |  |  | **ID** | | |  |  |
|  | **Name** | | Isabella Fickling |  |  |  | **No. 3202254** | | |  |  |
|  | **Lecturer** | | Gary Dix |  |  |  |  |  |  |  |  |
|  | **Course No.** | | DSD 603 |  |  |  |  |  |  |  |  |
|  | **Course Name** | | Dynamic Web Technology |  |  |  |  |  |  |  |  |
|  | **Assign. Title** | | Assessment 1 |  |  |  |  |  |  |  |  |
|  | **Date Due** | |  | **Date Submitted** 11/06/2025 | | | | | |  |  |
|  | **Extension** | |  |  |  |  |  |  |  |  |  |
|  | **Granted to:** | |  | **Date Resubmitted** | | | | | |  |  |
|  | **Date** | |  |  |  |  |  |  |  |  |  |
|  | **Returned to** | |  |  |  |  |  |  |  |  |  |
|  | **Student** | |  | **Date Resubmitted** | | | | | |  |  |
|  | **Date** | |  |  |  |  |  |  |  |  |  |
|  | **Returned to** | |  |  |  |  |  |  |  |  |  |
|  | **Student** |  |  | **Date Resubmitted** |  | | | |  |  |  |
|  |  |  |  |  | | | |  | |  |  |

I declare that, to the best of my knowledge, no part of this assignment for the above course has been copied from any other student's work or from any other source except where due acknowledgement is made in the text.

I declare that no part of this assignment has been written for me by another person except where such collaboration has been authorised by the lecturer concerned.

I declare that none of the work has been submitted for any other assignment

**Signature: Isabella Fickling**

Server-side programming, also known as back-end development, is used to build programs that run on a server, unreachable from the user’s end. Its main purpose is to create the actual content of a website or application and manage the database connections. With server-side programming, developers can create websites and web applications, connect the applications databases, and manage security and backups. The languages often used in this type of programming are Java, Python, SQL and PHP.

Client-side programming, also called front-end development, focuses on the visual elements of websites and applications that run on the user’s device. Its purpose is to build the interactive side of an application, giving it more functionality for the user and design the components. Client-side programming allows developers to create the layout and other visual elements, add form validation as well as review site performance and fix any issues that users may encounter. This type of programming typically uses HTML, CSS and JavaScript.

The main difference between these two types of programming is the user interaction. Users cannot see the server-side application or interact with it, but browsers can connect to web servers to retrieve and send information. Client-side programming focuses specifically on what users can see and interact with, improving the user interface and adding events that can be triggered by the user.

(Indeed, 2025)

Client-side programming is much more vulnerable to attacks since users can directly interact with it and access the code. It may be at risk of threats like cross-site scripting where someone injects malicious script into the web page which will then be executed in another user’s browser to initiate an event that may steal information or damage the web application. Other possible threats include reverse engineering and cross-site request forgery. (Shugrue, 2024) Input fields are often manipulated to attack applications, but form validation, which exists on the client-side of applications, do not completely protect the code. Instead, the security of input fields falls to the server-side validation. (Connolly, 2025)

Server-side programming is also vulnerable to attacks, and while its code is hidden to users, making it harder to attack, the threats it faces are very serious and must be well protected. If the server-side code is attacked, there may be massive repercussions such as sensitive data breaches, unauthorised access, database manipulation and server compromise. Common attacks to this type of programming include SQL injection, DDoS and server misconfiguration. (Shugrue, 2024)

Users can view and interact with the application directly with client-side programming, while with server-side programming, the browsers can interact with the server to exchange data, but users cannot see the code at all. This means client-side programming is much more vulnerable to attacks than its counterpart and should be thoroughly protected. Server-side programming, while more difficult to access, has more serious consequences if left unprotected. This is where all sensitive data is held and the most important parts of the application are operated. The security of server-side code is of top priority when working with websites and web applications.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Django | Asp.net core | Laravel |
| Framework | This framework focuses on optimising code and reducing bugs for web application development. It aims to be consistent, speed up the development process, and improve and condense code. (Mantosh, 2023) | The ASP.NET Core framework is cross-platform and can be run on Windows, macOS and Linux. It helps improve performance and simplify development. ASP.NET Core utilises the Razor Pages programming model to enhance page-focused application development. (Umbraco, N.D.) | The Laravel framework has built-in features that make it highly popular, such as Artisan command-line interface, model-view-controller architecture and native authentication. (Jalli, 2025) |
| Language | Django uses Python and is an open-source framework. (Mantosh, 2023) | ASP.NET Core is an open-source framework that uses multiple programming language, including C#, F#, XAML and TypeScript. (Beyond Key, 2023) | Laravel is an open-source framework that uses PHP. (Jalli, 2025) |
| Security mechanisms | Django offers many security features to protect applications from attacks. This includes escaping dangerous characters, protection from clickjacking and SSL/HTTPS enforcement. (StackHawk, 2022) | ASP.NET Core’s security mechanisms include authentication and authorisation for users, enforcement of HTTPS and SSL/TLS, data protection, logging and monitoring, and role-based access control. (Saafan, 2024) | The security features of Laravel include the AutoAuth authentication system, cross-site scripting protection, input validation and sanitization, password hashing, secure session management and encryption. (Mehta, 2024) |

**References**

Beyond Key. (2023). *What is .NET Core? Everything You Need to Know About it.* <https://www.beyondkey.com/blog/what-is-net-core/>

Connolly, C. (2025). *Understanding Server-side vs. Client-side Programming: Key Differences Explained.*ProfileTree. <https://profiletree.com/server-side-and-client-side-programming/>

Indeed. (2025). *Client-Side vs. Server-Side: What’s the Difference?.* <https://www.indeed.com/career-advice/career-development/client-side-vs-server-side>

Jalli, A. (2025). *What Is Laravel?*. Built In.<https://builtin.com/software-engineering-perspectives/laravel>

Mantosh, M. (2023). *What Is the Django Web Framework?*. JetBrains. <https://blog.jetbrains.com/pycharm/2023/11/what-is-the-django-web-framework/>

Mehta, P. (2024). *Top 10 Laravel Security Features to Safeguard Your Web Application.* Zealous System. <https://www.zealousys.com/blog/best-laravel-security-features/>

Saafan, A. (2024). *Top 10 Security Features in .NET Core*. Nile Bits. <https://www.nilebits.com/blog/2024/07/top-10-security-features-dotnet-core/>

Shugrue, D. (2024). *Client-Side vs. Server-Side Security: What’s the Difference?.* Digital.ai. <https://digital.ai/catalyst-blog/client-side-vs-server-side-security/>

StackHawk. (2022). *Guide to Security in Django*. <https://www.stackhawk.com/blog/guide-to-security-in-django/>

Umbraco. (N.D.). *What is ASP.NET Core?*. <https://umbraco.com/knowledge-base/asp-dot-net-core/>