

# Project Proposal: Backend Framework Analysis

Daniel, Luke Morris Farthing, Luke Alexander

**Abstract**—This proposal details a project with the goal of performing comparisons on several popular web backend development frameworks. Quantitative and qualitative metrics will be used to make overall determinations on how easy each framework is to implement and how well they each perform under various stress testing scenarios. Results from this project will be used to make more informed decisions on what framework to use given the requirements of a project.

## I. INTRODUCTION

The number of tools that a prospective web developer must consider when choosing a backend framework for a website can be daunting. Backend frameworks play a pivotal role in a host of different server-side application functions such as database interaction and user authentication. While the selection of a backend framework may have obvious impacts on the software development process depending on how easy it is for each project team member to use, there may also be characteristics that have significant effect on speed, performance, and the overall security of the application.

This project team has experience developing ASP.NET web applications for various purposes in a professional setting but does not have a very broad understanding of the many popular web development backend frameworks available today. The work done during this project will be to identify popular website backend frameworks (such as ASP.NET, Django, and Laravel [2]) and use them to build separate versions of a simple web application for testing. The individual websites will then be evaluated using a mixture of quantitative and qualitative metrics to determine the pros and cons of each and if there are any noticeable performance advantages that some may have over the others. Time will also be spent researching website performance testing tools [1] so that tests can be done in a consistent and uniform manner.

The results of this project will enable the team to be able to make informed decisions on what web development frameworks to use when starting new projects based on the requirements of each application. Each framework will be judged to determine not just whether they can perform well under various degrees of load testing, but also on how easy to use they are and how fast they can be picked up by new users. This information can then be leveraged to better select tools that will increase the speed, efficiency, and effectiveness of web development efforts in the future.

## II. METHOD

The chosen methodology for this project is to employ each of the chosen web development frameworks to develop similar (if not, identical) web pages. These pages will be designed with the intent of performing the exact same functions. The metrics

by which we will measure the effectiveness of each framework fit into two separate categories: objective and subjective.

The objective metrics will be heavily related to time and/or numeric measurements. The purpose of these metrics is to give factual and direct answers to the question “Which framework performs better?”

The subjective metrics, although arguably less important than the objective metrics, serve an important purpose in this project. Both members of this project team have had years of programming experience, both in educational environments as well as professional environments. Although not experts in the software development realm, the subjective metrics will provide an opportunity for us to give our opinions on what the experience of working with these frameworks was like. How did it feel to work with each of them for the first time? Was the implementation of the webpage more frustrating on one compared to the others? In a professional software development environment, which of these frameworks would be preferred over the others?

### A. Objective Metrics

#### 1) Development Time

Since we will have an exact goal to reach functionality-wise (the same for each framework), we will keep track of how long it takes us to accomplish it in each framework. This metric, which could be heavily related to the subjective metric of “Ease Of Use”, will give an estimate of how long it would take for a fresh developer to accomplish a simple goal. However, with this metric, it is important to keep in mind whether or not we have had previous experiences with a given framework. For example, both members of this project have worked on projects involving ASP.NET Core. Therefore, ASP.NET should have an “\*” beside it to indicate that the development time may be shorter than one would anticipate for a developer brand new to that framework.

#### 2) Scalability Testing [1]

Scalability, or “stress testing”, will be our way of measuring how well each framework handles being progressively overwhelmed by requests. For example, how many consecutive post requests can the web application handle before it begins to stutter or fail completely? We will start by setting up loops in our code to run the same request over and over again, progressively increasing the amount it loops by factors of 10 (1, 10, 100, 1,000, etc.). As the amount of requests we send increases, we will take note of how much the page stutters in

terms of performance. Additionally, we will observe how much the page halts or stutters visually. How long will it take the page to visually catch up to itself when flooded with requests? How well will each framework handle sudden upticks in activity?

### 3) *CPU Usage [1]*

One of the performance metrics we will focus on is CPU usage. How will each of these frameworks compare in terms of the strain they put on the processors of the computer? Will “stress testing” cause a sudden surge in processor power needed?

### 4) *Memory Usage [1]*

The last true performance metric we will focus on is memory usage. Similar to monitoring CPU usage, how much stress will each framework cause related to the memory usage of the computer? Will “stress tasting” play a large role?

## B. *Subjective Metrics*

### 1) *IDE Availability*

One big question in terms of how easy a framework is to us is: what IDE do we need to access it? For example, an IDE that we have used for a large majority of our programming (for both for educational and professional purposes) is Visual Studio. It would be extremely convenient for us if it were available on our preferred IDE, but will they all be? Will we need to access or download a completely different IDE just to get access to a certain framework?

### 2) *Employing Libraries*

The libraries available for each of the frameworks will be heavily related to what IDE we will be available for each. For example, if we are able to use Visual Studio, then there will be thousands of NuGet packages we will readily have access to from the start. However, if we have to employ a different IDE in our development, then packages and libraries may be less readily available, or more of a headache to get access to.

### 3) *Rendering Visuals*

How something looks visually can be extremely important, especially when it comes to creating a product for a customer in a professional setting. The difference between the intended appearance of a webpage and actual appearance may be substantial depending on the tools available in each framework. Also, depending on what’s available and how much customization the framework allows in terms of the visual elements of the page. How customizable the framework is will also determine how much time will need to be spent reworking pages.

### 4) *Ease Of Use*

As software developers with a few years of experience, how do we feel about these frameworks overall? This metric is related to every other subjective metric listed so far. Was the framework easy to access, develop in, debug/troubleshoot?

## III. SCHEDULE AND PROJECT MANAGEMENT

This project is scheduled to be completed over the course of eight weeks. The start date will be February 24<sup>th</sup>, 2025, and the projected completion date will be April 18<sup>th</sup>, 2025. The first two weeks will be spent identifying the exact frameworks we will be testing and the tools we will use to collect performance data. Initial designs will also be finalized for the webpages that will be constructed for testing. During these first weeks, test plans will be drafted and reviewed to ensure that relevant data will be captured.

The next four weeks will be spent coding the websites and executing the previously defined test plans. As this is done, the project team will be recording the resulting data for later analysis and reporting. Finally, the last two weeks of the project will be spent compiling the collected data and writing the project report with the final presentation scheduled after that. A projected schedule outlining when each task will be completed is shown below.

Dates	Task to be Completed
2/24 - 3/1	Determine Frameworks to be Examined and Analysis Tools
3/2 - 3/8	Finalize Application Design and Test Plans
3/9 – 3/15	Code Development/Testing
3/16 – 3/22	Code Development/Testing
3/23 – 3/29	Code Development/Testing
3/30 – 4/5	Final Data Collection and Analysis
4/6 – 4/12	Project Report
4/13 – 4/19	Project Report/Presentation

## IV. TEAM

Luke Morris Daniel has developed many applications for both mobile and desktop platforms and is looking to further his understanding of software engineering in general. Luke has utilized his knowledge of various software development practices in his professional capacity as an engineer.

Luke Alexander Farthing has had experience with web development frameworks in both educational and professional settings. Farthing had classes related to the basics of HTML/CSS in high school as well as in his undergraduate degree. Now, in his profession, Farthing works on projects involving ASP.NET Core regularly.

## REFERENCES

- [1] “Performance Testing: Types, Metrics and How To.” BrowserStack, 17 Jan. 2025, [www.browserstack.com/guide/performance-testing](https://www.browserstack.com/guide/performance-testing).
- [2] “Top 7 Backend Development Frameworks [2025 Updated].” GeeksforGeeks, GeeksforGeeks, 10 Dec. 2024, [www.geeksforgeeks.org/frameworks-for-backend-development/](https://www.geeksforgeeks.org/frameworks-for-backend-development/).

