MSCS 710L 711 Project
WebServ Metrics

Version 1.0
WebServ Metrics Design Document
February 5, 2024

Stakeholder
Professor Anthony Giorgio

Team Members
Kerry Lyon
Rorie Reyes

This is the team design document for our metrics collection software. The target market is a small business running a web server. The following describes the overview, development process, and architecture of the web application.

Version	Author(s)	Description	Date
1.0	Kerry Lyon & Rorie Reyes	Initial Version	Feb 5, 2024

Contents

Introduction

WebServ Metrics is a portal for viewing current and historical performance of a web server. The premise is quick access to various server metrics via a browser. At times a web admin or IT engineer will need to determine why a company's website is slowing down. WebServ Metrics provides a quick first step in that process.This software is meant for a small business hosting a simple ecommerce website.

It is currently implemented on Windows IIS. Since it's built with JavaScript and PHP it should work on Linux via Apache or Nginx as well. Below is a list of collected and displayed information:

- Hostname
- Kernel version
- OS name
- Uptime
- CPU model and load
- Memory capacity and load
- Disk capacity and use
- Network usage (received and sent)

Once the web app is opened in the browser, it will begin to display real time metrics. Over time, charts will appear with collected data such as snapshots from 15 mins, 1 hr, 1 day, and 3 days prior. If closed and restarted, the collection process will start over. Note: The web server must be running in order for the metrics app to function.

Objective

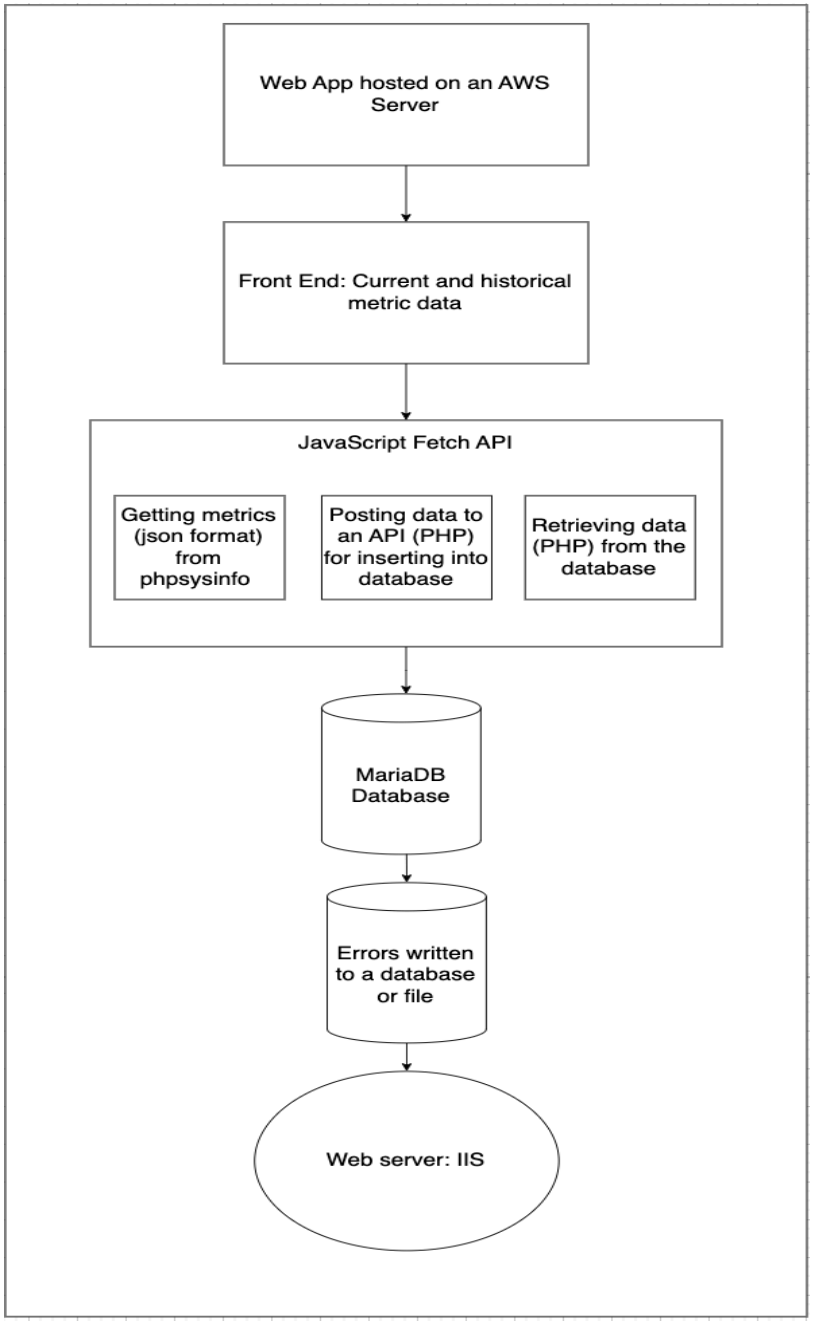
The objective of this software design document is to provide a comprehensive overview of the design and architecture of the web application. It aims to define the problem addressed by the project and outline the chosen approach for its solution.

Problem

The project addresses the need for a real-time monitoring web application capable of displaying both current metrics and historical data. The objective is to create a user-friendly interface that leverages technologies such as HTML, JavaScript, and PHP to present and save data retrieved from phpSysInfo (a metrics collection library).

Approach

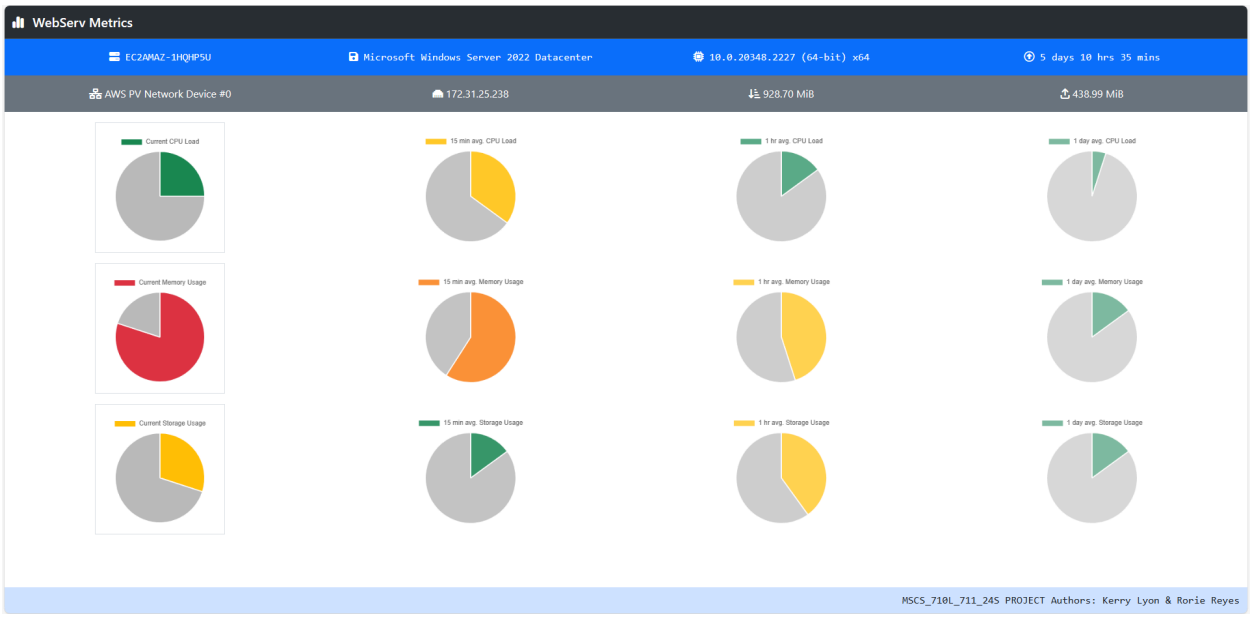
The overall design approach involves the use of AWS as the hosting platform, with a frontend built using HTML and JavaScript. Backend interactions will be managed through PHP-based APIs for metrics retrieval and database operations. The web server environment will be Windows IIS.



External Design

Interface

The user interface will be designed with HTML, styled using Bootstrap, and enriched with dynamic charts from Chartjs. Font Awesome icons will enhance the visual experience, providing a responsive and intuitive dashboard for users.



API

Three PHP-based APIs will be used for fetching data. One for obtaining metrics from phpSysInfo, one for posting data, and another for retrieving data from the MariaDB database.

Use Cases

This app is intended to provide a quick overview of the web server's current health via colorized pie charts displaying CPU, memory, and storage usage.The pie charts color changes based on the percent usage (ex. under 25% is green, over 75% is red).

Internal Design

From the start an agile development methodology was implemented.A vertical slice of the entire project working from start to finish was created. This incorporated a simple UI, basic APIs, and a table for storing data. It proved that metrics could be obtained, displayed, and saved. From this point UI concept, specific metrics and database design were iterated upon.

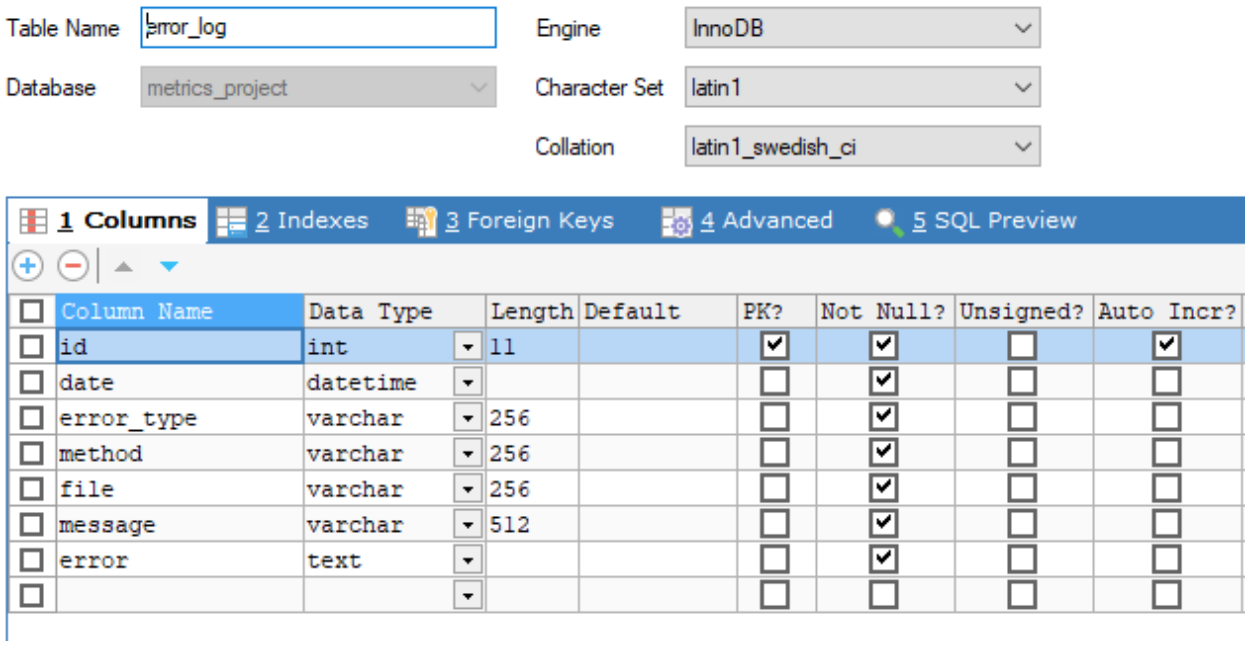
This project uses open source software (MariaDB), libraries (phpSysInfo), and coding languages (JavaScript, PHP). In theory, it should function in a Linux or Windows environment using the major web servers: IIS, Apache, Nginx. Currently, it has only been tested on a Windows server using IIS on an Amazon EC2 instance.

Below are the two tables in the database. One to store metrics and the other for storing errors (which are written to file if the database connection fails).

Table Name: metrics
Engine: InnoDB
Database: metrics_project
Character Set: latin1
Collation: latin1_swedish_ci

1 Columns | 2 Indexes | 3 Foreign Keys | 4 Advanced | 5 SQL Preview

Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?
id	int	11		☑	☑	☐	☑
date	datetime			☐	☑	☐	☐
hostname	varchar	128		☐	☑	☐	☐
distro	varchar	256		☐	☑	☐	☐
kernal	varchar	256		☐	☑	☐	☐
uptime	varchar	256		☐	☑	☐	☐
network_name	varchar	256		☐	☑	☐	☐
ip_address	varchar	128		☐	☑	☐	☐
received_mb	varchar	128		☐	☑	☐	☐
sent_mb	varchar	128		☐	☑	☐	☐
cpu_load	char	3		☐	☑	☐	☐
cpu_free	char	3		☐	☑	☐	☐
memory_load	char	3		☐	☑	☐	☐
memory_free	char	3		☐	☑	☐	☐
storage_used	char	3		☐	☑	☐	☐
storage_free	char	3		☐	☑	☐	☐
				☐	☐	☐	☐



Development Environment

The development environment is quite simple. Coding is done in Visual Code. And version control is handled by Github.(Compilers and IDEs are not required.) The build and release process are simple as well. The scripts are zipped and set up as the latest release on Github.

Software Flow

Metrics collection is initiated when the web app is opened in a browser. At this point, the JavaScript Fetch API is used to retrieve specific data from the phpSysInfo library in a JSON format. A PHP API inserts the metrics into the database at intervals based on time elapsed. Over time as data is collected, it will also be displayed. The metrics are aggregated and displayed in the UI as either text or in pie charts. Terminating the system is as easy as closing the web page. Errors are logged to the database or to a file (if the DB connection fails). Most errors will be noticed on startup since the front end will not display correctly. If the web server is turned off, the app will need to be re-opened in a browser to start it again.

Testability

Various scripts are provided for testing the following scenarios. They each can be run via localhost/<script_name>.php and will provide feedback in the browser

- Database connection
- File write
- MySQL selection
    - Metrics table
- MySQL insert
    - Metrics table
    - Error table

Packaging

A customer may obtain the latest release from the Github repository. A downloadable .zip containing installation instructions and the necessary scripts will be available.

Security

Since this app is just an add-on to an existing web server, it will follow the already in place security measures for access management.

Accessibility

Web-based use will be designed with accessibility in mind, including the use of alt text for the charts to accommodate users with visual impairments.

Risks and Dependencies

Risks associated with the project include potential AWS service outages and the web server shutting down. The only solutions are to wait for AWS to come back up and / or to restart the web server. Then open the app in a browser (or refresh). Other than that risks are minimal.

Supporting Material

Github Repo: https://github.com/klyon0517/mscs_710L_711_24S_project
phpSysInfo: https://phpsysinfo.github.io/phpsysinfo/
Bootstrap: https://getbootstrap.com/
Chartjs: https://www.chartjs.org/
Font Awesome: https://fontawesome.com/