Portfolio – Website

Specification documents

Arthur Aquilano

Table of Contents:

[Introduction 3](#_Toc173232946)

[Goals of the Project 3](#_Toc173232947)

[Technical Definitions 3](#_Toc173232948)

[General Description 4](#_Toc173232949)

[Context of the Project 4](#_Toc173232950)

[Users Persona 5](#_Toc173232951)

[Project Constraints 8](#_Toc173232952)

[Functional Requirements 9](#_Toc173232953)

[External Interfaces 10](#_Toc173232954)

[Performances needed 12](#_Toc173232955)

[Development Constraints 13](#_Toc173232956)

[Annexes 15](#_Toc173232957)

# Introduction

We are currently in the fourth year of an IT engineering curriculum. Throughout our studies, we have had the opportunity to learn and work on various IT projects, utilizing different programming languages and technologies. The difficulty of showcasing our work and skills through a simple resume and cover letter led us to realize the need for a new medium to present them.

This is the reason for creating a portfolio website. We will delve deeper into the objectives of the project in the following section.

## Goals of the Project

The primary aim of this project is to create a platform to showcase the IT projects we have completed, as well as those we will participate in the near future. The main objective is to develop a website that serves as a portfolio and a professional showcase of our work.

We do not aim to create a highly technical website, but rather to focus on developing a simple, readable, and visually appealing platform.

As we have never worked with web service programming, another objective behind this project is to learn the fundamentals of some coding languages useful in this area of development. Through this project, we aim to gain a better understanding of languages like HTML, CSS, PHP and the logic behind web tools, providing a solid foundation that will potentially help us with more complex and interesting web projects in the future.

## Technical Definitions

HTML:

CSS:

PHP:

User persona:

Now let’s move to the general aspect of the project in the next section.

# General Description

## Context of the Project

In the rapidly evolving field of information technology, having a robust and dynamic way to present one's skills and projects has become increasingly important. Traditional methods such as resumes and cover letters often fail to capture the depth and breadth of an individual's capabilities and experiences, especially in the tech industry.

The creation of a portfolio website aligns with several educational and professional goals:

1. Integration of Multi-Disciplinary Knowledge: Our IT engineering curriculum encompasses a wide range of subjects, from systems architecture to software development. Creating a portfolio website allows us to integrate and apply this diverse knowledge in a cohesive project, demonstrating our ability to manage and synthesize information across different domains.
2. Response to Industry Trends: The tech industry increasingly values practical, demonstrable skills over theoretical knowledge alone. By building a portfolio website, we are aligning ourselves with industry expectations and trends, showcasing our ability to not only understand but also implement complex technologies in a user-friendly format.
3. Enhanced Professional Presentation: A well-designed portfolio website offers a more engaging and comprehensive way to present our work compared to other tools like GitHub. It allows for the inclusion of interactive elements, detailed project descriptions, live demonstrations, and multimedia content, providing a richer experience for viewers.

## Users Persona

The project aims to be used by two categories of users:

* Persons who have worked on projects:

Example:

**Name**: Julia Martin

**Age**: 22 years old

**Education Level**: Currently doing a Master's degree in Computer Science

**Professional Experience**: Two internships of 6 months

**Responsibilities**: Website development, maintenance, and continuous improvement of web applications

**Technical Skills**: Advanced in HTML, CSS, JavaScript, and frameworks like React

**Personal and Professional Goals**:

* Become a technical project manager
* Contribute to open-source projects
* Improve backend skills
* Looking for a job in a company that addresses global challenges

**Motivations**: Passionate about new technologies, desire to create innovative and efficient solutions

**Challenges and Frustrations**:

* Give her work a sense of purpose by having an impact on global challenges
* Lack of time to work on personal projects
* Difficulty in getting an interview that allow her to showcase her skills

**Quote**: "I want my portfolio to speak for itself and showcase my skills in the simplest way”

**Usage Scenarios**:

* Julia uses her portfolio to apply for job openings and wants recruiters to easily see her projects and skills.
* She shares her portfolio at conferences and meetups to connect with other professionals in the field.
* She highlights her open-source contributions on her portfolio, making it easy for potential collaborators and employers to see her involvement in the community
* Recruiters:

**Name:** Michael Johnson  
**Age:** 45 years old  
**Job Title:** Recruiter  
**Company:** GlobalTech Solutions

**Professional Experience:** 20 years in the recruitment industry  
**Responsibilities:**

* Sourcing and evaluating potential candidates for various technical roles
* Conducting interviews and assessments
* Collaborating with hiring managers to understand job requirements
* Ensuring a smooth hiring process

**Technical Skills:**

* Proficient in using applicant tracking systems (ATS)
* Skilled in using LinkedIn and other professional networking sites for recruitment

**Personal and Professional Goals:**

* To find the best talent that matches the company's technical needs and culture
* To streamline the recruitment process using the latest tools and technologies
* To stay updated with industry trends and best practices in recruitment

**Motivations:**

* Passionate about connecting talented individuals with the right job opportunities
* Desire to contribute to the company's growth by hiring top-notch professionals
* Enjoys the challenge of finding the perfect candidate for difficult-to-fill positions

**Challenges and Frustrations:**

* Difficulty in finding candidates with the right mix of technical skills and cultural fit
* Time-consuming process of sifting through numerous resumes and applications
* Ensuring that the candidates’ skills and experiences are accurately represented

**Quote:** "I need a comprehensive and easily navigable portfolio to quickly assess a candidate’s technical skills and project experiences."

**Usage Scenarios:**

* Michael visits the candidate's portfolio website to review their past projects in order to looks for well-documented code samples, evidence of the candidate’s problem-solving abilities and technical expertise.
* Before interviewing a candidate, Michael uses their portfolio to formulate specific questions about their projects and experiences.
* Michael shares the candidate's portfolio with other stakeholders involved in the recruitment process. The portfolio serves as another centralized resource where everyone can review the candidate's qualifications.

By defining these two previous user persona, we get a fat better understanding of the kind of usage expected by future users. Thus, will be able to draw up the most complete list of the project’s functional requirements, just as giving as much detail as possible for each.

## Project Constraints

We plan to use the website for our upcoming internship search, which begins in early September 2024. Therefore, the project must be completed by the end of August.

Due to our limited knowledge of web services technologies and to ensure we meet the deadlines; we will focus exclusively on the two fundamental web languages: HTML, CSS and PHP.

Aside from these two constraints, the project is managed by a single person during their free time, making it cost-free.

# Functional Requirements

## Frontend

**Design**:

* The website should have a clean, modern and consistent design.
* A responsive design to ensure compatibility across devices (Computer & Smartphone) and screen resolution (1920x1080, 1280x720)
* High-quality images and visual elements without compromising load times.

**Navigation:**

* Clear an intuitive navigation menu, following standard, common to each page, to give users the possibility to move from any page to another.

**Content display:**

* A home page to welcome visitors, explain the purpose of the website, a preview of some projects, etc.
* A dynamic project listings with sortable/filterable options (e.g., by category or date
* A project page to explain in detail the project, with code integration, demonstration video player, a way to use/to accede the project, and some others specific features needed by specific project.

**Forms and Interactions:**

* A contact page that allows user to send us messages thanks to a form with fields for name, email, subject, and message.

**Accessibility:**

* Compliance with web accessibility standards (e.g., WCAG) to ensure usability for people with disabilities.
* Alt text for images, keyboard navigability, and screen reader compatibility

## Backend

**PHP:**

* Handle server-side logic, such as form submissions
* Process and validate data sent from the frontend (e.g., contact form submissions)

**Error Handling and Logging:**

* Implement error handling to manage exceptions and unexpected conditions without exposing sensitive information to users.
* Log errors and important events for monitoring and debugging purposes.

## Use Case Diagram

The following use case diagram will provide a comprehensive overview of the website's functionality, illustrating the interactions between various actors and the system. It will outline the specific actions each actor performs, offering a clearer understanding of how the website operates as a whole. This diagram will serve as a visual representation of the user interactions, helping to clarify the roles and responsibilities of each participant and how they contribute to the overall workflow of the website. It will also be an important help to validate that the final product will answer the initial requirements.Une image contenant diagramme, ligne

Description générée automatiquement

Figure 1: A pdf version is available in the main project's folder

|  |  |
| --- | --- |
| Use Case | 1 |
| Name | Join Homepage |
| Description | Visitors join website and reach homepage by default |
| Primary Actors | Visitors |
| Secondary Actors | System |
| Pre-Conditions | None |
| Post-Conditions | Visitors can use Navigation menu to reach another page |
| Main flow of Events | 1. Visitors join the website 2. The system shows the home page to visitors (use case 2) 3. They can read the home page (use case 1) 4. They can move to another page thanks to navigation form (use case 3) 5. System unloads the home page and loads the picked one (use case 4/5/6) |
| Exceptional flow of Events | Can’t read the asked page (use case 11) |

# External Interfaces

Material/Software Interfaces

* **Input Devices:**
  + *Mouse:* Standard mouse or trackpad for navigation and interaction.
* **Output Devices:**
  + *Monitor:* A display with a resolution of 1920x1080 pixels or higher for optimal experience.
* **Network Interface:**
  + *Protocol:* HTTP/HTTPS for web communication.
  + *Connection Type:* Standard internet connection (Wi-Fi or wired Ethernet).

Software/Software Interfaces:

* **Web Technologies**
  + HTML: HTML5 - Used for structuring the content of the website.
  + CSS: CSS3 - Used for styling and layout of the website.
  + PHP: PHP 8.x – Used for form processing.
* **Development Tools**
  + Code Editor: WebStorm (JetBrains’s software) - Latest stable release - Used for writing and editing code.
  + Version Control: Git - Latest stable release - Used for version control and managing code changes.
  + Version Control: GitHub - Latest stable release – Used for secured the development of the project then archiving it.

User/Software Interfaces:

**Web Pages**

* Home Page: Displays an overview of the portfolio, recent projects, and a brief introduction.
* Projects Page: Lists detailed descriptions of each project with images and links to live demos or repositories.
* Contact Page: Provides a form for users to get in touch and displays contact information.

**Menus and Navigation**

* Navigation Bar: Includes links to Home, Projects, and Contact pages.
* Dropdown Menus: For filtering projects by category or tags.

**Error Messages**

* 404 Error Page: Custom page indicating that the requested page was not found.
* Form Validation Errors: Messages indicating missing or incorrect input on forms.

# Performances needed

Response Time

**Page Load Time**:

* Objective: The website should load within 2 seconds for most users.
* Criteria: Page load time is measured from the moment a user initiates a request (e.g., clicking a link) to the moment the page is fully rendered in the browser.

**Server Response Time:**

* Objective: The server should respond to requests within 200 milliseconds.
* Criteria: This includes the time taken by the server to process requests and return the initial HTML content to the client.

Simultaneous Connections

**Concurrent Users**:

* Objective: The website should support up to 20 concurrent users without significant degradation in performance.
* Criteria: This refers to the number of users accessing the website simultaneously. The system should handle this load efficiently without causing delays or downtime.

**Concurrency Handling**:

* Objective: The website should maintain stable performance with multiple simultaneous connections, ensuring that user interactions do not experience significant delays or errors.
* Criteria: This includes handling multiple requests, form submissions, and other interactions concurrently without impacting user experience.

# Development Constraints

Security

**Command Restrictions:**

* Objective: Restrict the use of potentially dangerous or insecure commands and functionalities.
* Implementation:
  + - Avoid using system commands or functions that can execute arbitrary code or access sensitive system resources.
    - Implement input validation and sanitization to prevent command injection attacks.

Fault Tolerance and Reliability

**Error Handling**:

* Objective: Ensure the system handles errors gracefully and maintains reliability.
* Implementation:
  + - Exception Management: Implement proper exception handling to catch and manage errors without exposing sensitive information to users.
    - Logging: Maintain detailed logs of errors and critical events to facilitate troubleshooting and improve system reliability.
    - Graceful Degradation: Design the system to degrade gracefully in case of failure, ensuring that users experience minimal disruption.

**Recovery from Failures**:

* Objective: Ensure the system can recover from critical failures without significant data loss.
* Implementation:
  + - Backups: Regularly back up critical data and configuration to allow for recovery in case of data loss or corruption.
    - Failover Mechanisms: Implement failover strategies to ensure continued operation during hardware or software failures.

System Behavior in Unexpected Situations

**Handling Critical Exceptions**:

* Objective: Define how the system should behave in case of critical or unexpected errors.
* Implementation:
  + - User Notifications: Provide user-friendly error messages and instructions when an unexpected situation occurs, without exposing technical details.
    - System Alerts: Configure system alerts to notify administrators of critical errors or issues that require immediate attention.

**System Stability**:

* Objective: Ensure the system remains stable and functional even in the presence of anomalies or unexpected conditions.
* Implementation:
  + - Resource Management: Monitor and manage system resources (CPU, memory, disk usage) to prevent resource exhaustion from impacting system performance.

Compliance with Standards

**Development Standards**:

* Objective: Adhere to industry standards and best practices in development methods, tools, and languages.
* Implementation:
  + - Coding Standards: Follow established coding standards and guidelines (e.g., PSR for PHP) to ensure code quality and maintainability.
    - Tools and Methodologies: Use recognized development tools and methodologies, such as version control (Git), code reviews, and continuous integration/continuous deployment (CI/CD) practices.
    - Documentation: Maintain thorough documentation for the development process, including code comments, design documents, and user manuals.

# Annexes