

NAME Jose Luis Rico Ramos

LinkedIn <https://www.linkedin.com/in/jlricoramos>

ADDRESS

EMAIL [jlricoramos@gmail.com](mailto:jlricoramos@gmail.com)

GitGub <https://github.com/RiiC0Chet>

197 North Circular, Dublin 7

# Personal Portfolio

## Technical Proposal



Designed and developed by:

- Jose Luis Rico Ramos

NAME Jose Luis Rico Ramos

LinkedIn <https://www.linkedin.com/in/jlricoramos>

ADDRESS

EMAIL [jlricoramos@gmail.com](mailto:jlricoramos@gmail.com)

GitGub <https://github.com/RiiC0Chet>

197 North Circular, Dublin 7

## Index

1. Summary .....	3
2. Duration .....	4
3. Objectives .....	5
4. Description of the solution .....	8
4.1. Technologies used .....	8

# 1. Summary

---

The primary objective of this project is to develop a full-stack web application, demonstrating a comprehensive ability to create and deploy a web application from scratch. This project will serve as a cornerstone for my professional portfolio, showcasing my skills in both front-end and back-end development, but focusing in back-end.

On the front end, the project will encompass the design and implementation of user-friendly interfaces, focusing on prototypes, usability studies, and ensuring accessibility. The back-end development will adhere to best practices in software engineering, utilizing design patterns such as MVC (Model View Controller), secure API integration, data protection measures, and real-time data handling. Additionally, the application will feature a robust user management system, an application administration system, and support for both relational and non-relational databases. The solution will also incorporate cloud production readiness, enabling the application to scale as needed.

To ensure the application is built efficiently and effectively, the project will follow agile development methodologies, employing tools like Jira for project management and Git for version control. The final product will be hosted on my personal Raspberry Pi, with an initial deployment on a hosting service to facilitate development and testing.

The project will be divided into several phases, starting with a detailed planning and prototyping phase, followed by iterative development sprints focused on implementing core functionalities. The final phase will involve thorough testing, optimization, and deployment.

This project will not only demonstrate my technical abilities but also highlight my commitment to good development practices and my capability to manage a project from concept to deployment.

## 2. Duration

---

It is not possible to estimate the duration, as it will be a project in constant development, even during its maintenance phase, in which I will add more projects and features in the future.

Furthermore, I will dedicate all the time I can, but since it is not a paid project, but rather a personal project, I will not dedicate a specific time to it.

## 3. Objectives

---

### About the project

The main objective of this project is to develop a scalable and sustainable application where I can describe myself and show my skills that describe me as a worker, both techniques and personality.

In addition to showing my experiences, both at university and in the professions, to help recruiters in making decisions when hiring.

### About the application

The application will respond to functional and non-functional requirements:

### **Functional Requirements**

#### 1. User Authentication and Management:

- The application must support user registration, login, and logout functionalities.
- It should allow users to manage their profiles, including updating personal information and changing passwords.

#### 2. Project Showcase:

- The application should enable users to view a portfolio of projects with detailed descriptions, images, and links to repositories or live demos.
- Projects must be categorized and searchable by technology, date, or other relevant filters.

#### 3. Admin Dashboard:

- The application should provide an admin dashboard where I can manage content, including adding, updating, or deleting projects.
- Admin functionalities should also include managing user roles and access permissions.

#### 4. Real-Time Data and API Integration:

- The application must integrate with external APIs to display real-time data, such as GitHub activity or live project statistics.
- It should also allow users to submit data through forms that are processed and stored in the database.

#### 5. Responsive Design:

- The front-end must be fully responsive, ensuring a seamless user experience across devices with different screen sizes, from desktops to mobile devices.

#### 6. Accessibility Compliance:

- The application should adhere to accessibility standards (e.g., WCAG), ensuring that all users, including those with disabilities, can navigate and use the application effectively.

#### 7. Database Management:

- The application must support both relational (e.g., MySQL, PostgreSQL) and non-relational (e.g., MongoDB) databases.
- It should allow for efficient data retrieval, storage, and management, with an emphasis on data security and integrity.

#### 8. Search and Filter Capabilities:

- Users should be able to search and filter through projects using keywords, tags, or other relevant criteria.

#### 9. Contact and Feedback System:

- The application should include a contact form where visitors can send inquiries or feedback.
- Submitted forms should be stored in the database and accessible via the admin dashboard.

### Non-Functional Requirements

#### 1. Performance:

- The application must load quickly, with a target page load time of under 2 seconds for the main pages.
- It should be optimized to handle multiple simultaneous users without performance degradation.

#### 2. Scalability:

- The application architecture should be designed to scale easily, allowing for future expansion, including more users, projects, or additional features.

#### 3. Security:

- The application must implement robust security measures, including encryption for sensitive data (e.g., passwords), protection against SQL injection, and other common vulnerabilities.
- User data must be handled securely, following best practices for data protection and privacy.

#### 4. Reliability and Availability:

- The application should be highly reliable, with a target uptime of 99.9%.
- Backup and recovery mechanisms must be in place to ensure data integrity in case of a system failure.

#### 5. Usability:

- The application should have an intuitive interface that is easy to navigate, even for non-technical users.
- The design should be user-friendly, with clear instructions and feedback mechanisms.

**NAME** Jose Luis Rico Ramos

**LinkedIn** <https://www.linkedin.com/in/jlricoramos>

**ADDRESS**

**EMAIL** [jlricoramos@gmail.com](mailto:jlricoramos@gmail.com)

**GitGub** <https://github.com/RiiC0Chet>

197 North Circular, Dublin 7

**6. Maintainability:**

- The codebase should be well-documented, following clean code principles to ensure that future updates and maintenance can be performed efficiently.
- The application should be modular, allowing for easy addition or modification of features without affecting the overall system.

**7. Compliance:**

- The application must comply with relevant regulations and standards, such as GDPR for data protection and accessibility standards like WCAG.

**8. Cross-Browser Compatibility:**

- The application should work consistently across all major web browsers (e.g., Chrome, Firefox, Safari, Edge).

**9. Version Control:**

- The project must use Git for version control, with a clear branching strategy to manage development, testing, and production environments.

## 4. Description of the solution

---

This section outlines the key aspects of the web application, detailing the technologies that will be used and providing an overview of the database structure.

It should be noted that the proposed solution (at least the documentation part) is not a complete representation that determines 100% of the work done. Since the time dedicated to this project is not infinite and only a part of this will be represented to imply that we have knowledge in all that is covered.

### 4.1. Technologies used

---

The project will leverage a variety of modern technologies to build a robust and scalable web application:

#### Front-End Technologies:

- **HTML5/CSS3:** For structuring and styling the user interface, ensuring a responsive and visually appealing design.
- **JavaScript (ES6+):** To add interactivity and dynamic features on the client side.
- **Bootstrap CSS:** For quick and flexible UI design, ensuring a responsive layout across different devices.

#### Back-End Technologies:

- **Ruby on Rails:** Serving as the core framework for back-end development, Rails will handle routing, controllers, and models, providing a clean MVC (Model-View-Controller) architecture.
- **Version Control and Collaboration:**
  - o Git and GitHub: For version control, enabling collaborative development, and managing code changes across different environments.
- **Deployment and Hosting:**
  - o Raspberry Pi: As the primary hosting environment, showcasing the ability to self-host and manage a web application on a local server.
  - o Docker: For containerizing the application, ensuring consistent deployment across different environments.
- **Agile Development Tools:**
  - o Jira: For managing the development process using agile methodologies, tracking progress, and organizing tasks.



NAME Jose Luis Rico Ramos

LinkedIn <https://www.linkedin.com/in/jlricoramos>

ADDRESS

EMAIL [jlricoramos@gmail.com](mailto:jlricoramos@gmail.com)

GitGub <https://github.com/RiiC0Chet>

197 North Circular, Dublin 7

- **Security Tools:**

- Environment Variables (.env): For securely managing sensitive configuration data like API keys and database credentials.