Ingeniería de Software

Fourth semester Software Engineering and Data Mining

Fourth partial Project

20 de Febrero del 2025

Members: Oliver Suarez Mora, Samantha de la Mora Lòpez, Karen Melissa Mancilla Jiménez

Project No.5 - Digital Portfolio 1. Description

The Digital Portfolio is a web platform designed for developers to organize and showcase their software projects. Through this tool, users will be able to personalize their profile with a photo, description, skills, and links to their social networks. They will also be able to manage their projects by adding key information such as descriptions, screenshots, technologies used, and links to repositories or live demos.

One of the platform's most notable features is its integration with the GitHub API, allowing easy import of projects stored in repositories. Additionally, a commenting and rating system will be included to foster interaction and idea exchange within the community.

To ensure a secure and efficient experience, authentication and access control mechanisms will be implemented, along with protection measures against cyber attacks.

2. Objectives General Objective

Develop a digital platform that enables developers to manage and share their software projects, providing a user-friendly interface and key features for organizing and presenting their work. **Specific Objectives**

- Implement a user registration and authentication system with secure access control.
- Develop an intuitive interface for project management.
- Integrate an API to automatically import projects from GitHub.
- Offer visibility configuration options for projects.
- Include a commenting and rating system to encourage user interaction.
- Apply appropriate security measures to protect user data.
- Optimize the platform for proper functionality across different devices and browsers.

3. Requirements

3.1 Functional Requirements

1. User registration and authentication with password recovery.
2. User profile creation and editing.
3. Project management with title, description, images, technologies used, and relevant links.
4. Tag and category assignment for projects.
5. Integration with GitHub to import stored projects.
6. Project visibility configuration.
7. Implementation of a search and filtering system for projects.
8. Functionality for commenting and rating projects.
9. Protection against inappropriate content in comments.

3.2 Non-Functional Requirements

1. Use of HTTPS for secure communication.
2. Password encryption using robust algorithms such as bcrypt or Argon2.
3. Implementation of authentication via JWT or OAuth.
4. Responsive design for compatibility with different devices.
5. A user-friendly interface to enhance user experience.
6. Scalability to support a growing number of users.
7. Structured and documented code for easy maintenance.

4. Main Use Cases: Registration and Authentication

Actor: User. **Description:** Allows users to register on the platform and securely log in. **Main Flow:**

1. The user completes the registration form with their information.
2. The system validates the data and creates the account.
3. A confirmation email is sent.
4. The user logs into the platform.

Project Management

Actor: Authenticated user. **Description:** Enables the creation, editing, and deletion of projects within the platform. **Main Flow:**

1. The user accesses the projects section.
2. The necessary project information is entered.
3. Changes are saved, and the project is stored in their profile.

Project Import from GitHub

Actor: Authenticated user. **Description:** Allows integration with GitHub for automatic project import. **Main Flow:**

1. The user connects their GitHub account from profile settings.
2. Authorizes access to their repositories.
3. Selects the projects to import.

4. The selected projects are synchronized with their digital portfolio.

`Comments and Ratings

Actor: Authenticated user. **Description:** Users can comment on and rate projects within the platform. **Main Flow:**

1. A user accesses a published project.
2. Adds a comment and/or rating.
3. The system verifies the content and publishes it if it meets the established guidelines.

5. Technologies to Use Backend

- Language: Python with Django or Node.js with Express.js.
- Database: PostgreSQL or MongoDB.
- Authentication: JWT or OAuth with GitHub integration.

Frontend

- Frameworks: React.js or Vue.js.
- Responsive Design: CSS with TailwindCSS or Bootstrap.

Infrastructure

- Server: Nginx + Gunicorn (for Django) or PM2 (for Node.js).
- Image Storage: AWS S3 or Cloudinary.
- Deployment: Docker + Kubernetes.

6. Security Considerations

1. Password encryption with bcrypt or Argon2 for enhanced security.
2. Implementation of HTTPS to protect data transmission.
3. Strict input validation to prevent SQL injection and XSS attacks.
4. Protection against CSRF attacks in forms.
5. Access control to restrict the editing and deletion of projects and comments.
6. Implementation of logging and activity auditing to detect potential threats.

7. Conclusion

The Digital Portfolio will be a valuable tool for developers, providing a structured platform to manage their projects and connect with the community. Thanks to its GitHub integration, customization options, and security measures, it will serve as a reliable and efficient solution for digital portfolio management.