



UNIVERSITÉ
DE LORRAINE



IUT Saint-Dié-des-Vosges

BUT informatique 2ème année

SAE4

Report on the Analysis and Improvement of a Web Application

MEHIAOUI Mohamed

MERIMI Ayat

Table of Contents :

1. Project Objective
2. Working Method
3. Technical Tools Used
4. Difficulties Encountered and Solutions
5. Skills Acquired and Professional Impact
6. Additional Notes

1. Project Objective:

The purpose of this Directed Study (SAé4) was to analyze and improve a web application previously developed during Semester 3. The chosen application is a website for a figure skating club, which provides functionalities for members, visitors, and administrators, each having a dedicated role and interface.

This project was commissioned by our academic institution as part of our second-year coursework in the BUT Informatique program. The website is intended for use by:

- Club members (for accessing information, registering for competitions, viewing schedules, etc.),
- Visitors (to discover club activities),
- And administrators (for managing content, registrations, and data).

Our group consisted of two members: *Mehiaoui Mohamed* and *Merimi Ayat*. We were assigned the task of evaluating another group's application, identifying issues, and delivering an improved version of the website, maintaining the same functionalities while ensuring enhanced quality, performance, and security.

2. Working Method

Our approach was divided into several key phases:

➤ Phase 1 – Analysis of the Existing Application

We conducted a thorough technical audit of the code and database. Each PHP file was reviewed to identify security flaws, code anomalies, or usability issues. We documented these in an "Analyse de l'existant" report, which served as our baseline.

➤ Phase 2 – Task Distribution

We shared the workload evenly:

One member focused on analyzing and improving the front-end (accessibility, interface structure, JS optimization).

The other addressed back-end concerns (security issues, database optimization, PHP refactoring).

➤ Phase 3 – Implementation

We corrected critical issues, such as:

SQL injection vulnerabilities (replaced raw queries with PDO prepared statements),

Session mismanagement,

Error visibility (sensitive errors were previously printed in the UI),

Database integrity constraints (foreign keys, type definitions),

Code redundancy and JavaScript optimizations.

➤ Phase 4 – Testing and Validation

We performed:

Unit and functional testing,

Security testing (CSRF protection, session regeneration),

Performance testing (especially database performance after index and engine updates).

3. Technical Tools Used

To carry out our work efficiently, we relied on :

Git- for version control and collaboration.

Browser Developer Console – Used for debugging front-end behavior, inspecting elements, and tracking JavaScript errors in real time.

MySQL Workbench – Used to visualize and monitor real-time changes in the database structure and content.

Visual Studio Code (VSCode) – Main code editor used for development with useful extensions for syntax highlighting, version control, and collaboration.

AI Assistance (e.g., ChatGPT) – Used for code suggestions, debugging help, and improving development productivity.

Docker – Used to containerize the database and web server environments, ensuring consistency across all development machines and simplifying deployment.



4. Difficulties Encountered and Solutions

We encountered several major issues during the project:

- Repeated SQL injection flaws: Resolved by using `prepare()` and `bindParam()` in all database queries.
- Improper session handling: Implemented `session_status()` checks and `session_destroy()` to prevent misuse.
- Security of login forms: We added password hashing with `password_hash()` and protected against login bypass.
- PDF output conflicts: Solved by ensuring `ob_end_clean()` is called before `pdf->Output()`.
- HTML structure issues: Fixed improper nesting of HTML elements (e.g., `<video>` inside `<p>`), and added missing ARIA attributes for accessibility.
- **Migration to Docker**
Transitioning the project to Docker required learning how to properly write a `docker-compose.yml` file and correctly link different services and folders. This was challenging at first, but allowed for a better understanding of containerization and improved environment consistency.
- **Reading and Understanding Legacy Code**
Reviewing and working with code that was not originally written by us demanded time and effort to understand the original developers' logic and intentions. We had to carefully trace the application flow and identify what each part was supposed to do before making changes.

- **Proposing and Implementing Reliable Solutions**

It was important not only to detect problems, but to suggest and apply fixes that actually worked without breaking other parts of the application. This required testing, validation, and sometimes going through several iterations to ensure stability.

Each issue was documented and addressed methodically.

5. Skills Acquired and Professional Impact

Throughout this project, we developed and strengthened key skills, including:

- Secure web development practices,
- Code auditing and debugging,
- Database optimization and integrity enforcement,
- Teamwork and task coordination within a small group,
- Technical documentation and delivery of a professional-grade product.

These skills are highly relevant to the professional world, especially in positions involving full-stack web development, software maintenance, cybersecurity, and project management.

6. Additional Notes

The final version of the application we delivered includes:

- All functionalities from the original application,
- Major improvements in terms of **security, performance, accessibility, and usability**,
- A professional technical and user documentation set,
- A production-ready, deployable codebase.

Conclusion

This project enabled us to deepen our understanding of secure and maintainable web development through the critical analysis and improvement of an existing application. By identifying and correcting key issues—such as security flaws, database inconsistencies, and poor accessibility—we delivered a fully functional and optimized version of the original system.