Project Reflection

LEO-Based Assessment Tool

---

1. Introduction

This document provides a **project reflection** for the *LEO-Based Assessment Tool*, developed as part of the **Software Engineering Project (SENGPRJ)** course at FHTW.

The reflection summarizes the overall project experience, key challenges, solutions, lessons learned, and personal as well as team development outcomes.

---

2. Project Overview

The goal of the project was to design and implement a **learning outcome–based assessment system** that supports teachers and students through transparent progress tracking, structured assessments, and recommendation logic.

The system consists of:

- A **Spring Boot backend** handling business logic, grading rules, and persistence
- An **Electron-based frontend** providing an intuitive user interface
- A **cloud-based PostgreSQL database (Neon)**

The project was developed iteratively using agile methods and delivered as a fully deployed and runnable system on AWS.

---

3. Challenges Faced

3.1 LEO Dependency Graph & Cascade Logic

One of the main technical challenges was implementing the **LEO dependency graph** and the **cascade grading logic**. A single assessment change could affect multiple dependent learning outcomes.

Ensuring that: - dependencies were handled correctly - updates were applied consistently - no invalid states were created

required careful design of the service layer and extensive testing.

3.2 Frontend–Backend Integration

Another challenge was coordinating frontend and backend development in parallel. Changes to API contracts or data structures required close communication and frequent adjustments.

This highlighted the importance of: - clear API design - early integration testing - shared understanding of domain concepts

---

3.3 Deployment & Environment Configuration

Deploying the system using **Docker on AWS EC2** and connecting it to a **cloud-based database (Neon)** introduced additional complexity.

Handling environment variables, CORS configuration, and container orchestration helped the team gain practical experience with real-world deployment scenarios.

---

4. Solutions & Improvements

To address these challenges, the team:

- Centralized complex logic (cascade grading, validation) in the backend
- Used a clear layered architecture to reduce coupling
- Established consistent API contracts between frontend and backend
- Relied on Docker and environment-based configuration to standardize deployment

Iterative refinement during sprints allowed issues to be identified and resolved early.

---

5. Teamwork & Collaboration

Effective teamwork played a crucial role in the success of the project.

Key aspects included:

- Clear task assignment using GitHub Projects and Azure DevOps Boards
- Regular communication during development
- Collaborative problem-solving
- Mutual code reviews and support

The agile workflow helped ensure transparency and shared responsibility.

---

6. Learning Outcomes

6.1 Technical Learning

Through this project, the team gained hands-on experience in:

- Designing a full-stack application
- Implementing REST APIs with Spring Boot
- Managing role-based security
- Modeling complex domain logic using graphs

- Frontend development with React, Vite, and Electron
- Cloud deployment with Docker and AWS

6.2 Personal & Professional Development

Beyond technical skills, the project contributed to personal growth in:

- Communication and collaboration in a team setting
- Time management and prioritization
- Handling complexity and uncertainty
- Working with agile processes in practice

---

7. What Went Well

- Clear separation of frontend and backend responsibilities
- Successful implementation of core project requirements
- Stable deployment and runnable system
- Comprehensive documentation covering all aspects of the project

---

8. What Could Be Improved

If more time were available, potential improvements would include:

- Automated frontend testing
- More advanced LEO visualizations
- Performance optimizations for large LEO graphs
- Additional usability testing with real users

---

9. Conclusion

The *LEO-Based Assessment Tool* project was a valuable learning experience that combined software engineering theory with practical implementation.

Despite technical and organizational challenges, the team successfully delivered a complete, well-structured, and deployed system. The project strengthened both technical expertise and teamwork skills and provided strong preparation for future software engineering tasks.

Group 6 — SENGPRJ
Supervisor: *Thomas Mandl*