Personal Contribution, Personal Development & Evidence for Grading

LEO-Based Assessment Tool

---

1. Introduction

This document describes the **personal contributions**, **personal and technical development**, and the **evidence provided for grading** for the *LEO-Based Assessment Tool* project. The project was developed as part of the **Software Engineering Project (SENGPRJ)** course at FHTW by **Group 6**.

The purpose of this document is to transparently demonstrate individual and team contributions, learning outcomes, and how grading criteria are fulfilled and evidenced.

---

2. Personal Contribution

All group members contributed actively to the project across multiple phases, including analysis, design, implementation, testing, deployment, and documentation.

2.1 Contribution Areas

Individual and shared contributions included:

- Backend development using **Spring Boot** and **Java 17**
- Frontend development using **Electron** and modern web technologies
- Design and implementation of REST APIs
- Implementation of LEO graph logic and cascade grading rules
- Recommendation logic for next possible LEOs
- Database modeling and integration with **Neon PostgreSQL**
- Deployment using **Docker** and **AWS EC2**
- Testing (unit, integration, and API tests)
- Documentation (user handbook, setup guides, technical documentation)

2.2 Task Assignment & Traceability

Individual responsibilities and task assignments were tracked using project management tools:

- **GitHub Projects** for implementation tracking
- **Azure DevOps Boards** for backlog and sprint management

Each task was assigned to specific team members, enabling clear traceability of personal contributions. Completed tasks, commits, and pull requests provide verifiable evidence of individual work.

---

3. Personal & Technical Development

3.1 Technical Development

Through this project, team members strengthened their technical skills in the following areas:

- Full-stack development and frontend–backend integration
- Backend development with **Spring Boot** and layered architectures
- REST API design and role-based security
- Graph-based domain modeling (LEO dependency graphs)
- Implementation of business rules such as cascade grading
- Database integration with cloud-based PostgreSQL (Neon)
- Containerization and deployment using Docker and Docker Compose
- Cloud deployment and configuration on AWS EC2
- Testing strategies using JUnit 5, Spring Boot Test, and Testcontainers

---

3.2 Personal Development

In addition to technical skills, the project supported personal development in:

- Team communication and collaboration
- Task coordination and responsibility sharing
- Time management and prioritization
- Problem-solving in a collaborative environment
- Working with agile development processes

The iterative sprint-based workflow helped improve planning accuracy and adaptability to changing requirements.

---

4. Evidence for Grading Criteria

All grading criteria defined in the course are supported by concrete and verifiable evidence. The evidence is structured according to the three main grading categories.

4.1 Solution (40%)

Functionality (30%)

Evidence:

- Fully implemented frontend and backend applications
- Role-based authentication and authorization (Teacher / Student)
- LEO creation and dependency management
- Assessment recording and editing
- Cascade grading logic
- Progress visualization and recommendation logic
- Deployed and runnable system on AWS using Docker

Artifacts: - Backend Repository: https://github.com/piy678/SENGPRJ_Group6 - Frontend Repository: https://github.com/piy678/SENGPRJ_Group6_FrontendPart - Deployed application (AWS EC2)

Quality (10%)

Evidence:

- Clean layered backend architecture (controller, service, persistence)
- Use of Spring Boot, JPA/Hibernate, and Electron
- Validation, error handling, and security checks
- Unit and integration tests
- Code reviews and refactoring during development

Artifacts: - Test classes in backend repository - Structured package organization - Commit history documenting refactoring and improvements

---

4.2 Process (40%)

Evidence:

- Agile development with multiple iterations (sprints)
- Clearly defined user stories and requirements
- Continuous backlog refinement
- Task assignment and progress tracking
- Sprint planning and sprint reviews

Artifacts: - GitHub Project Board: https://github.com/users/piy678/projects/7 - Azure DevOps Boards (Backlog & Sprints): https://dev.azure.com/BWI-25WS-SEPR-Team06/LEOBasedAssessment/_sprints/backlog/LEOBasedAssessment%20Team/LEOBasedAssessment/Sprint%201

4.3 Presentation (20%)

Presentation (10%)

Evidence:

- Structured project presentation
- Explanation of system architecture and core concepts
- Live demonstration of implemented features

Artifacts: - Presentation slides (submitted via Moodle and GitHub)

Project Reflection (10%)

Evidence:

- Written reflection document

- Description of challenges, solutions, and lessons learned
- Evaluation of teamwork, tools, and development process

Artifacts: - Project reflection document submitted via Moodle

---

5. Summary

The documented personal contributions, personal development, and provided artifacts demonstrate that all grading criteria for the *Software Engineering Project* course are fulfilled.

Clear task traceability, comprehensive documentation, and a fully implemented and deployed system provide transparent and verifiable evidence for grading.

Group 6 — SENGPRJ
Supervisor: *Thomas Mandl*