

Grading criteria

- 0 point - The work is insufficient
- 0.1 point - The work is done
- 0.2 point - The work is well done (without the need of being perfect)

Maximum grade: 25 points * 0.2 + 1 = 6

[!IMPORTANT]

While the grading criteria might not be as detailed as in the previous practical works for each section, you **must** continue to apply all the good practices you have learned so far.

If elements that are supposed to be acquired through the course or previous practical works are omitted, forgotten or poorly implemented, we might penalize you.

Remember the UNIX philosophy and the KISS principle: *Keep it simple, silly!*

Category 1 - Git, GitHub and Markdown

If your repository is private, you must add us as collaborators to your repository!

#	Criterion	Points
1	The README is well structured and explains the purpose of your web with the authors' names so new users can understand it and know who is behind the application	0.2

Category 2 - Java, IntelliJ IDEA and Maven

#	Criterion	Points
2	The README describes explicit commands to clone and build your network application with Git and Maven so new developers can start and develop your project on their own computer	0.2

#	Criterion	Points
3	The codebase is well structured, easy to access, easy to understand and is documented so it is easier for new comers to understand the codebase	0.2

Category 3 - Docker and Docker Compose

#	Criterion	Points
4	The network application is packaged and published to GitHub Container Registry with Docker so other people can use your network application with Docker	0.2
5	The README describes explicit commands to build and publish your network application with Docker	0.2
6	The README describes explicit commands to use your network application with Docker Compose so other people can easily use it	0.2
7	The Docker applications (Traefik and your web application) are split into multiple directories and make usage of networks to communicate together	0.2

Category 4 - Java network concurrency

#	Criterion	Points
8	The data structures used in the network application are resilient to concurrent accesses	0.2

Category 5 - SSH and SCP

#	Criterion	Points
9	You and the teaching staff can access the virtual machine without a password using a SSH key	0.2

Category 6 - HTTP and curl

#	Criterion	Points
10	The README (or repository) contains the application protocol interface (API) that describes the web application	0.2
11	The web application makes usage of at least the following HTTP methods: GET , POST , PATCH / PUT and DELETE	0.2
12	The web application return status codes must be consistent and reflect the HTTP methods	0.2
13	The web application offers at least two resources (= domains) on which to operate CRUD operations	0.2
14	The README explains how to use your web application with explicit examples using curl with outputs to demonstrate how to interact with your web application deployed on the Internet	0.2

Category 7 - Web infrastructures

#	Criterion	Points
15	The README (or repository) contains instructions how to install and configure the virtual machine with each step	0.2
16	The README (or repository) contains explains how to configure the DNS zone to access your web application	0.2
17	The README (or repository) contains instructions how to deploy, run and access the web applications with Docker Compose	0.2
18	At least Traefik and your web application are deployed on the virtual machine	0.2
19	The README displays the domain names configuration in the DNS zone to validate everything is set up right	0.2
20	The web applications (the Traefik dashboard and your own application) are accessible using a domain name and/or subdomain names	0.2
21	The web applications (the Traefik dashboard and your own application) use automatic HTTPS/TLS certificate generations with Let's Encrypt	0.2

Category 8 - Caching and performance

#	Criterion	Points
22	The requests are cached to improve performance until the data changes	0.2

Category 9 - Presentation and questions

#	Criterion	Points
23	The web application is presented and a demo is made as you would do it to a colleague/another team/boss/client/investor so they can understand what you created, why and how	0.2
24	The presentation is clear and well prepared - everyone speaks during the presentation	0.2
25	The answers to the questions are correct	0.2

Constraints

- The application must be written in Java, compatible with Java 21
- The application must be built using Maven with the `maven-shade-plugin` plugin
- The application must use the Javalin dependency
- You can only use the Java classes seen in the course to implement the network application (you can use any other libraries for other aspects of the application, such as UI, etc.)
- Your application must be slightly more complex and slightly different than the examples presented during the course (we emphasize the word **slightly**, no need to shoot for the moon!)
- The web application can only use the HTTP/HTTPS protocols