SEC

Lab #2 - Authentication

- This lab will be graded.
- The quality of your code will be graded.
- The quality of your design will be graded.
- Your submission has to be in Rust.
- Your code should be tested and you should provide your tests.
- You do **not** need to submit a report.

1 Two-factor Authentication

The goal of this lab is to implement a **two-factor authentication mechanism** (2FA). The two factors will be

- A password
- A `Google Authenticator` token.

You have to implement the following features:

1. A **registration process** asking for a password, and an email address. The email is going to be used as a username and to reset the password.
2. A **login process** asking for an email address, a password, and a Google Authenticator token (if 2FA is active).
3. A **password reset** process sending a token by email that is valid **15 minutes** and using Google Authenticator (if 2FA is active).
4. A possibility to disable the 2FA for a user.

2 Indications

- The focus of this lab is the security part. You can simulate all the other parts (no need having a server, real emails, databases, . . .)

- Be careful on the requirements when disabling 2FA. An attacker should not be able to do it.

- Be also careful on how you store client data.

- For Google Authenticator, you can use the following crate: `https://docs.rs/google-authenticator`.