# SLH 23-24

# Labo 3 – Restaurants

- You have to submit your code and a very short report explaining your modifications.

## 1 A Restaurant Grading Application

Due to the terrible quality of some restaurants in Yverdon-les-Bains, a company decided to develop an application grading these restaurants. The goal is to help restaurants' owners to improve their offer. You just got hired as a security developer by this company to review their application. Unfortunately, the security of the application was completely ignored. Do your job and secure the application! You will find the source code on cyberlearn.

### 1.1 Indications

- There are three types of accounts in the app
  - A `Reviewer` can read **their own reviews** and review any restaurant.
  - An `Owner` can read **the reviews they wrote** and **the reviews of their own restaurant**. They are **not** allowed to write reviews for their own restaurant.
  - An `Admin` can read **any review**, add reviews, and **delete any review**.
- Only `Reviewer` and `Owner` accounts can be created. The `Admin` accounts have to be set up at the start of the database (see `Database::init()`).
- Access control should be managed with an access control tool (ex. casbin) and not by hand.
- A grade is an **integer** between 1 and 5.
- Since this lab is earlier in the semester, you do **not** need to take care of encryption or logging.
- Think in term of authentication, authorization, error management, and input/output validation.
- Write in a very short report what mistakes you found and which changes you did to the software (high level report).
- You should **not** touch the file `db.rs`. You can decide how to populate the database in the beginning by redefining the `Database::init()` method in the file `main.rs`.

- The content of the db is stored in the file `database.json`. To reinitialize the database, simply delete the file.
- The database system performs already some checks: one review per restaurant per person, maximum one owner per restaurant, unique usernames.
- You are allowed to change completely the software as long as it does (at least) what it should (except `db.rs`).
- The goal of the project is to secure the application and not to improve its UX. Nevertheless, if you wish to improve on this side, you can. Thus, there is no need to change for a more complex database system.