The goal is to design a database that has different access authorizations.

This is to guarantee that the respective users of the database (patients, receptionists, doctors and admins) only get the necessary privileges to the database.

A database must first be created with Mysql, which can be accessed with a username and password.

In the next step, the access authorizations must be distributed. Since all user information should be in a database, the access authorizations for the clinic's staff (receptionists, doctors and admins) must be entered manually. Patients, on the other hand, are implemented in the system with predefined usage authorizations so that no manual assignment should be made when a new patient is created.

In order to protect the users' passwords in the event of a security breach or by a malicious admin, they should be encrypted.

Threats to Consider:

If the privileges are configured incorrectly, users can make malicious modifications to the databases.

Privileges to be considered:

ALL

ALTER

CREATE

CREATE USER

DELETE

DROP

FILE

INDEX

INSERT

LOCK TABLES

SELECT

GRANT OPTION

SUPER

Patients have no privileges.

Receptionists need access to the appointment appointments table (SHOW) and must be able to enter new appointments (CREATE), change them (ALTER) and discard them (DELETE).

Doctors need access to the Appointment Appointments (SHOW) table. They must also be able to access the patient's medical files (SHOW) and make entries (CREATE).

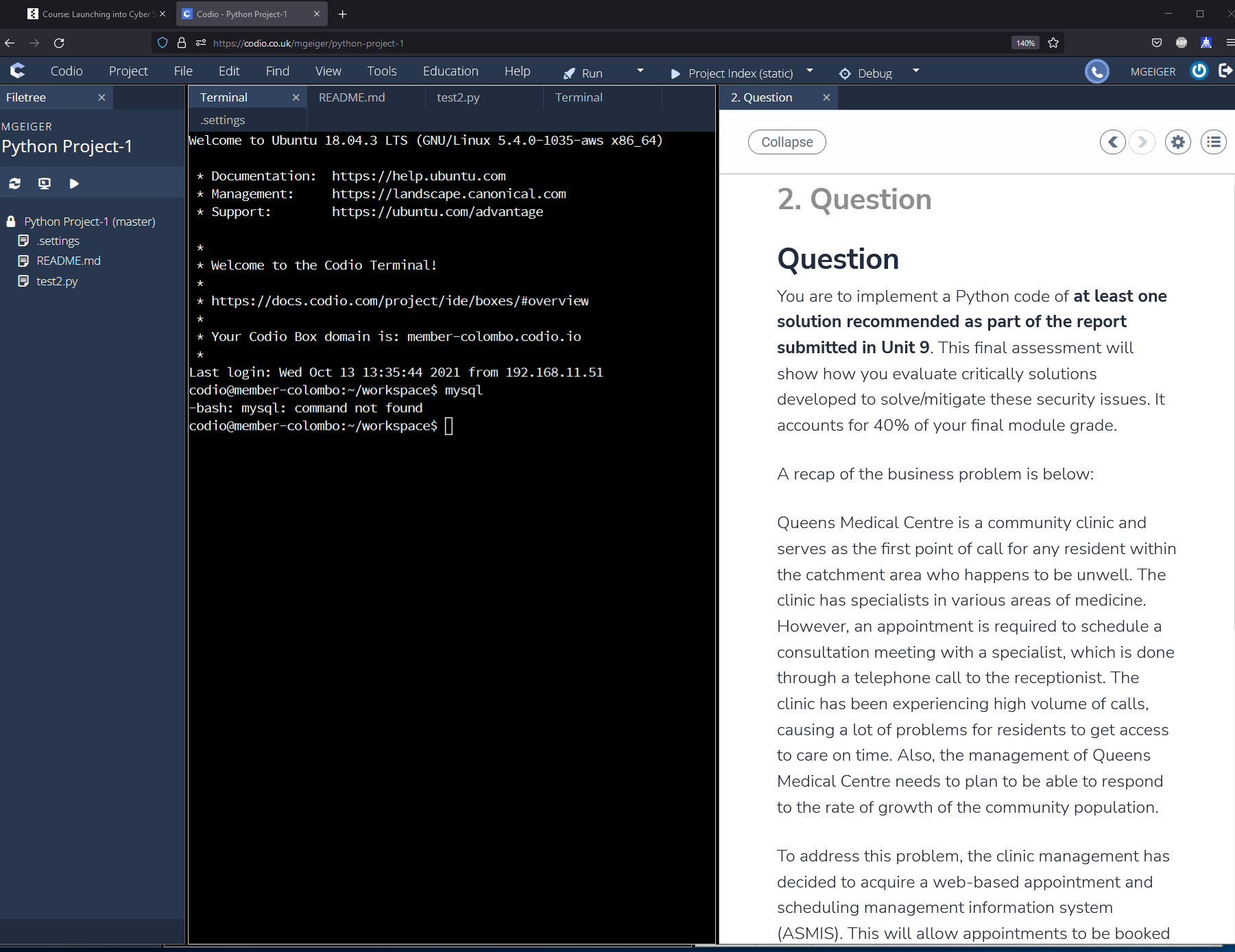
The maintenance of the ASMIS system has two aspects. On the one hand, it must be possible to delete user data, as patients could die or no longer want to be entered in the system (GDPR). Also, staff can no longer work in the hospital and therefore access to the database must be withdrawn from the person (DELETE). Passwords can be forgotten and a new password must be entered (AGE) or new personnel must be entered into the system (CREATE).

On the other hand, a person must also have access to the server administration in order to close security gaps and to keep the system up to date.

So it takes two types of admins.

Here is a question: is it possible to give a user access to the administrative privileges without this person being able to access the some contents of the tables? A person with administrative privileges can give himself such privileges.

I face a problem where I just don't know what I am doing wrong:

When I open the "Python Project" in Codio and try to access SQL with "mysql", I get an error message (see picture below).  The command works perfectly with other Codi tabs. From this problem two questions developed for me:

1. What am I doing wrong, or what I have not considered.
2. Is it basically necessary to work on the task in Codio or can I also work with, for example, MySQL Workbench 8.0 and PyCharm and submit the folder?