Criptografie Aplicată

Applied Cryptography / Cryptographie Appliquée

DEADLINE: 26/MAY/2026 23:59

Subject:

1) Create a **Database**:

- Must be a RELATIONAL database (example MariaDB, Postgres)
- You must be able to interact with it in Python
- Database must contain ONE table
    - TABLE users
        - **ID (**auto increment int)
        - **Username** (string)
        - **User email** (string)
        - **User password** (string)
        - **User type** (enum) [‘user’, ‘admin’]

2) Create a **Web Application** in **Python** that:

- Has a fully functional **USER** system
    - Users can login and register
    - Users can be normal users or admin users
        - For admin users, make an additional page accessible only to them
    - Integrated with a **DATABASE**
        - Users are stored in the database (**RELATIONAL database**)
    - User information is stored in a JWT
    - You apply password security practices (hash and compare)

- Displays a page of your choosing (HTML + CSS) [**content is not important**, I just want to see it rendering]

- Has available route to retrieve user information
    - Example: GET /users/3 -> returns id, username, user email for user 3 in the database

- Has available route to create user
    - Example: POST /users/new
    - JSON payload:
        - Username
        - User email
        - Password
    - User is automatically of USER type

- Has available route to create admin user
    - Example: POST /users/admin
    - JSON payload:
        - Username
        - User email
        - Password
    - User is automatically of ADMIN type

3) **BONUS** Create a **separate** Python script that generates the database

Notes:

You can use POSTMAN to test HTTP Requests: https://www.postman.com/downloads/

If you use Postgres for the database, make sure you install the right Python package: https://pypi.org/project/psycopg2-binary/