01UDFOV/01TXYOV Applicazioni Web I / Web Applications I [2024/2025]

Lab 09: Multi-user Application with Authentication

In this lab, you will complete your application by including the authentication and the related features.

1. Login Your Users

Add the possibility of **having multiple users** in your application, enabling them to **authenticate** (i.e., login and logout functionalities).
In the front-end, add a new page (with a dedicated route) containing a form, which will be used to log in. The login form will have two mandatory fields: email and password. The login form should be validated before its submission, and you must use proper error messages when inconsistencies are found.

Specifically, at least the following checks should be executed:

- When a field is missing or empty it must be forbidden to send a log-in request to the server.
- The email should be properly formatted (i.e., *something@something*).

In Express, implement the login process by exploiting the Passport authentication middleware, as shown in class.

Add users in the database using a suitable username (in the format name.surname@polito.it) and an hashed password of your choice (see notes).

Assign at least three items from your chosen exam topic from those already in the database to each of the newly created users.

When the login process fails, the front-end should display a suitable error message (e.g., "*Incorrect username or password*") and continue to show the login form.

Instead, when the login is successful, the application redirects the users to their home page, showing a message like: "*Welcome, {name_of_the_user}*".

2. Manage protected routes and authorization

Identify the Express routes that need to be authenticated and protect them accordingly. Implement all the necessary modifications to your react application to prevent unauthenticated users from trying to access pages and commit actions they should not have access to.

3. Logout Your Users

Finally, implement the logout functionality, again by exploiting the Passport authentication middleware. When the users are logged out, redirect them to the login form.

Notes:

1. **DO NOT** store plain text passwords in the database!
    a. Use scrypt (see below) to generate a hash of the passwords before saving them.
    b. Each password MUST have a different salt!

2. You can use the following website to generate the hash of a password, according to scrypt: https://www.browserling.com/tools/scrypt.
3. If you need to generate a salt by hand, you can use https://www.browserling.com/tools/random-hex, considering the length of the generated hex as the one of the salt (e.g., 16).