Sistemas de Gestão de Identidade / Identity Management Systems
Universidade Lusófona - 2024/2025

Assignment #2 – Web Identity Management

Prof. Dr. João Craveiro

This assignment aims to demonstrate the techniques of **Identity Management on the Web** presented during theoretical classes, through the implementation of a proof of concept.

Requirements

The work must cover the following aspects:

1. **Architecture Study:** Analysis of the utilized architecture and presentation of its components.
2. **Libraries Determination:** Identification of the necessary libraries for the implementation.
3. **Environment Setup:** Installation on the student's PC, virtual machine, or another environment of choice suitable for the demonstration.

Implementation Requirements

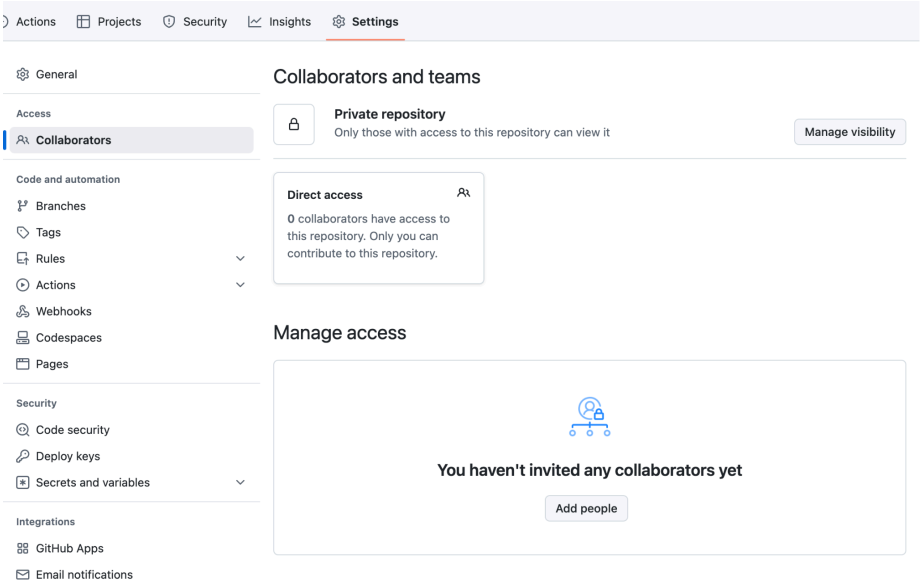
The implementation must include the following features:

1. **Application Access via Authorization Protocol:** Access to the application website should utilize an authorization protocol (e.g., OAuth2) with explicit consent, based on a Google account.
2. **User Profile Access:** Authorization must provide access to basic user profile data (google_id, username, and email), to be displayed on a success page.
3. **Session Persistence:** Session persistence must be managed implicitly by session middleware, while the user's profile should be explicitly stored in a separate collection within the same database (e.g., MongoDB Atlas).
4. **Resource Creation:** Once authenticated, the user should access a specific form to create a personal resource (e.g., an *item*), which should be stored in a separate collection (`items`). Feel free to get creative on the name/purpose of your item (Blog posts? Shopping lists?).
   Suggested fields:
   - "Title"
   - "Description"
   - "Creation Date"
5. **Resource Protection:** The created resource must be protected and accessible only by the user who created it while authenticated.
6. **Resource Deletion:** The user should be able to delete the resources they have created.
7. **Unauthorized Access Handling:** An unauthorized user attempting to access the protected resource should receive an error message or, ideally, be redirected to the authorization flow.

8. **Logout Management:** Upon logout, the session should be terminated, but the user's profile and resources stored in the database should remain intact.
9. **Page Rendering with Template Engine:** Resource creation and access pages should utilize a template engine similar to the one provided in examples (e.g., EJS).
10. **Code Submission and Deployment:** The application code must be submitted as a repository on GitHub, with additional credit for deployment on a cloud platform (e.g., Evennode, Vercel, Heroku).

Instructions for submission

- The submission must reflect the student's **individual and independent** solution to the assignment. This fact may be validated in a later demo/discussion session.
- The application code must be submitted as a repository on GitHub, with additional credit for deployment on a cloud platform (e.g., Evennode, Vercel, Heroku).
- *Optional, but valued in the grade*: Upload a 1 minute demo of your implementation to YouTube (as an **Unlisted** video), and include the link in your GitHub repo's `README.md`.
- On Moodle, submit only a text file with the link to your GitHub repo (and eventually the link to your deployed application).
- Keep your repo private, but don't forget to add the course instructor (https://github.com/jpgcc) as a collaborator:



Collaborators and teams

Private repository

Only those with access to this repository can view it

Direct access

0 collaborators have access to this repository. Only you can contribute to this repository.

Manage access

You haven't invited any collaborators yet

- ***The submission may be in either English or Portuguese.***