**Django API Implementation Test**

**Objective:**

The goal of this task is to evaluate the candidate’s ability to implement authentication, API documentation, and security measures using Django, Django REST Framework (DRF), and Swagger.

**Project Requirements:**

Develop a Django-based authentication system with cookie-based authentication. The system should include user registration, login, and a protected endpoint to retrieve the logged-in user’s details.

**Task Breakdown**

**1. Project Setup & Configuration**

* Create a new Django project and an app named authentication.
* Install and configure:
  + Django REST Framework (DRF) for API development.
  + Install  Swagger package.
* Ensure that Swagger (/swagger/) automatically generates a **CSRF token** when accessed.

**2. User Registration API (POST /api/register/)**

* Accepts email, password.
* Send a one time password to email and verfiy
* Return a success response upon successful registration.

**3. User Registration API (POST /api/register/verify)**

* Use this api to verify the registration

**4. User Login API (POST /api/login/)**

* Accepts email and password.
* Validates credentials and authenticates the user.
* On successful authentication:
  + **Set an authentication token (auth\_token)** in an **HTTP-only cookie**.
* API access should only work **after login**.

**5. User Details API (GET /api/me/)**

* Returns details of the **logged-in user**.
* Should only be accessible **if the user is authenticated** using the auth\_token from cookies.

**6. Logout API (POST /api/logout/)**

* Clears the **auth\_token cookie** from the client.
* Prevents further API access until the user logs in again.

**7. Security Considerations**

* **CSRF Protection:**
  + The csrftoken should be automatically generated when Swagger is opened.
  + All requests should require a CSRF token for validation.
* **Cookie-based authentication only** (No token headers).
* **Secure Cookies:**
  + Use HttpOnly and Secure flags to prevent client-side access.

**Bonus (Optional but Preferred)**

* Implement a simple frontend (HTML + JavaScript) to test authentication flow. If not, swagger docs will be used to test authentication flow.

**Evaluation Criteria**

* Proper implementation of **cookie-based authentication**.
* Swagger API setup with **automatic CSRF token generation**.
* Secure handling of **user authentication** and **protected API endpoints**.
* Code quality and best practices.

**Make sure your repository includes:**  
✅ A **README.md** file with setup instructions.  
✅ A **requirements.txt** file for dependencies.  
✅ Proper **commit history** to track your progress.

Looking forward to your submission!

**IMPORTANT:**

**Once you have completed the implementation, please upload your project to GitHub and share the repository link, and send an unlisted demo YouTube link. This will allow us to review your code, test the API, and evaluate your implementation.**