🧪 PHP Exercise – User Profile Manager (without database)

🎯 Objectives

- Master PHP forms
- Handle files (uploads, JSON)
- Use sessions to manage login
- Secure user data

🧰 Instructions

1. Login page (`login.php`)

- A form with a **"username" field** (e.g., "jdupont").
- On submission:
  - Check that the username is alphanumeric.
  - Save the username in the **session**.
  - Redirect to `profile.php`.

2. Profile management page (`profile.php`)

- Check that the user is logged in via the session.
- Display a **form** with:
  - First name (text input)
  - Bio (textarea)
  - Profile photo (image file)
- On submission:
  - Sanitize fields (`htmlspecialchars`)
  - Validate and upload the photo (images only, max 2 MB)
  - Save the data in a `users.json` file (format given below)

3. Format of the `users.json` file

Here is a possible format example for the JSON:

```json
{
  "jdupont": {
    "prenom": "Jean",
    "bio": "Junior web developer.",
    "photo": "../uploads/jdupont.jpg"
  },
  "amartin": {
    "prenom": "Alice",
    "bio": "Web mobile student.",
    "photo": "../uploads/amartin.png"
    }
}
```

4. 🔐 Bonus Improvements (optional)

These elements help strengthen security and prepare learners for more advanced concepts:

- **Add a password to the login**
  - Add a "password" field in `login.php`.
  - Save the hashed password with `password_hash()` in `users.json`.
  - Verify the password using `password_verify()` during login.
- **Protection against CSRF attacks**
  - Generate a CSRF token and store it in the session.
  - Add a hidden input field containing this token in all forms.
  - Check that the submitted token matches the one stored in the session.
- **Add a simple captcha**
  - Generate a code (text or simple calculation) stored in the session.
  - The user must enter the result in a form field.
  - Compare the values during form submission.
- **Server-side and client-side validation**
  - Server-side:
    - Check file MIME types (`mime_content_type()`).
    - Validate email formats (`filter_var()` with `FILTER_VALIDATE_EMAIL`).
  - Client-side:
    - Add HTML attributes like `required`, `pattern`, etc.
    - Add JavaScript validation to guide the user.
- **Logging user logins**
  - On each successful login, write a line to a `log.txt` file including:
    - Date and time
    - Username
    - User's IP address (`$_SERVER['REMOTE_ADDR']`)