🧪 PHP演習 – ユーザープロファイル管理（データベースなし）

🎯 目的

- PHPのフォーム処理を習得する
- ファイル操作（アップロード、JSON）を扱う
- セッションを使ったログイン管理
- ユーザーデータのセキュリティを確保する

🧰 指示事項

1. ログインページ (`login.php`)

- **「識別子」フィールド**を持つフォーム（例: "jdupont"）
- 送信時に：
  - 識別子が英数字のみであることを検証する
  - セッションに識別子を保存する
  - `profil.php`にリダイレクトする

2. プロファイル管理ページ (`profil.php`)

- セッションでユーザーがログイン済みであることを確認する
- 次の入力を含む**フォーム**を表示する：
  - 名前（テキスト入力）
  - 自己紹介（テキストエリア）
  - プロファイル写真（画像ファイル）
- 送信時に：
  - 入力値をセキュアに処理（`htmlspecialchars`）
  - 写真の検証とアップロード（画像のみ、最大2MB）
  - `users.json`ファイルにデータを保存する（下記フォーマット）

3. `users.json`ファイルのフォーマット例

```
{
  "jdupont": {
    "prenom": "Jean",
    "bio": "ジュニアウェブ開発者。",
    "photo": "../uploads/jdupont.jpg"
  },
  "amartin": {
    "prenom": "Alice",
    "bio": "モバイルウェブの学生。",
    "photo": "../uploads/amartin.png"
  }
}
```

4. 🎧 ボーナス改善（任意）

これらはセキュリティを強化し、より高度な概念への理解を助けます：

- **ログイン時のパスワード追加**
  - `login.php`に「パスワード」フィールドを追加
  - `users.json`に`password_hash()`でハッシュ化したパスワードを保存
  - ログイン時に`password_verify()`でパスワードを検証
- **CSRF攻撃の防止**
  - CSRFトークンを生成し、セッションに保存
  - 全てのフォームに隠しフィールドとしてトークンを挿入
  - フォーム送信時にセッションのトークンと照合
- **シンプルなキャプチャの追加**
  - セッションに保存したコード（テキストまたは簡単な計算式）を生成
  - ユーザーがフォームで結果を入力
  - 送信時に値を比較して検証
- **サーバー側・クライアント側でのバリデーション**
  - サーバー側：
    - ファイルのMIMEタイプを検証（`mime_content_type()`）
    - メールアドレスの形式検証（`filter_var()`と`FILTER_VALIDATE_EMAIL`）
  - クライアント側：
    - HTMLに`required`や`pattern`属性を追加
    - JavaScriptでユーザー入力のガイドと検証を追加
- **ログイン記録の保存**
  - ログイン成功時に`log.txt`に以下を記録：
    - 日時
    - 識別子
    - ユーザーのIPアドレス（`$_SERVER['REMOTE_ADDR']`）