



Tutorial 3: PHP

Required software:

- Web browser: Internet Explorer/...
- HTML editor: Notepad/...
- WAMP

Exercise 1

Write `add_student.php` that:

1. Takes a form with fields:
 - `student_id`
 - `name`
 - `group`
2. Validates them.
3. Loads existing students from `students.json` (if it exists).
4. Adds the new student to the array.
5. Saves back to `students.json`.
6. Displays a confirmation message.

Exercise 2

Create `take_attendance.php` that:

1. Loads students from `students.json`.
2. Shows a list with Present / Absent.
3. On submit:
 - Create a file named `attendance_YYYY-MM-DD.json`.
 - Save attendance as an array of:

```
[  
  ["student_id" => "101", "status" => "present"],  
  ["student_id" => "102", "status" => "absent"]  
]
```
 - If the file for today already exists, show:
"Attendance for today has already been taken."



Exercise 3

1. Create a file named config.php containing:
 - database host
 - username
 - password
 - database name
2. Create a second file named db_connect.php that:
 - uses a try/catch block
 - returns a connection object
 - handles errors cleanly
 - logs failures to a file (optional)
3. Test the connection by writing a short script:
 - connect to the database
 - display “Connection successful” or “Connection failed”

Exercise 4

1. Create a table students (id, fullname, matricule, group_id)
2. Implement PHP scripts:
 - add_student.php
 - list_students.php
 - update_student.php
 - delete_student.php

Exercise 5

1. Create a table attendance_sessions (id, course_id, group_id, date, opened_by, status)
2. Create a script create_session.php that:
 - receives course + group + professor ID
 - stores the session in the database
 - returns a session ID
1. Create a script close_session.php that updates the status to "closed".
2. Test by inserting 2–3 sessions manually.