



Assignment 1 (Registration webpage)

Students number: Min 5 – Max 6 (groups must be in the same lab) **Deadline:** week starting 6 April

Design a registration web page that inserts a user data into MySQL database, the details are as follow:

1. The form includes a user personal detail: **full_name, user_name, birthdate, phone, address, password, confirm_password, user_image, and email**. Use three appropriate client-side validations. (All fields mandatory, email and birthdate and full_name are correct types, password match with confirm_password and is at least 8 characters with at least 1 number literal and 1 special character) [Client-side VALIDATIONS]
2. Include a header and footer pages of your own design inside the registration webpage.
3. Maintain a **User's** table in the database to check if the username is not registered before. If the user inputs a username that exists before, the form alerts the user to choose another username [Server-side VALIDATIONS]
4. Upload the user image (the image will be stored on the server, while the image name is stored in the database).
5. Beside the **birthdate** field add a button that checks actors born in the same day using a third-party MDBI API "<https://rapidapi.com/apidojo/api/imdb8/>" (shown in the figure below)

You need to show the list with actor's names born on the same day.

Hint: using the "actors/list-born-today" endpoint retrieves a list of actors born in a same input day. Using the "actors/get-bio" endpoint retrieves actor details by passing an actor ID.

The screenshot shows the RapidAPI interface for the IMDb8 API. The 'GET actors/list-born-today' endpoint is selected and highlighted with a red box. The 'Test Endpoint' button is visible next to it. The 'GET actors/get-bio' endpoint is also highlighted with a red box. The 'Test Endpoint' button is visible next to it. The 'Test Endpoint' button is visible next to it.

Project Structure:

You need to have 6 php pages providing different **classes**:

- a. "Header.php": will include the header.
- b. "Footer.php": will include the footer.
- c. "Index.php": will be your main page and will include header and footer.
- d. "DB_Ops.php": will include your DB functions like connection and Insert and Select.
Also, it will handle any checking server side.
- e. "Upload.php": will include your code for processing the uploaded image.
- f. "API_Ops.php": will include all the necessary code to connect to the API **or** JavaScript
File "API_Ops.js": will include all the necessary code to connect to the API.

Hint:

- connecting to an API can be implemented using PHP or JavaScript, therefore you can use any of them.

Grading Rubrics:

The grading for this project will be broken down as follows:

1. Following the project structure and all the necessary pages (3 marks)
2. Required client-side validations listed and server-side validation listed (2 marks).
3. Including header and footer pages in index page (0.5 mark).
4. Using born-in the same day API (2.5 marks).
5. Inserting the user data in database (1 mark).
6. Supporting AJAX (1 mark).
7. Uploading the user image (**1 bonus mark**).