

# *Programmation Web*

## *A faire*

TD et TP :

- TD 1 framework CSS, exemple foundation, énoncé sur l'ENT
- **TD 2 : Programmation web : formulaire & PHP**
- TP 1 formulaire en HTML 5, CSS foundation énoncé sur l'ENT
- TP 2 ajouter du PHP et enregistrer les données dans un fichier XML bien formaté, énoncé sur l'ENT
- TP 3 ajouter du PHP et une base de données, énoncé sur l'ENT
- TP 4 conception de site avec MVC perso énoncé sur l'ENT

Les TP sont à rendre !

## TD 2 : Programmation web : formulaire & PHP

**Paul-Antoine BISGAMBIGLIA**

*Faculty of Science*

UMR-CNRS-6134

University of Corsica – 20250 Corte

*bisgambiglia@univ-corse.fr*

*Mots-clés : PHP, DB, html, javascript, MVC*

### Résumé

Vous avez une heure pour développer un petit module type forum.

## 1 Introduction

Il va falloir développer 2 pages : index.html et form.php. Elles doivent être dans le même dossier nommé nom\_prenom (exemple bisgambiglia\_paulantoine).

Il faudra poster sur l'ENT une archive (nom.prenom.zip) qui contient seulement ces deux fichiers.

Pour mes tests, votre code devra fonctionner avec ma base de données et mon fichier CSS, les instructions de connexion et les classes CSS sont données en annexe.

Le fichier **index.html** doit contenir un formulaire très simple qui demande à l'utilisateur un login. Il doit être impossible d'envoyer des données si le champ est vide, voir Figure 1.

Les données sont à envoyer en POST à la page form.php

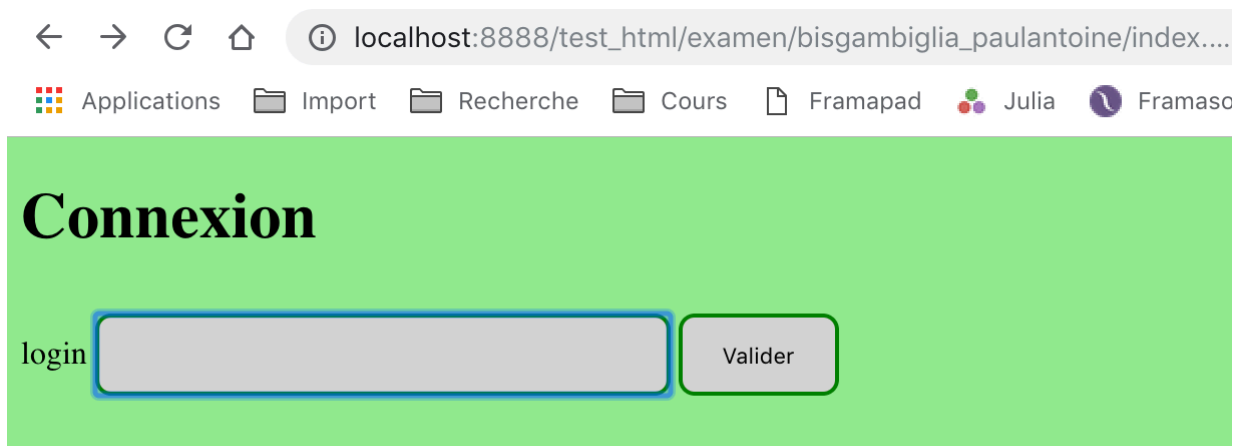
Le fichier **form.php** doit (c.f. Figure 2) :

- Se connecter à la base
- Afficher « hello \$LOGIN » si l'utilisateur est dans la base et sinon « hello \$LOGIN ajouté »
- Permettre à \$LOGIN d'ajouter un message
- Afficher les messages de la base de données

Bonus, le fichier form.php peut :

- Permettre à \$LOGIN de modifier uniquement ses messages
- Permettre à \$LOGIN de supprimer uniquement ses messages
- D'afficher les messages 10 par 10

## 2 Screens



The screenshot shows a web browser window with the address bar displaying `localhost:8888/test_html/examen/bisgambiglia_paulantoine/index...`. The browser's toolbar includes icons for Applications, Import, Recherche, Cours, Framapad, Julia, and Framaso. The main content area has a light green background and features the heading **Connexion**. Below the heading, there is a login form with a text input field labeled 'login' and a button labeled 'Valider'.

Figure 1 : `index.html`



The screenshot shows a web browser window with the address bar displaying `localhost:8888/test_html/examen/bisgambiglia_paulantoine/form...`. The browser's toolbar includes icons for Applications, Import, Recherche, Cours, Framapad, Julia, Framasoft, Apple, and GTrad. The main content area has a light green background and features the heading **Hello Jean**. Below the heading, there are two main sections: **Post** and **Read**. The **Post** section contains a text input field labeled 'New message:' and a button labeled 'Poster'. The **Read** section contains a table with alternating light green and light gray rows, displaying messages and their timestamps.

Paul a écrit le 2018-11-27
nouveau test
Toto a écrit le 2018-11-27
hello, je suis toto
Jean a écrit le 2018-11-27
est-ce que ça marche
Jean a écrit le 2018-11-27
est-ce que ça marche

Figure 2 : `form.php`

Hello Paul compte  
ajouté

## Post

New message:

Veuillez compléter ce champ.

## 3 Annexes

### 3.1 CSS

```
body {  
    background-color: lightgreen;  
}
```

```
.intro {  
    color: black;  
    text-align: left;  
}
```

```
input[type=text] {  
    width: 30%;  
    padding: 12px 20px;  
    margin: 8px 0;  
    box-sizing: border-box;  
    border: 2px solid green;  
    border-radius: 8px;  
}
```

```
input[type=text]:focus {  
    background-color: lightgrey;  
}
```

```
input[type=submit] {  
    background-color: lightgrey;  
    padding: 12px 20px;  
    margin: 8px 0;  
    box-sizing: border-box;  
    border: 2px solid green;  
    border-radius: 8px;  
}
```

```
div.un {  
    width: 320px;  
    border: 0px solid green;  
    margin: 0;  
    border-radius: 8px;
```

```

}

div.deux {
    width: 80%;
    padding: 10px;
    border: 5px solid green;
    margin: 0;
    border-radius: 8px;
}

table {
    border-collapse: collapse;
    width: 100%;
}

table, th, td {
    border: 1px solid green;
    padding: 15px;
    text-align: left;
    border-bottom: 1px solid green;
}

tr:hover {background-color: #f5f5f5;}
tr:nth-child(even) {background-color: #f2f2f2;}

```

## 3.2 Base de données

**Base** td\_php

**Table** user\_table

```

CREATE TABLE `td_php`.`user_table` ( `id_user` INT NOT NULL AUTO_INCREMENT , `login_user` varchar(30) NOT NULL , `date_crea` DATE NOT NULL , `date_con` DATE NOT NULL , PRIMARY KEY (`id_user`)) ENGINE = InnoDB;

```

**Données :**

```

INSERT INTO `user_table` (`id_user`, `login_user`, `date_crea`, `date_con`)
VALUES (NULL, 'paul', '2018-11-27', '2018-11-27');

```

**Table** message\_table

```

CREATE TABLE `td_php`.`msg_table` ( `id_msg` INT NOT NULL AUTO_INCREMENT , `user_msg` INT NOT NULL , `txt_msg` TEXT CHARACTER SET utf8 COLLATE utf8_bin NOT NULL , `date_msg` DATE NOT NULL , PRIMARY KEY (`id_msg`)) ENGINE = InnoDB;

```

**Données :**

```

INSERT INTO `msg_table` (`id_msg`, `user_msg`, `txt_msg`, `date_msg`) VALUES
(NULL, '1', 'bonjour, message de test de la table', '2018-11-27');

```

**Jointure :**

```
SELECT * FROM msg_table LEFT JOIN user_table ON msg_table.user_msg =  
user_table.id_user
```

### 3.3 Connexion

```
<?php  
$servername = "localhost";  
$username = "root";  
$password = "root";  
$port = 8889;  
$message = "";  
$greetings = "Hello ";  
  
try {  
    $conn = new PDO("mysql:host=$servername;dbname=td_php", $username,  
$password);  
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
    $message = "Connected successfully";  
}  
catch(PDOException $e)  
{  
    $message = "Connection failed: " . $e->getMessage();  
}  
?>
```