

Programmation web : formulaire + PHP + MVC

Paul-Antoine BISGAMBIGLIA

Faculty of Science

UMR-CNRS-6134

University of Corsica – 20250 Corte

bisgambiglia@univ-corse.fr

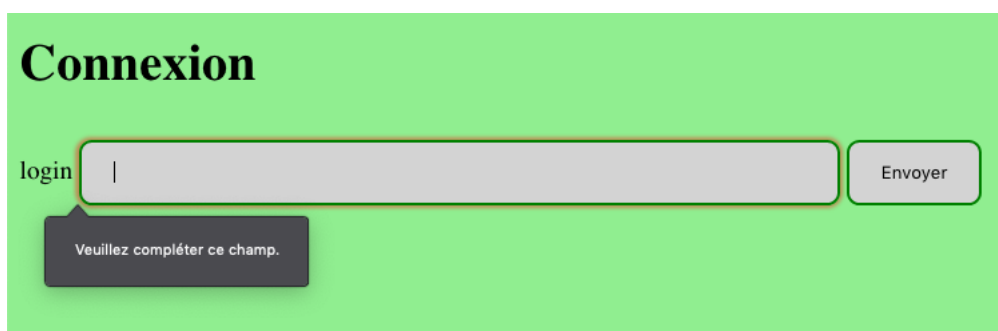
Mots-clés : PHP, DB, html, javascript, MVC

ETAPE 1

Pour commencer, il va falloir développer 2 pages : index.html et form.php. Elles doivent être dans le même dossier nommé nom_prenom (exemple bisgambiglia_paulantoine1).

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

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

ETAPE 2

Il va maintenant falloir développer et ajouter les pages :

- `admin.php` pour gérer les utilisateurs
- `compte.php` pour compléter son compte (photo de profil, mail, téléphone, nom, prénom, adresse, etc.)
- `cours.php` propose un ensemble de cours

Adaptez la base de données en ajoutant les tables permettant de construire une application web pour proposer les fonctionnalités suivantes, et notamment :

- Une (ou plusieurs) page(s) cours accessible après connexion pour s'inscrire à un cours (lister les cours) et voir ses notes et sa moyenne.
- Une (ou plusieurs) page(s) enseignant pour créer, modifier, supprimer un cours, voir ses élèves et mettre les notes.
- A la première connexion un message doit demander la modification du mot de passe.

Il y a trois enseignants à créer :

1. Paul Dupont enseigne le Pipo, et le Jonglage.
2. Eric Dupond enseigne la lutte, le yoga et le tire à l'arc
3. Stéphane Lefrançois enseigne la communication, et l'art de la guerre

Un élève suit des cours, il doit s'inscrire au moins à trois cours, et peut consulter ses notes.

Un enseignant peut créer un cours, et saisir les notes des élèves inscrits dans ses cours.

Un administrateur peut créer les comptes enseignants et étudiants, avec login : prénom et mot de passe : le nom

ETAPE 3

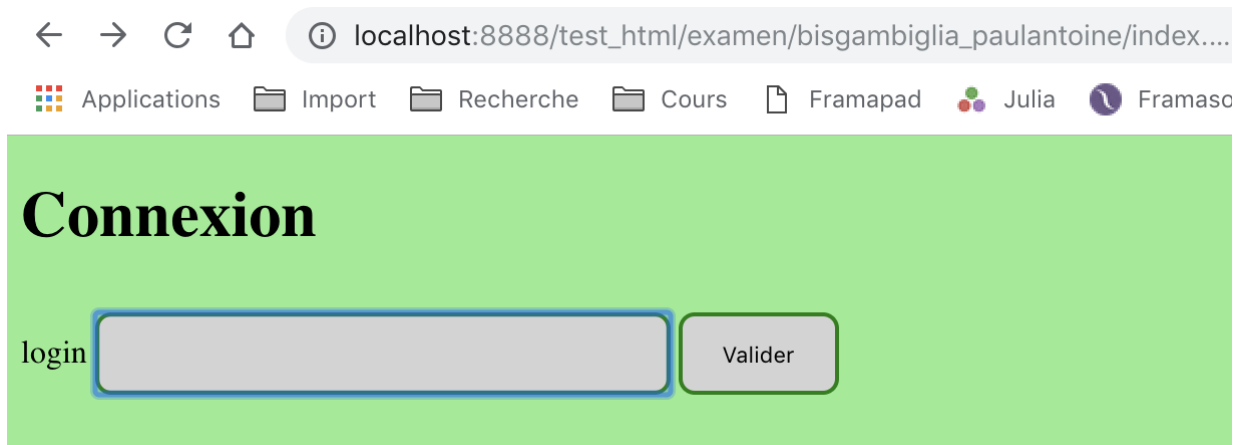
Dans un nouveau dossier (exemple `bisgambiglia_paulantoine2`), vous allez devoir recommencer tout le codage de vos pages et mettre en place le framework MVC.

Ce dernier va nécessiter le codage d'un modèle par table de la base de données. Un modèle est une classe qui modélise une table grâce à ses données et permet sa mise à jour grâce à ses méthodes.

Les contrôleurs vont faire le lien entre les différents formulaires et les modèles.

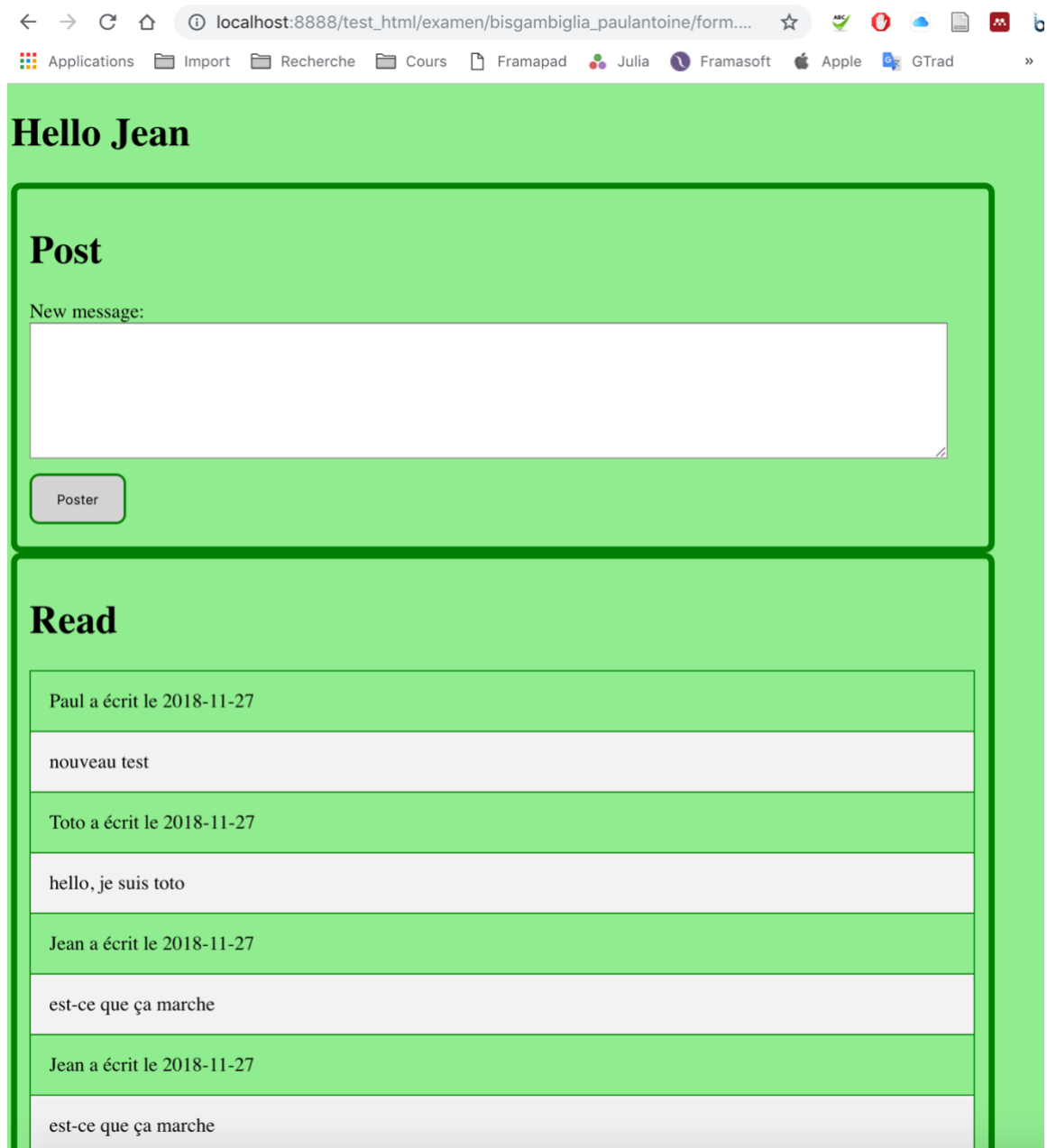
Généralement, ils vérifient et valident les données des formulaires avant d'instancier le bon modèle pour le mettre à jour ou créer un nouvel objet.

ANNEXES : Screens et code



The screenshot shows a web browser window with the address bar displaying 'localhost:8888/test_html/examen/bisgambiglia_paulantoine/index...'. The browser's top bar includes icons for Applications, Import, Recherche, Cours, Framapad, Julia, and Framaso. The main content area has a light green background and a large black 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 top bar includes icons for Applications, Import, Recherche, Cours, Framapad, Julia, Framasoft, Apple, GTrad, and a right arrow. The main content area has a light green background and a large black 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 list of messages, each with a header indicating the sender and date, and a body containing the message text.

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.

deconnexion

admin enseignant élève

Stéphane Lefrançais



Ajouter un cours

Nom

Descriptif

Envoyer

Modifier mes cours

Nom	Description	Action
Lecture	il faut beaucoup lire pour etre bon	<input type="checkbox"/>
Base de données	SQL et noSQL	<input type="checkbox"/>
RPG en php	good game	<input type="checkbox"/>
Modifier		

Supprimer un cours

Nom	Description	Créateur	Action
Lecture	il faut beaucoup lire pour etre bon	Stéphane Lefrançais	<input type="checkbox"/>
Base de données	SQL et noSQL	Stéphane Lefrançais	<input type="checkbox"/>
RPG en php	good game	Stéphane Lefrançais	<input type="checkbox"/>
Supprimer			



Tes cours :

Nom	Description	Enseignant	Note
programmation	apprendre à parler à un pc	Paul Dupont	
pipo	top le pipo, le pipo c'est cadeau	Eric Dupond	15
Lecture	il faut beaucoup lire pour etre bon	Stéphane Lefrançais	
programmation	apprendre à parler à un pc	Paul Dupont	
MySQL	vive les bases de données	Eric Dupond	

S'inscrire :

Nom	Description	De	Action
programmation	apprendre à parler à un pc	Paul Dupont	<input type="checkbox"/>
Lecture	il faut beaucoup lire pour etre bon	Stéphane Lefrançais	<input type="checkbox"/>
Base de données SQL et noSQL		Stéphane Lefrançais	<input type="checkbox"/>
RPG en php	good game	Stéphane Lefrançais	<input type="checkbox"/>
MySQL	vive les bases de données	Eric Dupond	<input type="checkbox"/>
pipo	top le pipo, le pipo c'est cadeau	Eric Dupond	<input type="checkbox"/>



Add user

Login	Mot de passe	Rôle
<input type="text" value="votre login"/>	<input type="text" value="mdp de + de 6 char"/>	<input type="text" value="eleve"/>
<input type="button" value="Envoyer"/>		

Date de connexion	Login	Rôle	Action
2019-12-10 10:02:48	Pierre-Paul Blanc	eleve	<input type="button" value="supprimer"/>
2019-12-10 10:02:35	Jean-Pierre Lelac	enseignant	<input type="button" value="supprimer"/>
2019-12-05 10:16:30	jacques	eleve	<input type="button" value="supprimer"/>
2019-12-05 10:12:34	Eric Dupond	enseignant	<input type="button" value="supprimer"/>
2019-12-05 11:15:00	Stéphane Lefrançais	enseignant	<input type="button" value="supprimer"/>
2019-12-05 10:03:23	Paul Dupont	enseignant	<input type="button" value="supprimer"/>

< précédent <

> suivant >

1 Code

1.1 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;}

```

1.2 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

```

1.3 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();
}

?>
```