

Bonjour à tous,

Voici le déroulé du cours de Programmation Web (Intro DFS)

L'objectif du cours du semestre est de vous faire comprendre les différentes méthodes de développement web et de renforcer vos compétences dans le domaine.

Vous aurez une note de participation, c'est-à-dire de présentation de vos codes, une note de TP, le sujet est en fin de mail et une note d'examen.

L'examen aura lieu en janvier, c'est du commentaire de code et/ou des questions sur des définitions de cours.

Table des matières

I. Déroulement des cours :	1
II. Les types de programmation web :	5
III. Le roulement :	5
IV. Les rendus et la Progression :	6
V. Sujet du TP noté :	7

I. Déroulement des cours :

jeu. 18/9 : Introduction du module // cours d'intro à l'UX/UI // exercices.

Exercice fil rouge => coder un système de login avec le backend en python (**Django, Flask, FastAPI**, puis faire la même chose en PHP (sans framework).

L'exercice est à faire seul. Vos codes doivent être fonctionnels, testés, commentés et déposés sur un GITxxx. **Ils seront présentés le 22/09 avec une présentation et une justification de votre choix de techno.**

Info base de données à utiliser :

```
CREATE TABLE `2025_M1`.`user` (`user_id` INT NOT NULL AUTO_INCREMENT ,  
`user_login` TEXT NOT NULL , `user_password` LONGTEXT NOT NULL ,  
`user_compte_id` INT NOT NULL , `user_mail` TEXT NOT NULL ,  
`user_date_new` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ,  
`user_date_login` TIMESTAMP on update CURRENT_TIMESTAMP NOT NULL  
DEFAULT CURRENT_TIMESTAMP , PRIMARY KEY (`user_id`), UNIQUE `cle-  
etrangere` (`user_compte_id`)) ENGINE = InnoDB;
```

Les dates sont gérées automatiquement, le champ 'user_compte_id' va permettre de faire le lien avec le compte client (clé étrangère) de l'utilisateur. Le compte client sera à ajouter par la suite pour gérer le panier et les commandes.

lun. 22/9 : Présentation des codes // Cours HTML - CSS // Exercices.

Exercice fil rouge => coder un CRUD pour une gestion des utilisateurs et des produits (ajouter un produit, le supprimer, le modifier).

Exercice à faire en groupe de 6 en suivant les modes de développement :

1. Mode API, (2 membres du groupe)
2. Mode MVC, (2 membres du groupe)
3. Mode classique, (2 membres du groupe)

Les codes seront présentés le 14/10.

```
CREATE TABLE produit (  
  id_p INT AUTO_INCREMENT PRIMARY KEY,  
  type_p VARCHAR(100) NOT NULL,  
  designation_p VARCHAR(255) NOT NULL,  
  prix_ht DECIMAL(10,2) NOT NULL,  
  date_in DATE NOT NULL,  
  times_in TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ,  
  stock_p INT DEFAULT 0  
);
```

Vous pouvez compléter cette table, pour gérer l'historique des prix d'un produit et aussi proposer une promotion, il faut donc ajouter un champ pourcentage de promotion (ppromo), s'il n'est pas complété le produit est à afficher à son prix

hors taxe et s'il y a une valeur, on affiche la promotion et on calcule le nouveau prix.

Ajouter aussi la possibilité de déposer une image.

Si aucune image n'est liée au produit, afficher une image générique.

Un produit est un élément à vendre sur votre site, vous pouvez choisir les désignations. Vous devez ajouter des fonctions pour gérer les produits.

Pour les types de produit, vous devez faire un fichier .json coté serveur avec la liste des types (categorie) et côté client, il faudra afficher un select html.

```
[
  {
    "id": 1,
    "categorie": "Smartphone"
  },
  {
    "id": 2,
    "categorie": "Ordinateur portable"
  },
  {
    "id": 3,
    "categorie": "Casque audio"
  },
  {
    "id": 4,
    "categorie": "Montre connectée"
  },
  {
    "id": 5,
    "categorie": "Tablette"
  },
  {
    "id": 6,
    "categorie": "Caméra"
  }
]
```

Pour le développement en API, voici les fonctions à implémenter :

GET : La méthode GET demande une représentation de la ressource spécifiée. Les requêtes GET doivent uniquement être utilisées afin de récupérer des données.

HEAD : La méthode HEAD demande une réponse identique à une requête GET pour laquelle on aura omis le corps de la réponse (on a uniquement l'en-tête).

POST : La méthode POST est utilisée pour envoyer une entité vers la ressource indiquée. Cela entraîne généralement un changement d'état ou des effets de bord sur le serveur.

PUT : La méthode PUT remplace toutes les représentations actuelles de la ressource visée par le contenu de la requête.

DELETE : La méthode DELETE supprime la ressource indiquée.

Lun. 22/septembre : point sur l'exercice fil rouge et point cours

mar. 14/10 : Présentation des codes // Cours SEO // exercices.

Exercice de cours => **faire un crawler (robot d'indexation) avec CURL ou bs4** (<https://www.scrapingbee.com/blog/crawling-python/>)

Exercice fil rouge => présentation des codes de CRUD et **premier roulement** :

- **pour le cours suivant => Ajouter dans le CRUD les fonctionnalités suivantes : système de Sign Up / Log In (couplage des codes du premier cours avec le code de votre CRUD).**

Les codes seront présentés le 10/11.

Mardi 21/10 et jeudi 23/10 : point sur l'exercice fil rouge et point cours

Exercice fil rouge => présentation des codes de CRUD et ajout d'un **BackOffice** (Une interface pour gérer les utilisateurs et les produits).

lun. 10/11 : Cours JS // exercices.

Exercice fil rouge => présentation des codes « Sign Up / Log In »

- pour le cours suivant => Ajouter les fonctionnalités suivantes : système de Trie par type de produit (ajouter des objets avec faker <https://www.docstring.fr/blog/generer-des-donnees-aleatoires-avec-faker/>)
- Deuxième roulement.

13, 14/11: point sur l'exercice fil rouge et point cours

Exercice fil rouge => ajouter un papier, il doit être possible d'ajouter des produits, de modifier la quantité et de voir l'évolution dynamique du montant, en ajoutant la TVA et des frais d'expédition.

lun. 17/11 : Cours API + exercice

Exercice fil rouge => présentation des codes de « Trie »

mer. 18/12 : Présentation et **notes** des codes de TP (le sujet de TP est en fin de mail)

II. Les types de programmation web :

Il y a trois types de programmation : 1. Mode API, 2. Mode MVC, 3. Mode classique.

Le Mode API implique de développer un backend qui suit les recommandations REST ou SOA. Il s'agit de développer plusieurs fonctions qui seront appelées par une méthode http (POST, GET, UPDATE, PUT, etc.). Elles devront renvoyer un fichier XML, JSON ou du texte.

Le Mode MVC implique de suivre ce pattern de conception avec un modèle qui gère les interactions avec la base de données, un contrôleur qui fait les liens entre les vues et les modèles et des vues consultables par l'utilisateur.

Le mode classique implique de coder sans modèle, les interactions entre requêtes du client et la base de données peuvent être codées dans les mêmes fichiers.

III. Le roulement :

Vous allez vous regrouper par 6, en trois équipes de 2, et dans chaque équipe faire 3 binômes.

Explication par l'exemple, pour chaque équipe :

- Le binôme 1 va faire le premier exercice en développant une API (1. Mode API)
- Le binôme 2 va faire le premier exercice en suivant le pattern MVC (2. Mode MVC)
- Le binôme 3 va faire le premier exercice en développant sans modèle from scratch (3. Mode classique)

A chaque nouveau cours, vous allez vous transmettre vos codes afin de travailler sur toutes les approches de programmation web, c'est ça le roulement, et c'est pour ça que le code est présenté en cours et qu'il doit être bien commenté.

IV. Les rendus et la Progression :

- 1) Bases et mise en place (déjà acquis)
 - a) Cours : rappels web (HTTP, client/serveur, API), Git/GitHub, introduction à HTML/CSS/JS modernes.
 - b) TP : création d'une page produit statique (HTML/CSS), mise en ligne sur GitHub Pages.
- 2) Front-end dynamique (septembre)
 - a) Cours : introduction à React (ou Vue), composants, état, routing.
 - b) TP : catalogue produits en React avec navigation entre catégories.
- 3) Back-end et base de données (octobre)
 - a) Cours : Node.js + Express, REST API, CRUD, PostgreSQL/MongoDB.
 - b) TP : API de gestion des produits (ajouter, modifier, supprimer, lister).
- 4) Authentification & panier (novembre)
 - a) Cours : sessions, JWT, sécurité (hashage mots de passe, XSS/CSRF).
 - b) TD : inscription, connexion utilisateur, panier d'achat stocké côté serveur.
- 5) Optimisation & SEO (décembre)
 - a) Cours : SSR avec Next.js, SEO (balises, sitemap), performance (lazy loading, code splitting).
 - b) TP : transformer le front en Next.js, améliorer SEO du site e-commerce.
- 6) Déploiement & DevOps (lien avec le cours de DevOps)

- a) Cours : Docker, CI/CD, hébergement (Vercel/Netlify pour le front, Render/Heroku pour l'API).
- b) TP : déployer l'application complète en ligne.

V. Sujet du TP noté :

Faire un jeu simple perso 100% JS et 100% perso, l'utilisation de code existant est proscrit.

Faire la gestion des comptes (Log In) et des scores en Python ou en PHP sans framework.

La partie front doit être réalisée avec les technologie suivante : Twing => templating HTMLX, astro, Tailwind CSS.

Bien cordialement,

Paul-Antoine BISGAMBIGLIA

Maître de Conférences - HDR en informatique

Université de Corse PP

CNRS UMR SPE 6134

bisgambiglia_pa@univ-corse.fr

@pabisgambiglia