

# Projet de développement Web

## A. Présentation du projet

Dans le cadre de ce projet, vous allez réaliser une application Web de votre choix.

Cette application devra répondre au **besoin d'un client**. Cela signifie que vous devez trouver une personne qui exprime un besoin et qui testera votre application. Cette dernière devra être utile, agréable et efficace à utiliser. Il est important que le sujet choisi vous plaise et vous motive.

Vous devrez développer le **front-end** (interface utilisateur) et le **backend** (traitements serveur) de l'application Web. Votre backend qui se trouvera sur un **serveur** doit avoir l'usage d'une **base de données** et être accessible via une **API**. Votre application devra être efficacement **testée** (tests unitaires et d'intégration).

A priori il s'agira d'un **site internet** (responsive), mais il est envisageable de se diriger vers une **application mobile** (ou les deux, ou une Progressive Web App) si vous faites bien intervenir les technologies web.

Vous serez en grande partie libre de vos **choix technologiques**. Des explications, des workshops et des ressources en ligne seront proposés durant le projet, mais il relève de votre responsabilité **de vous renseigner et de vous former** dans les technologies utilisées.

Ce projet s'étalant sur tout le quadrimestre, un résultat conséquent avec un **travail régulier** est attendu. De nouvelles technologies sont à apprendre et à mettre en place, cela demande du temps et de l'investissement. Il est conseillé de vous concentrer sur la **qualité** du développement et du résultat ainsi que sur la **bonne compréhension** des technologies utilisées, plutôt que sur la quantité.

## B. Contraintes

Ce projet est à réaliser en groupe de 3 étudiants. Choisissez comme partenaires des étudiants avec qui vous pourrez travailler efficacement, et qui présentent des compétences complémentaires par rapport aux vôtres. Chaque membre du groupe devra s'impliquer techniquement dans le projet. (Exemple : il ne faut pas qu'un membre s'occupe exclusivement de tâches liées à la gestion du groupe ou du client.) Il est également attendu à ce que chaque membre comprennent lui-même les différents aspects de son projet (idéalement chacun travaille sur les différentes parties).

Au niveau des outils, vous devrez obligatoirement utiliser:

- un **gestionnaire de source** (Github ou Gitlab) et effectuer vos commits **personnels** très régulièrement. Ces commits seront utilisés pour mesurer la contribution de chaque membre du groupe.
- un **gestionnaire de tâches** (type Trello ou Taiga) permettant de visualiser l'avancement du projet
- un **gestionnaire de suivi du temps de travail** (type Clockify) permettant de connaître le temps passé par chacun sur les différents aspects du projet

## C. Critères de choix de la problématique

Les besoins fonctionnels pouvant fort varier d'une problématique à l'autre, votre sujet de projet doit impérativement être validé par un enseignant avant de se lancer dans la mise en œuvre. N'hésitez pas à venir discuter de vos idées/envies dès que possible.

A priori, votre application devrait au minimum contenir :

- un système de comptes utilisateurs
- des interfaces dynamiques, des formulaires
- de l'insertion et de l'affichage de données
- des envois d'emails ou de notifications
- éventuellement du chargement ou téléchargement de fichiers
- éventuellement l'utilisation de services tiers en ligne (ex: google map, récupérations de données météo, de logging Facebook/Google, de vérification d'humain, etc...)
- un service de mesure de l'audience de votre site web est intéressant (Google Analytics ou alternatives).

Vous devrez fournir un document expliquant clairement la problématique de votre client et l'objectif et l'intérêt de votre application web. En plus de cette description textuelle claire, vous devrez fournir les "**user stories**" de votre application pour bien définir clairement quelles seront les fonctionnalités à implémenter.

## D. Critères technologiques à respecter

- **Choix des technologies**

Les technologies utilisées dans la mise en œuvre de votre application sont laissées à votre libre choix, à la condition que ces choix soient présentés et justifiés à l'équipe enseignante. Celle-ci devra valider ces choix avant la mise en œuvre.

L'appropriation et l'utilisation de technologies modernes, utiles et efficaces, seront valorisées lors de l'évaluation.

Vous trouverez un peu plus bas des exemples de technologies envisageables. N'hésitez pas à éventuellement en proposer d'autres et en discuter avec les enseignants.

Vous devrez fournir un document **expliquant** quels sont les technologies que vous avez choisies, quelles sont leurs particularités et les **raisons de ces choix**.

Dans le cadre de ce projet, vous pouvez expérimenter une technologie par curiosité, mais vous devez au préalable être capables d'expliquer ses forces et ses faiblesses ainsi que les différences par rapport à d'autres technologies envisageables.

- **Architecture**

L'architecture de votre application devra reposer sur une Web API qui définira la manière dont le backend et le frontend interagissent. Cette API devra être correctement documentée, au niveau des URLs utilisées et de la manière dont les requêtes sont effectuées.

- Workflow de développement

Une bonne approche DevOps pour la gestion de votre projet est une belle plus value pour votre projet, avec une bonne gestion des étapes de développement, intégration, test et déploiement.

Un environnement de développement virtuel peut s'avérer intéressant (Vagrant, ...).

- Liste de technologies envisageables (non limitante) :

#### *Front-end*

- React
- Vue
- JQuery
- Bootstrap, Materialize, Foundation, ...
- templating (twig, ...)
- scss, sass, ... / flexbox, grid, ...
- Angular
- ...

#### *Back-end*

- Node.js
- PHP (existe frameworks Symfony, Laravel, ...)
- Python (existe frameworks Django, Flask, ...)
- Ruby on Rails
- .NET
- Java Spring
- ...

#### *Données*

- SQL (PostgreSQL, MariaDB, MySQL, ...)
- NoSQL [justifier alors l'emploi !] (MongoDB, ...)
- ...

(Possibilité d'utilisation d'ORM.)

#### *Testing*

Dépendra des technologies employées pour vos tests unitaires.

Eventuellement tests d'intégration automatisés (Selenium, ...)

(Utilisation d'outils peut s'avérer pratique pour tester votre API : curl, postman...)

#### *Hébergement*

L'hébergement choisi dépendra des technologie employées

A priori, vous pouvez utiliser votre propre serveur (sur VPS). Vous aurez l'occasion de l'installer et le configurer dans le projet d'Administration Système et Réseaux. Il pourrait être intéressant d'utiliser Docker dans ce cadre.

D'autres options sont également envisageables : services cloud (azure, amazon, heroku) , Firebase ou alternative, ...

#### IDE

Il est important de développer dans de « bonnes conditions ». De très nombreux outils existent, idéalement intégrés avec votre gestionnaire de source. Le choix du langage peut bien évidemment influencer le choix de l'IDE. Quelques possibilités : Visual Studio, PHPStorm (gratuit pour étudiant ephec), Eclipse, NetBeans, Pycharm, Atom, RJ TextEd, etc., etc...

#### Sécurité

Quels que soient les choix technologiques effectués, vous devrez assurer la sécurisation de votre réalisation, en évaluant les risques et les vulnérabilités possibles et en mettant en place les contre-mesures appropriées.

### E. Échéances :

Vous serez coachés et **évalués tout au long de l'avancement** du projet. Un calendrier définissant les dates à respecter pour la réalisation de différentes étapes sera disponible sur le Moodle du cours, et les modalités seront précisées au fur et à mesure.

### F. Evaluation :

La note du projet sera constituée de deux éléments :

- L'évaluation continue, à savoir les évaluations intermédiaires
- L'évaluation finale, à savoir la qualité et l'ampleur de l'application finale et la défense du projet

**Attention** : la **cote** finale du projet sera le cas échéant **individualisée** en fonction de l'apport et de la maîtrise de chacun.