



Projet de fin d'étude / Soutenance

Objectif

Le projet de fin de formation doit permettre de mettre en pratique le maximum de notions vues au cours de la formation. Il doit également servir à expérimenter de manière concrète la collaboration au sein d'une équipe de développeurs.

Taille et composition des équipes

3 développeurs : bonne taille d'équipe

4 développeurs : taille d'équipe idéale, permet de faire de la vraie gestion de projet avec répartition des tâches, répartition des rôles, etc.

5 développeurs : grand maximum car à 5 ce sera déjà compliqué de répartir le travail de manière équitable et mettre tout le monde d'accord

Les équipes sont choisies par les étudiants, sauf si le formateur juge utile de les composer lui-même.

Types de projets et validation du formateur

Chaque équipe doit réfléchir au projet qu'elle souhaite développer tout en gardant à l'esprit les recommandations suivantes :

- Le projet devra faire appel à des notions de Back-office et de Front-office
- Une base de données devra être utilisée
- La partie Front-office devra être intégrée en utilisant les techniques du Responsive Web Design
- Le code PHP devra être organisé selon l'architecture MVC en utilisant le framework pédagogique WebForce3 (ou un autre framework)
- Le code PHP devra être orienté objet
- Le code JavaScript devra faire appel à au moins une API externe, ou à défaut, intégrer des fonctionnalités AJAX
- Le code HTML/CSS devra être valide et accessible
- Le projet devra permettre de mettre en pratique quelques notions liées à la sécurité, par exemple le hashage des mots de passe ou encore des mesures permettant d'éviter des attaques de type "Injection SQL"

En dehors de ces recommandations, les projets peuvent porter sur n'importe quel sujet qui intéresse l'équipe (cf. "Exemples de projets").

Une fois trouvée l'idée de projet, les étudiants la transmettent au formateur en lui détaillant :

- Le nom du projet s'il a déjà été trouvé
- Le but du site ou de l'application web qu'ils souhaitent réaliser
- Les fonctionnalités principales

- Les recommandations que l'équipe s'engage à mettre en pratique
- Les membres de l'équipe et les grandes lignes de la répartition des rôles

Sur la base de ces informations, le formateur validera le projet, proposera des modifications ou le déconseillera fortement s'il s'écarte de manière trop importante des recommandations et de l'esprit général des projets tel qu'il est décrit dans ce document.

Dans le cas où l'équipe ne parvient pas à trouver un projet, par manque d'idées ou faute de consensus, le formateur mettra à disposition de celle-ci le cahier des charges d'un projet prédéfini.

Périmètre fonctionnel du projet

La liste complète des fonctionnalités que l'équipe souhaite intégrer au projet représente le périmètre fonctionnel. Celui-ci doit être réalisable dans le temps à disposition, soit une quinzaine de jours.

Il est fondamental de ne pas surcharger le projet de fonctionnalités que l'équipe n'aura pas le temps de développer.

Voici une méthode qui peut vous aider à déterminer une liste de fonctionnalités réalisable :

- Brainstorming général afin que tous les membres de l'équipe puissent proposer des fonctionnalités
- Pour chaque fonctionnalité on nomme un développeur qui en serait le responsable
- Chaque développeur fait une estimation du temps qu'il mettrait à développer chacune des fonctionnalités dont il est le responsable (cf. "Estimation du temps nécessaire pour chaque fonctionnalité")
- Grâce à l'estimation du temps disponible, l'équipe est en mesure de retenir uniquement les fonctionnalités qui peuvent être développées de manière réaliste
- Estimation du temps disponible

Le temps disponible correspond au nombre d'heures que chaque développeur pourra consacrer au développement du projet. Il se détermine en additionnant :

- les heures en classe en présence du formateur
- les heures que chacun est prêt à mettre dans le projet en soirée (peut être variable d'un jour à l'autre en fonction des contraintes de chacun)
- les heures sur les jours fériés ou weekends (peut être variable d'un jour à l'autre en fonction des contraintes de chacun)

Conseil important : tout le temps disponible ne doit pas être consacré au développement des fonctionnalités. D'une manière générale, il faut compter 4/5 du temps disponible pour le développement des fonctionnalités et 1/5 pour :

- imprévus
- debuggage

- petites fonctionnalités non planifiées mais indispensables
- préparation de la présentation du projet
- Estimation du temps nécessaire pour chaque fonctionnalité

Pour estimer le temps nécessaire à la réalisation d'une fonctionnalité, il peut être utile de découper la fonctionnalité en sous-parties. Par exemple, pour une fonctionnalité "Formulaire d'inscription avec création d'un compte utilisateur" :

- Réflexion sur la liste des champs à faire figurer sur le formulaire
- Intégration HTML/CSS du formulaire
- Validation des champs en JavaScript et gestion des erreurs
- Validation des champs en PHP et gestion des erreurs
- Hashage du mot de passe
- Stockage des informations dans la base de données
- Envoi d'un mail permettant de valider l'adresse mail
- Envoi d'un mail de confirmation
- Tests et évaluation de l'accessibilité du formulaire
- Vérification de la validité du code HTML/CSS
- Tests de fonctionnement par un membre de l'équipe n'ayant pas contribué au développement du formulaire

Chaque sous-partie est estimée individuellement et la somme permet de déterminer le temps de la fonctionnalité.

Conseil très important : quand on pense avoir déterminé le temps nécessaire au développement d'une fonctionnalité, on le multiplie par 2 ! :) Il est largement préférable de se tromper en excès et avoir du temps en plus pour développer de nouvelles fonctionnalités plutôt qu'en défaut et devoir bâcler le projet et faire l'impasse sur des fonctionnalités qui avaient initialement été prévues.

Exemples de projets

À titre d'exemple, voici des projets réalisés par des étudiants :

- Site d'offres d'emploi permettant la mise en relation entre candidats et entreprises
- Site de mise en relation entre touristes et artisans locaux
- Proposition d'extranet pour la WebForce3
- Plateforme inspirée de Stackoverflow permettant aux étudiants WebForce3 de valoriser l'aide qu'ils pouvaient apporter à d'autres étudiants
- Site d'une bibliothèque en ligne, avec catalogue, livraison en point-relais et gestion de paiements de retard
- Site inspiré d'Allo-Ciné, avec listes utilisateurs de "films à voir"
- Clone de Twitter
- Commercialisation des projets et propriété intellectuelle
- Site de carte de naissance
- Site d'un humoriste
- Site inspiré d'Apolearn

Versioning

Tous les projets devront utiliser Git pour leur versioning. Chaque équipe est libre de choisir la plateforme qu'elle préfère (GitHub, Bitbucket, etc.).

À l'intérieur de chaque équipe, chaque développeur est libre d'utiliser le client Git qu'il préfère qu'il soit en ligne de commande ou avec interface graphique (GitHub, SourceTree, etc.).

Chaque équipe doit donner un accès au dépôt (au moins en lecture) au formateur.

Soutenance : Présentation finale

Lors du dernier jour de la formation, les équipes présenteront leur projet devant un jury. Il sera composé au minimum d'un membre de l'équipe pédagogique, un membre de la direction WebForce3 et d'un membre externe à l'école. Les autres élèves de la classe seront également présents.

Ces présentations dureront entre 35 et 45 minutes par équipe selon la composition. Le temps de parole devra être réparti équitablement entre chaque membre de l'équipe.

Vous déroulerez votre présentation selon les critères suivants :

1. Description du projet (avec ou sans slides)
2. Présentation des différents membres de l'équipe et leurs rôles respectifs
3. Présentation des étapes de production collective, les outils et technologies utilisés, justifiez les choix techniques
4. Démonstration
5. Difficultés rencontrées au cours du projet
6. Éventuelles perspectives d'évolution du projet après la formation
7. Questions/réponses avec le public