

Projet V5.0.0



Cette œuvre est mise à disposition selon les termes de la [licence Creative Commons Attribution – Pas d'Utilisation Commerciale – Partage à l'Identique 3.0 non transposé](#).

Document en ligne : www.mickael-martin-nevot.com

Date de rendu : **29 octobre 2018 à minuit**

Travail : **groupe de quatre**

1 Assistance

Vous pouvez contacter l'enseignant en cas de besoin en formalisant et en ciblant précisément votre demande. Pour ce faire, vous devez respecter les règles de communication et d'envoi (ci-dessous).

2 Communication et envoi

2.1 Généralités

En joignant vos coordonnées (*e-mail* et téléphone portable notamment) à un message ou à votre livraison, vous pourrez être joint en cas de problème.

2.2 Communication

Chaque communication devra être faite :

- à destination de votre enseignant responsable :
 - Olivier Gérard : oliv.gerard@gmail.com ;
 - Philippe Mondou : philippe.mondou@gmail.com ;
 - Axel Prat : axelprat@gmail.com ;
 - Mickaël Martin-Nevot : mmartin.nevot@gmail.com ;
- en faisant figurer [AMU][IUT][S3] en début de sujet.

2.3 AMeTICE

2.3.1 Présentation

Pour votre projet, vous devez utiliser la plateforme pédagogique de l'Université d'Aix-Marseille : **AMeTICE** (<http://ametice.univ-amu.fr>), à l'adresse suivante : Accueil / Cours / IUT / DUT – secteur Électronique – Informatique Mécanique / DUT – Informatique (INFO) – Aix-en-Provence / 2^e année / [18]–S3–WIF200–201–M3104– Programmation Web côté serveur (M. Martin-Nevot).

2.3.2 Sélection du groupe

Sélectionnez, sur AMeTICE, le **groupe** d'étudiant auquel vous avez été affecté.

2.3.3 Livraison

Votre livrable devra être :

- nommé de la manière suivante (Nom1, Nom2, Nom3, Nom4 étant vos noms et Prénom1, Prénom2, Prénom3, Prénom4 vos prénoms) : Nom1 Prénom1 – Nom2 Prénom2 – Nom3 Prénom3 – Nom4 Prénom4 ;
- compressé dans une seule archive au format ZIP **n'excédant pas 10 Mo** ;
- remis, avant la date de rendu, sur AMeTICE.

3 Disponibilité

Votre site Web doit rester disponible jusqu'à la fin de votre année universitaire de formation. En outre, l'ensemble des identifiants de connexion du site Web et de la base de données doivent également rester valides durant toute cette période.

4 Sujet

4.1 Présentation

Cook & Burn, toute jeune startup française, conçoit des barbecues connectés.

Afin d'accélérer son développement elle vous confie la réalisation d'une petite application Web dédiée à la création et au partage des recettes des membres. L'interface permettra aux utilisateurs, quel que soit leur terminal (mobile / tablette / bureau), de naviguer simplement et rapidement.

Ces utilisateurs sont des membres disposant d'un compte (créé et activé préalablement lors de l'achat d'un barbecue de la marque), qui peuvent donc créer et partager leurs propres recettes.

Les recettes sont notées par les utilisateurs et se voient attribuer des « *burns* » par ceux-ci. Une recette ayant dix *burns* devient publique et visible par les visiteurs du site Web ne disposant pas de compte.

4.2 Organisation générale de l'application

4.2.1 Page d'accueil

La page d'accueil doit présenter les caractéristiques suivantes :

- une description du service ;
- en Une, la dernière recette à avoir reçu quinze *burns* ;
- la liste des recettes accessibles selon le rôle de l'utilisateur – détails ci-dessous (avec une pagination allant de deux en deux par défaut mais paramétrable par le super administrateur) ;
- chaque recette pointe vers une page dédiée ;
- un système de connexion/déconnexion avec lien vers le compte utilisateur, lequel affiche les informations de celui-ci et la possibilité de les modifier ;
- un moteur de recherche pour trouver une recette via des mots clefs (recherche dans les titres, descriptions, ingrédients) ;

- un système de **mot de passe oublié ou perdu** (au choix : possibilité de réinitialiser un nouveau mot de passe et/ou par *e-mail*, génération automatique d'un mot de passe temporaire ou non, etc.).

4.2.2 Page recette

La page recette doit présenter les caractéristiques suivantes :

- toutes les informations permettant de réaliser la dite recette ;
- un système d'attribution des *burns* pour les membres ;
- les membres connectés peuvent mettre en favori certaines recettes ; accessibles ensuite très rapidement ;
- **(optionnel)** la possibilité de partager les recettes sur les principaux réseaux sociaux : si la recette n'est pas publique, le lien fera apparaître le contenu tronqué et l'utilisateur sera invité à se connecter pour accéder au contenu (via création d'un compte ou une connexion via le compte du réseau social utilisé).

4.3 Détail des fonctionnalités

4.3.1 Comptes utilisateurs

La page compte utilisateur doit présenter les caractéristiques suivantes :

- nom (ou pseudo) ;
- *e-mail* ;
- mot de passe (justifiez les mesures de sécurité mises en œuvre dans votre fichier `lisez-moi.txt`).

4.3.2 Rôles

Voici les différents rôles :

- (super) administrateur (tous les droits) ;
- membre **Cook & Burn** (disposant d'un compte avec des privilèges limités) ;
- visiteur (tous publics).

4.3.3 Privilèges/rôles

Voici la répartition des rôles par rapport aux privilèges :

(Super) administrateur	Tous les privilèges : <ul style="list-style-type: none"> – CRUD¹ Utilisateurs, recettes, ingrédients etc. ; – ne peut pas modifier les <i>burns</i> attribués à une recette existante ; – ne peut pas modifier le mot de passe d'un membre.
Membres	<ul style="list-style-type: none"> – CRUD recettes lui appartenant ; – CRUD ingrédients lui appartenant ; – R recettes des autres ; – R ingrédients des autres ; – <i>C burn</i> ; – CRUD favoris lui appartenant.
Visiteurs	<ul style="list-style-type: none"> – R recettes publiques.

¹ CRUD : *create, read, update, delete*.

4.3.4 Données

Jeu de test des recettes

Voici, **en respectant la consigne à la lettre !** le jeu de test de 18-20 recettes que vous devez mettre en place :

- 10 recettes créées par au moins 3 utilisateurs différents ;
- 2 recettes disposant de 15 *burns* ;
- 2 recettes disposant de 10 *burns* ;
- 2 recettes disposant de 9 *burns* ;
- 2 recettes à 0 *burn* ;
- **(optionnel)** 1 recette enregistrée mais non soumise (brouillon utilisateur) ;
- **(optionnel)** 1 recette en mode privé (accessible uniquement au membre qui l'a créé).

Recette

Voici les caractéristiques d'une recette :

- nom ;
- nombre de convives ;
- description courte ;
- description longue ;
- ingrédients ;
- un déroulé sous forme d'étapes ;
- *burns* attribués ;
- **(optionnel)** statut (brouillon, privé, publique).

Partage réseaux sociaux

Suivez les indications des fournisseurs via la documentation de leurs API pour procéder aux tests.

4.4 Evènement perturbateur

(optionnel) Un élément peut venir perturber le bon déroulement de l'avancement de votre projet à tout moment : malheureusement il est imprévisible, aussi bien sur sa date d'apparition que sur son objet. Vous savez seulement que vous devrez alors adapter l'application.

5 Réalisation

Vous devez réaliser, **sans** système de gestion de contenu préexistant (CMS open source) ou *framework* PHP, votre projet ; et livrer à votre enseignant responsable, avant la date de rendu :

- le **code source** (sans ressources binaires) correctement **commenté** ;
- un fichier nommé `lisez-moi.txt` succinct comportant **au minimum** :
 - le **nom** de l'équipe et de tous ces membres ;
 - l'URL de l'**index** du site Web ;
 - une **présentation** concise du projet réalisé ;
 - les **choix techniques** (technologies, algorithmes, etc.) et leurs raisons ;
 - la **configuration logicielle** minimale et conseillée ;
 - l'ensemble des **identifiants de connexion** :
 - du site Web (au minimum un compte de test et un d'administration) ;

- de la base de données ;
- de l'hébergement ;
- du compte du système de version (**obligatoire**).

Vous utiliserez pour cela, au **minimum**, PHP, MySQL, HTML et CSS.

Vous devez apporter un soin tout particulier à la **programmation orientée objet** et à l'utilisation du modèle **MVC** et éventuellement de divers **modèles de conception**.

Votre application doit être **évolutive**, **modulaire** et professionnelle (**robuste**, **fiable** et intégralement **fonctionnelle**).

En outre, vous devrez impérativement utiliser des comptes reliés à un système de version (Git, SVN etc.). Chaque membre du groupe disposera d'un compte permettant de tracer son activité. La gestion du temps et le partage des tâches donneront lieu à une **évaluation**.

Enfin, vous devez respecter les impératifs suivants :

- bonne indentation et commentarisation de l'ensemble des codes sources ;
- bonne architecture des répertoires sources ;
- validation W3C de toutes vos pages HTML (documents au minimum de type HTML5) ;
- validation W3C de toutes vos pages CSS (documents au minimum de profil CSS niveau 3 avec aucun avertissement et en tenant compte des extensions propriétaires comme avertissement).

6 Conseils

Voici quelques conseils :

- faites une recherche des solutions concurrentes ;
- n'oubliez pas de tester le site Web sur différents systèmes d'exploitation et différents navigateurs Web (en utilisant des versions portables ou des services Web comme <http://www.browsershots.org/>) ;
- pour les redoublants et ceux qui sont à l'aise, vous pouvez mettre en place quelques fonctionnalités intéressantes supplémentaires. À faire **uniquement** après avoir terminé tous les points essentiels notés ci-dessus et après avoir optimisé et approfondi les fonctionnalités demandées ;
- pensez à utiliser des services gratuits de stockage de ressources en ligne (Youtube, ImageShack, Scribd, etc.) afin d'éviter d'encombrer inutilement la taille restreinte du projet (et, bien souvent, de gagner en performance) ;
- faites un site Web adaptatif (*responsive*) pouvant être notamment utilisable sur PC, *smartphone* et tablettes ;
- réalisez quelques éléments d'une bibliothèque (logicielle) ou même un *framework* minimaliste ;
- utilisez des bibliothèques PHP (pour la pagination, etc.) ou d'autres langages (jQuery, etc.), des préprocesseurs CSS (Less, Sass etc.) et autres outils Web (Twitter Bootstrap, Foundation etc.) mais **pas** de système de gestion de contenu préexistant (CMS open source) ou de *frameworks* PHP ;

- utilisez des gestionnaires de dépendance (Composer, etc.), des systèmes de gestion de version (Git : Bitbucket et <https://try.github.com> ; etc.), des *frameworks* de test unitaire (PHPUnit, etc.) et d'intégration continue (Jenkins, TravisCI, etc.) ;
- activez le SSH par défaut pour votre solution d'hébergement ;
- utilisez les outils de développement de vos navigateurs.