

## Graduate Développeur web full stack

**Prénom :** Julien

**Nom :** DENIAUD

**Adresse :** 2 Rue des épinettes 44830 Bouaye

Nom du projet : ECF\_GraduateDeveloppeur\_juin2023\_Deniaud\_Julien  
Restaurant le Quai Antique

Lien Github du projet : <https://github.com/julien44830/restaurant.git>

URL du site (si vous avez mis votre projet en ligne) : <https://restaurantlequaiantique.000webhostapp.com/>

## Description du projet

1. Liste des compétences du référentiel qui sont couvertes par le projet

Activité – Type 1 : Développer la partie front-end d’une application web ou web mobile en intégrant les recommandations de sécurité

- 1 Maquetter une application
- 2 Réaliser une interface utilisateur web statique et adaptable
- 3 Développer une interface utilisateur web dynamique
- 4 Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce

Activité – Type 2 : Développer la partie back-end d’une application web ou web mobile en intégrant les recommandations de sécurité

- 5 Créer une base de données
- 6 Développer les composants d’accès aux données
- 7 Développer la partie back-end d’une application web ou web mobile
- 8 Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce

2. Résumé du projet en français d’une longueur d’environ 20 lignes soit 200 à 250 mots, ou environ 1200 caractères espaces non compris

## LE PROJET Restaurant

Le Chef Arnaud Michant aime passionnément les produits - et producteurs - de la Savoie.

C’est pourquoi il a décidé d’ouvrir son troisième restaurant dans ce département.

Le Quai Antique sera installé à Chambéry et proposera au déjeuner comme au dîner une expérience gastronomique, à travers une cuisine sans artifice.

Plus encore que ses deux autres restaurants, Arnaud Michant le voit comme une promesse d’un voyage dans son univers culinaire.

Lors de l’inauguration de son deuxième établissement, le chef Michant a pu constater l’impact positif que pouvait avoir un bon site web sur son chiffre d’affaires. C’est pourquoi il a fait appel à l’agence web dont vous faites partie.

Dans le cadre de cette mission qui vous est affectée, vous aurez à créer une application web vitrine pour le Quai Antique avec ce goût de la qualité que recherche Arnaud Michant.

3. Cahier des charges, expression des besoins, ou spécifications fonctionnelles du projet

US1. Se connecter.

*Utilisateurs concernés: Administrateur, Clients*

Le compte administrateur sera créé pour un employé du restaurant en particulier: l'hôte d'accueil. C'est lui qui gèrera les informations sur le site web.

Toutefois, un autre type de compte sera possible: le "client" (voir US7 - Mentionner des allergies).

Quel que soit le type d'utilisateur souhaitant se connecter, il pourra le faire grâce au même formulaire de connexion. Les identifiants à entrer seront l'adresse e-mail et un mot de passe sécurisé.

## US2. Créer une galerie d'images.

*Utilisateurs concernés: Administrateur*

Sur la page d'accueil, le chef Michant aimerait afficher les photographies de ses plats les plus appréciés afin de donner l'eau à la bouche de ses potentiels convives.

Toute photo devra pouvoir être ajoutée, modifiée ou supprimée sur la plateforme d'administration.

Chaque photo aura aussi un titre. Il sera visible sur la page d'accueil lors du survol de son image.

Un bouton d'appel à l'action vers le module de réservation (Voir US6 - Réserver une table) devra être positionné juste après la galerie.

## US3. Publier la carte.

*Utilisateurs concernés: Administrateur*

La carte du restaurant devra être présente sur une page dédiée.

Les plats seront listés dans des catégories (ex: Entrées, Desserts, Burgers, etc).

Les informations nécessaires pour chaque plat sont:

- un titre
- une description
- un prix

## US4. Présenter les menus.

*Utilisateurs concernés: Administrateur*

En plus des plats incontournables à proposer, le chef Michant voudrait des menus.

Pour chaque menu, on aura:

- un titre
- une à plusieurs formules, ayant chacune un prix et une description.

Un exemple est disponible dans les annexes

## US5. Définir les horaires d'ouverture.

*Utilisateurs concernés: Administrateur*

Pour un restaurant, cette information est CAPITALE.

Pour chaque jour de la semaine, les horaires d'ouverture devront donc être affichés dans le pied de chaque page du site.

Le chef Michant souhaite aussi que l'administrateur puisse modifier les horaires à sa guise.  
Un exemple est disponible dans les annexes.

## US6. Réserver une table.

*Utilisateurs concernés: Visiteurs, Clients*

Dans le menu, un bouton d'appel à l'action sera particulièrement mis en valeur: "réserver"

Au clic de ce dernier, le visiteur est redirigé sur un formulaire à remplir.

Plusieurs champs seront nécessaires:

- le nombre de couverts
- la date
- l'heure prévue
- la mention des allergies

Le visiteur doit savoir si des places sont disponibles sans rechargement de la page.

On pourra sélectionner un horaire par tranche de 15 minutes entre l'ouverture et la fermeture du restaurant.

La dernière heure avant la fermeture, aucun convive n'est accepté.

Attention! Le restaurant est limité en nombre de convives.

Pour éviter toute déconvenue sur place, il faudra refuser les réservations au-delà d'un certain seuil.

Ce seuil de convives maximum pourra être précisé dans le panel d'administration.

## US7. Mentionner des allergies.

*Utilisateurs concernés: Visiteurs, Clients*

Lors de la réservation d'une table, le visiteur peut indiquer si une personne qui l'accompagne a des allergies.

Si le visiteur vient régulièrement dans ce restaurant, il peut aussi créer un compte client et donc gagner du temps lors de la complétion du formulaire.

Quand le visiteur créera son compte, on lui proposera d'entrer une adresse email, un mot de passe sécurisé, un nombre de convives par défaut ainsi que la mention des allergies.

Dorénavant, si le visiteur se connecte avant de remplir le formulaire de réservation d'une table, le nombre de convives et les allergies seront remplis automatiquement avec les réglages du client.

4. Spécifications techniques du projet, élaborées par le candidat, y compris pour la sécurité et le web mobile

Serveur :

- <https://www.000webhost.com/>
- Version PHP 8.2.0
- Extension PHP : PDO
- MariaDB (version 10.4.27)

Pour le front :

- HTML 5
- CSS 3
- Bootstrap
- JavaScript

pour le back :

- PHP 8.2.0 sous PDO
- MySQL

5. Description de la veille, effectuée par le candidat durant le projet, sur les vulnérabilités de sécurité

Le site a été utilisé durant sa production par diverse personnes pour permettre de soulever diverse bug et faille, les recherche en vers ses problème de fonctionnalités ont été réalisé pour les corriger.

6. Description d'une situation de travail ayant nécessité une recherche, effectuée par le candidat durant le projet, à partir de site anglophone

Pour la création et l'identification de l'utilisateur sur le site, un mot de passe sécurisé est nécessaire. Il se compose pour moi au minimum d'un caractère spécial, majuscule, minuscule et de chiffre. Un simple code pour vérifier cela et empêcher l'utilisateur de créer un mot de passe trop facile à hacker est suffisant. Mais pour empêcher certains utilisateurs malveillants, il est indispensable de crypter le mot de passe en cas de fuite de la base de donnée.

7. Extrait du site anglophone, utilisé dans le cadre de la recherche décrite précédemment, accompagné de la traduction en français effectuée par le candidat sans traducteur automatique (environ 750 signes).

Source : [https://www.positioniseverything.net/php-password\\_hash/](https://www.positioniseverything.net/php-password_hash/)

## What Is PHP password\_hash?

The PHP `password_hash` function is a function that implements a **one-way algorithm to hash passwords**. It accepts a password, a hashing algorithm, and an optional associative array of options—the syntax: `password_hash(password, algorithm, options)`. In the end, it returns the hashed password.

Please note that the PHP `password_hash` function generates an **irreversible hashed password**. Interestingly, the resulting hashed password contains the used algorithm as well as the cost and the salt. So, you don't need to remember and provide the given details separately while verifying the hashed password.

Qu'est-ce que PHP `password_hash`?

La fonction PHP `password_hash` est une fonction qui implémente un algorithme unidirectionnel pour hasher les mots de passe. Il accepte un mot de passe, un algorithme de hachage et un tableau associatif facultatif d'options — la syntaxe: `password_hash ( mot de passe, algorithme, options )`. Au final, il renvoie le mot de passe hachuré.

Veuillez noter que la fonction PHP `password_hash` génère un mot de passe hachuré irréversible. Fait intéressant, le mot de passe hachuré résultant contient l'algorithme utilisé ainsi que le coût et le sel. Vous n'avez donc pas besoin de vous souvenir et de fournir les détails donnés séparément lors de la vérification du mot de passe hachuré.

( \*je ne peux pas me passer de traducteur automatique )

### 8. Autres ressources

Source : <https://www.php.net/manual/fr/function.password-hash.php>

## Description [¶](#)

```
password_hash(string $password, string|int|null $algo, array $options = []): string
```

La fonction `password_hash()` crée un nouveau hachage en utilisant un algorithme de hachage fort et irréversible.

Les algorithmes suivants sont actuellement supportés :

- `PASSWORD_DEFAULT` - Utilisation de l'algorithme `bcrypt` (par défaut depuis PHP 5.5.0). Notez que cette constante est conçue pour changer dans le temps, au fur et à mesure que des algorithmes plus récents et plus forts sont ajoutés à PHP. Pour cette raison, la longueur du résultat issu de cet algorithme peut changer dans le temps, il est donc recommandé de stocker le résultat dans une colonne de la base de données qui peut contenir au moins 60 caractères (255 caractères peut être un très bon choix).
- `PASSWORD_BCRYPT` - Utilisation de l'algorithme `CRYPT_BLOWFISH` pour créer la clé de hachage. Ceci va créer une clé de hachage standard [crypt\(\)](#) utilisant l'identifiant `"$2y$"`. Le résultat sera toujours une chaîne de 60 caractères, ou `false` si une erreur survient.
- `PASSWORD_ARGON2I` - Utilise l'algorithme de hachage Argon2i pour créer le hachage. Cet algorithme est seulement disponible si PHP a été compilé avec le support d'Argon2
- `PASSWORD_ARGON2ID` - Utilise l'algorithme de hachage Argon2id pour créer le hachage. Cet algorithme est seulement disponible si PHP a été compilé avec le support d'Argon2
- Exemple #1 Exemple avec `password_hash()`

```
<?php
/**
 * Nous voulons juste hacher notre mot de passe en utilisant
 l'algorithme par défaut.
 * Actuellement, il s'agit de BCrypt, ce qui produira un résultat
 sous forme de chaîne de
 * caractères d'une longueur de 60 caractères.
 *
 * Gardez à l'esprit que DEFAULT peut changer dans le temps, aussi,
 vous devriez vous
 * y préparer en autorisant un stockage supérieur à 60 caractères
 (255 peut être un bon choix)
 */

echo password_hash("rasmuslerdorf", PASSWORD_DEFAULT);
?>
```

## 9. Informations complémentaires