

NOM	PEREZ
Prénom	Florent
Date de naissance	17/09/1978

Copie à rendre

Graduate Développeur

(Android, Angular, Flutter, Front End, Full Stack, IOS, PHP/Symfony)

Documents à compléter et à rendre

Objectif & Livrables

Livrables :

Le lien du (ou des) dépôt(s) Github public, où sera présent le code de vos applications
Voici le lien du Github :

https://github.com/FloAFDEV/ECF_QUAI_ANTIQUE

7. e-mail et mot de passe administrateur 'adminquaiantique@quaiaint.fr' mdp: 'Quaiaintique'
8. Adresse de phpMyadmin sur votre navigateur http://127.0.0.1:8080/index.php?route=/database/structure&db=db_ecf_quai_antique
9. Les relations des entités et les diagramme de séquence sont disponibles dans le dossier /config/Docs à la racine du projet.
10. Le lien du Trello: <https://trello.com/b/SF5ceWq5/quai-antique>

Votre application Symfony est maintenant déployée et accessible à l'adresse <http://localhost:8000>.

Le lien de la version en ligne de l'application web

Le lien vers votre logiciel de gestion de projet pour cet ECF (Trello, Jira, etc...)

Voici le lien de mon trello :

<https://trello.com/b/SF5ceWq5/quai-antique>

Devra aussi se trouver:

- Un fichier readme.md contenant la démarche à suivre pour l'exécution en local
 - Ne pas oublier dans ce readme l'explication de la création d'un administrateur pour le back-office de l'application web
- Voici le readme :
-

Déploiement de votre application Symfony Quai Antique en local

Pour déployer une application Symfony en local, vous avez besoin de suivre les étapes suivantes :

Assurez-vous d'avoir les prérequis suivants installés sur votre ordinateur :

1. PHP 7.1 ou version ultérieure

MySQL ou un autre serveur de base de données compatible avec Symfony, je recommande PhpMyAdmin

Un serveur Web (par exemple Apache ou Nginx), je recommande Xampp.

Composer ou docker

2. Téléchargez et installez Symfony en utilisant l'installateur Composer :

composer create-project symfony/skeleton:"5.4." my_project_directory*

cd my_project_directory

composer require webapp

3. Configurez votre base de données en modifiant les paramètres dans le fichier .env :

*Adresse de phpMyadmin sur votre navigateur DATABASE_URL=mysql ou
http://127.0.0.1:8080/index.php?route=/database/structure&db=db_ecf_quai_antique*

4. Créez la base de données en exécutant la commande suivante :

php bin/console doctrine:database:create

Créez les tables en utilisant les entités de votre application, attention vous aurez besoin de MakerBundle de Symfony pour exécuter ces commandes :

php bin/console make:migration

php bin/console doctrine:migrations:migrate

5. Démarrez le serveur Web interne de Symfony en exécutant la commande suivante :

php bin/console server:run

6. e-mail et mot de passe administrateur 'adminquaiaantique@quaiaant.fr' mdp: 'Quaiaantique'

// Sous mac:

** docker volume create db_ecf_quai_antique*

** 1) Lancer le container Docker : docker-compose up -d (modifié) 2) Lancer le serveur PHP : symfony serve -d ou symfony server:start -d (modifié) ou symfony server:start -d 3) Arrêter le serveur PHP : symfony server:stop*

** 4) Stopper le container Docker : docker-compose down -v*

** Passer en https en local , dans le terminal : symfony server:ca:install*

Votre application Symfony est maintenant déployée et accessible à l'adresse http://localhost:8000.

☐ About Me:

Jeune développeur tout juste reconverti, passionné de code et autres techno.

☐ Socials:

[![Discord](https://img.shields.io/badge/Discord-%237289DA.svg?logo=discord&logoColor=white)](https://discord.gg/https://discord.gg/hz7

```
C5qFA)                                [![Facebook](https://img.shields.io/badge/Facebook-
%231877F2.svg?logo=Facebook&logoColor=white)](https://facebook.com/Flo          Skat)
[![LinkedIn](https://img.shields.io/badge/LinkedIn-
%230077B5.svg?logo=linkedin&logoColor=white)](https://linkedin.com/in/Florent Perez)
```

☐ Tech Stack:

```
![Symfony](https://img.shields.io/badge/symfony-%23000000.svg?style=for-the-
badge&logo=symfony&logoColor=white)      ![HTML5](https://img.shields.io/badge/html5-
%23E34F26.svg?style=for-the-badge&logo=html5&logoColor=white)
![JavaScript](https://img.shields.io/badge/javascript-%23323330.svg?style=for-the-
badge&logo=javascript&logoColor=%23F7DF1E)    ![PHP](https://img.shields.io/badge/php-
%23777BB4.svg?style=for-the-badge&logo=php&logoColor=white)
![Trello](https://img.shields.io/badge/Trello-%23026AA7.svg?style=for-the-
badge&logo=Trello&logoColor=white)          ![Docker](https://img.shields.io/badge/docker-
%230db7ed.svg?style=for-the-badge&logo=docker&logoColor=white)
![IOS](https://img.shields.io/badge/IOS-%2320232a.svg?style=for-the-
badge&logo=apple&logoColor=white)      ![ANDROID](https://img.shields.io/badge/android-
%2320232a.svg?style=for-the-badge&logo=android&logoColor=%a4c639)
![Apache](https://img.shields.io/badge/apache-%23D42029.svg?style=for-the-
badge&logo=apache&logoColor=white)                                ![Adobe
Lightroom](https://img.shields.io/badge/Adobe%20Lightroom-31A8FF.svg?style=for-the-
badge&logo=Adobe%20Lightroom&logoColor=white)
![ESLint](https://img.shields.io/badge/ESLint-4B3263?style=for-the-
badge&logo=eslint&logoColor=white)
```

☐ GitHub Stats:

```
<br/>
<br/>

```

☐ GitHub Trophies

```

```

- Les fichiers de création et d'alimentation de la base de données de votre application web (migrations, fixtures ou scripts SQL à la main)

- **Attention:** Générer du SQL par le biais d'une interface graphique (ex: PHPMyAdmin), et donc sans avoir à coder, conduira à la non-validation d'une des compétences du titre professionnel!

59 commits disponibles ici :

https://github.com/FloAFDEV/ECF_QUAI_ANTIQUUE/commits/main

```
// just set up a fresh $task object (remove the example data)
$reservation = new Reservations();

$form = $this->createForm(ReservationType::class, $reservation);

$restaurant = $restaurantRepository->find(1);

$form->handleRequest($request);
// dd($reservation); Remove this commented out code.
if ($form->isSubmitted() && $form->isValid()) {
    // $form->getData() holds the submitted values
    // but, the original `$task` variable has also been updated
    $reservation = $form->getData();

    if ($reservation->getPersons() > $restaurant->getMaximumCapacity()) {
        $this->addFlash('danger', 'Il n\'y a plus de place pour cette date');
        return $this->redirectToRoute('app_reservation');
    }

    $entityManager->persist($reservation);
    $entityManager->flush();
    return $this->redirectToRoute('reservation_success');
}

return $this->render('reservation/index.html.twig', [
    'form' => $form->createView()
]);
```

You, last month • création register et auth ...

```
Route('/reservation/success', name: 'reservation_success')]
lie function success(ReservationsRepository $reservationRepository): Response

return $this->render('reservation/success.html.twig');
```

```

class RegistrationController extends AbstractController
{
    private EmailVerifier $emailVerifier;

    public function __construct(EmailVerifier $emailVerifier)
    {
        $this->emailVerifier = $emailVerifier;
    }

    #[Route('/inscription', name: 'app_register')]
    public function register(Request $request, UserPasswordHasherInterface $userPasswordHasher,
    UserAuthenticatorInterface $userAuthenticator, AppAuthenticator $authenticator, EntityManagerInterface
    $entityManager): Response
    {
        $user = new User();
        $form = $this->createForm(RegistrationFormType::class, $user);
        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {
            // encode the plain password
            if ($user->getEmail() === "adminquaiantique@quaiant.fr") {
                $user->setRoles(["ROLE_ADMIN"]);
            } else {
                $user->setRoles(["ROLE_USER"]);
            }
            $user->setPassword(
                $userPasswordHasher->hashPassword(
                    $user,
                    $form->get('plainPassword')->getData()
                )
            );

            $entityManager->persist($user);
            $entityManager->flush();

            // generate a signed url and email it to the user
            $this->emailVerifier->sendEmailConfirmation(
                'app_verify_email',
                $user,
                (new TemplatedEmail())
                    ->from(new Address('mailer@quai-antique.com', 'Quai-Antique Bot'))
                    ->to($user->getEmail())
                    ->subject('Please Confirm your Email')
            );
        }
    }
}

```



```

class AppAuthenticator extends AbstractLoginFormAuthenticator
{
    public function authenticate(Request $request): Passport
    {
        $email = $request->request->get('email', '');

        $request->getSession()->set(Security::LAST_USERNAME, $email);

        return new Passport([
            new UserBadge($email),
            new PasswordCredentials($request->request->get('password', '')),
            [
                new CsrfTokenBadge('authenticate', $request->request->get('_csrf_token')),
            ]
        ]);
    }

    public function onAuthenticationSuccess(Request $request, TokenInterface $token, string $firewallName)
    : ?Response
    {
        if ($targetPath = $this->getTargetPath($request->getSession(), $firewallName)) {
            return new RedirectResponse($targetPath);
        }

        $email = $request->request->get('email', '');
        if ($email === "adminquaiaantique@quaiaant.fr") {
            return new RedirectResponse($this->urlGenerator->generate('admin'));
        }

        // For example:
        return new RedirectResponse($this->urlGenerator->generate('app_home'));
        ... // throw new \Exception('TODO: provide a valid redirect inside ' . __FILE__); Remove this commented out code
    }

    protected function getLoginUrl(Request $request): string
    {
        return $this->urlGenerator->generate(self::LOGIN_ROUTE);
    }
}

```

```

public function remove(Reservations $entity, bool $flush = false): void
{
    $this->getEntityManager()->remove($entity);

    if ($flush) {
        $this->getEntityManager()->flush();
    }
}

public function findAllByDateAndHour(\DateTimeInterface $date, \DateTimeInterface $hour): array
{
    return $this->createQueryBuilder('r')
        ->andWhere('r.date = :date')
        ->andWhere('r.hour = :hour')
        ->setParameter('date', $date)
        ->setParameter('hour', $hour)
        ->getQuery()
        ->getResult();
}

public function getTotalPersonsByDateAndBetweenHours(\DateTimeInterface $date, \DateTimeInterface $startHour, \DateTimeInterface $endHour): int
{
    return $this->createQueryBuilder('r')
        ->select('SUM(r.persons)')
        ->andWhere('r.date = :date')
        ->andWhere('r.hour BETWEEN :startHour AND :endHour')
        ->setParameter('date', $date)
        ->setParameter('startHour', $startHour)
        ->setParameter('endHour', $endHour)
        ->getQuery()
        ->getSingleScalarResult();
}

```

-
- Une documentation technique au format .pdf avec vos réflexions initiales sur le projet (technologies choisies, configuration de l'environnement de travail, diagramme de classe ou Méthode MERISE, diagramme de cas d'utilisation, diagramme de séquence...)
 - **Attention:** Chaque diagramme précisé dans l'annexe doit être obligatoirement présent! Ils sont en lien direct avec une des compétences du référentiel.
- Une charte graphique au format .pdf regroupant:
 - La palette de couleurs
 - La palette des polices d'écriture choisies
 - L'export des maquettes attendues
 - wireframe et/ou mockup pour la version mobile ET desktop

--Merci de vous reporter au fichier fourni en annexe au format PDF.--

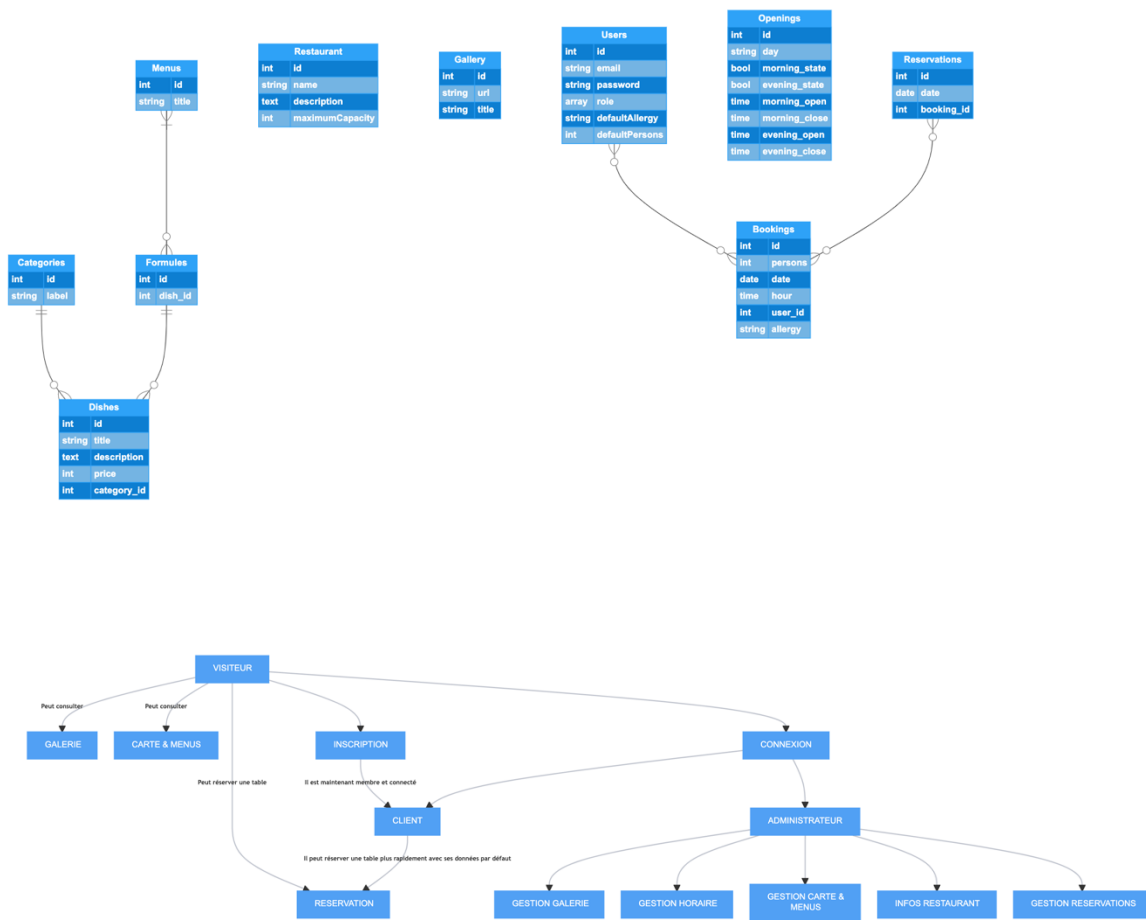
Attention! Pour respecter les contraintes de temps et valider toutes les compétences du référentiel, il est vivement conseillé de ne pas passer trop de temps sur le design de vos applications. S'il est minimaliste, mais cohérent, alors c'est suffisant.

Quai Antique ECF / Florent PEREZ

- Dans ma réflexion j'ai rapidement su que je voulais faire le site sous symfony avec docker comme serveur local, étant sous mac le choix le plus « simple » pour un déploiement prochain étant celui-ci.

L'approche des diagrammes est disponible ci-dessous.

Les diagrammes / UML :



N'étant pas au bout de ma formation, tout n'est pas implémenté à ce jour, je vous transmets, malgré tout, le fruit de mes efforts (nombreux).

La palette de couleur :

#444444 Color Hex



#444444
(68,68,68)

#ffc56e Color Hex



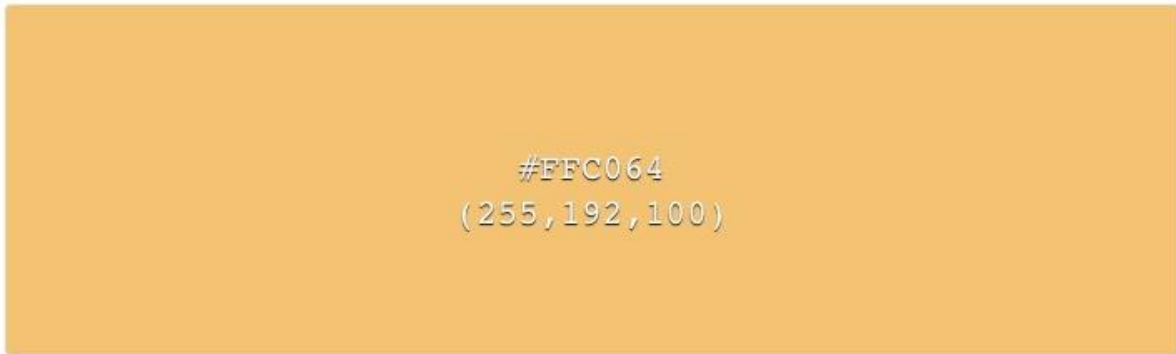
#FFC56E
(255,197,110)

#ffb03b Color Hex



#FFB03B
(255,176,59)

#ffc064 Color Hex



★ 0 Favorites ■ 0 Comments

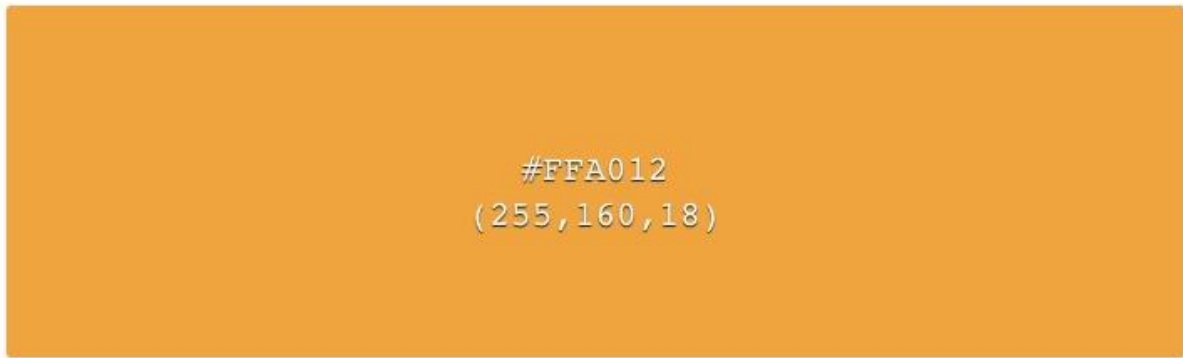
#1a1816 Color Hex



#ffffff Color Hex White

#FFFFFF
(255,255,255)

#ffa012 Color Hex



#FFA012
(255,160,18)

Concernant les polices, simples et sobre :

Satisfy
Sideshow

1 style

*Whereas recognition of the
inherent dignity*

Whereas recognition
of the inherent
dignity

Ainsi que « sans-serif ».

Visuals

Concernant les visuels, voici la version bureau ainsi que la version mobile.

La version desktop :

Page d'accueil du Quai Antique

Pour les besoins de l'école Studi / ECF



La version mobile :

Rendu visuel sur mobile

Voici la même page
d'accueil en format mobile -
Il ne s'agit pas de
Wireframes présentement
mais du rendu visuel final
uniquement pour la page
d'accueil du site



Tableau de bord

Categories

Plats

Formules

Galerie

Menus

Heures d'ouvertures

Reservations

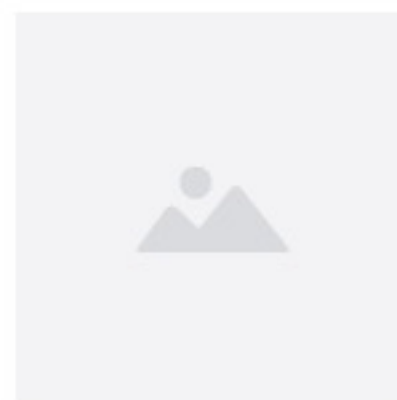
Restaurant

Clients

Administration de votre restaurant

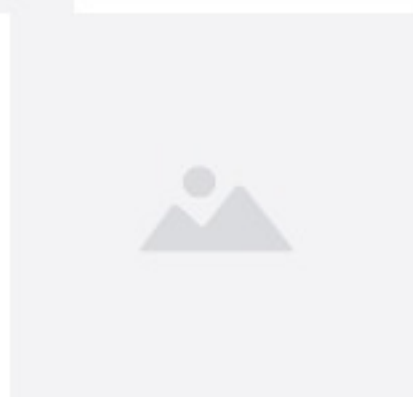
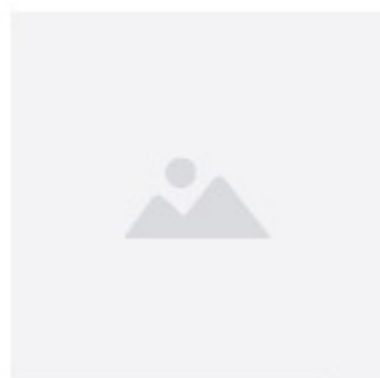
Bienvenue dans votre interface d'administration pour le site du restaurant Quai Antique.

Qu'est-ce qu'on vous sert aujourd'hui ?



Quai Antique

Le Quai Antique est installe a Chambéry et propose au déjeuner comme au diner une experience gastronomique, a 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.

[NOS MENUS](#)[DEVENIR MEMBRE ANTIQUE](#)



M'inscrire

Pourquoi notre Restaurant

Venez decouvrir les saveurs authentiques de notre cuisine locale dans un cadre chaleureux et accueillant. Nos plats sont prepares avec les ingredients frais et de qualite pour vous offrir une experience culinaire inoubliable. Rejoignez-nous pour un repas en famille ou entre amis et laissez-vous tenter par nos delieux plats. Nous vous attendons avec impatience pour vous faire vivre une experience culinaire unique



Notre Tartiflette

Plat traditionnel de la Savoie compose de pommes de terre, oignons, lardons, reblochon et creme fraiche.



Notre Raclette

Plat egalement originaire de la Savoie, fromage fondu servi avec des pommes de terre, des charcuteries et des legumes

03

Notre Fondue Savoyade

Une fondue de fromage a base de fromages de la region tels que le comte, la beaufort et le gruyere, accompagnee de pain et de vin blanc.

Nos meilleurs plats gourmands

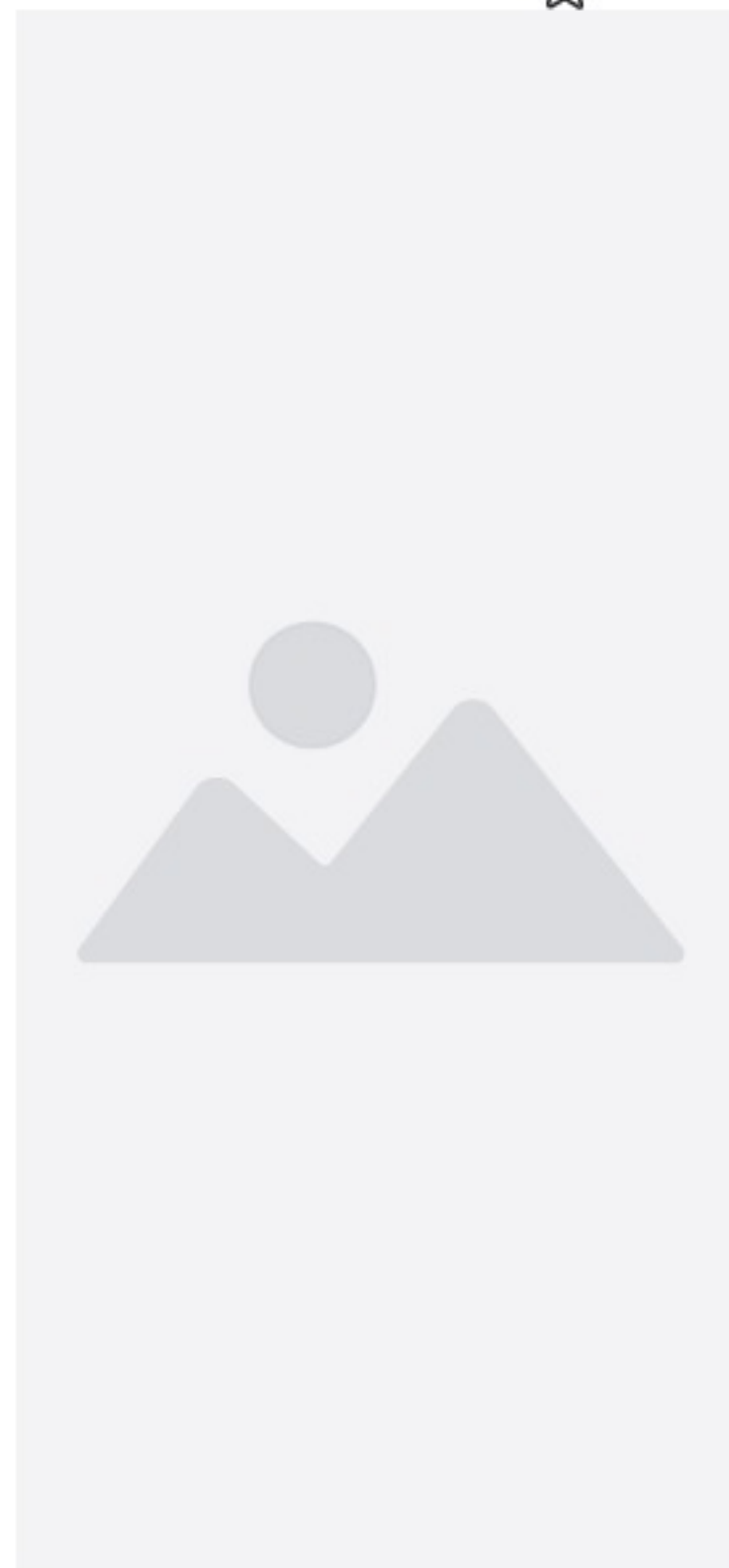


Arnaud Michant Le Chef

Le Chef Arnaud Michant aime passionnément les produits - et producteurs - de la Savoie. C'est pourquoi il a decide d'ouvrir son troisieme restaurant dans ce dejeuner comme au diner une experience gastronomique, a 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.

Arnaud Michant Le Chef

Le Chef Arnaud Michant aime passionnément les produits - et producteurs - de la Savoie. C'est pourquoi il a decide d'ouvrir son troisieme restaurant dans ce dejeuner comme au diner une experience gastronomique, a 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.



Arnaud Michant

Master Chef



Nous Contacter

Pour toutes reservations ou demandes d'informations
supplementaires, veuillez nous contacter directement au
numero de telephone suivant : +33 xx xx xx. Nous nous
ferons un plaisir de vous repondre dans les plus brefs
delais.



Adresse

Quai Antique
2 rue du Mont Blanc
73000 Chambéry,
France

Nos heures
d'ouvertures:

Lundi -- Samedi:
11:00 - 23:00

Email:

info@example.com
contact@example.cor

Telephone:

+3360000606
+33479001234

