

FICHE DE PROJET 1 : CRÉATION D'UNE APPLICATION DE GESTION DE RESTAURANT

1. Description du projet

Concevoir un site landing page de présentation d'un restaurant en utilisant les technologies HTML, CSS, BOOTSTRAP & JAVASCRIPT.

La page doit avoir le menu, un header, une section réservation, galerie, témoignages clients et des informations de contact, un footer.

Chaque plat aura une image et un descriptif, une section pour commander, une section pour faire une réservation de table.

2. Activités

Tâches à réaliser / Étapes / Livrables	Consignes
Concevoir les maquettes	Utiliser Figma pour créer les maquettes du site.
Nav (menu)	<ul style="list-style-type: none">- Accueil- Notre menu (le menu du restaurant) en pop up- Offres spéciales- Réservation (de table)- Galerie- Témoignages client- Contact
Header :	<ul style="list-style-type: none">- Le nom du restaurant,- un slogan accrocheur et une image attrayante de la nourriture ou de l'intérieur du restaurant (en background).- Call-to-action: Inclure des boutons d'appel à l'action clairs pour inciter les visiteurs à réserver une table, de passer une commande ou de contacter le restaurant <p>Proportion (98vh en hauteur sur desktop et 100vh en hauteur sur mobile)</p>
Le menu du restaurant	Mettre en évidence les plats phares du restaurant avec de belles images et des descriptions alléchantes. Inclure également des options pour les menus spéciaux, les boissons et les desserts.
Réservation	Offrir la possibilité de réserver une table en ligne avec un formulaire simple à remplir. <ul style="list-style-type: none">- Nom et prénom :- Date et heure :- Nombre de personnes :- Informations de contact :- Commentaires ou demandes spéciales :- Bouton de soumission :

Tâches à réaliser / Étapes / Livrables	Consignes
Galerie	Présenter des images attrayantes de plats, de l'intérieur du restaurant et de l'équipe pour donner aux visiteurs un aperçu de l'expérience qu'ils peuvent attendre (6 images maximum) <ul style="list-style-type: none"> - 3 images par lignes sur Desktop - 2 images par ligne sous mobile
Témoignages client	Afficher des avis de clients satisfaits pour renforcer la crédibilité du restaurant. (En Slide 3 témoignages : Prénom & Nom, Image du témoin et le texte de témoignage)
Contact	Fournir un formulaire de contact ou des coordonnées pour que les visiteurs puissent poser des questions ou obtenir plus d'informations <ol style="list-style-type: none"> 1. Adresse du restaurant (Mettre votre adresse) 2. Le téléphone du restaurant (Mettre votre numéro et le rendre cliquable : au clic on doit pouvoir automatiquement vous appeler) 3. Email du restaurant (Mettre votre email et le rendre cliquable : au clic on doit pouvoir envoyer un email) Éléments du formulaire <ul style="list-style-type: none"> ● Prénom & Nom ● Téléphone ou Email ● Sujet ● Objet
Footer:	Inclure des liens utiles, les liens vers les réseaux sociaux du restaurant, des mentions légales et des informations sur la politique de confidentialité

3. Critères d'évaluation

Votre projet sera évalué en fonction des critères suivants :

- Conception d'une belle interface utilisateur avec le respect du responsive design
- Apparition de tous les éléments dans le menu
- Fonctionnalités complètes de l'interface utilisateur et convivialité de l'application.
- Inclure un bouton clair et visible pour que le client puisse envoyer sa demande
- Utilisation de Trello pour la gestion des tâches
- Utilisation de git pour le versionning
- Qualité et clarté du rapport présentant votre projet.

FICHE DE PROJET 2 : Réalisation d'une application météo

4. Description du projet

Pour la conception de cette application Angular, les apprenants doivent utiliser une API météo publique, telle que [OpenWeatherMap](#), pour afficher les prévisions météorologiques en temps réel pour une ville donnée au Sénégal.

L'application doit inclure l'affichage de la température actuelle, les conditions météorologiques, les prévisions à court et long terme, ainsi que d'autres informations pertinentes. Cela va permettre aux apprenants de mettre en pratique les concepts de consommation d'API, de manipulation de données et d'affichage des résultats dans une interface utilisateur

5. Activités

Tâches à réaliser / Étapes / Livrables	Consignes
Mise en place du projet Angular	Créer un nouveau projet Angular en utilisant Angular CLI.
Création de composants	Créer des composants pour afficher la météo actuelle, les prévisions à court terme, les prévisions à long terme, etc.
Consommation de l'API	Utiliser HttpClient pour effectuer des requêtes vers l'API météo publique, en récupérant les données météorologiques nécessaires.
Affichage des données	Utiliser le binding de données pour afficher les informations météorologiques récupérées de l'API dans l'interface utilisateur.
Gestion des erreurs	Gérer les erreurs éventuelles lors de la récupération des données depuis l'API.
Ajout de fonctionnalités supplémentaires	Ajouter des fonctionnalités telles que la possibilité de rechercher la météo pour différentes villes, l'affichage des icônes météorologiques, etc.
Styling	Appliquer des styles CSS pour rendre l'application météo attrayante et conviviale. Ces tâches permettent aux apprenants de mettre en pratique les concepts de consommation d'API, de manipulation de données et d'affichage des résultats dans une application Angular. Les critères d'évaluation pour une application Angular utilisant une API publique pour afficher les données météorologiques pourraient inclure :

6. Critères d'évaluation

Votre projet sera évalué en fonction des critères suivants :

Fonctionnalité : L'application doit fonctionner correctement et afficher les informations météorologiques pour une ville donnée.

Interface utilisateur : L'interface utilisateur doit être conviviale, facile à utiliser et attrayante.

Consommation de l'API : Les apprenants doivent démontrer leur capacité à consommer l'API météo publique et à récupérer les données nécessaires pour afficher les informations météorologiques.

Manipulation de données : Les apprenants doivent être capables de manipuler les données récupérées de l'API pour les afficher de manière appropriée dans l'interface utilisateur.

Gestion des erreurs : Les apprenants doivent être en mesure de gérer les erreurs éventuelles lors de la récupération des données depuis l'API.

Code : Le code doit être propre, bien organisé et facile à comprendre pour les autres développeurs.

Documentation : Les apprenants doivent fournir une documentation claire et concise pour leur application, y compris les instructions d'installation et d'utilisation. (**Readme**)

<https://openweathermap.org/api/one-call-3>