**Application de Prévision des Commandes en Pâtisserie**

**Plan du cahier des charges source travail Next :**

1. Le projet

2. Étude de la concurrence

3. Charte graphique

4. Périmètre du projet (Arborescence, Administration du contenu, Fonctionnalité, Cadre technique, Conception des contenus, Stratégie, Intégrations des contenus, Hébergement, Maintenance et suivi d’évolution)

5. Déroulement du projet

6. Ressources

7. Format attendu pour la réponse du prestataire

**Cahier des Charges : Application de Prévision des Commandes en Pâtisserie**

📍 **Client** : Pâtisserie Angelina (Paris, Versailles)  
💡 **Objectif** : Anticiper les commandes en fonction de la fréquentation prévisionnelle, en analysant plusieurs facteurs (historique des ventes, météo, événements, tourisme, économie).

**1. Présentation du Projet**

L’application permettra aux responsables de boutiques de **mieux anticiper leurs commandes** en fonction de prévisions intelligentes, réduisant ainsi les pertes et optimisant la production.

Elle reposera sur un **algorithme prédictif** qui analysera les tendances passées et les événements à venir pour estimer l'affluence et recommander les quantités de produits à préparer.

**2. Fonctionnalités**

**🔹 2.1 Gestion des utilisateurs (Symfony - API Restful)**

👤 **Types d’utilisateurs** :

* **Administrateurs** (gestionnaires des prévisions et de l’application).
* **Responsables de boutique** (consultation des prévisions et ajustement des commandes).
* **Pâtissiers** (visualisation des commandes prévues).

🔑 **Fonctionnalités** :

* Connexion / Inscription sécurisée (JWT, OAuth).
* Gestion des rôles et permissions.

**📊 2.2 Système de Prévisions des Commandes**

📌 **Données prises en compte** :  
✅ Historique des ventes 📉  
✅ Météo 🌤️  
✅ Événements politiques, économiques, culturels, sportifs 📅  
✅ Tourisme (fréquentation touristique, saisonnalité) 🏨

🛠 **Calcul d’impact (%)** :

* Variation de fréquentation estimée en comparant avec les années précédentes.
* Affichage des tendances (hausse/baisse).

💡 **Affichage des prévisions** :

* Recommandations de quantités par produit.
* Graphiques interactifs et tendances.
* Alertes en cas de forte variation prévue.

**🏪 2.3 Gestion des Boutiques**

* 📍 **Gestion des emplacements** (Paris 1er, Versailles, etc.).
* 🏷️ **Capacité maximale de production et stockage** par boutique.
* 🔢 **Données spécifiques à chaque boutique** (ventes passées, fréquentation habituelle).

**🍰 2.4 Gestion des Produits**

* 📜 Catégorisation (viennoiseries, entremets, chocolats, etc.).
* 📈 Suivi des ventes et tendances par produit.
* 📊 Ajustement des prévisions en fonction des stocks disponibles.

**📸 2.5 Module Média**

* Ajout de **photos et vidéos** pour améliorer la communication interne.
* Partage de **documents (PSD, Excel, PDF)** pour optimiser les prévisions.

**3. Technologies utilisées**

**🌐 Back-end : Symfony (API RESTful)**

* Framework PHP robuste et sécurisé.
* Gestion des données avec Doctrine ORM et MySQL.
* Authentification avec JWT/OAuth2.
* Intégration de **scripts Python** pour l’algorithme prédictif.

**💻 Front-end : React.js**

* Interface utilisateur moderne et dynamique.
* Tableau de bord interactif avec **Chart.js / Recharts** pour la visualisation des tendances.

**📊 Machine Learning & Analyse des Données : Python**

* **Pandas, Scikit-learn** pour l’analyse et la modélisation des prévisions.
* **API météo et bases de données touristiques** pour enrichir les prévisions.

**📡 Sources de données externes**

* OpenWeather API (météo).
* Données de l’INSEE / Office du Tourisme de Paris.
* API d’événements (Ville de Paris, TicketMaster pour concerts).

**4. Structure de la Base de Données**

**📌 Entités principales**

1. **User** (id, nom, email, rôle).
2. **Boutique** (id, nom, adresse, capacité\_max).
3. **Vente** (id, date, boutique\_id, produit\_id, quantité).
4. **Produit** (id, nom, catégorie, prix\_vente, coût\_production).
5. **Événement** (id, type, date, impact\_estimé).
6. **Météo** (id, date, température, précipitations).
7. **Tourisme** (id, date, nb\_touristes).
8. **Prévision** (id, date, boutique\_id, fréquentation\_prévue, impact\_estimé).
9. **Media** (id, fichier, type, boutique\_id, utilisateur\_id).

**5. Développement et Déploiement**

**🚀 Étapes de développement**

1. **Conception et structure de la base de données** (Symfony, Doctrine).
2. **Développement de l’API Symfony** (gestion des utilisateurs, ventes, prévisions).
3. **Développement du front-end en React** (tableaux de bord interactifs).
4. **Mise en place de l’algorithme de prévision avec Python**.
5. **Tests et validation avec les responsables**.
6. **Déploiement sur serveur cloud (ex: AWS, OVH, Digital Ocean)**.

**6. Bénéfices attendus**

✅ Meilleure anticipation des commandes ⏳  
✅ Réduction du gaspillage alimentaire ♻️  
✅ Amélioration de la rentabilité 💰  
✅ Gain de temps pour les responsables 🏆

**7. Prochaines étapes**

💡 Souhaites-tu qu’on commence par une maquette de la base de données ou le développement du back-end Symfony ? 🎯

**📌 Structure de la Base de Données**

* **Modèle relationnel avec MySQL (Doctrine ORM pour Symfony)**
* **Clés primaires (PK)** et **Clés étrangères (FK)**
* **Indexation** sur les champs de recherche les plus utilisés (dates, boutiques, produits)

**📍 Schéma des Entités**

**1️⃣ User (Utilisateurs)**

| **Champ** | **Type** | **Description** |
| --- | --- | --- |
| id (PK) | INT (AUTO) | Identifiant unique |
| nom | VARCHAR(100) | Nom de l'utilisateur |
| email | VARCHAR(150) | Email (unique) |
| mot\_de\_passe | VARCHAR(255) | Hash du mot de passe |
| rôle | ENUM | admin, responsable, pâtissier |
| créé\_le | DATETIME | Date de création |

🔹 **Relations** :

* Un utilisateur peut gérer plusieurs boutiques (responsable\_id dans Boutique).

**2️⃣ Boutique (Magasins)**

| **Champ** | **Type** | **Description** |
| --- | --- | --- |
| id (PK) | INT (AUTO) | Identifiant unique |
| nom | VARCHAR(100) | Nom de la boutique |
| adresse | TEXT | Adresse complète |
| capacité\_max | INT | Stock maximal possible |
| responsable\_id (FK) | INT | Responsable de la boutique (User) |

🔹 **Relations** :

* Une boutique appartient à des **responsables** (User).

**3️⃣ Vente (Historique des ventes)**

| **Champ** | **Type** | **Description** |
| --- | --- | --- |
| id (PK) | INT (AUTO) | Identifiant unique |
| date | DATE | Date de la vente |
| boutique\_id (FK) | INT | Boutique concernée |
| produit\_id (FK) | INT | Produit vendu |
| quantité | INT | Nombre d’unités vendues |

🔹 **Relations** :

* Une **vente** est liée à une **boutique** et un **produit**.

**4️⃣ Produit (Produits de la pâtisserie)**

| **Champ** | **Type** | **Description** |
| --- | --- | --- |
| id (PK) | INT (AUTO) | Identifiant unique |
| nom | VARCHAR(100) | Nom du produit |
| catégorie | ENUM | viennoiserie, gâteau, chocolat… |
| prix\_vente | DECIMAL(10,2) | Prix de vente en € |
| coût\_prod | DECIMAL(10,2) | Coût de production en € |

🔹 **Relations** :

* Un produit est référencé dans les ventes.

**5️⃣ Prévision (Données de prévision de fréquentation)**

| **Champ** | **Type** | **Description** |
| --- | --- | --- |
| id (PK) | INT (AUTO) | Identifiant unique |
| date | DATE | Date prévue |
| boutique\_id (FK) | INT | Boutique concernée |
| fréquentation\_prev | INT | Nombre estimé de clients |
| impact\_estimé | FLOAT | Variation en % (hausse/baisse) |

🔹 **Relations** :

* Prédictions générées à partir des **ventes passées** et **événements**.

**6️⃣ Événement (Données des événements influençant la fréquentation)**

| **Champ** | **Type** | **Description** |
| --- | --- | --- |
| id (PK) | INT (AUTO) | Identifiant unique |
| type | ENUM | politique, sportif, économique… |
| nom | VARCHAR(255) | Nom de l’événement |
| date | DATE | Date de l’événement |
| impact\_est | FLOAT | Impact prévu sur la fréquentation (%) |

🔹 **Relations** :

* Un événement peut influencer plusieurs prévisions.

**7️⃣ Météo (Facteurs météorologiques influençant la fréquentation)**

| **Champ** | **Type** | **Description** |
| --- | --- | --- |
| id (PK) | INT (AUTO) | Identifiant unique |
| date | DATE | Date d’observation |
| température | FLOAT | Température moyenne (°C) |
| pluie | BOOLEAN | 1 = pluie, 0 = pas de pluie |

🔹 **Relations** :

* La météo influence les prévisions (Prévision).

**8️⃣ Tourisme (Influence du tourisme sur les ventes)**

| **Champ** | **Type** | **Description** |
| --- | --- | --- |
| id (PK) | INT (AUTO) | Identifiant unique |
| date | DATE | Date d’enregistrement |
| nb\_touristes | INT | Nombre de touristes à Paris ce jour |

🔹 **Relations** :

* Impacte la fréquentation (Prévision).

**9️⃣ Media (Partage de fichiers, vidéos et documents pour gestion interne)**

| **Champ** | **Type** | **Description** |
| --- | --- | --- |
| id (PK) | INT (AUTO) | Identifiant unique |
| fichier | VARCHAR(255) | Nom du fichier |
| type | ENUM | image, vidéo, document |
| utilisateur\_id (FK) | INT | Propriétaire du fichier |

🔹 **Relations** :

* Permet le partage d’informations entre responsables.

**📌 Relations Clés**

✔ **Un User (responsable) gère une Boutique**  
✔ **Une Boutique a plusieurs Vente et Prévision**  
✔ **Un Événement influence les Prévision**  
✔ **La Météo et le Tourisme impactent les Prévision**

**📍 Prochaine étape**

📌 **Modélisation en Symfony avec Doctrine**  
➡ Faut-il générer les entités Symfony avec php bin/console make:entity ? 🚀

Recherche du développement d’application discussion première premier prompte :

Je cherche à améliorer et anticiper aux mieux mes commandes de pâtisserie au labo. Existe t il des application qui pourraient aider à anticiper le nombre de client qui pourrait venir dans mes boutique en fonction des événements politiques et économiques de Paris ou je me situ ou en fonction d’autres paramètres auxquelles je n’aurais pas pensé ou que l’on pourrait fixer ou créer. Il faut pouvoir assister au mieux les responsable qui passe commande . Par exemple en fonction de la fréquentation des années précédente. Peut-être existe-t-il des moyens d’avoir des données statistiques de la fréquentation touristique en fonction du type de population par exemple. Je pourrais par exemple proposer une application à mon employeur qui permettrait une amélioration de la prise de commande d’anticiper mieux les commandes pour que nous les Pâtissier nous puissions Puissions ajuster le tir plus facilement derrière et éviter de trop gros écart de commande qui serait trop grosse ou trop petite et donc qui engendrerait trop de perdre. Peut-être un algorithme en informatique pourrait nous aider à mieux se préparer pour éviter les pertes et passer de meilleures commandes en anticipant au mieux le nombre de clients potentiels qui vont venir dans lesjours suivant ou pour les commandes du jour au lendemain. Dis-moi s’il existe des choses comme ça, dis-moi ce que tu en penses et ce que nous pourrions créer.

**ChatGPT a dit :**

Ton idée est excellente et très pertinente pour optimiser la gestion des commandes en pâtisserie. Il existe déjà des solutions qui s’appuient sur l’analyse de données et l’intelligence artificielle pour prévoir la demande en fonction de plusieurs paramètres, mais elles ne sont pas toujours adaptées spécifiquement à un laboratoire de pâtisserie. Voici quelques pistes et solutions possibles :

**1. Applications existantes et données disponibles**

Il existe des logiciels de gestion des stocks et de prévision de la demande, mais peu intègrent directement des facteurs comme les événements politiques, économiques, ou la fréquentation touristique spécifique à Paris. Voici quelques outils intéressants :

* **Too Good To Go & Phenix** : Ces applications luttent contre le gaspillage, mais elles ont aussi des algorithmes qui optimisent les stocks en fonction des ventes passées.
* **Lightspeed, Square ou Openbravo** : Ces logiciels de caisse pour commerces intègrent parfois des modules de prévision des ventes basés sur l’historique.
* **PredictHQ** : Fournit des analyses prédictives basées sur des événements à venir (concerts, grèves, fêtes locales…).
* **Google Trends & INSEE** : Ces sources permettent d’analyser des tendances sur la consommation et le tourisme en fonction de la période de l’année.

**2. Ce qu’on pourrait créer pour un laboratoire de pâtisserie**

Si aucune solution actuelle ne répond exactement à tes besoins, on pourrait concevoir une application ou un algorithme personnalisé. Voici ce qu’il pourrait inclure :

* **Analyse des ventes passées** : Enregistrer et analyser l’historique des ventes en fonction des jours, saisons, événements et conditions économiques.
* **Données externes (tourisme, météo, événements, grèves, fêtes locales)** : En intégrant des API publiques (Open Data Paris, Google Trends, prévisions touristiques), on pourrait prédire la fréquentation dans tes boutiques.
* **Modèle d’intelligence artificielle (Machine Learning)** : Un algorithme pourrait apprendre des ventes passées et ajuster automatiquement les prévisions de commande.
* **Tableau de bord interactif** : Accessible aux responsables pour ajuster les commandes en fonction des recommandations du modèle.

**3. Prochaines étapes pour développer cela**

Si tu veux proposer une solution à ton employeur, voici comment avancer :

1. **Recueillir des données internes** : Analyser plusieurs mois (voire années) de ventes, comparer avec des événements et conditions météo.
2. **Identifier les sources externes utiles** : Trouver les données sur le tourisme, les événements et l’économie locale.
3. **Créer un prototype** : Un petit programme qui prend ces données et fait une première analyse des tendances.
4. **Automatiser la prédiction** : Entraîner un algorithme pour donner des prévisions et suggérer les quantités idéales à commander.

👉 **On pourrait commencer par un outil simple en Python qui analyse tes ventes et les compare à des événements et conditions météo**. Intéressé par cette approche ?

4o

**Vous avez dit :**

Ne vaut-il pas mieux créer un site en symfony et react en front intégrant les script python ? c'est plutôt ma spécialité

**ChatGPT a dit :**

Mémoire mise à jour

Oui, totalement ! Puisque Symfony et React sont ta spécialité, le mieux serait de développer une application web structurée comme suit :

**Architecture de l'application**

* **Back-end (Symfony)** : Gérer l’authentification, stocker les données de ventes, récupérer les données externes (météo, événements, tourisme).
* **Front-end (React)** : Afficher des tableaux de bord interactifs avec les prévisions de commandes et les recommandations.
* **Scripts Python (ML & analyse des données)** : Analyser l’historique des ventes et prévoir la demande avec du Machine Learning.

**Les étapes du développement**

1. **Concevoir la base de données** (Symfony + Doctrine) :
   * Table ventes (produit, quantité, date, jour de la semaine, événement associé).
   * Table événements (nom, date, impact potentiel).
   * Table prévisions (générées par l’algorithme).
2. **Créer un script Python** qui analyse ces données et prédit la demande.
   * Récupérer les ventes passées, les corréler avec les jours de la semaine, la météo, les événements.
   * Utiliser un modèle ML (ex. scikit-learn) pour faire une prévision.
   * Exposer un endpoint API en Symfony pour exécuter et récupérer les prédictions.
3. **Créer une interface React** :
   * Tableau de bord avec visualisation des tendances.
   * Système de recommandations pour aider à la prise de commande.

Cette solution serait complète et pourrait même évoluer vers une application SaaS pour d’autres pâtissiers ! Veux-tu qu’on commence par la structure de la base de données et un premier script Python ? 🚀

4o

**Vous avez dit :**

quelle autre entité pourrions nous utiliser ? vente, événement, prevision, user, media pour rentrer des photos des vidéos que les responsables pourraient se communiquer Et pour rentrer les nouvelles données récolter par exemple sous forme de PSD. Je pense qu'il faut créer un algorithme qui puisse prendre en compte pour son modèle prédictif les évènements politiques de la capitale sportif économique international mondial la météo et peut-être des évènements musicaux et de théâtre notamment sur des grosses programmations qui pourraient attirer du monde. La boutique s'appelle Angelina et est à Paris Premier. Il y a plusieurs boutiques dans Paris dont une notamment à Versailles. Qu'en penses-tu partage-moi de nouvelles idées ou de nouvelles choses qui pourraient être utile

**ChatGPT a dit :**

Mémoire mise à jour

Excellente approche ! Ajouter une entité Media pour stocker des photos, vidéos et documents comme des fichiers PSD est une très bonne idée pour faciliter la communication entre responsables.

**Autres entités utiles pour enrichir l'application :**

1. **Boutique** : Chaque boutique a des spécificités différentes (localisation, affluence).
   * id, nom, adresse, capacité\_max, responsable\_id
   * Relation avec Vente, Prévision, User
2. **Produit** : Pour suivre les ventes par type de pâtisserie.
   * id, nom, catégorie, coût\_production, prix\_vente, stock
   * Relation avec Vente
3. **Tourisme** : Pour intégrer des données sur la fréquentation touristique à Paris et Versailles.
   * id, date, nb\_touristes, source\_données
   * Relation avec Prévision
4. **Météo** : Indispensable pour voir l’impact de la météo sur la fréquentation.
   * id, date, température, précipitations, conditions
   * Relation avec Prévision
5. **Événement** (Politique, Sportif, Culturel) : Déjà prévu, mais on pourrait ajouter une catégorisation plus fine.
   * id, type, nom, date, lieu, impact\_estimé
   * Relation avec Prévision
6. **Prévision** : Stocke les prédictions générées par l’algorithme.
   * id, date, boutique\_id, prévision\_clientèle, taux\_confiance
   * Relation avec Boutique, Vente, Tourisme, Météo

**Idées complémentaires :**

* **Ajouter une API externe pour récupérer des données en temps réel** (météo, événements, affluence touristique).
* **Créer un dashboard interactif avec des graphiques dynamiques (React + Chart.js)**.
* **Développer une gestion des alertes** pour informer les responsables des prévisions critiques (affluence exceptionnelle ou faible).

Avec ces entités et fonctionnalités, l’application pourrait devenir un outil puissant pour Angelina et potentiellement être élargie à d’autres pâtisseries ! 🚀

Veux-tu que je t’aide à structurer la base de données en Symfony pour commencer ?

4o

**Vous avez dit :**

Il faut pouvoir calculer des données statistiques de l'impact estimé en pourcentage par exemple de l'augmentation ou de la diminution de la fréquentation

**ChatGPT a dit :**

Mémoire mise à jour

Bonne idée ! On pourrait ajouter un système de calcul d'impact basé sur plusieurs critères :

**Comment calculer l’impact estimé en pourcentage ?**

L’idée serait de comparer la fréquentation actuelle avec des périodes similaires du passé pour estimer une variation en pourcentage.

**1. Formule de base**

Impact estimeˊ=Freˊquentation preˊvue−Freˊquentation historiqueFreˊquentation historique×100\text{Impact estimé} = \frac{\text{Fréquentation prévue} - \text{Fréquentation historique}}{\text{Fréquentation historique}} \times 100Impact estimeˊ=Freˊquentation historiqueFreˊquentation preˊvue−Freˊquentation historique​×100

Où :

* **Fréquentation prévue** = Résultat de l’algorithme basé sur événements, météo, etc.
* **Fréquentation historique** = Moyenne des fréquentations lors d’événements similaires.

**2. Facteurs influençant l’impact**

On pourrait pondérer les résultats en fonction de plusieurs critères :

* **Impact des événements passés similaires** (ex : Roland-Garros augmente la demande de +20%)
* **Météo** (pluie = moins de clients, beau temps = plus de touristes)
* **Tendances économiques** (pouvoir d’achat, inflation)
* **Jours fériés et vacances scolaires**

**3. Implémentation dans l’application**

* Ajouter une colonne impact\_estimé dans Prévision pour stocker le pourcentage.
* Développer un script Python qui recalcule cet impact automatiquement.
* Afficher les variations sur le dashboard avec des indicateurs visuels (flèches rouges/vertes, graphiques interactifs).

Avec ce système, les responsables auraient une meilleure visibilité pour ajuster leurs commandes. Tu veux qu’on commence par structurer la table Prévision et définir comment récupérer les données historiques ?

Synthèse de la création des entités :

**Entités principales**

1. **User** (id, nom, email, rôle).
2. **Boutique** (id, nom, adresse, capacité\_max).
3. **Vente** (id, date, boutique\_id, produit\_id, quantité).
4. **Produit** (id, nom, catégorie, prix\_vente, coût\_production).
5. **Événement** (id, type, date, impact\_estimé). Données des événements influençant la fréquentation) -> (Politique, Sportif, Culturel) : Déjà prévu, mais on pourrait ajouter une catégorisation plus fine.
6. **Météo** (id, date, température, précipitations).
7. **Tourisme** (id, date, nb\_touristes).
8. **Prévision** (id, date, boutique\_id, fréquentation\_prévue, impact\_estimé). (Données de prévision de fréquentation)
9. **Media** (id, fichier, type, boutique\_id, utilisateur\_id).
10. **Commande** *(une commande générée pour une boutique et envoyée au labo)*
11. **CommandeLigne** *(détail des produits dans une commande)*
12. **Production :** **Rassembler toutes les commandes de toutes les boutiques sur une seule feuille distribuée au labo.**
13. **Notification** *(alertes et messages pour ajustements des commandes et stocks)*

**📌 Ajustement pour Production**

👉 **Objectif : Rassembler toutes les commandes de toutes les boutiques sur une seule feuille distribuée au labo.**

**Nouveaux champs pour Production :**

* id *(Identifiant unique de la production globale du jour)*
* date\_production *(Jour concerné par la production)*
* statut *(en attente, en cours, terminé...)*
* pdf\_commande *(Fichier PDF généré avec toutes les commandes consolidées, imprimable pour distribution)*

**📌 Relations Clés**

✔ **des User (responsables) gère une Boutique**  
✔ **Une Boutique a plusieurs Vente et Prévision**  
✔ **des Événement influence les Prévision**  
✔ **La Météo et le Tourisme impactent les Prévision**

**📌 Schéma des relations**

* **Une Boutique peut passer plusieurs Commandes.**
* **Une Commande contient plusieurs CommandeLigne.**
* **Une CommandeLigne est liée à un seul Produit.**
* **La Production regroupe toutes les Commandes d’un même jour et génère un document unique.**

Faire une navbar avec les onglets pour :

Entités principales 1. User (id, nom, email, rôle). 2. Boutique (id, nom, adresse, capacité\_max). 3. Vente (id, date, boutique\_id, produit\_id, quantité). 4. Produit (id, nom, catégorie, prix\_vente, coût\_production). 5. Événement (id, type, date, impact\_estimé). Données des événements influençant la fréquentation) -> (Politique, Sportif, Culturel) : Déjà prévu, mais on pourrait ajouter une catégorisation plus fine. 6. Météo (id, date, température, précipitations). 7. Tourisme (id, date, nb\_touristes). 8. Prévision (id, date, boutique\_id, fréquentation\_prévue, impact\_estimé). (Données de prévision de fréquentation) 9. Media (id, fichier, type, boutique\_id, utilisateur\_id). 10. Commande (une commande générée pour une boutique et envoyée au labo) 11. CommandeLigne (détail des produits dans une commande) 12. Production : Rassembler toutes les commandes de toutes les boutiques sur une seule feuille distribuée au labo. 13. Notification (alertes et messages pour ajustements des commandes et stocks)

Onglets : boutique – commande – offre – concours meilleur anticipation de commande( ou challenge ou recompense ) – evenements – statistique

**Voici la nouvelle structure de navigation que j'ai créée, organisée autour des fonctionnalités clés de l'application :**

* 1. **Tableau de bord** (/)

Vue d'ensemble des KPIs

Alertes et notifications importantes

Actions rapides

* 1. **Gestion Commandes** (/orders)

Nouvelles Commandes

Historique

Production

* 1. **Prévisions & Analytics** (/analytics)

Prévisions Ventes

Historique Ventes

Performance KPIs

* 1. **Boutiques** (/stores)

Liste des Boutiques

Performance

* 1. **Catalogue** (/products)

Produits

Catégories

Saisonnalité

* 1. **Événements** (/events)

Calendrier

Météo

Tourisme

* 1. **Challenges** (/challenges)

Objectifs

Classement

Récompenses

* 1. **Équipe** (/team)

Membres

Rôles

* 1. **Tableau de bord** (/)
  2. **Gestion Commandes** (/orders)
  3. **Prévisions & Analytics** (/analytics)
  4. **Boutiques** (/stores)
  5. **Catalogue** (/products)
  6. **Événements** (/events)
  7. **Challenges** (/challenges)
  8. **Équipe** (/team)