Rapport

Enjoy

Sommaire

Présentation du projet

Enjoy! est une plateforme de livraison de repas à domicile, faites pour offrir aux utilisateurs un large choix de restaurants et aux établissements partenaires une solution pratique et efficace pour gérer les commandes et les livraisons.

Grâce à **Enjoy!**, les clients peuvent accéder à une variété de cuisines et de restaurants à proximité, tout en profitant de la simplicité de commander depuis chez eux ou depuis n'importe où. La plateforme se charge de transmettre les commandes aux restaurants partenaires et met à leur disposition une flotte de livreurs qualifiés pour acheminer les plats préparés directement chez les clients, rapidement et en toute sécurité.

En se positionnant comme un simplifieur pour la vie de nos clients, **Enjoy!** permet aux restaurants de se concentrer sur leur cœur de métier, préparer des plats savoureux tout en éliminant les contraintes logistiques liées à la gestion de la livraison.

Partie Technique

Modèle relationnel :

restaurant(idresto, nom, adresse, frais, idcarte, codepostal)
client(email, nom, prenom, numtel, pointfid, idparrain, codepostal)
livreur(matricule, nom, prenom, numpro)
plat(idplat, nom, prix, description, photo)
carte(idcarte)
commande(idcommande, montant, adresse, statutlivreur, statutcommande, codepostal)
motsclefs(motcle)
cb(numcarte, propriétaire, codesecret, datexp, cvc, email)
ville(codepostal, nom)
jour(jour)
fermetureexep(idferm, motif)
compose(idplat, idcommande, quantite)
travaille(matricule, codepostal)
figure(idcarte, idplat)
ferme(idferm, idresto, heuredebut, heurefin, date)
ouvre(jour, idresto, heuredebut, heurefin)contient(motclefs, idresto)
panier(email, idplat, quantite)
compteclient(email, motdepasse)
comptelivreur(matricule, motdepasse)
compterestau(idresto, motdepasse)
contient(motclefs, idresto)

FK :

idcarte dans restaurant fait référence à idcarte dans carte
codepostal dans restaurant fait référence à codepostal dans ville
idparrain dans client fait référence à email dans client
codepostal dans client fait référence à codepostal dans ville
codepostal dans commande fait référence à codepostal dans ville
idplat dans compose fait référence à idplat dans plat
idcommande dans compose fait référence à idcommande dans commande
matricule dans travaille fait référence à matricule dans livreur
codepostal dans travaille fait référence à codepostal dans ville
idcarte dans figure fait référence à idcarte dans carte
idplat dans figure fait référence à idplat dans plat
idresto dans ferme fait référence à idresto dans restaurant
idferm dans ferme fait référence à idferm dans fermetureexep
idresto dans ouvre fait référence à idresto dans restaurant
jour dans ouvre fait référence à jour dans jour
motsclefs dans contient fait référence à motscles dans motsclefs
idresto dans contient fait référence à idresto dans restaurant
email dans panier fait référence à email dans client
idplat dans panier fait référence a idplat dans plat
email dans compteclient fait reference à email dans client
matricule dans comptelivreur fait référence à email dans livreur
idresto dans compterestau fait référence à idresto dans restaurant

Types de données :

Pour les clés primaires, telles que les identifiants des restaurants (idresto), des commandes (idcommande) et des livreurs (matricule), nous avons opté pour des types serial pour faciliter la gestion des identifiants.
Les attributs qui contiennent des valeurs textuelles dont on connaît plus ou moins la longueur, comme les noms et les adresses, ont été définis avec des types varchar avec une longueur maximale adaptée.

Contraintes :

Les contraintes NOT NULL ont été appliquées sur les colonnes où une valeur est indispensable au bon fonctionnement de l'application (par exemple, le montant des commandes ou le nom des clients).
Concernant les statuts de commandes et de livreurs, nous avons utilisé des contraintes CHECK pour limiter les valeurs possibles à un ensemble précis de chiffres, entre 0 et 3. Cela permet de garantir l'intégrité des données concernant l'état des commandes et des livreurs :
Pour statutcommande, les valeurs possibles sont :

- 0 : Indisponible,
- 1 : En attente,
- 2 : En livraison,
- 3 : Livré.

Pour statutlivreur, les valeurs possibles sont :

- 0 : Indisponible,
- 1 : En service,
- 2 : En attente de commande
- 3 : En livraison.
- 4 : Commande livré

Le client verra les restaurants présent dans son code postal.

Vue pour les statistiques :

Nous avons créé une vue pour calculer les statistiques des commandes en fonction des mots-clés des plats. Par exemple, si deux commandes incluent le mot-clé "pizza" et une autre "burger", la vue retournera des statistiques indiquant que 66 % des commandes contiennent "pizza" et 33 % contiennent "burger".

Requêtes de test :

Des requêtes de test ont été ajoutées en commentaires dans le script de création de la base de données.

Fonctionnalités

Inscription et connexion :

Les pages de connexion pour les clients, les livreurs et les restaurants sont tous gérer de la même manière utilisant leur table respectivement (compteclient, comptelivreur, compterestau) pour récupérer les identifiants, et les mots de passes (les mots de passes sont stockés hachés avec passlib) on récupère simplement le mot de passe de la table associé à l'identifiant entré dans le formulaire, et on récupère le résultat si nous avons un résultat on vérifie que le mot de passe présent dans la table et le mot de passe entré sont les même grâce à pwd_context.verify de passlib.
Si on a bien réussi à trouvé un identifiant et un mot de passe correct dans la table on le renvoi vers l'accueil_connecte de son genre (client, livreur ou restaurant) et en même temps on l'ajoute à la session pour avoir son email, son matricule ou son idresto.

Les pages d'inscriptions reprennent également tous le même concept, on récupère les informations entrés, et on les insère dans la table de leur genre (restaurant, client, livreur) et on ajoute les informations de connexion dans leur compte (livreur, client ou restaurant).
On vérifie également que le code postal est existant dans la table ville, si elle ne l'est pas on ajoute à ville le code postal entré et la ville entré.

Les pages d'accueil et accueil connecté :

La page d'accueil nous montre une page de connexion et d'inscription pour les clients, une page de connexion pour les livreurs (les livreurs ne peuvent pas créer leur compte, ils doivent passer par les admins de la plateforme pour créer leur compte), et une page de connexion pour les restaurants (avec un lien en dessous si ils ne sont pas inscrit).

Les pages d'accueils connectés donnent accès pour chacune des catégories (client, restaurant et livreur) un logo profil qui leur donne accès à leur profile, ainsi qu'un bouton déconnexion. Et un bouton panier pour les clients ainsi qu'une liste déroulante de recherche avec les mots clés de tout les mots clés présent dans la table mot clé.

Les pages de profils :

Un client peut accéder via son profil à son total de points de fidélité, à son historique de commande, et les commandes en cours, si les commandes en cours sont encore en attente (c-a-d qu'aucun livreur n'a pas encore récupéré la commande), le client peut annuler la commande qui sera supprimer de la table (via un DELETE). Une fois la commande livré, le client à la possibilité de noter la commande et cela lui ajoutera 5 points dans son montant de points de fidélité.

Un livreur peut voir ses commandes traités, et les commandes présentes dans son code postal qui sont en attente pour les récupérer. Une fois récupérer le client peut mettre à jour son état, (en livraison, livré).

Un restaurant peut voir tout ses plats présent dans la carte, en ajouter à sa carte, supprimer des plats

Recherche :

Nous avons une barre déroulante de recherche qui nous montre tout les mots clés présent dans la table mot clé, en choisissant l'un deux, on aura accès à tout les restaurant qui possède ce mot clé. (SELECT idresto, nom, adresse, codepostal FROM restaurant NATURAL JOIN contient WHERE motsclefs LIKE %s", (motclef,)

Restaurant :

Dans la page du restaurant, nous avons les informations du lieu, adresse, code postal, et ville. Les horaires d'ouvertures, les fermetures exeptionnelle. Et le menu qui nous permet d'ajouter la quantité que nous souhaitons dans le panier. INSERT INTO panier VALUES(%s, %s, %s)", (session['user_id'], idplat, quantite)

Panier :

Dans la page du panier nous avons les produits que nous avons ajoutés auparavant dans le panier. Et nous voyons également les prix, et le montant total de la commande, puis nous pouvons entrer nos informations pour commander ce qui ajoutera les informations à la table commande. (problème d'insertion des données de la commande dans commande et de la suppresion du panier une fois commander.)

Améliorations

Pour améliorer notre projet, nous aurions pu avec plus de temps essayer de faire un rayon autour du code postal du client pour afficher ses restaurants à proximité.

Nous aurions aussi pu corriger les problèmes du panier quand on veut passer commande, actuellement quand on rempli les informations et qu'on commande les informations ne s'insèrent pas correctement dans la table commande.

Nous aurions pu ajouter un système premium pour avoir la livraison gratuite et gagnez plus de points de fidélités en ajoutant simplement une table premium et qu'on insère les informations de connexion du client et on vérifierait à chaque commande et à chaque notation si ils sont dans la table premium si oui on ajoutera + de points et on rendrait les frais à 0€.

On aurait aussi pu mieux gérer les CB.

On aurait pu ajouter dans le panier des boutons pour changer la quantité des plats qu'on a déjà dans le panier, et également supprimer les plats qu'on veux.

Organisation et répartition du travail

Nous avons répartis le travail de manière équitable. Nous travaillons principalement à distance et s'envoyer les améliorations par discord. Et pendant les TP nous essayons de faire les parties de codes plus compliqués. Nous avons répartis le travail en 50/50.