

Rapport Final - Projet Bon Voyage

Sommaire

1. Introduction
2. Manuel utilisateur
3. Section technique (étapes 1 et 2)
4. Description des fonctionnalités techniques
5. Pistes d'amélioration
6. Organisation et répartition du travail

1.Introduction

Le projet Bon Voyage consiste à développer une plateforme web permettant à une compagnie de voyages de gérer efficacement ses activités. Cette plateforme offre deux interfaces distinctes : une pour les employés (administrateurs) et une pour les clients. L'objectif principal est de faciliter la planification, la gestion et la réservation des voyages, tout en garantissant une expérience utilisateur optimale.

Contexte

La compagnie Bon Voyage souhaite moderniser ses processus de gestion des voyages. L'objectif est de fournir un outil centralisé permettant :

- Une gestion fluide des comptes employés et clients.
- La création, modification et suppression de voyages et d'étapes.
- Une interface intuitive pour les clients, facilitant la recherche et la réservation des voyages.

Ce projet est réalisé dans le cadre d'un module universitaire sur les bases de données, mettant en pratique des compétences en modélisation, conception relationnelle, et développement web avec Flask et PostgreSQL.

2.Manuel utilisateur

2.1. Côté Administrateur

Fonctionnalités principales :

1. Connexion :

- Les administrateurs se connectent via une interface sécurisée avec identifiant et mot de passe.

2. Gestion des comptes utilisateurs :

- Créer, modifier ou supprimer des comptes employés au sein de leur agence.
- Exemple de scénario : Ajouter un nouvel employé avec un identifiant unique.

3. Gestion des voyages et étapes :

- Ajouter un nouveau voyage, définir ses dates et son tarif.
- Configurer les étapes d'un voyage (ville, transport, logement, etc.).
- Supprimer ou modifier les informations des voyages existants.

4. Suivi des réservations :

- Consulter les listes des clients réservés par voyage.

Accès rapide aux pages :

- Page de gestion des comptes : `/configUser`
- Page de gestion des voyages : `/configTrip`

2.2. Côté Client

Fonctionnalités principales :

1. Inscription et connexion :

- Les clients peuvent créer un compte ou se connecter à leur espace personnel.
- Exemple : Créer un compte avec un e-mail et un mot de passe haché pour la sécurité.

2. Recherche de voyages :

- Filtrer les voyages par destination, date ou disponibilité.
- Exemple : Trouver un voyage vers Paris pour le mois de juillet.

3. Consultation des détails :

- Voir les informations complètes sur un voyage (coût, étapes, transport, etc.).

4. Gestion des réservations :

- Réserver un voyage si les dates ne se chevauchent pas avec une autre réservation.
- Consulter l'historique des voyages réservés.

Accès rapide aux pages :

- Page de recherche des voyages : `/page_recherche`
- Page de détails des voyages : `/detail/<id_voyage>`

3.Section technique (étapes 1 et 2)

Schéma relationnel :

client = (id_utilisateur, nom, sexe , courriel, prenom , tel , adresse);

type_etape = (id_et_type , valeur);

moyen_transport = (id_transport , valeur);

type_logement = (id_type_logement, valeur);

Agence = (id_agence ,nom);

langue = (id_langue , langue);

Travailleur = (id_travailleur , est_Responsable,mdp , login , #id_agence);

voyage = (id_voyage, nom, reservation,date_debut , date_de_fin, cout_par_personne , #id_agence);

pays = (id_pays, nom, description , #id_langue);

Ville = (id_ville, nom , #id_pays);

logement = (id_logement, #id_ville, #id_type_logement);

Etape = (id_etape, visa, date_arrivée , date_depart , #id_ville, #id_et_type, #id_logement, #id_transport, #id_voyage);

participe = (#id_utilisateur, #id_voyage);

FOREIGN KEY : Travailleur(id_agence) fait référence à agence(id_agence)

 voyage(id_agence) fait référence à agence(id_agence)

 pays(id_langue) fait référence à langue(id_langue)

 ville(id_pays) fait référence à pays(id_pays)

 logement(id_ville) fait référence à ville(id_ville)

 logement(id_type_logement) fait référence à type_logement(id_type_logement)

 etape(id_ville) fait référence à ville(id_ville)

 etape(id_et_type) fait référence à type_logement(id_et_type)

 etape(id_logement) fait référence à logement(id_logement)

 etape(id_transport) fait référence à moyen_transport(id_transport)

 etape(id_voyage) fait référence à voyage(id_voyage)

Création de la Base de Données :

Cette base de données a pour but de gérer les informations nécessaires pour une plateforme de voyages.

Tables :

- **Client** : Stocke les informations des clients obligatoires pour l'inscription.
- **Agence** : Stocke les informations des agences.
- **Travailleur** : Stocke les employés. Grâce à ON DELETE SET NULL, on garde les informations sur les employés même si leur agence est supprimée.
- **Voyage** : Liste les voyages, avec la même logique pour l'agence (ON DELETE CASCADE).
- **Pays et Ville** : Pour chaque ville, on a un lien avec un pays. En cas de suppression d'un pays, les villes associées sont également supprimées (ON DELETE SET NULL).
- **Logement** : Associe chaque logement à une ville, avec suppression en cascade.
- **Etape** : Stocke les informations sur les étapes (modes de transport, logement, etc.).
- **Participe** : Gère les réservations des clients sur les voyages, avec suppression automatique des réservations si le client ou le voyage est supprimé.

Pour réaliser cette base de données, nous nous sommes appuyés sur notre modèle entité-association. Nous avons ajouté des informations supplémentaires qui ne pouvaient pas être définies directement dans le modèle, comme les contraintes de suppression.

Nous avons également remodelé la base avec l'ajout de vues et ajouté un attribut réservation (booléen) à la table Voyage pour indiquer si un voyage est disponible ou non. Les vues ont été créées selon les demandes du client.

Avec ces modifications, la base est prête à gérer les données comme prévu dans le cahier des charges, tout en maintenant un bon équilibre entre suppression automatique et conservation de l'historique.

4.Description des fonctionnalités techniques

4.1. Gestion des comptes administrateurs

- **Requête pour afficher les employés d'une agence :**

```
SELECT * FROM Travailleur WHERE id_agence = %s;
```

- **Requête pour modifier les informations d'un employé :**

```
UPDATE Travailleur SET login = %s, mdp = %s WHERE id_travailleur = %s;
```

4.2. Gestion des voyages

- **Requête pour ajouter un voyage :**

```
INSERT INTO voyage (nom, date_debut, date_de_fin, cout_par_personne,  
reservation, id_agence)  
VALUES (%s, %s, %s, %s, %s, %s);
```

- **Requête pour mettre à jour un voyage :**

```
UPDATE voyage SET nom = %s, date_debut = %s, date_de_fin = %s WHERE  
id_voyage = %s;
```

4.3. Réservations client

- **Requête pour afficher les voyages disponibles :**

```
SELECT * FROM voyage WHERE reservation = true AND date_debut >=  
CURRENT_DATE;
```

- **Requête pour ajouter une réservation :**

```
INSERT INTO participe (id_utilisateur, id_voyage) VALUES (%s, %s);
```

5.Pistes d'amélioration

Interface utilisateur :

- Ajouter des fonctionnalités avancées de filtrage pour les clients (par prix, type de voyage, etc.).
- Améliorer l'ergonomie des pages.

Notifications :

- Implémenter des notifications par e-mail pour les réservations et les mises à jour des voyages.

Internationalisation :

- Supporter plusieurs langues pour attirer une audience internationale.

6.Organisation et répartition du travail

- **Côté Administrateur** : Conçu et développé par Herreros Tanguy.
- **Côté Client** : Conçu et développé par Ben-Ammar Sasa.