

**Université Paris Nanterre**

Licence MIASHS - Parcours MIAGE

Semestre 4 • Programmation web et introduction PHP

# Rapport de Projet

## Application de Planification de Voyages

### *"TravelDream"*

**Réalisé par :**

Melissa AKLI (44008939)

Nikola KALETKA (42012015)

**Encadré par :**

Thibault Anani

Lien vers le projet : <https://github.com/NikolaKaletka/Project-S4.git>

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Contexte du projet . . . . .	3
1.2	Objectifs et problématique . . . . .	3
1.3	Méthodologie de travail . . . . .	3
<b>2</b>	<b>Analyse des Besoins</b>	<b>3</b>
2.1	Besoins fonctionnels . . . . .	3
2.2	Besoins non fonctionnels . . . . .	4
2.3	Spécificités techniques . . . . .	4
<b>3</b>	<b>Conception de la Base de Données</b>	<b>5</b>
<b>4</b>	<b>Architecture Technique</b>	<b>5</b>
4.1	Structure du code . . . . .	5
4.2	Organisation des fichiers . . . . .	6
4.3	Configuration du projet . . . . .	7
4.4	Sécurité . . . . .	8
<b>5</b>	<b>Fonctionnalités</b>	<b>8</b>
5.1	Authentification . . . . .	8
5.1.1	Inscription . . . . .	8
5.1.2	Connexion . . . . .	9
5.1.3	Réinitialisation du mot de passe . . . . .	9
5.1.4	Déconnexion . . . . .	9
5.2	Gestion des Voyages . . . . .	10
5.2.1	Création et Modification d'un Voyage . . . . .	10
5.2.2	Visualisation des voyages de l'utilisateur ( <code>mesvoyages.php</code> ) . . . . .	11
5.2.3	Visualisation Détailée d'un Voyage ( <code>voyage_detail.php</code> ) . . . . .	11
5.2.4	Partage d'un Voyage . . . . .	12
5.3	Carte interactive . . . . .	12
5.4	Assistant de voyage (chatbot) . . . . .	13
5.5	Recherche de destinations ( <code>destination.php</code> ) . . . . .	14
<b>6</b>	<b>Interface Utilisateur</b>	<b>14</b>
6.1	Charte graphique . . . . .	14
6.1.1	Palette de couleurs . . . . .	14
6.1.2	Typographie . . . . .	15
6.1.3	Iconographie . . . . .	15
6.1.4	Éléments d'interface . . . . .	15
6.2	Maquettes et wireframes . . . . .	15
6.3	Responsive Design . . . . .	15
6.3.1	Adaptations spécifiques . . . . .	16

<b>7 Tests et Validation</b>	<b>16</b>
7.1 Stratégie de test . . . . .	16
7.1.1 Environnement de test . . . . .	16
7.1.2 Outils et méthodes . . . . .	16
7.2 Tests fonctionnels . . . . .	16
<b>8 Difficultés Rencontrées et Solutions</b>	<b>17</b>
8.1 Problèmes techniques . . . . .	17
8.1.1 Compatibilité entre navigateurs . . . . .	17
8.1.2 Gestion dynamique des formulaires complexes . . . . .	17
8.1.3 Intégration des API externes (Leaflet/Nominatim) . . . . .	17
<b>9 Améliorations Futures</b>	<b>18</b>
9.1 Fonctionnalités à développer . . . . .	18
9.1.1 Application mobile native . . . . .	18
9.1.2 Amélioration des performances . . . . .	18
9.1.3 Renforcement de la sécurité . . . . .	18
9.1.4 Intelligence artificielle et personnalisation avancée . . . . .	18
<b>10 Conclusion</b>	<b>19</b>
10.1 Synthèse du projet . . . . .	19
10.2 Compétences acquises . . . . .	19
10.3 Perspectives . . . . .	20
<b>11 ANNEXE : Schéma de bases de données</b>	<b>21</b>
<b>12 ANNEXE : Lancement du projet</b>	<b>22</b>
12.1 Authentification . . . . .	22
12.1.1 Inscription . . . . .	22
12.1.2 Connexion . . . . .	22
12.1.3 Mot de passe oublié . . . . .	23
12.1.4 Déconnexion . . . . .	23
12.2 Accueil et liste des destinations . . . . .	23
12.2.1 Carte des destinations . . . . .	25
12.3 Chatbot . . . . .	26
12.4 Gestion des voyages . . . . .	30
12.4.1 Profil et liste des voyages . . . . .	30
12.4.2 Ajouter ou modifier un voyage . . . . .	31
12.4.3 Détail d'un voyage . . . . .	34
12.4.4 Partager un voyage . . . . .	35
12.4.5 Imprimer un voyage . . . . .	36

# 1 Introduction

## 1.1 Contexte du projet

Dans le cadre de ce projet, réalisé en binôme par Melissa et Nikola, nous avons développé une application web nommée "*TravelDream*" permettant aux utilisateurs de planifier leurs voyages de manière complète et organisée. Ce projet a aussi pour objectif d'augmenter nos compétences en développement web, en conception de bases de données et en modélisation.

## 1.2 Objectifs et problématique

La planification de voyages est souvent une tâche complexe qui implique la gestion de nombreux éléments : transport, hébergement, activités, budget, etc. Notre application vise à centraliser toutes ces informations en un seul endroit, offrant ainsi aux utilisateurs un outil pratique pour organiser leurs déplacements et profiter pleinement de leurs expériences de voyage.

La problématique centrale que nous avons cherché à résoudre est la suivante : comment simplifier et centraliser le processus de planification de voyage pour les utilisateurs, en leur offrant une interface intuitive et des fonctionnalités complètes ?

Pour ce faire notre application va combiner une base de données bien structurée, une interface utilisateur intuitive et responsive, des fonctionnalités adaptées aux besoins des voyageurs (création, modification et partage de voyages, carte interactive) et un chatbot d'aide à la décision

## 1.3 Méthodologie de travail

Pour mener à bien ce projet, nous avons adopté une approche méthodique en plusieurs phases :

- **Phase d'analyse** : Définition des besoins fonctionnels et non fonctionnels
- **Phase de conception** : Conception du schéma relationnel de la base de données, et définition de l'architecture technique du projet.
- **Phase de développement** : Implémentation des fonctionnalités par modules (authentification, gestion des voyages, activités, etc.), développement de l'interface utilisateur, et intégration des différentes parties.
- **Phase de test** : Vérification manuelles du bon fonctionnement des fonctionnalités et correction des bugs identifiés.
- **Phase de finalisation** : Création d'un rapport

Tout au long du projet, nous avons maintenu une communication constante entre les membres de l'équipe pour assurer la cohérence du développement et résoudre rapidement les problèmes rencontrés.

# 2 Analyse des Besoins

## 2.1 Besoins fonctionnels

Notre application de planification de voyages *TravelDream* a été conçue pour répondre à un ensemble de besoins fonctionnels précis, identifiés à partir des attentes des utilisateurs potentiels. Ces besoins définissent les capacités et services que notre système doit offrir :

- **Gestion des utilisateurs**

- Création d'un compte utilisateur avec nom, email et mot de passe
- Connexion sécurisée
- Réinitialisation du mot de passe
- Déconnexion et gestion de session
- **Gestion des voyages**
  - Création d'un nouveau voyage (destination, dates)
  - Consultation de la liste des voyages
  - Modification des informations d'un voyage
  - Partage d'un voyage avec d'autres utilisateurs
  - Vue détaillée d'un voyage
- **Gestion des activités** (associées à un voyage)
- **Gestion des transports** (associés à un voyage)
- **Gestion des logements** (associés à un voyage)
- **Gestion des restaurants** (associés à un voyage)
- **Gestion d'une checklist avant le départ** (associée à un voyage)
- **Gestion des transports urbains** (associés à un voyage)
- **Fonctionnalités supplémentaires**
  - Carte interactive des destinations
  - Assistant virtuel pour recommandations

## 2.2 Besoins non fonctionnels

Au-delà des fonctionnalités, notre application devait également répondre à des exigences non fonctionnelles, qui définissent la qualité et les performances du système :

- **Sécurité**
  - Protection des données personnelles des utilisateurs
  - Hachage des mots de passe pour éviter leur stockage en clair
  - Utilisation de requêtes préparées pour prévenir les injections SQL
  - Validation des entrées utilisateur pour éviter les attaques XSS
- **Utilisation**
  - Interface facile à prendre en main
  - Design responsive adapté aux différents appareils (ordinateurs, smartphones)
  - Temps de chargement rapide des pages
  - Feedback visuel pour les actions utilisateur (par exemple le curseur qui change quand c'est un bouton)

## 2.3 Spécificités techniques

Le développement de notre application a été soumis à plusieurs spécificités (parfois contraintes) techniques, qui ont guidé nos choix technologiques et notre approche tout au long du projet :

- **Technologies utilisées**
  - PHP 8.4 pour le développement backend avec l'extension PDO pour accéder à la base de données de manière sécurisée et permettant de se prémunir des injections SQL grâce à l'utilisation de requêtes préparées

- Base de données MySQL 8.0 pour le stockage des données
- HTML5, CSS et JavaScript pour le développement frontend
- Bootstrap 5.3 pour le design responsive ainsi que Font Awesome 6.4 comme bibliothèque d'icônes
- **Infrastructure**
  - Hébergement sur un serveur PHP local configuré pour le développement
- **API tierces utilisées**
  - Leaflet.js pour la carte interactive avec OpenStreetMap pour le fond de carte
  - Nominatim pour la recherche de lieux

## 3 Conception de la Base de Données

La base de données, nommée `planvoyages`, est le cœur de notre application. Elle est conçue pour stocker de manière structurée toutes les informations relatives aux utilisateurs, à leurs voyages, et aux détails de chaque voyage. Le schéma relationnel est implémenté en MySQL. Le schéma complet est présent en annexe.

Les tables principales sont :

- `utilisateur` : stocke les informations des utilisateurs (identifiants, nom, email, mot de passe haché).
- `voyage` : contient les informations générales sur un voyage (destination, dates, référence à l'utilisateur propriétaire).
- `voyage_partage` : table de liaison pour gérer le partage des voyages entre utilisateurs. Elle lie un voyage à un utilisateur avec qui il est partagé.
- `activite`, `logement`, `transport`, `depense`, `restaurant`, `transport_ville`, `ticket_activite`, `item_checklist_avant_depart` : ces tables stockent les détails spécifiques associés à chaque voyage (activités prévues, réservations de logement, moyens de transport, etc.). Chacune de ces tables est liée à la table `voyage` par une clé étrangère.

Différentes contraines sont définies (clés primaires, clés étrangères, vérifications de dates, ...). Par exemple, la date de début d'un voyage doit être avant sa date de fin.

## 4 Architecture Technique

### 4.1 Structure du code

**Organisation générale** Le code est organisé en s'inspirant du modèle MVC (Modèle-Vue-Contrôleur), que nous avons essayé d'implémenter au mieux. Pour l'implémenter complètement, un framework PHP comme Symfony aurait pu être utilisé. Voici comment nous avons fait :

- **Modèle** : Les interactions avec la base de données sont gérées par un ensemble de fonctions utilitaires dans `utils/database_utils.php`. Ce fichier contient des fonctions pour établir la connexion (`getDbConnection()`) et exécuter des requêtes (`executeQuery()`, `fetchAll()`, `fetchOne()`, `insert()`, `update()`). Les paramètres de connexion à la base de données sont définis dans `config/config.php`.

- **Vue** : Les fichiers PHP contenant principalement du HTML (par exemple, `creation_voyage.php`, `mesvoyages.php`, `voyage_detail.php`) et avec l'aide des fichiers CSS pour le style.
- **Contrôleur** : La logique de traitement est intégrée dans les fichiers PHP principaux (par exemple, `creation_voyage_envoyer.php`, `share_voyage.php`, `pageconnexion.php`), qui gèrent les requêtes utilisateur, traitent les données, interagissent avec le modèle et préparent les réponses pour la vue.

**Gestion des sessions** La gestion des sessions utilisateur est implémentée de manière centralisée. Le fichier `header.php` initialise la session avec `session_start()` (si pas déjà démarrée) et des fonctions utilitaires comme `isUserLoggedIn()` (définie dans `utils/user_utils.php`) sont utilisées pour vérifier l'état de connexion de l'utilisateur. Le fichier `utils/login_check_utils.php` est inclus par `utils/utils.php` et gère la redirection des utilisateurs non connectés pour les pages protégées. En effet, certaines de nos pages (liste, modification de voyage, création, partage) ne sont pas accessibles aux utilisateurs non connectés.

**Système de templates** L'application utilise un système simple de templates en incluant des fichiers PHP dans d'autres fichiers :

- `header.php` : Contient l'en-tête HTML commun à toutes les pages, incluant les balises meta, les liens vers les feuilles de style et la barre de navigation.
- `footer.php` : Contient le pied de page commun et les scripts JavaScript, ainsi que le bouton et le conteneur du chatbot flottant.

Ces fichiers sont inclus dans chaque page de l'application, assurant ainsi une cohérence visuelle et fonctionnelle. Le fichier `utils/utils.php` sert de point d'entrée pour inclure tous les fichiers utilitaires nécessaires (configuration, base de données, utilisateur, général, vérification de connexion).

## 4.2 Organisation des fichiers

L'organisation des fichiers de notre projet suit une structure logique qui facilite la navigation et la maintenance du code.

### Fichiers PHP principaux (à la racine)

- `header.php`, `footer.php` : Composants de l'interface utilisateur.
- `inscription.php`, `pageconnexion.php`, `Réinitialisation.php`, `logout.php` : Gestion de l'authentification.
- `destination.php` : Affiche les destinations populaires et permet la recherche.
- `mesvoyages.php` : Affiche et gère les voyages de l'utilisateur connecté (liste, création, partage).
- `creation_voyage.php` : Formulaire pour créer ou modifier un voyage.
- `creation_voyage_envoyer.php` : Traitement du formulaire de création/modification de voyage.
- `voyage_detail.php` : Affichage détaillé d'un voyage.
- `share_voyage.php` : Traitement du partage d'un voyage.
- `get_users.php` : Récupération de la liste des utilisateurs pour le partage.
- `map.php` : Implémente la carte interactive avec géolocalisation.
- `chat.php` : Implémente l'assistant virtuel de voyage.

## Dossier config

- `config.php` : Contient les paramètres de configuration, notamment pour la base de données et le site. Le détail de la configuration est donnée dans la partie 4.3 de ce rapport

## Dossier utils

- `utils.php` : Fichier central important tous les autres utilitaires.
- `database_utils.php` : Fonctions d’interaction avec la base de données (connexion, exécution de requêtes).
- `user_utils.php` : Fonctions relatives à la gestion des utilisateurs (ex : `isUserLoggedIn()`).
- `general_utils.php` : Fonctions utilitaires générales (redirection, sanitization, alertes, formatage de dates).
- `login_check_utils.php` : Logique de vérification de connexion pour les pages protégées.

## Dossier static (pour les fichiers CSS, images, etc.)

### 4.3 Configuration du projet

**Configuration de la base de données** La configuration de la connexion à la base de données est centralisée dans le fichier `config/config.php` pour n’avoir à mettre ses identifiants de connexion à la base de données une seule fois.

```
1 $config = [
2     'db' => [
3         'host' => '127.0.0.1',
4         'dbname' => 'PlanVoyages',
5         'username' => 'root',
6         'password' => '',
7         'charset' => 'utf8mb4',
8     ],
9     'site' => [
10        'name' => 'TravelDream',
11        'url' => 'http://localhost/projet_voyage',
12        'email' => 'contact@traveldream.com',
13    ],
14];
```

Cette approche permet de modifier facilement les paramètres de connexion sans avoir à modifier chaque fichier qui interagit avec la base de données.

**Fonctions utilitaires** Les fichiers du dossier `utils` contiennent plusieurs fonctions utilitaires importantes :

- `isUserLoggedIn()` (dans `user_utils.php`) : Vérifie si l’utilisateur est connecté en contrôlant la présence de l’ID utilisateur dans la session
- `redirect()` (dans `general_utils.php`) : Facilite la redirection vers une autre page
- `sanitize()` (dans `general_utils.php`) : Nettoie les données d’entrée pour prévenir les attaques XSS

- `generateAlert()` (dans `general_utils.php`) : Génère des messages d'alerte formatés avec Bootstrap
- `formatDate()` (dans `general_utils.php`) : Formate les dates selon le format spécifié
- Les fonctions d'accès à la base de données (`executeQuery()`, `fetchAll()`, etc. dans `database_utils.php`) qui gèrent aussi la connexion à la DB si besoin automatiquement

## 4.4 Sécurité

Plusieurs mesures de sécurité ont été mises en place :

- Requêtes préparées (PDO) pour éviter les injections SQL, utilisées via les fonctions de `utils/database_utils.php`
- Mots de passe hachés avec `password_hash()` lors de l'inscription et vérifiés avec `password_verify()` lors de la connexion. Cela évite les fuites de mots de passe en cas de fuite de données.
- Nettoyage des entrées avec `htmlspecialchars()` via la fonction `sanitize()` (dans `general_utils.php`)
- Gestion des sessions pour contrôler l'accès aux pages protégées (via `utils/login_check_utils.php`).

# 5 Fonctionnalités

## 5.1 Authentification

Le module d'authentification constitue la porte d'entrée de notre application et assure la sécurité des données utilisateurs. Il comprend quatre fonctionnalités principales : l'inscription, la connexion, la réinitialisation du mot de passe et la déconnexion.

### 5.1.1 Inscription

La page d'inscription (`inscription.php`) permet aux nouveaux utilisateurs de créer un compte sur la plateforme.

#### Processus d'inscription

1. L'utilisateur accède au formulaire d'inscription via le lien « Incription » dans la barre de navigation.
2. Il remplit les champs requis : nom d'utilisateur, adresse email et mot de passe.
3. À la soumission, le serveur (dans `inscription.php`) vérifie si l'email est déjà utilisé en interrogeant la table `Utilisateur`.
4. Si l'email est disponible, le mot de passe est haché avec `password_hash()` avant d'être stocké dans la table `Utilisateur`.
5. Un message validant l'inscription s'affiche, invitant l'utilisateur à se connecter.

#### Sécurité

- Utilisation de requêtes préparées (via `fetchValue` et `insert` de `utils/database_utils.php`) pour prévenir les injections SQL.
- Hachage du mot de passe avec `bcrypt` via `password_hash()`.
- Validation des données côté serveur (vérification de l'unicité de l'email).

### 5.1.2 Connexion

La page de connexion (`pageconnexion.php`) permet aux utilisateurs existants d'accéder à leur compte et à leurs voyages.

#### Processus de connexion

1. L'utilisateur accède au formulaire de connexion via le lien « Connexion » dans la barre de navigation.
2. Il saisit son adresse email et son mot de passe.
3. Le serveur (dans `pageconnexion.php`) vérifie l'existence de l'email dans la table `Utilisateur`.
4. Si l'email existe, le mot de passe fourni est vérifié par rapport au mot de passe haché stocké, via `password_verify()`.
5. En cas de succès, les données de l'utilisateur (ID, nom, email) sont stockées en session (avec la variable PHP `$_SESSION`), puis il est redirigé vers sa page de profil (`mesvoyages.php`).
6. En cas d'échec, un message d'erreur est affiché.

**Gestion de session** Après une connexion réussie, les informations suivantes sont stockées en session :

- ID de l'utilisateur (`$_SESSION['id_utilisateur']`)
- Nom d'utilisateur (`$_SESSION['nom_utilisateur']`)
- Adresse email (`$_SESSION['email']`)

Cela permet d'identifier l'utilisateur et de personnaliser son expérience (principalement sur les pages nécessitant une authentification).

### 5.1.3 Réinitialisation du mot de passe

La fonctionnalité de réinitialisation du mot de passe (`Réinitialisation.php`) permet aux utilisateurs ayant oublié leur mot de passe de le réinitialiser de manière sécurisée.

#### Processus de réinitialisation

- L'utilisateur accède au formulaire de réinitialisation via le lien « Mot de passe oublié » sur la page de connexion.
- Il saisit son adresse email, un nouveau mot de passe et la confirmation de ce mot de passe.
- Le serveur vérifie que les deux mots de passe correspondent.
- Si l'email existe dans la base de données (table `Utilisateur`), le nouveau mot de passe est haché avec `password_hash()` et mis à jour.
- Un message de confirmation s'affiche et l'utilisateur est invité à se connecter avec son nouveau mot de passe.

### 5.1.4 Déconnexion

La fonctionnalité de déconnexion (`logout.php`) permet aux utilisateurs de se déconnecter en toute sécurité de leur compte.

## Processus de déconnexion

1. L'utilisateur clique sur le lien « Déconnexion » dans la barre de navigation.
2. Le script `logout.php` appelle `session_start()` puis `session_destroy()`, effaçant toutes les variables de session.
3. L'utilisateur est redirigé vers la page de connexion (`pageconnexion.php`).

## Sécurité

- Destruction complète de la session pour éviter tout accès non autorisé.
- Redirection immédiate pour empêcher toute action après la déconnexion.

```
1 session_start();  
2 session_destroy();  
3 header("Location: pageconnexion.php");  
4 exit();
```

## 5.2 Gestions des Voyages

La gestion des voyages est une fonctionnalité centrale de *TravelDream*. Elle permet aux utilisateurs de créer, modifier, visualiser et partager leurs plans de voyage.

### 5.2.1 Crédation et Modification d'un Voyage

La création et la modification d'un voyage sont gérées via la page `creation_voyage.php` pour l'interface utilisateur et `creation_voyage_envoyer.php` pour le traitement des données.

**Interface de création/modification (`creation_voyage.php`)** L'utilisateur accède à cette page soit via un bouton "Nouveau voyage" sur `mesvoyages.php`, soit en cliquant sur "Modifier" dans les détails d'un voyage existant.

- Si un `id` de voyage est passé en paramètres, le formulaire est pré-rempli avec les informations du voyage existant et de tous ses éléments associés (checklist, transports, logements, activités, restaurants, transports urbains), récupérés depuis la base de données.
- L'utilisateur peut saisir ou modifier le titre du voyage, les dates de début et de fin.
- Des sections dynamiques permettent d'ajouter, modifier ou supprimer des éléments pour :
  - La checklist avant départ (description, état fait/non fait).
  - Les transports (type, villes départ/arrivée, dates, horaires, bagages, terminal).
  - Les hébergements (nom, dates, adresse, horaires check-in/out, numéro de réservation, petit déjeuner).
  - Les transports urbains (type de transport, type de ticket, prix, informations, lieu d'achat).
  - Les activités (nom, date, horaire, adresse, informations, gestion des tickets associés avec nom, prix et lien).
  - Les restaurants (nom, adresse, type, date, horaire).
- Du code JavaScript est utilisé pour gérer l'ajout et la suppression dynamiques de ces éléments dans le formulaire, ainsi que pour permettre de renommer les sections.

## Traitement du formulaire (`creation_voyage_envoyer.php`) Lors de la soumission du formulaire :

1. Le script récupère l'ID de l'utilisateur depuis la session.
2. Si un `voyage_id` est présent (modification), les informations de base du voyage (destination, dates) dans la table `Voyage` sont mises à jour.
3. Si aucun `voyage_id` n'est présent (création), un nouveau voyage est inséré dans la table `Voyage`, et le nouvel ID est récupéré.
4. Pour chaque catégorie d'éléments (checklist, transport, logement, etc.) :
  - Toutes les entrées existantes associées à ce `voyage_id` dans la table correspondante (par exemple, `item_checklist_avant_depart`, `Transport`) sont d'abord supprimées. Cela simplifie la gestion des modifications, ajouts et suppressions d'éléments multiples.
  - Ensuite, les nouvelles données soumises via le formulaire pour cette catégorie sont insérées dans la table. Pour les activités avec ticket, une insertion est également faite dans `ticket_activite`.
5. Après traitement, l'utilisateur est redirigé vers `mesvoyages.php` avec un message de succès.

Toutes les opérations de base de données sont effectuées en utilisant les fonctions utilitaires de `utils/database.php` (comme `insert`, `executeQuery`).

### 5.2.2 Visualisation des voyages de l'utilisateur (`mesvoyages.php`)

La page `mesvoyages.php` est le tableau de bord principal de l'utilisateur connecté. Elle affiche :

- Un résumé du profil utilisateur (nom, email, statistiques sur le nombre de voyages et de destinations).
- Un bouton pour créer un nouveau voyage (redirige vers `creation_voyage.php`).
- Des statistiques globales (nombre total de jours de voyage, prochain voyage prévu avec décompte).
- La liste des voyages de l'utilisateur. Sont affichés les voyages créés par l'utilisateur ainsi que ceux qui ont été partagés avec lui (via la table `voyage_partage`).
- Les voyages sont séparés en deux sections : "Voyages en cours" et "Voyages passés / à venir", déterminés en comparant les dates du voyage avec la date actuelle.
- Pour chaque voyage, sont affichés : la destination, les dates de début et de fin, la durée, et un statut (en cours, à venir, terminé). Une icône indique si le voyage est partagé.
- Des boutons d'action pour chaque voyage :
  - "Modifier" : redirige vers `creation_voyage.php?id={id_voyage}` pour éditer le voyage.
  - "Partager" (uniquement si l'utilisateur est propriétaire du voyage) : ouvre un modal pour gérer le partage.
  - "Détails" : redirige vers `voyage_detail.php?id={id_voyage}` pour une vue détaillée.

### 5.2.3 Visualisation Détailée d'un Voyage (`voyage_detail.php`)

Cette page permet à l'utilisateur de consulter toutes les informations relatives à un voyage spécifique.

- L'accès se fait via le bouton "Détails" sur la page `mesvoyages.php`, en passant l'ID du voyage en paramètre GET (`voyage_detail.php?id={id_voyage}`).
- Le script vérifie que l'utilisateur connecté est soit le propriétaire du voyage, soit un utilisateur avec qui le voyage a été partagé (via la table `voyage_partage`).

- Toutes les informations du voyage sont récupérées depuis la base de données : informations générales (destination, dates), checklist, transports, logements, transports urbains, activités (y compris les détails des tickets), et restaurants.
- Les informations sont présentées section par section. Les dates et heures sont formatées pour une meilleure lisibilité.
- Des boutons d'action sont disponibles : "Modifier" (lien vers `creation_voyage.php`), "Partager" (si l'utilisateur est propriétaire, ouvre le modal de partage), et "Imprimer" (utilise la fonctionnalité d'impression du navigateur).

#### 5.2.4 Partage d'un Voyage

La fonctionnalité de partage permet au propriétaire d'un voyage d'autoriser d'autres utilisateurs de *TravelDream* à consulter les détails de ce voyage.

##### Interface de partage

- Sur la page `mesvoyages.php` (et `voyage_detail.php`), un bouton "Partager" est disponible pour chaque voyage dont l'utilisateur connecté est le propriétaire.
- Cliquer sur ce bouton ouvre une fenêtre modale (Bootstrap Modal) '#shareModal'.
- Au chargement du modal, une requête AJAX est envoyée au script `get_users.php` qui récupère la liste de tous les utilisateurs (sauf l'utilisateur actuel) depuis la table `Utilisateur`, ainsi que la liste des utilisateurs avec qui le voyage spécifique est déjà partagé (depuis la table `voyage_partage`).
- Le modal est ensuite peuplé avec une liste d'utilisateurs sous forme de cases à cocher. Les utilisateurs avec qui le voyage est déjà partagé ont leur case pré-cochée.

##### Traitement du partage (`share_voyage.php`)

- Lorsque l'utilisateur soumet le formulaire du modal de partage, les données (ID du voyage et liste des ID des utilisateurs sélectionnés) sont envoyées en POST à `share_voyage.php`.
- Le script vérifie d'abord que l'utilisateur connecté est bien le propriétaire du voyage.
- Toutes les entrées existantes pour cet `id_voyage` dans `voyage_partage` sont supprimées.
- De nouvelles entrées sont insérées pour chaque utilisateur coché dans le formulaire du modal.
- Une réponse JSON indiquant le succès ou l'échec de l'opération est retournée au client, qui peut alors afficher une notification et fermer le modal.

### 5.3 Carte interactive

#### 5.6.1 Carte interactive

La carte interactive (`map.php`) permet aux utilisateurs de visualiser et de rechercher des lieux sur une carte du monde.

##### Fonctionnalités :

- Affichage d'une carte interactive utilisant Leaflet.js et OpenStreetMap.
- Recherche de lieux via l'API Nominatim.
- Géolocalisation pour afficher la position actuelle de l'utilisateur.
- Ajout de marqueurs sur les lieux recherchés.

##### Extrait de code : Initialisation de la carte

```

1 const map = L.map('map-2').setView([20, 0], 2); % Vue globale
2
3 const streets = L.tileLayer('https://s.tile.openstreetmap.org/{z}/{x}/{y}.png', {
4     attribution: '&copy; OpenStreetMap contributors',
5     maxZoom: 19,
6     crossOrigin: true
7 }).addTo(map);

```

### Extrait de code : Recherche de lieux avec l'API Nominatim :

```

1 function performSearch() {
2     const query = searchInput.value;
3     if (!query) return;
4
5     const originalText = searchBtn.innerHTML;
6     searchBtn.innerHTML = '<i class="fas fa-spinner fa-spin"></i> Recherche ...';
7     searchBtn.disabled = true;
8
9     const url = 'https://nominatim.openstreetmap.org/search?format=json&q='
+ encodeURIComponent(query);
10
11     fetch(url)
12         .then(function(response) { return response.json(); })
13         .then(function(data) {
14             if (data.length > 0) {
15                 const lat = data[0].lat;
16                 const lon = data[0].lon;
17
18                 markerManager.clear();
19                 markerManager.addMarker(lat, lon, '<b>' + data[0].
display_name + '</b>');
20
21                 map.setView([lat, lon], 10);
22             } else {
23                 alert('Lieu introuvable !');
24             }
25             searchBtn.innerHTML = originalText;
26             searchBtn.disabled = false;
27         })
28
29 }

```

## 5.4 Assistant de voyage (chatbot)

L'assistant de voyage (`chat.php`) est un chatbot qui aide les utilisateurs à trouver des destinations adaptées à leurs préférences. Il est accessible via un bouton flottant sur toutes les pages (grâce au fichier `footer.php`).

### Fonctionnalités

- Interface de chat intégrée dans une iframe.
- Base de données de destinations (définie en JavaScript dans `chat.php`) avec leurs caractéristiques (climat, budget, activités, etc.).
- Algorithme de recommandation basé sur les préférences exprimées par l'utilisateur (continent, type de voyage, budget, climat, durée).
- Présentation des destinations recommandées avec leurs informations principales et un lien vers la page `destination.php` pour plus de détails.

**Implémentation technique** L'assistant est implémenté en JavaScript côté client dans le fichier `chat.php`. Il maintient un état de la conversation (`chatbotState`) et pose une série de questions pour affiner les préférences de l'utilisateur.

L'algorithme de recommandation filtre les destinations (stockées dans le tableau JavaScript `destinations`) selon ce que l'utilisateur écrit. Les destinations peuvent être filtrées par continent, budget, type de voyage et durée. Le mot "tous" permet de ne pas faire de choix.

## 5.5 Recherche de destinations (`destination.php`)

Cette fonctionnalité, accessible via la page `destination.php`, permet aux utilisateurs de découvrir des lieux de voyage populaires et d'obtenir des informations détaillées sur ceux-ci.

### Fonctionnalités

- Affichage d'une sélection de destinations populaires (définies dans un tableau PHP `$destinations_info`) avec images et descriptions.
- Barre de recherche pour trouver une destination spécifique parmi celles prédéfinies.
- Si une destination est recherchée ou sélectionnée, affichage détaillé de ses informations (description, meilleure période, budget, activités recommandées).
- Possibilité pour un utilisateur connecté de planifier directement un voyage vers la destination choisie (lien vers `creation_voyage.php` avec la destination pré-remplie).

**Implémentation technique** Les informations sur les destinations populaires sont codées en dur dans un tableau PHP au sein du fichier `destination.php`. La recherche filtre ce tableau.

# 6 Interface Utilisateur

## 6.1 Charte graphique

L'interface de l'application **TravelDream** a été conçue pour offrir une expérience utilisateur agréable. Elle repose sur une charte graphique harmonieuse appliquée à l'ensemble du site.

### 6.1.1 Palette de couleurs

- **Couleur principale** : Jaune-Orange – Représente le soleil et l'énergie.
- **Couleur secondaire** : Bleu ciel – Évoque le ciel, l'océan et la liberté.
- **Couleurs neutres** :
  - Blanc – Fond principal pour une lecture confortable.

- Gris clair – Fond secondaire pour distinguer certaines sections.
- Gris foncé – Texte principal pour un bon contraste.

### 6.1.2 Typographie

- **Titres** : Montserrat – Moderne, sans-serif, excellente lisibilité.
- **Corps de texte** : Open Sans – Clair et fluide pour le contenu principal.

#### Hiérarchie des tailles

- Titres principaux (h1) : 2.5rem
- Sous-titres (h2) : 2rem
- Titres de section (h3) : 1.5rem
- Corps de texte : 1rem
- Texte secondaire : 0.875rem

### 6.1.3 Iconographie

Nous utilisons la bibliothèque **Font Awesome** pour les icônes.

#### Utilisations principales :

- Illustration des fonctionnalités (ex : avion, lit)
- Navigation (flèches, menu hamburger)
- Actions (ajout, modification, suppression)
- Statuts (succès, erreur, information)

### 6.1.4 Éléments d'interface

- **Boutons** : Coins arrondis, couleur d'action ( primaire, secondaire, danger), icône si besoin.
- **Cartes** : Coins arrondis, ombre légère, marges internes.
- **Formulaires** : Champs avec bordures fines, labels clairs, retour visuel de validation.
- **Alertes** : Couleurs distinctes selon le type (succès, erreur, info, avertissement).
- **Barre de recherche** : Coin arrondi, intégration des boutons à l'intérieur, même design sur tout le site
- **Input de texte** : Positionnement du label et taille toujours identique

## 6.2 Maquettes et wireframes

Avant le développement, des maquettes et wireframes ont été réalisés afin de structurer l'interface et garantir une navigation fluide. Les captures d'écran sont disponibles en annexe.

## 6.3 Responsive Design

Notre application a été conçue selon les principes du "mobile-first" et du design responsive, garantissant une expérience utilisateur optimale sur tous les appareils, des smartphones aux grands écrans d'ordinateur.

### 6.3.1 Adaptations spécifiques

Des ajustements ont été appliqués pour améliorer l'expérience utilisateur :

- Menu hamburger sur petits écrans (géré par Bootstrap).
- Cartes de voyage et de destination : s'adaptent en colonnes (par exemple, 1 colonne sur mobile, 2 ou 3 sur écrans plus larges).
- Réorganisation verticale des sections sur mobile.
- Champs de formulaire adaptés en largeur.

## 7 Tests et Validation

### 7.1 Stratégie de test

Pour que la fiabilité de notre application *TravelDream* soit très bonne, nous avons mis en place une stratégie de test complète couvrant différents aspects du système. Cette approche nous a permis d'identifier et de corriger les problèmes avant le déploiement final.

#### 7.1.1 Environnement de test

Les tests ont été réalisés dans un environnement de développement local :

- Serveur local (XAMPP/WAMP/MAMP) avec PHP et MySQL.
- Base de données de test peuplée avec des jeux de données variés et réalistes.
- Navigateurs : Chrome, Firefox, Safari, Edge.
- Outils de développement des navigateurs pour simuler différents appareils et tailles d'écran.

#### 7.1.2 Outils et méthodes

Nous avons utilisé les outils et méthodes suivants pour le processus de validation :

- **Tests manuels** : exécution de scénarios d'utilisation typiques (créer un compte, planifier un voyage complet, partager, consulter,...) à partir d'une checklist.
- Vérification de la conformité du HTML et du CSS générés aux standards du web.
- **Checklists fonctionnelles** : listes de contrôle basée sur les besoins fonctionnels pour s'assurer que chaque fonctionnalité est testée.

## 7.2 Tests fonctionnels

Les tests fonctionnels ont permis de valider que chaque fonctionnalité respecte les besoins utilisateurs et les spécifications définies. Par exemple :

- **Inscription** : création de compte avec email unique, hachage du mot de passe.
- **Connexion** : accès avec identifiants valides, refus avec identifiants invalides, création de session.
- **Création de voyage** : enregistrement correct des informations de base et de tous les éléments associés (checklist, transports, etc.) dans la base de données.
- **Modification de voyage** : mise à jour correcte des informations, y compris la suppression et l'ajout d'éléments dynamiques.
- **Partage de voyage** : un propriétaire peut partager un voyage, les utilisateurs partagés peuvent le voir dans `mesvoyages.php` et `voyage_detail.php`.

- **Chatbot** : capacité à poser des questions et à recevoir des recommandations de destinations pertinentes.
- **Carte interactive** : recherche de lieux et géolocalisation fonctionnelles.

## 8 Difficultés Rencontrées et Solutions

### 8.1 Problèmes techniques

Durant le développement de l'application *TravelDream*, plusieurs défis techniques ont été rencontrés, nécessitant des solutions adaptées.

#### 8.1.1 Compatibilité entre navigateurs

**Problème** : Des différences mineures d'affichage des éléments d'interface (notamment avec les formulaires complexes et les mises en page Flexbox/Grid) ont été observées entre les navigateurs.

**Solution** : L'utilisation de Bootstrap 5 a grandement aidé à minimiser ces problèmes grâce à ses composants et son système de grille normalisés. Des tests réguliers sur Chrome, Firefox, Safari et Edge ont permis d'identifier et de corriger les écarts restants, parfois en utilisant des réinitialisations CSS (`normalize.css` implicitement inclus par Bootstrap) ou des ajustements CSS spécifiques.

#### 8.1.2 Gestion dynamique des formulaires complexes

**Problème** : Le formulaire de création/modification de voyage (`creation_voyage.php`) contient de nombreuses sections dynamiques (checklist, transports, activités, etc.) où l'utilisateur peut ajouter ou supprimer des éléments. La gestion des noms des champs (`name="transport[index][type]"`) et la synchronisation avec le backend pour la sauvegarde étaient complexes.

**Solution** :

- **JavaScript côté client** : Du JavaScript a été utilisé pour cloner les templates HTML des éléments, incrémenter les index dans les noms des champs, et gérer la suppression d'éléments.
- **Traitement côté serveur (`creation_voyage_envoyer.php`)** : Le PHP a été structuré pour boucler sur les tableaux de données reçus (par exemple `$_POST['transport']`). La stratégie "supprimer toutes les anciennes entrées puis insérer les nouvelles" pour chaque sous-section (transports, logements, etc.) a simplifié la logique de mise à jour, évitant de devoir déterminer quels éléments étaient nouveaux, modifiés ou supprimés individuellement.

#### 8.1.3 Intégration des API externes (Leaflet/Nominatim)

**Problème** : Assurer le chargement correct de la carte Leaflet, la gestion des appels à l'API Nominatim pour le géocodage (limites de taux, gestion des erreurs) et l'affichage des résultats de manière conviviale.

**Solution** :

- **Chargement asynchrone/Lazy loading (conceptuel)** : Pour la carte, s'assurer qu'elle ne bloque pas le rendu initial de la page. L'initialisation de la carte dans `map.php` est effectuée après le chargement du DOM.

- **Gestion des erreurs** : Des blocs `try...catch` en JavaScript pour les appels `fetch` à Nominatim et des messages d'alerte pour l'utilisateur en cas d'échec.
- **Performances** : Implémentation de fonctions `debounce` et `throttle` dans `map.php` pour optimiser les appels à l'API de recherche lors de la saisie et pour les animations au défilement et améliorer les performances car nous avions un problème de lenteur sur la page.

## 9 Améliorations Futures

### 9.1 Fonctionnalités à développer

Notre application "TravelDream" offre déjà un ensemble de fonctionnalités pour la planification de voyages, mais plusieurs améliorations pourraient être apportées pour enrichir l'expérience utilisateur et étendre les capacités du système.

#### 9.1.1 Application mobile native

Bien que notre site soit responsive, une application mobile native offrirait une meilleure expérience sur smartphone et permettrait d'accéder à des fonctionnalités spécifiques aux appareils mobiles. **Description** : Développer des applications natives pour iOS et Android qui se synchronisent avec la plateforme web. **Avantages** : - Accès hors ligne aux informations de voyage - Utilisation des fonctionnalités natives (appareil photo, GPS, notifications) - Expérience utilisateur optimisée pour mobile - Notifications push pour les rappels (check-in, activités planifiées, etc.)

#### 9.1.2 Amélioration des performances

- **Optimisation des requêtes SQL** : Analyse et optimisation des requêtes complexes, ajout d'index pertinents sur les tables de la base de données.
- **Lazy loading** : Généraliser le lazy loading pour les images et autres contenus lourds sur toutes les pages.

#### 9.1.3 Renforcement de la sécurité

**Mesures envisagées** :

- Authentification à deux facteurs (2FA).
- Chiffrement plus poussé de certaines données sensibles si nécessaire (au-delà du hachage des mots de passe).
- Protection CSRF (Cross-Site Request Forgery) en utilisant des tokens dans les formulaires.
- Rate limiting sur les tentatives de connexion et autres actions sensibles pour prévenir les attaques par force brute.

#### 9.1.4 Intelligence artificielle et personnalisation avancée

**Objectif** : Offrir une assistance encore plus intelligente et personnalisée.

**Applications possibles** :

- Améliorer le chatbot pour comprendre des requêtes plus complexes.
- Suggestions basées sur les voyages précédents de l'utilisateur ou les tendances populaires.

- Analyse prédictive des meilleures périodes pour voyager ou des fluctuations de prix (nécessiterait des sources de données externes).
- Génération automatique d'itinéraires optimisés en fonction des activités sélectionnées et des contraintes de temps.

En conclusion, ces évolutions permettraient à *TravelDream* de passer d'un outil de planification à une véritable plateforme personnalisée dédiée aux voyageurs.

## 10 Conclusion

### 10.1 Synthèse du projet

Le projet "TravelDream" a permis de développer une application web complète dédiée à la planification de voyages, offrant aux utilisateurs un outil centralisé pour organiser tous les aspects de leurs déplacements. Cette application répond à un besoin réel des voyageurs qui doivent souvent jongler entre différentes plateformes et documents pour gérer leurs réservations, activités et informations pratiques. Notre solution propose une approche intégrée qui couvre l'ensemble du processus de planification :

- Création et gestion de voyages avec leurs titres, dates et destinations.
- Organisation détaillée des transports, hébergements, activités (avec gestion des tickets), restaurants, transports urbains et checklists avant départ.
- Partage des plans de voyage avec d'autres utilisateurs pour consultation.
- Exploration des destinations via une carte interactive.
- Recommandations personnalisées grâce à un assistant virtuel.

L'architecture technique mise en place repose sur des technologies éprouvées (PHP, MySQL, JavaScript, Bootstrap) et utilise une structure organisée en dossiers (`utils`, `config`, `static`) facilitant la maintenance et l'évolution future. La base de données relationnelle a été conçue avec soin pour assurer l'intégrité et la cohérence des données, tout en permettant une gestion efficace des relations entre les différentes entités (utilisateurs, voyages, et tous les éléments d'un voyage). L'interface utilisateur, à la fois esthétique et fonctionnelle, offre une expérience identique sur tous les appareils grâce au responsive design et à l'utilisation de Bootstrap. Les nombreux tests que nous avons effectués nous ont permis d'avoir un site fiable.

### 10.2 Compétences acquises

#### Compétences techniques :

- Développement web full-stack (frontend HTML/CSS/JS, backend PHP).
- Conception et implémentation de bases de données relationnelles (MySQL).
- Introduction à la sécurité web (protection contre injections SQL via PDO, XSS via `htmlspecialchars`, hachage de mots de passe).
- Intégration d'APIs tierces (Leaflet.js, Nominatim).
- Développement de fonctionnalités dynamiques avec JavaScript (gestion de formulaires, AJAX pour le partage).
- Tests manuels et débogage d'applications web.

#### Compétences transversales :

- Gestion de projet (découpage en phases, suivi).

- Travail en équipe (collaboration, communication).
- Analyse des besoins et conception de solutions.
- Résolution de problèmes techniques.
- Rédaction de documentation technique (rapport).
- Conception de l'expérience utilisateur (UX) et de l'interface utilisateur (UI) de base. Formation au responsive design pour avoir une UX et UI excellente sur mobile comme sur PC.

### 10.3 Perspectives

*TravelDream* représente une base solide qui pourrait évoluer dans plusieurs directions. Les améliorations futures identifiées (application mobile, gestion collaborative, gestion des dépenses, IA avancée, etc.) ouvrent la voie à un écosystème complet dédié au voyage. Ce projet pourrait également servir pour explorer des domaines connexes comme :

- L'intégration de services de réservation externes (vols, hôtels, activités).
- L'analyse de données pour identifier des tendances et des préférences de voyage des utilisateurs
- L'utilisation de technologies plus modernes pour le frontend (frameworks JavaScript comme Vue.js ou ReactJS) ou backend (frameworks PHP comme Symfony ou Laravel) pour des projets futurs.

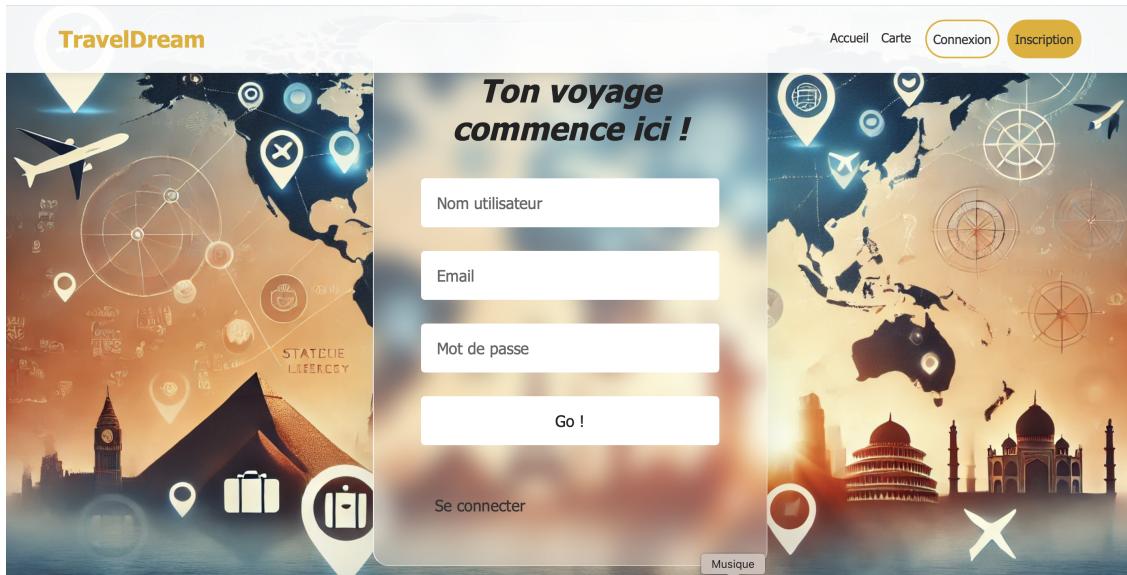
En conclusion, ce projet S4 a été une expérience formatrice et enrichissante, nous permettant d'appliquer concrètement les connaissances acquises durant ce cours et de développer nos compétences. "TravelDream" n'est pas seulement une application fonctionnelle, mais aussi tout un apprentissage autour de la conception, du développement, du déploiement et des tests sur un projet web complet pouvant répondre à des besoins d'utilisateurs.

## **11 ANNEXE : Schéma de bases de données**

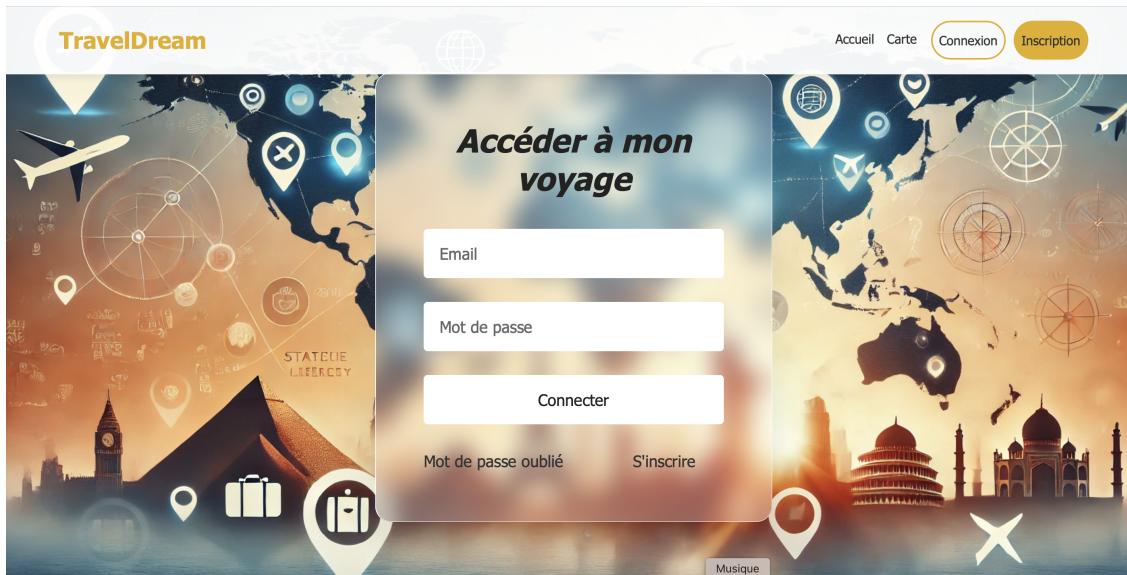
## 12 ANNEXE : Lancement du projet

### 12.1 Authentification

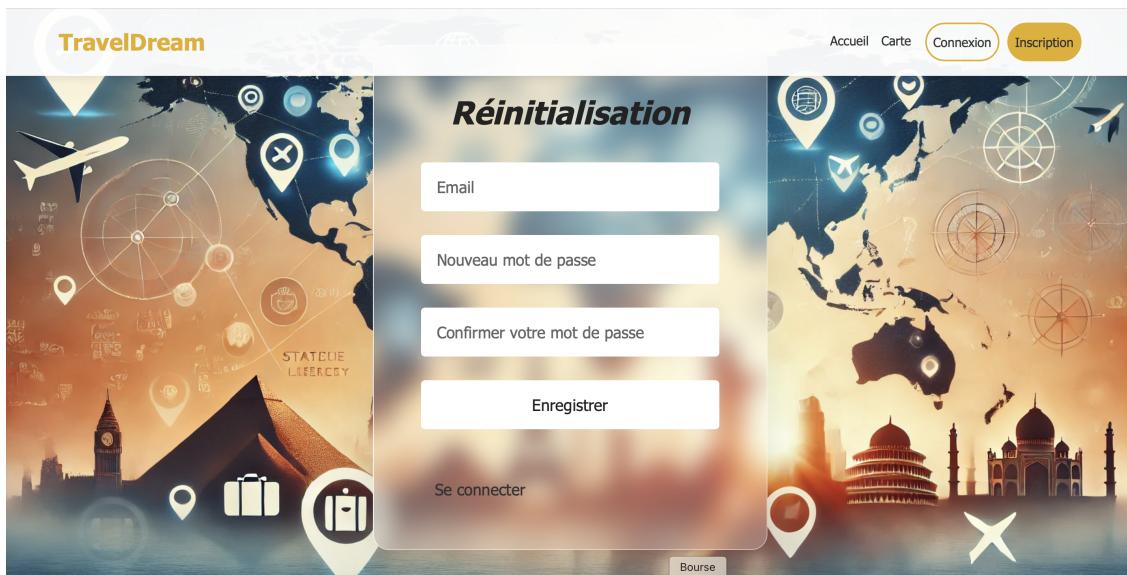
#### 12.1.1 Inscription



#### 12.1.2 Connexion



### 12.1.3 Mot de passe oublié



### 12.1.4 Déconnexion

La déconnexion est possible grâce au bouton déconnexion de la barre de navigation (voir le screenshot ci-dessous).

## 12.2 Accueil et liste des destinations

Accessible en cliquant sur "Accueil" dans la barre de navigation

A screenshot of the TravelDream homepage. At the top is a search bar with 'Rechercher une destination...' and a placeholder 'Paris'. To the right are buttons for 'Accueil', 'Carte', 'Mon Profil', and 'Déconnexion'. The main section is titled 'Destinations populaires' with the sub-instruction 'Explorez nos destinations les plus appréciées par nos voyageurs'. It features six destination cards: Paris (Eiffel Tower), Rome (Colosseum), Bali (temple), New York (Statue of Liberty), Tokyo (Mt. Fuji), and Barcelone (Catalan skyline). Each card includes a small photo, the destination name, a brief description, and seasonal information (e.g., Printemps et automne, Élevé, Moyen).

## Destinations populaires

Explorez nos destinations les plus appréciées par nos voyageurs



### Paris

La ville de l'amour avec ses monuments emblématiques, sa gastronomie et son ambiance romantique.

Printemps et automne

Moyen à élevé



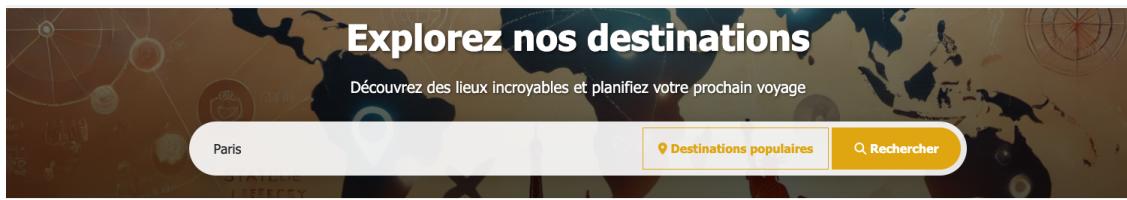
### Rome

La cité éternelle regorge de trésors historiques, d'art et d'une cuisine délicieuse.

Printemps et automne

Moyen





Résultats pour "Paris"

[Voir toutes les destinations](#)**Paris**

La ville de l'amour avec ses monuments emblématiques, sa gastronomie et son ambiance romantique.

[Printemps et automne](#)[Moyen à élevé](#)**À propos de Paris****Description**

La ville de l'amour avec ses monuments emblématiques, sa gastronomie et son ambiance romantique.

**Meilleure période**[Printemps et automne](#)**Budget**[Moyen à élevé](#)**Activités recommandées**[Tour Eiffel](#) [Musée du Louvre](#) [Montmartre](#) [Croisière sur la Seine](#) [Shopping](#)[Planifier un voyage à Paris](#)**TravelDream**

Votre compagnon de voyage idéal pour planifier des aventures inoubliables.

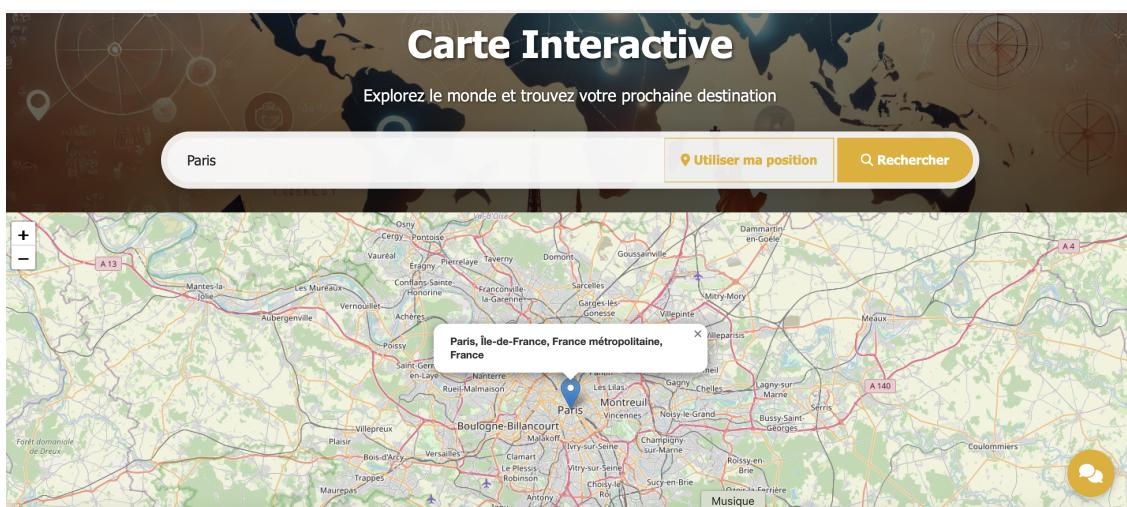
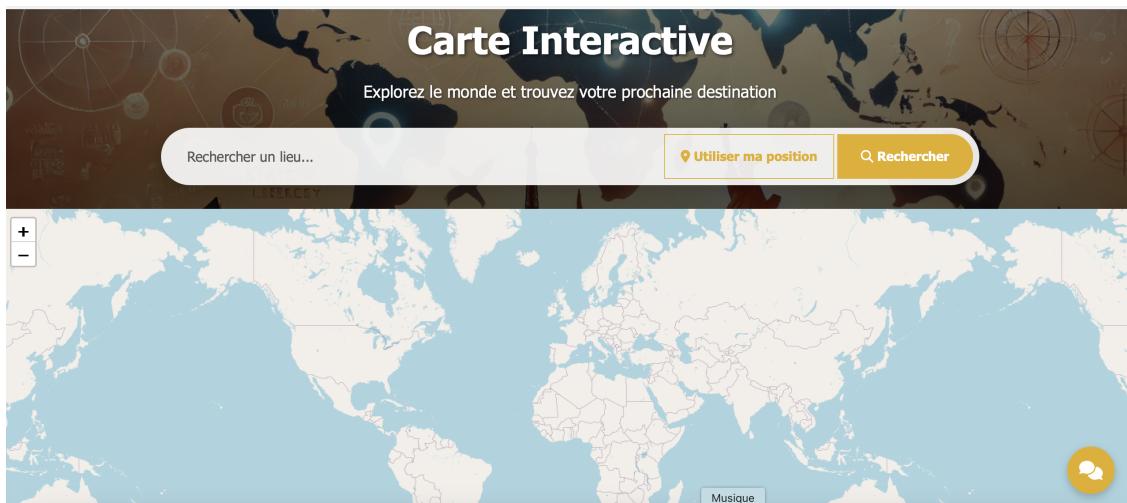
**Assistance**[Centre d'aide](#)[FAQ](#)[Conditions d'utilisation](#)[Politique de confidentialité](#)**Contact**[123 Rue du Voyage, Paris](#)[+33 1 23 45 67 89](#)[contact@traveldream.com](mailto:contact@traveldream.com)

© 2025 TravelDream. Tous droits réservés.



### 12.2.1 Carte des destinations

Accessible en cliquant sur "Carte" dans la barre de navigation



## 12.3 Chatbot

Accessible en cliquant sur la bulle en bas à droite (qui apparaît après la page de connexion)

## Assistant de Voyage

X



TravelDream Assistant  
En ligne

Bonjour ! Je suis l'assistant TravelDream. Je vais vous aider à trouver la destination idéale pour votre prochain voyage. Quel continent souhaitez-vous explorer ? (Europe, Asie, Afrique, Amérique du Nord, Amérique du Sud, Océanie, ou 'tous')

Amérique du Sud



## Assistant de Voyage

X



TravelDream Assistant  
En ligne

? (Europe, Asie, Afrique, Amérique du Nord, Amérique du Sud, Océanie, ou 'tous')

Amérique du Sud

Super ! Quel type de voyage recherchez-vous ? (ville, plage, nature, culture, aventure, détente, gastronomie, romantique)

plage



## Assistant de Voyage

X



TravelDream Assistant  
En ligne

Excellent choix ! Quel est votre budget approximatif ?  
(économique, moyen, élevé)

moyen

Parfait ! Quel climat préférez-vous ?  
(tropical, méditerranéen, tempéré, désertique, montagneux)

Ecrivez votre message...



## Assistant de Voyage

X



TravelDream Assistant  
En ligne

Parfait ! Quel climat préférez-vous ?  
(tropical, méditerranéen, tempéré, désertique, montagneux)

tropical

Dernière question : quelle est la durée idéale de votre voyage ?  
(week-end, une semaine, deux semaines, plus)

Ecrivez votre message...



Assistant de Voyage

TravelDream Assistant  
En ligne

une semaine

Merci pour vos réponses ! Voici les destinations que je vous recommande :

**Rio de Janeiro, Brésil**

Ville vibrante entre mer et montagnes, connue pour ses plages, sa musique et son ambiance festive.

Écrivez votre message...

This screenshot shows a travel assistant chat interface. At the top, it says "Assistant de Voyage" and "TravelDream Assistant En ligne". A yellow box at the top right has an "X" icon. Below that, a button says "une semaine". A message bubble says "Merci pour vos réponses ! Voici les destinations que je vous recommande :". Another message bubble below it says "**Rio de Janeiro, Brésil**" followed by its description. At the bottom, there's a text input field "Écrivez votre message..." and two icons: a yellow circle with a white envelope-like symbol and a yellow circle with a white speech bubble symbol.

Assistant de Voyage

TravelDream Assistant  
En ligne

**Rio de Janeiro, Brésil**

Ville vibrante entre mer et montagnes, connue pour ses plages, sa musique et son ambiance festive.

Budget: moyen

Idéal: 5-7 jours

Meilleure saison: décembre à mars

Activités: Christ Rédempteur

Écrivez votre message...

This screenshot shows a travel assistant chat interface with detailed information about Rio de Janeiro. At the top, it says "Assistant de Voyage" and "TravelDream Assistant En ligne". A yellow box at the top right has an "X" icon. Below that, a message bubble says "**Rio de Janeiro, Brésil**". Its description follows: "Ville vibrante entre mer et montagnes, connue pour ses plages, sa musique et son ambiance festive.". Below this, three callout boxes provide details: "Budget: moyen", "Idéal: 5-7 jours", and "Meilleure saison: décembre à mars". At the bottom, it says "Activités: Christ Rédempteur". There's a text input field "Écrivez votre message..." and a send message icon. A small preview of the chat history is visible at the bottom right.



## 12.4 Gestion des voyages

### 12.4.1 Profil et liste des voyages

Accessible en cliquant sur "Mon profil" dans la barre de navigation

# Statistiques



Jours de voyage

**7 jours**



Prochain voyage

**Rome**

Dans 7 jours

## 12.4.2 Ajouter ou modifier un voyage

Accessible en cliquant sur "Créer un voyage" depuis "Mon profil" et d'autres endroits dans l'application

Accessible en cliquant sur "Modifier" depuis "Mon profil" à partir d'un voyage déjà créé ou "Modifier" à partir des détails du voyage

The screenshot shows the 'TravelDream' app interface for creating a new trip. At the top, there are navigation links: Accueil, Carte, Mon Profil, and Déconnexion. Below this is the title 'Nouveau Voyage'. The main form is titled 'Informations du voyage'. It includes fields for 'Titre du voyage:' (Rome), 'Date de début:' (10/05/2025, selected from a calendar), and 'Date de fin:' (18/05/2025). A note 'Voyageant voyage' is visible near the calendar. There is a button 'Ajouter une tâche' (Add task) and a placeholder 'Pas de tâches' (No tasks) with a task icon. A small message bubble icon is in the bottom right corner.

## Checklist avant voyage

[Ajouter une tâche](#) faire check-in[Supprimer](#) préparer une valise[Supprimer](#)

## Transports

[Ajouter un transport](#)

## Pas de transports

Cliquez sur "Ajouter un transport" pour planifier vos déplacements

## Hébergements

[Ajouter un hébergement](#)

Musique

## Transports

[Ajouter un transport](#)

## Paris-Rome

[Modifier le nom](#)[Supprimer](#)

Type:

Avion

Ville de départ:

Paris

Ville d'arrivée:

Rome

Date:

26/05/2025

Horaire de départ:

12:30

Horaire d'arrivée:

14:30

Bagages:  Oui  Non

## Transports

[Ajouter un transport](#)

## Paris-Rome

[Modifier le nom](#)[Supprimer](#)

Type:

✓ Avion

Train

Bus

Bateau

Voiture

Date:

18/05/2025

Horaire de départ:

12:30

Horaire d'arrivée:

12:30

Bagages:  Oui  Non

Détails des bagages:

Taille, nombre de bagages...



## Hébergements

[Ajouter un hébergement](#)

## Rome-gelato

[Modifier le nom](#)[Supprimer](#)

Date d'arrivée:

26/05/2025

Date de départ:

01/06/2025

Adresse:

9 Pizaa Street

Horaire de check-in:

15:00

Horaire de check-out:

11:00

Numéro de réservation:

1234567890

Petit déjeuner:  Oui  Non[Bourse](#)

## Transports urbain

[Ajouter un transport](#)

## City transport

[Modifier le nom](#)[Supprimer](#)

Type de transport:

Bus, métro, train

Type de ticket:

one week ticket

Prix:

30

Informations:

ticket pour tout ville sauf aéroport

Lieu d'achat:

en ligne

[Musique](#)

## Activités

[Ajouter une activité](#)

## Coloseum

[Modifier le nom](#)[Supprimer](#)

Date:

27/05/2025

Horaire:

10:00

Adresse:

9 Coloseum Street

Informations:

Il faut arriver 30 minutes en avance

Ticket:  Oui  Non

**Restaurants**

**Lucio Pizzeria**

Adresse: 10 Italiano Street

Type: cuisine Italienne

Date: 28/05/2025

Horaire: 18:00

**Ajouter un restaurant** **Modifier le nom** **Supprimer**

**Annuler** **Bourse** **Enregistrer**

### 12.4.3 Détail d'un voyage

Accessible en cliquant sur "Détail" depuis "Mon profil" à partir d'un voyage déjà créé

**Rome**

**Informations générales**

- Dates Du 26/05/2025 au 01/06/2025
- Durée 7 jours

**Actions**

**Modifier** **Partager** **Imprimer**

**Checklist avant départ**

- faire check-in
- préparer une valise

**Transports**

Avion Musique

**TravelDream**

**Transport**

**Avion**

- Lieu de départ Paris
- Date 26/05/2025
- Horaire de départ 12:30
- Horaire d'arrivée 14:30
- Bagages une petite valise

**Hébergements**

**Hôtel**

- Adresse 9 Pizaa Street
- Check-in 15:00
- Check-out 11:00
- Numéro de réservation 1234567890
- Petit déjeuner Non inclus

**Transports urbains**

Type de billet one week ticket  Prix 30.00 €	Lieu d'achat en ligne  Informations ticket pour tout ville sauf aéroport
Type de billet 1 way ticket  Prix 6.00 €	Lieu d'achat en ligne  Informations airport ticket

**Activités**

Activité	Informations
Date 27/05/2025  Horaire Musique	Il faut arriver 30 minutes en avance

**TravelDream**

airport ticket

**Activités**

Activité	Informations
Date 27/05/2025  Horaire 10:00  Adresse 9 Coloseum Street	Il faut arriver 30 minutes en avance

**Restaurants**

Restaurant	Date	Musique
Adresse 10 Italiano Street  Type de cuisine cuisine Italienne	Date 28/05/2025	

## 12.4.4 Partager un voyage

Accessible en cliquant sur "Partager" depuis "Mon profil" à partir d'un voyage déjà créé ou "Partager" à partir des détails du voyage

TravelDream

Nikola

nikola@gmail.com

2 Voyages 2 Destinations

+ Nouveau voyage

Statistiques

Jours de voyage 10 jours

Prochain voyage Rome Dans 7 jours

mes voyages

Aucun voyage en cours actuellement.

Partager le voyage

Sélectionnez les utilisateurs avec qui vous souhaitez partager ce voyage :

- Anna anna.priness@gmail.com
- Elsa elsa.queen@gmail.com
- Melissa 44008939@parislanterre.fr
- Melissa Melissa@gmail.com

Annuler Partager

26/05/2025 01/06/2025

À venir 3 jours

Modifier Détails

À venir 7 jours

Modifier Partager Détails

The screenshot shows the TravelDream mobile application interface. On the left, the user profile for "Melissa" is displayed, including her name, email (Melissa@gmail.com), and statistics for 2 Voyages and 2 Destinations. A yellow button "+ Nouveau voyage" is visible. On the right, the "Mes Voyages" section is shown, featuring a header "Voyages en cours" with a note "Aucun voyage en cours actuellement." Below it is a section for "Voyages passés / à venir" with a single entry for a trip to Berlin from June 9 to June 11, 2025, which is labeled as "À venir" (Upcoming) and "3 jours" (3 days). Buttons for "Modifier", "Partager", and "Détails" are present.

### 12.4.5 Imprimer un voyage

Accessible en cliquant "Imprimer" à partir des détails du voyage

The screenshot shows the TravelDream mobile application displaying a travel itinerary for a trip to Rome. The itinerary includes sections for "Informations générales" (Dates: Du 26/05/2025 au 01/06/2025, Durée: 7 jours), "Checklist avant" (faire-check-in, préparer une valise), "Transports" (Avion), and "Hébergements" (Hotel: Hotel Roma, Transport urbain: Taxi, Activités: Visite de la ville). To the right, a printer settings overlay is open, showing options for "Impriante" (HP OfficeJet Pro 9010 series), "Préréglages" (Copies: 1, Pages: Toutes les pages (3)), "Imprimer en couleur" (Activé), "Taille du papier" (A4 210 par 297 mm), "Orientation" (Portrait), "Mise à l'échelle" (100%), "Safari" (checkboxes for Imprimer l'arrière-plan and Imprimer les en-têtes et pieds de page), and "Support et qualité" (Type de support: Sélection automatique, Qualité: Normale). At the bottom, there are buttons for "?", "PDF", "Musique", and a large blue "Imprimer" button.