Système de Réservation de Tickets de Train du Gabon

Guide de Développement Complet

📋 Table des Matières

🎯 Aperçu du Projet

Le système de réservation de tickets de train du Gabon est une plateforme web responsive conçue pour faciliter la réservation de billets de train en ligne. Développé avec React.js et Tailwind CSS pour le frontend, Node.js avec Express pour le backend, et PostgreSQL hébergé sur Neon pour la base de données.

Objectifs du Projet

- **Plateforme intuitive** : Créer un système de réservation facile à utiliser
- **Mobile First** : Garantir une expérience optimale sur smartphones
- **Architecture modulaire** : Faciliter la maintenance et les améliorations futures
- **Gestion complète** : Système intégré de réservations, paiements et gestion utilisateur

---

🚂 Réseau Ferroviaire Gabonais

Ligne Complète (Toutes les Gares)

La ligne ferroviaire traverse le Gabon selon l'itinéraire suivant :

Owendo → N'toum → Andeme → M'Bel → Oyan → Abanga → **Ndjolé** → Alembé → Otoumbi → Bissouna → Ayem → **Lopé** → Offoué → **Booué** → Ivindo → Mayabi → Milolé → **Lastourville** → Doumé → Lifouta → Mboungou-Mbadouma → **Moanda** → **Franceville**

Types de Trains

🚆 Omnibus

- **Dessert** : Toutes les gares (22 arrêts)
- **Usage** : Transport local et régional
- **Durée** : Plus longue en raison des arrêts fréquents

🚄 TSA Express

- **Dessert** : Uniquement les gares principales (7 arrêts)
- **Gares desservies** : Owendo, Ndjolé, Lopé, Booué, Lastourville, Moanda, Franceville
- **Usage** : Transport rapide inter-villes
- **Durée** : Réduite grâce aux arrêts limités

Horaires de Départ

Les trains partent selon les créneaux suivants :

Jour	Heures de Départ
Mercredi	13h00
Jeudi	13h00
Samedi	13h00 et 20h00

Note importante : Les utilisateurs ne peuvent réserver que pour ces créneaux spécifiques.

⚡ Fonctionnalités Principales

🔐 Authentification et Gestion des Utilisateurs

- Inscription et connexion sécurisées
- Profils utilisateurs avec historique des réservations
- Gestion des informations personnelles

🔍 Recherche et Réservation de Trains

- **Recherche par critères** :
  - Gare de départ et d'arrivée
  - Date et heure (selon les créneaux disponibles)
  - Nombre de passagers
  - Type de train (Omnibus ou TSA Express)
- **Filtres avancés** :

    - Prix
    - Durée du trajet
    - Type de service
- **Sélection interactive des sièges**
- **Processus de réservation guidé**

💳 Gestion des Paiements

- Intégration de systèmes de paiement sécurisés
- Récapitulatif détaillé avant paiement
- Confirmation automatique et génération de facture
- Support de multiples méthodes de paiement

📱 Fonctionnalités Additionnelles

- Affichage des itinéraires populaires du réseau gabonais
- Informations sur les services disponibles dans chaque type de train
- Politique de réservation adaptée aux horaires fixes
- Section FAQ spécifique au transport ferroviaire gabonais
- Support client multilingue

🧬 Architecture Technique

Frontend

- **React.js** : Interface utilisateur moderne et réactive
- **Tailwind CSS** : Design responsive et personnalisé
- **React Router** : Navigation fluide entre les pages
- **React Icons** : Iconographie cohérente

Backend

- **Node.js** : Environnement serveur performant
- **Express** : Framework web robuste
- **JWT** : Authentification sécurisée par tokens
- **API RESTful** : Communication standardisée

Base de Données

- **PostgreSQL** : Gestion robuste des données

- **Hébergement Neon** : Service cloud fiable
- **Modèles spécialisés** : Tables optimisées pour le transport ferroviaire

---

📁 Structure des Dossiers

```
/train-reservation-gabon
├── /client                    # Frontend React
│   ├── /public                # Fichiers statiques
│   ├── /src
│   │   ├── /assets            # Images, fonts, etc.
│   │   ├── /components        # Composants réutilisables
│   │   │   ├── /buttons
│   │   │   ├── /cards
│   │   │   ├── /forms
│   │   │   ├── /modals
│   │   │   └── /ui            # Éléments d'interface génériques
│   │   ├── /features          # Modules par fonctionnalité
│   │   │   ├── /auth          # Authentification
│   │   │   ├── /booking       # Processus de réservation
│   │   │   ├── /payment       # Gestion des paiements
│   │   │   ├── /profile       # Profil utilisateur
│   │   │   ├── /search        # Recherche de trains
│   │   │   └── /stations      # Gestion des gares gabonaises
│   │   ├── /hooks             # Hooks personnalisés
│   │   ├── /layouts           # Composants de mise en page
│   │   ├── /pages             # Composants de page
│   │   ├── /services          # Services API
│   │   ├── /store             # Gestion d'état
│   │   ├── /styles            # Styles globaux
│   │   ├── /utils             # Fonctions utilitaires
│   │   └── /data              # Données statiques (gares, horaires)
│   └── Configuration files
│
├── /server                     # Backend Node.js/Express
│   ├── /config                # Configuration serveur
│   ├── /controllers           # Contrôleurs API
│   ├── /middleware            # Middleware Express
│   ├── /models                # Modèles de données
│   │   ├── Station.js         # Modèle des gares
│   │   ├── Train.js           # Modèle des trains
│   │   ├── Schedule.js        # Modèle des horaires
│   │   └── Booking.js         # Modèle des réservations
│   ├── /routes                # Routes API
│   ├── /services              # Services métier
│   └── /utils                 # Utilitaires
│
├── /database                  # Scripts de base de données
│   ├── /migrations            # Migrations PostgreSQL
│   ├── /seeds                 # Données initiales
│   │   ├── stations.sql       # Données des gares gabonaises
│   │   ├── trains.sql         # Configuration des trains
```

```
│   │   └── schedules.sql    # Horaires fixes
│   └── gabon_railway_schema.sql
│
└── Configuration files
```

📄 Pages et Composants Détaillés

1. 🏠 Page d'Accueil (HomePage)

Composants principaux :

- **Hero Section** : Bannière avec train gabonais et message d'accueil
- **Formulaire de Recherche** :
  - Sélection gare de départ/arrivée (avec autocomplétion)
  - Calendrier avec seulement les jours disponibles (mercredi, jeudi, samedi)
  - Choix de l'heure (13h ou 20h selon le jour)
  - Sélection du type de train (Omnibus/TSA Express)
  - Nombre de passagers
- **Itinéraires Populaires** : Routes fréquemment empruntées
- **Services** : Présentation des avantages de chaque type de train

2. 🔍 Page de Résultats (SearchResultsPage)

Fonctionnalités :

- **Barre de recherche modifiable**
- **Filtres spécialisés** :
  - Type de train (Omnibus/TSA Express)
  - Prix
  - Durée estimée
  - Services à bord
- **Liste des résultats** avec :
  - Gares de départ/arrivée
  - Horaires et durée
  - Prix par catégorie
  - Disponibilité des places
  - Type de train (badge distinctif)

3. 🚂 Page Détails du Train (TrainDetailsPage)

Informations détaillées :

- **Caractéristiques du train** : Type, capacité, services
- **Itinéraire complet** : Liste des arrêts avec horaires
- **Plan des sièges interactif**
- **Services à bord** : Différences entre Omnibus et TSA Express
- **Politique de bagages**
- **Conditions d'annulation**

4. 💳 Page de Paiement (CheckoutPage)

Processus sécurisé :

- **Récapitulatif détaillé** :
  - Informations du trajet
  - Sièges sélectionnés
  - Prix total avec détail
- **Informations passagers** : Formulaires validés
- **Méthodes de paiement** : Options locales et internationales
- **Conditions générales** : Spécifiques au transport ferroviaire

5. 🎫 Page de Facture (InvoicePage)

Éléments essentiels :

- **Billet numérique** avec QR code
- **Détails complets** du voyage
- **Instructions d'embarquement**
- **Contact d'urgence**
- **Options de téléchargement/impression**

📱 Approche Mobile First

Principes de Conception

Design Mobile en Premier :

```css
/* Style de base pour mobile */
.search-form {
  display: flex;
  flex-direction: column;
  gap: 1rem;
  padding: 1rem;
}

/* Adaptation pour tablettes */
@media (min-width: 768px) {
  .search-form {
    flex-direction: row;
    align-items: center;
    gap: 0.5rem;
  }
}

/* Optimisation desktop */
@media (min-width: 1024px) {
  .search-form {
    max-width: 1200px;
    margin: 0 auto;
  }
}
```

Composants Responsifs :

```jsx
import { useMediaQuery } from 'react-responsive';

function StationSelector() {
  const isMobile = useMediaQuery({ maxWidth: 767 });

  return (
    <div className={`station-selector ${isMobile ? 'mobile-layout' : 'desktop-layout'}`}>
      {isMobile ? <MobileStationList /> : <DesktopStationGrid />}
    </div>
  );
}
```

Optimisations Spécifiques

- **Navigation tactile** optimisée pour la sélection des gares

- **Calendrier mobile** adapté aux créneaux fixes
- **Formulaires simplifiés** avec validation en temps réel
- **Chargement progressif** des données de trains
- **Gestes tactiles** pour la sélection des sièges

⚙️ Installation et Configuration

Prérequis Système

- **Node.js** (v18 ou supérieur)
- **npm** ou **yarn**
- **PostgreSQL** (compte Neon recommandé)
- **Git** pour le contrôle de version

Installation Étape par Étape

1. Clonage du Projet

bash

```bash
git clone https://github.com/votre-utilisateur/train-reservation-gabon.git
cd train-reservation-gabon
```

2. Configuration de la Base de Données

Créer le fichier `/server/.env` :

env

```env
# Base de données
DATABASE_URL=postgres://user:password@neon.io:5432/gabon_railway
DB_SSL=true

# Authentification
JWT_SECRET=votre_secret_jwt_securise
JWT_EXPIRES_IN=7d

# Serveur
PORT=5000
NODE_ENV=development

# Paiements (à configurer selon le service choisi)
PAYMENT_API_KEY=your_payment_api_key
```

3. Installation des Dépendances

```bash
# Frontend
cd client
npm install

# Backend
cd ../server
npm install
```

4. Initialisation de la Base de Données

```bash
cd server
npm run db:migrate
npm run db:seed
```

5. Lancement en Développement

```bash
# Terminal 1 - Backend
cd server
npm run dev

# Terminal 2 - Frontend
cd client
npm run dev
```

🤝 Collaboration et Déploiement

Workflow GitHub

Structure des Branches

- **main** : Code de production stable
- **develop** : Branche d'intégration continue
- **feature/nom-fonctionnalité** : Nouvelles fonctionnalités
- **hotfix/correction-urgente** : Corrections critiques

Conventions de Commits

```
feat(stations): ajouter sélecteur de gares gabonaises
fix(booking): corriger validation des horaires
docs(readme): mettre à jour guide d'installation
style(mobile): améliorer responsive sur petits écrans
```

Déploiement Production

Frontend (Vercel/Netlify)

```bash
# Build de production
cd client
npm run build

# Variables d'environnement à configurer :
# REACT_APP_API_URL=https://votre-api.herokuapp.com
# REACT_APP_PAYMENT_KEY=clé_publique_paiement
```

Backend (Railway/Heroku)

```bash
# Préparation
cd server
npm run build

# Variables d'environnement production :
# DATABASE_URL=postgresql://prod-url
# JWT_SECRET=secret-production-fort
# NODE_ENV=production
```

🎯 Spécificités Techniques Gabonaises

Gestion des Gares

javascript

```javascript
// /client/src/data/stations.js
export const GABON_STATIONS = {
  PRINCIPAL_STATIONS: [
    { id: 'OWE', name: 'Owendo', type: 'principal', coordinates: [-0.3, 9.5] },
    { id: 'NDJ', name: 'Ndjolé', type: 'principal', coordinates: [-0.18, 0.18] },
    { id: 'LOP', name: 'Lopé', type: 'principal', coordinates: [0.2, -0.2] },
    { id: 'BOO', name: 'Booué', type: 'principal', coordinates: [0.1, -0.1] },
    { id: 'LAS', name: 'Lastourville', type: 'principal', coordinates: [0.8, -0.8] },
    { id: 'MOA', name: 'Moanda', type: 'principal', coordinates: [1.3, -1.3] },
    { id: 'FRA', name: 'Franceville', type: 'principal', coordinates: [1.6, -1.3] }
  ],
  ALL_STATIONS: [
    // Toutes les 22 gares avec leurs coordonnées
  ]
};
```

Gestion des Horaires

javascript

```javascript
// /server/models/Schedule.js
const FIXED_SCHEDULES = {
  WEDNESDAY: [{ departure: '13:00', arrival_estimates: {...} }],
  THURSDAY: [{ departure: '13:00', arrival_estimates: {...} }],
  SATURDAY: [
    { departure: '13:00', arrival_estimates: {...} },
    { departure: '20:00', arrival_estimates: {...} }
  ]
};
```

📚 Ressources et Documentation

Liens de Référence

- Structure de projet React 2025
- Guide de responsivité React
- Tutoriel vidéo

Technologies Utilisées

- **Frontend** : React 18+, Tailwind CSS 3+, React Router 6+
- **Backend** : Node.js 18+, Express 4+, JWT
- **Base de données** : PostgreSQL 14+, Neon Cloud

- **Outils** : Vite, ESLint, Prettier, Husky

✅ Conclusion

Ce guide complet fournit toutes les informations nécessaires pour développer le système de réservation de tickets de train du Gabon. L'architecture proposée, adaptée aux spécificités du réseau ferroviaire gabonais (gares, horaires fixes, types de trains), garantit une solution robuste et évolutive.

L'approche mobile-first assure une expérience utilisateur optimale sur tous les appareils, particulièrement importante dans le contexte gabonais où l'accès mobile est prédominant.

Prochaines étapes recommandées :

1. Configuration de l'environnement de développement
2. Implémentation des modèles de données spécifiques au Gabon
3. Développement des composants de recherche avec les gares gabonaises
4. Tests sur différents appareils mobiles
5. Intégration des systèmes de paiement locaux

Bonne chance dans le développement de votre plateforme de réservation ferroviaire gabonaise ! 🚂 GA