

=====

DOCUMENTATION DE PASSATION -
TEST PSYCHOTECHNIQUE PERMIS

Date de création: 11 Octobre 2025
Projet: Site de réservation de tests psychotechniques pour permis de conduire
Domaine: <https://test-psychotechnique-permis.com> (<https://test-psychotechnique-permis.com>)

=====

TABLE DES MATIÈRES

1. VUE D'ENSEMBLE DU PROJET
2. ARCHITECTURE TECHNIQUE
3. CONFIGURATION ENVIRONNEMENT
4. BASE DE DONNÉES SUPABASE
5. SYSTÈME D'EMAIL
6. ESPACE ADMINISTRATEUR
7. SYSTÈME DE RÉSERVATION
8. DÉPLOIEMENT
9. SCRIPTS UTILES
10. MAINTENANCE ET DÉPANNAGE
11. CONTACTS ET ACCÈS

=====

1. VUE D'ENSEMBLE DU PROJET

=====

DESCRIPTION: Site web professionnel pour la réservation de tests psychotechniques obligatoires pour la récupération du permis de conduire. Centre agréé situé à Clichy (92).

FONCTIONNALITÉS PRINCIPALES: ☐ Réservation en ligne avec calendrier interactif ☐ Paiement sur place (pas de paiement en ligne) ☐ Envoi automatique d'emails de confirmation ☐ Rappels automatiques 24h avant le rendez-vous ☐ Dashboard administrateur complet ☐ Gestion des créneaux horaires ☐ Système de notifications ☐ Export des statistiques (CSV, PDF) ☐ Nettoyage automatique des anciens rendez-vous

INFORMATIONS DU CENTRE:

- Nom: Test Psychotechnique Permis - Centre Agréé Clichy

• Adresse: 82 Rue Henri Barbusse, 92110 Clichy

• Téléphone: 07 65 56 53 79

• Email contact: contact@test-psychotechnique-permis.com

• Email réservation: reservation@test-psychotechnique-permis.com

• Responsable: Fatiha Sebti (sebtifatih@live.fr)

=====

2. ARCHITECTURE TECHNIQUE

STACK TECHNIQUE:

- Framework: Next.js 14.2.3 (App Router)

• Langage: TypeScript

• Styling: Tailwind CSS + Framer Motion (animations)

• Base de données: Supabase (PostgreSQL)

• Email: Elastic Email API

• Hébergement: OVH (à déployer)

• Gestion de version: Git + GitHub

STRUCTURE DU PROJET:

```

permis-expert/
├─ app/                                # Pages et routes Next.js
│  └─ api/                             # API Routes
│     └─ admin/                        # APIs admin (auth, CRUD)
│        └─ appointments/              # Gestion rendez-vous
│           └─ available-slots/         # Créneaux disponibles
│              └─ cron/                 # Tâches automatisées
│                 └─ email/             # Gestion emails
│                    └─ admin/          # Interface admin
│                       └─ components/  # Composants admin
│                          └─ dashboard/ # Tableau de bord
│                             └─ creneaux/ # Gestion créneaux
│                                └─ notifications/ # Notifications
│                                   └─ settings/ # Paramètres admin
│                                      └─ prendre-rendez-vous/ # Page de réservation
│                                         └─ contact/ # Page contact
│                                            └─ a-propos/ # Page à propos
│                                               └─ [...autres pages]/ # Pages informatives
├─ components/                        # Composants réutilisables
├─ lib/                               # Utilitaires et services
│  └─ supabase.ts                     # Client Supabase
│     └─ emailService.ts              # Service d'envoi email
│        └─ emailTemplates.ts         # Templates HTML emails
│           └─ adminLogger.ts          # Logging admin
│              └─ env.ts               # Validation variables env
├─ public/                            # Assets statiques
├─ styles/                            # Styles globaux
└─ scripts/                           # Scripts Node.js utiles

```

DÉPENDANCES PRINCIPALES:

- @supabase/supabase-js: ^2.39.3 (BDD)
- framer-motion: ^11.0.8 (Animations)
- date-fns: ^3.3.1 (Manipulation dates)
- bcryptjs: ^2.4.3 (Hash mots de passe)
- jsPDF & jsPDF-autotable: Export PDF
- papaparse: Export CSV

3. CONFIGURATION ENVIRONNEMENT

FICHIERS REQUIS:

- .env.local (développement)
- .env.production (production)

VARIABLES D'ENVIRONNEMENT CRITIQUES:

Supabase

NEXT_PUBLIC_SUPABASE_URL=https://hzfpscgdyrqbp1mhgwhi.supabase.co
NEXT_PUBLIC_SUPABASE_ANON_KEY=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJkdXBhYmFzZSIsInJlZiI6Imh6ZnBzY2dkeXJxYnBsbWVnd2hpliwicm9sZSI6ImFub24

Elastic Email API

ELASTIC_EMAIL_API_KEY=B0D3C9F949F85DF5B9045463F6B4A04C1194929A06D05B8B972AAC0B14682CEFB03CA8FA79579D005F264103C6C92987
FROM_EMAIL=contact@test-psychotechnique-permis.com ADMIN_EMAIL=sebtifatiha@live.fr

Application

NEXT_PUBLIC_APP_URL=https://test-psychotechnique-permis.com NODE_ENV=production

⚠ IMPORTANT: Ces clés sont sensibles. Ne JAMAIS les commiter dans Git.

INSTALLATION:

```
# Cloner le repo
git clone https://github.com/adellkl/test-psychotechnique-permis.git
cd permis-expert

# Installer les dépendances
npm install

# Créer .env.local avec les variables ci-dessus
cp .env.example .env.local
# Éditer .env.local avec les vraies valeurs

# Lancer en développement
npm run dev

# Build production
npm run build

# Lancer en production
npm start
```

4. BASE DE DONNÉES SUPABASE

ACCÈS SUPABASE:

- URL Dashboard: <https://supabase.com/dashboard/project/hzfpocgdyrbqplmhgwhi> (<https://supabase.com/dashboard/project/hzfpocgdyrbqplmhgwhi>).
- Email: adellouka2@gmail.com
- Mot de passe: [Demander au propriétaire]

TABLES PRINCIPALES:

1. appointments (Rendez-vous)

- id (uuid, PK)
- first_name, last_name, email, phone
- appointment_date, appointment_time
- status (pending, confirmed, completed, cancelled)
- location, test_type, duration
- created_at, updated_at

2. admins (Administrateurs)

- id (uuid, PK)
- email (unique)
- password_hash (bcrypt)
- full_name, phone
- role (admin)
- is_active (boolean)
- created_at

3. time_slots (Créneaux horaires)

- id (uuid, PK)
- date, start_time, end_time
- location, max_capacity

- is_available (boolean)
- created_at

4. notifications (Notifications admin)

- id (uuid, PK)
- admin_id (FK)
- type (new_appointment, cancellation, reminder)
- message, is_read
- appointment_id (FK)
- created_at

5. admin_activity_logs (Logs d'activité)

- id (uuid, PK)
- admin_id (FK)
- action, details
- ip_address, user_agent
- created_at

6. email_templates (Templates emails)

- id (uuid, PK)
- name (unique)
- subject, html_content
- variables (jsonb)
- is_active, created_at, updated_at

POLITIQUES RLS (Row Level Security):

- Toutes les tables ont des politiques permettant l'accès complet
- Sécurité gérée côté application Next.js
- Pas de récursion dans les politiques

INDEXES POUR PERFORMANCE:

- idx_appointments_status_date (appointments)
- idx_appointments_email (appointments)
- idx_notifications_admin_unread (notifications)
- idx_admin_logs_admin_date (admin_activity_logs)

SCRIPT D'INITIALISATION: Le fichier supabase-tables-setup.sql contient tout le schéma. Exécuter dans l'éditeur SQL de Supabase.



5. SYSTÈME D'EMAIL

SERVICE UTILISÉ: Elastic Email API

- Plan: Payant (20€/mois)
- Limite: Illimitée
- Délivrabilité: Excellente

CONFIGURATION DNS (OVH): Les enregistrements DNS suivants sont configurés pour test-psychotechnique-permis.com:

SPF: v=spf1 include:_spf.elasticemail.com include:mx.ovh.com ~all

DKIM: api_domainkey → Configuration Elastic Email

CNAME bounce: bounce → bounces.elasticemail.net

EMAILS AUTOMATISÉS:

1. Confirmation de rendez-vous (Client)

- Envoyé immédiatement après réservation
- Template: appointment_confirmation_client
- Contient: date, heure, lieu, instructions

2. Notification admin (Nouveau RDV)

- Envoyé à sebtifatiha@live.fr
- Template: appointment_notification_admin
- Contient: infos client, détails RDV

3. Rappel 24h avant

- Envoyé automatiquement via CRON
- Template: appointment_reminder
- Contient: rappel avec détails

4. Annulation

- Envoyé lors d'annulation
- Template: appointment_cancellation
- Contient: confirmation annulation

TEMPLATES: Les templates HTML sont stockés dans:

- Base de données: table email_templates
- Code: lib/emailTemplates.ts

VARIABLES DISPONIBLES: {{first_name}}, {{last_name}}, {{email}}, {{phone}}, {{appointment_date}}, {{appointment_time}}, {{duration}}, {{location}}, {{address}}, {{location_details}}, {{contact_phone}}, {{website}}, {{test_type}}

FICHIER PRINCIPAL: lib/emailService.ts contient toutes les fonctions d'envoi.

TEST D'EMAIL:

```
node test-send-email.js
```

6. ESPACE ADMINISTRATEUR

ACCÈS ADMIN: URL: <https://test-psychotechnique-permis.com/admin> (<https://test-psychotechnique-permis.com/admin>)

COMPTES ADMINISTRATEURS ACTUELS:

1. Compte Principal

- Email: sebtifatiha@live.fr
- Mot de passe: Admin123!
- Rôle: Administrateur principal
- Nom: Fatiha Sebti

2. Compte Développeur

- Email: adelloukal2@gmail.com
- Mot de passe: Adelloulou93350
- Rôle: Développeur
- Nom: Adel Loukal

FONCTIONNALITÉS ADMIN:

1. Dashboard (/admin/dashboard)

- Statistiques en temps réel
- Graphiques (rendez-vous par jour, statuts, lieux)
- Liste des rendez-vous récents
- Notifications non lues
- Export CSV/PDF

2. Gestion des créneaux (/admin/creneaux)

- Création de créneaux horaires
- Modification/suppression
- Gestion de la disponibilité
- Vue calendrier

3. Notifications (/admin/notifications)

- Nouveaux rendez-vous
- Annulations
- Rappels
- Marquage lu/non lu

4. Paramètres (/admin/settings)

- Modification profil admin
- Changement mot de passe
- Envoi d'emails de test
- Logs d'activité

5. Nettoyage BDD

- Suppression rendez-vous anciens
- Filtres par statut et date
- Prévisualisation avant suppression
- Suppression individuelle ou en masse

SÉCURITÉ ADMIN:

- Mots de passe hashés avec bcrypt (10 rounds)
- Vérification mot de passe avant changement
- Logging de toutes les actions
- Session sécurisée
- Pas d'inscription publique (création manuelle)

CRÉATION NOUVEAU ADMIN: Utiliser le script: scripts/create-admin.js

```
node scripts/create-admin.js
```



7. SYSTÈME DE RÉSERVATION

PROCESSUS DE RÉSERVATION:

1. Client visite /prendre-rendez-vous
2. Sélectionne date dans le calendrier
3. Choisit créneau horaire disponible
4. Remplit formulaire (nom, prénom, email, téléphone)
5. Valide la réservation
6. Reçoit email de confirmation immédiatement
7. Admin reçoit notification email

VALIDATION:

- Vérification disponibilité en temps réel
- Validation email format
- Validation téléphone français
- Prévention double réservation

API ENDPOINTS:

GET /api/available-slots

- Paramètres: date, location
- Retourne: créneaux disponibles

POST /api/appointments

- Body: infos client + créneau
- Retourne: confirmation + envoi emails

POST /api/appointments/cancel

- Body: appointment_id, email
- Retourne: confirmation annulation

CALENDRIER:

- Composant: components/Calendar.tsx
- Librairie: date-fns
- Désactive dates passées
- Affiche disponibilités

CRÉNEAUX HORAIRES:

- Configurables par l'admin
- Durée: 30min ou 1h
- Capacité max par créneau
- Plusieurs lieux possibles



8. DÉPLOIEMENT

HÉBERGEMENT: OVH

- Type: Hébergement web mutualisé ou VPS
- Domaine: test-psychotechnique-permis.com
- Certificat SSL: Let's Encrypt (gratuit)

PROCESSUS DE DÉPLOIEMENT:

1. Build local:

```
npm run build
```

2. Vérifier .env.production

3. Upload via FTP/SFTP:

- Dossier .next/
- Dossier public/
- Fichiers package.json, next.config.js
- node_modules/ (ou npm install sur serveur)

4. Configuration serveur:

- Node.js version 18+
- PM2 pour process management
- Nginx comme reverse proxy

5. Démarrage:

```
npm install --production
npm start
```

CONFIGURATION NGINX (exemple):

```
server {
    listen 80;
    server_name test-psychotechnique-permis.com;

    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

TÂCHES CRON: Configurer sur le serveur pour:

1. Rappels 24h avant:

```
0 9 * * * curl https://test-psychotechnique-permis.com/api/cron/send-confirmation-reminders
```

2. Marquer rendez-vous complétés:

```
0 0 * * * curl https://test-psychotechnique-permis.com/api/cron/mark-completed
```

MONITORING:

- Logs serveur: /var/log/nginx/
- Logs application: PM2 logs
- Supabase Dashboard pour BDD
- Elastic Email Dashboard pour emails



9. SCRIPTS UTILES

SCRIPTS NODE.JS DISPONIBLES:

1. check-all-admins.js

- Liste tous les comptes admin
- Affiche emails et statuts

```
node check-all-admins.js
```

2. reset-admin-account.js

- Crée/réinitialise compte admin
- Génère nouveau hash bcrypt

```
node reset-admin-account.js
```

3. get-admin-password.js

- Récupère infos de connexion
- Affiche emails des admins

```
node get-admin-password.js
```

4. test-send-email.js

- Test envoi email Elastic Email
- Vérifie configuration

```
node test-send-email.js
```

5. test-supabase-connection.js

- Test connexion Supabase
- Vérifie tables et données

```
node test-supabase-connection.js
```

COMMANDES NPM:


```
npm run dev      # Développement (port 3000)
npm run build    # Build production
npm start       # Démarrage production
npm run lint     # Vérification code
```

10. MAINTENANCE ET DÉPANNAGE

PROBLÈMES COURANTS:

1. Emails non reçus ✓ Vérifier clé API Elastic Email ✓ Vérifier DNS (SPF, DKIM) ✓ Consulter dashboard Elastic Email ✓ Vérifier logs: lib/emailService.ts
2. Erreur connexion Supabase ✓ Vérifier NEXT_PUBLIC_SUPABASE_URL ✓ Vérifier NEXT_PUBLIC_SUPABASE_ANON_KEY ✓ Tester avec test-supabase-connection.js ✓ Vérifier politiques RLS
3. Admin ne peut pas se connecter ✓ Vérifier email existe: check-all-admins.js ✓ Réinitialiser mot de passe: reset-admin-account.js ✓ Vérifier colonne password_hash en BDD ✓ Logs: admin_activity_logs table
4. Créneaux non disponibles ✓ Vérifier table time_slots ✓ Vérifier is_available = true ✓ Vérifier date >= aujourd'hui ✓ API: /api/available-slots
5. Build échoue ✓ npm install pour dépendances ✓ Vérifier erreurs TypeScript ✓ Supprimer .next/ et rebuild ✓ Vérifier versions Node.js (18+)

LOGS ET DEBUGGING:

Console navigateur:

- Ouvrir DevTools (F12)
- Onglet Console pour erreurs JS
- Onglet Network pour requêtes API

Logs serveur:

```
# PM2
pm2 logs

# Nginx
tail -f /var/log/nginx/error.log
tail -f /var/log/nginx/access.log
```

Logs Supabase:

- Dashboard Supabase > Logs
- Voir requêtes SQL et erreurs

SAUVEGARDE:

Base de données:

- Supabase fait des backups automatiques
- Export manuel: Dashboard > Database > Backup

Code:

- Git: <https://github.com/adellk/test-psychotechnique-permis> (<https://github.com/adellk/test-psychotechnique-permis>)
- Branches: main (production)

MISES À JOUR:

Dépendances:

```
npm outdated    # Voir packages obsolètes
npm update      # Mettre à jour
npm audit fix    # Corriger vulnérabilités
```

Next.js:

```
npm install next@latest react@latest react-dom@latest
```

11. CONTACTS ET ACCÈS

PROPRIÉTAIRE DU PROJET:

- Nom: Fatiha Sebti
- Email: sebtifatiha@live.fr
- Téléphone: 07 65 56 53 79
- Rôle: Gérante du centre

DÉVELOPPEUR ACTUEL:

- Nom: Adel Loukal
- Email: adelloukal2@gmail.com
- GitHub: adellkl

ACCÈS TECHNIQUES:

GitHub:

- Repo: <https://github.com/adellkl/test-psychotechnique-permis> (<https://github.com/adellkl/test-psychotechnique-permis>)
- Accès: Demander invitation au propriétaire

Supabase:

- Dashboard: <https://supabase.com/dashboard/project/hzfpscgdyrgbplmhgwhi> (<https://supabase.com/dashboard/project/hzfpscgdyrgbplmhgwhi>)
- Email: adelloukal2@gmail.com
- Mot de passe: [Demander]

Elastic Email:

- Dashboard: <https://elasticemail.com/account> (<https://elasticemail.com/account>)
- Email: adelloukal2@gmail.com
- Mot de passe: [Demander]
- API Key: Voir section 3

OVH:

- Manager: <https://www.ovh.com/manager/> (<https://www.ovh.com/manager/>)
- Email: [À définir]
- Mot de passe: [À définir]

DOMAINE:

- Registrar: OVH
- Domaine: test-psychotechnique-permis.com
- Expiration: [Vérifier sur OVH]
- DNS: Géré par OVH

EMAILS PROFESSIONNELS (OVH):

- contact@test-psychotechnique-permis.com
- reservation@test-psychotechnique-permis.com
- sebti.fatiha@test-psychotechnique-permis.com

☐ NOTES IMPORTANTES

1. SÉCURITÉ:

- Ne JAMAIS commiter les fichiers .env*
- Changer les mots de passe admin régulièrement

- Vérifier les logs d'activité admin
- Garder les dépendances à jour

2. PERFORMANCE:

- Build optimisé avec Next.js 14
- Images optimisées avec next/image
- Animations GPU avec Framer Motion
- Indexes BDD pour requêtes rapides

3. SEO:

- Metadata optimisés dans layout.tsx
- Sitemap.xml et robots.txt configurés
- Structured data (JSON-LD)
- URLs canoniques

4. ACCESSIBILITÉ:

- Animations respectent prefers-reduced-motion
- Alt text sur toutes les images
- Navigation au clavier
- Contraste WCAG AA

5. RESPONSIVE:

- Mobile-first design
- Breakpoints Tailwind
- Testé sur iOS/Android
- Touch-friendly



☐ SUPPORT

En cas de problème urgent:

1. Consulter cette documentation
2. Vérifier les logs (serveur, Supabase, Elastic Email)
3. Tester avec les scripts de diagnostic
4. Contacter le développeur actuel
5. Contacter le propriétaire du projet

Pour questions techniques:

- GitHub Issues: <https://github.com/adellki/test-psychotechnique-permis/issues> (<https://github.com/adellki/test-psychotechnique-permis/issues>).
- Email développeur: adelloukal2@gmail.com

Pour questions business:

- Email propriétaire: sebtifatiha@live.fr
- Téléphone: 07 65 56 53 79



☐ CHECKLIST DE PASSATION

Avant de passer le projet, vérifier:

- ☐ Accès GitHub fourni au nouveau développeur
- ☐ Accès Supabase transféré
- ☐ Accès Elastic Email transféré
- ☐ Accès OVH transféré
- ☐ Variables d'environnement partagées (sécurisé)
- ☐ Documentation lue et comprise
- ☐ Tests effectués (emails, réservations, admin)
- ☐ Mots de passe admin changés
- ☐ Backup base de données effectué
- ☐ Code à jour sur GitHub
- ☐ Build production testé
- ☐ Tâches CRON configurées
- ☐ Monitoring en place
- ☐ Contact propriétaire établi



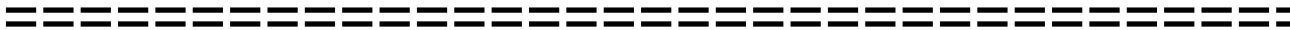
☐ HISTORIQUE DES VERSIONS

v1.0.0 - Octobre 2025

- Lancement initial du site
- Système de réservation complet
- Dashboard administrateur
- Emails automatisés
- Export statistiques

v1.1.0 - Octobre 2025

- Ajout nettoyage BDD
- Amélioration sécurité admin
- Optimisation performances
- Correction bugs emails



FIN DE LA DOCUMENTATION

Cette documentation a été créée le 11 Octobre 2025. Pour toute mise à jour, modifier ce fichier et commiter sur Git.

Bonne chance avec le projet ! ☐