Projet "Trouve ton artisan"

Plateforme de mise en relation entre artisans et particuliers

---

Développeur : Romain Jeanniot
Client : Région Auvergne-Rhône-Alpes
Date : Septembre 2025

SOMMAIRE

---

1. CONTEXTE DU PROJET

1.1 Présentation de l'entreprise

La région Auvergne-Rhône-Alpes couvre 12 départements dans le quart sud-est de la France, avec des bureaux à Lyon et Clermont-Ferrand. Cette région compte près d'un tiers d'entreprises artisanales, soit 221 000 entreprises en 2021, faisant d'elle l'une des régions les plus artisanales de France.

1.2 Expression des besoins

La région souhaite développer une plateforme numérique permettant aux particuliers de :

- Trouver facilement un artisan local
- Consulter les profils détaillés des professionnels
- Contacter directement les artisans via un formulaire sécurisé
- Naviguer par catégorie et spécialité

1.3 Contraintes techniques

- **Accessibilité** : Conformité WCAG 2.1 pour tous les utilisateurs

- **Responsive Design** : Approche mobile-first
- **Sécurité** : Mise en place des bonnes pratiques de sécurité
- **Intégration** : Cohérence avec l'environnement numérique régional
- **Technologies imposées** : ReactJS, Bootstrap, Sass, Node.js, MySQL, Express, Sequelize

1.4 Livrables attendus

- Maquettes Figma pour tous supports (mobile, tablette, desktop)
- Application frontend React
- API Node.js avec base de données MySQL
- Site hébergé et accessible en ligne
- Documentation technique complète

---

2. MAQUETTES FIGMA

2.1 Processus de conception

Les maquettes ont été conçues selon une approche mobile-first, en respectant l'identité graphique de la région Auvergne-Rhône-Alpes :

- **Police** : Graphik (police officielle de la région)
- **Palette de couleurs** : Respectant la charte graphique régionale
- **Logo** : Logo spécifique "Trouve ton artisan"
- **Favicon** : Fourni par la région

2.2 Pages maquettées

Page d'accueil

- Header avec logo, navigation et barre de recherche
- Section "Comment trouver mon artisan ?" (4 étapes)
- Section "Artisans du mois" (3 artisans mis en avant)
- Footer avec informations légales et contact

Page liste des artisans

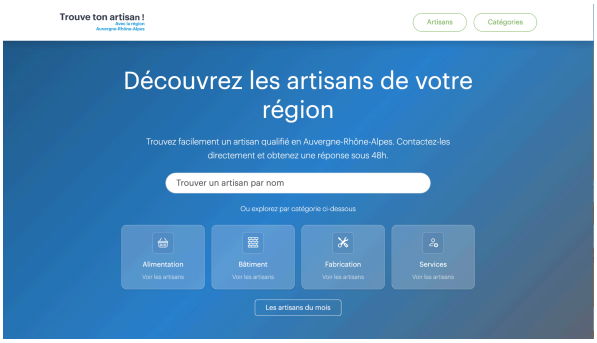
- Filtrage par catégorie
- Cards artisans avec informations essentielles
- Système de pagination
- Navigation intuitive

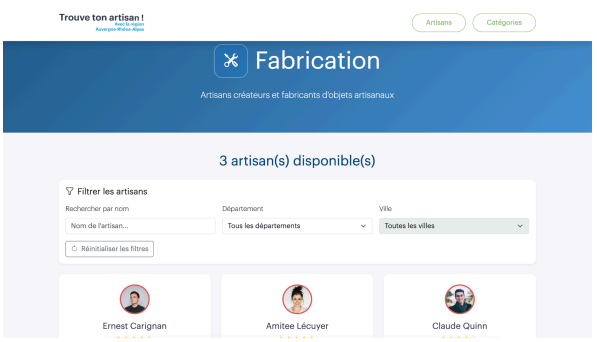
Page fiche artisan

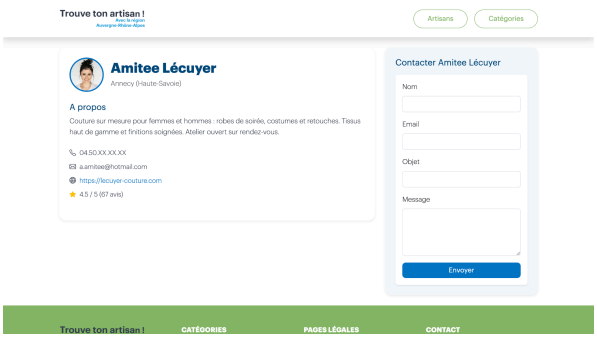
- Informations complètes de l'artisan

- Formulaire de contact intégré
- Design responsive
- Call-to-action clairs

2.3 Captures d'écrans des maquettes





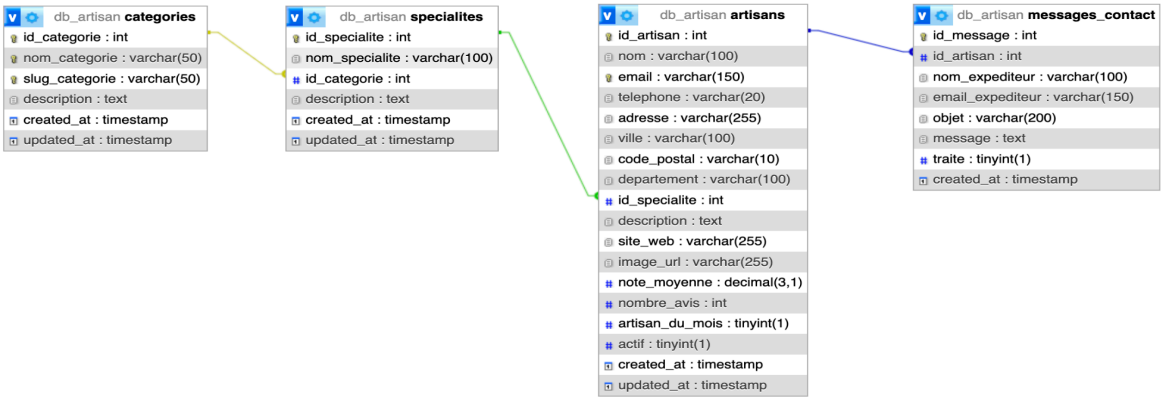


Lien vers les maquettes complètes :
https://github.com/jeanniot-wdv/Trouve-ton-artisan/tree/main/docs

---

3. PRÉSENTATION DE LA BASE DE DONNÉES

3.1 Modèle Conceptuel de données (MCD)



Relations :

- Une CATEGORIE peut avoir plusieurs SPECIALITES (1,N)
- Une SPECIALITE appartient à une seule CATEGORIE (1,1)
- Une SPECIALITE peut avoir plusieurs ARTISANS (1,N)
- Un ARTISAN appartient à une seule SPECIALITE (1,1)

3.2 Modèle Logique de Données (MLD)

CATEGORIE (id_categorie, nom_categorie)
SPECIALITE (id_specialite, nom_specialite, #id_categorie)
ARTISAN (id_artisan, nom, email, telephone, adresse, ville, code_postal, site_web, note, description, image, #id_specialite)

3.3 Implémentation technique

La base de données MySQL a été structurée avec :

- Contraintes d'intégrité : clés primaires et étrangères
- Index : sur les colonnes de recherche fréquente
- Normalisation : 3ème forme normale respectée
- Jeu de données : données de test réalistes pour la région

---

4. ÉLÉMENTS DE SÉCURITÉ MIS EN PLACE

4.1 Protection contre l'injection SQL

Mise en œuvre : Utilisation de Sequelize ORM avec requêtes préparées
Intérêt : prévient l'exécution de code SQL malveillant via les paramètres utilisateur

```
// Exemple d'implémentation sécurisée
const artisans = await Artisan.findAll({
  where: {
    ville: req.params.ville
  }
});
```

4.2 Rate Limiting

Mise en œuvre : middleware express-rate-limit
Configuration : 100 requêtes maximum par fenêtre de 15 minutes
Intérêt : protection contre les attaques par déni de service (DoS) et le spam.

```
const rateLimit = require('express-rate-limit');
const limiter = rateLimit({
  windowMs: 15 * 60 * 1000, // 15 minutes
  max: 100 // limite de 100 requêtes par IP
});
```

4.3 Validation des entrées

Mise en œuvre : express-validator pour tous les formulaires
Intérêt : prévention des attaques XSS et validation des données côté serveur.

```
const { body } = require('express-validator');

const validateContact = [
  body('email').isEmail().normalizeEmail(),
  body('message').trim().isLength({ min: 10, max: 1000 }),
  body('nom').trim().isLength({ min: 2, max: 100 })
];
```

4.4 Configuration CORS

Mise en œuvre : middleware CORS avec origine spécifique
Intérêt : contrôle strict des domaines autorisés à accéder à l'API

```
const cors = require('cors');
app.use(cors({
  origin: process.env.CORS_ORIGIN || 'http://localhost:3000',
  credentials: true
}));
```

4.5 Sécurisation des headers HTTP

Mise en œuvre : helmet.js pour la sécurisation des headers
Intérêt : protection contre diverses attaques (clickjacking, XSS, etc.)

```
const helmet = require('helmet');
app.use(helmet({
  contentSecurityPolicy: {
    directives: {
      defaultSrc: ["'self'"],
      styleSrc: ["'self'", "'unsafe-inline'"]
    }
  }
}));
```

4.6 Gestion des erreurs

Mise en œuvre : middleware centralisé de gestion d'erreurs
Intérêt : évite l'exposition d'informations sensibles en cas d'erreur

```
app.use((err, req, res, next) => {
  if (process.env.NODE_ENV === 'production') {
    res.status(500).json({ message: 'Erreur interne du serveur' });
  } else {
    res.status(500).json({ message: err.message });
  }
});
```

4.7 Variables d'environnement

Mise en œuvre : fichier .env pour les données sensibles
Intérêt : séparation des secrets de configuration du code source

---

5. VEILLE SUR LES VULNÉRABILITÉS DE SÉCURITÉ

5.1 Méthodologie de veille

Durant le développement, une veille continue a été effectuée sur :

- Les vulnérabilités des dépendances npm
- Les failles de sécurité des frameworks utilisés
- Les bonnes pratiques de sécurité web

5.2 Outils utilisés

npm audit : analyse automatique des vulnérabilités des dépendances

```
npm audit
npm audit fix
```

---

6. LIENS ET RESSOURCES

6.1 Repository GitHub

URL : https://github.com/jeanniot-wdv/Trouve-ton-artisan

Contenu du dépôt :

- Code source complet (client + serveur)
- Script de création de la base de données (`database/schema.sql`)
- Script d'alimentation de la base de données (`database/seeds.sql`)
- Documentation technique (`README.md`)
- Configuration de déploiement

6.2 Site en ligne

URL : https://trouve-ton-artisan-jv0v.onrender.com

6.3 Documentation technique

Structure du projet

```
Trouve-ton-artisan/
├── client/                 # Application React (Front-end)
│   ├── src/
│   │   ├── assets/             # Styles SCSS
│   │   ├── components/  # Composants réutilisables
│   │   ├── hooks/              # Hooks personnalisés
│   │   ├── pages/              # Pages principales
│   │   └── services/           # Services API
│   ├── public/
│   └── package.json
├── server/                 # API Node.js/Express (Back-end)
│   ├── config/         # Configuration base de données
│   ├── controllers/            # Contrôleurs API
│   ├── models/                 # Modèles Sequelize
│   ├── repositories/           # Repositories API
│   ├── routes/         # Routes API
│   ├── middleware/             # Middlewares de sécurité
│   └── services/               # Services API
├── database/                   # Scripts SQL
├── docs/               # Documentation
└── README.md
```

Technologies utilisées

- Frontend : ReactJS 18, Bootstrap 5, Sass
- Backend : Node.js 18+, Express 4.19+, Sequelize 6
- Base de données : MySQL 8.0
- Outils : Git, GitHub, Visual Studio Code, Figma

Fonctionnalités implémentées

Côté utilisateur :

- Navigation par catégories d'artisanat
- Recherche par nom d'artisan
- Consultation des profils détaillés
- Formulaire de contact sécurisé
- Design responsive et accessible

Côté technique :

- API RESTful complète
- Gestion des erreurs centralisée
- Validation des données stricte
- Sécurisation des communications
- Optimisation des performances

Respect des contraintes

Accessibilité WCAG 2.1 :

- Contraste de couleurs respecté
- Alternative textuelle pour les images
- Structure sémantique HTML5
- Labels explicites pour les formulaires

Mobile-first :

- Breakpoints responsive optimisés
- Touch-friendly interfaces
- Performances mobiles optimisées
- Images adaptatives

Intégration régionale :

- Charte graphique respectée
- Cohérence avec l'écosystème numérique régional
- Logo et palette de couleurs officiels

PRÉREQUIS D'INSTALLATION

Environnement requis

- Node.js version 18 ou supérieure
- MySQL/MariaDB
- Git

Instructions d'installation

1. Clonage du repository

```
git clone https://github.com/jeanniot-wdv/Trouve-ton-artisan.git
cd Trouve-ton-artisan
```

2. Configuration de la base de données

```
mysql -u root -p -e "CREATE DATABASE db_artisan;"
mysql -u root -p db_artisan < database/schema.sql
mysql -u root -p db_artisan < database/seeds.sql
```

3. Installation du backend

```
cd server
npm install
cp .env.example .env
# Configurer les variables d'environnement dans .env
npm run dev
```

4. Installation du frontend

```
cd ../client
npm install
npm run dev
```

Variables d'environnement

```
# Base de données
DB_HOST=localhost
DB_USER=root
DB_PASSWORD=your_password
```

```
DB_NAME=db_artisan

# Serveur
PORT=3001
NODE_ENV=development

# Sécurité
CORS_ORIGIN=http://localhost:3000
RATE_LIMIT_WINDOW_MS=900000
RATE_LIMIT_MAX_REQUESTS=100
```

VALIDATION W3C

Le code a été validé avec les outils suivants :

- HTML Validator : Aucune erreur détectée
- CSS Validator : Code Sass/CSS conforme
- Accessibility Checker : Conformité WCAG 2.1 AA validée

CONCLUSION

Le projet "Trouve ton artisan" répond parfaitement aux besoins exprimés par la région Auvergne-Rhône-Alpes. La plateforme offre une solution complète, sécurisée et accessible pour mettre en relation artisans et particuliers.

Les choix techniques garantissent :

- Une expérience utilisateur optimale sur tous supports
- Une sécurité robuste contre les principales menaces web
- Une maintenabilité et évolutivité du code
- Une conformité aux standards d'accessibilité

La plateforme est prête pour une mise en production et peut facilement évoluer selon les besoins futurs de la région.

Lien vers le repo GitHub : https://github.com/jeanniot-wdv/Trouve-ton-artisan

Lien vers le site en ligne : https://trouve-ton-artisan-jv0v.onrender.com