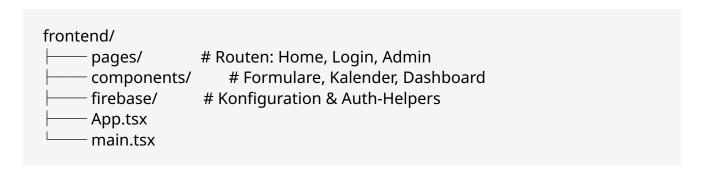
Tattoo-Termin-App

Eine moderne Web-App für selbstständige Tätowierer:innen zur Verwaltung von Kundenanfragen und privaten Terminen.

Tech-Stack

Bereich	Technologie
Frontend	React + Vite + Tailwind CSS
Backend (optional)	Node.js (Express)
Hosting	Railway + GitHub
Auth	Firebase Authentication
Datenbank	Firebase Firestore
Optional	Firebase Functions

Projektstruktur (Frontend)



Firebase Setup

Authentifizierung

- Aktiv: E-Mail + Passwort
- Admin-Login mit fester E-Mail-Adresse

Firestore Collections

appointments

```
{
"type": "beratung" | "stechen" | "nachstechen" | "privat",
"status": "neu" | "angenommen" | "abgelehnt",
"date": "Timestamp",
"customer": {
    "name": "string",
    "email": "string",
    "message": "string (optional)"
}
}
```

privateEvents

```
{
  "title": "string",
  "date": "Timestamp",
  "note": "string (optional)"
}
```

Firestore Regeln

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {

  match /appointments/{id} {
    allow create: if true;
    allow read, update, delete: if request.auth.token.email == "ihre-mail@domain.com";
  }

  match /privateEvents/{id} {
    allow read, write: if request.auth.token.email == "ihre-mail@domain.com";
  }
}
```

Deployment mit Railway

- 1. GitHub-Repo erstellen
- 2. Railway-Projekt → mit GitHub verbinden
- 3. .env -Variablen in Railway setzen: VITE_FIREBASE_API_KEY=... VITE_FIREBASE_AUTH_DOMAIN=... VITE_FIREBASE_PROJECT_ID=...

Funktionen

- Termin-Anfragen durch Kunden (Beratung, Stechen etc.)
- · Admin-Dashboard für Verwaltung
- Kalenderansicht (inkl. privater Termine)
- · Zugriffsschutz über Firebase Auth

Starten (lokal)

npm install npm run dev

ToDos

- [] E-Mail-Benachrichtigungen
- [] Datei-Upload für Motivideen
- $\bullet \ [\] \ Mehrsprachigkeit$
- [] Terminerinnerungen