

MorphCV - AI-Powered CV Generator

MorphCV is a full-stack web application that uses AI to generate professional CVs tailored to job requirements. This repository contains the React frontend that integrates with a Flask backend API.



Features

- **AI-Powered CV Generation:** Create professional CVs optimized for applicant tracking systems
- **Multiple Templates:** Choose from various professional CV templates
- **Real-time Editing:** Edit your CV with live preview
- **Google Authentication:** Secure login with Google OAuth
- **Subscription Management:** Premium features with Stripe integration
- **CV Management:** Save, edit, and download your CVs



Tech Stack

- **Frontend:** React, TypeScript, TailwindCSS
- **State Management:** React Context API
- **API Communication:** Axios
- **Authentication:** JWT with Google OAuth
- **Styling:** TailwindCSS with custom components
- **Payment Processing:** Stripe
- **Build Tool:** Vite

Project Structure

```
morphcv/  
├── public/                # Static assets  
│   ├── images/           # Image assets  
│   │   └── templates/    # CV template preview images  
├── src/  
│   ├── components/       # Reusable UI components  
│   │   ├── MainApp/      # Main application components  
│   │   ├── shared/       # Shared components  
│   │   └── ui/           # UI component library  
│   ├── contexts/         # React contexts  
│   │   └── AuthContext.tsx # Authentication context  
│   ├── hooks/            # Custom React hooks  
│   ├── lib/              # Utility libraries  
│   │   ├── api/          # API service modules  
│   │   └── utils/         # Utility functions  
│   ├── pages/            # Application pages  
│   ├── App.tsx           # Main application component  
│   └── main.tsx          # Application entry point  
├── .env                  # Development environment variables  
├── .env.production       # Production environment variables  
└── vite.config.ts        # Vite configuration
```

API Integration

The frontend integrates with a Flask backend API with the following services:

- **Authentication:** Login with Google OAuth, token refresh, logout
- **CV Management:** Create, read, update, delete CVs
- **User Profile:** User information and statistics
- **Subscription:** Payment processing with Stripe

Setup & Development

Prerequisites

- Node.js 16+ and pnpm
- Flask backend API running
- Google OAuth credentials
- Stripe account (for payment processing)

Environment Variables

Create a `.env` file with the following variables:

```
VITE_API_URL=http://localhost:8000/api/v1
VITE_GOOGLE_CLIENT_ID=your-google-client-id
VITE_STRIPE_PUBLIC_KEY=your-stripe-public-key
```

For production, create a `.env.production` file:

```
VITE_API_URL=/api/v1
VITE_GOOGLE_CLIENT_ID=your-google-client-id
VITE_STRIPE_PUBLIC_KEY=your-stripe-public-key
```

Installation

```
# Install dependencies
pnpm install

# Start development server
pnpm dev

# Build for production
pnpm build
```



Authentication Flow

1. User clicks "Login with Google" button
2. Google OAuth flow is initiated
3. Upon successful Google authentication, the frontend receives an authentication code
4. The code is sent to the backend `/api/v1/auth/google` endpoint
5. Backend verifies the code with Google and issues JWT tokens (access and refresh)
6. Frontend stores tokens securely and includes them in subsequent API requests
7. Tokens are automatically refreshed when needed



CV Generation & Management

1. User fills out CV information form
2. Data is sent to backend for processing
3. Backend generates CV using AI and creates PDF
4. Frontend displays real-time generation status
5. User can view, edit, download, or delete the generated CV

Subscription Management

1. User views available subscription plans
2. Selects a plan and initiates checkout
3. Stripe Checkout session is created by the backend
4. User completes payment on Stripe-hosted page
5. User is redirected back to application with updated subscription status

Responsive Design

The application is fully responsive and works on:

- Desktop
- Tablet
- Mobile devices

Deployment

For detailed deployment instructions, see [DEPLOYMENT.md](#).

Backend API

The frontend integrates with a Flask backend API with the following endpoints:

Authentication

- `POST /api/v1/auth/google` - Google OAuth login
- `POST /api/v1/auth/refresh` - Refresh access token
- `POST /api/v1/auth/logout` - Logout user
- `GET /api/v1/auth/me` - Get current user profile

CV Management

- `GET /api/v1/cvs` - List user CVs with pagination
- `POST /api/v1/cvs` - Generate new CV
- `GET /api/v1/cvs/{uuid}` - Get CV details
- `PUT /api/v1/cvs/{uuid}` - Edit CV
- `DELETE /api/v1/cvs/{uuid}` - Delete CV
- `GET /api/v1/cvs/{uuid}/status` - Get generation status
- `GET /api/v1/cvs/{uuid}/download` - Download CV file

Subscription

- `GET /api/v1/subscription` - Get subscription status
- `POST /api/v1/subscription/checkout` - Create payment session
- `POST /api/v1/subscription/portal` - Customer portal
- `GET /api/v1/subscription/prices` - Available prices

User Management

- `GET /api/v1/users/profile` - Get user profile
- `PUT /api/v1/users/profile` - Update profile
- `GET /api/v1/users/statistics` - User statistics



License

Copyright © 2025 MorphCV. All rights reserved.