

Germansphere SaaS Plattform - Production Deployment Guide

Übersicht

Germansphere ist eine vollständige SaaS-Plattform für deutsches Sprachlernen in Marokko mit umfassenden Features für Kursbuchungen, Schulverwaltung, Tutoren-Services und Visa-Unterstützung.

Features

Kursbuchungssystem

- Erweiterte Kurs-Suche und -Filterung (Standort, Niveau A1-C2, Preisbereich)
- Echtzeit-Verfügbarkeitsprüfung
- Integrierter Buchungsflow mit Zahlungsabwicklung
- Kursvergleichsfunktionalität

Schulverwaltungssystem

- Umfassendes Schulverzeichnis mit detaillierten Profilen
- Erweiterte Suche mit mehreren Filtern
- Schulvergleichsfeatures
- Bewertungs- und Ratingsystem

Visa-Services Plattform

- Dokumentenübersetzungsservices

- Visa-Antragsunterstützung
- Service-Buchung und -Tracking
- Preisrechner

Benutzerverwaltung & Authentifizierung

- Rollenbasierter Zugriff (Student, Tutor, Schuladmin, Systemadmin)
- Sichere JWT-basierte Authentifizierung
- OAuth2.0 Integration (Google, Facebook, GitHub)
- Benutzer-Dashboards für jede Rolle

Zahlungs- & Buchungsinfrastruktur

- Multi-Währungsunterstützung (MAD, EUR, USD)
- Stripe-Integration für internationale Zahlungen
- PayPal-Integration
- CMI/Wafacash für lokale marokkanische Zahlungen
- Buchungsbestätigungen und Rechnungsstellung

Administrative Features

- Admin-Panel für Systemverwaltung
- Analytics und Reporting
- Content-Management
- Benutzerverwaltungstools

Technologie-Stack

Frontend

- **React 18** mit TypeScript

- **Tailwind CSS** für Styling
- **Radix UI** für Komponenten
- **React Router** für Navigation
- **Zustand/Context API** für State Management

Backend

- **Node.js** mit Express und TypeScript
- **PostgreSQL** mit PostGIS für geografische Daten
- **JWT** für Authentifizierung
- **Stripe & PayPal** für Zahlungen
- **bcryptjs** für Passwort-Hashing

Infrastructure

- **Docker** & Docker Compose
- **Nginx** als Reverse Proxy
- **PM2** für Process Management
- **Redis** für Caching (optional)

Deployment-Optionen

1. Quick Deploy (Aktueller Server)

```
# Bereits deployed unter:  
https://40rc5kur8q.space.minimax.io
```

2. Heroku Deployment

```
# Automatisches Deployment  
./deploy.sh heroku
```

3. Vercel Deployment (Frontend)

```
# Frontend-only deployment  
./deploy.sh vercel
```

4. AWS EC2 Deployment

```
# Vollständiges Setup für EC2  
./deploy.sh aws
```

5. Docker Deployment

```
# Lokale Entwicklung  
docker-compose up -d  
  
# Production mit Docker  
docker-compose --profile production up -d
```



Installation & Setup

Voraussetzungen

- Node.js 18+

- pnpm
- PostgreSQL 15+
- Docker (optional)

1. Repository klonen

```
git clone <repository-url>  
cd smirisGermany
```

2. Environment-Variablen konfigurieren

```
# Frontend  
cp .env.example .env  
  
# Backend  
cp server/.env.example server/.env
```

3. Dependencies installieren

```
# Frontend  
pnpm install  
  
# Backend  
cd server && npm install
```

4. Datenbank einrichten

```
cd server  
npm run db:init  
npm run db:seed
```

5. Projekt bauen

```
# Frontend  
pnpm build  
  
# Backend  
cd server && npm run build
```

6. Produktionsserver starten

```
# Frontend (mit serve)  
npx serve dist -s -l 3000  
  
# Backend  
cd server && npm start
```

Sicherheitskonfiguration

Environment-Variablen (Produktion)

```
# Sichere JWT-Secrets (mindestens 64 Zeichen)
JWT_SECRET=your_ultra_secure_jwt_secret_key_here_min_64_chars

# Starke Datenbank-Passwörter
DB_PASSWORD=your_very_secure_database_password

# API-Keys für Zahlungsanbieter
STRIPE_SECRET_KEY=sk_live_your_stripe_secret_key
PAYPAL_CLIENT_SECRET=your_paypal_client_secret

# OAuth2 Credentials
GOOGLE_CLIENT_SECRET=your_google_oauth_client_secret
FACEBOOK_APP_SECRET=your_facebook_app_secret
```

SSL/TLS Konfiguration

```
# Nginx SSL Configuration
server {
    listen 443 ssl http2;
    server_name your-domain.com;

    ssl_certificate /path/to/ssl/cert.pem;
    ssl_certificate_key /path/to/ssl/private.key;

    # SSL Security Headers
    add_header Strict-Transport-Security "max-age=31536000;
includeSubDomains" always;
    add_header X-Content-Type-Options nosniff;
    add_header X-Frame-Options DENY;
    add_header X-XSS-Protection "1; mode=block";
}
```

Datenbank-Schema

Haupttabellen

- **users** - Benutzerverwaltung mit Rollen
- **schools** - Sprachschulverzeichnis
- **courses** - Kursangebote
- **tutors** - Tutorenprofile
- **bookings** - Buchungsverwaltung
- **payments** - Zahlungsabwicklung
- **reviews** - Bewertungssystem
- **visa_services** - Visa-Services

Geografische Features

- PostGIS-Integration für Standortsuche
- Automatische Distanzberechnung
- Kartenintegration

API-Dokumentation

Authentifizierung

```
POST /api/auth/login
POST /api/auth/register
POST /api/auth/refresh
POST /api/auth/oauth/google
POST /api/auth/oauth/facebook
```

Kurse & Buchungen

```
GET /api/courses
POST /api/bookings
GET /api/bookings/:id
PUT /api/bookings/:id
```

Zahlungen

```
POST /api/payments/create-intent
POST /api/payments/confirm
GET /api/payments/:id
POST /api/payments/refund
```

Admin-Funktionen

```
GET /api/admin/analytics
GET /api/admin/users
PUT /api/admin/users/:id
DELETE /api/admin/users/:id
```



Monitoring & Analytics

Health Checks

```
# Application Health
GET /api/health

# Database Health
GET /api/health/database
```

Metriken

- Benutzerregistrierungen
- Buchungsvolumen
- Zahlungserfolgsrate
- Systemperformance

Backup & Recovery

Automatische Backups

```
# Datenbank-Backup
pg_dump germansphere_db > backup_$(date +%Y%m%d).sql

# Datei-Uploads
tar -czf uploads_backup_$(date +%Y%m%d).tar.gz uploads/
```

Disaster Recovery

1. Datenbank aus Backup wiederherstellen
2. Anwendung neu deployen
3. SSL-Zertifikate erneuern
4. DNS aktualisieren

Internationalisierung

Unterstützte Sprachen

- **Deutsch** (Hauptsprache)
- **Französisch** (geplant)
- **Arabisch** (geplant)

Währungsunterstützung

- **MAD** (Marokkanischer Dirham)
- **EUR** (Euro)
- **USD** (US-Dollar)

Support & Wartung

Logs

```
# Anwendungslogs
tail -f logs/application.log

# Nginx-Logs
tail -f /var/log/nginx/access.log

# PM2-Logs
pm2 logs
```

Performance-Optimierung

- CDN für statische Inhalte
- Datenbank-Indizierung
- Code-Splitting
- Lazy Loading
- Caching-Strategien

Roadmap

Phase 1 (Abgeschlossen)

- Basis-Plattform mit allen Kernfunktionen
- Payment-Integration
- OAuth2-Authentifizierung
- Deployment-Konfiguration

Phase 2 (In Planung)

- Mobile App (React Native)
- Advanced Analytics Dashboard
- Mehrsprachige Unterstützung
- Video-Call Integration

Phase 3 (Geplant)

- KI-basierte Kursempfehlungen
- Blockchain-basierte Zertifikate
- Advanced Reporting
- Enterprise Features

Beitrag & Entwicklung

Entwicklungsumgebung

```
# Frontend Development
pnpm dev

# Backend Development
cd server && npm run dev

# Full Stack mit Docker
docker-compose up
```

Code-Qualität

- TypeScript für Typsicherheit
- ESLint für Code-Standards

- Prettier für Formatierung
 - Husky für Git-Hooks
-

Lizenz

MIT License - Siehe LICENSE-Datei für Details.

Kontakt

Für Support und Fragen:

- E-Mail: support@germansphere.com
 - Documentation: [docs.germansphere.com]
 - Status Page: [status.germansphere.com]
-

Germansphere SaaS Platform - Professionelle Deutschlern-Lösung für Marokko 

