**Chakrai - AI-Powered Wellness & Mental Health Platform**

Chakrai is a comprehensive wellness application that combines AI-powered chat, guided meditation, journaling, analytics, and therapeutic tools to support mental health and personal growth.

**Current Version:** 1.0.0 **Last Updated:** September 9, 2025 **Status:** Stable & Production Ready

**Core Features:**

1. **AI Chat Interface**
   * Multi-model AI support (GPT-4, Grok-4)
   * Voice input with speech-to-text
   * Text-to-speech responses with voice selection
   * Real-time typing indicators
   * Message history persistence
2. **Guided Meditation** 
   * 6 meditation types: mindfulness, breathing, guided, visualization
   * ElevenLabs voice synthesis (Natasha, Natasha's Husband)
   * Duration options: 10-25 minutes
   * Progress tracking with visual timer
   * Volume controls and mute functionality
   * Silent meditation fallback
3. **Enhanced Journal** 
   * Rich text journaling with mood tracking
   * Daily prompts and reflection questions
   * Entry categorization and tagging
4. **Analytics Dashboard** 
   * Mental health insights and trends
   * Journal entry analysis
   * Mood pattern recognition
   * Progress visualization with charts
5. **User Management** 
   * Anonymous user sessions with device fingerprinting
   * HIPAA-compliant data handling
   * Privacy controls and data export
   * Secure session management

**Architecture:**

**Frontend (React + TypeScript)**

* Framework: React 18 with TypeScript
* Build Tool: Vite with SWC
* UI Library: Tailwind CSS + Radix UI components
* State Management: React Query + Context providers
* Audio: Native Web Audio API with ElevenLabs integration

**Backend (Node.js + Express)**

* Runtime: Node.js with Express server
* Database: PostgreSQL with Drizzle ORM
* Authentication: HIPAA-compliant session management
* APIs: OpenAI GPT-4, ElevenLabs TTS, Stripe payments
* Security: CORS, rate limiting, helmet middleware

**Infrastructure**

* Development: Concurrently runs client (5173) & server (5001)
* Database: Neon PostgreSQL with connection pooling
* File Storage: Multer for audio uploads
* Environment: dotenv configuration management

**Setup & Development:**

**Prerequisites**

* Node.js 18+
* PostgreSQL database
* OpenAI API key
* ElevenLabs API key (for TTS)

**Environmental Variables**

Create a .env file:

# Database

DATABASE\_URL=postgres://user:pass@host:port/dbname

# APIs

OPENAI\_API\_KEY=sk-...

ELEVENLABS\_API\_KEY=...

# Security

COOKIE\_SECRET=your-secret-key

NODE\_ENV=development

# Server

PORT=5001

**Installation & Startup**

bash

# Install dependencies

npm install

# Start development servers (client + server)

npm run dev

# Client only (port 5173)

npm run dev:client

# Server only (port 5001)  
  
npm run dev:server

# Build for production

npm run build

# Start production server

npm start

**Database Setup**

# Push schema to database

npm run db:push

# Run migrations (if any)

npm run db:migrate

**Application Structure**

chakrai/

|--client/src/ # React Frontend

| |--components/ # UI components

| |--contexts/ # React contexts

| |--utils/ # Utility functions

| |--App.tsx # Main app component

|--server/ # Express backend

| |--routes/ # API endpoints

| |--auth/ # Authentication

| |--middleware/ # Server middleware

| |--index.ts # Server entry point

|--shared/ # Shared utilities

|--types/ # TypeScript definitions

|--vite.config.ts # Build configuration

**Key Components**

**Navigation & Layout**

* ModernLayout: Main shell with sidebar.
* ModernDashboard: Quick actions.
* NeonCursor: Animated cursor.

**Wellness Tools**

* BeautifulMeditation: Guided with TTS.
* PersonalityReflection: Mood tracking.
* VoluntaryQuestionDeck: Self-assessments.
* ChallengeSystem: Goals and challenges.

**Admin & Business**

* TherapistPortal: Provider dashboard.
* AdminPortal: System admin.
* ChakraiPlans: Subscriptions.
* AnalyticsDashboard: Insights.

**Security & Compliance**

**HIPAA Compliance**

* Encrypted transmission (HTTPS).
* Secure sessions.
* Audit logging.
* Consent management.
* Data anonymization.

**Authentication**

* Device fingerprinting.
* Session-based.
* CSRF protection.
* Rate limiting.

**Data Privacy**

* Minimal collection.
* User-controlled retention/deletion.
* Privacy-first principles.

**API Integrations**

* **OpenAI GPT-4:** Chat, speech-to-text, moderation.
* **ElevenLabs TTS:** Voice synthesis, streaming.
* **Stripe:** Subscription billing (ready).

**Performance & Monitoring**

**Client**

* Lazy loading.
* Optimized bundles.
* Efficient state.
* Audio optimization.

**Server**

* Connection pooling.
* Rate limiting.
* Session management.
* Memory optimization.

**Health Monitoring**

* /healthz endpoint.
* Server status.
* Database verification.

**Known Issues & Limitations**

**Minor Issues**

1. Debug logs (non-critical).
2. Enhance error boundaries.
3. Aggressive audio cleanup.

**Planned Improvements**

See Future Roadmap below.

**Development Workflow**

**Code Quality**

# Type checking

npm run check

# Linting

npm run lint

# Formatting

npm run format

# Testing

npm run test

**Database Management**

bash

# Generate migrations

npx drizzle-kit generate

# Push changes

npm run db:push

# Introspect

npx drizzle-kit introspect

**Deployment**

# Build

npm run build

# Start

npm start

* Configure production DB, env vars, HTTPS, domain/SSL.

**Future Roadmap**

**Short Term (1-3 months)**

* Enhanced meditation library
* Mobile responsive improvements
* Advanced analytics features
* Therapist portal enhancements

**Medium Term (3-6 months)**

* Mobile app development
* AI therapy session analysis
* Group meditation features
* Advanced personalization

**Long Term (6+ months)**

* VR therapy integration
* Biometric data integration
* Clinical trial capabilities
* Global wellness community

**Contributing:**

**Guidelines**

1. Follow TypeScript best practices.
2. Use React hooks.
3. Proper error handling.
4. Descriptive commits.
5. Thorough testing.

**Standards**

* ESLint for quality.
* TypeScript for safety.
* Prettier for formatting.
* Conventional commits.

**Support**

* Review console logs.
* Check API keys.
* Verify DB connectivity.
* Ensure env setup.

**License**

MIT License - See LICENSE file for details.