## # Fiber Friends - Morgellons Disease Health Tracking Platform ## Overview

6.44 KB •139 lines•Formatting may be inconsistent from source

# Fiber Friends - Morgellons Disease Health Tracking Platform

## Overview

Fiber Friends is a comprehensive health tracking application specifically designed for individuals with Morgellons Disease. The platform combines symptom logging, AI-powered insights, community support, and gamification elements to help users manage their condition effectively. The application uses a modern full-stack architecture with React/TypeScript frontend and Node.js/Express backend, integrated with Firebase for authentication and data storage.

## System Architecture

### Frontend Architecture

- \*\*Framework\*\*: React 18 with TypeScript and Vite for development

- \*\*UI Components\*\*: Shadcn/UI with Radix UI primitives for accessible components

- \*\*Styling\*\*: Tailwind CSS with custom design system

- \*\*State Management\*\*: TanStack Query (React Query) for server state management

- \*\*Routing\*\*: Wouter for lightweight client-side routing

- \*\*Authentication\*\*: Firebase Auth with Google OAuth integration

### Backend Architecture

- \*\*Runtime\*\*: Node.js with Express.js framework

- \*\*Language\*\*: TypeScript with strict type checking

- \*\*API Design\*\*: RESTful endpoints with Firebase Firestore integration

- \*\*Real-time Communication\*\*: WebSocket server for chat functionality

- \*\*AI Integration\*\*: Multiple AI services including Google Gemini and Anthropic Claude

### Data Storage Strategy

- \*\*Primary Database\*\*: Firebase Firestore (NoSQL document database)

- \*\*Schema Design\*\*: Prepared for PostgreSQL migration with Drizzle ORM schemas

- \*\*File Storage\*\*: Firebase Storage for user uploads and AI-generated content

- \*\*Session Management\*\*: Custom session handling for authentication

## Key Components

### 1. Health Tracking System

- \*\*Daily Symptom Logging\*\*: Comprehensive symptom tracking with severity scales

- \*\*Symptom Wheel Interface\*\*: Interactive circular interface for multi-dimensional symptom tracking

- \*\*Food Logging\*\*: Nutritional analysis with Morgellons-specific dietary recommendations

- \*\*Pattern Recognition\*\*: Advanced analytics to identify symptom correlations and triggers

### 2. AI Companion System (Luna)

- \*\*Personalized AI Assistant\*\*: Customizable AI companion with specialized Morgellons knowledge

- \*\*Multi-modal Interaction\*\*: Text and voice-based communication

- \*\*Contextual Responses\*\*: Health-specific advice based on user's condition and history

- \*\*Personality Customization\*\*: User-configurable AI personality and appearance

### 3. Community Platform

- \*\*Discussion Forums\*\*: Category-based community discussions

- \*\*Real-time Chat\*\*: WebSocket-powered instant messaging

- \*\*Peer Support\*\*: User-to-user support and experience sharing

- \*\*Content Moderation\*\*: Built-in safety and moderation features

### 4. Gamification Engine

- \*\*Points System\*\*: Comprehensive scoring for health activities

- \*\*Challenges\*\*: Daily, weekly, and personalized health challenges

- \*\*Achievements\*\*: Milestone-based achievement system

- \*\*Leaderboards\*\*: Community engagement through friendly competition

### 5. Analytics and Insights

- \*\*Pattern Detection\*\*: Machine learning algorithms for symptom pattern recognition

- \*\*Environmental Correlation\*\*: Weather and environmental trigger analysis

- \*\*Personalized Recommendations\*\*: AI-driven health activity suggestions

- \*\*Research Contribution\*\*: Anonymized data contribution for Morgellons research

## Data Flow

### Authentication Flow

1. User signs in via Google OAuth through Firebase Auth

2. User profile created/updated in Firestore

3. Session established with custom middleware

4. JWT tokens managed by Firebase SDK

### Health Data Flow

1. User inputs symptom/health data through various interfaces

2. Data validated and stored in Firestore collections

3. Background analytics processing identifies patterns

4. AI services generate personalized insights and recommendations

5. Results displayed in dashboard and pattern analysis views

### AI Interaction Flow

1. User sends message to AI companion

2. Context gathered from user's health history and preferences

3. AI service processes request with Morgellons-specific knowledge

4. Response generated and stored in conversation history

5. Points awarded for engagement

## External Dependencies

### AI Services

- \*\*Google Gemini\*\*: Primary AI service for health insights and companion responses

- \*\*Anthropic Claude\*\*: Secondary AI service for specialized analysis

- \*\*Firebase Genkit\*\*: AI orchestration and flow management

### Third-party APIs

- \*\*OpenWeather API\*\*: Environmental data for trigger analysis

- \*\*Stripe\*\*: Payment processing for premium features

- \*\*Firebase Services\*\*: Authentication, Firestore, Storage, Hosting

### UI Libraries

- \*\*Radix UI\*\*: Accessible component primitives

- \*\*Tailwind CSS\*\*: Utility-first styling framework

- \*\*Lucide React\*\*: Icon library

- \*\*React Hook Form\*\*: Form management with validation

## Deployment Strategy

### Development Environment

- \*\*Local Development\*\*: Vite dev server with hot module replacement

- \*\*Database\*\*: Firebase Firestore with development project

- \*\*Authentication\*\*: Firebase Auth with test users

- \*\*WebSocket\*\*: Local WebSocket server for chat testing

### Production Deployment

- \*\*Platform\*\*: Firebase App Hosting with automatic scaling

- \*\*Build Process\*\*: Vite production build with TypeScript compilation

- \*\*Environment Variables\*\*: Firebase environment configuration

- \*\*CDN\*\*: Firebase Hosting with global edge caching

- \*\*Monitoring\*\*: Firebase Analytics and Crashlytics

### Database Migration Strategy

- \*\*Current\*\*: Firebase Firestore for rapid development

- \*\*Future\*\*: PostgreSQL with Drizzle ORM (schemas already defined)

- \*\*Migration Path\*\*: Data export/import scripts with schema transformation

- \*\*Backward Compatibility\*\*: Dual-write pattern during transition

## Changelog

- June 16, 2025. Initial setup

- June 16, 2025. Migrated from Replit Agent to standard Replit environment for Firebase deployment

- Removed Replit-specific dependencies (@replit/vite-plugin-cartographer, @replit/vite-plugin-runtime-error-modal)

- Configured Firebase integration with user's project credentials (fiber-friends)

- Fixed vite configuration and server startup issues

- Added Firebase configuration files for App Hosting deployment

- All data operations configured for Firebase Firestore and Storage

## User Preferences

Preferred communication style: Simple, everyday language.

\*\*\*Be Advised\*\*\*

A lot of changes have been made since the creation of this document. Please check all files very carefully before making assumptions or suggested code or structure changes.