Progress Journal - TODO Liste & Roadmap

🚨 Phase 1: MVP-Fertigstellung (DRINGEND)

Backend-Fixes

- [ ] **ImportError im Backend beheben**
- [ ] `api/serializers.py` und `api/views.py` überprüfen
- [ ] Project-Modell und Serializer-Klasse korrigieren
- [ ] Backend-Server zum Laufen bringen
- [ ] API-Endpunkte testen

KI-Assistent Implementation

- [ ] **Backend-Logik für Roadmap-Generierung**
- [ ] API-Endpunkt `/api/generate-roadmap/` erstellen
- [ ] Input-Verarbeitung (Ziel, Ausgangslage, Deadline)
- [ ] Mock-KI-Logik für Meilenstein-Generierung implementieren
- [ ] JSON-Response für generierte Pläne
- [ ] **Frontend-Integration**
- [ ] Wizard-Form mit Backend verbinden
- [ ] API-Anfrage in `dashboard-empty.html` integrieren
- [ ] Loading-States und Error-Handling hinzufügen
- [ ] Generierte Roadmap im UI anzeigen

Dashboard-Funktionalität

- [ ] **Projekt-Verwaltung**
- [ ] CRUD-Operationen für Projekte implementieren
- [ ] Projektkarten mit echten Backend-Daten füllen
- [ ] Progress-Berechnung implementieren
- [ ] Projekt-Detail-Ansicht funktionsfähig machen
- [ ] **Task-Management**
- [ ] CRUD für Tasks innerhalb von Milestones
- [ ] Task-Checkbox-Funktionalität
- [ ] Task-Status-Updates an Backend senden
- [ ] Progress-Updates basierend auf Task-Completion

Authentifizierung

- [ ] **Basis-Login-System**
- [ ] Django TokenAuthentication einrichten
- [ ] Login/Logout-Funktionalität
- [ ] User-spezifische Projekte
- [ ] Permissions von AllowAny auf IsAuthenticated ändern

📈 Phase 2: Feature-Erweiterung (nach MVP-Launch)

Reflexion & Inbox

- [ ] **Inbox-Funktionalität**
- [ ] InboxItem CRUD-Operationen
- [ ] Notizen zu Projekten zuordnen
- [ ] Quick-Add-Button implementieren
- [ ] Drag & Drop für Inbox-Items

Visuelle Daten & Timeline

- [ ] **Progress-Visualisierung**
- [ ] Timeline-Ansicht mit echten Daten
- [ ] Progress-Charts und Statistiken
- [ ] Meilenstein-Timeline in Projekt-Details
- [ ] Erfolgs-Tracking und Visualisierung
- [ ] **Pomodoro-Timer Integration**
- [ ] Timer mit Task-Verknüpfung
- [ ] Pomodoro-Sessions speichern
- [ ] Einstellungen für Timer-Dauer
- [ ] Timer-Statistiken

Streak-Tracking

- [ ] **Gewohnheits-Tracking**
- [ ] Tägliche/wöchentliche Streaks

- [ ] Streak-Visualisierung
- [ ] Motivation durch Streak-Belohnungen
- [ ] Kalender-Ansicht für Streaks

🚀 Phase 3: Strategische Features (V2.0+)

Automatisierung & Integrationen

- [ ] **API-Integrationen**
- [ ] Strava-Integration für Lauf-Projekte
- [ ] Lichess-Integration für Schach-Projekte
- [ ] Kalender-Synchronisation
- [ ] Automatisches Progress-Tracking

Community Features

- [ ] **Sharing & Kollaboration**
- [ ] "Dein Jahr im Journal" Feature (Spotify Wrapped-Style)
- [ ] Teilbare Jahresrückblicke
- [ ] Team-Roadmaps für gemeinsame Projekte
- [ ] Public Progress-Sharing

Feedback & Verbesserung

- [ ] **Community-Hub**
- [ ] Integrierter Feedback-Hub
- [ ] Feature-Voting-System
- [ ] User-Analytics für App-Verbesserung
- [ ] A/B-Testing für neue Features

🔧 Technische Verbesserungen

Frontend-Optimierungen

- [ ] **Performance**
- [ ] React-Komponenten optimieren
- [ ] Lazy Loading für große Listen
- [ ] PWA-Features hinzufügen
- [ ] Mobile-First Responsive Design verfeinern
- [ ] **UX/UI-Verbesserungen**
- [ ] Animations und Micro-Interactions
- [ ] Dark Mode implementieren
- [ ] Accessibility-Features
- [ ] Keyboard-Navigation

Backend-Skalierung

- [ ] **Database & Performance**
- [ ] PostgreSQL für Production
- [ ] Database-Indizierung optimieren
- [ ] API-Rate-Limiting
- [ ] Caching-Strategy implementieren
- [ ] **Deployment & DevOps**
- [ ] Docker-Setup für Development/Production
- [ ] CI/CD-Pipeline einrichten
- [ ] Environment-Configuration
- [ ] Monitoring und Logging

📝 Content & Marketing

Onboarding & Tutorial

- [ ] **User-Experience**
- [ ] Onboarding-Flow verfeinern
- [ ] Interactive Tutorial erstellen
- [ ] Help-System und FAQ
- [ ] Video-Tutorials erstellen

Launch-Vorbereitung

- [ ] **Marketing-Material**
- [ ] Landing Page erstellen
- [ ] Demo-Videos produzieren
- [ ] Beta-Tester-Programm
- [ ] Social Media Content

🎯 Nächste Schritte (Diese Woche)

Priorität 1 (Sofort)

1. **Backend-ImportError beheben** - Ohne funktionierendes Backend ist nichts möglich
2. **API-Endpunkt für Roadmap-Generierung** - Kernfunktionalität der App
3. **Frontend-Backend-Verbindung testen** - Sicherstellen dass Wizard funktioniert

Priorität 2 (Diese Woche)

4. **Projekt-CRUD vollständig implementieren** - Basis für alle anderen Features
5. **Authentifizierung einrichten** - Notwendig für User-spezifische Daten
6. **Task-Management funktionsfähig machen** - Für funktionales MVP

Messbare Ziele für MVP

- [ ] Wizard erstellt erfolgreich Projekte mit Milestones
- [ ] Dashboard zeigt echte Projekte mit korrektem Progress
- [ ] Tasks können als erledigt markiert werden
- [ ] Progress wird automatisch berechnet und angezeigt
- [ ] Einfaches Login-System funktioniert

💡 Implementierungs-Tipps

Für Backend-Fixes

- Schauen Sie zuerst in die Django-Logs für spezifische Error-Messages
- Überprüfen Sie Model-Relationships in `models.py`
- Testen Sie API-Endpunkte mit Tools wie Postman oder curl

Für Frontend-Integration

- Implementieren Sie erst Mock-Responses, dann echte API-Calls
- Verwenden Sie Try-Catch für Error-Handling
- Implementieren Sie Loading-States für bessere UX

Für Deployment

- Starten Sie mit SQLite für Development
- Verwenden Sie Environment-Variablen für Konfiguration
- Implementieren Sie Logging von Anfang an