

# FLOWPLAN

LEAN & SIMPLE MVP 2026

AI-Native Audit Management System

מערכת ניהול ביקורות מבוססת בינה מלאכותית

LEAN MVP

PRD V2.0

תאריך: ינואר 2026

סטטוס: Internal MVP

גישה: De-Bloated & Task-First

עקרון מנחה: DE-BLOATING מוחלט

המטרה: לבנות כלי שהוא "Instantly Usable" - ללא שום bloat טכני או פונקציונלי. כל פיצ'ר שלא משרת ישירות את ליבת המוצר - לא נכנס.

## תוכן עניינים

01 מה הוסר (De-Bloating)

02 חזון ומטרות

03 Tech Stack

04 מודל הרשאות פשוט

05 מודל הנתונים (Lean Schema)

06 Scheduling Engine (CPM)

07 ארכיטקטורת AI מבוססת מקורות

08 ממשק משתמש (Brutalist)

09 דרישות פונקציונליות

10 אסטרטגיית בדיקות

11 שלבי פיתוח

---

## 01. מה הוסר (DE-BLOATING)

---

לפני שמתחילים - הנה רשימת כל מה שלא נפתח ב-MVP:

✕ **תמחור וחיוב** - אין Pricing Tiers, אין Stripe, אין Billing Logic.  
המערכת פנימית בלבד בשלב MVP.

---

✕ **הרשאות מורכבות** - אין RBAC מורכב.  
מודל פשוט: Admin / Member בלבד.

---

✕ **Dashboard Builders** - אין בונה דשבורדים.  
תצוגת רשימה אחת נקייה = מקור אמת יחיד.

---

✕ **ניהול משאבים מורכב** - אין Resource Leveling.  
הקצאה פשוטה למשימות בלבד.

---

✕ **Sprints / Agile מורכב** - אין ספרינטים, אין Velocity.  
פוקוס על Tasks + Critical Path בלבד.

---

✕ **אינטגרציות חיצוניות** - אין Slack, Jira, GitHub sync.  
מערכת עצמאית. API לעתיד.

---

✕ **Multi-language** - עברית בלבד.  
פשוטות מקסימלית.

---

## 02. חזון ומטרות

מה זה FLOWPLAN?

FlowPlan היא מערכת ניהול ביקורות (Audit Management System) מינימליסטית ומבוססת AI. המערכת מתמקדת בדבר אחד: ניהול משימות/מצאים עם תזמון חכם.

### עקרונות ליבה

עיקרון	משמעות
Task-First	הכל סובב סביב ה-Task/Finding. זה מקור האמת היחיד.
Instantly Usable	משתמש חדש מתחיל לעבוד תוך 30 שניות. אפס Onboarding.
Offline-First	עובד בשטח ללא אינטרנט (CRDT sync).
Grounded AI	ה-AI עונה רק על בסיס מסמכים שהועלו. Source Attribution.

### מדדי הצלחה (MVP)

0	100%	30s>
EXTERNAL DEPENDENCIES	OFFLINE CAPABILITY	TIME TO FIRST TASK

# TECH STACK .03

עקרון: פשטות ומהירות

Stack מינימלי שמאפשר פיתוח מהיר עם יכולת Offline-First.

שכבה	טכנולוגיה	סיבה
Framework	Next.js 15 (App Router)	SSR + API Routes במקום אחד
Styling	Tailwind CSS + Shadcn/UI	עיצוב מהיר, קומפוננטים מוכנים
Database	Supabase (PostgreSQL)	Auth + DB + Realtime מובנה
Real-time / Offline	CRDT (Loro או Yjs)	Offline-first, מניעת קונפליקטים
AI	Claude API (Anthropic)	Grounded AI עם Source Attribution

## למה CRDT?

**CRDT (Conflict-free Replicated Data Types)** מאפשר לכמה משתמשים לערוך בו-זמנית ללא "Digital Soup" - בעיקר קריטי לביקורות שטח ללא אינטרנט.

- משתמש א' עורך Offline
- משתמש ב' עורך Online
- כשמשתמש א' חוזר לרשת - Merge אוטומטי ללא קונפליקטים

## 04. מודל הרשאות פשוט

FLAT MODEL בלבד

אין RBAC מורכב. שני תפקידים בלבד:

תפקיד	הרשאות
ADMIN	<ul style="list-style-type: none"><li>יצירה/עריכה/מחיקה של פרויקטים</li><li>ניהול משתמשים (הזמנה/הסרה)</li><li>גישה לכל המשימות והדוחות</li><li>העלאת מסמכים ל-AI</li></ul>
MEMBER	<ul style="list-style-type: none"><li>צפייה בפרויקטים שמוקצים אליו</li><li>יצירה/עריכה של משימות</li><li>עדכון סטטוס משימות</li><li>שימוש ב-AI לשאלות</li></ul>

---

## 05. מודל הנתונים (LEAN SCHEMA)

---

5 טבלאות בלבד

סכמה מינימלית שמכסה את כל הצרכים של MVP.

### 1. PROJECTS

```
CREATE TABLE projects ( id UUID PRIMARY KEY DEFAULT gen_random_uuid(), name TEXT NOT NULL, description TEXT, status TEXT DEFAULT 'active', -- active, completed, archived start_date DATE, end_date DATE, created_by UUID REFERENCES auth.users(id), created_at TIMESTAMPTZ DEFAULT now(), updated_at TIMESTAMPTZ DEFAULT now() );
```

### 2. TASKS

```
CREATE TABLE tasks ( id UUID PRIMARY KEY DEFAULT gen_random_uuid(), project_id UUID REFERENCES projects(id) ON DELETE CASCADE, title TEXT NOT NULL, description TEXT, status TEXT DEFAULT 'pending', -- pending, in_progress, done priority TEXT DEFAULT 'medium', -- low, medium, high, critical assignee_id UUID REFERENCES auth.users(id), -- Scheduling fields (CPM) duration INTEGER DEFAULT 1, -- Duration in working days start_date DATE, end_date DATE, es DATE, -- Early Start ef DATE, - - Early Finish ls DATE, -- Late Start lf DATE, -- Late Finish slack INTEGER DEFAULT 0, -- Slack = LS - ES is_critical BOOLEAN DEFAULT false, -- On Critical Path? created_at TIMESTAMPTZ DEFAULT now(), updated_at TIMESTAMPTZ DEFAULT now() );
```

## DEPENDENCIES .3

```
CREATE TABLE dependencies ( id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
predecessor_id UUID REFERENCES tasks(id) ON DELETE CASCADE, successor_id UUID
REFERENCES tasks(id) ON DELETE CASCADE, type TEXT DEFAULT 'FS', -- FS, SS, FF, SF
lag_days INTEGER DEFAULT 0, created_at TIMESTAMPTZ DEFAULT now(),
UNIQUE(predecessor_id, successor_id) );
```

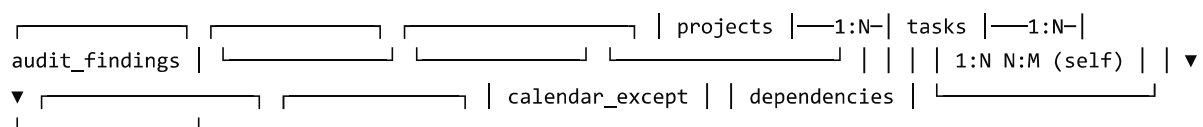
## AUDIT\_FINDINGS .4

```
CREATE TABLE audit_findings ( id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
task_id UUID REFERENCES tasks(id) ON DELETE CASCADE, severity TEXT NOT NULL, --
critical, high, medium, low finding TEXT NOT NULL, root_cause TEXT, capa TEXT, --
Corrective Action / Preventive Action due_date DATE, status TEXT DEFAULT 'open', -
- open, in_progress, closed created_at TIMESTAMPTZ DEFAULT now() );
```

## CALENDAR\_EXCEPTIONS .5

```
CREATE TABLE calendar_exceptions ( id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
project_id UUID REFERENCES projects(id) ON DELETE CASCADE, date DATE NOT NULL,
type TEXT NOT NULL, -- holiday, non_working name TEXT, -- e.g., "יום כיפור"
UNIQUE(project_id, date) );
```

## מפושט ERD





## SCHEDULING ENGINE (CPM) .06

### CRITICAL PATH METHOD

מנוע תזמון שמזהה אילו משימות לא ניתן לעכב ללא השפעה על מועד הסיום.

### אלגוריתם ב-3 שלבים

שלב 1: Forward Pass (חישוב EF, ES)

$$lag + (predecessors \text{ של } EF)ES = max$$
$$EF = ES + duration - 1$$

שלב 2: Backward Pass (חישוב LF, LS)

$$lag - (successors \text{ של } LS)LF = min$$
$$LS = LF - duration + 1$$

שלב 3: Slack Calculation

$$Slack = LS - ES$$

אם  $Slack = 0$  → משימה על Critical Path

## CALENDAR AWARENESS

כללי לוח שנה

- **שבת/ראשון:** המנוע מדלג אוטומטית (ניתן להגדרה)
- **חגים:** נשלפים מטבלת `calendar_exceptions`
- **כלל ברזל:** משימה יכולה להתחיל ביום לא-עבודה, אך לא להסתיים בו

## TYPESCRIPT PSEUDOCODE

```
class SchedulingService {
  async calculateCriticalPath(projectId: string) {
    // 1. Get all tasks & dependencies
    const tasks = await this.getTasks(projectId);
    const deps = await this.getDependencies(projectId);
    const holidays = await this.getHolidays(projectId);
    // 2. Forward Pass
    for (const task of this.topologicalSort(tasks, deps)) {
      task.es = this.calculateEarlyStart(task, deps, holidays);
      task.ef = this.addWorkingDays(task.es, task.duration, holidays);
    }
    // 3. Backward Pass
    for (const task of this.topologicalSort(tasks, deps).reverse()) {
      task.lf = this.calculateLateFinish(task, deps, holidays);
      task.ls = this.subtractWorkingDays(task.lf, task.duration, holidays);
    }
    // 4. Calculate Slack & Mark Critical
    for (const task of tasks) {
      task.slack = this.workingDaysBetween(task.es, task.ls, holidays);
      task.is_critical = task.slack === 0;
    }
    return tasks;
  }
}
```

## 07. ארכיטקטורת AI מבוססת מקורות

GROUNDING AI = NOTEBOOKLM STYLE

ה-AI לא "ממציא" - הוא עונה רק על בסיס מסמכים שהועלו לפרויקט. כל תשובה כוללת ציטוט המקור.

איך זה עובד

שלב	פעולה
1. העלאה	Admin מעלה מסמכי פרויקט (PDF, DOCX, TXT)
2. עיבוד	המסמכים מפורקים ל-chunks ונשמרים עם embeddings
3. שאילתה	משתמש שואל שאלה בשפה טבעית
4. RAG	המערכת מוצאת chunks רלוונטיים (similarity search)
5. תשובה	Claude מגיב עם ציטוטים מדויקים מהמקורות

שאלה: "מה דרישות הבטיחות לפי תקן ISO?"

תשובה:

לפי סעיף 4.2.1 במסמך "ISO\_Requirements.pdf", דרישות הבטיחות כוללות:

1. בדיקות תקופתיות כל 6 חודשים
2. תיעוד כל אירוע חריג
3. הדרכת עובדים שנתית

[מקור: ISO\_Requirements.pdf, עמוד 12, שורות 45-52]

## SOURCE ATTRIBUTION חובה

כלל ברזל: אם ה-AI לא מוצא מידע במסמכים - הוא עונה:

"לא נמצא מידע רלוונטי במסמכי הפרויקט. אנא העלה מסמכים נוספים או נסח את השאלה מחדש."

# 08. ממשק משתמש (BRUTALIST)

## עקרונות עיצוב

- Brutalist: ניגודיות גבוהה, טיפוגרפיה חזקה, ללא עיטורים מיותרים
- Progressive Disclosure: פרטים נוספים רק כשצריך
- Task-First: הכל מוביל למשימה

## מסך יחיד: TASK LIST

				FLOWPLAN [+ חדש] [AI] [👤]			
				[▼] סינון [Q1 2026 פרויקט: ביקורת]			
				[CRITICAL] ■       בדיקת מערכת כיבוי אש			
דני   אחראי: 15/02   תאריך יעד: Slack: 0							
				[HIGH] □       עדכון נהלי בטיחות			
מיכל   אחראי: 20/02   תאריך יעד: Slack: 3							
				[MEDIUM] □       הכנת דוח סיכום			
אחראי   28/02   תאריך יעד: -   Slack: 8							
				Critical Path משימות על       מסומנות			
ב- ■ אדום							

## חוקי UI

עיצוב	אלמנט
רקע אדום בהיר, אייקון ■ במקום □	משימות Critical Path
תגית אדומה	Priority: Critical
תגית כתומה	Priority: High
טקסט מודגש באדום	Slack = 0
קו חוצה, אפור	משימות Done

## 09. דרישות פונקציונליות

### 9.1 ניהול פרויקטים

מזהה	דרישה	עדיפות
FR-001	יצירת פרויקט (שם, תיאור, תאריכים)	CORE
FR-002	הגדרת ימים לא-עובדים (חגים, שבתות)	CORE
FR-003	ארכיון פרויקטים ישנים	V2

### 9.2 ניהול משימות

מזהה	דרישה	עדיפות
FR-010	CRUD משימות (כותרת, תיאור, עדיפות)	CORE
FR-011	הגדרת Duration (ימי עבודה)	CORE
FR-012	הקצאת משימה למשתמש	CORE
FR-013	שינוי סטטוס (Pending → In Progress → Done)	CORE
FR-014	הגדרת תלויות בין משימות	CORE
FR-015	חישוב Critical Path אוטומטי	CORE
FR-016	הצגת Slack לכל משימה	CORE

9.3 ממצאי ביקורת (AUDIT FINDINGS)

מזהה	דרישה	עדיפות
FR-020	הוספת ממצא למשימה (Severity, Description)	CORE
FR-021	הגדרת Root Cause	CORE
FR-022	הגדרת CAPA (Corrective Action)	CORE
FR-023	מעקב סטטוס ממצא (Open → Closed)	CORE

AI 9.4

מזהה	דרישה	עדיפות
FR-030	העלאת מסמכי פרויקט	CORE
FR-031	שאלות בשפה טבעית	CORE
FR-032	Source Attribution בכל תשובה	CRITICAL



# 10. אסטרטגיית בדיקות

## בדיקות קריטיות

קטגוריה	בדיקה	סוג
Scheduling	Forward Pass מחשב ES/EF נכון	Unit
Scheduling	Backward Pass מחשב LS/LF נכון	Unit
Scheduling	זיהוי Critical Path (Slack=0)	Unit
Scheduling	דילוג על שבתות וחגים	Unit
CRDT	Merge ללא קונפליקטים אחרי Offline	Integration
AI	תשובה כוללת Source Attribution	Integration
AI	תשובה "לא נמצא" כשאין מידע	Integration
E2E	יצירת משימה + תלות + ראיית CP	E2E

## כלי בדיקה

סוג	כלי
Unit	Vitest
Integration	Vitest + Supabase Local
E2E	Playwright

11. שלבי פיתוח

שלב	תוצרים	אבן דרך
1. Setup	פרויקט Next.js + Supabase סכמת DB (5 טבלאות) Auth בסיסי	Skeleton פעיל
2. Tasks + CPM	CRUD משימות SchedulingService הצגת Critical Path	MVP פנימי
3. Audit Findings	ממצאים + CAPA סינון לפי Severity	Audit Flow שלם
4. CRDT + Offline	Yjs integration Offline-first editing	Field-ready
5. Grounded AI	העלאת מסמכים RAG + Source Attribution	AI Enabled