

# FLOWPLAN

---

LEAN & SIMPLE MVP 2026

AI-Native Audit Management System

מערכת ניהול ביקורת מבוססת בינה מלאכותית

LEAN MVP PRD V2.0

תאריך: ינואר 2026

סטטוס: Internal MVP

גישה: De-Bloated & Task-First

## עקרון מנהה: DE-BLOATING מוחלט

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

### תוכן ענייניים

01 מה הופר (De-Bloating)

02 חזון ומטרות

Tech Stack 03

04 מודל הרשות פשוט

05 מודל הנתונים (Lean Schema)

Scheduling Engine (CPM) 06

07 ארכיטקטורת AI מבוססת מקורות

08 ממשק משתמש (Brutalist)

09 דרישות פונקציונליות

10 אסטרטגיית בדיקות

11 שלבי פיתוח

# 01. מה הוסר (DE-BLOATING)

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

**תמhor וחייב** - אין **Billing Logic**, **Stripe**, **Pricing Tiers**.  
המערכת פנימית בלבד בשלב MVP.



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



**Dashboard Builders** - אין בונה דשبورדים.  
תצוגת רשימה אחת נקייה = מקורה אמת ייחד.



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



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



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



**Multi-language** - עברית בלבד.  
פתרונות מקסימלית.



## 02. חזון ומטרות

מה זה FlowPlan?

FlowPlan היא מערכת ניהול ביקורת (Audit Management System) מינימליסטית ובסיסת AI. המערכת מתחזקת בדבר אחד: ניהול משימות/ממצאים עם תזמון חכם.

### עקרונות ליבה

משמעות	עיקרונו
הכל סובב סביב ה-Task/Finding. זה מקור האמת היחיד.	<b>Task-First</b>
משתמש חדש מתחילה לעבוד תוך 30 שניות. אף	<b>Instantly Usable</b>
עובד בשטח ללא אינטרנט (CRDT sync).	<b>Offline-First</b>
ה-AI עונה רק על בסיס מסמכים שהועלו.	<b>Grounded AI</b>

### מדדי הצלחה (MVP)



## TECH STACK .03

עקרון: פשוטות ו מהירות

.Offline-First Stack

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

למה ?CRDT

CRDT (Conflict-free Replicated Data Types) מאפשר לכמה משתמשים לעורך בו-זמנית ללא בעיקר קרייטי לביקורת שטח ללא אינטרנט.

- משתמש א' עורך Offline
- משתמש ב' עורך Online
- כמשתמש א' חזר לרשא - Merge אוטומטי ללא קונפליקטים

## 04. מודל הרשאות פשוט

FLAT MODEL בלבד

אין RBAC מורכב. שני תפקידים בלבד:

הרשאות	תפקיד
יצירה/עריכה/מחיקה של פרויקטים ניהול משתמשים (זמן/הסра) גישה לכל המשימות והדוחות העלאת מסמכים ל-AI	ADMIN
צפיה בפרויקטים שמוקצים אליו יצירה/עריכה של משימות עדכון סטטוס משימות שימוש ב-AI לשאלות	MEMBER

---

## 05. מודל הנתונים (LEAN SCHEMA)

---

### 5 טבלאות בלבד

סכמה מינימלית שמכסה את כל הצרcis של MVP.

### PROJECTS .1

```
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() );
```

### TASKS .2

```
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) );
```

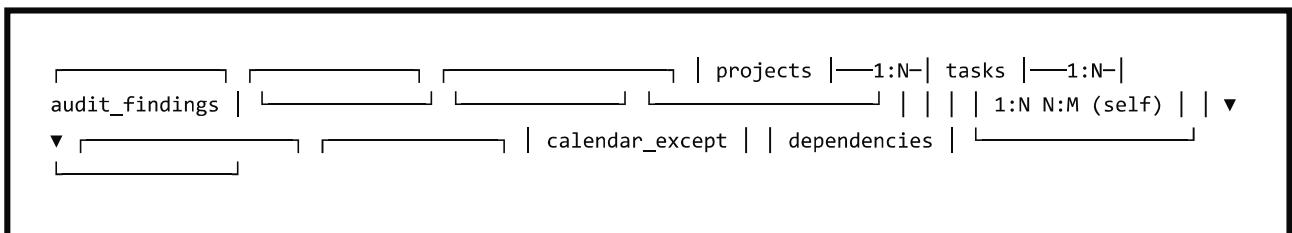
## AUDIT\_FINDINGS .4

```
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) );
```

## הפושט ERD



# SCHEDULING ENGINE (CPM) . 06

## CRITICAL PATH METHOD

מנוע תזמון שמצויה אילו משימות לא ניתן לעכבר ללא השפעה על מועד הסיום.

### אלגוריתם ב- 3 שלבים

שלב 1 : Forward Pass (ES, EF חישוב)

$$lag + (predecessors-EF)ES = max$$

$$EF = ES + duration - 1$$

שלב 2 : Backward Pass (LS, LF חישוב)

$$lag - (successors-LS)LF = min$$

$$LS = LF - duration + 1$$

שלב 3 : Slack Calculation

$$Slack = LS - ES$$

Critical Path → משימה על  $Slack = 0$  אם

## CALENDAR AWARENESS

כלי לוח שנה

- **שבת/ראשון:** המנוע מדריך אוטומטית (ניתן להגדלה)
- **הגים:** נשלפים מטבלת calendar\_exceptions
- **כלל ברזל:** משימה יכולה להתחיל ביום לא-עובד, 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

GROUNDED AI = NOTEBOOKLM STYLE

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

איך זה עובד

שלב	פעולה
1. העלאה	Admin מעלת מסמכים פרויקט (PDF, DOCX, TXT)
2. עיבוד	המסמכים מפורקים ל-chunks ונשמרים עם embeddings
3. שאלתה	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, עמ"ו 12, שורות 45-52]

### **SOURCE ATTRIBUTION**

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

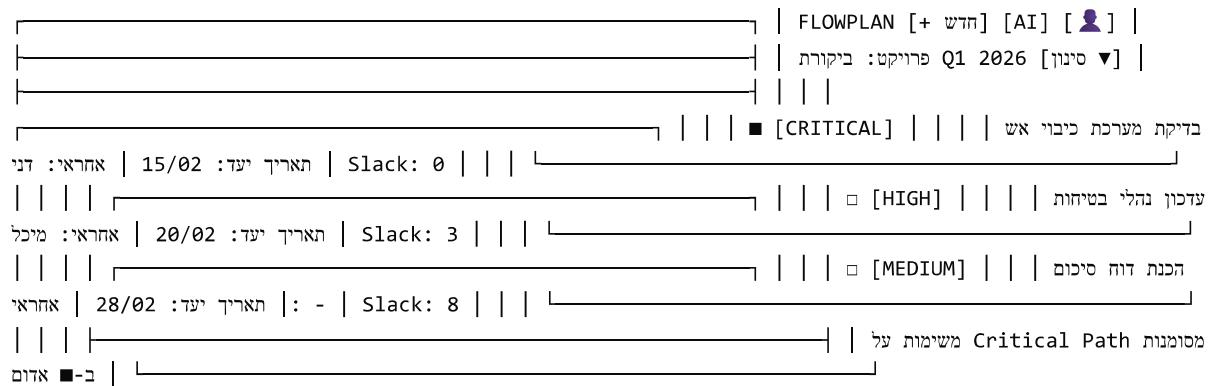
"לא נמצא מידע רלוונטי במסמכי הפרויקט. אנא העלה מסמכים נוספים או נסה את השאלה מחדש."

## 08. ממשק משתמש (BRUTALIST)

### עקרונות עיצוב

- ניגודיות גובהה, טיפוגרפיה חזקה, ללא עיטורים מיוחדים :Brutalist
- פרטימ נספפים רק כנדרין :Progressive Disclosure
- הכל מוביל למשימה :Task-First

### מסך יחיד : TASK LIST



עיצוב	אלמנט
רקע אדום בהיר, איקון ■ במקומות □	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

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

### 9.2 ניהול משימות

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

### 9.3 ממצאים ביקורת (AUDIT FINDINGS)

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

### AI 9.4

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

## 10. אסטרטגיית בדיקות

### בדיקות קריטיות

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

### כלי בדיקה

כלי	סוג
Vitest	Unit
Vitest + Supabase Local	Integration
Playwright	E2E

## 11. שלבי פיתוח

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

**FLOWPLAN PRD v2.0 – LEAN MVP**

Internal Use Only • January 2026