

"תוכנית מערכת ניהול תקין לעמותות" - מהטפת

Context

מלאה לניהול תקין של עמותות בישראל עם אוטומציה של 100%. המערכת מספקת שירות SaaS בניית פלטפורמת וכוללת: ניהול כספי, תורמים, ועד מנהל, מתנדבים, דיווחים רגולטוריים, ומנוע (Multi-tenant) למספר עמותות אוטומציות מלא. הפרויקט מתחיל מאפס בתיקייה ריקה.

Tech Stack

שכבה	טכנולוגיה
Framework	Next.js 14+ (App Router), TypeScript
UI	Tailwind CSS + shadcn/ui (RTL מובנה)
Database	PostgreSQL + Prisma ORM
Auth	NextAuth.js (Auth.js v5)
State	React Query (TanStack Query) + Zustand
Jobs/Queue	BullMQ + Redis
File Storage	S3-compatible (MinIO dev / AWS S3 prod)
PDF	@react-pdf/renderer + Puppeteer
Email	Nodemailer + React Email templates
Monorepo	Turborepo

מבנה פרויקט (Project Structure)

```
mahatefet/
├── turbo.json
├── package.json
├── .env.example
├── docker-compose.yml          # PostgreSQL, Redis, MinIO
├── apps/
│   ├── web/                   # Next.js App (Frontend + API)
│   │   ├── app/
│   │   │   ├── (auth)/        # Login, Register, Forgot Password
│   │   │   ├── (public)/      # Landing page, Pricing
│   │   │   ├── (dashboard)/   # Main app (protected)
│   │   │   ├── layout.tsx     # Dashboard shell + sidebar
│   │   │   ├── page.tsx       # Overview dashboard
│   │   │   ├── finance/       # ניהול כספי
│   │   │   └── donors/        # ניהול תורמים
```

```

├── compliance/ # ניהול תקין
├── board/      # ועד מנהל
├── volunteers/ # מתנדבים
├── automation/ # אוטומציות
├── reports/    # דוחות
├── integrations/ # אינטגרציות
├── settings/   # הגדרות ארגון
├── api/v1/     # API Routes
├── components/ # UI Components
├── lib/        # Utils, hooks, services
├── middleware.ts # Auth + Tenant + Locale
├── worker/     # Background Job Processor
├──   └── src/
├──     ├── workers/ # BullMQ workers
├──     ├── jobs/    # Job definitions
├──     └── index.ts  # Worker entry
├── packages/
├──   ├── db/        # Prisma schema + client
├──   │   ├── prisma/
├──   │   │   ├── schema.prisma
├──   │   │   ├── seed.ts
├──   │   │   └── migrations/
├──   │   └── src/
├──   │       ├── client.ts # Prisma client singleton
├──   │       └── tenant.ts  # Multi-tenant Prisma extension
├──   ├── shared/      # Shared types, validators, constants
├──   │   ├── types/
├──   │   ├── validators/ # Zod schemas
├──   │   └── constants/
├──   └── integrations/ # External service adapters
├──     └── src/
├──         ├── base-adapter.ts
├──         ├── invoice-green/
├──         ├── icount/
├──         ├── tranzila/
├──         ├── cardcom/
├──         ├── whatsapp/
├──         ├── google-workspace/
├──         ├── tax-authority/
├──         └── registrar/

```

מודלים מרכזיים - Database Schema

Multi-Tenancy: Row-Level Security

query מוסיף פילטר אוטומטי לכל `organizationId` - Prisma Client Extension כל טבלה מכילה

(מודלים עיקריים (~40 טבלאות):

ליבה:

- **Organization** - (עמותה (שם, מספר עמותה, כתובת, הגדרות
- **User** - (משתמש (אימייל, סיסמה, שם
- **Membership** - (ADMIN/MANAGER/ACCOUNTANT/VIEWER) תפקיד + Organization ב User חברות של
- **AuditLog** - לוג מלא של כל פעולה במערכת

כספי:

- **Invoice** - (חשבוניות (מספור רציף, סטטוס, סכום
- **Receipt** - (קבלות (כולל סעיף 46
- **BankTransfer** - העברות בנקאיות
- **Budget** - תקציב שנתי
- **BudgetLine** - שורות תקציב
- **FinancialTransaction** - תנועות כספיות
- **DocumentSequence** - מספור רציף חוקי ללא פערים

תורמים:

- **Donor** - פרופיל תורם
- **Donation** - תרומות
- **RecurringDonation** - הוראות קבע
- **Campaign** - קמפיינים
- **DonorCommunication** - תקשורת עם תורמים

ועד מנהל:

- **BoardMeeting** - ישיבות ועד
- **MeetingProtocol** - פרוטוקולים
- **BoardResolution** - החלטות
- **BoardMember** - חברי ועד

מתנדבים:

- **Volunteer** - מתנדב
- **VolunteerShift** - משמרות
- **VolunteerHours** - שעות

ניהול תקין:

- **ComplianceItem** - (deadline פריטי ציות (עם
- **ComplianceCertificate** - אישורי ניהול תקין
- **AnnualReport** - דוחות שנתיים
- **OrganizationDocument** - מסמכי יסוד

אוטומציות:

- **Workflow** - (trigger → conditions → actions) תהליך עבודה
- **WorkflowExecution** - הרצות

- `WorkflowStep` - צעדים
- `Notification` - התראות
- `ScheduledTask` - משימות מתוזמנות

אינטגרציות:

- `IntegrationConfig` - הגדרות אינטגרציה (credentials מוצפנים AES-256-GCM)
- `IntegrationLog` - לוגים של אינטגרציות

Automation Engine) מנוע אוטומציות

ארכיטקטורה תלת-שכבתית:

1. **Trigger Layer** - ידני (Cron), לוח זמנים (Event), מה מפעיל: אירוע
2. **Matching Layer** - תנאים: סינון לפי כללים (if/else)
3. **Execution Layer** - פעולות: שליחת מייל, יצירת מסמך, עדכון רשומה - API call

Event Bus (Redis Pub/Sub):

workflows מנוע האוטומציות מאזין ומפעיל → event כל פעולה במערכת מפרסמת

(Pre-built Templates): תבניות מוכנות מראש

1. קבלה אוטומטית בקבלת תרומה
2. תזכורת לפני תום תוקף אישור ניהול תקין
3. סיכום ישיבת ועד אוטומטי
4. דוח חודשי אוטומטי לנציגי הארגון
5. תזכורת להגשת דוח שנתי
6. ברכת שנה טובה/חג לתורמים
7. אישור שעות מתנדבים חודשי
8. התראה על חריגה מתקציב
9. גיבוי מסמכים אוטומטי
10. עדכון סטטוס ניהול תקין מרשם העמותות

UI - Visual Workflow Builder:

React Flow לבניית תהליכים בגרירה (drag & drop).

שכבת אינטגרציות (Integration Layer)

Adapter Pattern:

`BaseIntegrationAdapter` כל אינטגרציה ממשת

```
abstract class BaseIntegrationAdapter {
  abstract connect(config: IntegrationConfig): Promise<void>
  abstract disconnect(): Promise<void>
  abstract testConnection(): Promise<boolean>
}
```

```
abstract executeAction(action: string, params: any): Promise<any>
}
```

אינטגרציות מתוכננות:

אינטגרציה	סוג	זמין API?
חשבונית ירוקה	REST API	כן
iCount	REST API	כן
Tranzila	REST API	(כן) (סליקה)
CardCom	REST API	(כן) (סליקה)
WhatsApp Business	Cloud API	כן
Google Workspace	OAuth2 + APIs	כן
מס הכנסה	XML/SOAP	חלקי
רשם העמותות	Web Scraping + API	(בסיסי API חלקי) (יש)
בנקים	OFX/MT940	(ישיר API קבצים) (לא)

שלבי מימוש (Implementation Phases)

שבועות 1-3 - (Foundation) שלב 1: בסיס

קבצים קריטיים ליצירה:

- turbo.json, package.json, docker-compose.yml
- packages/db/prisma/schema.prisma - כל הסכמה
- apps/web/middleware.ts - Auth + Tenant
- apps/web/app/(auth)/ - Login/Register
- apps/web/app/(dashboard)/layout.tsx - Shell + Sidebar
- apps/web/components/ui/ - shadcn/ui components

מה נבנה:

- ☐ Turborepo monorepo אתחול
- ☐ Docker Compose (PostgreSQL, Redis, MinIO)
- ☐ Prisma schema מודלים (40~)
- ☐ NextAuth.js authentication
- ☐ Multi-tenant middleware + Prisma extension
- ☐ Dashboard layout עם sidebar (RTL)
- ☐ shadcn/ui components בסיסיים
- ☐ RBAC (Role-Based Access Control)
- ☐ Audit logging infrastructure

שלב 2: כספי + תורמים - שבועות 4-7

קבצים קריטיים:

- `apps/web/app/(dashboard)/finance/` - כל עמודי הכספים
- `apps/web/app/(dashboard)/donors/` - ניהול תורמים
- `apps/web/app/api/v1/finance/` - כספי API
- `apps/web/app/api/v1/donors/` - תורמים API
- `packages/db/src/services/document-sequence.ts` - מספור רציף

מה נבנה:

- ☐ PDF, ניהול חשבוניות (יצירה, עריכה, שליחה)
- ☐ (ניהול קבלות) כולל קבלות סעיף 46
- ☐ (DocumentSequence) מספור רציף חוקי
- ☐ ניהול העברות בנקאיות
- ☐ תקציב שנתי + מעקב
- ☐ דוחות כספיים בסיסיים
- ☐ (תורמים) פרופיל, היסטוריה CRM
- ☐ (ניהול תרומות) חד-פעמי + הוראת קבע
- ☐ קמפיינים
- ☐ הפקת קבלות אוטומטית בקבלת תרומה

שלב 3: ניהול תקין + ועד מנהל - שבועות 8-10

קבצים קריטיים:

- `apps/web/app/(dashboard)/compliance/` - ניהול תקין
- `apps/web/app/(dashboard)/board/` - ועד מנהל
- `apps/web/app/api/v1/compliance/`
- `apps/web/app/api/v1/board/`

מה נבנה:

- ☐ צ'קליסט ניהול תקין אוטומטי
- ☐ מעקב אישורים ותעודות
- ☐ ניהול מסמכי יסוד
- ☐ יצירת דוח שנתי אוטומטי
- ☐ deadlines התראות
- ☐ ניהול ישיבות ועד
- ☐ יצירת פרוטוקולים אוטומטית
- ☐ מעקב החלטות ומשימות
- ☐ הצבעות דיגיטליות

שלב 4: מתנדבים + מנוע אוטומציות - שבועות 11-14

קבצים קריטיים:

- `apps/web/app/(dashboard)/volunteers/`
- `apps/web/app/(dashboard)/automation/`
- `apps/worker/src/workers/automation.worker.ts` - הלב של האוטומציות
- `apps/worker/src/workers/` - workers-כל ה

סך גבנה:

- ☐ ניהול מתנדבים (רישום, שיבוץ, מעקב שעות)
- ☐ Event Bus (Redis Pub/Sub)
- ☐ Workflow engine (trigger → condition → action)
- ☐ Visual Workflow Builder (React Flow)
- ☐ תכניות אוטומציה מוכנות 10
- ☐ Scheduled tasks (cron)
- ☐ Notification system (email, push)
- ☐ BullMQ workers setup

שלב 5: אינטגרציות + אנליטיקס - שבועות 15-19

קבצים קריטיים

- packages/integrations/src/ - כל adapters
- apps/web/app/(dashboard)/integrations/
- apps/web/app/(dashboard)/reports/

סך גבנה:

- ☐ Integration adapter base class
- ☐ iCount adapter / חשבונית ירוקה
- ☐ Tranzila / CardCom adapter (סליקה)
- ☐ WhatsApp Business adapter
- ☐ Google Workspace adapter
- ☐ Email automation (Nodemailer + React Email)
- ☐ adapter (API + scraping) רשם העמותות
- ☐ דשבורד אנליטיקס
- ☐ מותאם אישית Report builder

ייצור - שבועות 20-24 + SaaS Billing: שלב 6

קס: גבנה

- ☐ Landing page + Pricing page
- ☐ Subscription management (Stripe/PayPlus)
- ☐ Onboarding wizard לעמותות חדשות
- ☐ E2E בדיקות (Playwright)
- ☐ Performance optimization
- ☐ Security audit
- ☐ Deployment (Vercel + Railway/Render)
- ☐ Monitoring (Sentry + analytics)
- ☐ Documentation

החלטות עיצוב מפתח

1. **Multi-tenancy: Row-Level Security** - מוסיף פילטר Prisma extension `organizationId`, כל טבלה עם `schema-per-tenant`-אוטומטי, פשוט יותר מ-10,000 ארגונים, מספיק טוב ל-10,000 ארגונים.

- מבטיח - transaction בתוך `SELECT FOR UPDATE` טבלה נפרדת עם: **DocumentSequence (מספור רצף)**: מספור ללא פערים כנדרש בחוק הישראלי.
 - env vars**, ב DB, AES-256-GCM-**מוצפנים**: כל הרשאות אינטגרציה מוצפנות ב **Credentials**.
 - workflows. מנוע אוטומציות מאזין → מפעיל → event מפרסמת mutation כל **Automation** → **Event Bus**. event-driven ארכיטקטורה.
 - PDF Generation**: `@react-pdf/renderer` (מהיר), Puppeteer (לקבלות/חשבוניות), לדוחות מורכבים.
-

אימות (Verification)

בדיקת שלב 1:

```
# הרצת DB + Redis
docker-compose up -d
# מיגרציה
npx prisma migrate dev
# הרצת אפליקציה
npm run dev
# בדיקה: כניסה, יצירת ארגון, הרשאות
```

בדיקת כל שלב:

- Unit tests עם Vitest לכל service
- Integration tests ל-API routes
- E2E tests עם Playwright (לתהליכים קריטיים (יצירת חשבונית, קבלת תרומה
- ידינית בכל עמוד RTL בדיקת

בדיקת אוטומציות:

- וידוא הפעלה → trigger event → workflow יצירת
- Scheduled task → וידוא הרצה בזמן
- Integration test → mock external API → וידוא adapter

בדיקת אינטגרציות:

- Sandbox/test environments לכל שירות חיצוני
- sandbox חשבונית ירוקה
- Tranzila test mode