

Web Application כ-Modern DTP overview TAG+ :name: בניית תוכנת עימוד שולחני מודרנית וחדשנית עם עבודה משותפת בזמן אמת, תמיכה מלאה ב-RTL, ויכולות AI מתקדמות - מחליפה את TAG.EXE הישנה.

id: setup-monorepo content :todos: - id: design-system content :completed: - id: canvas-engine content :completed dependencies: - id: text-editor content :completed dependencies: - id: document-model content :canvas-engine: - id: backend-api content :status: completed dependencies: - id: file- content :operations content :dependencies: - id: typography content :status: completed dependencies: - id: page-management content :status: completed dependencies: - id: style-system content :completed dependencies: - id: image-handling content :status: completed dependencies: - id: websocket-collab content :status: completed dependencies: - id: ai-features content :status: completed dependencies: - id: linked-frames content :completed dependencies: - id: auto-indent-engine content :status: completed dependencies: - id: reflow-engine content :status: completed dependencies: - id: sets-groups-system content (& Groups status: completed dependencies: - id: multi- content :format-frames content :status: completed dependencies: - id: mirror-layout content :status: completed dependencies: - id: sty-import content :status: completed dependencies: - id: ini-import content (Berech הגדרות) TAG המקורי :status: completed dependencies: - id: sets-groups-system

+TAG - תוכנת עימוד שולחני חדשנית

חזון הפרויקט

בניית תוכנת (Desktop Publishing) DTP מודרנית שתחליף את TAG.EXE הישנה עם:

- Figma/Google Docs כמו **Real-time Collaboration**
- **AI-Powered Features** ליצירת תוכן ועיצוב אוטומטי
- **Cloud-Native Architecture** עם עבודה אופליין
- **RTL-First Design** תמיכה מלאה בעברית/ערבית

ארכיטקטורה טכנית

```
graph TB
  subgraph frontend [Frontend - React/Next.js]
    Editor[Canvas Editor]
    Toolbar[Rich Toolbar]
    Panels[Side Panels]
    Collab[Collaboration Layer]
  end

  subgraph backend [Backend - Node.js/Fastify]
    API[REST API]
    WS[WebSocket Server]
    Queue[Job Queue]
  end

  subgraph services [External Services]
    DB[(PostgreSQL)]
    Redis[(Redis)]
    S3[(Object Storage)]
    AI[AI Service]
  end

  Editor --> Collab
  Collab --> WS
  Toolbar --> API
  API --> DB
  WS --> Redis
  API --> S3
  Queue --> AI
```

מבנה הפרויקט

```

TAGPLUS/
├── apps/
│   ├── web/ # Next.js Frontend
│   │   ├── src/
│   │   │   ├── app/ # App Router
│   │   │   ├── components/ # React Components
│   │   │   │   ├── editor/ # Canvas & Editor
│   │   │   │   ├── toolbar/ # Toolbars
│   │   │   │   ├── panels/ # Side Panels
│   │   │   │   └── ui/ # Design System
│   │   │   ├── hooks/ # Custom Hooks
│   │   │   ├── stores/ # Zustand Stores
│   │   │   ├── services/ # API Clients
│   │   │   └── lib/ # Utilities
│   │   └── public/
│   └── api/ # Fastify Backend
│       ├── src/
│       │   ├── routes/ # API Routes
│       │   ├── services/ # Business Logic
│       │   ├── repositories/ # Database Layer
│       │   ├── websocket/ # Real-time
│       │   └── workers/ # Background Jobs
│       └── prisma/
├── packages/
│   ├── editor-core/ # Editor Engine
│   ├── document-model/ # Document Schema
│   ├── collab/ # CRDT Sync
│   └── shared/ # Shared Types
└── docker/

```

טכנולוגיות מרכזיות

| קטגוריה | טכנולוגיה | סיבה |

|-----|-----|-----|

| מודרני ,**Frontend** | Next.js 14 + React 18 | SSR, App Router |

| פשוט ויעיל | **State** | Zustand + Immer |

| עריכה גרפית מתקדמת | **Canvas** | Fabric.js / Konva.js |

| עריכת טקסט עשירה | **Text Editor** | Lexical / ProseMirror |

| סנכרון בזמן אמת | **Collaboration** | Y.js (CRDT) |

| מהיר ו-Type-safe | **Backend** | Fastify + TypeScript |

| **Database** | PostgreSQL + Prisma | Relational + ORM |

| **Cache** | Redis | Sessions + Pub/Sub |

| **Storage** | MinIO / S3 | Assets + Documents |

| יצירת PDF מקצועי | **PDF** | pdf-lib + Puppeteer |

| יצירת תוכן | **AI** | OpenAI API |

פיצ'רים מרכזיים (לפי שלבים)

שלב 1: Core Editor (MVP)

Document Canvas .1

- Multiple pages with zoom/pan
- Rulers and guides (horizontal + vertical)
- Grid system with snap
- Page templates (A4, Letter, Custom)
- Preview mode

Text Frames .2

- Rich text editing (Lexical)
- Font selection and styling
- Paragraph formatting

- RTL/LTR support with BiDi
- Linked text frames (text flow between frames)

.3 Basic Typography (כמו TAG המקורי)

- Font family, size, color
- Hebrew fonts + English fonts (dual font support)
- Bold, Italic, Underline, Strikethrough
- Line height (leading) - auto/fixed/percentage
- Letter spacing (tracking)
- Kerning (manual + automatic from font tables)
- Width tables for precise character positioning
- Baseline shift

.4 File Operations

- Create new document (with size dialog)
- Save/Load (cloud + local)
- Auto-save every X minutes
- Export to PDF
- Import: RTF, TXT, ASCII
- Export: RTF, PDF, HTML

שלב 2: Advanced Layout

.1 Page Management (כמו SECTION 7)

- Master pages with inheritance
- Page numbering (auto + manual)
- Headers/Footers (different first page, odd/even)
- Section breaks (new page, continuous)
- Page breaks (forced, keep with next)
- Margins dialog (top, bottom, left, right, gutter)

.2 Columns and Grids (כמו COLUMN dialog)

- Multi-column layouts (1-12 columns)
- Gutter (חרוז) settings per column
- Text flow between columns

- Column rules (separator lines)
- Column balancing

(!TAG כמו 5 Tabs) **Style System .3**

- **Tab 1**: Basic formatting (font, size, color)
- **Tab 2**: Paragraph settings (alignment, indents)
- **Tab 3**: Tabs and indents (tab stops, first line)
- **Tab 4**: Hyphenation settings
- **Tab 5**: Advanced (keep options, inheritance)
- Style library with import/export
- Based-on styles + Next style

(SECTION 9 כמו) **Graphics .4**

- Image import (PNG, JPG, SVG, WebP, BMP, TIFF)
- Image cropping and scaling (percentage)
- Drop shadows and borders
- Text wrap (square, tight, through)
- **Drop Caps** (אותיות פתיחה חוגגוליות)
- Image frames with customizable borders
- Position: X, Y, Rotation

שלב 3: **Professional Features**

(SECTION 11 כמו) **Tables .1**

- Create and edit tables (rows, columns)
- Cell formatting (borders, background)
- Merge/split cells
- Table styles
- Column width / Row height
- Border styles (solid, dashed, double)
- Accent tables

(SECTION 12 כמו) **Find and Replace .2**

- Text search with options
- **Find Attributes** (search by formatting)

- **Wildcards** support (regex-like)
- Replace all / Replace one by one
- Find special characters
- Case sensitive / Whole word options

(SECTION 10 כמו) **Color Management** .3

- Color picker (RGB, HSL, Hex)
- Swatches palette
- Gradients (linear, radial)
- CMYK support for print
- 2-color / 3-color modes
- Named colors

(SECTION 14 כמו) **Spelling and Grammar** .4

- Hebrew spell check
- Grammar suggestions
- Custom dictionary (add words)
- Ignore list
- **Hyphenation** (חיקוק אוטומטי)
- Hyphenation dictionary
- Word breaks definition

שלב 4: Collaboration and AI

Real-time Collaboration .1

- Multiple cursors with names
- Presence indicators (who's viewing)
- Comments and annotations
- **Version history** (unlimited undo!)
- Branching and merging

(חדשנות!) **AI Features** .2

- Text generation (GPT-powered)
- Auto-layout suggestions
- Image enhancement (upscale, remove bg)

- Translation (Hebrew/English/Arabic)
- Smart text reflow
- Content summarization

Templates and Assets .3

- Template gallery (professional designs)
- Asset library (images, icons, logos)
- Drag-and-drop from library
- Reusable components (save frame as template)
- Cloud sync assets

(SECTION 24 כמו) Notes and Annotations .4

- Side notes (הערות שוליים)
- Note markers in text
- Footnotes and endnotes
- Note reference links
- Remark tags

(SECTION 19 כמו) Special Characters .5

- Symbol picker
- Fixed space (non-breaking)
- Em-dash, En-dash
- Hebrew special chars (ניקוד, טעמים)
- Insert by ASCII code
- Custom character shortcuts

שלב 5: Print and Export

(SECTION 13 כמו) Print Production .1

- Bleed and crop marks
- Color separation (CMYK)
- PDF/X export (print-ready)
- Print preview (WYSIWYG)
- Printer setup dialog

- DPI settings (300/600/1200)
- Virtual printer support

.2 Scripting (כמו 9 - SECTION 20 דיאלוגים!)

- JavaScript scripting API
- Macro recording (record actions)
- Batch processing (multiple files)
- Custom commands
- Script library
- Script editor with syntax highlighting
- Scheduled scripts

.3 Undo/Redo System (כמו 15 SECTION)

- Unlimited undo/redo
- History panel (visual timeline)
- Snapshots (named save points)
- Restore to any point
- Undo per-frame (selective)

.4 Clipboard Operations (כמו 16 SECTION)

- Copy/Cut/Paste
- Paste Special (with/without formatting)
- Multiple clipboard items
- Cross-document paste
- Clipboard history

שלב 6: Advanced Layout Engine (מתוך סקריפט "ברך")

.1 Linked Frames System (מערכת מסגרות מקושרות)

מבוסס על ניתוח BERECH.PRG - מערכת מתקדמת לפריסת ספרות תורנית:

- ****Sets (סטטים)****: קבוצות של מסגרות טקסט מקושרות
 - text IDs, format indices, box indices הגדרת
 - החלפת צדדים בדפים זוגיים/אי-זוגיים - `swap`
 - מרווח מינימלי בין מסגרות - `min_space`
 - דרישה שכל המסגרות קיימות - `all`

- **Groups** (קבוצות): אוספים של סטים
 - רשימת סטים מופרדת בפסיקים
 - תמיכה בטווחים (1-5)
 - על כל הסטים בקבוצה batch ביצוע
- **Visual Selection**: בחירה ויזואלית עם קליק
 - `clickhandler` לזיהוי מסגרות
 - מילוי אוטומטי של הגדרות מהמסגרת שנבחרה

.1 Auto-Indent Engine (מנוע Indent אוטומטי)

אלגוריתם מתקדם מ-Berech_f():

- **חישוב חפיפות**: זיהוי אוטומטי של מסגרות חופפות
- ...


```
overlap = (Top[i] < Total[j]) AND (Total[i] > Top[j])
```

- ...
- **Depth Points** (נקודות עומק):
 - `rTopDP` - עומק עליון של מסגרת
 - `rBoxDP` - גובה המסגרת
 - `rTotalDP` - עומק כולל (top + height)
 - `rKotDP` - עומק כותרת
 - `rContDP` - עומק המסך
- **Dynamic Indents**: לפי מיקום Indents שינוי
 - Left Indent לפי מסגרות משמאל
 - Right Indent לפי מסגרות מימין
 - Indent תמיכה ב-4 רמות (0-3)
- **Grid Snapping**: עיגול לרשת
 - `RoundToGrid(n, rGrid, bDown)` - עיגול למעלה/למטה

.1 Reflow Engine (מנוע זרימה מחדש)

- **Automatic Reflow**: כאשר מסגרת משתנה
- **Word Breaks**: שבירת מילים נכונה
- **Continuation Handling**: טיפול בהמשכים בין עמודים
- **Header Integration**: שילוב כותרות עליונות

CODE System .2 (מערכת קודי עיצוב)

מבוסס על mishna.sty - קודי עיצוב מהירים:

```

...
CODE 01 FT7 BO SZ14 JF QC LD20
↓
Font: 7, Bold, Size: 14pt, Justify Full Center, Leading: 20pt
...

```

- ****קודים זמינים****:
 - `FT{n}` - Font number
 - `BO/NO` - Bold / No Bold
 - `IT` - Italic
 - `SZ{n}` - Size
 - `LD{n}` - Leading
 - `JF/JL/JR/JC` - Justify (Full/Left/Right/Center)
 - `QC/QL/QR` - Quick align
 - `SUP{n}` - Superscript
 - `SUB{n}` - Subscript
 - `CLR{nnn}` - Color (3-digit RGB: 0-7 per channel)
 - `ll{n}cm` - Line length
 - `col{n}` - Column count
- ****Quick Apply****: יישום מהיר עם קיצורי מקלדת
- ****Code Library****: ספריית קודים שמורים

Multi-Format Frames .1 (מסגרות עם פורמטים מרובים)מבוסס על הגדרות `[format N,M]` ב-mishna.sty:

- ****Format per Element****: פורמט שונה לכל אלמנט במסגרת
- ****Format Attributes****:
 - `grid` - רשת אופקית/אנכית
 - `tur2` - רוחב טור שני (multi-column)
 - `x_r/x_l` - לדפים ימניים/שמאליים X מיקום
 - `ltab_r/rtab_r` - Indents שמאליים/ימניים
 - `tsura_r/tsura_l` - צורת התסגרת (width, height, indents)
 - `zerem_r/zerem_l` - זרימת טקסט
- ****Mirror Layouts****: פריסה הפוכה לדפים זוגיים

```

```
Right page: ltab_r=0, rtab_r=6.2cm
Left page: ltab_l=6.2cm, rtab_l=0
```

```

.1 Page Hierarchy (היררכיית עמודים)

מבוסס על הגדרות `[text N]` ב-`mishna.sty`:

- **Ramah** (רמה): היררכיה של טקסטים
 - רמה 0: טקסט ראשי (משנה)
 - רמה 1: פירושים ראשיים (תפארת, עובדיה)
 - רמה 2: פירושים משניים (ע_משנה)
- **Text Properties**:
 - ``shem`` - שם הטקסט
 - ``kovets`` - קובץ מקור
 - ``format_count`` - מספר פורמטים זמינים
 - ``revah`` - ריווח בין טקסטים
 - ``use_kot`` - שימוש בכותרת
 - ``lang_ltr`` - כיוון שפה
 - ``newpage`` - התחלה בעמוד חדש
 - ``lahtoch_ara`` - חיתוך ערה

.1 Layout Links (קישורי פריסה)

מבוסס על `[netunehaluka]` ב-`mishna.sty`:

- **Main/Dependent Links**: טקסט ראשי ותלויים


```

```ini
main_id=1
link_ids=1,2,3
link1=1,3,0 # text 1 3-ל-
```

```
- **Auto-Balance**: איזון אוטומטי בין טקסטים מקושרים

.1 Border and Divider System (מערכת גבולות ומפרידים)

מבוסס על הגדרות `format` ב-`mishna.sty`:

- ****Column Dividers****:
 - `column_divider` - קו מפריד בין טורים
 - `divider_width` - רוחב המפריד
 - `divider_gray` - צבע אפור למפריד
- ****Box Borders****:
 - `border_type` - סוג הגבול
 - `border_pen` - עובי קו
 - `border_fill` - מילוי
 - `border_excess` - חריגה מהמסגרת
 - `border_use` - שימוש בגבול

1. Header System (מערכת כותרות)

מבוסס על [netunekoteret] ב-mishna.sty:

- ****Header Types****: כותרות עליונות ותחתונות
- ****Header Switching****: החלפת כותרות לפי תנאים
 - `hachlafat_koteret` - חתי להחליף
 - `batel_acharon` - ביטול אחרון
- ****Header Spacing****:
 - `merhakilit` - מרחק מיותרת עילית
 - `merhaktatit` - מרחק מיותרת תחתית
 - `merhakara` - מרחק מערה

1. Configuration System (מערכת הגדרות)

מבוסס על INI files ופונקציות GetFS/SetFS:

```

- **Persistent Settings**:
  - Dialog positions (`DialogXY`, `DialogXY:Set`)
  - User preferences per project
  - Set/Group definitions

- **INI Structure**:
  ``ini
  [Settings]
  set_count=3
  kvuza_count=2

```

```
color=024,400
```

```
[set 1]
```

```
shem=טט צד עובדיה
```

```
text=3,4
```

```
format=1,1
```

```
box=1,1
```

```
swap=1
```

```
min_space=5
```

```
...
```

- ****Import/Export****: ייבוא וייצוא הגדרות

שלב 7: RTA Engine (מתוך ניתוח RTA.LIB Bytecode)

מבוסס על ניתוח מלא של 6 קבצי LIB (94,003 bytes, 356 פונקציות):

1. LinkCalc Engine (מנוע חישוב קישורים)

מבוסס על RTAID.LIB - המנוע המרכזי:

- ****LinkCalculator****: חישוב קישורים בין מסגרות

- `LinkCalc_d` - דיאלוג חישוב

- `LinkCalc_f` - פונקציית חישוב ראשית

- `LinkCalc_m` - חישוב מרובה

- `LinkCalc_r` - reverse חישוב

- `LinkCalc_s` - single חישוב

- ****LinkTable****: ניהול טבלת קישורים

- `LinkTable_d` - דיאלוג טבלה

- `LinkTable_ID` - מזהים בטבלה

- `link_table` - מבנה הנתונים

- ****LC Internal Engine****:

- `LC_LastFrame` - מסגרת אחרונה בשרשרת

- `LC_indents` - חישוב הזחות

- `LC_just` - יישור טקסט

- `LC_theo` - חישוב תיאורטי

- `LC_newpage` - מעבר דף

- `LC_info` - מידע על הקישור

MultiLinks System .1 (מערכת קישורים מרובים)

מבוסס על RTAID.LIB ו-RTAID2.LIB:

- ****MultiLinks Processor****:
 - `MultiLinks_a` - קישורים מרובים - all
 - `MultiLinks_f` - פונקציה ראשית
 - `MultiLinks_fix` - תיקון קישורים שבורים
 - `MultiLinks_m` - main processing
 - `MultiLinks_s` - single link processing
- ****Link Validation****: בדיקת תקינות קישורים
- ****Auto-Fix****: תיקון אוטומטי של קישורים

Hatem Engine .1 (מערכת חתימות)

מבוסס על RTAID1.LIB - 16 פונקציות חתימה:

- ****Signature Types****:
 - `hatem_d` - דיאלוג חתימה
 - `hatem_f` - פונקציית חתימה
 - `hatem_x` - extra חתימה
 - `hatem_s` - single חתימה
 - `hatem_t` - type חתימה
 - `hatem_m` - main חתימה
 - `hatem_theo_d` - חתימה תיאורטית
- ****Signature Lists****:
 - `hatem_list` - רשימת כל החתימות
 - Multiple levels (0, 1, 2) לכל סוג
- ****Signature Properties****:
 - Size, Font, Position
 - Min/Max constraints
 - Auto vs Manual

Hatzmed Engine .1 (מערכת הצמדות)

מבוסס על RTAID1.LIB - 12 פונקציות הצמדה:

- **Attachment Types**:
 - `hatzmed_a` - all הצמדה
 - `hatzmed_d` - דיאלוג הצמדה
 - `hatzmed_e` - edit הצמדה
 - `hatzmed_x` - extra הצמדה
 - `hatzmed_m` - main הצמדה
 - `Hatzmed_YK` - הצמדה ימין/קיצוני
- **Attachment Lists**:
 - `hatzmed_list` - רשימת הצמדות
 - Multi-level support
- **Dynamic Attachment**: הצמדה דינמית בזמן אמת

1. Marks System (מערכת סימנים)

מבוסס על RTA.LIB ו-RTAID2.LIB:

- **Mark Operations**:
 - `CreateMarksFile` - יצירת קובץ סימנים
 - `AddMarks_f` - הוספת סימן
 - `RemoveMarks_d` - הסרת סימן (דיאלוג)
 - `RemoveMarks_m` - הסרת סימן (main)
 - `Marks_f` - פונקציית סימנים
- **Mark Properties**:
 - `MarkSize` - גודל
 - `MarkFont` - פונט
 - `MarkCut` - חיתוך
 - `MarkMainSize` / `MarkMainFont` - סימן ראשי

1. Process Management (ניהול תהליכים)

מבוסס על כל קבצי ה-LIB:

- **Process Lifecycle**:
 - `InProcess_d` / `InProcess_plugin` - התחלה
 - `EndProcess_d` / `EndProcess_plugin` - סיום
 - `wait_dlog` - המתנה
 - `UpdateWindow_` - עדכון חלון

- ****Plugin Integration****:
 - `Plugin_d` - דיאלוג plugin
 - `dUpdatePlug` - עדכון plugin

.1 TextID System (מערכת מזהי טקסט)

מבוסס על RTA1.LIB ו-RTAID1.LIB:

- ****TextID Operations****:
 - `dlg_TextID` - דיאלוג TextID
 - `GetIniFile_TextID` - INI-קריאה ח
 - `SetIniFile_TextID` - INI-שמירה ל
 - `dlg_formatik` - דיאלוג פורמט
 - `dlg_kaftor` - דיאלוג כפתור
- ****kvutsa_textids****: מזהי טקסט לפי קבוצה
- ****text_sorted****: טקסטים ממוינים

.1 AutoRTA Engine (עיבוד RTA אוטומטי)

מבוסס על RTA.LIB - הפונקציה המרכזית:

- ****Auto Processing****:
 - `AutoRTA` - עיבוד אוטומטי מלא
 - `AutoStiya` - סטייה אוטומטית
 - `auto_izun` - איזון אוטומטי
 - `Auto_Izun` - איזון מתקדם
- ****Settings****:
 - `SetStiya` / `GetStiya` - סטייה
 - `Theo_Shetach` - שטח תיאורטי
 - `RoundStiya` - עיגול סטייה

.1 FindBreak Engine (מנוע מציאת שבירות)

מבוסס על RTAID3.LIB:

- ****Break Detection****:
 - `FindBreak_` - מציאת שבירה
 - `FindBreak_init` - אתחול חיפוש

```

- `Return_` - החזרה למקור
- **Border System**:
  - `border_use` - שימוש בגבול
  - `just_0` - יישור בסיסי

```

חדשנות מול TAG המקורי

| תכונה | TAG הישן | TAG+ החדש |

|-----|-----|-----|

| פלטפורמה | Windows בלבד | Web (כל דפדפן) |

| עבודה משותפת | אין | Real-time כמו Figma |

| AI | אין | יצירת תוכן, תרגום, הצעות |

| ענן | אין | שמירה אוטומטית בענן |

| מובייל | אין | רספונסיבי מלא |

| RTL | בסיסי | BiDi מתקדם |

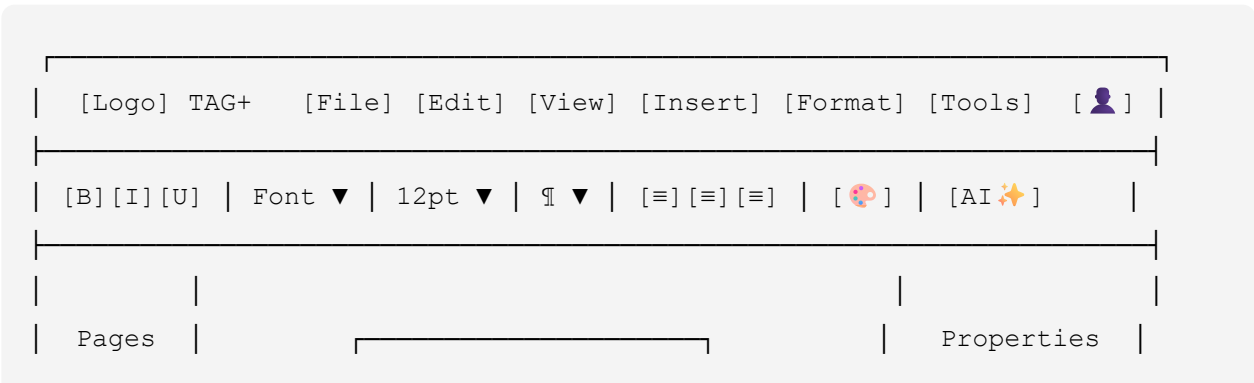
| הגנה | דונגל USB | ללא (קוד פתוח) |

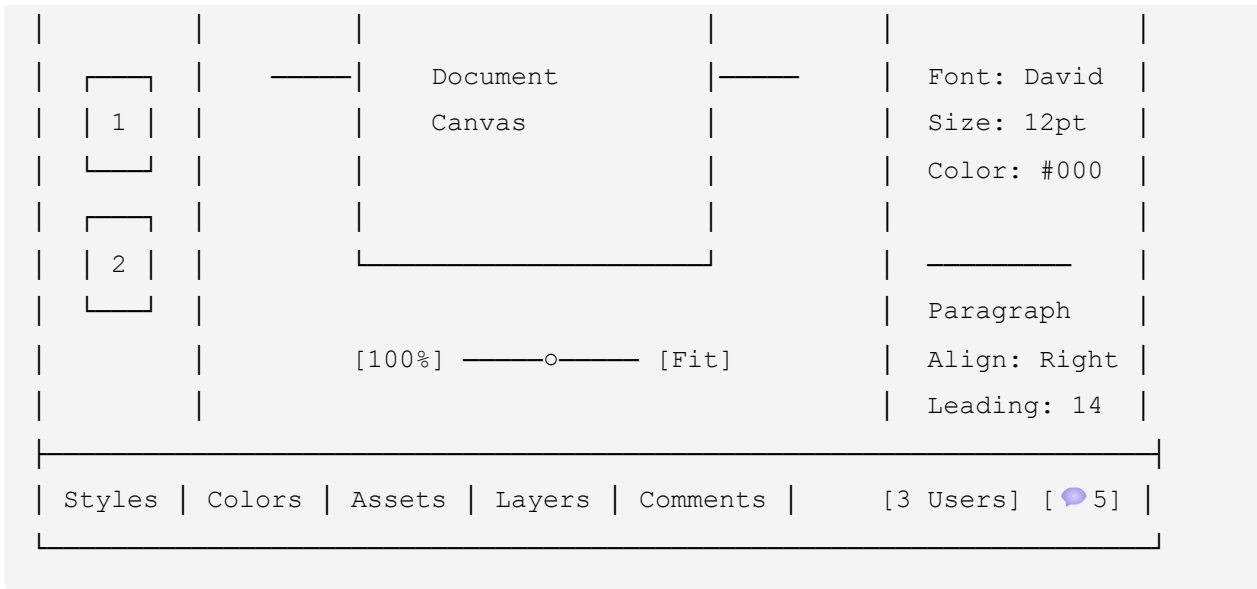
| עדכונים | ידני | אוטומטי |

| Templates | מוגבל | גלריה עשירה |

| Undo | מוגבל | אינסופי + History |

ממשק משתמש





API Structure

```
// Document Model
interface Document {
  id: string;
  name: string;
  pages: Page[];
  styles: StyleLibrary;
  codes: CodeLibrary;           // CODE System (from mishna.sty)
  sets: FrameSet[];           // Linked Frames Sets (from Berech)
  groups: FrameGroup[];       // Groups of Sets
  assets: Asset[];
  metadata: DocumentMetadata;
}

interface Page {
  id: string;
  size: PageSize;
  margins: Margins;
  frames: Frame[];
  masterPageId?: string;
  isOddPage: boolean;         // For swap/mirror layouts
}

interface Frame {
  id: string;
```

```

type: 'text' | 'image' | 'shape';
position: Position;
size: Size;
rotation: number;
content: TextContent | ImageContent | ShapeContent;
linkedTo?: string;           // Linked text frames
formats: FrameFormat[];     // Multi-format support
ramah: number;              // Hierarchy level (0, 1, 2)
}

// From Berech.prg - Linked Frames System
interface FrameSet {
  id: string;
  name: string;
  textIds: number[];        // Frame IDs in set
  formatIndices: number[];  // Format index per frame
  boxIndices: number[];    // Box index per frame
  swap: boolean;           // Mirror on even pages
  all: boolean;            // Require all frames
  minSpace: number;       // Minimum space between frames
}

interface FrameGroup {
  id: string;
  name: string;
  sets: string[];          // Set IDs or ranges ("1-5")
}

// From mishna.sty - CODE System
interface CodeDefinition {
  id: string;               // e.g., "01", "11"
  font?: number;           // FT{n}
  bold?: boolean;          // BO/NO
  italic?: boolean;        // IT
  size?: number;           // SZ{n}
  leading?: number;        // LD{n}
  justify?: 'full' | 'left' | 'right' | 'center';
  color?: string;          // CLR{nnn}
  lineLength?: string;     // ll{n}cm
  columns?: number;        // col{n}
  superscript?: number;    // SUP{n}
}

```

```

    subscript?: number;          // SUB{n}
}

// From mishna.sty - Format definitions
interface FrameFormat {
    id: string;
    grid: [number, number];      // [x, y] grid
    columns?: string;           // tur2 - second column width
    rightPage: FormatSide;       // Settings for right (odd) pages
    leftPage: FormatSide;       // Settings for left (even) pages
    divider?: DividerSettings;
    border?: BorderSettings;
}

interface FormatSide {
    x: number;
    leftIndents: number[];      // ltab_r/ltab_l - up to 4 levels
    rightIndents: number[];     // rtab_r/rtab_l
    shape: FrameShape;          // tsura_r/tsura_l
    flow: boolean;              // zerem_r/zerem_l
}

interface FrameShape {
    width: string;               // e.g., "12cm"
    height: number;              // in points
    leftIndent: string;          // e.g., "6.2cm"
    rightIndent: string;         // e.g., "0"
}

// Auto-Indent calculation (from Berech_f algorithm)
interface DepthInfo {
    topDP: number;               // Top depth
    boxDP: number;               // Box height
    totalDP: number;             // Total depth (top + height)
    kotDP: number;               // Header depth
    contDP: number;              // Continuation depth
    grid: number;                // Y grid
}

interface IndentElement {
    startDepth: number;
}

```

```

endDepth: number;
leftIndent: number;
rightIndent: number;
type: 'top' | 'bottom';
}

// =====
// RTA Engine Interfaces (from RTA.LIB bytecode analysis)
// =====

// LinkCalc Engine (32 functions from RTAID.LIB)
interface LinkTable {
  id: string;
  links: Link[];
  kvutsaTextIds: string[];
  textInfo: TextInfo;
  theoShetach: number;           // Theoretical area
  lastFrame: string;           // LC_LastFrame
}

interface Link {
  sourceFrameId: string;
  targetFrameId: string;
  type: 'single' | 'multi' | 'reverse';
  indents: number[];           // LC_indents
  just: 'left' | 'right' | 'center' | 'full'; // LC_just
}

interface LCConfig {
  iConst: number;             // LC_iConst
  sConst: string;           // LC_sConst
  theo: number;             // LC_theo
  newpage: boolean;         // LC_newpage
  info: LCInfo;             // LC_info
}

// MultiLinks System (from RTAID.LIB, RTAID2.LIB)
interface MultiLinksConfig {
  links: MultiLink[];
  autoFix: boolean;         // MultiLinks_fix
}

```

```

interface MultiLink {
  id: string;
  frameIds: string[];          // Multiple frames linked
  mode: 'all' | 'single' | 'main'; // MultiLinks_a/s/m
}

// Hatem Engine - Signatures (16 functions from RTAID1.LIB)
interface HatemConfig {
  type: 'x' | 's' | 't' | 'm'; // hatem_x/s/t/m
  list: HatemItem[];          // hatem_list
  theo: number;                // hatem_theo_d
  levels: HatemLevel[];       // 0, 1, 2 levels
}

interface HatemItem {
  id: string;
  size: number;                // MarkSize
  font: string;                // MarkFont
  position: Position;
  minMax: [number, number];    // MarkMinMax
  cut: number;                 // MarkCut
}

interface HatemLevel {
  level: number;               // 0, 1, 2
  items: HatemItem[];
}

// Hatzmed Engine - Attachments (12 functions from RTAID1.LIB)
interface HatzmedConfig {
  type: 'a' | 'e' | 'x' | 'm'; // hatzmed_a/e/x/m
  list: HatzmedItem[];         // hatzmed_list
  yk: boolean;                 // Hatzmed_YK (right/extreme)
}

interface HatzmedItem {
  id: string;
  sourceFrameId: string;
  targetFrameId: string;
  attachPoint: 'top' | 'bottom' | 'left' | 'right';
}

```

```

    offset: number;
}

// Marks System (from RTA.LIB, RTAID2.LIB)
interface MarksConfig {
    file: string;           // CreateMarksFile
    marks: Mark[];
    mainSize: number;      // MarkMainSize
    mainFont: string;     // MarkMainFont
}

interface Mark {
    id: string;
    size: number;         // MarkSize
    font: string;        // MarkFont
    cut: number;         // MarkCut
    position: Position;
}

// Process Management (from all LIB files)
interface ProcessState {
    isProcessing: boolean; // InProcess_d
    processType: 'document' | 'plugin';
    updatePending: boolean; // dUpdatePlug
}

// TextID System (from RTA1.LIB, RTAID1.LIB)
interface TextIDConfig {
    textIds: TextID[];
    kvutsaSorted: string[]; // kvutsa_sorted
    textSorted: string[]; // text_sorted
}

interface TextID {
    id: string;
    name: string;
    format: string;        // dlg_formatik
    button: string;       // dlg_kaftor
    iniPath: string;      // GetIniFile_TextID
}

```

```
// AutoRTA Engine (from RTA.LIB)
interface AutoRTAConfig {
    enabled: boolean;           // AutoRTA
    stiya: number;              // SetStiya/GetStiya
    autoStiya: boolean;        // AutoStiya
    theoShetach: number;       // Theo_Shetach
    roundStiya: boolean;       // RoundStiya
    autoIzun: boolean;         // Auto_Izun
}

// FindBreak Engine (from RTAID3.LIB)
interface FindBreakConfig {
    initialized: boolean;      // FindBreak_init
    borderUse: boolean;        // border_use
    just0: 'left' | 'right' | 'center'; // just_0
}

interface BreakPoint {
    position: number;
    type: 'word' | 'line' | 'page';
    frameId: string;
}

// Speed Settings (from multiple files)
interface SpeedConfig {
    speed: number;             // speed_f
    func: string;              // speed_func
}

// Dialog Positions (from INI files)
interface DialogPosition {
    name: string;              // DlogName
    x: number;
    y: number;
    width: number;             // DialogWidth
    height: number;            // DialogHeight_d
    plusMinus: number;         // DialogPlusMinus
}
```

Database Schema (Prisma)

```
model Document {
  id          String    @id @default(cuid())
  name        String
  content     Json      // Full document JSON
  ownerId     String
  createdAt   DateTime @default(now())
  updatedAt   DateTime @updatedAt

  owner       User      @relation(fields: [ownerId], references: [id])
  versions    Version[]
  comments    Comment[]
  styles      Style[]
  codes       Code[]
  sets        FrameSet[]
  groups      FrameGroup[]
}

model User {
  id          String    @id @default(cuid())
  email       String    @unique
  name        String
  documents   Document[]
}

model Version {
  id          String    @id @default(cuid())
  documentId  String
  content     Json
  createdAt   DateTime @default(now())
  createdBy   String

  document    Document @relation(fields: [documentId], references: [id])
}

// CODE System (from mishna.sty)
model Code {
  id          String    @id @default(cuid())
  documentId  String
```

```

codeId      String    // e.g., "01", "11"
definition  Json       // CodeDefinition object

document    Document @relation(fields: [documentId], references: [id])

@@unique([documentId, codeId])
}

// Linked Frames System (from Berech.prg)
model FrameSet {
  id          String    @id @default(cuid())
  documentId  String
  name        String
  textIds     Int[]     // Array of frame IDs
  formatIds   Int[]     // Format index per frame
  boxIds      Int[]     // Box index per frame
  swap        Boolean  @default(false)
  all         Boolean  @default(false)
  minSpace    Float    @default(0)

  document    Document @relation(fields: [documentId], references: [id])
  groups      FrameGroup[] @relation("SetGroups")
}

model FrameGroup {
  id          String    @id @default(cuid())
  documentId  String
  name        String

  document    Document @relation(fields: [documentId], references: [id])
  sets        FrameSet[] @relation("SetGroups")
}

// Style definitions
model Style {
  id          String    @id @default(cuid())
  documentId  String
  name        String
  type        StyleType
  definition  Json     // Full style definition
  basedOn    String?   // Parent style ID
}

```

```

nextStyle String? // Next style after Enter

document Document @relation(fields: [documentId], references: [id])

@@unique([documentId, name])
}

enum StyleType {
    PARAGRAPH
    CHARACTER
    TABLE
    OBJECT
}

// User settings per document (from INI files)
model DocumentSettings {
    id String @id @default(cuid())
    documentId String @unique
    dialogXY Json? // Dialog positions
    lastSetId Int?
    lastGroupId Int?
    colorScheme String?
    crossEnabled Boolean @default(true)
    tabsEnabled Boolean @default(true)
}

```

סדר עבודה מוצע

(1-4 שבועות) Phase 1: Foundation

- Setup monorepo with Turborepo
- Design system and UI components
- Basic canvas with Fabric.js
- Text editing with Lexical
- Document model and persistence

(5-8 שבועות) Phase 2: Core Features

- Page management

- Typography controls
- Image handling
- Style system
- PDF export

!Berech - חדש מ- (שבועות 9-12) Phase 3: Advanced Layout

- **CODE System**: קודי עיצוב מהירים
- **Linked Frames**: מסגרות מקושרות
- **Sets & Groups**: ניהול סטים וקבוצות
- **Auto-Indent Engine**: חישוב Indents אוטומטי
- **Reflow Engine**: זרימת טקסט מחדש
- **Multi-Format Frames**: פורמטים מרובים למסגרת
- **Mirror Layouts**: פריסה הפוכה לדפים זוגיים

(שבועות 13-16) Phase 4: Collaboration

- WebSocket infrastructure
- Y.js CRDT integration
- Multi-cursor support
- Comments system
- Version history

(שבועות 17-20) Phase 5: AI & Polish

- AI features integration
- Template gallery
- Performance optimization
- Accessibility (A11y)
- RTL perfection

קבצים מרכזיים ליצירה

Core Components

- .1 `apps/web/src/components/editor/Canvas.tsx` - הקנבס הראשי
- .2 `apps/web/src/components/editor/TextFrame.tsx` - מסגרת טקסט
- .3 `apps/web/src/stores/documentStore.ts` - State Management
- .4 `apps/api/src/routes/documents.ts` - API Routes

`packages/document-model/src/schema.ts` - Document Schema .5

`packages/collab/src/sync.ts` - CRDT Sync .6

(Berech.prg-מ) Advanced Layout

- מקושרות מסגרות - `packages/layout-engine/src/linkedFrames.ts` .1
- מנוע Indent אוטומטי - `packages/layout-engine/src/autoIndent.ts` .2
- מנוע זרימה מחדש - `packages/layout-engine/src/reflow.ts` .3
- ניהול סטים וקבוצות - `packages/layout-engine/src/sets.ts` .4
- חישוב עומקים - `packages/layout-engine/src/depthCalculator.ts` .5

(mishna.sty-מ) CODE System

- פרסר לקודי עיצוב - `packages/code-system/src/parser.ts` .1
- יישום קודים על טקסט - `packages/code-system/src/applier.ts` .2
- ספריית קודים - `packages/code-system/src/library.ts` .3

Format System

- פורמטי מסגרת - `packages/format-system/src/frameFormat.ts` .1
- פריסה הפוכה - `packages/format-system/src/mirrorLayout.ts` .2
- לרשת `packages/format-system/src/gridSnap.ts` - Snap .3

UI Components

- דיאלוג ניהול סטים - `apps/web/src/components/dialogs/SetDialog.tsx` .1
 - דיאלוג קבוצות - `apps/web/src/components/dialogs/GroupDialog.tsx` .2
 - עורך קודים - `apps/web/src/components/dialogs/CodeEditor.tsx` .3
 - פאנל מסגרות - `apps/web/src/components/panels/LinkedFramesPanel.tsx` .4
- מקושרות

אלגוריתמים מרכזיים (מ-Berech.prg)

Auto-Indent Algorithm

```
/**
 * אוטומטי למסגרות חופפות Indents חישוב
 * Based on Berech_f() from BERECH.PRG
 */
```

```

function calculateAutoIndents(frames: Frame[], pageNumber: number): IndentRes
  const results: IndentResult[] = [];

  // 1. איסוף מידע על כל המסגרות
  const depthInfo = frames.map(f => getDepthInfo(f));

  // 2. חישוב מטריצת חפיפות
  const overlaps: boolean[][] = [];
  for (let i = 0; i < frames.length; i++) {
    overlaps[i] = [];
    for (let j = 0; j < frames.length; j++) {
      if (i === j) continue;
      overlaps[i][j] =
        depthInfo[i].topDP < depthInfo[j].totalDP &&
        depthInfo[i].totalDP > depthInfo[j].topDP;
    }
  }

  // 3. לכל מסגרת Indents חישוב
  for (let i = 0; i < frames.length; i++) {
    const elements: IndentElement[] = [];
    const depthPoints: number[] = [];

    // עבור על מסגרות חופפות
    for (let j = 0; j < frames.length; j++) {
      if (!overlaps[i][j]) continue;

      // Top indent (מסגרת מתחילה מעל)
      if (depthInfo[i].topDP < depthInfo[j].topDP) {
        const indent = calculateIndentAmount(i, j, frames);
        elements.push({
          startDepth: depthInfo[i].topDP,
          endDepth: depthInfo[j].topDP,
          leftIndent: j < i ? indent : 0,
          rightIndent: j > i ? indent : 0,
          type: 'top'
        });
        depthPoints.push(depthInfo[j].topDP);
      }

      // Bottom indent (מסגרת מסתיימת למטה)
    }
  }

```

```

    if (depthInfo[i].totalDP > depthInfo[j].totalDP) {
      elements.push({
        startDepth: depthInfo[j].totalDP,
        endDepth: depthInfo[i].totalDP,
        leftIndent: 0,
        rightIndent: 0,
        type: 'bottom'
      });
      depthPoints.push(depthInfo[j].totalDP);
    }
  }

  // 4. מיון נקודות עומק
  depthPoints.sort((a, b) => a - b);

  // 5. יצירת חטורת מדורגת
  results.push({
    frameId: frames[i].id,
    elements,
    depthPoints
  });
}

return results;
}

/**
 * עיגול לרשת
 * Based on RoundToGrid() from BERECH.PRG
 */
function roundToGrid(value: number, grid: number, roundDown: boolean): number {
  if (!grid || !value) return 0;
  if (roundDown) {
    return -(value % grid);
  }
  return (grid - (value % grid)) % grid;
}

/**
 * חישוב הפרש עומק
 * Based on GetDepthDiff() from BERECH.PRG

```

```

*/
function getDepthDiff(a: number, b: number, minSpace: number, grid: number, c
  let diff = a - b + minSpace;
  return diff + roundToGrid(diff, grid, down);
}

```

Reflow Algorithm

```

/**
 * זרימת טקסט מחדש למסגרת שהשתנתה
 * Based on reflow logic from Berech
 */
function reflowText(frame: Frame, indents: IndentElement[]): ReflowResult {
  // 1. חישוב צורת המסגרת החדשה
  const shape = calculateFrameShape(frame, indents);

  // 2. חלוקת הטקסט לאלמנטים
  const elements = splitTextByDepth(frame.content, indents);

  // 3. חישוב שבירות שורה מחדש
  const lineBreaks = calculateLineBreaks(elements, shape);

  // 4. טיפול בהמשכים
  const continuations = handleContinuations(frame, lineBreaks);

  return { shape, elements, lineBreaks, continuations };
}

```

Set Execution Algorithm

```

/**
 * הפעלת סט על עמוד
 * Based on Berech_set() from BERECH.PRG
 */
function executeSet(set: FrameSet, page: Page): void {
  // 1. קריאת הגדרות הסט
  const { textIds, formatIds, boxIds, swap, minSpace } = set;

```

```

// 2. החלפת צדדים לדפים זוגיים
const effectiveIds = page.isOddPage || !swap
  ? textIds
  : textIds.slice().reverse();

// 3. איסוף מידע על כל המסגרות
const frameInfos = effectiveIds.map((id, i) =>
  getFrameInfo(page, id, formatIds[i], boxIds[i])
);

// 4. חישוב Indents
const indents = calculateAutoIndents(frameInfos, page.number);

// 5. עדכון המסגרות
for (let i = 0; i < frameInfos.length; i++) {
  applyIndents(frameInfos[i], indents[i]);
  reflowText(frameInfos[i]);
}
}

```

הערות חשובות

1. **RTL-First**: כל הקוד ייכתב עם תמיכה מלאה ב-RTL מההתחלה
2. **Accessibility**: עמידה בתקני WCAG 2.1 AA
3. **Performance: Target** של 60fps בזמן עריכה
4. **Offline-First**: עבודה אופליין עם סנכרון אוטומטי
5. **Type-Safety**: TypeScript strict mode בכל הקוד
6. **Berech Compatibility**: תמיכה בייבוא קבצי INI מ-TAG המקורי
7. **CODE Import**: תמיכה בייבוא קבצי STY מ-TAG המקורי