---

name: Hebrew OCR System overview: מערכת זיהוי כתב יד בעברית עם Next.js, Express, Firebase ומודל ML היברידי. תומכת בזיהוי אותיות, מילים ושורות עם אימון שיתופי todos:

- id: setup-monorepo content: הקמת מבנה פרויקט - frontend, backend, ml-service status: pending
- id: firebase-setup content: הגדרת Firebase - Firestore, Storage, Auth status: pending
- id: backend-api content: בניית Express API עם routes/controllers/services status: pending
- id: frontend-upload content: ממשק העלאת תמונות ב-Next.js status: pending
- id: image-enhancement content: שלב שיפור וחידוד תמונות עם תצוגה בזמן אמת status: pending
- id: segmentation-service content: שירות Python לחלוקת תמונות (שורות/מילים/אותיות) status: pending
- id: annotation-interface content: ממשק תיוג אינטראקטיבי status: pending
- id: character-model content: מודל זיהוי אותיות (EfficientNet) status: pending
- id: word-model content: מודל זיהוי מילים (CRNN) status: pending
- id: line-model content: מודל זיהוי שורות (TrOCR) status: pending
- id: model-fusion content: שילוב תוצאות מכל המודלים status: pending
- id: training-pipeline content: צינור אימון מתמשך status: pending
- id: deployment content: פריסה ופרסום ב-GitHub status: pending

---

מערכת זיהוי כתב יד בעברית

ארכיטקטורה כללית

```
Parse error on line 2:
...  subgraph frontend [Frontend - Next.js]
----------------------^
Expecting 'SEMI', 'NEWLINE', 'SPACE', 'EOF', 'GRAPH', 'DIR',
'TAGEND', 'TAGSTART', 'UP', 'DOWN', 'subgraph', 'end', 'SQE',
'PE', '-)', 'DIAMOND_STOP', 'MINUS', '--', 'ARROW_POINT',
'ARROW_CIRCLE', 'ARROW_CROSS', 'ARROW_OPEN',
'DOTTED_ARROW_POINT', 'DOTTED_ARROW_CIRCLE',
'DOTTED_ARROW_CROSS', 'DOTTED_ARROW_OPEN', '==',
'THICK_ARROW_POINT', 'THICK_ARROW_CIRCLE', 'THICK_ARROW_CROSS',
'THICK_ARROW_OPEN', 'PIPE', 'STYLE', 'LINKSTYLE', 'CLASSDEF',
'CLASS', 'CLICK', 'DEFAULT', 'NUM', 'PCT', 'COMMA', 'ALPHA',
'COLON', 'BRKT', 'DOT', 'PUNCTUATION', 'UNICODE_TEXT', 'PLUS',
'EQUALS', 'MULT', got 'SQS'
```

מבנה תיקיות

```
hebrew-ocr/
├── frontend/                 # Next.js App
│   ├── src/
│   │   ├── app/              # App Router
│   │   ├── components/       # React Components
│   │   │   ├── upload/       # Upload components
│   │   │   ├── annotation/   # Annotation interface
│   │   │   └── common/       # Shared components
│   │   ├── hooks/            # Custom hooks
│   │   ├── services/         # API calls
│   │   └── types/            # TypeScript types
│   └── package.json
│
├── backend/                  # Express API
│   ├── src/
│   │   ├── routes/           # API routes
│   │   ├── controllers/      # Request handlers
│   │   ├── services/         # Business logic
│   │   ├── middleware/       # Auth, validation
│   │   └── config/           # Firebase config
│   └── package.json
│
├── ml-service/               # Python ML Microservice
│   ├── models/               # Trained models
│   ├── services/
│   │   ├── segmentation.py   # Image segmentation
│   │   ├── character_ocr.py  # Character recognition
│   │   ├── word_ocr.py       # Word recognition
│   │   └── line_ocr.py       # Line recognition
│   ├── training/             # Training scripts
│   └── requirements.txt
│
└── shared/                   # Shared types/utils
```

שלב 1: הקמת תשתית בסיסית

1.1 Backend - Express + Firebase

- הגדרת Firebase Admin SDK
- מבנה API בסיסי עם routes/controllers/services
- Middleware לאימות ו-rate limiting
- Upload endpoint לתמונות

1.2 Frontend - Next.js

- הגדרת Firebase Client SDK
- ממשק העלאת תמונות עם drag & drop
- תצוגת תמונות שהועלו

1.3 Firebase Setup

- Firestore collections: `users`, `images`, `annotations`, `training_data`
- Storage buckets: `uploads/`, `processed/`, `models/`
- Security rules

שלב 2: שיפור וחידוד תמונות (CRITICAL)

שלב קריטי לפני אימון - שיפור איכות התמונות עם תצוגה בזמן אמת.

2.1 Image Enhancement Pipeline

```
Parse error on line 1:
flowchart LR    Ori
^
Expecting 'NEWLINE', 'SPACE', 'GRAPH', got 'ALPHA'
```

2.2 פילטרים זמינים

| פילטר | תיאור | פרמטרים |

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

| Grayscale | המרה לגווני אפור | - |

| Denoise | הסרת רעש | strength: 1-10 |

| Contrast | שיפור ניגודיות | CLAHE clipLimit: 1-5 |

| Binarization | המרה לשחור-לבן | method: Otsu/Adaptive, blockSize, C |

| Deskew | תיקון הטיה | auto-detect angle |

| Sharpen | חידוד | kernel size, strength |

| Morphology | פעולות מורפולוגיות | dilate/erode, kernel size |

2.3 ממשק שיפור תמונות (Real-time)

```
interface EnhancementSettings {
  grayscale: boolean;
  denoise: {
    enabled: boolean;
    strength: number; // 1-10
  };
  contrast: {
    enabled: boolean;
    clipLimit: number; // 1-5
    tileSize: number; // 8-16
  };
  binarization: {
    enabled: boolean;
    method: 'otsu' | 'adaptive' | 'sauvola';
    blockSize: number; // 11-51 (odd)
    constant: number; // 2-20
  };
  deskew: {
    enabled: boolean;
    maxAngle: number; // max rotation degrees
  };
  sharpen: {
    enabled: boolean;
    strength: number; // 0.5-3
  };
}

interface EnhancementPreview {
  originalImage: string; // base64
  enhancedImage: string; // base64
  settings: EnhancementSettings;
  processingTime: number; // ms
  qualityScore: number; // 0-100
}
```

2.4 תכונות ממשק המשתמש

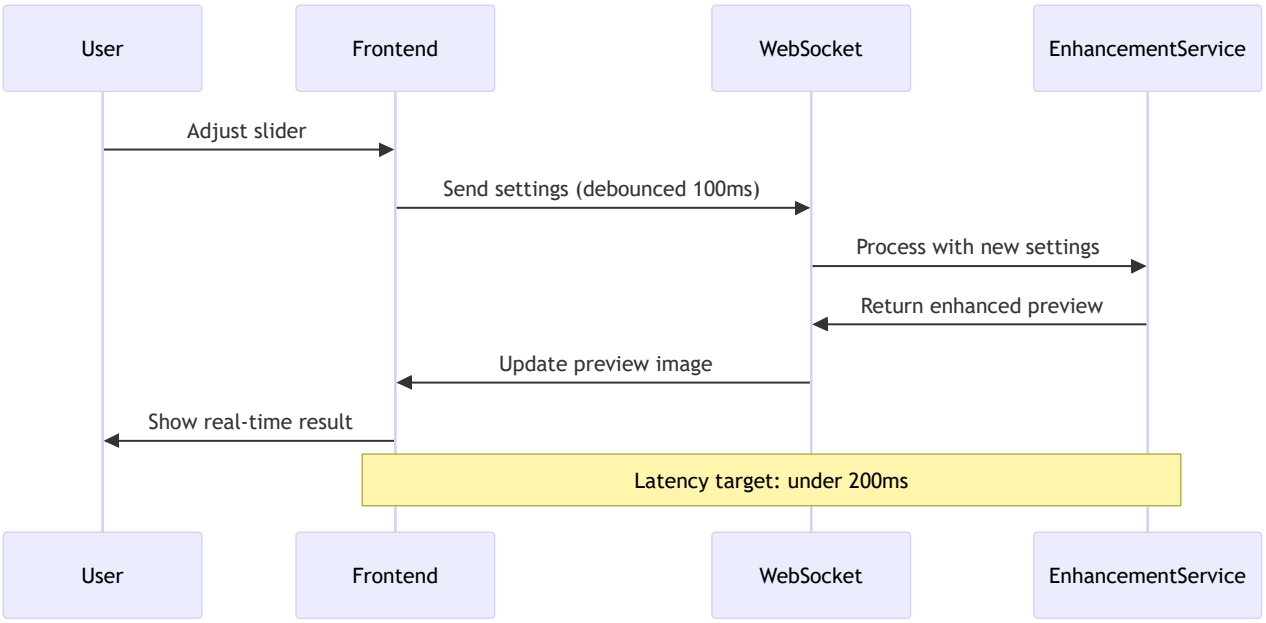
- **תצוגה בזמן אמת**: כל שינוי בפרמטר מעדכן את התצוגה מיידית
- **השוואה Side-by-Side**: תמונה מקורית לצד תמונה משופרת

- **Slider comparison**: גרירה להשוואה על אותה תמונה
- **Zoom & Pan**: התקרבות לפרטים קטנים
- **Undo/Redo**: חזרה לשלבים קודמים
- **Presets**: הגדרות מוכנות מראש לסוגי כתב יד שונים
- **Quality Score**: ציון איכות אוטומטי לתמונה

2.5 Presets מוכנים

```
const ENHANCEMENT_PRESETS = {
  light_ink: {
    name: 'דיו בהיר',
    settings: { contrast: { clipLimit: 4 }, binarization: { constant: 5 } }
  },
  dark_background: {
    name: 'רקע כהה',
    settings: { binarization: { method: 'adaptive', blockSize: 31 } }
  },
  faded_document: {
    name: 'מסמך דהוי',
    settings: { contrast: { clipLimit: 5 }, sharpen: { strength: 2 } }
  },
  noisy_scan: {
    name: 'סריקה רועשת',
    settings: { denoise: { strength: 8 }, binarization: { method: 'sauvola' } }
  },
  tilted_image: {
    name: 'תמונה מוטה',
    settings: { deskew: { enabled: true, maxAngle: 15 } }
  }
};
```

2.6 WebSocket לתצוגה בזמן אמת



2.7 API Endpoints לשיפור תמונות

| Method | Endpoint | תיאור |

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

| POST | `/api/enhance/preview` | קבלת תצוגה מקדימה עם הגדרות |

| POST | `/api/enhance/apply` | החלת שיפורים ושמירה |

| GET | `/api/enhance/presets` | קבלת רשימת presets |

| POST | `/api/enhance/auto` | שיפור אוטומטי מומלץ |

| GET | `/api/enhance/quality/:imageId` | ציון איכות לתמונה |

2.8 שמירת הגדרות לכל תמונה

```
interface ImageEnhancementDoc {
  imageId: string;
  originalPath: string;
  enhancedPath: string;
  settings: EnhancementSettings;
  qualityBefore: number;
  qualityAfter: number;
  appliedAt: Timestamp;
}
```

שלב 3: עיבוד תמונות וסגמנטציה

3.1 Image Processor

- קבלת תמונה וחיתוך אוטומטי
- זיהוי שורות (Line Detection)
- זיהוי מילים בתוך שורות (Word Segmentation)
- זיהוי אותיות בתוך מילים (Character Segmentation)

```
Parse error on line 1:
flowchart LR    Ima
^
Expecting 'NEWLINE', 'SPACE', 'GRAPH', got 'ALPHA'
```

3.2 מבנה נתונים ב-Firestore

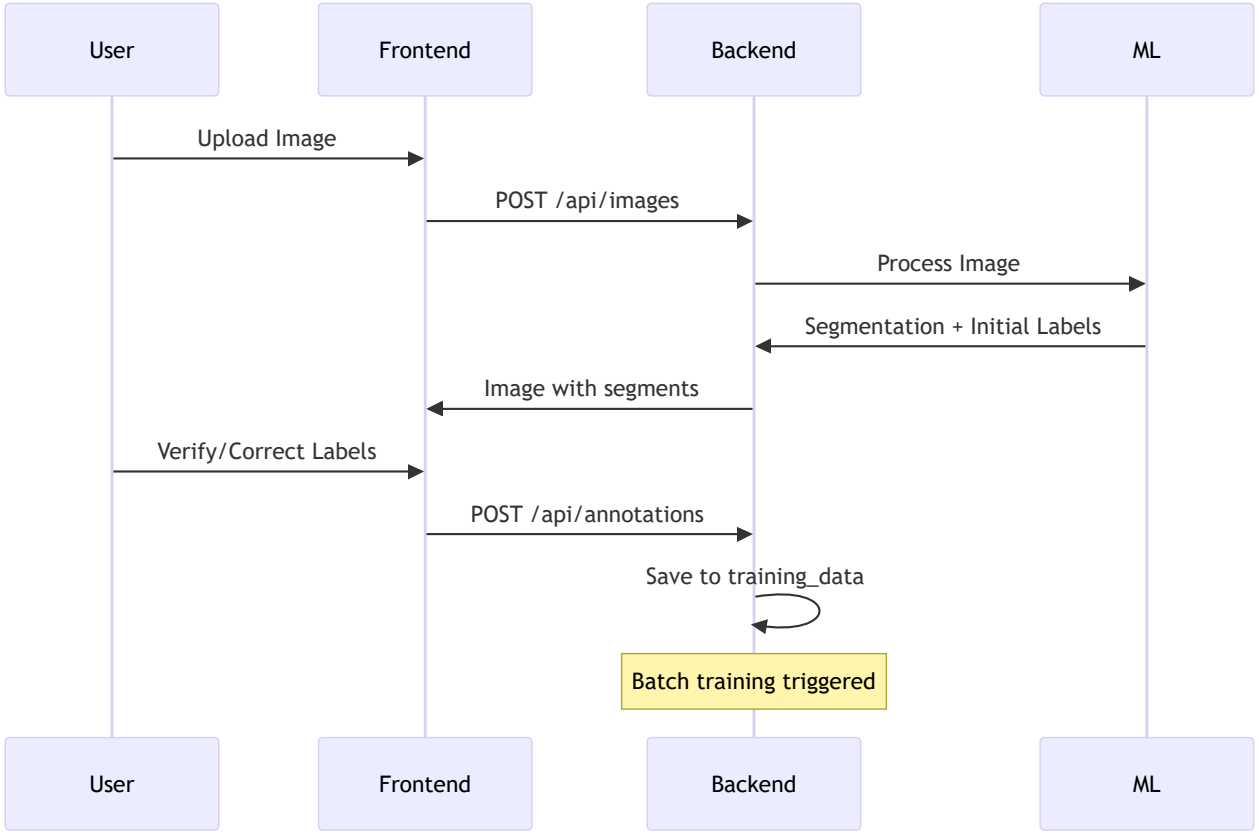
```
// Image document
interface ImageDoc {
  id: string;
  userId: string;
  storagePath: string;
  status: 'uploaded' | 'processing' | 'ready' | 'annotated';
  segmentation: {
    lines: LineSegment[];
    words: WordSegment[];
    characters: CharSegment[];
  };
  createdAt: Timestamp;
}

// Annotation document
interface AnnotationDoc {
  id: string;
  imageId: string;
  userId: string;
  type: 'character' | 'word' | 'line';
  boundingBox: BoundingBox;
  label: string;
  confidence: number;
  isVerified: boolean;
  createdAt: Timestamp;
}
```

שלב 4: ממשק תיוג (Annotation Interface)

4.1 תכונות הממשק

- תצוגת תמונה עם bounding boxes
- מעבר בין מצבי תיוג: אותיות / מילים / שורות
- מקלדת וירטואלית בעברית
- אישור/תיקון תיוג אוטומטי
- התקדמות ומשוב למשתמש

4.2 עבודה Flow



שלב 5: מודלים היברידיים

5.1 Character Model (EfficientNet-based)

- Fine-tuned EfficientNet-B0
- 27 אותיות + סימני פיסוק

- אימון על אותיות מבודדות

5.2 Word Model (CRNN)

- CNN לחילוץ features
- RNN (LSTM) לסדר אותיות
- CTC Loss לאימון

5.3 Line Model (TrOCR / Vision Transformer)

- Pre-trained transformer
- Fine-tuning על שורות בעברית
- תמיכה בהקשר מלא

5.4 שילוב המודלים

```
Parse error on line 1:
flowchart TB    Inp
^
Expecting 'NEWLINE', 'SPACE', 'GRAPH', got 'ALPHA'
```

שלב 6: אימון מתמשך

6.1 Training Pipeline

- איסוף annotations מאומתות
- יצירת training batches
- אימון incremental למודלים
- שמירת model versions

5.2 Metrics Dashboard

- דיוק לפי סוג (אות/מילה/שורה)
- כמות נתונים מתויגים

- התקדמות אימון

API Endpoints

| Method | Endpoint | תיאור |

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

| POST | `/api/auth/register` | הרשמת משתמש |

| POST | `/api/images/upload` | העלאת תמונה |

| GET | `/api/images/:id` | קבלת תמונה עם segmentation |

| POST | `/api/annotations` | שמירת תיוג |

| GET | `/api/annotations/:imageId` | קבלת תיוגים לתמונה |

| POST | `/api/ocr/recognize` | זיהוי טקסט בתמונה |

| GET | `/api/stats` | סטטיסטיקות מערכת |

סדר עבודה מומלץ

1. **שבוע 1**: הקמת תשתית בסיסית (Firebase, Express, Next.js)
2. **שבוע 2**: Upload flow + Image storage
3. **שבוע 3**: Segmentation service (Python)
4. **שבוע 4**: Annotation interface
5. **שבוע 5**: Character model + training
6. **שבוע 6**: Word/Line models
7. **שבוע 7**: Model fusion + API
8. **שבוע 8**: Testing, optimization, deployment