## **אפיון מערכת MedSync**

## תיאור כללי

## MedSync היא אפליקציה Saas לניהול תיק רפואי דיגיטלי אישי. המערכת מאפשרת למשתמשים להעלות מסמכים רפואיים, לבצע פענוח חכם של המסמכים, לזהות מצבים רפואיים הדורשים טיפול ולקבוע תורים לרופאים מתאימים. המידע מסונכרן בין המטופל לרופא, ומאפשר תהליך רפואי רציף ומבוסס נתונים.

## 1.1 מטרת הפרויקט

## לאפשר למטופלים לנהל מסמכים רפואיים בצורה נוחה ומאובטחת.

## לאפשר לרופאים לנהל מטופלים, לצפות בתיקים רפואיים ולתאם תורים.

## לשפר את נגישות המידע ולייעל את תהליך הזימון.

## לספק פענוח מסמכים רפואיים בעזרת AI לצורך ניתוח רפואי חכם.

## להוריד עומס ניהולי ממוסדות רפואיים באמצעות אוטומציה של תהליכים.

## 1.2 קהל היעד

## מטופלים הזקוקים למעקב רפואי ושמירת מסמכים.

## רופאים ומרכזים רפואיים המנהלים פגישות ומסמכים רפואיים.

## גופי ביטוח וחברות רפואיות הדורשות גישה למסמכים וניתוח נתונים.

## מוסדות רפואיים המעוניינים לשפר את ניהול התורים והמסמכים.

## 1.3 פונקציונליות של המערכת

## העלאת מסמכים רפואיים ושמירתם בענן.

## חיפוש מסמכים לפי תגים ומילות מפתח.

## פענוח מסמכים רפואיים באמצעות OCR ו-AI לזיהוי טקסט ותמצות מידע חשוב.

## יצירת סיכום רפואי אוטומטי ממסמכים שהועלו.

## הצגת בעיות רפואיות פוטנציאליות שהתגלו במסמכים ושאילת המטופל אם ברצונו לקבוע תור לרופא.

## התאמת רשימת הרופאים לפי התמחות ובעיות שזוהו במסמך הרפואי.

## ניהול תורים וזימון רופאים על בסיס התמחות וזמינות.

## לאחר ביקור רופא, העלאת טופס ביקור רפואי לתיק הרפואי של המטופל.

## גישה של רופאים לתיק הרפואי של המטופל בעת קביעת תור.

## שליחת התראות למטופלים על בדיקות עתידיות.

## הפקת דוח חודשי המסכם את הפעילות, הבעיות שהתגלו, והבדיקות המומלצות.

## שיתוף מאובטח של מסמכים רפואיים עם רופאים מורשים.

## הפקת דוחות רפואיים מותאמים אישית.

## ממשק ניהול מתקדם למנהלים לצורך פיקוח על המערכת.

## שילוב מנוע המלצות רופאים המבוסס על היסטוריית המטופל.

## 1.4 בעיות שהמערכת פותרת

## ריכוז כל המידע הרפואי של המטופל במקום אחד ונגיש מכל מקום.

## גישה מאובטחת ושיתוף מסמכים בצורה נוחה ומהירה.

## שיפור יעילות קביעת תורים מול רופאים.

## ביטול הצורך בניירת רפואית פיזית ושיפור תהליכי העבודה.

## צמצום טעויות רפואיות באמצעות ניתוח אוטומטי של מסמכים רפואיים.

## מנגנון המלצות חכם לרופאים ולמטופלים על בדיקות נדרשות.

## הקטנת זמני המתנה למטופלים באמצעות אופטימיזציה חכמה של התורים.

## 1.5 טכנולוגיות בשימוש

## Backend: .NET Core לניהול API.

## Frontend: React לניהול קבצים ותורים, Angular לניהול משתמשים והרשאות.

## Database: PostgreSQL / MongoDB.

## Cloud Storage: AWS S3 / Firebase.

## AI & OCR: שימוש במודלים של AI לניתוח מסמכים רפואיים וזיהוי טקסט.

## Security: JWT, OAuth2, הצפנת נתונים.

## CI/CD: GitHub Actions לפריסה אוטומטית.

## 1.6 מתודולוגיית פיתוח (Agile)

## עבודה לפי ספרינטים בני שבוע.

## שימוש ב-Jira / Trello לניהול משימות.

## בדיקות איכות והפצות מדורגות.

## שילוב CI/CD לפריסה מהירה.

## שימוש במתודולוגיית TDD לבדיקת רכיבים מרכזיים.

## עיצוב ממשק משתמש ויזואלי (Wireframes)

## 📌 מסכים מרכזיים:

## מסך התחברות/הרשמה – הזדהות מאובטחת.

## מסך ניהול קבצים רפואיים – הצגת מסמכים, העלאה, מחיקה ושיתוף.

## מסך פענוח מסמך רפואי – הצגת OCR, ניתוח AI ועריכה ידנית.

## מסך הצגת בעיות רפואיות – זיהוי בעיות רפואיות במסמך, הצעת קביעת תור.

## מסך חיפוש רופאים וקביעת תור – התאמת רופאים לפי התמחות ובעיות שהתגלו במסמך.

## מסך ניהול משתמשים (Angular) – ניהול משתמשים, הרשאות, ודוחות פעילות.

## מסך הצגת דוח חודשי – גרפים וניתוחים סטטיסטיים על מצב הבריאות של המטופל.

## 

## **3. איפיון תמציתי**

### **3.1 רשימת פונקציות המערכת**

| **פונקציה** | **Route** | **פרמטרים** | **פלט** |
| --- | --- | --- | --- |
| העלאת מסמך רפואי | POST /api/files/upload | file: File | { fileId, url } |
| חיפוש מסמכים | GET /api/files/search | query: string | רשימת מסמכים |
| פענוח מסמך רפואי | POST /api/files/parse | fileId: string | { parsedText } |
| קביעת תור | POST /api/appointments | doctorId, date | { appointmentId } |
| ניהול משתמשים | GET /api/users | - | רשימת משתמשים |

### **3.2 אימות והרשאות**

* **JWT Authentication** להגנה על API.
* רמות גישה: **מטופל**, **רופא**, **מנהל מערכת**.
* הרשאות מבוססות תפקידים (RBAC).

### **3.3 אפליקציית ניהול**

* צפייה וניהול משתמשים והרשאות (Angular).
* עדכון סטטוסי תורים.
* הפקת דוחות פעילות למנהלים.

#### 📂 טבלאות המערכת:

#### 4.1 טבלת משתמשים (Users)

| **עמודה** | **סוג נתונים** | **תיאור** |
| --- | --- | --- |
| id | INT (PK) | מזהה משתמש |
| full\_name | VARCHAR(100) | שם מלא |
| email | VARCHAR(100) | אימייל ייחודי |
| role | ENUM('patient', 'doctor', 'admin') | סוג משתמש |
| created\_at | TIMESTAMP | תאריך יצירה |

#### 4.2 טבלת קבצים רפואיים (Files)

| **עמודה** | **סוג נתונים** | **תיאור** |
| --- | --- | --- |
| id | INT (PK) | מזהה קובץ |
| user\_id | INT (FK) | מזהה משתמש |
| file\_name | VARCHAR(255) | שם הקובץ |
| file\_url | TEXT | מיקום הקובץ בענן |
| parsed\_text | TEXT | טקסט OCR |
| file\_type | VARCHAR(50) | סוג הקובץ (PDF, תמונה וכו') |
| status | ENUM('pending', 'approved', 'rejected') | סטטוס קובץ |
| uploaded\_at | TIMESTAMP | תאריך העלאה |
| updated\_at | TIMESTAMP | תאריך עדכון |

#### 4.3 טבלת תורים (Appointments)

| **עמודה** | **סוג נתונים** | **תיאור** |
| --- | --- | --- |
| id | INT (PK) | מזהה תור |
| patient\_id | INT (FK) | מזהה מטופל |
| doctor\_id | INT (FK) | מזהה רופא |
| appointment\_date | TIMESTAMP | תאריך ושעה |
| status | ENUM('pending', 'confirmed', 'canceled') | סטטוס |
| reason | TEXT | סיבת התור |
| updated\_at | TIMESTAMP | תאריך עדכון |

#### 4.4 טבלת התכתבויות (Messages)

| **עמודה** | **סוג נתונים** | **תיאור** |
| --- | --- | --- |
| id | INT (PK) | מזהה הודעה |
| sender\_id | INT (FK) | מזהה השולח |
| receiver\_id | INT (FK) | מזהה המקבל |
| message\_text | TEXT | תוכן ההודעה |
| sent\_at | TIMESTAMP | תאריך ושעת שליחה |

#### 4.5 טבלת היסטוריית תורים (Appointment\_History)

| **עמודה** | **סוג נתונים** | **תיאור** |
| --- | --- | --- |
| id | INT (PK) | מזהה רשומה |
| appointment\_id | INT (FK) | מזהה תור |
| status | ENUM('pending', 'confirmed', 'canceled', 'rescheduled') | סטטוס קודם |
| changed\_at | TIMESTAMP | תאריך שינוי |
| changed\_by | INT (FK) | מי ביצע את השינוי |

## **5📌 ספרינט 1 - תשתיות ופיתוח בסיסי**

#### **מטרות:**

## **הקמת פרויקט .NET Core לפיתוח API.**

## **הגדרת מסד נתונים PostgreSQL / MongoDB.**

## **יצירת מערכת אימות משתמשים (JWT) והצפנת נתונים.**

## **פיתוח API ראשוני לניהול משתמשים, כולל הרשאות RBAC (Patient, Doctor, Admin).**

## **בניית מערכת העלאת מסמכים רפואיים ואחסונם ב- AWS S3 / Firebase.**

## **הגדרת CI/CD באמצעות GitHub Actions לפריסה אוטומטית.**

#### **תוצרים:**

## **✅ בסיס נתונים מוגדר עם טבלאות: Users, Files, Appointments ✅ API ראשוני: הרשמה, התחברות, העלאת מסמכים רפואיים, חיפוש מסמכים ✅ אימות והצפנה: JWT, OAuth2 ✅ תשתית CI/CD לפריסה ראשונית**

## 

### **📌 ספרינט 2 - פיתוח ממשק משתמש וניהול משתמשים**

#### **מטרות:**

## **פיתוח ממשק ניהול משתמשים ב- Angular (הרשאות, עדכון מידע).**

## **פיתוח מערכת לניהול קבצים רפואיים ב- React (העלאה, חיפוש, תצוגה).**

## **חיבור ה-Frontend ל-API בעזרת Axios / Fetch.**

## **בניית ממשק התחברות מאובטח עם הזדהות JWT.**

## **יצירת מערכת הרשאות מתקדמת לניהול משתמשים לפי תפקיד.**

#### **תוצרים:**

## **✅ ממשק התחברות והרשמה מאובטחים ✅ מסך ניהול קבצים רפואיים (React) ✅ מסך ניהול משתמשים והרשאות (Angular) ✅ חיבור מלא של ה-Frontend ל-Backend**

## 

### **📌 ספרינט 3 - ניהול תורים ואינטגרציה עם רופאים**

#### **מטרות:**

## **פיתוח מערכת קביעת תורים.**

## **יצירת ממשק לזימון רופאים בהתאמה לנתוני המסמכים הרפואיים.**

## **הוספת תזכורות חכמות לבדיקות עתידיות.**

## **אינטגרציה עם API של ספקי שירותי בריאות (אם רלוונטי).**

## **ניהול התראות ועדכון מטופלים ורופאים על פגישות קרובות.**

#### **תוצרים:**

## **✅ מסך בחירת רופא והתאמת תור לפי התמחות ✅ אפשרות זימון רופא והצגת זמינות ✅ מערכת תזכורות והתראות (SMS / Email)**

## 

### **📌 ספרינט 4 - AI, OCR ואבטחה מתקדמת**

#### **מטרות:**

## **שילוב OCR לזיהוי טקסט במסמכים רפואיים.**

## **עיבוד מסמכים באמצעות AI לזיהוי בעיות רפואיות.**

## **הצגת סיכום רפואי אוטומטי על סמך המסמכים שהועלו.**

## **הטמעת שכבות אבטחה נוספות – הצפנת מסמכים, הגבלת גישה.**

## **שיפור ביצועים ואופטימיזציה לטעינת קבצים.**

#### **תוצרים:**

## **✅ פענוח אוטומטי של מסמכים רפואיים עם OCR ✅ AI לזיהוי מצבים רפואיים והצעת זימון תור ✅ אבטחה מתקדמת (הצפנת מסמכים, גישה לפי הרשאות)**

## 