**אפיון מערכת לניהול קבצים ותמונות**

**1. תיאור המערכת**

**1.1 מטרת הפרויקט**

מערכת לניהול קבצים ותמונות בענן, המאפשרת למשתמשים פרטיים לנהל קבצים ותמונות בצורה מסודרת, לשתף אותם, ולגשת אליהם מכל מקום. המערכת כוללת אפליקציה עיקרית לניהול קבצים ב-React, ואפליקציה ניהולית קטנה ב-Angular לניהול משתמשים והרשאות.

המערכת נועדה לספק פתרון נוח ויעיל לניהול קבצים ותמונות עבור משתמשים פרטיים ומשפחות.

**תכונות עיקריות:**

* העלאה ואחסון של קבצים ותמונות בענן.
* ארגון קבצים בתיקיות ותיוג חכם.
* חיפוש וסינון קבצים לפי קטגוריות, תאריכים ותוויות.
* שיתוף קבצים ותמונות בצורה מאובטחת.
* עיבוד תמונות מתקדם (יצירת מצגות, GIFs, קולאז'ים ועוד).
* ניהול משתמשים והרשאות (באמצעות אפליקציית Angular למנהלים).
* הפקת דוחות פעילות וניהול מערכת.

**1.2 קהל היעד**

* משפחות המעוניינות לנהל אלבומי תמונות משפחתיים בצורה חכמה.
* משתמשים פרטיים הזקוקים לאחסון מאובטח ומסודר לקבצים ותמונות.
* עסקים קטנים הזקוקים לפלטפורמה לניהול ושיתוף מסמכים וקבצים.

**1.3 פונקציונליות עיקרית**

**אפליקציית React (משתמשים רגילים):**

1. העלאת קבצים ותמונות.
2. ארגון הקבצים בתיקיות עם אפשרות תיוג.
3. חיפוש מתקדם לפי שם, תאריך, תגיות וסוג קובץ.
4. אפשרות ליצירת מצגות, GIFs, קולאז'ים.
5. שיתוף קבצים עם משתמשים אחרים.
6. מחיקת קבצים וניהול תיקיות.
7. מסך פרופיל משתמש וניהול חשבון אישי.

**אפליקציית Angular (מנהלנים):**

1. מסך התחברות מאובטח.
2. ניהול משתמשים (הוספה, עדכון, מחיקה, הקצאת הרשאות).
3. צפייה בלוגים של המערכת.
4. דוחות שימוש וסטטיסטיקות על אחסון ופעילות משתמשים.
5. ניהול הגדרות מערכת (הגבלת שטח אחסון, סוגי קבצים מותרים).

**פונקציונליות של המערכת**

1. העלאת קבצים ותמונות ושמירתם ב-Amazon S3.
2. יצירת תיקיות וארגון קבצים לפי קטגוריות.
3. תיוג קבצים ותמונות לזיהוי מהיר.
4. חיפוש מתקדם לפי תאריכים, תגיות וסוגי קבצים.
5. שיתוף קבצים ותמונות עם משתמשים אחרים.
6. יצירת ועיבוד תמונות (מצגות, GIFs, קולאז'ים ועוד).
7. ניהול משתמשים והרשאות באפליקציית המנהל.
8. תהליכי AI לזיהוי פנים, תיוג אוטומטי ותיאור תמונות.

**1.4 בעיות שהמערכת פותרת**

* מפחיתה את הצורך בחיפוש קבצים במכשירים שונים על ידי אחסון מסודר בענן.
* מספקת פתרון לשיתוף קבצים בקלות ללא צורך בשליחת מיילים.
* שומרת על סדר באמצעות תיקיות, תיוגים וחיפוש חכם.
* מספקת אבטחת מידע ושמירה על פרטיות באמצעות הרשאות וניהול משתמשים.

**1.5 טכנולוגיות בשימוש**

* **Backend**: .NET 9 לפיתוח API מרכזי.
* **Frontend**:
  + **React.js** - לניהול קבצים ותמונות (משתמשים רגילים).
  + **Angular 19** - לאפליקציית ניהול משתמשים והרשאות.
* **Database**: PostgreSQL לאחסון נתוני משתמשים, תיקיות ותיוגים.
* **Storage**: Amazon S3 לאחסון קבצים ותמונות.
* **Authentication**: JWT לאימות משתמשים.
* **State Management**: Redux (ב-React) ו-Services (ב-Angular) לניהול נתונים בין רכיבים.

**2. עיצוב ממשק משתמש ויזואלי**

* מסך התחברות ורישום.
* מסך ניהול קבצים (רשימת קבצים, תצוגת תיקיות, העלאה).
* מסך צפייה ועריכה של קובץ/תמונה.
* מסך חיפוש וסינון מתקדם.
* מסך ניהול משתמשים (באפליקציית מנהל).

**3. איפיון תמציתי**

**3.1 רשימת פונקציות המערכת**

* **Route**: POST /api/upload
  + **פרמטרים**: קובץ (file), תגיות (tags)
  + **פלט**: מזהה קובץ (fileId), כתובת (url)
* **Route**: GET /api/search
  + **פרמטרים**: תגית (tag), תאריך (date), סוג קובץ (type)
  + **פלט**: רשימת קבצים תואמים
* **Route**: POST /api/share
  + **פרמטרים**: מזהה קובץ (fileId), מזהה משתמש (userId)
  + **פלט**: אישור שיתוף

**3.2 אימות והרשאות**

* אימות באמצעות JWT.
* תפקידים: משתמש רגיל, מנהל.
* הרשאות ניהול עבור מנהל בלבד (מחיקה, דוחות, ניהול משתמשים).

**3.3 אפליקצית ניהול**

* דוחות פעילות משתמשים.
* ניהול משתמשים והרשאות.
* הגדרת פרמטרים מערכתיים (מכסת אחסון, תצוגה).

**4. תרשים מבנה הטבלאות (Database Schema)**

**טבלת Users**

| **עמודה** | **סוג נתונים** | **מאפיינים** | **תיאור** |
| --- | --- | --- | --- |
| id | INT | PK | מזהה ייחודי |
| first\_name | VARCHAR(50) | NOT NULL | שם פרטי |
| email | VARCHAR(100) | UNIQUE | אימייל |
| created\_at | TIMESTAMP | DEFAULT NOW() | תאריך יצירה |

**טבלת Files**

| **עמודה** | **סוג נתונים** | **מאפיינים** | **תיאור** |
| --- | --- | --- | --- |
| id | INT | PK | מזהה קובץ |
| user\_id | INT | FK → Users.id | בעלים |
| file\_name | VARCHAR(255) | NOT NULL | שם הקובץ |
| tags | TEXT |  | תגיות |
| created\_at | TIMESTAMP | DEFAULT NOW() | תאריך העלאה |

**1.6 מתודולוגיית פיתוח (Agile)**

הפיתוח יתבצע בשיטת Agile עם חלוקה ל-5 ספרינטים.

**ספרינט 1 - תשתיות ופיתוח API (שבועות 1-2)**

* הגדרת מסד נתונים PostgreSQL.
* פיתוח API לניהול משתמשים, העלאת קבצים ותיוגים.
* הטמעת אימות משתמשים באמצעות JWT.
* קביעת הרשאות משתמשים וניהול רמות גישה.

**ספרינט 2 - ממשק משתמש ב-React (שבועות 3-4)**

* פיתוח דף התחברות.
* עיצוב דף ניהול קבצים (העלאה, מחיקה, תיוגים).
* הטמעת ניהול תיקיות ותיוגים.
* אינטגרציה בין React ל-API.

**ספרינט 3 - אפליקציית ניהול ב-Angular (שבועות 5-6)**

* פיתוח מסך התחברות למנהלים.
* יצירת ממשק לניהול משתמשים והרשאות.
* הוספת תצוגת דוחות וסטטיסטיקות.

**ספרינט 4 - שיפור ביצועים ואבטחה (שבועות 7-8)**

* הטמעת אחסון קבצים מאובטח ב-S3.
* בדיקות עומסים ואופטימיזציות למסד הנתונים.
* שיפור חוויית משתמש והוספת אפשרויות חיפוש מתקדמות.

**ספרינט 5 - פריסה והשקה (שבועות 9-10)**

* פריסת המערכת על AWS.
* בדיקות אחרונות ושיפור חוויית משתמש.
* הכנת תיעוד למנהלים ולמשתמשים.

**הגדרת סבבי פיתוח (ספרינטים)**

**ספרינט 1 - תשתית ופיתוח בסיסי**

* הקמת סביבת עבודה (React.js, .NET 9, PostgreSQL).
* פיתוח API בסיסי לניהול משתמשים ואלבומים.
* יצירת טבלאות מסד נתונים והתחברות ל-S3.

**ספרינט 2 - ממשק משתמש ראשוני**

* עיצוב מסך התחברות וניהול קבצים בסיסי.
* העלאת קבצים ותמונות ואחסון ב-S3.

**ספרינט 3 - פיצ'רים מתקדמים**

* חיפוש וסינון לפי תגיות ותאריכים.
* יצירת ממשק צפייה ועריכה.
* שיתוף קבצים עם משתמשים אחרים.

**ספרינט 4 - אבטחה ושיפורים**

* הטמעת הרשאות וגיבוי קבצים.
* אופטימיזציה ובדיקות ביצועים.

**ספרינט 5 - פריסה והשקה**

* פריסת האפליקציה על AWS.
* עיצוב דף נחיתה למשתמשים חדשים.

זהו אפיון מעודכן שמשלב בין ניהול קבצים ותמונות תוך שמירה על המבנה ההיררכי. האם יש משהו שתרצי להוסיף? 😊