

מסמך אפיון // אפליקציית אלבום תמונות משפחתי

1. תיאור המערכת

1.1 מטרת הפרויקט

האפליקציה נועדה לאפשר למשתמשים לנהל את התמונות המשפחתיות שלהם בצורה נוחה ואינטואיטיבית. היא תאפשר העלאת תמונות, ארגון בתיקות, חיפוש לפי סינון, שיתוף עם אחרים, יצירת מצגות GIFs, וקולאז'ים. המטרה היא לפשט את ניהול התמונות האישיות ולהפוך את חווית השימוש למהנה ויעילה יותר.

1.2 קהל היעד

קהל היעד העיקרי של האפליקציה הוא אמהות לילדים, המעוניינות לנהל ולארגן את התמונות המשפחתיות שלהן בצורה מסודרת. האפליקציה מתאימה גם למשפחות שרוצות לשתף תמונות בין בני המשפחה.

1.3 פונקציונליות של המערכת

האפליקציה תכלול את הפונקציונליות הבאה:

1. העלאת תמונות לענן.
2. ארגון תמונות בתיקות.
3. תיוג תמונות וחיפוש לפי תגיות או תאריכים.
4. יצירת מצגות GIFs, וקולאז'ים.
5. שיתוף תמונות עם משתמשים אחרים.
6. ניהול משתמשים ואפליקציית מנהל.

1.4 בעיות שהמערכת פותרת

האפליקציה פותרת את הבעיה של ניהול תמונות מבולגן ומפוזר בין מכשירים שונים. היא מאפשרת גישה נוחה לכל התמונות ממקום אחד, מבטיחה גיבוי אוטומטי כדי למנוע אובדן תמונות, היא מאפשרת ממשק חוויתי עם פיצ'רים קלים ומוצלחים ומאפשרת שיתוף קל עם אחרים.

1.5 טכנולוגיות בשימוש

הפרויקט יפותח באמצעות הטכנולוגיות הבאות:

1. צד שרת NET 9: (Backend) לבניית ה-API-
2. צד לקוח React.ts (Frontend): לבניית ממשק משתמש אינטראקטיבי.
3. מוד נתונים PostgreSQL: לאחסון נתונים של משתמשים ותמונות.

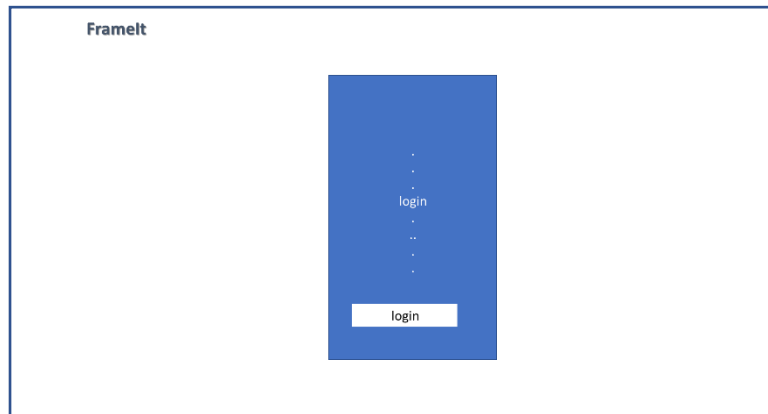
4. אחסון קבצים: שימוש ב Amazon S3-לאחסון תמונות.

1.6 מתודולוגיית פיתוח(Agile)

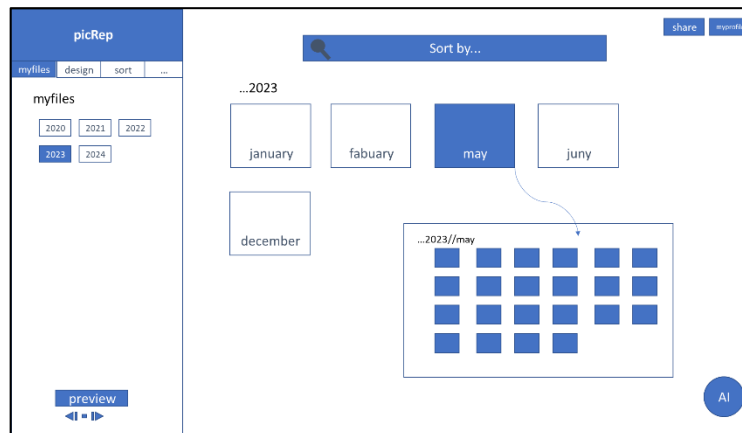
הפרויקט יפותח במתודולוגיית Agile עם חלוקה ל-5 ספרינטים בני שבועיים כל אחד. כל ספרינט יתמקד במשימות ליבה עיקריות.

2. עיצוב ממשק משתמש ויזואלי

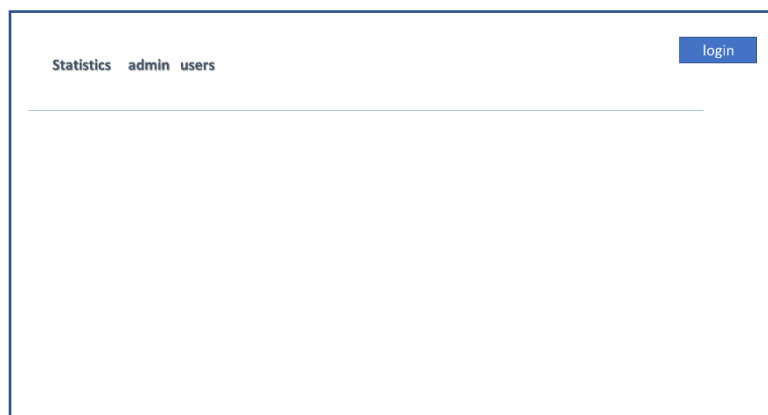
- **מסך כניסה/רישום:** כולל שדות למילוי פרטי משתמש, כפתור רישום והתחברות.



- **מסך ניהול קבצים:** מציג תקיות, תמונות ואפשרויות שונות.



- **מסך אדמין:** סטטיסטיקות שימוש, ניהול משתמשים.



3. איפיון תמציתי

3.1 רשימת פונקציות המערכת

העלאת קובץ

- **Route:** POST /api/upload

- **פרמטרים:**

- **Body:** { file: File, tags: string[] }

- **פלט:**

- data: { fileId: string, url: string }

- **לוגיקה עסקית:**

- הגבלת גודל קובץ ל-MB50

- הוספת חתימת זמן לקובץ

- שמירה ב-S3-

הורדת קבצים

- **Route:** GET /api/download/{folderId}

- **פרמטרים:**

- **Path:** { folderId: string }

- **פלט:**

- קובץ ZIP להורדה

- **לוגיקה עסקית:**

- יצירת ZIP של כל התמונות בתיקייה

- שליחת לינק להורדה

סינון תמונות

- **Route:** POST /api/filter

- **פרמטרים:**

- **Body:** { filterType: string, fileIds: string[] }

- **פלט:**

- data: { filteredFiles: File[] }

- **לוגיקה עסקית:**

- שימוש ב AI-לזיהוי תמונות מטושטשות, פנים/חוך, מספר אנשים

יצירת תיקייה חדשה

- **Route:** POST /api/folders

- **פרמטרים:**

Body: { name: string, parentId?: string } ○

• **פלט:**

data: { folderId: string, name: string } ○

• **לוגיקה עסקית:**

○ בדיקה אם שם התיקיה כבר קיים

○ יצירה במסד הנתונים

מחיקת תמונה

• **Route:** DELETE /api/files/{fileId}

• **פרמטרים:**

Path: { fileId: string } ○

• **פלט:**

status: 204 No Content ○

• **לוגיקה עסקית:**

○ בדיקה אם הקובץ קיים

○ מחיקה מ S3 וממסד הנתונים

יצירת קולאז'

• **Route:** POST /api/collage

• **פרמטרים:**

Body: { fileIds: string[], template: string } ○

• **פלט:**

data: { collageUrl: string } ○

• **לוגיקה עסקית:**

○ שילוב תמונות לפי תבנית

○ ייצוא כ-PDF

3.2 אימות והרשאות

האפליקציה מבצעת אימות באמצעות JWT Authentication. התפקידים במערכת כוללים משתמש רגיל ומנהל, כאשר למנהל יש גישה להפקת דוחות וניהול משתמשים.

• **מנגנון אימות:**

○ המערכת משתמשת ב-JWT Authentication-

○ המשתמש מתחבר עם מייל/סיסמה ומקבל JWT Token.

- **תפקידים:**

- **משחמש רגיל:** יכול להעלות, להוריד ולנהל את הקבצים שלו בלבד.
- **מנהל:** יכול לנהל משתמשים, לראות דוחות, להגדיר הרשאות ומגבלות אחסון.

3.3 אפליקצית ניהול

יכולות המנהל כוללות:

- הפקת דוחות פעילות משתמשים.
- CRUD למשתמשים + שיוך תפקידים.
- עדכון הרשאות למשתמשים.

מסכים עיקריים

1. **מסך התחברות** – אימות באמצעות JWT
2. **ניהול משתמשים** – הוספה, עדכון, מחיקה, שינוי הרשאות
3. **דוחות פעילות** – נתונים סטטיסטיים על שימוש במערכת
4. **ניהול הגדרות מערכת** – ניהול מכסות אחסון, הרשאות גישה, מגבלות מערכת
5. **צפייה בלוגים** – מעקב אחרי פעולות במערכת (העלאות, מחיקות, כניסות וכו')

פונקציות עיקריות

- **ניהול משתמשים:**

- GET /api/admin/users – שליפת רשימת משתמשים
- POST /api/admin/users – יצירת משתמש חדש
- PUT /api/admin/users/{id} – עדכון משתמש
- DELETE /api/admin/users/{id} – מחיקת משתמש

- **דוחות מערכת:**

- GET /api/admin/reports – שליפת נתונים סטטיסטיים

- **ניהול הרשאות:**

- PUT /api/admin/users/{id}/permissions – שינוי תפקידים והרשאות

- **ניהול הגדרות מערכת:**

- PUT /api/admin/settings – עדכון פרמטרים כמו מגבלת אחסון או פורמט ברירת מחדל לקולאז'ים

4. תרשים מבנה הטבלאות (Database Schema)

טבלת Users

שם עמודה	סוג נתונים	מאפיינים	תיאור
id	INT	PRIMARY KEY	מזהה ייחודי
first_name	VARCHAR(50)	NOT NULL	שם פרטי
last_name	VARCHAR(50)	NOT NULL	שם משפחה
email	VARCHAR(100)	UNIQUE, NOT NULL	אימייל
password	VARCHAR(255)	NOT NULL	סיסמה מוצפנת
role	ENUM('admin', 'user')	DEFAULT 'user'	סוג המשתמש
created_at	DATETIME	DEFAULT NOW()	תאריך יצירה
updated_at	DATETIME	ON UPDATE NOW()	תאריך עדכון

טבלת Files

שם עמודה	סוג נתונים	מאפיינים	תיאור
id	INT	PRIMARY KEY	מזהה ייחודי של הקובץ
user_id	INT	FOREIGN KEY → Users(id)	בעל הקובץ
folder_id	INT	FOREIGN KEY → Folders(id)	התיקייה שבה נמצא הקובץ
filename	VARCHAR(255)	NOT NULL	שם הקובץ
url	TEXT	NOT NULL	קישור לקובץ בS3-
size	INT	NOT NULL	גודל הקובץ (בבתים)
created_at	DATETIME	DEFAULT NOW()	תאריך יצירה
updated_at	DATETIME	ON UPDATE NOW()	תאריך עדכון

טבלת Directory

שם עמודה	סוג נתונים	מאפיינים	תיאור
id	INT	PRIMARY KEY	מזהה תיקייה

user_id	INT	FOREIGN KEY → Users(id)	משתמש שיצר את התיקייה
parent_id	INT	FOREIGN KEY → Folders(id), NULLABLE	תיקיית אב (אם יש)
name	VARCHAR(100)	NOT NULL	שם התיקייה
created_at	DATETIME	DEFAULT NOW()	תאריך יצירה
updated_at	DATETIME	ON UPDATE NOW()	תאריך עדכון

קשרים בין הטבלאות:

- משתמשים יכולים להעלות מספר תמונות. (One-to-Many)
- תמונה יכולה להיות במספר תיקיות. (Many-to-Many)

5. הגדרת סבבי פיתוח (ספרינטים)

ספרינט 1

- מטרה: הקמת סביבת עבודה ובסיס נתונים + תהליך התחברות והרשמה
- משימות:
 - פתיחת פרויקטים חדשים ב.NET, React.
 - יצירת טבלאות במסד הנתונים. (PostgreSQL)
 - פיתוח תהליך התחברות והרשמה ב-API.
 - פיתוח מסך התחברות והרשמה ב-React.

ספרינט 2

- מטרה: פיתוח ממשק משתמש ראשוני והעלאת תמונות
- משימות:
 - בניית רכיבי React.js לעיצוב אלבומים וגלריות.
 - אינטגרציה בין Frontend ל API באמצעות Axios.
 - הטמעת העלאת תמונות בסיסית עם Drag & Drop.

- יצירת ממשק משתמש עבור תצוגת אלבומים.

ספרינט 3

- **מטרה:** פיתוח פיצ'רים ליבה

- **משימות:**

- פיתוח מערכת שיתוף אלבומים עם הרשאות גישה.
- הטמעת חיפוש תמונות לפי תגיות או תאריכים.
- יצירת ממשק ניהול אלבומים עם אפשרויות עריכה ומחיקה.
- הוספת אפשרות לתיוג תמונות.

ספרינט 4

- **מטרה:** שיפורים ואבטחה

- **משימות:**

- הטמעת גיבוי אוטומטי יומי לתמונות.
- הוספת אופציות סינון מתקדמות (למשל: לפי טקסט חופשי).
- ביצוע בדיקות אבטחה ואופטימיזציות ב-API.
- שיפור חווית המשתמש בממשק.

ספרינט 5

- **מטרה:** פריסה ופרסום

- **משימות:**

- פריסת האפליקציה בשרת AWS.
- הגדרת דומיין אישי.
- עיצוב דף הבית לשיווק ומשיכת משתמשים פוטנציאליים.
- הכנת דוקומנטציה למשתמשים ולמפתחים.