**מסמך אפיון לפרויקט: אלבום תמונות משפחתי**

1. תאור המערכת

**1.1 מטרת הפרויקט:**

האפליקציה נועדה לאפשר למשתמשים לנהל אלבומי תמונות בצורה נוחה ואינטואיטיבית. היא תאפשר העלאת תמונות, ארגון באלבומים, שיתוף עם אחרים וגיבוי אוטומטי בענן. המטרה היא לפשט את ניהול התמונות האישיות ולהפוך את החוויה למהנה יותר.

האפליקציה תאפשר:

* העלאת התמונת וסידורן באלבומים ותקיות
* תיוג תמונות וחיפוש חכם לפי תאריכים תגיות וזיהוי פנים
* יצירת מצגות קולאז'ים וgifs
* שיתוף תמונות בין משתמשים ויצוא להדפסה
* ניהול משתמשים והרשאות דרך ממשק אדמיין
* שימוש ב AI ליצירת תיוג תמונות, עיבודאוטומטי,זיהוי פנים

**2.1 יעילות המערכת:**

* ניהול מבולגן של תמונות במכשירים שונים
* קושי במציאת תמונות ישנות ללא חיפוש חכם
* שיתוף מסורבל של אלבומים בין בני משפחה
* צורך בשימוש במספר כלים נפרדים ליצירת קולאז'ים או מצגות

זמן בעידן הדיגיטלי, תמונות מפוזרות במקומות שונים, אפליקציות מגוונות ושירותי ענן שונים, מה שמקשה על ארגון ושיתוף נוח. בנוסף, חיפוש תמונות ישנות לפי זיכרון אישי או תאריך הופך למורכב. האפליקציה שלנו מספקת פתרון מרכזי AIלניהול תמונות משפחתיות במקום אחד, עם כלי חיפוש חכם המבוסס על על,תיוג אוטומטי, ויכולת שיתוף מאובטחת עם בני משפחה. בכך, היא משפרת את חוויית המשתמש ומבטיחה שהתמונות החשובות ביותר יישארו מסודרות ונגישות לאורך

**3.1 קהל היעד**

* משפחות המעוניינות לשמור ולשתף זיכרונות משותפים
* משתמשים פרטיים שרוצים לנהל ולארגן את התמונות שלהם
* צלמים חובבים המעוניינים בארגון יעיל של תמונותיהם

קהל היעד העיקרי הוא משפחות המעוניינות לשמור ולארגן את הזיכרונות שלהן בצורה חכמה ונגישה. בנוסף, האפליקציה מתאימה גם לאנשים פרטיים שרוצים לנהל את ספריית התמונות שלהם, וכן לעסקים קטנים או צלמים פרטיים הזקוקים לכלי ניהול מתקדם לתמונות שלהם.

**4.1 הערך שהאפליקציה מביאה למשתמש**

* ניהול תמונות יעיל-ארגון תמונות לפי אלבומים,תיוגים וזיהוי פנים אוטומטי
* חיפוש חכם-ארגון תמונות לפי אלבומים תיוגים זיהוי אוביקטם ותאריכים

### 5.1 טכנולוגיות בשימוש

* .NET 9 API **:Backend**
* : React.js : **Frontend** אפליקציה ראשית
* -Angular ניהול משתמשים
* PostgreSQL **:Database**
* Amazon S3: **Cloud Storage** לאחסון תמונות
* **AI**:שימוש ב-OpenAI לזיהוי תמונות ויצירת תיאורים
* : JWT + OAuth **:Authentication**

### 1.6 מתודולוגיית פיתוח (Agile)

הפיתוח יתבצע במתודולוגיית **Agile**, עם חלוקה ל-56 ספרינטים של שבועיים כל אחד

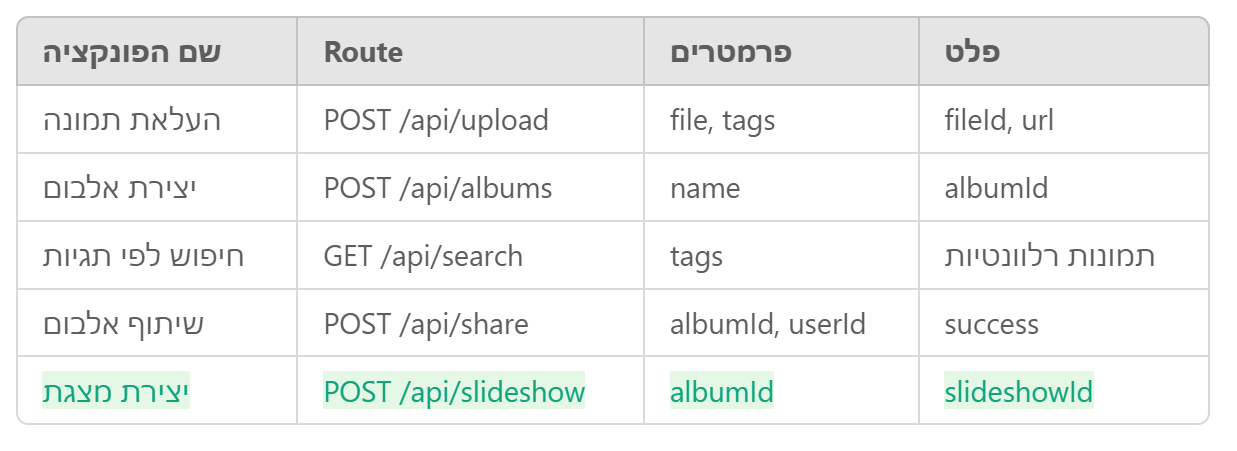
## .2 עיצוב ממשק משתמש ויזואלי

### Wireframes מסכים מרכזיים

1. **מסך התחברות / הרשמה** – טופס רישום והתחברות
2. **דף ראשי** – תצוגת אלבומים וגלריות תמונות
3. **מסך העלאת תמונות** – אפשרות לגרירת קבצים, הוספת תגיות ובחירת אלבום
4. **ניהול משתמשים (Admin)** – ניהול הרשאות, מעקב אחר פעילות משתמשים
5. **מסך חיפוש מתקדם** – חיפוש לפי זיהוי פנים, תיוגים ותאריכים
6. **מסך יצירת מצגת/קולאז'** – כלים לעריכת תמונות וקבצי וידאו

## 3 .איפיון תמציתי

**3.1 פונקציות עיקריות במערכת**



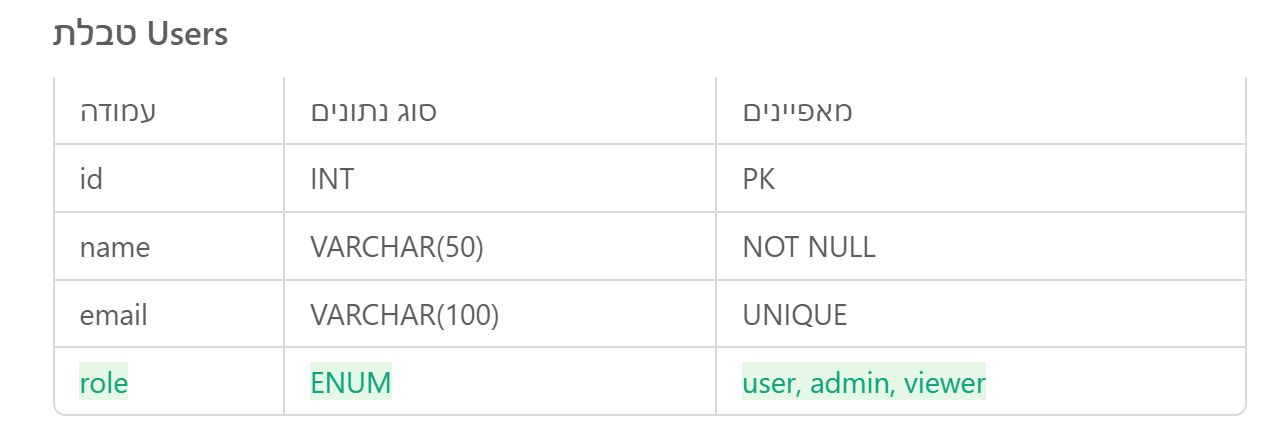
### 3.2 אימות והרשאות

* **JWT Authentication** לניהול הרשאות משתמשים
* **OAuth** לאפשרות התחברות דרך Google ו-Facebook
* רמות גישה: **משתמש רגיל, מנהל מערכת, צופה בלבד**

### 3.3 אפליקציית ניהול (Admin Panel)

* דוחות פעילות משתמשים
* ניהול הרשאות ושיוך תפקידים
* סטטיסטיקות שימוש
* ניהול גיבויים ושחזור נתונים

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

טבלאות עקריות:  


תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

## 5. הגדרת סבבי פיתוח (ספרינטים)

### ****ספרינט 1: תשתיות וניהול משתמשים****

* תכנון מסד נתונים (PostgreSQL)
* פיתוח API לניהול משתמשים ב-.NET
* אימות משתמשים עם JWT + OAuth
* הגדרת אחסון Amazon S3

### ****ספרינט 2: ממשק משתמש ראשוני****

* בניית רכיבי React.js להצגת אלבומים
* חיבור בין Frontend ל-API
* העלאת תמונות בסיסית

### ****ספרינט 3: פיצ'רים מתקדמים****

* חיפוש חכם באמצעות AI
* ניהול תגיות ותאריכים
* יצירת מצגות וקולאז'ים
* מערכת הערות ותגובות

### ****ספרינט 4: אבטחה ושיפורים****

* גיבויים אוטומטיים
* שיפור ביצועים ב-API
* בדיקות אבטחה

### ****ספרינט 5: פריסה ופרסום****

* פריסת האפליקציה ב-AWS
* הגדרת דומיין
* שיווק ופרסום ראשוני

### ****ספרינט 6: תיקונים אחרונים ושיפורים****

* בדיקות משתמשים
* תיקוני באגים
* שיפור UX/UI בהתאם למשוב משתמשים