**1. תיאור המערכת:**

האפליקציה נועדה לספק מערכת לניהול שמחות שנתוניות (כגון חתונות, בר מצוות, ימי הולדת ועוד) ומאפשרת למשתמשים לנהל את פרטי השמחות, להעלות קבצים ולהתעדכן בנוגע להן. בנוסף, ישנו פיצ'ר ייחודי המאפשר למנהל להעלות קבצים מעניינים כל שבוע, שיתעדכנו אוטומטית ויהיו זמינים לכלל המשתמשים לצפייה.

**1.1 מטרת הפרויקט:**

המטרה של האפליקציה היא לאפשר למשתמשים לשמור על קשר עם בני כיתותיהם ולחוות את השמחות השונות בצורה דינמית. האפליקציה תאפשר למנהל להעלות קבצים (כגון הזמנות חתונה, מצגות, סרטונים או עדכונים אחרים) באופן שבועי, וכל משתמש יוכל לצפות בקבצים שהועלו.

**1.2 קהל היעד:**

האפליקציה מיועדת לבני שנתונים/כיתות המעוניינים לשמור על קשר לאורך השנים ולהתעדכן בשמחות ובחגיגות. בנוסף, המנהל יוכל לשדרג את חוויית המשתמש על ידי העלאת תכנים מעניינים שמותאמים לקהל המשתמשים.

**1.3 פונקציונליות של המערכת:**

הפונקציות העיקריות של המערכת כוללות את:

* **ניהול שמחות:** כל משתמש יכול להעלות שמחות פרטיות (חתונה, בר מצווה וכו') ולשתף עם חברי השנתון.
* **צפייה בקבצים שהועלו:** כל משתמש יכול לצפות בקבצים שהועלו על ידי המנהל או משתמשים אחרים.
* **העלאת קבצים שבועית על ידי המנהל:** המנהל יוכל להעלות קבצים מעניינים (כגון סרטונים, מצגות, הזמנות וכו') אחת לשבוע.
* **צפייה בקבצים על פי תאריכים:** אפשרות להציג קבצים לפי תאריכים (השבוע, החודש, שנה).
* **ניהול משתמשים:** כל משתמש יכול להוסיף או לעדכן את פרטי השמחות והפרופיל שלו.
* **הרשאות גישה:** רק המנהל יכול להעלות את הקבצים המיוחדים.

**1.4 בעיות שהמערכת פותרת:**

המערכת פותרת את הבעיה של אי-יכולת לעקוב אחרי השמחות של חברי כיתה לאורך השנים, ומספקת מערכת ניהול שמחות שמרכזת את כל המידע במקום אחד.

**1.5 טכנולוגיות בשימוש:**

הפרויקט יפותח בעזרת הטכנולוגיות הבאות:

* **צד שרת (Backend):** NET 9 (ASP.NET Core) – לבניית ה-API.
* **צד לקוח (Frontend):**
  + **React.js** – ממשק משתמש אינטראקטיבי שבו המשתמשים יכולים לצפות בקבצים, להעלות אותם ולעדכן את השמחות.
  + **Angular** – עבור המנהל, כולל ניהול משתמשים וסטטיסטיקות של המערכת.
* **מסד נתונים:** PostgreSQL – לאחסון נתונים של משתמשים, שמחות, קבצים ועוד.
* **אחסון קבצים:** שימוש ב-Amazon S3 לאחסון קבצים.

**2. עיצוב ממשק משתמש ויזואלי:**

**מסך הכניסה / רישום**

* **מה יהיה ב-React:** כל המשתמשים יראו את המסך הזה עם אפשרויות להרשמה והתחברות.

**מסך ראשי:**

* **מה יהיה ב-React:** כל משתמש יוכל לצפות בשמחות הקרובות, להעלות קבצים ולהתעדכן בזמני השמחות.
* **מה יהיה ב-Angular:** המנהל יוכל להעלות קבצים חדשים שבועיים, לנהל משתמשים, ולראות סטטיסטיקות על השמחות שהועלו.

**מסך ניהול קבצים (מנהל בלבד):**

* **מה יהיה ב-Angular:** המנהל יוכל להעלות קבצים ולנהל את התוכן (העלאה, עדכון, מחיקה).
* **הפונקציות:**
  + העלאת קבצים.
  + עדכון קבצים שנוספו.
  + ניהול קטגוריות של קבצים.

**מסך פעילות (משתמשים בלבד):**

* **מה יהיה ב-React:** המשתמשים יראו את רשימת הקבצים שהועלו בשבועות קודמים, ויוכלו לצפות בהם.

**3. איפיון תמציתי:**

**3.1 רשימת פונקציות המערכת:**

* **העלאת קבצים:**
  + **Route:** POST /api/upload
  + **פרמטרים:** { file: File, tags: string[], category: string }
  + **פלט:** { fileId: string, url: string }
  + **לוגיקה:** כל קובץ יישמר בענן (S3) ויוחזר URL לקובץ.
* **צפייה בקבצים שהועלו:**
  + **Route:** GET /api/files
  + **פלט:** { files: [{ id: string, title: string, url: string, uploadDate: Date }] }
* **ניהול משתמשים (מנהל):**
  + **Route:** POST /api/users (הוספת משתמשים)
  + **Route:** PUT /api/users/{id} (עדכון משתמשים)
  + **Route:** DELETE /api/users/{id} (מחיקת משתמשים)

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

**טבלת Users**

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

**טבלת Files**

| **שם העמודה** | **סוג נתונים** | **מאפיינים** | **תיאור** |
| --- | --- | --- | --- |
| id | INT | PK | מזהה ייחודי |
| title | VARCHAR(100) | NOT NULL | שם הקובץ |
| url | VARCHAR(255) | NOT NULL | כתובת הקובץ |
| created\_at | DATETIME | DEFAULT NOW() | תאריך יצירה |
| updated\_at | DATETIME | DEFAULT NOW() | תאריך עדכון |

**5. הגדרת סבבי פיתוח (ספרינטים):**

**ספרינט 1 - פיתוח בסיסי:**

* בניית התשתית של ה-API.
* הגדרת מודל נתונים בסיסי (Users, Files).
* יצירת פונקציות אימות משתמשים והתחברות.

**ספרינט 2 - פיתוח ממשק React:**

* יצירת מסך כניסה/הרשמה.
* יצירת ממשק להעלאת קבצים והצגת קבצים שהועלו.
* יצירת דפים עם שיתוף ומעקב אחר שמחות.

**ספרינט 3 - פיתוח ממשק Angular (למנהל):**

* יצירת מערכת ניהול קבצים עבור המנהל.
* הוספת אפשרות להעלות קבצים ולהציג סטטיסטיקות.

**ספרינט 4 - שיפורים ופריסה:**

* הטמעת אבטחה ושיפורים נוספים.
* פריסת האפליקציה על שרת.