בס"ד

**מסמך אפיון פרויקט גמר: אפליקציה להעברת קבצים מוצפנים**  
**1. תיאור המערכת**  
1.1 **מטרת הפרויקט:**  
האפליקציה נועדה לאפשר העברה מאובטחת של מסמכים רגישים בין ארגונים ועורכי דין. היא תאפשר העלאה, הצפנה, ושיתוף קבצים באמצעות קישור מוגן, עם הגבלת זמן גישה וקוד אימות. המטרה היא לספק פתרון מאובטח ויעיל להעברת מידע רגיש.  
1.2 **קהל היעד:**  
קהל היעד העיקרי של האפליקציה הוא ארגונים ועורכי דין המעבירים מסמכים רגישים. האפליקציה מתאימה גם לכל מי שזקוק להעברה מאובטחת של קבצים.  
1.3 **פונקציונליות של המערכת:**  
האפליקציה תכלול את הפונקציונליות הבאה:

* העלאת קבצים.
* הצפנת קבצים.
* שיתוף קבצים באמצעות קישור מוגן.
* הגבלת זמן גישה לקבצים.
* קבלת קוד אימות במייל לפתיחת מסמך.
* ניהול משתמשים (הוספה, עדכון, מחיקה).
* הפקת דוחות.
* ניהול הגדרות אבטחה.
* זיהוי קבצים חשודים (אנטי-וירוס).

1.4 **בעיות שהמערכת פותרת:**  
האפליקציה פותרת את הבעיה של העברת מסמכים רגישים בצורה לא מאובטחת. היא מאפשרת העברה מאובטחת של קבצים, תוך שמירה על פרטיות המידע.  
1.5 **טכנולוגיות בשימוש:**  
הפרויקט יפותח באמצעות הטכנולוגיות הבאות:

* צד שרת (Backend): Net.
* צד לקוח (Frontend): React, Angular.
* מסד נתונים: (בחירה שלך).
* אחסון קבצים: שירותי ענן (AWS S3, Azure Blob Storage, וכו').
* AI: שירותי זיהוי קבצים חשודים (אנטי-וירוס) בענן.

1.6 **מתודולוגיית פיתוח (Agile):**  
הפרויקט יפותח במתודולוגיית Agile עם חלוקה ל-5 ספרינטים בני שבועיים כל אחד. כל ספרינט יתמקד במשימות ליבה עיקריות. להלן פירוט הספרינטים הצפויים:

* ספרינט 1 - בניית תשתיות וניהול משתמשים:
* תכנון מודל מסד הנתונים.
* פיתוח API בסיסי ב-.NET.
* הטמעת מערכת אימות משתמשים (Registration/Login) עם JWT.
* הגדרת אחסון קבצים בענן.
* ספרינט 2 - ממשק משתמש ראשוני:
* בניית רכיבי React.js לעיצוב העלאת קבצים ושיתוף.
* אינטגרציה בין Frontend ל-API באמצעות Axios.
* הטמעת העלאת קבצים בסיסית.
* ספרינט 3 - פיצ'רים ליבה:
* פיתוח מערכת שיתוף קבצים עם הרשאות גישה.
* הטמעת הגבלת זמן גישה לקבצים.
* יצירת ממשק ניהול משתמשים ב Angular.
* ספרינט 4 - שיפורים ואבטחה:
* הטמעת זיהוי קבצים חשודים (אנטי-וירוס).
* הוספת אפשרות לקבלת קוד אימות במייל.
* בדיקות אבטחה ואופטימיזציות ב-API.
* ספרינט 5 - פריסה ופרסום:
* פריסת האפליקציה בשרת ענן.
* הגדרת דומיין אישי.
* עיצוב דף הבית לשיווק ומשיכת משתמשים פוטנציאליים.

**2. עיצוב ממשק משתמש ויזואלי**

* שרטוטים (wireframes) של כל מסך מרכזי:
* מסך כניסה/רישום.
* מסך ניהול קבצים (רשימת קבצים עם כפתורי עריכה/מחיקה).
* מסך אדמין (סטטיסטיקות שימוש, ניהול משתמשים).
* פירוט תהליכים מרכזיים (העלאת קובץ, שיתוף קובץ, ניהול משתמשים).

**3. איפיון תמציתי**  
3.1 **רשימת פונקציות המערכת:**

* העלאת קובץ: POST /api/upload
* שיתוף קובץ: POST /api/share
* הורדת קובץ: GET /api/download/{fileId}
* קבלת קוד אימות: POST /api/verify
* ניהול משתמשים: CRUD /api/users
* ניהול הגדרות אבטחה: PUT /api/settings
* זיהוי קבצים חשודים: POST /api/scan

3.2 **אימות והרשאות:**

* אימות באמצעות JWT Authentication.
* תפקידים: משתמש, מנהל.
* הרשאות: משתמש (העלאה, שיתוף, הורדה), מנהל (ניהול משתמשים, הגדרות, דוחות).

3.3 **אפליקציית ניהול:**

* דוחות: הפקת דוחות פעילות משתמשים, סטטיסטיקות שימוש.
* ניהול משתמשים: CRUD למשתמשים + שיוך תפקידים.
* הרשאות: עדכון הרשאות למשתמשים.
* הגדרות מערכת: ניהול פרמטרים גלובליים (מכסות אחסון, הגדרות אבטחה).

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

* רשימת טבלאות עם שדות, סוגי נתונים, ומפתחות (Users, Files, Shares, Logs).
* תרשים המציג את הקשרים בין הטבלאות (אחד-לרבים, רבים-לרבים).

**5. הגדרת סבבי פיתוח (ספרינטים)**

* חלוקת הפרויקט ל-5 ספרינטים, כאשר לכל ספרינט מטרה ברורה וחלוקת משימות, כפי שפורט בסעיף 1.6.

דוקר סטיק סרצ

AWS של אמזון

לשים הרבה ספריות

AI

זה ההוראות לשרטוטים

**. עיצוב ממשק משתמש ויזואלי**

המסכים מרכזיים:

1. מסך כניסה/רישום (אופציונלי - רק אם אפליקציית ניהול תיושם):
   * מסך כניסה: טופס עם שדות שם משתמש וסיסמה, כפתור "התחבר".
   * מסך רישום: טופס עם שדות שם משתמש, אימייל, סיסמה ואישור סיסמה, כפתור "הרשם".
   * זרימת משתמש: משתמשים חדשים יכולים להירשם, משתמשים קיימים יכולים להתחבר.
2. מסך "קבצים שלי" (My Files Screen):
   * רשימת קבצים שהמשתמש העלה, בטבלה או רשימה.
   * לכל קובץ יוצגו: שם קובץ, תאריך העלאה, מספר הורדות, כפתורים/אייקונים לפעולות:
     + "צור קישור מוגן"
     + "מחק קובץ"
   * כפתור "העלה קובץ" - לפתיחת דיאלוג בחירת קובץ.
   * זרימת משתמש: לאחר התחברות, משתמש מגיע למסך זה ורואה רשימת הקבצים שלו. לחיצה על "העלה קובץ" פותחת דיאלוג לבחירה מהמחשב. לחיצה על "צור קישור מוגן" מעבירה למסך יצירת קישור מוגן (סעיף 3). לחיצה על "מחק קובץ" מציגה אישור מחיקה ומבצעת מחיקה.
3. מסך יצירת קישור מוגן (Generate Protected Link Screen):
   * טופס עם השדות הבאים:
     + "סיסמה לקישור" (שדה סיסמה).
     + "הגבל לשימוש פעם אחת" (תיבת סימון).
     + כפתור "צור קישור מוגן".
     + אזור להצגת הקישור שנוצר (לאחר יצירה מוצלחת).
   * זרימת משתמש: מגיעים למסך זה ממסך "קבצים שלי" דרך "צור קישור מוגן" עבור קובץ מסוים. המשתמש מגדיר סיסמה (אופציונלי) ומגדיר אם הקישור לשימוש פעם אחת. לחיצה על "צור קישור מוגן" שולחת בקשה ל-Backend ויוצרת קישור. לאחר יצירה מוצלחת, הקישור מוצג למשתמש (פעם אחת בלבד!).
4. דף גישה לקישור מוגן (Protected Link Access Page):
   * טופס להזנת סיסמה עבור גישה לקישור מוגן.
     + שדה "סיסמה" (שדה סיסמה).
     + כפתור "אמת והורד".
     + הודעת שגיאה (אם הסיסמה שגויה).
   * זרימת משתמש: כאשר משתמש פותח קישור מוגן בדפדפן, הוא מגיע לדף זה. הוא מזין סיסמה ולוחץ "אמת והורד". אם הסיסמה נכונה, מתחילה הורדת הקובץ המוצפן.
5. מסך אדמין (Admin Dashboard Screen) (אופציונלי - אם אפליקציית ניהול תיושם):
   * דשבורד ראשי עם סטטיסטיקות שימוש כלליות (מספר משתמשים, מספר קבצים, נפח אחסון בשימוש וכו').
   * ניווט לאזורי ניהול נוספים (אם יהיו - ניהול משתמשים, דוחות וכו').
   * זרימת משתמש: מנהלי מערכת מגיעים למסך זה לאחר התחברות לאפליקציית הניהול ורואים סקירה כללית של המערכת.