**פרויקט פולסטאק /** מסמך אפיון

**תיאור המערכת:**

האפליקציה נועדה לאפשר לבעל עסק לצפות במצב העסק (הוצ' הכנ') בצורה ויזואלית ונוחה. היא תאפשר העלאת קבלות וחשבוניות, שימוש בנתוני הטפסים למאזן העסק, שיתוף עם מנהלי החשבונות וגיבוי אוטומטי בענן. המטרה היא לשקף לבעל העסק את המאזן הכלכלי בצורה פשוטה ונוחה.

**קהל היעד:**

קהל היעד העיקרי של האפליקציה הוא בעל עסק פרטי וכן עצמאי שרוצים לנהל את נתוני העסק בעצמם. האפליקציה מתאימה גם למשפחות שרוצות לנהל את המאזן הכלכלי בבית.

**פונקציונליות של המערכת:**

האפליקציה תכלול את הפונקציונליות הבאה:

* העלאת חשבוניות וקבלות שונות.
* ניתוח הנתונים מקובץ ועדכון המאזן.
* יצירת דוחות לעסק.
* גיבוי אוטומטי של קבצי ונתוני העסק.
* משוב על התנהלות העסק
* פילוח נתוני העסק.
* אינטגרציה עם תכנת הנהלת חשבונות.
* **AI** זיהוי טקסט בקבלות וניתוח דפוסי הוצ'.

**בעיות שהמערכת פותרת:**

האפליקציה פותרת את הבעיה של אי ודאות על מצב העסק וכן חוסכת עבודה בניהול החשבון. היא מאפשרת גישה נוחה לנתוני העסק, מבטיחה גיבוי אוטומטי כדי למנוע אובדן מידע, ועולה על דפוסי התנהלות שגויים בעסק.

**טכנולוגיות** **בשימוש**:

הפרויקט יפותח באמצעות הטכנולוגיות הבאות:

* **צד שרת (Backend):** NET 8. לבניית הAPI .
* **צד לקוח (Frontend):** React.js לבניית ממשק משתמש (מנהל) אינטראקטיבי.
* **מסד נתונים:** Code First לאחסון נתוני העסק משתמשים וטפסי העסק.
* **אחסון קבצים:** שימוש ב Amazon S3 (Simple Storage Service).

**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 בPostgreSQL טבלאות משתמשים קבלה \ חשבונית ונתוני העסק.

-דיאגרמת UML להמחשת מבנה המערכת.  
- פיתוח API בסיסי ב-.NET 8 עם endpoints לניהול משתמשים קבלה \ חשבונית ונתוני העסק.  
- הטמעת מערכת אימות משתמשים (Registration/Login) עם JWT.  
- הגדרת גיבוי קבצים ב Amazon S3.

ספרינט 2 - ממשק משתמש ראשוני:  
- בניית רכיבי React.js לעיצוב ממשק מנהל.  
- אינטגרציה בין Frontend ל-API באמצעות Axios.

- בניית רכיבי Angular לעיצוב ממשק מנהל.

-גישת משתמש נוסף להעלאת קבצים וצפיה חלקית.

ספרינט 3 - פיצ'רים ליבה:  
- ניתוח נתונים מקובץ ועדכונים רלוונטיים.  
- יצירת דוחות.

ספרינט 4 - שיפורים ואבטחה:

-בדיקות אבטחה ואופטימיזציות ב-API.

-משוב על התנהלות העסק

-פילוח נתוני העסק.

-אינטגרציה עם תכנת הנהלת חשבונות.

AI- זיהוי טקסט בקבלות וניתוח דפוסי הוצ'.

ספרינט 5 - פריסה ופרסום:  
- פריסת האפליקציה בשרת AWS.  
- הגדרת דומיין אישי.  
- עיצוב דף הבית לשיווק ומשיכת משתמשים פוטנציאליים.

**פירוט על הטכנולוגיות**

.NET 8

מה זה?: . NET 8הוא פלטפורמת פיתוח תוכנה מאת מיקרוסופט, המאפשרת לפתח אפליקציות עבור Windows, Linux ו-Mac.

יתרונות:

ביצועים גבוהים: .NET 8 מציע שיפורים בביצועים ובזיכרון.

תמיכה רחבה: ישנה קהילה גדולה ותמיכה רבה בספריות ובכלים.

פיתוח API: קל ליצור APIs באמצעות ASP.NET Core, מה שמאפשר בניית שירותים מהירים ויעילים.

React.js

מה זה?: React היא ספריית JavaScript לבניית ממשקי משתמש, שנוצרה על ידי פייסבוק.

יתרונות:

רכיבים: מאפשרת בניית רכיבים ניתנים לשימוש חוזר, מה שמקל על התחזוקה והפיתוח.

ביצועים: משתמשת ב-Virtual DOM כדי לשפר את הביצועים על ידי הפחתת מספר השינויים שנעשו ב-DOM האמיתי.

קהילה גדולה: ישנן המון ספריות ותוספים זמינים.

PostgreSQL

מה זה?: PostgreSQL הוא מערכת ניהול מסדי נתונים רלציונית (RDBMS) קוד פתוח.

יתרונות:

תמיכה ב-SQL: תומך בשפת SQL ובפונקציות מתקדמות כמו טריגרים, פרוצדורות מאוחסנות ועוד.

גמישות: מאפשרת עבודה עם סוגי נתונים שונים, כולל JSON.

אבטחה: מציעה אפשרויות אבטחה מתקדמות כמו הצפנה וגישות מבוקרות.

Amazon S3 (Simple Storage Service)

מה זה?: שירות אחסון בענן של אמזון המאפשר אחסון נתונים בצורה מאובטחת ובעלת זמינות גבוהה.

יתרונות:

סקלאביליות: יכולת לאחסן כמויות גדולות של נתונים מבלי לדאוג לתשתית.

זמינות גבוהה: נתונים מאוחסנים בצורה מבוזרת, כך שהגישה אליהם מהירה ואמינה.

אבטחה: מציע אפשרויות אבטחה כמו הצפנה ושליטה על גישת משתמשים

**רשימת פונקציות המערכת**

פונקציות על נתוני העסק:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **שם הפונקציה** | **ניתובים** | **פרמטרים** | **פלט** | **לוגיקה עסקית** |
| העלאת קובץ | POST /api/upload | Body: { file: File, details: string[] } | data: { fileId: string, url: string } | הגבלת גודל קובץ ל-50MB,  הוספת חתימת זמן לקובץ.  בדיקת תקינות סוג הקובץ |
| ניתוח נתונים מקובץ | POST /api/analyze | Body: { fileId: string } | data: { analysisResult: object } | ניתוח תוכן הקובץ והפקת נתונים רלוונטיים למאזן העסק. |
| הפקת דוח | GET /api/reports | Query Params: { startDate: string, endDate: string } | data: { report: object } | יצירת דוח על בסיס הנתונים המנותחים בטווח התאריכים שניתן |
| גיבוי קבצים | Route: POST /api/backup | Body: { fileIds: string[] } | data: { success: boolean } | העברת הקבצים ל-Amazon S3 לגיבוי. |

פונקציות המטפלות במשתמשים:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **שם הפונקציה** | **ניתובים** | **פרמטרים** | **פלט** | **לוגיקה עסקית** |
| רישום משתמש חדש | POST /api/register | Body: { username: string, password: string, role: string } | data: { userId: string, message: string } | יצירת משתמש חדש במסד הנתונים עם תפקיד מוגדר. |
| כניסת משתמש | POST /api/login | Body: { username: string, password: string } | data: { token: string, userId: string } | אימות פרטי הכניסה והחזרת טוקן JWT למשתמש. |
| מחיקת משתמש | Route: DELETE /api/users/:userId | Headers:  Authorization: Bearer <token> | data: { message: string } | מחיקת משתמש מהמערכת |
| קבלת פרטי משתמש | Route: GET /api/users/:userId | Headers:  Authorization: Bearer <token> | data: { userId: string, username: string, role: string } | החזרת פרטי המשתמש המבוקש |
| החזרת קובץ | Route: GET /api/files/:fileId | Headers: Authorization: Bearer <token>  Params: fileId - מזהה הקובץ שברצונך להחזיר. | data:  file: התוכן של הקובץ (בפורמט המתאים, לדוגמה, Blob או Base64).  filename: שם הקובץ. | אימות ההרשאות של המשתמש.  חיפוש הקובץ במסד הנתונים והחזרתו אם הוא קיים. |

**אימות והרשאות**

אימות: האפליקציה מבצעת אימות באמצעות JWT Authentication

**תפקידים וההרשאות**:

**בעל עסק**: יכול לנהל משתמשים, להעלות קבצים, לצפות בנתונים, להפיק דוחות.

**מנהלי חשבונות**: גישה לקריאת דוחות ונתונים, אך לא יכול להעלות קבצים.

**משתמש אחר:** יכול להעלות קבצים.

**אפליקציית ניהול**

**דוחות**: הפקת דוחות.

**ניהול משתמשים:**  CRUD (Create, Read, Update, Delete)למשתמשים, כולל שיוך תפקידים.

**הרשאות**: עדכון הרשאות למשתמשים בהתאם לתפקידים.

**הגדרות מערכת**: ניהול פרמטרים גלובליים כמו מכסות אחסון, הגדרות גיבוי ועוד.