**מסמך אפיון לפרויקט ניהול חומרים לסטודנטים עם שימוש בבינה מלאכותית**

**תיאור המערכת**

המערכת מאפשרת למורים לנהל חומרים לסטודנטים בצורה יעילה. המורה יוכל לפתוח שיעור, לסמן נוכחות של תלמידים, ובסוף השיעור להעלות את כל הקבצים והקוד שנלמדו במהלך השיעור. הקבצים ייכנסו לאזור האישי של כל תלמיד. בנוסף, המערכת תשתמש בבינה מלאכותית לתמלול אוטומטי של הקבצים המועלים ולזיהוי תוכן חשוב.

**מטרת הפרויקט**

* יצירת פלטפורמה מרכזית לניהול חומרים לסטודנטים.
* מתן אפשרות למורים לנהל שיעורים בצורה נגישה וידידותית.
* שילוב כלים טכנולוגיים מתקדמים לשיפור חוויית המשתמש (ניהול קבצים, ניהול נוכחות, תמלול אוטומטי).
* ניהול גישה לתוכן באמצעות הרשאות שונות למשתמשים.

**קהל היעד**

* מורים המעוניינים לנהל את החומרים שלהם בצורה מסודרת.
* תלמידים המעוניינים לקבל גישה לחומרים שנלמדו בשיעורים.

**פונקציונליות של המערכת**

1. **ניהול משתמשים**
   * הרשמה והתחברות של מורים ותלמידים.
   * ניהול פרופיל אישי.
2. **ניהול שיעורים**
   * פתיחת שיעור חדש.
   * סימון נוכחות של תלמידים.
   * העלאת קבצים בסוף השיעור.
3. **ניהול קבצים**
   * העלאת קבצים לאזור האישי של התלמידים.
   * תמלול אוטומטי של קבצים באמצעות בינה מלאכותית.
   * הצגת קבצים לפי שיעורים.
4. **אזור אישי לתלמידים**
   * הצגת קבצים שהועלו על ידי המורה.
   * הצגת שיעורים קודמים.

**טכנולוגיות בשימוש**

1. **Backend**: ASP.NET Core לניהול ה-API.
2. **Frontend**: Angular לפיתוח ממשק משתמש דינמי.
3. **Database**: SQL Server לאחסון נתוני משתמשים ושיעורים.
4. **Cloud Storage**: Amazon S3 לאחסון מאובטח של קבצים.
5. **AI Services**: שימוש ב-AI לתמלול אוטומטי וזיהוי תוכן חשוב.
6. **Authentication**: JWT לניהול הרשאות ואבטחת גישה.

**אימות והרשאות**

* אימות מבוסס JWT.
* תפקידים שונים: Admin, Teacher, Student.
* שליטה על פרטיות הקבצים: ציבורי, פרטי, תלמידים בלבד.

**API Routes עיקריים**

1. **ניהול משתמשים**
   * POST /api/users/register – רישום משתמשים.
   * POST /api/users/login – התחברות למערכת.
   * GET /api/users/profile – קבלת נתוני משתמשים.
2. **ניהול שיעורים**
   * POST /api/lessons/create – יצירת שיעור חדש.
   * GET /api/lessons/{id} – צפייה בפרטי שיעור מסוים.
   * POST /api/lessons/{id}/attendance – סימון נוכחות בשיעור.
3. **ניהול קבצים**
   * POST /api/lessons/{id}/upload – העלאת קבצים לשיעור.
   * GET /api/lessons/{id}/files – קבלת רשימת קבצים לשיעור מסוים.
4. **שימוש בבינה מלאכותית**
   * POST /api/ai/transcribe – תמלול אוטומטי של קבצים.
   * POST /api/ai/analyze – ניתוח תוכן וזיהוי תוכן חשוב.

**שלבי הפיתוח (Agile)**

**ספרינט 1 – תשתיות וניהול משתמשים**

* הקמת מסד הנתונים SQL Server (טבלאות משתמשים, שיעורים, קבצים).
* פיתוח API לניהול משתמשים (הרשמה, כניסה, הרשאות).
* אימות משתמשים באמצעות JWT.
* הגדרת אחסון קבצים בענן Amazon S3.

**ספרינט 2 – פיתוח ממשק משתמש וניהול תוכן**

* יצירת דפי פתיחת שיעורים וסימון נוכחות.
* פיתוח מערכת ניהול קבצים והעלאתם.
* אינטגרציה בין ה-Frontend (Angular) ל-Backend (ASP.NET Core).

**ספרינט 3 – שילוב בינה מלאכותית**

* שילוב שירותי AI לתמלול אוטומטי של קבצים.
* פיתוח מערכת לניתוח תוכן וזיהוי תוכן חשוב.

**ספרינט 4 – פריסה, אבטחה ושיפור ביצועים**

* פריסת המערכת ב-Azure.
* בדיקות אבטחה וניטור ביצועים.
* פיתוח דוחות סטטיסטיים על נתוני שימוש ומעורבות משתמשים.
* שיפור חוויית משתמש עם מערכת המלצות מותאמת אישית.

**עיצוב ממשק משתמש**

* **מסך כניסה/רישום** – טופס הרשמה והתחברות עם אימות משתמשים.
* **דף ראשי** – תצוגת שיעורים פתוחים וניהול נוכחות.
* **עמוד שיעור** – הצגת פרטי שיעור, נוכחות וקבצים שהועלו.
* **אזור אישי לתלמידים** – ניהול הקבצים האישיים של התלמידים.

**תרשים מבנה הטבלאות (Database Schema)**

**טבלת משתמשים:**

* user\_id (PK)
* name
* email
* role
* date\_created

**טבלת שיעורים:**

* lesson\_id (PK)
* title
* description
* teacher\_id (FK)
* date\_created

**טבלת קבצים:**

* file\_id (PK)
* lesson\_id (FK)
* file\_path
* upload\_date
* transcription

**טבלת נוכחות:**

* attendance\_id (PK)
* lesson\_id (FK)
* student\_id (FK)
* timestamp

**סיכום**

המערכת תספק פלטפורמה גמישה ואינטראקטיבית לניהול חומרים לסטודנטים, תוך שימוש בטכנולוגיות מתקדמות לשיפור חוויית המשתמש, כולל תמלול אוטומטי וניתוח תוכן באמצעות בינה מלאכותית.