**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.

תפקידי המערכת:

יש עמוד בית ואפשרות של כניסה לאזור אישי ניתן להיכנס כמנהל כמורה או כסטודנט, מנהל יכול ליצור קבוצות לימוד של סטודנטים (ולמחוק ולערוך) המכיל את פרטי הסטודנטים ולהוסיף למחוק ולערוך מורים מורה יכול להוסיף, למחוק ולערוך מקצוע לימוד ובתוכו לבחור את קבוצות הלימוד ובתוכו להוסיף למחוק ולערוך שיעורים ולהעלות לתוכם קבצים של השיעור ובעת לחיצה נכנס לאזור האישי של כל התלמידות בקבוצה התכנים שבשיעור זה. ולסטודנט יש באזור האישי לפי תאריכים השיעורים שנשלחו באותו תאריך

בנוסף מערכת לניהול כל האזורים האישיים המאפשר לנהל את מוסדות הלימוד שקיים להם אזור אישי (להוסיף, למחוק לערוך) כגון ניהול הגדרות מערכת, דוחות פעילות כלליים, צפייה בלוגים של המערכת.

**Admin:**

1. **GET**: https://localhost:7253/api/admin
   * **שימוש**: מחזיר את כל המנהלים (Admins) מהמסד נתונים.
2. **GET**: https://localhost:7253/api/admin/{id}
   * **שימוש**: מחזיר מנהל ספציפי לפי id שניתן בכתובת.
3. **POST**: https://localhost:7253/api/admin
   * **שימוש**: מוסיף מנהל חדש למסד הנתונים עם הנתונים שנשלחים בגוף הבקשה.
4. **PUT**: https://localhost:7253/api/admin/{id}
   * **שימוש**: מעדכן מנהל קיים לפי id עם הנתונים שנשלחים בגוף הבקשה.
5. **DELETE**: https://localhost:7253/api/admin/{id}
   * **שימוש**: מוחק מנהל קיים לפי id.

**Auth**:

1. **POST**: https://localhost:7253/api/auth
   * **שימוש**: מבצע כניסה (Login) למערכת. הבקשה כוללת את פרטי המשתמש (דוא"ל וסיסמה) בגוף הבקשה.
   * **תשובה**: אם הכניסה מצליחה, מוחזר טוקן JWT. אם הכניסה נכשלת, מוחזר קוד סטטוס 401 Unauthorized.

**Course**:

1. **GET**: https://localhost:7253/api/course
   * **שימוש**: מחזיר את כל הקורסים (Courses) מהמסד נתונים.
2. **GET**: https://localhost:7253/api/course/{id}
   * **שימוש**: מחזיר קורס ספציפי לפי id שניתן בכתובת.
3. **POST**: https://localhost:7253/api/course
   * **שימוש**: מוסיף קורס חדש למסד הנתונים עם הנתונים שנשלחים בגוף הבקשה.
   * **דרישות**: רק משתמשים עם תפקיד "Teacher" יכולים לבצע פעולה זו.
4. **PUT**: https://localhost:7253/api/course/{id}
   * **שימוש**: מעדכן קורס קיים לפי id עם הנתונים שנשלחים בגוף הבקשה.
   * **דרישות**: רק משתמשים עם תפקיד "Teacher" יכולים לבצע פעולה זו.
5. **DELETE**: https://localhost:7253/api/course/{id}
   * **שימוש**: מוחק קורס קיים לפי id.
   * **דרישות**: רק משתמשים עם תפקיד "Teacher" יכולים לבצע פעולה זו.

**File:**

1. **GET**: https://localhost:7253/api/file
   * **שימוש**: מחזיר את כל הקבצים (Files) מהמסד נתונים.
2. **GET**: https://localhost:7253/api/file/{id}
   * **שימוש**: מחזיר קובץ ספציפי לפי id שניתן בכתובת.
3. **POST**: https://localhost:7253/api/file
   * **שימוש**: מוסיף קובץ חדש למסד הנתונים עם הנתונים שנשלחים בגוף הבקשה.
   * **דרישות**: רק משתמשים עם תפקיד "Teacher" יכולים לבצע פעולה זו.
4. **PUT**: https://localhost:7253/api/file/{id}
   * **שימוש**: מעדכן קובץ קיים לפי id עם הנתונים שנשלחים בגוף הבקשה.
   * **דרישות**: רק משתמשים עם תפקיד "Teacher" יכולים לבצע פעולה זו.
5. **DELETE**: https://localhost:7253/api/file/{id}
   * **שימוש**: מוחק קובץ קיים לפי id.
   * **דרישות**: רק משתמשים עם תפקיד "Teacher" יכולים לבצע פעולה זו.

**Group:**

1. **GET**: https://localhost:7253/api/group
   * **שימוש**: מחזיר את כל הקבוצות (Groups) מהמסד נתונים.
   * **דרישות**: רק משתמשים עם תפקידים "Admin" או "Teacher" יכולים לבצע פעולה זו.
2. **GET**: https://localhost:7253/api/group/{id}
   * **שימוש**: מחזיר קבוצת ספציפית לפי id שניתן בכתובת.
   * **דרישות**: רק משתמשים עם תפקידים "Admin" או "Teacher" יכולים לבצע פעולה זו.
3. **POST**: https://localhost:7253/api/group
   * **שימוש**: מוסיף קבוצת חדשה למסד הנתונים עם הנתונים שנשלחים בגוף הבקשה.
   * **דרישות**: רק משתמשים עם תפקיד "Admin" יכולים לבצע פעולה זו.
4. **PUT**: https://localhost:7253/api/group/{id}
   * **שימוש**: מעדכן קבוצת קיימת לפי id עם הנתונים שנשלחים בגוף הבקשה.
   * **דרישות**: רק משתמשים עם תפקיד "Admin" יכולים לבצע פעולה זו.
5. **DELETE**: https://localhost:7253/api/group/{id}
   * **שימוש**: מוחק קבוצת קיימת לפי id.
   * **דרישות**: רק משתמשים עם תפקיד "Admin" יכולים לבצע פעולה זו.

**Institution:**

1. **GET**: https://localhost:7253/api/institution
   * **שימוש**: מחזיר את כל המוסדות (Institutions) מהמסד נתונים.
   * **דרישות**: רק משתמשים עם מדיניות "DeveloperOnly" יכולים לבצע פעולה זו.
2. **GET**: https://localhost:7253/api/institution/{id}
   * **שימוש**: מחזיר מוסד ספציפי לפי id שניתן בכתובת.
   * **דרישות**: רק משתמשים עם מדיניות "DeveloperOnly" יכולים לבצע פעולה זו.
3. **GET**: https://localhost:7253/api/institution/admin
   * **שימוש**: מחזיר מוסד ספציפי עבור מנהל, לפי InstitutionId שנמצא בטוקן של המשתמש.
   * **דרישות**: רק משתמשים עם תפקיד "Admin" יכולים לבצע פעולה זו.
4. **POST**: https://localhost:7253/api/institution
   * **שימוש**: מוסיף מוסד חדש למסד הנתונים עם הנתונים שנשלחים בגוף הבקשה.
   * **דרישות**: אין דרישות מיוחדות עבור פעולה זו.
5. **PUT**: https://localhost:7253/api/institution
   * **שימוש**: מעדכן מוסד קיים לפי InstitutionId שנמצא בטוקן של המשתמש עם הנתונים שנשלחים בגוף הבקשה.
   * **דרישות**: רק משתמשים עם תפקיד "Admin" יכולים לבצע פעולה זו.
6. **DELETE**: https://localhost:7253/api/institution/{id}
   * **שימוש**: מוחק מוסד קיים לפי id.
   * **דרישות**: רק משתמשים עם מדיניות "DeveloperOnly" יכולים לבצע פעולה זו.
7. **DELETE**: https://localhost:7253/api/institution/admin
   * **שימוש**: מוחק מוסד קיים לפי InstitutionId שנמצא בטוקן של המשתמש.
   * **דרישות**: רק משתמשים עם תפקיד "Admin" יכולים לבצע פעולה זו.

**Lesson:**

1. **GET**: https://localhost:7253/api/lesson
   * **שימוש**: מחזיר את כל השיעורים (Lessons) מהמסד נתונים.
   * **דרישות**: כל משתמש מחובר יכול לבצע פעולה זו.
2. **GET**: https://localhost:7253/api/lesson/{id}
   * **שימוש**: מחזיר שיעור ספציפי לפי id שניתן בכתובת.
   * **דרישות**: כל משתמש מחובר יכול לבצע פעולה זו.
3. **POST**: https://localhost:7253/api/lesson
   * **שימוש**: מוסיף שיעור חדש למסד הנתונים עם הנתונים שנשלחים בגוף הבקשה.
   * **דרישות**: רק משתמשים עם תפקיד "Teacher" יכולים לבצע פעולה זו.
4. **PUT**: https://localhost:7253/api/lesson/{id}
   * **שימוש**: מעדכן שיעור קיים לפי id.
   * **דרישות**: רק משתמשים עם תפקיד "Teacher" יכולים לבצע פעולה זו, והם יכולים לעדכן רק שיעורים שהם מלמדים.
5. **DELETE**: https://localhost:7253/api/lesson/{id}
   * **שימוש**: מוחק שיעור קיים לפי id.
   * **דרישות**: רק משתמשים עם תפקיד "Teacher" יכולים לבצע פעולה זו, והם יכולים למחוק רק שיעורים שהם מלמדים.

**Teacher:**

1. **GET**: https://localhost:7253/api/teacher
   * **שימוש**: מחזיר את כל המורים (Teachers) מהמסד נתונים.
   * **דרישות**: רק משתמשים עם תפקיד "Admin" יכולים לבצע פעולה זו.
2. **GET**: https://localhost:7253/api/teacher/{id}
   * **שימוש**: מחזיר מורה ספציפי לפי id שניתן בכתובת.
   * **דרישות**: רק משתמשים עם תפקיד "Admin" יכולים לבצע פעולה זו.
3. **GET**: https://localhost:7253/api/teacher/me
   * **שימוש**: מחזיר את המידע של המורה המחובר.
   * **דרישות**: כל משתמש עם תפקיד "Teacher" יכול לבצע פעולה זו.
4. **POST**: https://localhost:7253/api/teacher
   * **שימוש**: מוסיף מורה חדש למסד הנתונים עם הנתונים שנשלחים בגוף הבקשה.
   * **דרישות**: רק משתמשים עם תפקיד "Admin" יכולים לבצע פעולה זו.
5. **PUT**: https://localhost:7253/api/teacher/{id}
   * **שימוש**: מעדכן מורה קיים לפי id.
   * **דרישות**: רק משתמשים עם תפקיד "Admin" יכולים לבצע פעולה זו, והם יכולים לעדכן רק מורים השייכים לאותה מוסד.
6. **PUT**: https://localhost:7253/api/teacher/me
   * **שימוש**: מעדכן את הסיסמה של המורה המחובר.
   * **דרישות**: כל משתמש עם תפקיד "Teacher" יכול לבצע פעולה זו.
7. **DELETE**: https://localhost:7253/api/teacher/{id}
   * **שימוש**: מוחק מורה קיים לפי id.
   * **דרישות**: רק משתמשים עם תפקיד "Admin" יכולים לבצע פעולה זו, והם יכולים למחוק רק מורים השייכים לאותה מוסד.

### שושי תלם

כיתה חכמה - ניהול קורסים

ריאקט:

מורות - מסך קורסים, בכל קורס יש שיעורים, הוספת קורס, שיוך קורס לקבוצות, סטטוס שיעור: טיוטה/פרסום, העלאת שיעור, עדכון ומחיקה

תלמידות - אותו מסך כמו מורות, כמובן עם הנתונים שלהן וללא אפשרויות עריכה.

אנגולר: ניהול תלמידות ומורות, ניהול קבוצות, התחברות והרשמה

**קיים**

סרבר: API

קליינט: כלום

מסד נתונים: כן

**עד ר"ח ניסן**

התחברות

מורות - מסך קורסים, בכל קורס יש שיעורים, הוספת קורס, העלאת שיעור

**אחרי פסח**

שיוך קורס לקבוצות, סטטוס שיעור: טיוטה/פרסום, עדכון ומחיקת שיעור

צפיה לתלמידות

אנגולר

פיצ'ר AI