**LiveInsight  
1. הקדמה**

המערכת מיועדת לניהול קבצים ושאלות במערכת שבה משתמשים יכולים להעלות קבצים או שאלות לאחר שצברו מספיק נקודות. כל משתמש יכול להגיב על כל שאלה או קובץ, ובכך לצבור נקודות. קבצים שנעלים או שאלות נבדקות על ידי מערכת AI, וישנה אפשרות לבדיקות אנושיות אם מבוקש. מנהל המערכת אחראי לניהול המשתמשים וביצוע הבדיקות האנושיות.

**2. דרישות מערכת**

**2.1 דרישות פונקציונליות**

1. **צברה של נקודות על תגובות**:
   * כל משתמש יכול להגיב על קבצים או שאלות.
   * כל תגובה מקנה למשתמש מספר נקודות.
   * לאחר שהמשתמש צבר מספר נקודות מינימלי, הוא יכול להעלות קובץ או שאלה למערכת.
2. **העלאת קבצים ושאלות**:
   * כל משתמש יכול להעלות קובץ או שאלה רק אם צבר מספיק נקודות.
   * כאשר קובץ או שאלה מועלים, הם נבדקים אוטומטית על ידי מערכת AI.
3. **בדיקה אוטומטית על ידי AI**:
   * כל קובץ או שאלה שעולה נשלחת למערכת AI לצורך בדיקה.
   * AI בודק אם הקובץ או השאלה עומדים בקריטריונים שנקבעו (למשל, אם השאלה ממוקדת וברורה, אם הקובץ תואם את הפורמט).
4. **בקשה לבדיקת אנוש**:
   * אם קובץ או שאלה לא עומדים בדרישות של AI, המשתמש יכול לבקש בדיקה אנושית.
   * בקשות אלו נשלחות למנהל המערכת, שהוא בודק אם יש לאשר את הקובץ או השאלה או להחזיר אותן לשימוש.
5. **ניהול משתמשים**:
   * המנהל יכול להוציא משתמשים שהפרו את תנאי השימוש.
   * המנהל יכול להפעיל אפשרויות כמו חסימת משתמשים או העלאת דרישות צבירת נקודות.
6. **היסטוריית תגובות ונקודות**:
   * למשתמשים תהיה היסטוריה של כל התגובות שהשאירו, כולל מספר הנקודות שהרוויחו עבור כל תגובה.
   * יש מעקב אחרי מספר הנקודות הנוכחיות של המשתמש.

**3. מבנה נתונים**

**3.1 טבלאות**

**3.1.1 טבלת Users**

| **עמודה** | **סוג נתון** | **תיאור** |
| --- | --- | --- |
| user\_id | INT (PK) | מזהה ייחודי למשתמש |
| username | VARCHAR(255) | שם משתמש |
| email | VARCHAR(255) | דוא"ל המשתמש |
| password | VARCHAR(255) | סיסמה מוצפנת |
| points | INT | מספר הנקודות של המשתמש |
| created\_at | DATETIME | תאריך יצירת המשתמש |
| updated\_at | DATETIME | תאריך עדכון המשתמש |

**3.1.2 טבלת Files**

| **עמודה** | **סוג נתון** | **תיאור** |
| --- | --- | --- |
| file\_id | INT (PK) | מזהה ייחודי לקובץ |
| user\_id | INT (FK) | מזהה המשתמש שעלה את הקובץ (מפתח זר ל-Users) |
| file\_name | VARCHAR(255) | שם הקובץ |
| file\_url | VARCHAR(255) | כתובת הקובץ (למשל ב-S3 או מערכת קבצים) |
| status | VARCHAR(50) | סטטוס הקובץ (למשל: "מתקבל", "נבדק על ידי AI", "דורש בדיקה אנושית") |
| created\_at | DATETIME | תאריך העלאת הקובץ |
| updated\_at | DATETIME | תאריך עדכון הקובץ |

**3.1.3 טבלת Questions**

| **עמודה** | **סוג נתון** | **תיאור** |
| --- | --- | --- |
| question\_id | INT (PK) | מזהה ייחודי לשאלה |
| user\_id | INT (FK) | מזהה המשתמש שהעלה את השאלה (מפתח זר ל-Users) |
| question\_text | TEXT | טקסט השאלה |
| status | VARCHAR(50) | סטטוס השאלה (למשל: "מתקבל", "נבדק על ידי AI", "דורש בדיקה אנושית") |
| created\_at | DATETIME | תאריך העלאת השאלה |

**3.1.4 טבלת Comments**

| **עמודה** | **סוג נתון** | **תיאור** |
| --- | --- | --- |
| comment\_id | INT (PK) | מזהה ייחודי לתגובה |
| user\_id | INT (FK) | מזהה המשתמש שכתב את התגובה (מפתח זר ל-Users) |
| target\_id | INT | מזהה השאלה או הקובץ שעליהם הגיב (מפתח זר ל-Files או Questions) |
| target\_type | VARCHAR(50) | סוג היעד (File או Question) |
| comment\_text | TEXT | תוכן התגובה |
| created\_at | DATETIME | תאריך כתיבת התגובה |
| points\_awarded | INT | מספר הנקודות שהוענקו למשתמש עבור התגובה |

**3.1.5 טבלת Admin\_Requests**

| **עמודה** | **סוג נתון** | **תיאור** |
| --- | --- | --- |
| request\_id | INT (PK) | מזהה ייחודי לבקשה |
| user\_id | INT (FK) | מזהה המשתמש ששלח את הבקשה (מפתח זר ל-Users) |
| request\_type | VARCHAR(50) | סוג הבקשה (למשל "בדיקה אנושית") |
| request\_text | TEXT | פרטי הבקשה (למשל "בקשה לבדוק קובץ זה") |
| status | VARCHAR(50) | סטטוס הבקשה (למשל "בעבודה", "הושלם") |
| created\_at | DATETIME | תאריך שליחת הבקשה |

**3.2 קשרים בין הטבלאות**

* **Users → Files**: קשר "אחד לרבים", כל משתמש יכול להעלות מספר קבצים.
* **Users → Questions**: קשר "אחד לרבים", כל משתמש יכול להעלות מספר שאלות.
* **Users → Comments**: קשר "אחד לרבים", כל משתמש יכול להגיב על שאלות וקבצים.
* **Files → Comments**: קשר "אחד לרבים", כל קובץ יכול להיות מושא למספר תגובות.
* **Questions → Comments**: קשר "אחד לרבים", כל שאלה יכולה להיות מושא למספר תגובות.
* **Admin\_Requests → Users**: קשר "אחד לרבים", כל בקשה נשלחת על ידי משתמש אחד.

**4. דרישות טכנולוגיות**

**4.1 טכנולוגיות צד שרת**

* **שפה**: .NET
* **Database**: MySQL
* **AI Integration**: שימוש במודל AI לבדיקת קבצים ושאלות (אפשר להשתמש ב-OpenAI או מודלים אחרים)

**4.2 טכנולוגיות צד לקוח**

* **Frontend**: React.js Angular
* **Styling**: CSS/SASS
* **HCI**: חווית משתמש אינטואיטיבית, כולל חיפושים חכמים וגרפיים

**5. חוויית משתמש (UX/UI)**

1. **עמוד העלאת קבצים ושאלות**:
   * כפתור העלאה + שדות עבור שם הקובץ או השאלה.
   * הצגת מספר הנקודות הנדרש להעלאת קובץ או שאלה.
2. **עמוד תגובות**:
   * הצגת התגובות על קובץ או שאלה.
   * הצגת מספר הנקודות שהרוויח המשתמש עבור כל תגובה.
3. **עמוד בקשות לבדיקות אנושיות**:
   * אפשרות למשתמש לשלוח בקשה לבדיקה אנושית.
   * אפשרות למנהל לראות את כל הבקשות ולבצע את הבדיקות.

**6. אבטחת מידע**

1. **אימות משתמשים**: באמצעות JSON Web Tokens (JWT) או OAuth.
2. **הצפנת סיסמאות**: שימוש ב- bcrypt לאחסון סיסמאות בצורה מאובטחת.
3. **מניעת התקפות SQL Injection**: שימוש בפרמטרים (Prepared Statements) בשאילתות SQL.