**מסמך אפיון מערכת סימולציית ראיונות עבודה עם AI**

**1. תיאור המערכת**

**1.1 מטרת הפרויקט**

המערכת נועדה לספק למשתמשים סימולציה של ראיונות עבודה עם שאלות מותאמות אישית לפי תחום התמחותם. המערכת תסייע למחפשי עבודה להתכונן בצורה מיטבית לראיונות, לקבל פידבק על תשובותיהם ולשפר את סיכוייהם להתקבל לעבודה.

**תכונות ויתרונות עיקריים:**

* שאלות מותאמות אישית לפי תחום התמחות המשתמש.
* פידבק אוטומטי על התשובות.
* דוח מפורט עם המלצות לשיפור.
* פלטפורמת ניהול למנהלים לצפייה בסטטיסטיקות והפקת דוחות.

**1.2 קהל היעד**

המערכת מיועדת ל:

* מחפשי עבודה המעוניינים להתכונן לראיונות עבודה בצורה מקצועית.
* סטודנטים ובוגרים טריים ללא ניסיון קודם בראיונות עבודה.
* אנשים המעוניינים לעבור הסבה מקצועית וללמוד תחום חדש.

**1.3 פונקציונליות של המערכת**

המערכת תכלול את הפונקציונליות הבאה:

1. העלאת קורות חיים או בחירת תחום התמחות.
2. יצירת שאלות מותאמות אישית לפי תחום התמחות.
3. מענה על שאלות תוך מדידת זמן.
4. הפקת דוח מפורט עם ניתוח תשובות והמלצות לשיפור.
5. פאנל ניהול למנהלים לצפייה בסטטיסטיקות והפקת דוחות.
6. מערכת הרשאות מבוססת תפקידים (Admin, User).

**1.4 בעיות שהמערכת פותרת**

* מחסור בפלטפורמות חינמיות להתכוננות לראיונות עבודה.
* צורך בקבלת פידבק אובייקטיבי ומיידי על תשובות.
* שיפור יכולות התנסחות וביטחון עצמי.
* הכנה מותאמת אישית לפי תחום התמחות.

**1.5 טכנולוגיות בשימוש**

* **צד שרת (Backend):** ASP.NET Core 8
* **צד לקוח (Frontend):** React.js לאפליקציית המשתמש, Angular 17 לאפליקציית המנהל
* **מסד נתונים:** SQLite
* **AI:** מודל GPT לאיתור שאלות מתאימות וניתוח תשובות
* **Authentication:** JWT
* **אחסון קבצים:** AWS S3

**1.6 מתודולוגיית פיתוח (Agile)**

הפרויקט יפותח במתודולוגיית Agile עם חלוקה ל-4 ספרינטים בני שבוע אחד כל אחד:

|  |  |  |
| --- | --- | --- |
| ספרינט | מטרה | משימות עיקריות |
| 1 | תשתיות וניהול משתמשים | - תכנון מסד הנתונים |
|  |  | - פיתוח API לניהול משתמשים |
|  |  | - פיתוח מסך התחברות והרשמה ב-React |
| 2 | יצירת שאלות מותאמות | - פיתוח מערכת יצירת שאלות ב-AI |
|  |  | - אינטגרציה עם API |
|  |  | - פיתוח ממשק מענה לשאלות |
| 3 | הפקת דוחות | - פיתוח לוגיקת ניתוח תשובות |
|  |  | - פיתוח מערכת הפקת דוחות |
|  |  | - אינטגרציה עם צד לקוח |
| 4 | פאנל ניהול | - פיתוח אפליקציית המנהל ב-Angular |
|  |  | - ניהול משתמשים והרשאות |
|  |  | **- צפייה בסטטיסטיקות** |

**2. עיצוב ממשק משתמש ויזואלי**

**Wireframes**

* מסך התחברות
* מסך שאלות ומענה
* מסך דוח מפורט

**פאנל ניהול למנהלים**

**תרשימי זרימה לתהליכי עבודה**

**תהליך רישום משתמש חדש**

1. הזנת פרטי משתמש.
2. אימות פרטים.
3. שליחת מייל אישור.
4. התחברות ראשונית.
5. הגדרת תחום התמחות.
6. התחלת סימולציה.

**תהליך יצירת שאלות מותאמות**

1. הזנת תחום התמחות.
2. שליחת בקשה ל-AI.
3. קבלת שאלות מותאמות.
4. הצגת שאלות למשתמש.

**3. איפיון תמציתי**

**3.1 רשימת פונקציות המערכת**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| פונקציה | Route | Method | פרמטרים | פלט |
| הרשמה | /api/register | POST | { email, password } | { userId, token } |
| התחברות | /api/login | POST | { email, password } | { token } |
| יצירת שאלות | /api/questions | POST | { domain } | { questions[] } |
| מענה על שאלה | /api/answer | POST | { questionId, answer } | { feedback } |
| דוח מפורט | /api/report | GET | { userId } | { report } |

3.2 אימות והרשאות

* JWT Authentication
* תפקידים במערכת:
  + **User:** מענה על שאלות וקבלת דוחות.
  + **Admin:** צפייה בסטטיסטיקות, ניהול משתמשים והפקת דוחות.

**3.3 אפליקצית ניהול**

* צפייה בסטטיסטיקות משתמשים.
* הפקת דוחות שימוש.
* ניהול משתמשים והרשאות.

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

**טבלת Users**

|  |  |  |  |
| --- | --- | --- | --- |
| שם העמודה | סוג נתונים | מאפיינים | תיאור |
| id | INT | PK | מזהה ייחודי |
| email | VARCHAR(100) | UNIQUE | כתובת אימייל |
| password | VARCHAR(255) | NOT NULL | סיסמה |
| role | VARCHAR(50) |  | תפקיד |
| createdAt | DATETIME | DEFAULT | תאריך יצירה |

**קשרים בין טבלאות**

* אחד לרבים: משתמש → תשובות
* אחד לרבים: משתמש → דוחות

**5. הגדרת סבבי פיתוח**

**ספרינט 1: תשתיות וניהול משתמשים**

**מטרת הספרינט:**

הקמה של תשתית בסיסית למערכת עם התמחות בניהול משתמשים, הרשאות ואבטחת גישה.

**משימות עיקריות:**

1. **תכנון מסד הנתונים**
   * יצירת טבלאות ב-SQLite (Users, Questions, Answers, Reports).
   * הגדרת קשרים בין טבלאות (אחד לרבים בין משתמשים לתשובות ולדוחות).
   * הגדרת מבנה אינדקסים ויחסים (indexing) למבני הנתונים.
2. **פיתוח API לניהול משתמשים**
   * יצירת API עבור הרשמה והתחברות:
     + /api/register: יצירת משתמש חדש עם אימייל וסיסמה.
     + /api/login: התחברות באמצעות אימייל וסיסמה עם יצירת JWT Token.
     + טיפול בשגיאות אפשריות (כמו אימייל כבר קיים או סיסמה לא נכונה).
   * אבטחת הנתונים ו-Encryption של סיסמאות.
3. **פיתוח מסך התחברות והרשמה ב-React**
   * יצירת טופס הרשמה והתחברות בממשק המשתמש.
   * אינטגרציה עם ה-API (שליחת בקשות POST ו-GET).
   * טיפול בהודעות שגיאה והצלחה (כמו הודעות על קבלת קוד אישור או שגיאת סיסמה).
   * יצירת מנגנון לאחסון JWT טוקן ב-LocalStorage לשימוש עתידי.

**תוצאות הצפויות:**

* API פעיל לניהול משתמשים.
* מסך התחברות והרשמה פועל ב-React.
* בסיס נתונים מוכן.

**ספרינט 2: יצירת שאלות מותאמות**

**מטרת הספרינט:**

יצירת מערכת יצירת שאלות בהתאמה אישית לכל תחום התמחות.

**משימות עיקריות:**

1. **פיתוח מערכת יצירת שאלות ב-AI**
   * אינטגרציה עם GPT-3 (או מודל מתאים אחר) לצורך יצירת שאלות אוטומטיות.
   * API שיקבל תחום התמחות ויחזור עם שאלות מותאמות (למשל: POST /api/questions).
   * פיתוח מנגנון לעיבוד תחום התמחות (כגון חיפוש מילות מפתח).
2. **אינטגרציה עם API**
   * פיתוח ממשק API שמקבל בקשות מהלקוח ומחזיר את השאלות המותאמות.
   * יצירת ממשק לניהול השאלות במערכת (כגון שמירה, עריכה, מחיקה).
3. **פיתוח ממשק מענה לשאלות ב-React**
   * יצירת מסך שאלות בו המשתמש יכול לבחור את תחום ההתמחות ולקבל את השאלות.
   * טיימר המודד את זמן התשובה לכל שאלה.
   * אפשרות לשלוח תשובות ולשמור אותן.

**תוצאות הצפויות:**

* ממשק ליצירת שאלות מותאמות ב-React.
* אינטגרציה עם API של GPT לצורך יצירת שאלות.
* מערכת פעילה לשליחת שאלות ומענה עליהם.

**ספרינט 3: הפקת דוחות**

**מטרת הספרינט:**

פיתוח מערכת לניתוח תשובות והפקת דוחות עם פידבק.

**משימות עיקריות:**

1. **פיתוח לוגיקת ניתוח תשובות**
   * יצירת מנגנון לניתוח תשובות המשתמשים.
   * אינטגרציה עם מודל AI (למשל: GPT-3) לצורך הערכת איכות התשובות (כגון הערכת בהירות, רלוונטיות, ונכונות).
2. **פיתוח מערכת הפקת דוחות**
   * יצירת API להפקת דוחות על בסיס התשובות שניתנו.
   * כל דוח יכיל את התשובות עם הפידבק, ציון על תשובה נכונה או לא נכונה והמלצות לשיפור.
   * הצגת הדוחות בפורמט קריא (JSON או HTML).
3. **אינטגרציה עם צד לקוח**
   * פיתוח ממשק ב-React להצגת הדוח למשתמש עם מידע אישי.
   * הצגת דוחות בפורמט ידידותי למשתמש.

**תוצאות הצפויות:**

* API להפקת דוחות.
* דוחות ניתוח תשובות עם פידבק.
* ממשק ב-React להציג את הדוחות למשתמש.

**ספרינט 4: פאנל ניהול**

**מטרת הספרינט:**

פיתוח אפליקציית ניהול שתאפשר למנהלים לצפות בסטטיסטיקות ולנהל משתמשים.

**משימות עיקריות:**

1. **פיתוח אפליקציית המנהל ב-Angular**
   * יצירת ממשק ניהול ב-Angular שיאפשר למנהל להיכנס עם הרשאות גבוהות.
   * הצגת סטטיסטיקות כמו: מספר המשתמשים, אחוזי הצלחה בתשובות, ופעילויות מערכת.
2. **ניהול משתמשים והרשאות**
   * אפשרות למנהל לנהל את המשתמשים: יצירה, מחיקה, עדכון תפקידים.
   * ניהול הרשאות: Admin יכול להיכנס לפאנל הניהול, בעוד שמשתמש רגיל יוכל רק לקבל תשובות ודוחות.
3. **צפייה בסטטיסטיקות**
   * הצגת סטטיסטיקות על פעילות המשתמשים, כמות השאלות שנשאלו, זמן ממוצע על כל שאלה, ועוד.
   * אפשרות להפיק דוחות סטטיסטיים לפיתוח עתידי.

**תוצאות הצפויות:**

* אפליקציית ניהול פעילה ב-Angular.
* פאנל ניהול שמאפשר ניהול משתמשים וסטטיסטיקות.
* אפשרות לצפייה בסטטיסטיקות ולהפקת דוחות ניהול.

**סיכום**

במהלך כל הספרינטים תפותח גרסה עובדת של המערכת עם פונקציות מוגדרות, כך שבסיום כל ספרינט יהיה ניתן לבדוק את התקדמות הפרויקט, לבצע אופטימיזציה ולוודא שהתכונות פועלות כמצופה.