**- "LingoFlow"** לדבר שוטף, ללמוד בכיף!! 📊💬

By Chana Vandervalde

**תיאור המערכת:**

LingoFlow היא מערכת ללמידה עצמאית של שפות עם דגש על הקלטת שיחות ומתן משוב מפורט לשיפור הדיבור והרחבת אוצר המילים. המערכת מאפשרת למשתמש להקליט שיחות, לשמור אותן בהיסטוריה ולחזור להן בכל עת, מה שמאפשר לו להשוות את הביצועים הנוכחיים לאלו שהיו בעבר ולהבחין בהתקדמות ובתחומים שדורשים שיפור.

באמצעות ניתוח מתקדם של ההיסטוריה האישית, המערכת מספקת משוב מותאם אישית – המצביע על שיפורים, מזהה טעויות חוזרות ומציע המלצות ממוקדות לתרגול נוסף. חוויית הלמידה האינטראקטיבית משלבת תרגול שיחות אמיתי, פידבק מיידי בזמן אמת והכוונה להתקדמות אישית, כך שהמשתמש יכול לתכנן את מסלול הלמידה שלו בהתאם לצרכיו האישיים ולחזק את ביטחונו בדיבור השוטף בשפה הנלמדת.

**קהל היעד:**

המערכת מיועדת לאנשים בכל גיל המעוניינים לשפר את ביטחונם בדיבור ולהרחיב את אוצר המילים שלהם, תוך תמיכה במעקב אחרי ההתקדמות האישית והכוונה לשיפור מתמשך.  
כולל:

**-משתמשים פרטיים** המעוניינים לשפר את יכולות השפה שלהם באופן עצמאי ונח.

**-תלמידים וסטודנטים** שמחפשים דרך גמישה ויעילה לשפר את כישוריהם בשפה זרה לצורך לימודים אקדמיים או קריירה מקצועית.

**פונקציונליות של המערכת:**

ומפורט על הגייה, מבנה משפטים ושימוש נכון במילים, ובכך לתרגל שיחות מדומות בצורה מדויקת ומבוקרת.

**בעיות שהמערכת פותרת:**

LingoFlow מציעה פתרון לבעיית חוסר התרגול בשיחות אמיתיות בלימוד שפות. במקום תרגולים פסיביים בלבד, המערכת מאפשרת למשתמש לתרגל דיבור אמיתי עם משוב מותאם אישית על הגייה, זרימה ומבנה משפטים. בנוסף, היא מאפשרת מעקב רציף אחרי התקדמות הלמידה ומספקת מסלול לימוד אישי, המותאם לצרכים של כל משתמש. (מעבר לכך, LingoFlow מציעה חוויית למידה אינטראקטיבית הכוללת סרטוני לימוד, דפי תרגול, בחנים ומטלות יומיות, שמסייעות למשתמשים לבדוק ולהתאים את ההתקדמות שלהם בצורה אקטיבית וממוקדת(.

**טכנולוגיות בשימוש:**

1. **צד שרת (Backend):** - .NET9 לבנייית API מהיר וגמיש.
2. **צד לקוח (Frontend):** React.js לבניית ממשק משתמש אינטראקטיבי.Angular**:** בשימוש לפיתוח ממשקי משתמש עבור מסכים עם ניהול משתמשים והגדרות מערכת.
3. מסד נתונים relational,sql מתאים לאחסון נתונים על משתמשים, קבצים, שיחות ומטלות. יעיל בניהול מערכות גדולות עם נתונים קשורים.
4. **Amazon S3**: -אחסון ענן לאחסון הקלטות דיבור, סרטוני לימוד, קבצים שנוצרו על ידי המשתמשים ומטלות יומיות.
5. AI & Machine Learning -   
   **TensorFlow.js / Speech Recognition API**: כדי לאפשר ניתוח אוטומטי של ההקלטות, תיקון שגיאות דיבור, משוב אישי על שיפור הביטוי הלשוני וההגייה.

**NLP (Natural Language Processing)**: עבור ניתוח טקסטים ממילים שהוזנו על ידי המשתמשים, להפקת משוב על שגיאות או הצעות לשיפור.

1. אופציונלי: אבטחה: **JWT (JSON Web Token)**: לאימות וגישה מאובטחת של משתמשים במערכת, יאפשר חיבור מבוקר עם מסדי הנתונים והAPI-

**מתולוגיית פיתוח Agile**

הפרויקט יפותח באמצעות טכנולוגיית Agile עם חלוקה ל-5 ספרינטים. כך ספרינט יתמקד במשימה עיקרית שונה על מנת להבטיח עבודה גמישה וממוקדת במשימות מרכזיות, כל ספרינט כשבועיים עבודה.  
להלן פירוט של הספרינטים הצפויים בס"ד:

**ספרינט 1-** תכנון תשתיות והגדרת אפיון בסיסי:

🎯 המטרה: הקמת הבסיס לפיתוח המערכת – מסד נתונים Api ותשתיות פרונטד.

* תכנון מסד הנתונים (SQL) למודלים של משתמשים, שיחות והקלטת דיבור לניהול משתמשים.
* פיתוח API ביסי לניהול המשתמשים הרשמה התחברות ואחסון קבצים.
* יצירת תשתיתי לפיתוח צד לקוח בעזרת react & angular

**ספרינט 2 –** ממשק משתמש ראשוני:

🎯 **מטרה:** פיתוח מסכים בסיסיים ויצירת אינטגרציה חלקה בין ה-Frontend ל-Backend.

* פיתוח מסך התחברות והרשמה עבור משמשים ב-angular .
* בניית רכיבי ממשק בסיסיים ב-react.
* אינטגרציה בין צד הפרונטד לבין ה-API לניהול הרשמה והתחברות.
* פיתוח תצוגה ראשונית להצגת נושאי שיחה ואוצר מילים מותאם, כדי להנחות את המשתמש במהלך תהליך הלמידה.

**ספרינט 3 –** תכנים ופיצ'רים עיקריים:

🎯 **מטרה:** הוספת הפיצ'רים המרכזי למערכת, כולל יכולות AI ותוכן לימודי.

* פיתוח מערכת הקלטת דיבור ומענה אוטומטי (AI) באמצעות TensorFlow או Speech Recognition API.

**ספרינט 4 –** שיפורים וייעול:

🎯 **מטרה:** שיפור חוויית המשתמש, הוספת פיצ'רים מתקדמים וייעול הביצועים.

* פיתוח פיצ'ר של משוב אישי והצעות לשיפור על דיבור המשתמש.
* אופטימזציה למערכת למידה מתקדמת (כגון הצגת נתוני שיפור בדיבור עם כל הקלטה).
* חיבור לדיווח על התקדמות וסטטיסטיקות אישי
* בדיקות מערכת ואבטחת מידע.

**ספרינט 5 –** אבטחת איכות ופריסה:

🎯 **מטרה:** בדיקות סופיות והשקת המערכת.

* ביצוע בדיקות מקיפות על יכולת המערכת להתמודד עם קבצים גדולים ונתונים.
* הטמעת גיבוי אוטומטי למידע אישי.
* פריסת המערכת על גבי AWS וייעול קוד.
* עיצוב דף הבית והשיווק כדי למשוך משתמשים חדשים.

כלים לניהול משימות:

**Jir**a לניהול המשימות, תכנון הספרינטים, מעקב אחרי התקדמות ומדדי ביצועים.

**Trello**: לניהול מטלות יומיומיות ובדיקות.

**GitHub/GitLab**: לניהול קוד ושיתוף עבודה עם צוות.

הספרינטים יתמקמו בכל שלב בהתמקדות בפיצ'רים העיקריים ובדיקות עם משתמשים בשטח כדי לוודא שהממשק קל לשימוש והפיצ'רים פועלים כמצופה.

**עיצוב ממשק משתמש ויזואלי**

Wireframes (שרטוטי מסכים) לכל מסך מרכזי:

* מסך כניסה- והרשמה שדות למייל סיסמא ושם משתמש וכפתורי התחברות/ הרשמה.
* **מסך ראשי -תרגול שיחה -** חלוקה לאזור תמלול, הצעות לנושאי שיחה ואוצר מילים תואם, כפתור הקלטה ומשוב.
* **(מסך ניהול אדמין** (?): ניתוח שימוש במערכת, ניהול משתמשים ותכנים.)

**טבלאות הנתונים:**

משתמשים (רבים לרבים לשיחות ליסט של שיחות.)  
 עמודות: ID-זיהוי אוטומטי, שם משתמש, מייל ,סיסמא ורמת ידע לפי המשוב בAI.

נושאי שיחה יחיד לרבים למילים ID הval יש ליסט של מילים  
 עמודות: מפתח מזהה, מחרוזת הנושא.

אוצר מילים יחיד לרבים לנושאי שיחה.

עמודות: מפתח מזהה-ID, ערך באנלית וערך עברית (string)

שיחות/ הקלטות   
 עמודות: מפתח מזהה-ID, נושא השיחה, אורך השיחה, תאריך שליחת השיחה, שם המשתמש שמקליט את השיחה.

**קשרים בין הטבלאות:**

Users 🡨🡪 Subjects  
**one-to-many** לכל משתמש יש ליסט של כל נושאי השיחה שדיבר.

Users 🡨🡪 Words

One-to-many לכל משתמש יהיו ליסטים של מילים עליהן תתבסס ההקלטה שלו.

Subject 🡨🡪words

One-to-many לכל נושא שיחה יש רשימה של הרבה מילים שמתאימות רק לנושא הנוכחי.

Users 🡨🡪 SpeechRecord

Many-to-many לכל משתמש יכולים להיות הרבה נושאי שיחה. וכל נושא שיחה יכול להופיע אצל כמה משתמשים.

**פונקציות מערכת:**

**הוספת משתמש חדש-** – AddUser(User user)  
**Route:** POST /api/users

📌 מאפשר יצירת משתמש חדש במערכת

.

-כניסת משתמש קיים- LoginUser(LoginReq req)

Route POST /api/users/login

📌 אימות פרטי משתמש והחזרת טוקן התחברות (JWT)

-בחירת נושא לשיחה- .ChooseSubject()

**Route:** GET /api/subjects

📌 מחזיר רשימת נושאים זמינים לתרגול שיחה.

-קבלת אוצר המילים בהתאם לנושא הניבחר - GetVocabulary(int subjectId)

**Route:** GET /api/subjects/{subjectId}/words

📌 מחזיר מילים עבור נושא מסוים.

- הקלטת שיחה - StartRecording(int userId, int subjectId)

**Route:** POST /api/conversations/start

📌 מאפשר התחלת הקלטת שיחה בנושא נבחר.

**-שמירת הקלטה בסיום השיחה** - – SaveConversation(Conversation conversation)

**Route:** POST /api/conversations/save

📌 שומר את ההקלטה בבסיס הנתונים ומחזיר מזהה שיחה.

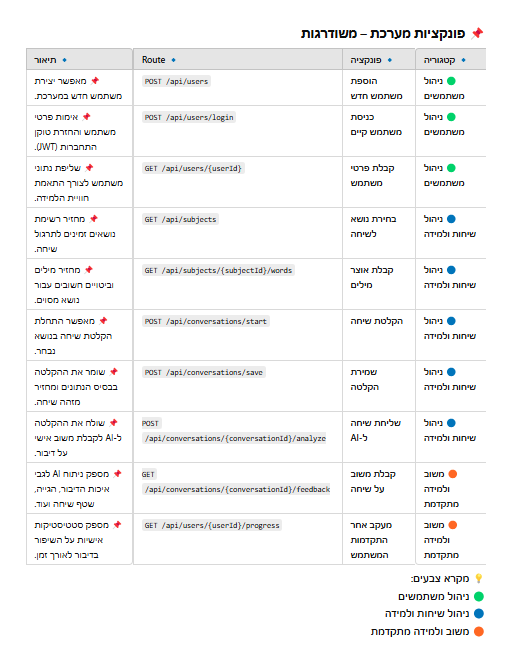
**-שליחת שיחה לניתוח AI** – AnalyzeConversation(int conversationId) -   
**Route:** POST /api/conversations/{conversationId}/analyze   
📌 שולח את ההקלטה ל-AI לקבלת משוב אישי על דיבור.

**-קבלת משוב על שיחה** – GetFeedback(int conversationId)  
**Route:** GET /api/conversations/{conversationId}/feedback   
📌 מספק ניתוח AI לגבי איכות הדיבור, הגייה, שטף שיחה ועוד.

**אפליקציית הניהול:**

**הוספה/עדכון/מחיקת משתמשים:** אפשרות להוסיף משתמשים חדשים, לערוך פרטי משתמשים קיימים ולמחוק משתמשים לא פעילים או לא רלוונטיים.

ניהול תוכן – המנהל יכול להעלות תוכן חדש כגון סרטוני לימוד והוספת נושאי שיחה וכן אוצר מילים.



USERS:

פונקציית קבלת משתמשים:

קבלת משתמש לפי ID

SUBJECT:

קבלת כל הנושאים וקבלת אוצר המילים לפי ID של הנושא

CONVERSATION:

חיפוש שיחה לפי תאריך ההקלטה שלה

FEEDBACK:

חיפוש משוב לפי תאריך

להוסיף את הפוסטמודל

קישור שיכול לעזור בענין ההקלטה:  
<https://blog.logrocket.com/how-to-create-video-audio-recorder-react/>