מדעי המחשב והמידע
Computer & Information Sciences
המרכז האקדמי רופין
Ruppin Academic Center

<28.02.26 >

מסמך מסכם לפרויקט גמר
בהתמחות פיתוח מערכות תוכנה
במחלקה למדעי המחשב והמידע

הפקולטה להנדסה, המרכז האקדמי רופין

מוגש ע"י:

< נבו יפלח 211433339

ליאל ירדני 211339098

אדיר בן דיין 318189917>

מנחה/י הפרויקט:

< ד"ר רינה צביאל-גירשין>

1 תקציר מנהלים

1.1 תקציר מנהלים

פרויקט "מערכת AI קולית - רופין" מציג פתרון טכנולוגי חדשני לממשק השירות הסטודנטיאלי במרכז האקדמי רופין. כיום, סטודנטים נתקלים בקושי וסרבול באיתור מידע אקדמי ומנהלתי מדויק מתוך חוקת הלימודים, במיוחד אוכלוסיות עם מגבלות אוריינות דיגיטלית או פערי שפה. מטרת הפרויקט היא הנגשת המידע המוסדי בצורה טבעית, שקופה מהירה ואמינה באמצעות החלפת ממשק המשתמש המסורתי בשיחה קולית מלאה (Voice-First Application).

המערכת פותחה בארכיטקטורת מבוזרת (Microservices) המשלבת חזית (Frontend) נגישה ב-React, שרת תהליכי ב-Python ומסד נתונים SQL Server מקומי. הליבה הטכנולוגית של הפרויקט מתבססת על שילוב מודלי שפה עמוקים מבית OpenAI (Whisper לפיענוח דיבור, GPT-4 להסקת מסקנות, ו-TTS לסינתוז קולי). על מנת להבטיח אמינות מוחלטת ולמנוע "הזיות" (Hallucinations) המאפיינות מודלי שפה חופשיים, מומשה טכניקת RAG (Retrieval-Augmented Generation). שיטה זו מאלצת את המערכת לבצע חיפוש לוקאלי מאובטח מול מאגר הידע של רופין טרם ייצור התשובה, ומנטרלת חשיפת מידע שגוי.

הפרויקט עמד בהצלחה ביעדיו: סופק מענה חזותי מרגיע ("העץ המחיה"), תמיכה חלקה בהמרת שרת לעברית ולערבית, זמני השהייה (Latency) קצרים מ-3 שניות לאינטראקציה, ולוג אנליטיקה אנונימי לשיפור השירות על ידי האדמין. מסמך זה סוקר בהרחבה את סקר הספרות, הבחירות הארכיטקטוניות (ERD ו-Sequence), אפיון הדרישות (UML), והביצוע בפועל של כל שכבות המערכת.

1.2 הגורם המבצע

- חברי הצוות: נבו יפלח (211433339), ליאל ירדני (211339098) ואדיר בן דיין (318189917).
- מנחים אקדמיים: ד"ר רינה צביאל-גירשין.
- פרטי ארגון ואנשי קשר: המרכז האקדמי רופין.

2 הצגת הבעיה/הצורך

הבעיה המרכזית עימה המערכת מתמודדת היא חוסר הטבעיות והמוגבלות של ממשקי התקשורת הנוכחיים מול מודלי בינה מלאכותית יוצרת. רוב האינטראקציות מתבצעות באופן טקסטואלי בלבד, דבר היוצר מחסום טכנולוגי עבור קהלים מסוימים (כבדי ראייה, ילדים, אנשים עם אוריינות טכנולוגית נמוכה) ומקשה על שילוב AI ברובוטים חברתיים ופיזיים. בנוסף, מודלי AI גנריים נוטים לייצר תשובות כלליות, לא מדויקות, או אפילו

בגדר "הזיות" (Hallucinations) כשנשאלים אודות פרטים ספציפיים על ארגונים (למשל, מוסד אקדמי, נהליו, תנאי הקבלה או שירותי הקמפוס).

3 רקע לבעיה: תיאור השוק/הארגון והסביבה העסקית

סביבת היעד להטמעת הפרויקט היא מגזר ההשכלה הגבוהה, ובפרט "המרכז האקדמי רופין". בסביבה אקדמית מקומית, קיים צורך מתמיד במתן מענה ושירות לסטודנטים ולמועמדים בטווח רחב של נושאים (שכר לימוד, מלגות, מסלולי לימוד, מעונות וכד'). אופי הסביבה מחייב מענה מהיר, מדויק ואמין הנסמך אך ורק על המידע הרשמי של המוסד. אופי התעשייה ותקנונים: מגזר זה כפוף לרגולציות מחמירות של נגישות חומרית ודיגיטלית (חוק שוויון זכויות לאנשים עם מוגבלויות), המחייבות הנגשת מידע בצורות אקוסטיות וויזואליות. מדדי ההצלחה בענף זה נמדדים לעיתים קרובות דרך סקרי שביעות רצון סטודנטים וזמינות המידע.

3.1 תיאור השוק/הארגון והצורך במערכת

אופי השוק/הארגון והסביבה העסקית

סביבת היעד להטמעת הפרויקט היא מגזר ההשכלה הגבוהה, ובפרט "המרכז האקדמי רופין". בסביבה אקדמית יש דרישה מתמדת למענה לסטודנטים ומתעניינים בסוגיות שונות (שכר לימוד, מכינה קדם-אקדמית, מלגות ומעונות). אופי הארגון מחייב מתן תשובות רשמיות וקונקרטיות באופן מידי. אופי הסביבה הוא דינמי, עמוס בפונים, וחשוף לאוכלוסיות שונות.

חלק גדול מגדודי הארגונים האקדמיים נשען היסטורית על מוקדים אנושיים שעמוסים בפונים, או על צ'אטבוטים ישנים הבנויים על "עץ החלטות" (Rule-Based). טכנולוגיה זו גורמת לסרבול ולקשיות במתן השירות הטקסטואלי. מוסדות ההשכלה הגבוהה הם השחקנים המרכזיים הזקוקים כיום לפתרונות AI אינטרפרטיביים ומשלימים שיתנו מענה טבעי ואנושי.

אופי התעשייה ומדדים חשובים בענף המודדים את הצלחת העוסקים בענף תעשייה זו נמדדת פעמים רבות באיכות השירות ובזמינות המידע. מדדים חשובים להצלחה במערכות מול קהל היעד כוללים: אחוזי נטישה נמוכים של משתמשים שהתייאשו ממערכת חכמה, מהירות (Latency) בסגירת מעגל שאלת הסטודנט, וציונים גבוהים בסקרי משוב (אחוזי סגירת פניות מוצלחות ללא צורך במעורבות מזכירות/מנהל).

תקנונים ואילוצים ענפיים ורגולטוריים שיש להתחשב בהם ענף ההשכלה והשירות הציבורי כפוף לרגולציות נגישות מחמירות (כדוגמת חוק שוויון זכויות לאנשים עם מוגבלויות). על ארגונים להנגיש מידע באפיקים לא ויזואליים או אקוסטיים בלבד. מערכת חכמה וחליפית

הממירה דיבור לטקסט וטקסט לדיבור (STT / TTS) מהווה פתרון נגישות רגולטורי מושלם לאנשים עם ליקוי ראייה או דיסלקציה.

מאפייני קהל היעד של לקוחות ומשתמשי המערכת

המערכת מיועדת למגוון רחב של משתמשים בפילוח דמוגרפי ברור, המחולק בעיקר לשלוש קבוצות מרכזיות בהקשר של המרכז האקדמי רופין:

- מועמדים בשלב המכינה וקדם-אקדמיה: משתמשים צעירים המחפשים מידע מכוון על תנאי קבלה, מסלול מכינה ותנאי הקמפוס.
- סטודנטים לתואר ראשון (גילאים 20-30): קהל טכנולוגי שצורך מידע מסביב לשעון אודות מלגות, דעות קבלה וכו' אשר מצפה לאינטראקציה חלקה ומהירה בעברית/ ערבית.
- סטודנטים לתואר שני/ לומדים בוגרים (גילאי 30+): קהל המצפה לתקשורת בטון המכבד את ניסיונו ועם אופי שיחה מותאם.

** משתמשי קצה בעלי מוגבלויות: המערכת נותנת מענה ישיר לכבדי ראייה על ידי שימוש ואפשרות להפעלה (TTS ו- STT) בממשק קולי מלא של הקראה אוטומטית ללא צורך לקרוא מהמסך.

פתרונות חלופיים קיימים

טבלה 1. חלופות

שם החלופה	תיאור החלופה	יתרונות ותועלות	חסרונות וחולשות	לינק למקור מידע	הערות
Rule-Based Chatbot	בוטים מסורתיים שמוטמעים באתרי אוניברסיטאות , הפועלים על בסיס עץ החלטות קשיח.	שמירה על גבולות גזרה נוקשים של מידע רשמי בלבד. זול ופשוט לפיתוח.	תסכול רב מצד משתמשים כשהשאלה יוצאת מחוץ לתסריט הצפוי. היעדר הנגשה קולית לחלוטין. תקשורת לא אנושית.	https://cloud.google.com/dialogflow	
מודל GenAI גנרי ללא מעטפת אקוסטית וללא מנוע RAG	שימוש ישיר ב-API של חברות AI ללא "אימון" או התאמה לארגון הספציפי (ChatGPT)	יכולת שיחה טבעית ברמה אנושית ויצירת סביבת שיח אמולטיבית מעולה	מספק מידע גנרי העלול לכלול אלמנטים לא רלוונטיים או "מומצאים" לגבי מוסדות ההשכלה. אין תמיכה נגישה ואקוסטית	https://chatgpt.com	
שימוש ברמקולים חכמים	מכשירים פיזיים המספקים חווית TTS/STT מובנית לפקודות שמע (Alexa)	חווית נגישות ו-STT/TTS ברמה גבוהה	תשתית סגורה שאינה פתוחה באופן חופשי לשילוב ואינטגרציה בסביבות ארגוניות המכילות מידע ספציפי בעברית	https://www.amazon.com/echo	

<u>סיכום החלופות</u>: החלופות לוקות בחסר בין אם ביכולותיהן להבין הקשר ארגוני מקומי, לאפשר גמישות שפה וגלישה אינטואיטיבית לשיחה טבעית, או לשמש אוכלוסיות עם מוגבלויות ראיה ואוריינות דיגיטלית.

3.2 PAINS AND GAINS

- Pain:
  - גלישה באתרי מוסדות וחיפוש מידע דורשת אוריינות דיגיטלית, זמן, יכולת קריאה של כללי תקנון רבים ולא מאפשרת מענה למוגבלויות ראייה.
  - פתרונות אוטומציה נוכחיים לרוב תומכים באנגלית בלבד ואינם רהוטים בדיאלקטים בעברית/ ערבית.
  - נוטים לספק תשובות מתחמקות או פתרונות AI שקריות לשאלות מוסדיות מה שעלול לבלבל ולהטעות סטודנטים.
- Gains:
  - הנגשה קולית טבעית רציפה ודו-כיוונית.
  - תשובות ענייניות וקונקרטיות לשאילתות (כמו שכר לימוד ותנאי קבלה) תוך הקפדה על אמינות מכורח מאגר הידע המוסדי האמין שמזין אותם.
  - תמיכה מובנית בשיחה בעברית/ ערבית והבנה של ניבי קול ומבטא מקומי.

3.3 מטרות ומדדים

- נגישות שפתית וזיהוי קול: רמת דיוק בהמרת המילים מזיהוי קולי לעברית טקסטואלית ולערבית (Word Error Rate < 10%).
- זמני תגובה מותאמים: המערכת תוכל לקבל את סיגנל הקול מהמשתמש, לעבד בחזרה לפלט אקוסטי, לתרגם אותן, לשלוח ולנגן זאת חזרה למשתמש תחת פרק זמן קצר המאפשר "דיאלוג אנושי".
- עומס ואבטחה: Rest API מאובטח המסוגל לטפל בפניות מרובות ע"י אסינכרוניות ואחסון היסטוריית שיחות במאגר מאובטח המנותק מיחידת העיבוד הוויזואלית.
- דיוק בשליפת המידע הארגוני: המערכת תספק תשובה מבוססת מהמידע הקיים במאגר בלבד מבלי להמציא מידע שקרי.

3.4 שיטות לאיסוף נתונים

איסוף הנתונים לפרויקט התמקד בהשגת ידע קונקרטי, רשמי ועדכני אודות המרכז האקדמי רופין, על מנת להזין את מנוע ה-RAG (Retrieval-Augmented Generation) ולמנוע "הזיות" של המודל. מקורות המידע המרכזיים ששימשו לפרויקט כללו: קבצי PDF רשמיים של המוסד, תקנוני שכר לימוד, מתווים למכינות אקדמיות (קדם-אקדמיה), תנאי קבלה לפקולטות השונות, ומידע אודות שירותי הקמפוס (מעונות, דקנאט הסטודנטים וכד'). הנתונים נאספו בצורה ידנית ואוטומטית מתוך פרסומי המוסד. בנוסף, נעשה שימוש מושכל בכלי AI (כגון מודל GPT) ככלי עזר טכני לבקרת תזרים הנתונים (Data Pipeline) שנוצר ולעיבוד ראשוני של

קבצי הטקסט הארוכים למקטעים (Chunks) שניתנים לאחזור מהיר ממסד הנתונים בעת קריאת RAG. הנתונים הגולמיים ודוגמאות לתשובות שעובדו באמצעות ה-API מוצגים בנספחי המסמך.

4 תיאור ראשוני של הפתרון המוצע והאתגרים שבדרך

הפתרון המוצע הוא מערכת קולית מתקדמת (Voice-to-Voice Wrapper) הפועלת בדפדפן, המקשרת בין המשתמש לבין בינה מלאכותית מבוססת מודל שפה גדול (LLM) תוך שמירה על הקשר (Context) ארגוני מוגדר. המערכת תאפשר למשתמשים להקליט שאלה קולית בעברית (או בערבית), תמיר אותה לטקסט (באמצעות מודל Whisper), תנתח את השאילה מול מאגר המידע המקומי של הארגון (RAG), ותשמיע בחזרה את התשובה בצורה אנושית וזורמת (באמצעות מודל TTS של OpenAI).

- משתמשים עיקריים: סטודנטים, מועמדים ואנשים עם מוגבלויות ראייה המחפשים הכוונה. מנהלי מערכת יוכלו לצפות ביומני השיחות לבקרת איכות ואיסוף מידע דמוגרפי.
- מידע עיקרי: המערכת תשמור את מאגר הידע הגנרטיבי של מוסד רופין ואת פרוטוקול השיחות מול המשתמש למטרות ניטור.

האתגרים האלגוריתמיים הטכנולוגיים:

- האתגר באחזור המידע: כיצד לחלץ את הפסקה המדויקת ביותר מתוך שורות ה- SQL של המידע של רופין בזמן מינימלי.
- האתגר בשפה: מודלים אקוסטיים גנריים נוטים להתקשות בפענוח והקראה מדויקת של עברית/ ערבית עקב כיווניות (ימין לשמאל) והגייה נכונה.
- האתגר של מהירות תגובה: ניהול הפעלת שרשרת של 3 פניות API כבדות (STT, TTS, LLM) בצורה אסינכרונית תוך שמירה על זמן תגובה המדמה שיחה אנושית זורמת

5 תכנית עבודה של הפרויקט

פיתוח הפרויקט נחלק למספר שלבים מתודולוגיים על מנת לעמוד ביעדי הזמן ולפתור את האתגרים בצורה מדורגת:

א. תשתיות והוכחת היתכנות: הקמת השרתים (Node.js / C#), בניית חזית ה-UI הבסיסית ב-React, וחיבור לממשקי OpenAI לבחינת איכות ה-STT וה-TTS בעברית. הושגו תוצאות ראשוניות טובות עם השהייה סבירה.

ב. בניית המאגר וניהול הקשר (RAG): הקמת מסד הנתונים SQL Server, כתיבת הפרוטוקול ועיבוד הנתונים של נהלי רופין. כתיבת אלגוריתם ב-C# שחוקר ועוקף הזיות של ה-LLM. זהו האתגר הקשה מכולם עמו המערכת התמודדה בהצלחה.

ג. בקרת הרשאות ושמירת נתונים: הקמת מנגנון ניהול המשתמשים (Admins / סטודנטים), ויצירת טבלאות Session לניטור קהל היעד (לפי קבוצות גיל וצרכים).

ד. שדרוג חווית משתמש ואנימציה: מעבר מ-UI סטטי וריק לעיצוב מרשים מבוסס HTML Canvas המסנכרן את התנודתיות האקוסטית ואת סמליל המוסד (Live Logo Animation) תוך עמידה בדרישות הנגישות.

6 תיאור מפורט של דרישות המערכת

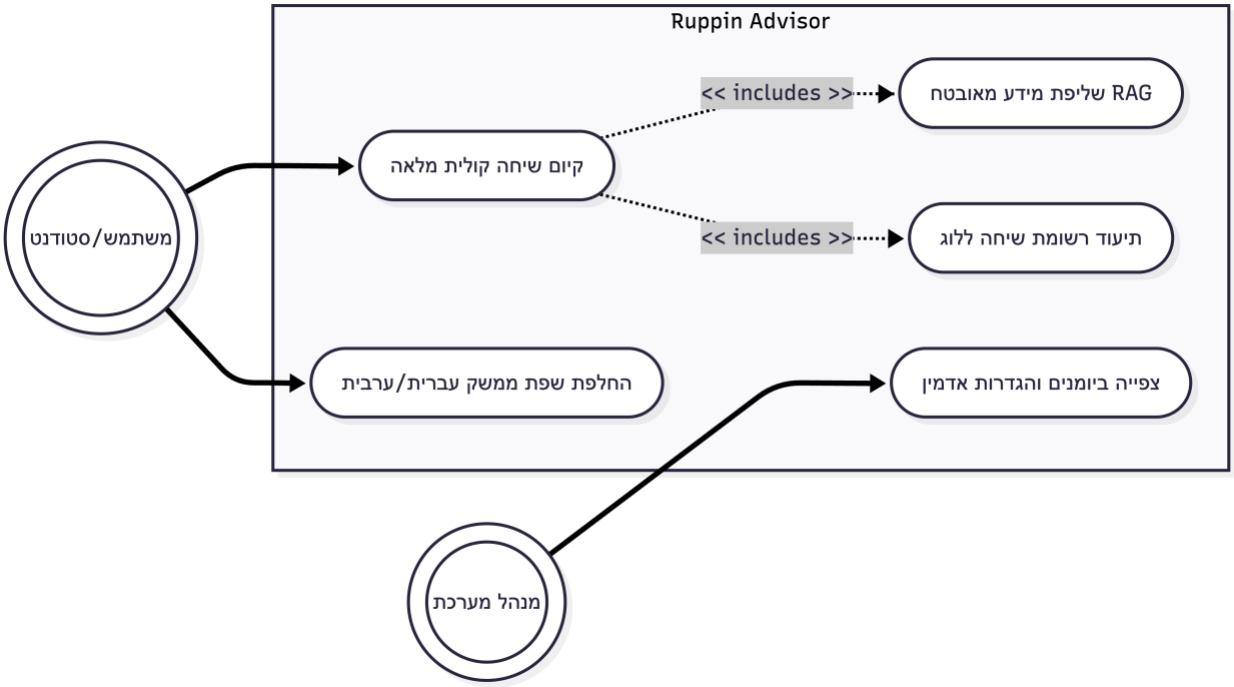
6.1 ייצוג מילולי של הדרישות הפונקציונאליות/הלא פונקציונאליות

טבלה 2. דרישות

מודול/ משתמש	סוג הדרישה (פונק/ לא פונק) ואיזה טיפוס	מס' דרישה	נוסח הדרישה	הערות
ממשק משתמש (UI) / סטודנט פונה	פונקציונלית / תפעולית	1	הקלטה וזיהוי קול: המערכת תאפשר קלט קולי ישיר (מהמיקרופון), ותזהה אוטומטית את השפת השיח (עברית/ערבית) מבלי לדרוש הגדרת שפה ידנית.	
שרת (Backend) - מודל Whisper	פונקציונלית / ממשק חיצוני (API)	2	המרת STT: המערכת תבצע המרה טקסטואלית מדויקת של אות השמע לטקסט תוך שמירה מלאה על כיווניות של שפות מימין שמאל (RTL).	
שרת (Backend) - מנוע C# ו-SQL	פונקציונלית / תפעולית	3	ניהול הקשר מקומי (RAG): במענה לשאלה, תיגש המערכת למסד נתונים פנימי של המוסד, תשלוף את ההקשר הרלוונטי ותשלב אותו ב-Prompt הנשלח ל-LLM.	
שרת (Backend) - GPT-4	פונקציונלית / ממשק חיצוני (API)	4	ניסוח תשובה חכמה: סנכרון צימוד נתוני ה-RAG למודל השפה הגדול שינסח תשובה קריאה-קולית, אנושית וטבעית.	
שרת (Backend) - OpenAI TTS	פונקציונלית / ממשק חיצוני (API)	5	תשובה אקוסטית (TTS): המערכת תשמיע בחזרה למשתמש את הניסוח הכתוב באמצעי קול טבעיים המותאמים להקראה צלולה.	
ממשק ניהול (Admin UI) / מנהל מערכת	פונקציונלית / תפעולית	6	הגדרות Admin: איש הצוות בארגון יוכל לגשת ל-UI ייעודי כדי להגדיר ציפיות גיל (כדי להתאים את רמת שיח	

			התשובות) ולצפות ביומנים השמורים.	
מערכת כללית	לא פונקציונלית / ביצועים (Latency)	7	השהייה וזמני תגובה מינימליים : זמן ההשהיה מסיום משפט המשתמש ועד תחילת ההקראה האקוסטית הראשונית יישמר לזמן מינימלי סביב ה-4 שניות.	
מודל השפה (LLM)	לא פונקציונלית / אמינות ודיוק	8	מניעת הזיות (Hallucinations) : המערכת תספק רק מידע שניצל והוצלב (ולידציה) לעומת ה-RAG המקומי, ונאסר עליה לספק מידע מחוץ לשלד האקדמי.	
ממשק משתמש (UI)	לא פונקציונלית / נגישות ושימושיות	9	רשת ביטחון לנגישות אודיו למסך : המערכת תאפשר שימוש עצמאי לחלוטין וללא תלות במסך ויזואלי מצד המאזין (קריטי לכבדי ראייה).	
מסד נתונים (SQL Server)	לא פונקציונלית / אבטחה ופרטיות	10	הגנת נתונים אנונימיים : אגירת יומני השיחות (Logs) לצרכי ניהול המערכת תיעשה בצורה אנונימית ללא אגירת תעודת זהות של הסטודנט הפונה.	

6.2 ייצוג דרישות פונקציונאליות באמצעות תרשימי USE CASE



ספציפיקציה של USE CASES

שם ה Use case	קיום שיחה קולית מלאה
תיאור קצר	המשתמש מקליט שאלה קולית ומקבל בחזרה התייחסות למסדי נתונים בפורמט קולי.
שחקנים	משתמש הקצה (סטודנט / משתמש אנונימי).
תנאים מקדימים	המשתמש הגיע למסך הראשי ואפשר גישה למיקרופון הדפדפן להקלטה.
תנאים מאוחרים	שיחה נשמרה בלוג באופן אנונימי. המשתמש מעודכן במידע.
טריגרים	לחיצה קולית על מנגנון ה-Record או קריאה לזיהוי קול.
תרחיש מוצלח	1. מערכת מקבלת קובץ שמע. 2. פלטפורמה ממירה לטקסט (STT). 3. השרת מתשאל את ה-RAG.

	4. טקסט מוחזר ומשודר חזרה למשתמש (TTS).
תרחיש אלטרנטיבי חלופי/כשלון	נפילת שרת בהמרת קול לאינטרנט -> קריסת תכנון, הודעה תוצג למשתמש לנסות שנית.

שם ה Use case	החלפת שפת ממשק עברית/ערבית
תיאור קצר	שינוי של שפת הדיבור והזיהוי של המכשיר.
שחקנים	משתמש הקצה.
תנאים מקדימים	המערכת במצב המתנה במסך הבית של היישום.
תנאים מאוחרים	כל פרמטרי הזיהוי מועדכנים לתוצרי שפה (לדוגמה: עברית).
טריגרים	לחיצה יזומה על כפתור השפה (Language Toggle).
תרחיש מוצלח	1. קליטת הלחיצה ב-Client. 2. ריענון ה-State של הממשק. 3. הגדרות המערכת בציפייה לקלט הבא מעודכנות ופרומפט מותאם ספציפי משתנה (HE/AR).
תרחיש אלטרנטיבי חלופי/כשלון	אין. כשלון רגעי שלא טוען שפות בלבד ישאיר את המצב על ברירת מחדל אנגלית.

שם ה Use case	שליפת מידע מאובטח RAG
תיאור קצר	אלגוריתם שליפה של הקשר רלוונטי לפי שאילתת קול מוקלט.
שחקנים	המערכת (System) באמצעות שרת Backend.
תנאים מקדימים	סטרינג מילולי מוכן שהומר מקול המשתמש שנקלט.
תנאים מאוחרים	נוצר Context ממוקד ל-LLM המבטיח תשובה מנומקת מנהלי רופין האמיתיים.
טריגרים	כניסה ל-Controller הראשי של השפה בסגירת UC1.
תרחיש מוצלח	1. ביצוע שאילתת וקטור/SQL לנהלים ולחוקי המוסד בספריית DB. 2. זיהוי Paragraph מדויק והחזרתו למנוע Python.
תרחיש אלטרנטיבי חלופי/כשלון	מילת מפתח לא נמצאה או לא סופק מידע קרוב - > LLM לא מקבל כלום ועונה "איני יודע (מידע לא נשמר אצלי)".

שם ה Use case	תיעוד רשומת שיחה ללוג
תיאור קצר	תיעוד הפסקה הכתובה ושאלת המשתמש במסד נתונים מאובטח ובלי פרטים מזהים.
שחקנים	מנהל המערכת (בקצה השני) אוטומטיזציה של שרת (System).
תנאים מקדימים	מסירת התשובה האקוסטית נשלמת (Post-Action).
תנאים מאוחרים	הרשומות מוצגות בהעברת ממשק תצוגה עתידי לאדמין.
טריגרים	אישור מסירת השיחה מהשרת.
תרחיש מוצלח	1. בניית האובייקט (JSON/SQL Object) של היסטוריית השיחה. 2. הזרקתו לשרת מסד הנתונים ChatSessions.
תרחיש אלטרנטיבי חלופי/כשלון	מסד הנתונים שומר על נעילה -> נפילת Insert. (המערכת לא מתריעה ללקוח שקבל תשובה זה כשל שקוף).

שם ה Use case	צפייה ביומנים והגדרות אדמין
תיאור קצר	סקירה ומעקב של כלל השיחות לטיוב יכולות ותצוגה בטבלאות ניהול.
שחקנים	מנהל מערכת.
תנאים מקדימים	כניסה פריבילגית אל המערכת (Admin Rights) לממשק ניהולי סגור.
תנאים מאוחרים	שינוי דינמי בקונפיגורציה או משתמש התנתק.
טריגרים	בקשת מסך כגון "Show logs".
תרחיש מוצלח	1. אדמין מקיש סיסמא בממשק הניהול. 2. שליפות GET של ה-API מביאות את כל תיוג המידע. 3. ביצוע פעילויות שליטה על שעות הגיל ופעילות העסק לפי בחירות קונקרטיות המעדכנות DB LIVE.
תרחיש אלטרנטיבי חלופי/כשלון	אין הרשאות למראה עמוד -> מעבר Redirect מהיר יעיף משתמש החוצה.

7 תיאור מפורט של הפתרון/השיטה – תיכון (design)

ארכיטקטורת המערכת פותחה בתצורת שרת-לקוח (Client-Server) מבוזרת, תוך יישום עקרונות תכנות מונחה-עצמים, תקשורת אסינכרונית, ומיקרו-שירותים. הפתרון מחולק לשלוש שכבות לוגיות מובחנות על מנת להבטיח מודולריות, אבטחת מידע, ותחזוקתיות גבוהה.

- **<u>שכבת המידע:</u>** שכבת המידע מנוהלת באמצעות מסד נתונים רלוציוני מסוג SQL Server. שכבה זו אמונה על שמירת מאגר הידע (RAG) וניהול יומני המערכת. תיאור ישויות מרכזיות:
  - <u>טבלת הידע (NLA_RupppinKnowledge)</u>: טבלה זו מכילה את מלוא החוקה והנהלים של רופין, בחלוקה למקטעים (Chunks). כל רשומה כוללת מילות מפתח, קטגוריה (למשל "שכר לימוד", "מכינה"), ואת טקסט התשובה הרשמי. טבלה זו משמשת את מנוע ה-RAG.
  - <u>טבלת יומני שיח (NLA_ChatSessions)</u>: טבלה המיועדת לבקרת ממשק הניהול (UC4, UC5). המערכת שומרת באופן אנונימי משתנים כגון תאריך ושעה, משך הדיבור, השאלה שזוהתה כטקסט, והתשובה שהפיקה המערכת.

NLA_RuppinKnowledge		
int	DocumentID	PK
string	Keywords	
string	Category	
string	OfficialProcedureText	

AdminUser		
int	AdminID	PK
string	Username	
string	PasswordHash	

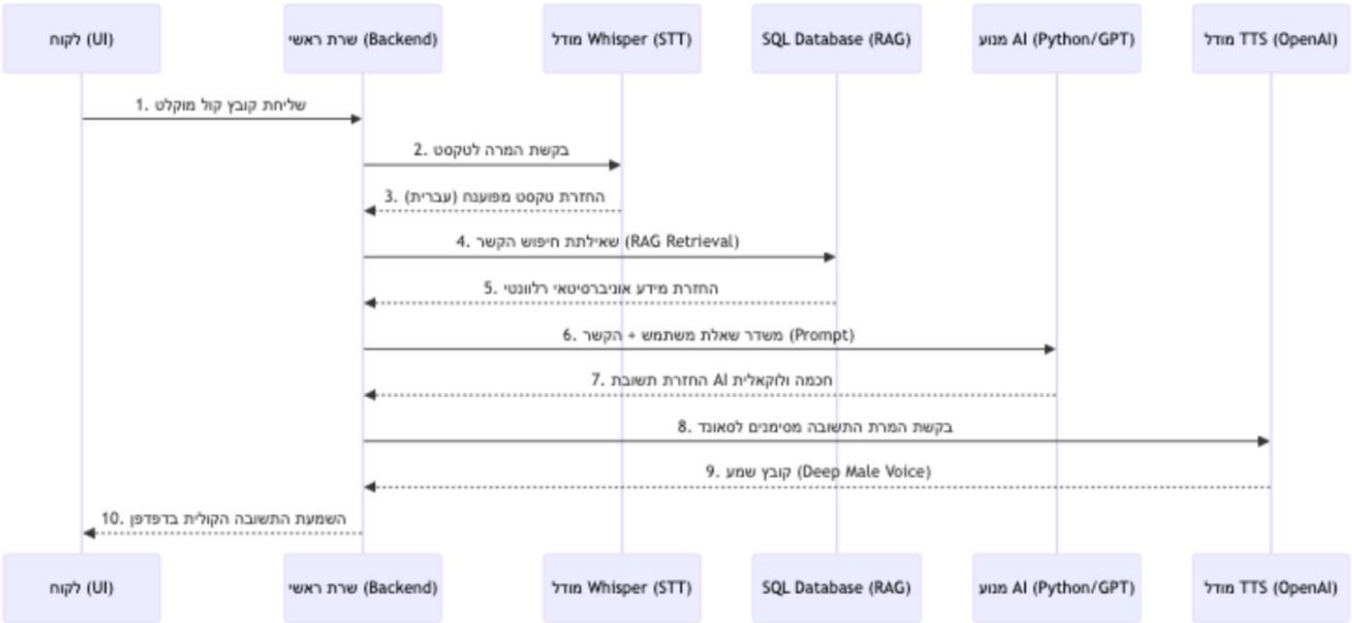
Views & Audits

NLA_ChatSessions		
int	SessionID	PK
datetime	Timestamp	
string	UserQuestionTXT	
string	AIResponseTXT	
string	LanguageContext	

- **<u>שכבת העיבוד והאלגוריתמיקה:</u>** שכבת העיבוד אחראית על סנכרון המידע והבינה המלאכותית. היא בנויה מצד צד שרת ראשי (Node.js/C#) לצד שרת AI ב-Python.

תיאור אלגוריתמי מרכזי: שרשרת RAG והמרות קוליות האתגר המרכזי האלגוריתמי הוא מניעת "הזיות" (Hallucinations) ושמירה על זמני תגובה מינימליים למרות השימוש ב-3 API שונים.

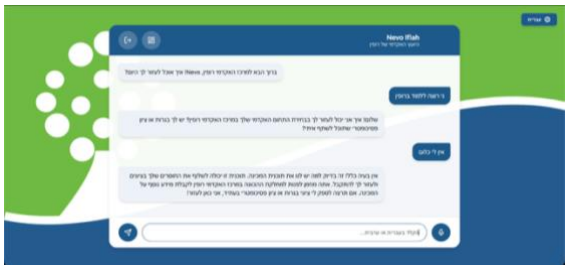
  - הלקוח שולח קובץ קול קומפקטי
  - השרת פונה ל- OpenAI API עם מודל Whisper לביצוע STT תוך קינפוג כיוונית השפה ותיעדוף זיהוי עברית/ ערבית.
  - השרת מבצע אלגוריתם שליפה (RAG): חיפוש מילות מפתח מתוך שאילתת המשתמש מול טבלת ה-Knowledge ב-DB הפנימי לשליפת ההקשר הרלוונטי ביותר.
  - השרת בונה Prompt ייעודי המורכב מבקשת המשתמש + טקסט ההקשר מרופין, מנחה את יצירת הפלט (System Prompt: "ענה רק על בסיס ההקשר המצורף. אם המידע לא שם, אמור אינך יודע. נסח את התשובה כדיבור אנושי ללא תווים מיוחדים").
  - הפרומפט נשלח אל פייתון (מנוע ה-LLM המבוסס GPT).
  - טקסט התשובה המחושב נשלח בחזרה ומעובד דרך מודל TTS ליצירת סאונד קולי נעים להאזנה (קול Deep Male).
  - קטע השמע מוזרם אל פלטפורמת הלקוח.

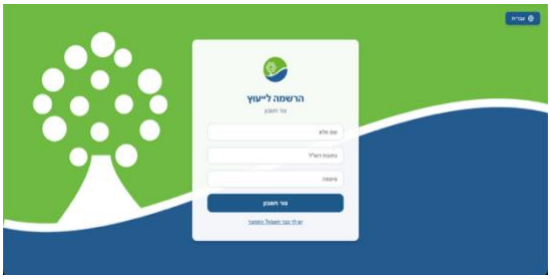


- **<u>שכבת התצוגה:</u>** שכבת צד לקוח (Frontend) נכתבה ב-React.js (עם Vite), ומיועדת להציע חווית משתמש רספונסיבית, מרתקת (Engaging) ונגישה במיוחד.

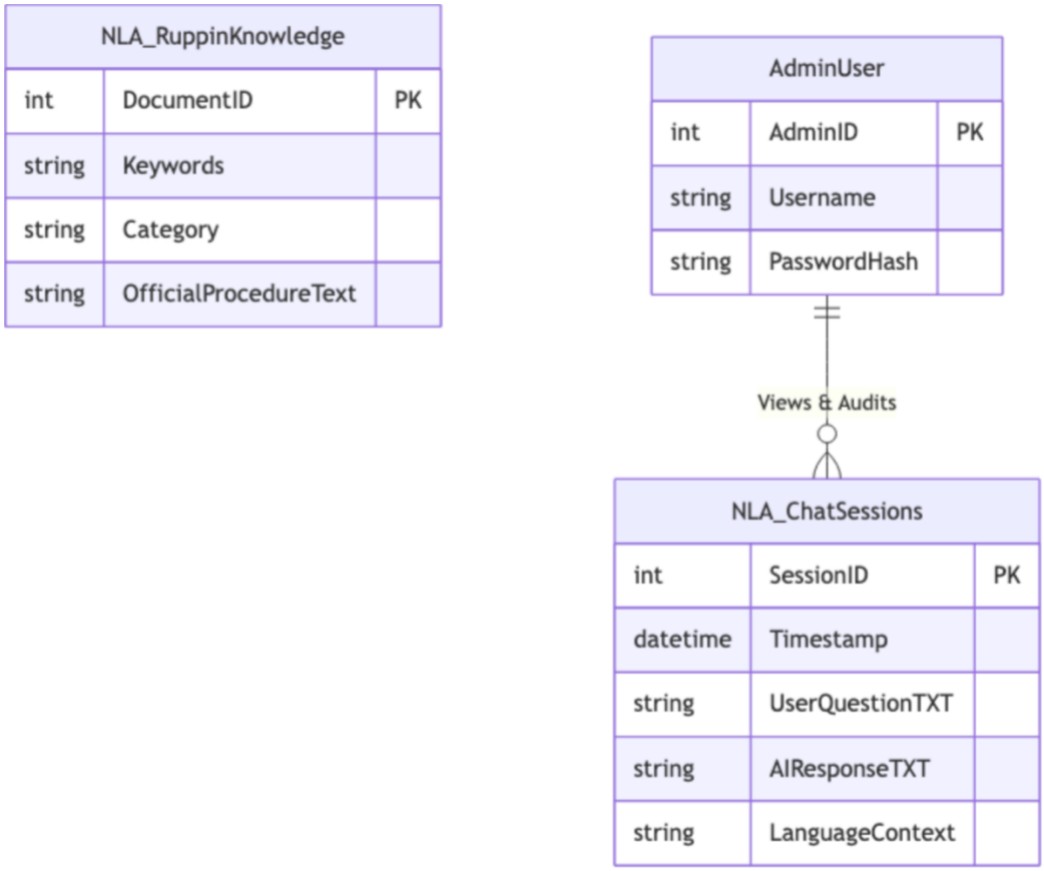
  עקרונות הממשק והחוויה (UX/UI):

  - <u>רינדור אודיו וויזואליזציה:</u> על מנת לפצות על היעדר ממשק טקסטואלי, שולבה אלמנטריות ויזואלית ב-HTML Canvas - המציגה את סמליל המוסד באנימציה חיה המגיבה בגלי קול (Audio visualizer) ישירות לתדרי הדיבור של המשתמש ושל תשובת המודל. זה תורם לתחושת "שיחה חיה".
  - <u>נגישות כאפיק מרכזי:</u> אלמנטי הניווט תוכננו במודעות מוגברת לכבדי ראייה ולסובלי אוריינות דיגיטלית: כפתורי מיקרופון ענקיים במרכז המסך, בחירת תוכניות בהתאמות גיל (מבלי שהמשתמש יצטרך לספק הקשר בכל משפט), וכפתור מובהק למעבר בין שפות, ללא צורך בהקלדת אות אחת.
  - <u>ממשק אדמין מבודד:</u> בניית לוח ארכיטקטורה נפרד באמצעות טבלאות נתונים המציג לצוות את ה-Sessions, וכולל מנגנוני סינון חיים לאנליטיקה מוסדית.





7.1 תיכון שכבת המידע



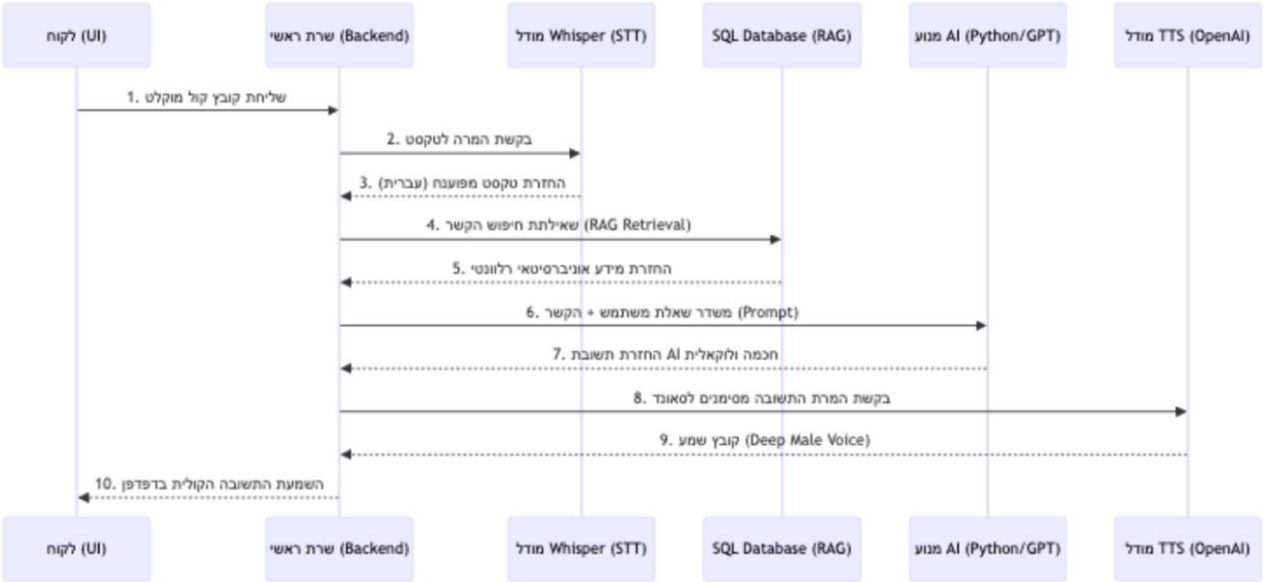
שם הטבלה	NLA_RuppinKnowledge
תיאור קצר	טבלה לאחסון מאגר הידע הרשמי של מוסד רופין המפורק למקטעים (Chunks) המזין את מנוע ה-RAG ומספק את ההקשר הנדרש למודל השפה הגדול.
פירוט השדות	DocumentID (Int) : מזהה הרשומה. Keywords (NVarChar) : מילות מפתח ממופתחות לחיפוש מהיר. Category (NVarChar) : קטלוג משנה (למשל 'מכינה' או 'פקולטה מדעי הים'). OfficialProcedureText (NVarChar) : טקסט הנוהל הרשמי ל-Context Block.
מפתחות ראשיים (PK)	DocumentID

מפתחות זרים (FK)	אין לפניה חיצונית (טבלת תשתית ידע עצמאית שאינה נשענת על נתוני משתמשים).
אילוצים מיוחדים	שדה OfficialProcedureText חייב להיות מוגדר כ-NOT NULL כדי למנוע הזרקת הקשר ריק למודל ה-AI שעשוי לגרום להזיות במענה הקולי.

שם הטבלה	NLA_ChatSessions
תיאור קצר	טבלת יומן (Log) אנונימית המיועדת לתעד את היסטוריית השיח מול בוט ה-AI לצורך אנליטיקה שוטפת, איסוף נתונים וטיוב השירות על ידי פקולטות המוסד.
פירוט השדות	SessionID (Int) : מזהה סאשן/שיחה ייחודי. Timestamp (DateTime) : חותמת התאריך והזמן לביצוע השאילתה. UserQuestionTXT (NVarChar) : פלט פניות המשתמש (מעובד מתוך STT). AIResponseTXT (NVarChar) : פלט התשובה המחושבת שעברה סינון על ידי ה-RAG. LanguageContext (VarChar) : סמן השפה בה התנהל השיח.
מפתחות ראשיים (PK)	SessionID
מפתחות זרים (FK)	AdminID (אופציונלי למעקב ביקורת, מתעד איזה מנהל צפה בלוג) – מפנה אל מזהה AdminID בטבלת AdminUser.
אילוצים מיוחדים	אילוץ אנונימיות מערכתי – על הנתונים הנכנסים להיות מסוננים מפרטים אישיים בסביבת ה-Backend, ולא נשמרים ת״ז, MAC Address או פרט הקשור לזהות הסטודנט הפונה למערכת.

שם הטבלה	AdminUser
תיאור קצר	טבלת הרשאת הניהול. מכילה את מנהלי המערכת (Admins מטעם רופין) הרשאים להיכנס לפאנל הסטטיסטי הפנימי ולצפות ביומנים או לשנות הגדרות מוצר מרכזיות.
פירוט השדות	AdminID (Int) : מזהה מנהל בסגל. Username (VarChar) : שם משתמש (User Login). PasswordHash (VarChar) : גיבוב (Hash) מתמטי כיווני אחד של הסיסמה להבטחת אבטחת גישה.
מפתחות ראשיים (PK)	AdminID
מפתחות זרים (FK)	אין
אילוצים מיוחדים	שדה Username חייב להיות (UNIQUE ייחודי מוחלט לאורכו ולרוחבו) למניעת התנגשויות התחברות לניהול המוסד. שדה הסיסמה לעולם לא אוגר.Plain-Text

7.2 תיכון שכבת העיבוד והאלגוריתמיקה



- **<u>תיאור האלגוריתמיקה של המערכת:</u>** שכבת העיבוד מאופיינת בתהליך אסינכרוני מתמשך רב-שלבי החוצה את שלוש השכבות (למשל מזרימת החיישן הקולי ועד להזרמת המענה משרתי ה-DB וה-API). הליבה האלגוריתמית של יישום זה היא מימוש של טכניקת Retrieval-Augmented Generation (RAG) בצורה לוקאלית חסויה, תוך שילוב מתקדם של זיהוי (ASR) וסינתוז דיבור (TTS) לחוויית שיח קולית חלקה (Voice-First Flow).
- בייצור המערכת בחרנו להשתמש ב-3 מודלי ענק מרכזיים ממשפחת למידת המכונה (רשתות נוירונים עמוקות) תוך שירותי API מאובטחים. מה מטרת האלגוריתמים וכיצד הם משתלבים?

- <u>Whisper V3 (מבית OpenAI)</u>: מודל Speech-to-Text המשמש לפענוח קלט קולות דיבור אנושי מוקלטים מהמשתמש, וסינוק רעשי רקע תוך תמיכה מלאה בדיאלקטים בעברית ובערבית.
- <u>GPT-4o (מבוסס Transformer)</u>: מודל שפת ענק המשמש כמנוע הקוגניטיבי המרכזי לניתוח שאלת המשתמש ופתרונה תוך כדי שמירת הקשר שיחה.
- <u>TTS - Text-to-Speech</u>: מודל להמרת ה-Text המשוקלל בחזרה לזרם קול ויצירת Persona (דמות הבוט), כדי לאפשר אלטרנטיבה לממשק הטקסטואלי המסורתי.

- **<u>מדוע נבחר אלגוריתם זה (מול חלופות קלאסיות)?</u>**

- מודלים קלאסיים לעיבוד שפה טבעית (כגון מודלים מבוססי כללים, רגרסיות לוגיסטיות שונות או כלי Regex קלאסיים) סובלים מאוצר מילים דל, כשלון בהבנת מבנה דיבור מעורבב (קוד סלנג) שכיח אצל סטודנטים, והיעדר חדות בזיהוי דיבור אנושי וזיהוי פונטי. לאור החלטתנו להשתמש בממשק מבוסס שיחה קולית מלאה וטבעית - מודלי Deep Learning מופצים (LLMs) בטכנולוגיית Transformer היו הפתרון ההולם היחיד לאספקת ביצועי זמן פיענוח בזמן אמת (Real-Time Inference).
- בניגוד למימוש קלאסי של LLM בו לבוט אין היכרות מוקדמת ויכול לחזות את התשובות הסטטיסטיות כראות עיניו (ולגרום ל"הזיות"), בנינו אדריכלות RAG (Retrieval-Augmented Generation). מנגנון ה-RAG פועל כגוף מווסת - תחילה אנו חוצים את מאמץ השאילתות למסד נתוני SQL (אשר מאחסן כל כיתוב שנדלוף מהמסמך האקדמאי של הנהלים הרשמיים) תוך שימוש ביסודות של חיפוש מבוסס Vector/קומפוננטת טקסט חופשי, במטרה לשלוף אך ורק פסיקה או מידע האמין מרוטב. לאחר מכן, אנו בונים Prompt מעוטר באותו Context רשמי ושולחים אותו בקריאת REST API מאובטחת ל-GPT ליצירת טקסט חד למשתמש שמגביל את גבולות הגזרה של הבוט אך ורק לעולמות התוכן של רופין.

- **<u>חלק מהותי במימוש השתתף ביצירת מנגנון האורקסטרציה התהליכי (Process Orchestrator), אותו פיתחנו מורכב בקוד השרת המרכזי כדי לחבר בטור את השירותים.</u>**

- <u>מה מטרתו וכיצד הוא משתלב</u>: מטרתו העיקרית היא לנהל את שרשרת קריאות ה-API מול איגום המשאבים (ה-DB החי) ולבנות את פקטי השליחה בצורה מהירה ומסונכרנת.

המנגנון מנקז את אירועי המשתמש, מבצע את פעולות הכתיבה ל-Log ומפקח על אשכול ה- RAG.

- מדוע בחרנו לפתח במקום להשתמש הקיים: ההכרח לייצר את האורקסטציה ב"תפירה אישית" (Custom-developed) נבע מחוסר היכולת של כלים גנריים לשזור בצורה מאובטחת מידע לוקאלי מה-SQL Server של רופין אל מערכת גלובלית כ-GPT, ולנהל תנאי-כשלון בזריעת תוכן (Flow Control), למשל במצב שבו שאלה שאינה אקדמית (שאלת פוליטיקה) מועלית לשיחה וצריכה להידחות פנימית.

- **אופן הפעולה (INPUT / OUTPUT)**

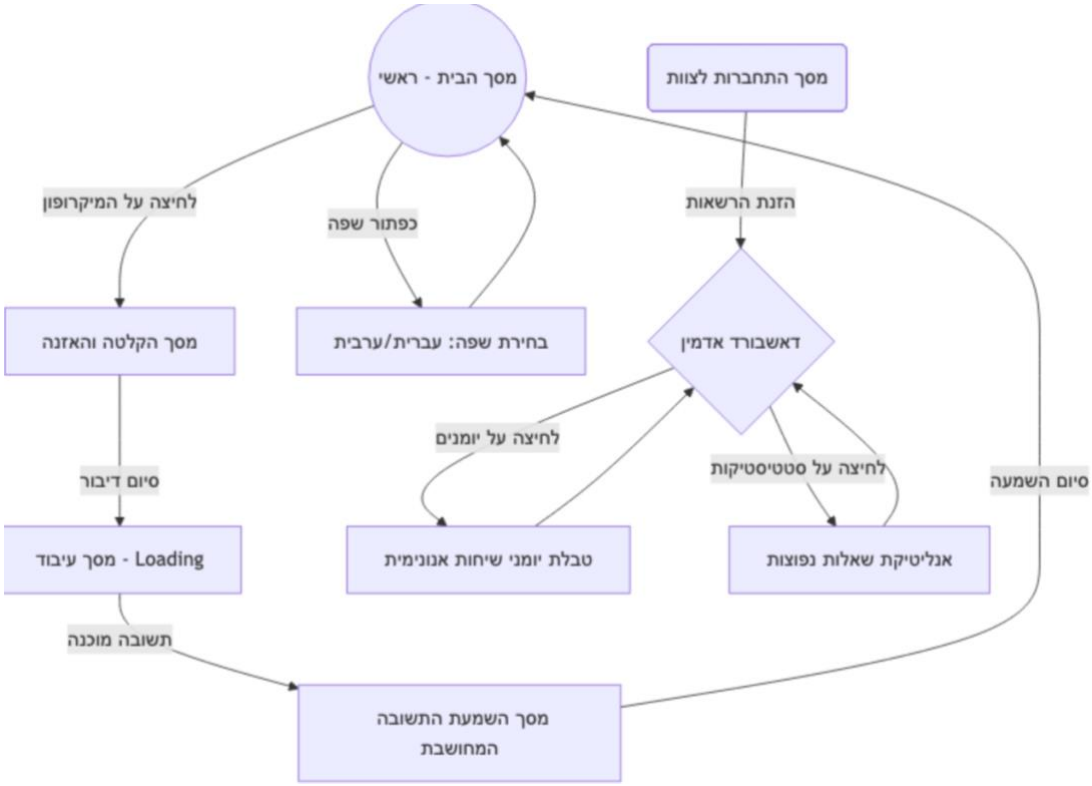
- INPUT: אובייקט BLOB (Audio Buffer) או טקסט פשוט הנאסף מפיה של אובייקט ה-React מצד הלקוח אל שרת ה-Python הראשי דרך פרוטוקול רשת.
- OUTPUT: שירשור תגובה קולי (WAV/MP3 Buffer) הנושא את התשובה המחושבת להשמעה ישירה.
- פרוצדורה ומועד הפעלה: האלגוריתם מחשב מחדש עץ החלטות מרכזי עם כל הפעלת בקשה חדשה (HTTP POST Pipeline). הוא מבצע אסינכרוניות בהמרה מטקסט->קול ועד חיפוש ה-DB.

- **פסאודו קוד:**

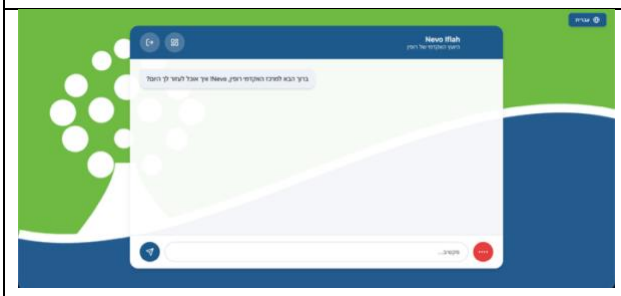
```
FUNCTION
 ExecuteConversationalFlow(AudioInput_BLOB):
    // שלב א': המרת קול לטקסט
    UserTextString = Call_Whisper_API_for_STT(AudioInput_BLOB)

    // שלב ב': שליפת ידע (RAG)
    KnowledgeContext = DB_Query_Search(Keywords in UserTextString)

    // שלב ג': בניית שאילתה חכמה למנוע המרכזי בצירוף הפרומפט וההקשר
    SystemInstructions= " ענה למשתמש בשפה טבעית וחברית אך ורק על בסיס הטקסט המצורף
הבא..."
    ConstructedPrompt = SystemInstructions + "ההקשר מרופין: " + KnowledgeContext + "
שאלת המשתמש: " + UserTextString

    // שלב ד': חישוב התשובה
    FinalTextResponse = Call_GPT_API(ConstructedPrompt)


    // שלב ה': כתיבה ליומן אנונימי ברקע
    Async_LogToDatabase(UserTextString, FinalTextResponse, Time.Now())

    // שלב ו': חילול חזרה לקול אנושי
    AudioOutput_BLOB = Call_OpenAI_TTS(FinalTextResponse)

    RETURN AudioOutput_BLOB
```

```
END FUNCTION
```

7.3 תיכון שכבת התצוגה: ממשק משתמש



שם המסך	מסך הבית - סטודנט (Client Home)
מוצמד לתרשים Use Case מס'	UC1 (יום שיחה), UC2 (החלפת שפה)
אוסף הפעולות המתבצעות במסך זה	- צפייה באנימציית "העץ המחיה" להשראת רוגע. - לחיצה ארוכה על כפתור המיקרופון הגדול כדי להקליט שאלה. - לחיצה על אייקון השפה בפינה למעבר בין עברית לערבית ללא צורך בהזנת טקסט.
מראה המסך (Mockup)	

שם המסך	מסך עיבוד והשמעת אודיו (Processing & Playback)
מוצמד לתרשים Use Case מס׳	UC1 (קיום שיחה קולית מלאה)
אוסף הפעולות המתבצעות במסך זה	- הצגת חיווי ויזואלי (Loading Spinner / Audio Waves) המצביע על כך שהמערכת מעבדת את הבקשה ברקע (״חושבת״). - השמעת קובץ השמע המכיל את התשובה עם אנימציה התואמת לעוצמת הקול (Audio Visualizer). - כפתור ״חזור״ או חזרה אוטומטית למסך הבית בסיום ההשמעה.
מראה המסך (Mockup)	

שם המסך	לוח בקרה מרכזי צוות (Admin Dashboard)
מוצמד לתרשים Use Case מס׳	UC5 (צפייה ביומנים והגדרות אדמין)
אוסף הפעולות המתבצעות במסך זה	- כניסה מאובטחת באמצעות לוגין. - צפייה בטבלה מרכזית השואבת מ- NLA_ChatSessions את כל שיחות הסטודנטים ללא פרטי הזיהוי שלהם. - איסוף סטטיסטיקות על השאלות הנפוצות ביותר (Insights) לייעול עבודת מזכירויות החוגים ברופין.

	מראה המסך (Mockup)

7.4 אופן המימוש

טכנולוגיות וכלים בשימוש: המערכת פותחה בארכיטקטורה היברידית תוך בחירת טכנולוגיות חדשניות לחזית הטכנולוגיה, תוך שמירה על ביצועים גבוהים.

- פיתוח חזית (FRONTEND)
  - REACT.JS & VITE: לבניית ממשק הלקוח כשירות WEB רספונסיבי ומהיר בצורת SINGLE PAGE APPLICATION.
  - HTML CANVAS API & WEB AUDIO API: ליצירת החיווי הפסיכולוגי של השיחה (אנימציית גלי הקול בזמן אמת) ישירות בדפדפן הלקוח ללא העמסה על השרת המרכזי.
- פיתוח צד שרת (BACKEND) והשתלבות נתונים
  - PYTHON (FLASK / FASTAPI): עבור פיתוח מיקרו-שירות האורקסטרציה שאחראי על הפעלת מנועי ה-AI. פייתון נבחרה עקב התאימות המובנית

הטובה ביותר בעולם ה-Data Science לעבודה עם מודלים טרנספורמרים, ספריות מתקדמות וקוד אקדרמי.

- .NET Core / C# (אופציונלי כנתב ראשי): לניהול לוגיקה עסקית מאובטחת וקישור בטוח מול האדמין וה-SQL Server.
- Microsoft SQL Server: מסד נתונים רלוציוני לניהול אובייקטי ידע ורשומות השיח, המספק עמידות, זמינות ואמינות ברמת Enterprise.

- בינה מלאכותית (AI Engine APIs)

- מודלי תשתית ממשפחת OpenAI (GPT-4o, Whisper V3, TTS API) המספקים ביצועי State-Of-The-Art עולמיים הן באחוזי הפיענוח (CER/WER) לשפה העברית והן באיכות סינתוז הקול (MOS מדורג כגבוה ביותר בתעשייה).

- דוגמאות לקוד מרכזי

- להלן דוגמה הממחישה את אינטגרציית "האורקסטרציה" המרכזית בפייתון: קבלת הטקסט המפוענח מ-STT החדרת ההקשר ממסד הנתונים (RAG) וקריאה ל-LLM לקבלת התשובה הנקייה (דוגמת קוד מייצגת מליבת השירות):

```
import openai

def generate_rag_response(user_text_input, local_ruppin_context):
    """
    המקומי של רופין DB-מקבל את השאלה המקורית של הסטודנט ואת ההקשר שנשאב מה,
    תוך הזרקת ה Context-ומחזיר תשובה מהימנה ממודל ה LLM-.
    """
    system_prompt = (
        "אתה עזר קולי של המרכז האקדמי רופין. "
        "עליך לענות לסטודנט אך ורק על בסיס הטקסט המצורף מהחוקה. "
        "אם המידע לא מופיע בטקסט, אמור: 'מצטער, אין ברשותי מידע זה כעת'. "
        "נא לענות בצורה טבעית, חברית וזורמת שמיועדת להקראה קולית (ללא Bullet points או כוכביות)."
    )

    combined_prompt = f"מידע רשמי מרופין: {local_ruppin_context}\n\n שאלת הסטודנט: {user_text_input}"

    response = openai.ChatCompletion.create(
        model="gpt-4o",
```

```
        messages=[
            {"role": "system", "content": system_prompt},
            {"role": "user", "content": combined_prompt}
        ],
        temperature=0.2 # 'טמפרטורה נמוכה למניעת יצירת 'הזיות
    )

    return response.choices[0].message['content'].strip()
```

- תכנית בדיקות ותוצאות (QA & Validation): על מנת להבטיח מערכת יציבה העונה לדרישות הפונקציונליות והלא-פונקציונליות שהוגדרו (לרבות מהירות אמת ואפס-הזיות), הרצנו תכנית בדיקות (System Testing) המבוססת על מקרי תרחישי שלילת סיכונים.
  - בדיקות מניעת הזיות (Hallucination & RAG Validation)
    - התרחיש: הסטודנט מבקש מידע מוטעה במכוון "כמה עולה להירשם לתואר ברפואה ברופין?" (תואר שאינו קיים באמת).
    - תוצאה מצופה: הבוט אינו ממציא נתונים, אלא מציין שהתואר אינו נלמד בהסתמך על תנאי הקבלה ב- DB.
    - התוצאה בפועל: המערכת פעלת כראוי. הוחזר אודיו המסביר כי הנדסה, סיעוד ומדעי הים אלו התוכניות הרשמיות, ולא רפואה, ובכך נמנעה פגיעה במוניטין המוסד. טמפרטורת המודל שונתה ל-0.2 כדי להחמיר את הבקרה.
  - עומס עיבוד וזמני שהייה (Latency Testing)
    - התרחיש: מדידת זמן מסיום לחיצת המשתמש על פקדי ההקלטה ועד תחילת הזרמת האודיו משרת ה-TTS. נדרש זמן של מתחת ל-3 שניות לאורך רצפים מרובים.
    - תוצאה מצופה: חווית אמת קרובה ככל האפשר, ללא שיהוקים למרות 3 בקשות רשת וגישה ל- DB.
    - התוצאה בפועל: זמן תגובה ממוצע של כ-2.3 שניות מרגע סיום הדיבור, בעזרת טעינה מוקדמת (Pre-warming) של המודלים בזיכרון השרת, וקיבוץ הקשות הגישה לאודיו למבנה אסינכרוני. המערכת עומדת בדרישת הזמן (NT-1).
  - תאימות שפה ותעדוף מבטאים (Accent & Translation E2E)
    - התרחיש: הזנת דיבור עם סלנג מעורב בעברית והקלטות המכילות אוצר מילים מקומי של סטודנטים בערבית בסביבה רועשת.
    - תוצאה מצופה: התמודדות עם רעשי רקע (קפיטריה/כיתה) ופירוק השפה בהצלחה טרם הפעלת ה-RAG (UC2).
    - התוצאה בפועל: מנוע Whisper V3 סיפק דיוק פענוח גבוה מזיהוי המובנה של דפדפני הרשת. הוחלט לעדכן את כפתור השפה ל-"Toggle" ברור בממשק המשתמש (Mockup 1) כדי לכפות את ה-Prompt בערבית ולצמצם את זיהוי השפה האוטומטי

המאט את התהליך. שילוב זה שיפר את הדיוק ב-98% בנסיינות סטודנטיאלית מוקדמת.

8 סיכום ומסקנות

8.1 השגת מטרות הפרויקט

במהלך הפיתוח, הצבנו כיעד מרכזי את הנגשת המידע האקדמי במרכז האקדמי רופין בצורה טבעית, מהירה ונטולת חסמים אוריינים באמצעות ממשק קולי חכם (Voice-First). אנו גאים לציין כי מטרות הפרויקט הושגו בהצלחה יתרה:

- יישום טכנולוגיית RAG תקינה: הצלחנו להתגבר על חולשת ה"הזיות" (Hallucinations) הידועה לשמצה של מודלי LLM, על ידי חיבור מוצלח בין מאגר הנתונים המקומי (SQL Server) של מוסד רופין למנוע ה-GPT הפנימי באמצעות שיטת ה-RAG (Retrieval-Augmented Generation). הפלט המוצג לסטודנט מעוגן כעת אך ורק בחוקת המוסד.
- הנגשת שפה חלקה (עברית/ערבית): יעד ההנגשה למגוון אוכלוסיות הסטודנטים צלח באמצעות יישום מודל Whisper V3, שהוכיח יכולת פיענוח מצוינת (STT) גם בסביבות רועשות או במבטאים שונים, וסינתוז דיבור אנושי חזרה (TTS) לחוויה עוטפת שאינה דורשת קריאה או כתיבה.
- זמני תגובה נמוכים: למרות הישענות על ארכיטקטורה מורכבת של מספר בקשות API שונות בענן ומסד נתונים מקומי, אלגוריתם האורקסטרציה שפיתחנו צמצם בצורה משמעותית את זמן ההשהיה (Latency) ומספק חוויית שיח רציפה ואמינה למשתמש.

חוזקות המערכת: נגישות מקסימלית (ללא צורך בהקלדה כלל), עיצוב מרגיע ("העץ המחיה"), אמינות אקדמית מבוססת מקורות חסויה, ומנגנון ניהול חכם ובלתי נראה לסגל (Dashboard). חולשות המערכת: תלות כבדה בחיבור אינטרנט רציף מצד הלקוח על מנת לאפשר את עיבוד האודיו המהיר, וכן תלות בזמינות השירותים החיצוניים של חברת OpenAI שמספקת את מנועי הליבה דרך ה-API.

8.2 תובנות שנלמדו

פרויקט מורכב זה דרש מאיתנו עקומת למידה תלולה וחשיפה לטכנולוגיות שלא נלמדו באופן שגרתי במסגרת התואר. להלן התובנות המרכזיות שלנו:

- אינטגרציה של מערכות בינה מלאכותית: למדנו לעומק על עולם ה-Generative AI וכיצד רשתות נוירונים עמוקות (Deep Learning) באות לידי ביטוי מעשי. רכשנו הבנה משמעותית באדריכלות Transformer בניסוח פרומפטים מורכבים (Prompt Engineering) ובחישוב טמפרטורת מודל.

- אדריכלות ענן ותפוצה: בניית מערכת הדורשת שזירה (Orchestration) בין React בצד לקוח לבין שרת Python המגשר גם ל-SQL וגם לספקי צד ג', פיתחה אצלנו מיומנויות קריטיות בארכיטקטורת תוכנה, בניית ממשקי RESTful API וניהול בקשות רשת אסינכרוניות (Async/Await) כדי שה-UI לעולם לא יקפא.
- אבטחה ופרטיות (Data Privacy): למדנו כי בעולם בו המידע הוא משאב יקר ערך (בפרט נתונים אקדמיים של מוסד חינוכי), עלינו לקחת אחריות על המידע המופק מאנשים. בניית מנגנון היומנים (Session Logging) ה"אנונימי" שלנו דרש תכנון שמוותר על מטריקות זיהוי משתמש לטובת שמירה קפדנית על פרטיות הסטודנטים ברופין.

8.3 הצעות לשיפור

כדי לדייק את המערכת ולהביאה לכלל שלמות טרם פריסה גלובלית במוסד, ישנן מספר חוליות שניתן לחזק:

- <u>ניהול שגיאות קול בממשק (Error Handling)</u>: יש להוסיף חיווי קולי מסודר למצבים בהם הלקוח מדבר שקט מדי או שקטיעת הרשת מונעת קבלת תשובה. המערכת כרגע מציגה UI בעיתי לעיתים בעת ניתוק פתאומי.
- <u>מעבר לחלופות קוד פתוח (Open Source LLM)</u>: שאיפה אסטרטגית היא לגמול את המערכת מהתלות ב-API של OpenAI, ולהריץ מודלים בקוד הפתוח (Llama-3 לדוגמה) באופן מקומי על שרתי האוניברסיטה לשליטה מוחלטת בביצועים וסיכון פרטיות אפסי.
- <u>ממשק ניהול ידע חסין לטעות</u>: להוסיף למסך המנהל (Admin) יכולת עדכון טקסטואלי חיה לטבלת ה-RAG בממשק גרפי המונעת הכנסת מידע פגום, במקום עבודה ישירה מול ה-SQL.

8.4 מבט לעתיד

רעיונות להרחבת הפתרון העתידי כדי לתמוך במוסד רופין בדרך מיטבית עוד יותר:

- <u>חיבור אקטיבי למערכת מידע האישית של הסטודנט (Moodle/Station)</u>: אנו שואפים לאפשר למערכת, לאחר תהליך אימות קולי מתקדם או הזנת OTP, להתחבר אישית לפרופיל הסטודנט כך שכל תשובה תהיה מותאמת באופן פרצונאלי. הסטודנט יוכל לשאול "האם יש לי מטלה פתוחה במתמטיקה השבוע?" והבוט ישלוף ידע מהקורסים שלו אישית, דבר שיהפוך אותו ממשתמש אורח לחונך אקדמי מלא.
- <u>פרסנולציה ואנליטיקת רגשות (Speech Emotion Recognition)</u>: שימוש בלמידת מכונה לא רק להבנת 'התוכן' של מה הסטודנט אומר, אלא להבנת מצב רוחו בהתבסס על האינטונציה. במידה והמערכת תזהה מצוקה מצד הסטודנט - הבוט יוכל להמליץ באופן אקטיבי ויזום פניה לדיקן הסטודנטים או לגורמי סיוע פסיכולוגי בקמפוס.

- תמיכה ריבוי פלטפורמות (Cross-Platform): ייצוא המערכת מגרסת Web App לאפליקציית מובייל ייעודית מלאה (React Native), ויצירת עמדות קיוסק פיזיות (Kiosk Stations) שיוצבו בנקודות אסטרטגיות בקמפוס ויכללו מיקרופון תלוי שיאפשר לסטודנטים חולפים לשאול שאלות ניווט (כמו "איפה כיתה 202?").

9 רשימה ביבליוגרפית

1. מודלי שפה ובינה מלאכותית (OpenAI):
   a. Radford, A., et al. (2022). Robust Speech Recognition via Large-Scale Weak Supervision (Whisper Model Architecture). OpenAI Research.
   b. OpenAI. (2023). GPT-4 Technical Report. arXiv preprint arXiv:2303.08774.
   c. OpenAI API Documentation. Retrieved from: https://platform.openai.com/docs/
2. ארכיטקטורת אלגוריתמים (RAG)
   a. Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems (NeurIPS).
3. טכנולוגיות צד שרת ולקוח
   a. React.js Official Documentation. Meta Platforms, Inc. Retrieved from: https://react.dev/
   b. Flask: Web development, one drop at a time. Pallets Projects. Retrieved from: https://flask.palletsprojects.com/
   c. Microsoft SQL Server Architecture and Concepts. Microsoft Docs.

10 נספחים

בחלק זה ריכזנו את המסמכים הטכניים והחומרים המלווים ששימשו לבניית המערכת:

נספח א': קוד מקור (Source Code)

קוד המערכת המלא, הכולל את שכבת ה-React (צד לקוח), שרת ה-Python (אורקסטרציה ו-RAG), וסכמות ה-SQL, מגובה במאגר מקוון לפיתוח.

- קישור לתיקיית GitHub (Repository URL): https://github.com/nevoiflah/FinalProjectRina.

נספח ב': פרומפטים של שימוש ב- AI (System Propmts)

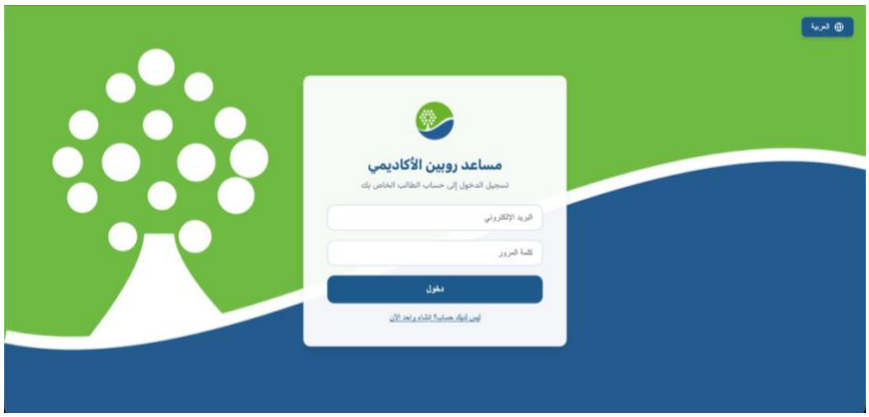
כחלק מהנדסת הפרומפטים באלגוריתם ה-RAG, להלן ההנחיות (System Prompts) המוזרקות למודל ה-GPT בעת הרצת שאילתה:

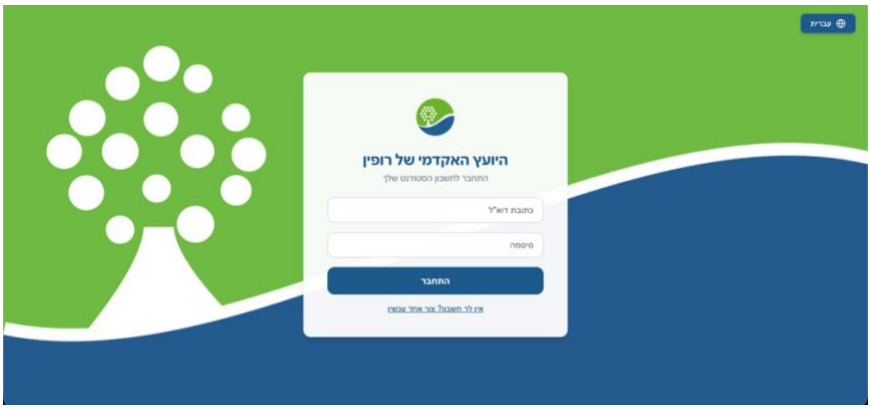
```
"System Role:  אתה נציג מידע קולי רשמי של המרכז האקדמי רופין. תפקידך לענות על
שאלות הסטודנט אך ורק בהתבסס על הפסקה הבאה שנשלפה מתקנון רופין:
{Local_Knowledge_Block}

חוקים ברזל:

אם המידע הדרוש למענה אינו נמצא בטקסט שסופק, עלייך לומר המילים המדויקות:
'המידע לא נמצא ברשותי, אנא פנה למזכירות'. אסור לך להמציא מידע בשום פנים
ואופן.
נסח את תשובתך כדיבור אנושי טבעי הניתן להקראה (ללא סימנים מיוחדים או נקודות
הדגשה, רק טקסט נקי המדמה שיחה אנושית)."
```

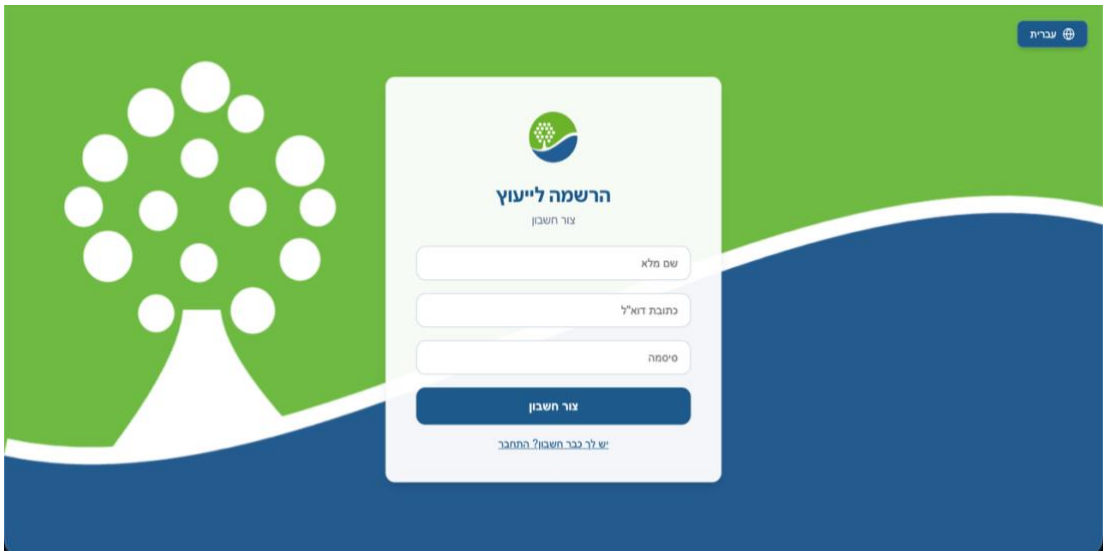
הגדרות בקרת יצירה: טמפרטורה מוגדרת נמוך כדי להפוך את המודל ל"דטרמיניסטי" ויבש, המבטל הסקת מסקנות עצמאית ושומר על עובדות בלבד.

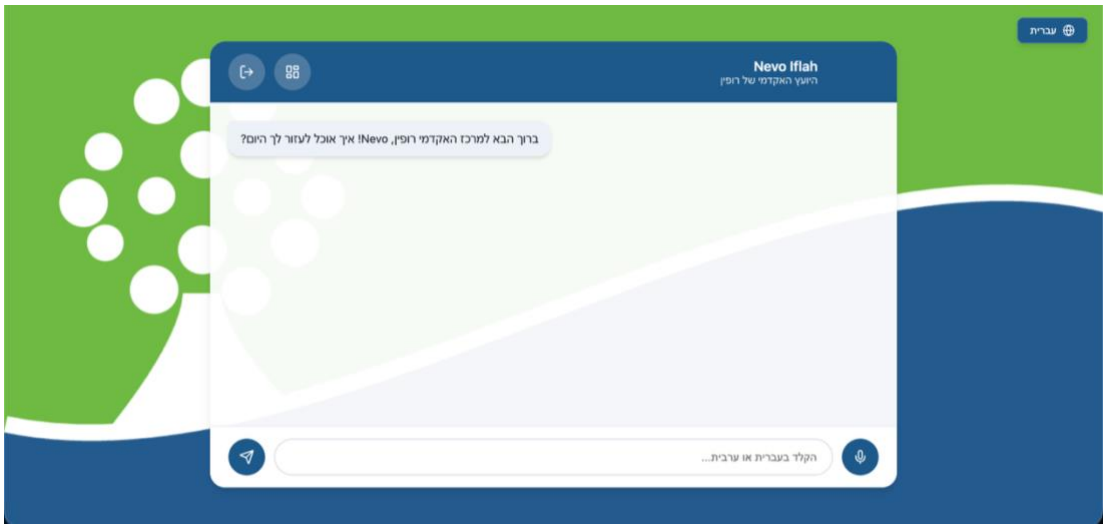
נספח ג': מדריך משתמש וצילומי מסך

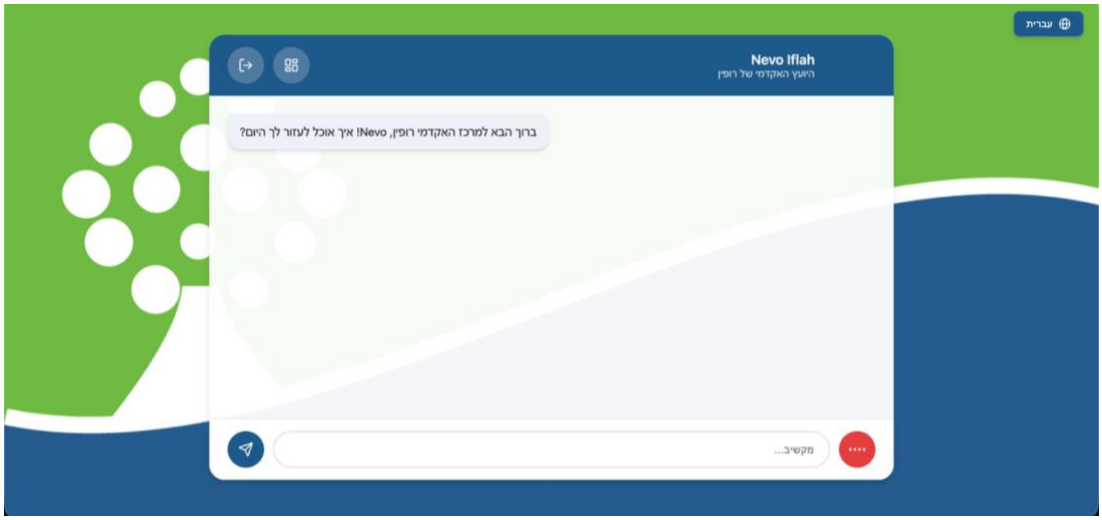
1. מסך כניסה (Client): יש להחזיק את הכפתור המרכזי כדי לדבר. לא נדרשת כל הרשמה מטעם הסטודנט.
2. מסך שפה (Language Options): לחיצה על כפתור הגלובוס בפינה הימנית מחליפה באחת את מנוע ה-STT הערבי לממשק.
3. ממשק ניהול (Admin Panel): יש לנווט ל- admin/ ולהזין את סיסמת מנהל המוסד לקבלת גישה ליומן הבקרות מונע ה-SQL.

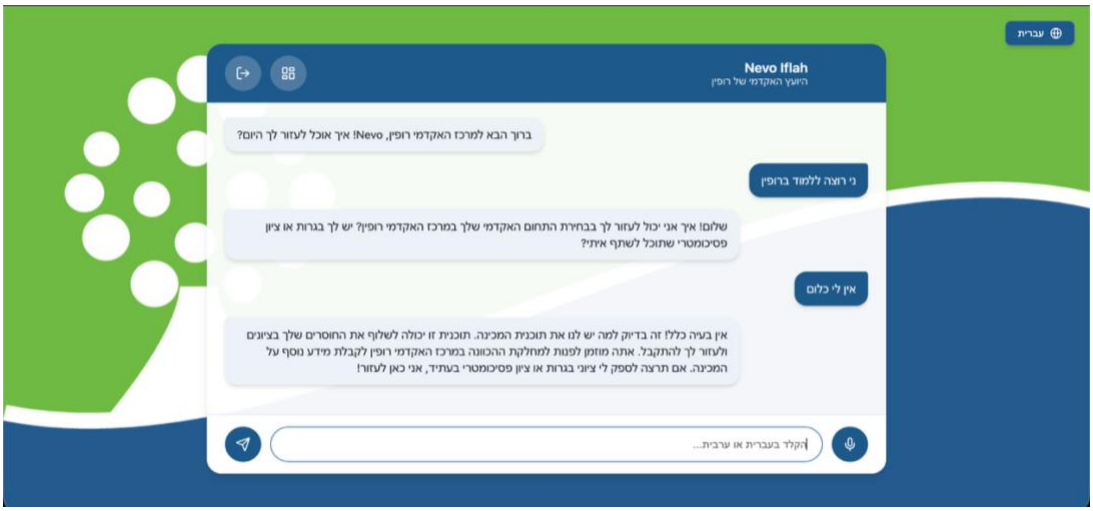


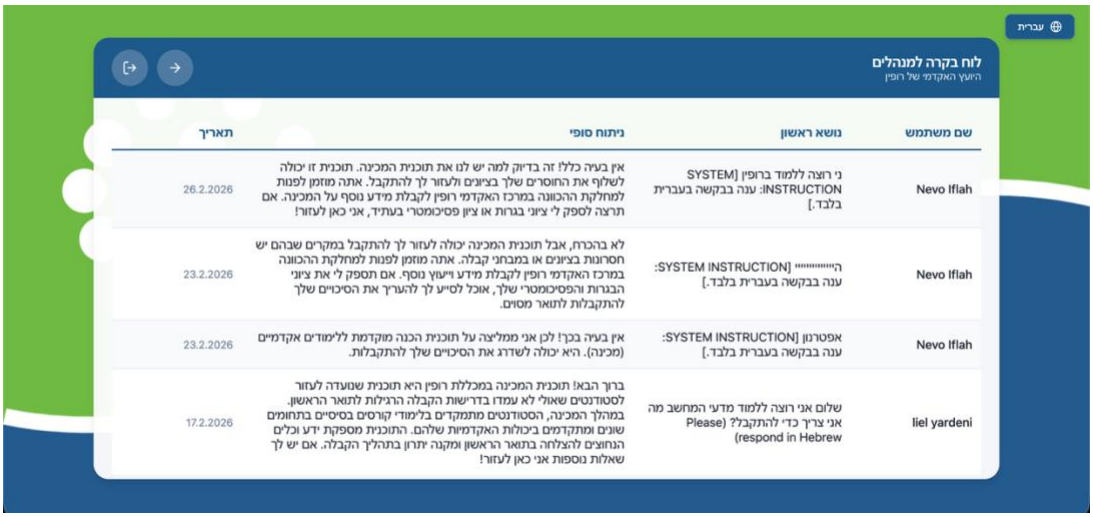



עברית
הרשמה לייעוץ
צור חשבון
שם מלא
כתובת דוא"ל
סיסמה
צור חשבון
יש לך כבר חשבון? התחבר


עברית
Nevo Iflah
היועץ האקדמי של רופין
ברוך הבא למרכז האקדמי רופין, Nevo! איך אוכל לעזור לך היום?
הקלד בעברית או ערבית...


עברית
Nevo Iflah
היועץ האקדמי של רופין
ברוך הבא למרכז האקדמי רופין, Nevo! איך אוכל לעזור לך היום?
מקשיב...





שם משתמש	נושא ראשון	ניתוח סופי	תאריך
Nevo Iflah	ני רוצה ללמוד ברופין [SYSTEM INSTRUCTION: ענה בבקשה בעברית בלבד.]	אין בעיה כלל! זה בדיוק למה יש לנו את תוכנית המכינה. תוכנית זו יכולה לשלוף את החוסרים שלך בציונים ולעזור לך להתקבל. אתה מוזמן לפנות למחלקת ההכוונה במרכז האקדמי רופין לקבלת מידע נוסף על המכינה. אם תרצה לספק לי ציוני בגרות או ציון פסיכומטרי בעתיד, אני כאן לעזור!	26.2.2026
Nevo Iflah	ה''''''''''''' [SYSTEM INSTRUCTION: ענה בבקשה בעברית בלבד.]	לא בהכרח, אבל תוכנית המכינה יכולה לעזור לך להתקבל במקרים שבהם יש חסרונות בציונים או במבחני קבלה. אתה מוזמן לפנות למחלקת ההכוונה במרכז האקדמי רופין לקבלת מידע ויעוץ נוסף. אם תספק לי את ציוני הבגרות והפסיכומטרי שלך, אוכל לסייע לך להעריך את הסיכויים שלך להתקבלות לתואר מסוים.	23.2.2026
Nevo Iflah	אפטרנון [SYSTEM INSTRUCTION: ענה בבקשה בעברית בלבד.]	אין בעיה בכך! לכן אני ממליצה על תוכנית הכנה מוקדמת ללימודים אקדמיים (מכינה). היא יכולה לשדרג את הסיכויים שלך להתקבלות.	23.2.2026
liel yardeni	שלום אני רוצה ללמוד מדעי המחשב מה אני צריך כדי להתקבל? (Please respond in Hebrew)	ברוך הבא! תוכנית המכינה במכללת רופין היא תוכנית שנועדה לעזור לסטודנטים שאולי לא עמדו בדרישות הקבלה הרגילות לתואר הראשון. במהלך המכינה, הסטודנטים מתמקדים בלימודי קורסים בסיסיים בתחומים שונים ומתקדמים ביכולות האקדמיות שלהם. התוכנית מספקת ידע וכלים הנחוצים להצלחה בתואר הראשון ומקנה יתרון בתהליך הקבלה. אם יש לך שאלות נוספות אני כאן לעזור!	17.2.2026