A black background with purple letters

Description automatically generated

**מחשוב ענן-**

**פרוייקט סופי**

**קישור למחברת-**

[**https://colab.research.google.com/drive/1eKglXPPUaIClw-tKEteh7gHfyOdFmg1x?usp=sharing**](https://colab.research.google.com/drive/1eKglXPPUaIClw-tKEteh7gHfyOdFmg1x?usp=sharing)

**מגישים- קרן קזצינסקי- 322845504**

**אוריאן חזיזה- 211921028**

**מאור סיבוני- 207284043**

**דניאל עייש- 315690396**

**דניאל שחר- 314983214**

**רחל אבידן קינג- 315329508**

**קבוצת Unicorn**

1. מהות המוצר ומרכיביו

המערכת הינה מנוע חיפוש חכם שמבצע זחילה על דפים אינטרנטיים ומייצרת אינדקס של מילים מהדפים הנסרקים. המנוע מבצע ניתוח של הדפים תוך כדי סינון של מילים שאינן תורמות למשמעות החיפוש, ומבצע Stemming על מנת לשפר את דיוק החיפושים. המשתמש יכול לבצע חיפושים (רגילים או חכמים-בעזרת AI) במנוע, והמערכת מציגה לו את התוצאות בצורה מסודרת ויעילה. כמו כן, המשתמש יכול לצפות בסטטיסטיקות מעניינות, יכול לנהל שיחה עם צ'אטבוט ומשתמש בעל הרשאות של מנהל יכול גם לערוך את האינדקס ולהוריד דו"ח מילים נפוצות כתמונה.

1.1 מודולים ופיצ'רים מעניינים

מודול: רשימה מקושרת  
מודול זה משמש לשמירת המידע מהאתר הנזחל. לכל מילה באינדקס ישנה רשימה מקושרת המכילה את הלינקים לכל האתרים בהם היא נמצאת, יחד עם תדירותה בכל אתר.  
מודול זה נמצא בתחילת הקוד תחת הכותרת util classes

פיצ'ר: הורדת דו"ח  
פיצ'ר זה מאפשר למשתמש מנהל להוריד את גרף חמשת החיפושים הנפוצים כקובץ PNG למחשב.   
הקוד לפיצ'ר זה נמצא בשכבת הUI תחת הכותרת generate\_search\_report

1.2 microservices שנעשה בהם שימוש

**1. עיבוד מידע (Processing Service)**

**תפקיד:** אחראי על עיבוד המידע לאחר הזחילה (Crawling). מספק את הפונקציונליות לביצוע ההתאמות הנדרשות לנתונים לאחר תהליך הזחילה כאשר מבצעים זחילה על כל הנתונים (ולא על מילה ספציפית).   
**יתרונות:**

* הורדת המילים שמופיעות לעיתים קרובות בחיפושים ואנחנו איננו נזקקים להם.
* הכנת המילים לכתיבה במסד נתונים- המרה לאותיות קטנות, זיהוי מילים עם שורש זהה ושמירת המידע במבנה ברור ומסודר (רשימה מקושרת) המוכן לכתיבה במסד נתונים. ההכנה חוסכת מקום במסד נתונים על ידי מניעת כפילויות ומאפשרת לשלוף נתונים בדיוק רב יותר.

**2. כתיבה למסד נתונים (Database Writing Service)  
תפקיד:** כותב את המידע המאונדקס למסד הנתונים בצורה יעילה.

**יתרונות:**

* כתיבה מרוכזת למסד בפעולת Batch אחת במקום שליחת בקשות נפרדות.

**3. ניהול בינה מלאכותית (AI Service)**

**תפקיד:** מספק תשובות חכמות לשאלות של המשתמש באמצעות חיבור ל- .Gemini  
**יתרונות:**

* תשובות אינטקרטיביות ומעמיקות למשתמש.
* הרחבת היכולת של המערכת מעבר לחיפושים באתר בלבד.

1.3 KPI מרכזיים

* עמידה במטרות הפרויקט:

הפרויקט שלנו עמד בזמני ההגשות עבור כל מטלה.

למדנו מהערות של המרצות ושל עמיתנו לקורס בסדנה ושיפרנו את האתר בעקבות כך.

כלל חברי הצוות היו שותפים לעשייה ועמדו כל אחד במשימתו לפי חלוקת התפקידים שנעשתה בכל שלב.

* עמידה בתנאי אתיקה וחוקיות:

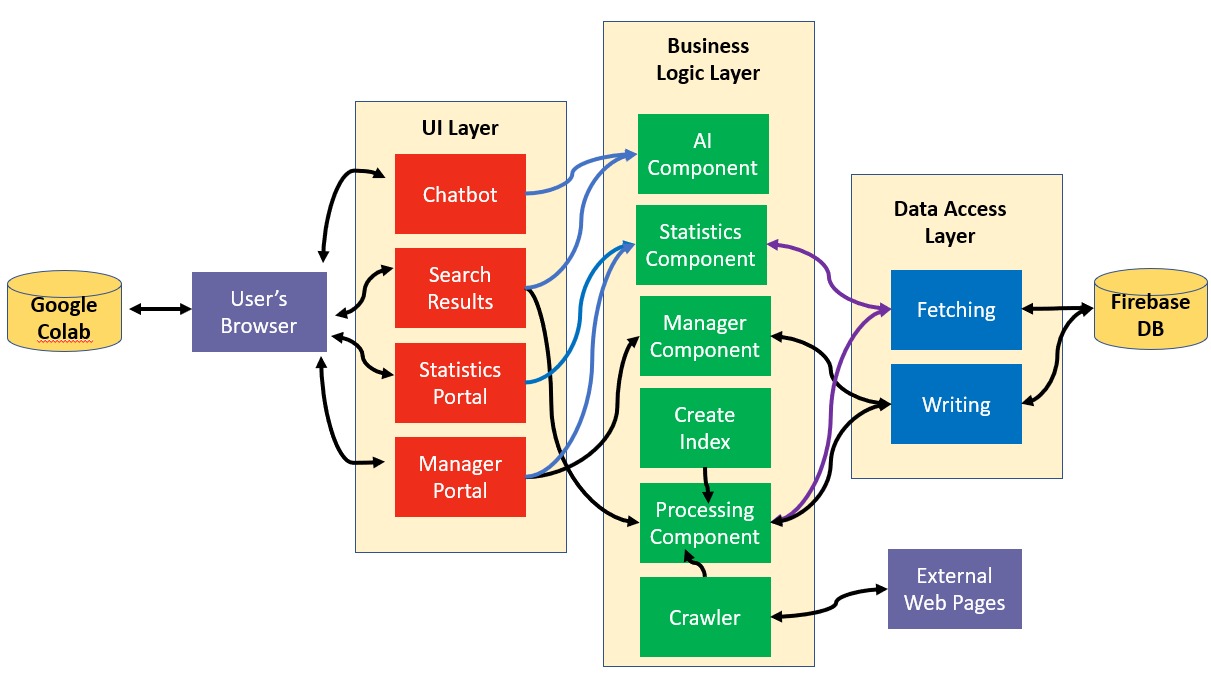
נמנענו ממצבים בעיתיים כגון, פניות לאתר הנזחל בתדירות גבוה שהייתה עלולה לגרום לעומס יתר על האתר שניגשנו אליו.

* יעילות:

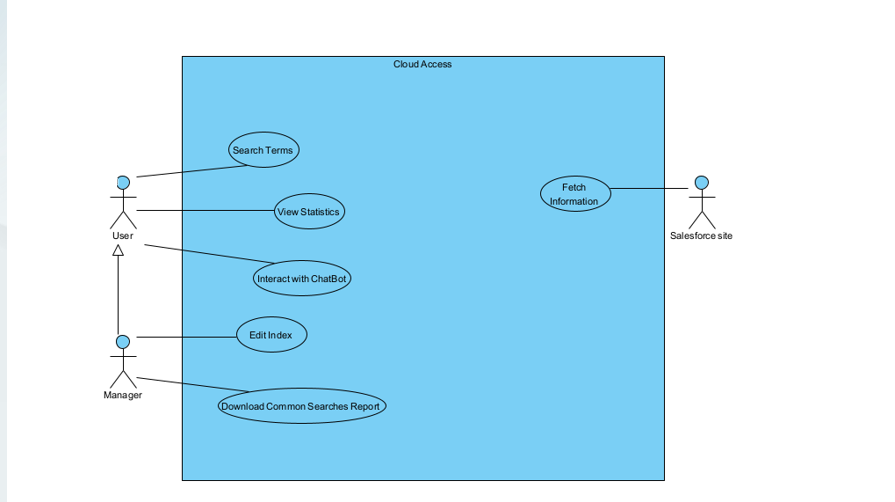
אנו כותבים למסד נתונים רק פעם אחת על-ידי יצירת אובייקט בקוד שלנו, שבעזרתו נעבד את המידע.

2. ארכיטקטורת המערכת

2.1 תרשים ומאפיינים מרכזיים (מעודכן)



2.2 Use case מעודכן של פונקציונליות המערכת.



2.3 דרישות פונקציונליות

* המערכת תאפשר לבצע חיפוש.
* המערכת תבצע עיבוד מידע.
* המערכת תאפשר לשוחח עם בינה מלאכותית.
* המערכת תאפשר הגדלת גופן.
* המערכת תאפשר שמירת נתונים.

1. דרישות לא פונקציונליות - אתגרים איתם הפרויקט מתמודד (מסווגים לפי קישור WIKIPEDIA).

**פרטיות (privacy)**

* **פרטיות למנהל:** יש דברים שרק למנהל יש גישה אליהם ומשתמש רגיל לא יכול לגשת אליהם.

**תקינות ואיכות נתוניםData Integrity) )**

* **שמירת פרטים:** הפרטים שנשמרים עבור כל מילה במסד הם: כותרות המסמכים, הקישורים ומספרם, כמויות בכל הדפים ומספרים של המסמכים שנוספו.

**שמישות Usabillity))**

* **מניעת טעויות:** המערכת כוללת מנגנונים למניעת טעויות משתמש.
* **שקיפות למשתמש:** במסך חיפוש המערכת כתוב למשתמש שהמידע נאסף לצורך נתונים סטטיסטיים ובמסך תוצאות כתוב שהתוצאות ממויינות לפי כמות הופעות.
* **שימוש עקבי באלמנטים ויזואליים:** כפתורים, סמלים ותפריטים נראים ופועלים בצורה עקבית בכל חלקי המערכת, כדי להפחית בלבול ולהגדיל את היעילות.

4. ביקורת עמיתים אשר ניתנה במהלך הצגת הפרויקט (שבוע 9), וכיצד התמודדתם איתה. נא לחשב גם את ציון SUS ולהתייחס אליו.

|  |  |  |
| --- | --- | --- |
| **הערת משוב** | **האם התבצע שינוי באפליקציה בעקבות ההערה?** | **נימוק** |
| יש לשפר את זמן התגובה של המערכת והתחברות כמנהל | לא | לא רלוונטי, היה בעיה באינטרנט הביתי של מי שהציגה מהקבוצה |
| כותרות לא לבנות | לא | המסך של המציגה היה על מצב dark mode |
| ניגוד צבעים בטקסט |
| צבע לבן על רקע ורוד נעלם בתוכו, הייתי בוחר בצבע אחר |
| המסך נראה ענק שלא לצורך, אולי כדאי להקטין אותו | לא | זוהי דעתם האישית ולכן השינויים שעשינו הם בהתאם לשאר הערות |
| אישית שילוב הצבעים קצת הפריע לי לעין אבל זה דעה אישית רק בסך הכל עבודה יפה מאד |
| צבעי כתיבה |
| גרפים יותר יפים |
| עוד סטטיסטיקה או פיצ'ר | כן | התווספה אופציית חיפוש חכם |
| צ'טבוט | כן | שניהם התווספו בש"ב 3 |
| הייתי מוסיף לadmin אופציה להוסיף מילה |
| לעשות שתוצאה זה hyper link | לא | כבר היה ממומש |
| מחיקת מילה מקישור ולא את כל המילה מכל הקישורים | לא | אנחנו מוחקים מילה מכל הקישורים ולא מקישור ספציפי כי לדעתנו זה פתרון אידיאלי |

ציון הSUS של המערכת לאחר חישוב הינו 82.6 . ציון זה הוא מעל הממוצע (68) ולכן מעיד על שביעות רצון גבוהה מהמערכת

5 . תיק תחזוקה – תיאור של כל הקבצים והאובייקטים המרכזיים, ותיעוד קצר של כל פונקציה בקוד.

**תיק מתכנת**

החלוקה בקוד היא לארכיטקטורת 4 שכבות וקלאסים לצרכי נוחות. לכן נשתמש בחלוקה זו גם עבור תיק המתכנת.

***2nd layer- business logic layer***

util classes:

מחלקת Node: מייצגת קישור בודד שנאסף במהלך הזחילה, יחד עם המידע שמשויך אליו.   
תכונות: docTitle- כותרת הדף שבו נמצא הקישור  
DocIDs- שומר את ה-URL.  
amountInPage- מספר סידורי ייחודי של הקישור ברשימה המקושרת.  
docNum- שומר את המספר של המסמך.  
-next מצביע לקישור הבא ברשימה המקושרת.   
מכילה מתודות של אתחול והדפסה.  
  
מחלקת LinkedList: מנהלת רשימה של קישורים שנאספו במהלך הזחילה.  
תכונות: head- מצביע על הקישור הראשון ברשימה המקושרת.  
counter- מונה של מספר הקישורים שנוספו לרשימה.   
frequencyInAllPages- סופר את כמות ההופעות של כל הקישורים בכל הדפים.  
מכילה מתודות בסיסיות של אתחול, הדפסה והוספה לרשימה מקושרת.   
 -def sort\_by\_amount(self)מתודה שמבצעת מיון בועות לצורך מיון החיפושים בהמשך במסך התוצאות כך שהחיפושים שהכי נפוצים יופיעו בתחילת העמוד.

מחלקת Processing Service: מיקרו-שירות שאחראי על העניין של עיבוד המידע. מכיל סה"כ 3 מתודות.  
מתודות:  
index\_words(self, soup, url)- מתודה שמקבלת את תוכן הhtml של הדף וכתובת ה-URL שלו. המתודה מוציאה את כל המילים שמופיעות בדף ושומרת את כמות ההופעות של כל מילה באמצעות word\_counts. אם המילה לא קיימת באינדקס, יוצרים רשימה מקושרת חדשה עם הקישור שלה. אם המילה קיימת, הרשימה המקושרת מעודכנת עם ה-count החדש. המתודה מחזירה את האינדקס המעודכן.

 -def remove\_stop\_words(self,index)מתודה שמקבלת אינדקס גלובלי שבו כל מילה מקושרת לנתוני הדפים שבה היא הופיעה. הפונקציה מגדירה רשימה של מילים נפוצות שאינן תורמות למשמעות החיפוש. עוברת על כל המילים ב-index ומסירה את המילים שמופיעות ב-Stop Words. המתודה מחזירה את האינדקס המעודכן.

def apply\_stemming(self, index)- מתודה שמקבלת אינדקס גלובלי שבו כל מילה מקושרת לנתוני הדפים שבה היא הופיעה. הפונקציה ממירה מילים לשורש שלהן. אם השורש כבר קיים ב-Stemmed Index, הנתונים מתווספים לשורש. אחרת, נוצרת רשימה חדשה תחת השורש. הפונקציה גם מחשבת מחדש את התדירות של מילים מאוחדות לפי השורש. המתודה מחזירה אינדקס חדש שבו כל מילה מופיעה בצורת השורש שלה, יחד עם כל המידע שמשויך אליה.

מתודות נוספות בשכבה:

def fetch\_page(url)- מתודה שמקבלת את הקישור של הדף. שולחת בקשת HTTP כדי לחלץ את התוכן ולהמיר אותו לאובייקט html. מחזירה את האובייקט המבוקש אם הבקשה הצליחה. אחרת, מחזירה None.

def search(query, index)- מתודה שמקבלת את השאילתה שהמשתמש מחפש ואת האינדקס שבו כל מילה מקושרת לנתוני הדף בו הופיעה. הפונקציה ממירה את השאילתה לרשימה של מילים ומפעילה Stemming. יש בדיקה עם המילה קיימת באינדקס. אם כן יש הוספה ל-results עם הנתונים שלה. מוחזר מילון שבו המילים מהשאילתה הם המפתחות, והערכים הם הנתונים מתוך האינדקס.  
  
def search\_engine(query)- מתודה שמקבלת את שאילתת החיפוש של המשתמש. מערך התוצאות מתקבל מהמתודה search. יש פיצול למילים, הסרה של stop words והמנכה לאותיות קטנות. בסוף מעדכנים את החיפושים במסד ומחזירים את התוצאות.

def startEngine(url)- מקבלת את כתובת הדף לסריקה. הפונקציה קוראת לזחילה על הדף ובודקת אם היא הצליחה. עושה אינדקס למילים עם Stemming ושומרת בfirebase. שומרת את הנתיב במסד הנתונים. מוחזר None אם הזחילה נכשלה ומתרחש עדכון של האינדקס במסד.

def extract\_links(html)- מתודה שמקבלת את תוכן הדף ומוציאה את כל הקישורים ממנו. מחזירה רשימה של קישורים שמוכנים לזחילה נוספת.

crawl(url) def- מתודה שמקבלת את הכתובת שמיועדת לסריקה. המתודה בודקת אם הדף כבר בוקר (כדי למנוע זחילה כפולה). כמו כן, היא שולפת את תוכן הדף, מאנדקסת את המילים ומוציאה קישורים תוך כדי זחילה. הפונקציה מחזירה 1 אם הדף כבר בוקר וממשיכה לזחול על קישורים חדשים אם הדף טרם בוקר.

מתודות שקשורות לסטטיסטיקה:

def fiveMostSearchedWords(dbName)- מתודה שמקבלת נתיב למסד נתונים שבו יש סטטיסטיקות על מילות החיפוש. הפונקציה קוראת ל- fetchQueryWordsFromDB(dbName) שמחזירה ממילון שבו המפתחות הם המילים והערכים הם מספר הפעמים שהמילה חופשה. אם לא נמצאו מילים מוחזר None. אחרת, יש מיון בסדר יורד לפי הערכים ולבסוף מוחזרות המילים הכי נפוצות.  
 def indexForRandom()- מתודה ששולפת מתוך האינדקס 5 מילים רנדומליות ושומרת במערך באינדקס של המילה את התדירות שלה. המערך של המילים הרנדומליות מוחזר.

מתודות שקשורות למסך מנהל:

def deleteWord(word)- מתודה שמקבלת את המילה שברצון המנהל למחוק. אם המילה לא קיימת באינדקס לא ניתן למחוק אותה. כמו כן, המילה עוברת Stemming. המילה נמחקת מהאינדקס וממסד הנתונים. אם המחיקה הצליחה המילה נמחקת גם מ- .resFromDB2ndLayerהפונקציה מחזירה True אם המחיקה עברה בהצלחה ו-False אחרת.

def sendDataToManager()- מתודה ששולחת את כל האינדקס המקומי למנהל לצורך בדיקה או עריכה.

def index\_specific\_word(soup, url, word)- מתודה שמקבלת את התוכן של הדף, הכתובת של הקישור והמילה הספציפית לצורך חיפוש/אינדקס. המתודה מבצעת מעבר על כל המילים בעמוד ומשווה אותם למילה הרצויה. כמו כן, היא סופרת כמה פעמים המילה מופיעה בעמוד. אם המילה נמצאת, מוחזר אובייקט של צומת עם הקישור, כותרת העמוד ותדירות ההופעה. אחרת, מוחזר None.

def crawlForSpecificWord(word)- המתודה מקבלת את המילה המבוקשת. כמו כן, מפעילה Stemming על המילה. אם המילה כבר קיימת ב- resFromDB2ndLayerמוחזר 0. יש זחילה על כל הדפים שבוקרו וחיפוש של המילה הספציפית. אם יש הופעות של המילה, נוצרת רשימה מקושרת חדשה עם הנתונים. אם הכתיבה למסד הצליחה האינדקס המקומי גם מעודכן.   
החזרות: 0- המילה קיימת במאגר, 1- המילה לא נמצאה בשום דף, 2- כשלון בכתיבה למסד, 3- המילה נוספה בהצלחה לאינדקס ולמסד הנתונים.

מחלקת connectToAi: מיקרו שירות שאחראי על החיבור עם ה-AI. מכיל סה"כ 3 מתודות.

מתודות:

to\_markdown(self, text)-מתודה שממירה טקסט לפורמט Markdown על ידי החלפת הסימן '•' ב-' \*' לצורך יצירת רשימות. בנוסף, המתודה מבצעת הזחה של הטקסט באמצעות textwrap.indent, מה שמאפשר להציג את הטקסט עם סימן > לפני כל שורה, באופן שתואם לפורמט Markdown ומסייע להציג את התוכן בצורה מסודרת וברורה.

askGemini(self, query)-מתודה שמבצעת שיחה עם מודל ה-Gemini של Google, שואלת את המודל שאלה בנוגע לנושא נתון ומחזירה את התשובה בצורה טקסטואלית המתאימה לחיפוש חכם- כלומר מועבר פורמפט המבקש הגדרה או הסבר פשוט של השאילתה הנשלחת. המתודה מקונפגת עם מפתח API ומבצעת קריאה למודל Gemini-1.5 כדי לקבל הסבר קצר על הנושא המבוקש. במקרה של חריגה "TooManyRequests", המתודה ממתינה 60 שניות ומבצעת ניסיון נוסף לשאול את השאלה.

  -def askGeminiChatbot(self,query)באופן דומה למתודה הקודמת רק שהפעם הprompt מנוסח לצ'אטבוט במקום חיפוש חכם.

***3rd layer- data access***

מחלקת writeToDbMicroservice: מיקרו שירות האחראי לכתיבה של מידע מהזחילה למסד נתונים.

מתודות:

writeWordsAfterCrawl(self, index) -אחראית על כתיבת מילים שאונדקסו לאחר הזחילה למסד הנתונים ב Firebaseהמתודה ממירה את הרשימות המקושרות של כל מילה לפורמט שניתן לשרשר למסד הנתונים, כולל שמות הדפים, ה-URLs וכמות ההופעות של כל מילה בדפים. לאחר מכן, המתודה כותבת את כל המילים בבת אחת למאגר הנתונים ב-Firebase באמצעות קריאת post   
אם הכתיבה נכשלת, המתודה מחזירה False.

-writeOneWordToDb(self, word, posting\_list) אחראית לכתיבת מילה בודדת למסד הנתונים ב-Firebase. המתודה ממירה את הרשימה המקושרת של המילה לפורמט שניתן לשרשר למסד הנתונים, כולל שמות הדפים, ה-URLs וכמות ההופעות של המילה. לאחר מכן, המתודה מבצעת עדכון בעזרת קריאת patch בנתיב שנשמר ב-newPath כדי להוסיף את המילה החדשה למאגר הנתונים. אם הכתיבה נכשלת, המתודה מחזירה False.

מתודות נוספות בשכבה:

writeQueryWords(dbName, query) -אחראית על עדכון תדירות המילים במסד הנתונים ב-Firebase, כאשר היא בודקת אם המילה כבר קיימת במסד. אם המילה קיימת, המתודה מעדכנת את תדירותה ב-1, ואם המילה לא קיימת, היא מוסיפה אותה עם ערך התדירות 1.

-fetchWordsFromDB(dbName) מבצעת שליפה מהירה של המילים מהמאגר לאחר הזחילה, וממירה את הנתונים לפורמט שניתן לעבוד איתו בקלות, כולל המילים, תדירותן, ורשימות הקישורים. אם הנתונים קיימים, המתודה מחזירה את המילים והקישורים בפורמט של מילון עם רשימות מקושרות.

deleteFromIndex(dbName, word) -אחראית למחוק מילה מסוימת ממסד הנתונים. המתודה מוצאת את המילה המתאימה, אם היא קיימת, ומוחקת אותה ממאגר הנתונים לאחר עדכון רשימת הנתונים.

fetchQueryWordsFromDB(dbName) -אחראית לשלוף את הנתונים של כל המילים במסד הנתונים. אם יש נתונים זמינים, המתודה מחזירה אותם, ואם לא, היא מציגה הודעת שגיאה.

\*בנוסף יש את הUI layer האחראית על עיצוב המסך (שכבה 1), ואת המסד המוזכר קודם (שכבה 4)

6. תיק למשתמש , הכולל הסבר כללי על המערכת , פירוט מסכים, מעברים בין מסכים והסבר על טעויות אפשרויות.

תיק למשתמש – הסבר כללי על המערכת

המערכת הינה מנוע חיפוש חכם שמבצע זחילה על דפים אינטרנטיים ומייצרת אינדקס של מילים מהדפים הנסרקים. המנוע מבצע ניתוח של הדפים תוך כדי סינון של מילים שאינן תורמות למשמעות החיפוש, ומבצע Stemming על מנת לשפר את דיוק החיפושים. המשתמש יכול לבצע חיפושים במנוע, והמערכת מציגה לו את התוצאות בצורה מסודרת ויעילה. כמו כן, המשתמש יכול לצפות בסטטיסטיקות מעניינות. משתמש בעל הרשאות של מנהל יכול גם לערוך את האינדקס ולהוריד דו"ח מילים נפוצות כתמונה.

מסכים:

1. מסך הבית:

תצוגה של שורת חיפוש בה המשתמש יכול להקליד את השאילתה.במסך זה יש כפתור לחיפוש שמפעיל את פעולת החיפוש במנוע, וכפתור לשליחת שאלה לAI עבור חיפוש חכם. בנוסף ישנם כפתורים נוספים לכניסה לדפים שונים באתר. כמו כן, נאמר למשתמש כי המילים שעליהן מתבצע החיפוש נשמרות לצורך ניתוח נתונים לסטטיסטיקות בעתיד.

1. מסך תוצאות חיפוש:

תוצאות החיפוש מוצגות לפי סדר תדירות ההופעות של המילים (דף שבו מילה מופיעה הכי הרבה פעמים מופיעה בתחילת הדף). כל תוצאה כוללת כותרת הדף, קישור ישיר לדף ותדירות הופעת המילה בדף. עבור חיפוש של מונח שמורכב מכמה מילים מודפס כמה פעמים כל מילה הופיעה בדף. במסך זה הבהרנו למשתמש כי התוצאות ממויינות כך שהתוצאות שמופיעות לעיתים יותר תכופות בדף מופיעות ראשונות.

1. מסך סטטיסטיקות:

הצגת שתי דיאגרמות- חיפושים נפוצים ומופע של כמה מילים רנדומליות באתר. המשתמש יכול לצפות בנתונים הסטטיסטיים שנאספו קודם לכן.

1. מסך מנהל:

משתמש בעל הרשאות של מנהל יכול למחוק ולהוסיף מילים לאינדקס. הוא גם יכול להפיק דו"ח של המילים הכי נפוצות שיורד בפורמט של תמונה.

1. מסך צ'אטבוט:

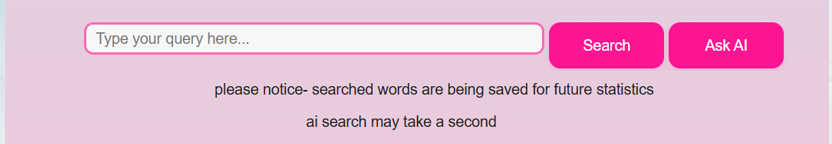
מסך ובו שורת כתיבת טקסט בה המשתמש יכול לשלוח הודעות ולנהל שיח עם בינה מלאכותית בלחיצת כפתור. התשובות של הבינה המלאכותית יופיעו בתחתית הדף, כאשר לכל תשובה כתוב בנוגע לאיזה שאילתה היא.

מעברים בין מסכים

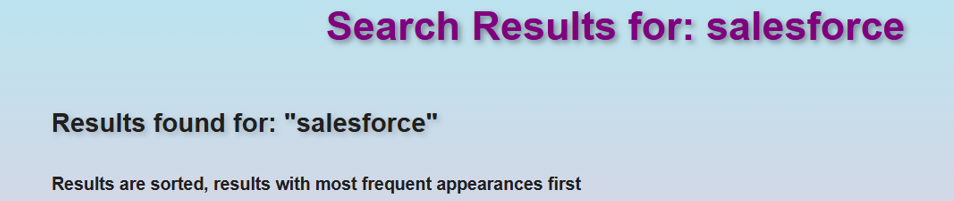
* המעבר בין המסכים מתבצע בצורה חלקה לפי בחירת המשתמש במסך הבית (כניסה למסך תוצאות, בינה מלאכותית, מנהל וסטטיסטיקות) בלחיצה על הכפתור הרלוונטי.
* יש כפתור Back שמאפשר למשתמש לחזור אחורה בקלות למסך הבית.
* לאחר ביצוע חיפוש בלחיצה על כפתור Search, המשתמש מועבר למסך תוצאות החיפוש בו מוצגות התוצאות המתאימות.
* המשתמש יכול לעשות חיפוש חכם של מושגים על ידי לחיצה על כפתור Ask AI.
* אם המשתמש בעל הרשאות מנהל, הוא יכול לגשת למסך המנהל לצורך ניהול המילים והאינדקס ע"י לחיצה על Manager Page. נפתח לו חלון בו הוא צריך להזין את הסיסמה 1234. אפשר גם לסגור את החלון במידה הצורך.
* המשתמש יכול לצפות בסטטיסטיקות מעניינות על ידי לחיצה על כפתור Statistics Page.
* המשתמש יכול לנהל שיחה עם צ'אטבוט ע"י לחיצה על כפתור chatbot ומעבר למסך זה.

הסבר על טעויות אפשריות

1. ייתכן כי המשתמש יזין מילה שלא מופיעה באתר. במקרה כזה המערכת תציג למשתמש הודעה שלא נמצאו תוצאות חיפוש.
2. ייתכן שתהיה שגיאה בכניסה למסך מנהל (סיסמה: 1234). אם משתמש לא מורשה נכנס למסך המנהל בטעות. לשם כך שמנו את האפשרות להזין סיסמה.
3. ייתכן שהמנהל ינסה למחוק מילה שלא קיימת באינדקס. לשם כך לא אפשרנו למנהל למחוק מילים שלא מופיעות בו. יש dropdown עם השלמה אוטומטית של כל המילים שקיימות באתר וברגע שמילה נמחקת מהאינדקס מוצג שהמילה נמחקה בהצלחה.
4. ייתכן שהמנהל ינסה להפיק את הדו"ח של החיפושים הכי נפוצים מבלי שיש חיפושים בכלל. במקרה זה מוחזר שאין חיפושים בכלל במערכת.
5. ייתכן שהמשתמש בטעות יבקש מהChatBot מידע על מונח ולא יקבל את התשובות הדרושות. לשם כך הוספנו תיבת טקסט שמאפשרת להזין מחדש את השאילתה המבוקשת.
6. אם המשתמש נכנס בטעות לדף שלא התכוון יש כפתורי Back בכל דף באתר שחוזרים למסך הראשי.
7. יתכן שהמנהל ינסה להוסיף לאינדקס מילה שלא קיימת בכלל באתר הנזחל. במקרה כזה, מוחזרת לו הודעה שהמילה לא קיימת.
8. שקיפות אלגוריתמית – כיצד הבהרתם למשתמשים את האלגוריתמים והנתונים הנאספים בקוד שלכם?

נציג את השני דרכים בהם השתמשנו בשקיפות אלגוריתמית בקוד-  


בדף החיפוש הבהרנו למשתמש כי המילים שעליהן מתבצע החיפוש נשמרות לצורך ניתוח נתונים לסטטיסטיקות בעתיד.



בדף הצגת תוצאות חיפוש הבהרנו למשתמש כי התוצאות ממויינות כך שהתוצאות שמופיעות לעיתים יותר תכופות בדף מופיעות ראשונות.

8. אתגרים אשר עלו במהלך העבודה, וכיצד התמודדתם איתם.

* בתחילת הקורס התחלנו לעבוד בפעם הראשונה עם סביבת העבודה גוגל קולב, מה שבעבר לא התנסנו. בעקבות כך היינו צריכים ללמוד דרך חדשה לתכנת וליצור GUI בסביבת עבודה דינאמית.
* נתקלנו בזמן חיבור הצ'אטבוט בבעיה בכתיבת הקוד מכיוון שגוגל קולב נותן לנו לעשות מקסימום 50 גישות ל- API כל יום. לשם כך ניסינו ליצר מפתח חדש עבור הגוגל קולב, אך לאחר 5 גישות שום נחסמנו. מצב זה גרם להגבלת זמן עבורנו לעבוד על חלק זה בקוד.
* מכיוון שהכתיבה למסד נעשתה כאובייקט אחד והמידע היה מאוד מקונן, נדרש עבודה רבה בעת שליפת המידע ממנו ע"מ להבין איך הוא חוזר, באיזה מבנה נתונים, ואיך לעבד אותו באופן המשרת את הצרכים שלנו במדויק.

9. משובים מהצגה- שבוע 11

מרבית המשובים היו חיוביים ולכן לא נתייחס אליהם כאן.

|  |  |  |
| --- | --- | --- |
| **הערת משוב** | **האם לדעתכם יש צורך בשינוי במערכת בעקבות ההערה?** | **נימוק** |
| קצת חסר דאטא בהצגה. מילה אחת. | לא | מטרתנו הייתה להכניס גרפים תמציתיים המציינים מילה ספציפית והמידע הרלוונטי אליה (כמות חיפושים/כמות הופעות באתר) |
| הגרפים ברורים אפשר אולי להחליף לצבע בולט יותר | לא | טעם אישי, בעינינו הצבע בולט |
| use-case יותר מפורט (אם יש מה להוסיף) | לא | דיאגרמה זו אמורה להציג תהליכים מרכזיים ומשמעותיים במערכת, ואכן זה מה שהדיאגרמה שלנו מציגה |
| אפשר להוריד את הגרף עם צילום מסך, הפיצ'ר לא הכרחי KPI של המערכת זה לא עמידה בזמני ההגשה נראה לי | לא | ניתן לומר זאת על כל מערכת שבה יש מסמכים. הפיצ'ר נועד לנוחות המשתמש וביצוע יעיל יותר של שמירת המידע עבורו.  בנוגע לKPI – נבדק באינטרנט וזה אכן עומד בדרישה |

התייחסות למשוב שקיבלתם בשבוע 12 – code review – יש להגיש את המשוב שקיבלתם מהצוות השני בחלק הקבוצתי , ולכתוב התייחסות מתאימה:

לכל הערה, יש לרשום אותה ולציין האם נעשה שינוי בקוד בעקבות זאת.

|  |  |  |
| --- | --- | --- |
| **הערה** | **תגובה** | **האם בכוונת הצוות לבצע שינוי בעקבות ההערה? יש לפרט. אם כן, מה השינוי. אם לא, מדוע** |
| פשטות | מסכימים | אין מה לשנות |
| מימוש | מסכימים | אין מה לשנות |
| מודלריות | מסכימים | אין מה לשנות |
| יעילות | מסכימים | אין מה לשנות |
| באגים | מסכימים | אין מה לשנות |
| טיפול בשגיאות | מסכימים | אין מה לשנות |

10. מקורות. יש לצטט את המקומות מהם אתם לוקחים את הנתונים השונים. יש לכלול פרומפטים לכלי AI, במידה והשתמשתם בהם.

אתר שהשתמשנו בו לזחילה: אתר salesforce <https://www.salesforce.com/eu/?ir=1>

מקור לעיבוד מידע: המחברת מהתרגול <https://colab.research.google.com/drive/1uQhw3EcBkVAHsTKI7PFvZqKH0TBeaBAn?usp=sharing#scrollTo=yfNa7qeMk7ZL>

מקור לחיבור לAI: המחברת מהתרגול <https://colab.research.google.com/drive/1TBo4saFn1BCSfgXsmREFrUl3zSQFg6CC#scrollTo=sThFl4fJtEQJ>

פרומפט לAI:

"Please create a website display for a search engine. It needs to have a search bar in the middle of the screen, with a search button next to it. At the top left of the screen add manager button and statistics button. Create the manager page- add a dropdown list with a delete button next to it, a text bar with add button next to it and generate report button, also add the graph from the given code before, and a back button at the top left. In the statistics page, add both given graphs. In the search result page create a search display, where each item in the linked list will be displayed separately and include the page title, link and number of appearances saved in the nodes. The code is written in Python and the developing environment is Google Colab so make sure to use widgets if possible. The design theme should be unicorn, so give it positive themes and vibrant colors like pink and blue but make it mature and classy"