**מבוא למחשוב ענן - סמסטר חורף התשפ"ה**

**תרגיל בית 3**- **עבודה בצוותי העבודה**

מועד הגשה: 26.1.25

המשימה בתרגיל זה: סיום בניית המערכת, ביצוע קוד ריוויו לצוות אחר

איטרצית פיתוח מס 2

קישור למחברת-

<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**

שימו לב: למטלה זו שלושה חלקים

**חלק ראשון.** עליכם להגדיר את בעלי התפקידים לאיטרציה זו (יש להחליף מתרגיל בית 2) יש לרשום מי מהנדס.ת המערכת, האחראי.ת בתרגיל זה.

טבלת הקצאת משימות:

|  |  |  |
| --- | --- | --- |
| **שם חבר הצוות** | **משימות שהוקצו** | משימות שהושלמו |
| רחל אבידן-קינג  scrum master | -עבודה על הצאט בוט  -תיאור מדדי KPI , שקיפות אלגוריתמית, מיקרו-שרתים ואתגרים -הקצאת משימות לשאר הצוות | כל המשימות הושלמו |
| אוריאן חזיזה  frontend developer | -עבודה על העיצוב  -עבודה על השלמת מסך מנהל | כל המשימות הושלמו |
| קרן קזצ'ינסקי  backend developer | -עבודה על הצאט בוט  -עבודה על בדיקות קבלה | כל המשימות הושלמו |
| מאור סיבוני  UI | -מימוש מיקרו שרתים  -עבודה על תיק מתכנת | כל המשימות הושלמו |
| דניאל עייש  product manager | -מימוש מיקרו שרתים -עבודה על תיק מתכנת | כל המשימות הושלמו |
| דניאל שחר  QA | -עבודה על השלמת מסך מנהל  -תיאור מדדי KPI , שקיפות אלגוריתמית, מיקרו-שרתים ואתגרים | כל המשימות הושלמו |

**טבלת בדיקות קבלה:**

|  |  |
| --- | --- |
| ***בדיקת קבלה*** | ***האם הבדיקה עברה?*** |
| ***זחילה בסיסית באתר והפקת אינדקס מילים: בדיקה האם המערכת מסיימת זחילה ומפיקה אינדקס עם לפחות 100 מילים.*** | ***הבדיקה עברה*** ***✅ המערכת זחלה בהצלחה באתר Salesforce והפיקה אינדקס תקין.*** |
| ***כתיבה למסד הנתונים: בדיקה אם כל הנתונים נכתבים בפעולה אחת וללא שגיאות.*** | ***הבדיקה עברה*** ***✅ כל הנתונים נכתבו למסד נתונים בכתיבה מרוכזת וללא שגיאות.*** |
| ***הסרת מילים נפוצות (Stop Words): בדיקה אם מילים כמו ""the ו-"and" מוסרות מהמסד נתונים.*** | ***הבדיקה עברה*** ***✅ המילים הנפוצות סוננו מהאינדקס בהצלחה.*** |
| ***יישום Stemming: בדיקה עם מילים עם שורש משותף מאוחסנות כשורש המילה בלבד.*** | ***הבדיקה עברה*** ***✅ המילים "run" ו-"running" אוחדו ל""run באינדקס.*** |
| ***חיפוש מילה קיימת במאגר: בדיקה האם המערכת מחזירה את כל המסמכים בהם מופיעה המילה.*** | ***הבדיקה עברה*** ***✅ המערכת החזירה מסמכים רלוונטים למילות חיפוש כמו "Salesforce".*** |
| ***חיפוש מילה שלא קיימת במאגר: בדיקה אם המערכת מחזירה הודעה שאין תוצאות.*** | ***הבדיקה עברה*** ***✅ המערכת החזירה "No Results" למילה שלא קיימת במאגר.*** |
| ***מחיקת מילה מהאינדקס: בדיקה אם המילה נמחקה מהאינדקס ומהמסד.*** | ***הבדיקה עברה*** ***✅ המילה נמחקה בהצלחה גם מהאינדקס וגם מהמסד נתונים.*** |
| ***ביצוע חיפוש עם בינה מלאכותית: בדיקה אם המערכת מחזירה תשובות מותאמות אישית דרך Gemini.*** | ***הבדיקה עברה*** ***✅ התקבלו תשובות אינטואיטביות לשאילתות המשתמש.*** |
| ***עבודה במקביל של מספר מיקרו שרתים: בדיקה אם השירותים פועלים במקביל ללא השפעה הדדית.*** | ***הבדיקה עברה*** ***✅כל השירותים עבדו בו זמנית ללא שגיאות או האטה בביצועים.*** |
| ***בדיקה שמנגנון המיון של התוצאות עובד כראוי: בדיקה אם התוצאות הכי נפוצות מופיעות בתחילת העמוד.*** | ***הבדיקה עברה*** ***✅כל התוצאות שהופיעו הכי הרבה בדפים מופיעות בתחילת דף תוצאות החיפוש.*** |
| ***בדיקת הפונקציונליות של המסכים: בדיקה שאין שגיאות במסכים.*** | ***הבדיקה עברה*** ***✅כל הכפתורים לחיצים וכל המעברים בין הדפים עובדים כמו שצריך.*** |
| ***בדיקה שההורדה של דו"ח החיפושים במסך מנהל עובד כראוי: בדיקה שאין שגיאות בהורדה.*** | ***הבדיקה עברה*** ***✅הדו"ח יורד ללא שגיאות עם התאריך של היום בפורמט של png.*** |

***חלק שני: בניית המערכת*** (80 נקודות)

עליכם לסיים את בניית המערכת.

בפרט, יש לוודא קיום של כלל האלמנטים הבאים:

1. כל הפיצ'רים שהתחלתם לממש בתרגיל בית 2 - יש לוודא שכל המסכים תקינים ועובדים כראוי, **בקולאב**, ולעבוד עם firebase לצורך שמירת נתונים . אין להשאיר קישורים או כפתורים למסכים שלא קיימים.

יש לוודא שניתן להריץ את הפרויקט מהמחברת בלבד , ללא כל צורך בהטענה של קבצים נוספים או יציאה למערכת חיצונית (כגון FLASK). (30 נקודות)

1. שימוש ב -microservices (תרגול 7) - עליכם לעשות שימוש לפחות בשניים. פרטו באילו שירותים עשיתם שימוש, ומה היתרון בשימוש בשירותים אלו.(10 נקודות)

השתמשנו בשלושה מיקרו שרתים לצורך פיתוח מודולרי ויעיל:

**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.

בין היתרונות בשימוש בשירותים אלו הם חלוקה לשירותים עצמאיים, מה שמקל על פיתוח ותחזוקה. באגים או שגיאות בשירות מסוים אינם משפיעים על המערכת כולה. בנוסף ה- microservices מאפשרים עיבוד של משימות שונות בו-זמנית על ידי שירותים שונים. אם שירות אחד נפרץ או נכשלה בו הפעולה, ניתן לבודדו מבלי לפגוע במערכת כולה.

השימוש ב-microservices עם AI ועיבוד מידע מבטיח מערכת מודולרית, יעילה ואמינה שניתן להתאים אותה לצרכים ספציפיים, להרחיב אותה בקלות ולהפיק ממנה תועלת מרבית לטווח הארוך.

1. צטבוט שישולב במערכת (כפי שלמדתם בתרגול 8), הכולל מידע אשר מגיע מה - DB שלכם, ועונה בצורה אינטליגנטית לשאלות המשתמש. (15 נקודות).
2. מהם ה -KPI הרלוונטיים בפרויקט שלכם? הסבירו? (10 נקודות)

ה- KPI הרלוונטיים בפרויקט שלנו הם:

* עמידה במטרות הפרויקט:

הפרויקט שלנו עמד בזמני ההגשות עבור כל מטלה.

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

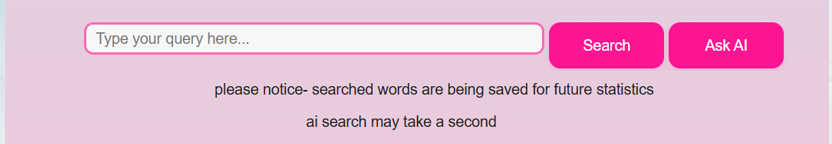
כלל חברי הצוות היו שותפים לעשייה ועמדו כל אחד במשימתו לפי חלוקת התפקידים שנעשתה בכל שלב.

* עמידה בתנאי אתיקה וחוקיות:

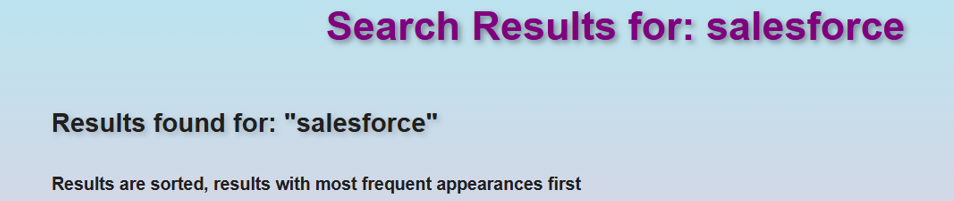
נמנענו ממצבים בעיתיים כגון, פניות לאתר הנזחל בתדירות גבוה שהייתה עלולה לגרום לעומס יתר על האתר שניגשנו אליו.

* יעילות:

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

1. שקיפות אלגוריתמית – כיצד הבהרתם למשתמשים את האלגוריתמים והנתונים הנאספים בקוד שלכם? (5 נקודות)  
   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.

1. אתגרים שעלו במהלך העבודה , וכיצד התמודדתם איתם (לדוגמא- DB גדול מדי, יצירת ויזואליזציה, מבנה הקוד וכו) (10 נקודות)

* 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 במדוייק.

***חלק שלישי : סגירת הפרויקט \_( 20 נקודות)***

בחלק זה תכינו מסמכים המתארים את הפרויקט שלכם:

1. יש לבנות תיק למתכנת הכולל את שמות כל הקבצים המרכזיים, פונקציות מרכזיות, קטעי קוד/תבניות עיצוב מעניינים שהשתמשתם בהם.
2. יש לבנות תיק למשתמש , הכולל הסבר כללי על המערכת , פירוט מסכים, מעברים בין מסכים והסבר על טעויות אפשרויות.
3. התייחסו בתיק המשתמש לשקיפות אלגוריתמית – כיצד הבהרתם למשתמשים את האלגוריתמים והנתונים הנאספים בקוד שלכם?
4. יש להכין סרטון קצר של 30-60 שניות, המתאר את השימוש במערכת. הסרטון משמש כ – elevator pitch  למערכת שלכם, כלומר יש לכלול בו הסבר מקצועי ועם זאת שיווקי , המדגיש את האלמנטים המיוחדים של המערכת שבניתם. יש להגיש את הסרטון בפורמט mp4.

**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 שמשוייך 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 שמשוייך 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 פרומפט 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)

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 לאינקדס. 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.
* השמתמש 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.