המכללה האקדמית להנדסה בראודה A black background with purple letters

Description automatically generated

**המחלקה להנדסת תוכנה - מבוא למחשוב ענן**

**פרויקט**

מועד הצגה: שבוע 11 - בזמן ההרצאה

מועד הגשה: **9.2.25**

**הסעיף נלקח מתרגיל בית 1**

**הסעיף נלקח מתרגיל בית 2**

**הסעיף נלקח מתרגיל בית 3**

בסיום הסמסטר, מגיש כל צוות סטודנטים את הפרויקט עליו עבדו במהלך הסמסטר. הפרויקט ימומש בסביבת גוגל קולאב, עם מסד נתונים FireBase, ויאוחסן בתיקיית גיט של הצוות.

על הפרויקט להכיל את כל האלמנטים הממומשים במלואם כפי שתוארו בתרגילי בית 2+3.

יש להגיש את הפרויקט בדו"ח אשר אורכו לא יעלה על 15 עמודים , בכתב Arial 12. יש להתייחס לנקודות הבאות:

- [**קישור למחברת הגוגל קולאב**](https://colab.research.google.com/drive/1UyGNd-QvtdfAn-u8FGnhTCUv0nSMXb0K?usp=sharing)

[**- קישור לתיקיית הגיט** של הצוות](https://github.com/TRiZKy/CloudAccessBear).

1. מהות המוצר ומרכיביו

מהות המוצר היא למקד את אתר אליבאבא יותר למתקני טלפונים.

אליבאבא מונעת בצורה אקטיבית זחלנים אז הפרויקט היה שונה משאר הקבוצות בכך ש

1.1 מודולים ופיצ'רים מעניינים

פיצ'ר **זיהוי מילים נרדפות בחיפוש:**

הוספנו למערכת פיצ'ר שמרחיב שאילתות חיפוש באמצעות מילים נרדפות ודומות. כך, משתמש יכול לחפש "mobile" ולקבל תוצאות הכוללות "smartphone", "cellphone", ועוד. הפיצ'ר משפר אתחוויית החיפוש על ידי התאמת התוצאות גם אם המשתמש לא השתמש במילים המדויקות.   
הפיצ'ר ממומש על ידי הפונקציה **expand\_query\_with\_synonyms** שמרחיבה את השאילתה על בסיס מילון של מילים נרדפות ומשולב בפונקציית החיפוש **search\_index** שתרחיב את החיפוש למוצרים התואמים למילים הנרדפות, כך שמשתמש יקבל את מירב התוצאות המתאימות לחיפושו.בממשק ניהול האינדקס, המנהל יכול לצפות, לערוך ולהוסיף מילים נרדפות, מה שמאפשר לו לשלוט בהשפעת הפיצ'ר על תוצאות החיפוש.

**פיצ'ר תיקון מילים- AutoCorrect :**

הפיצ'ר של תיקון אוטומטי (Autocorrect) נועד לשפר את חוויית המשתמש על ידי זיהוי ותיקון טעויות הקלדה או שגיאות חיפוש בשאילתות טקסטואליות. בפונקציה זו, השאילתה מפורקת למילים בודדות באמצעות ביטוי רגולרי שמזהה מילים בפורמט טקסטואלי. לאחר מכן, כל מילה מושווית למאגר של מונחים (index\_terms) באמצעות הפונקציה get\_close\_matches מספריית difflib - פונקציה זו מוצאת התאמות קרובות בין המילה שנבדקת לבין המונחים במאגר, בהתבסס על דמיון טקסטואלי.

כאשר נמצאת התאמה קרובה עם רמת דמיון של לפחות 80% (כפי שהוגדר על ידי הפרמטר cutoff) המילה הקרובה ביותר נבחרת ונוספת לשאילתה המתוקנת. אם לא נמצאה התאמה קרובה, המילה המקורית נשמרת כפי שהיא. בסיום התהליך, המילים המתוקנות מאוחדות חזרה לכדי שאילתה מתוקנת ומוחזרות כתוצאה.

הפיצ'ר מספק מענה שימושי במיוחד למקרים של שגיאות הקלדה, מילים חסרות או שגיאות איות, ומאפשר למשתמשים לקבל תוצאות מדויקות יותר למרות טעויות במקור.

1.2 microservices שנעשה בהם שימוש

PageFetcher אחראי לאיסוף נתונים מURL .

DataExtractor אחראי על חילוץ נתונים רלוונטיים מתוך תוכן HTML

IndexBuilder בונה אינדקס המבוסס על הנתונים שהופקו.

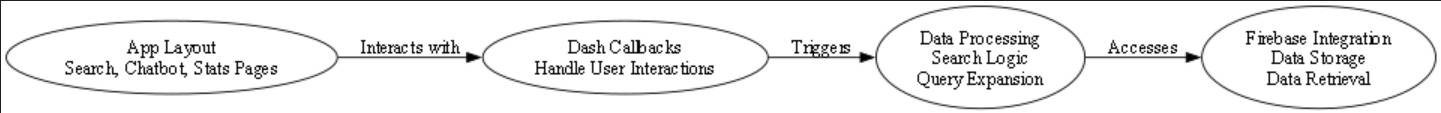
QueryProcessor מעבד שאילתות משתמש ומספק תוצאות חיפוש, כולל הרחבת שאילתות עם מילים נרדפות ותיקון אוטומטי.

FirebaseService מטפל בשירותי העלאה ואחסון נתונים ב .Firebase

1.3 KPI מרכזיים

1. זמן תגובה של השירותים:  
   זמן התגובה הממוצע של מנגנון החיפוש לאחר הזנת שאילתה (במילישניות).
2. דיוק המידע בצ'אט בוט:  
   אחוז השאלות שמקבלות מענה נכון בהתבסס על נתונים קיימים.
3. הצלחת פיצ'ר תיקון המילים  
   אחוז השאילתות בהן המשתמשים קיבלו תוצאות משופרות בעקבות תיקון שגיאות.
4. הצלחת פיצ'ר זיהוי מילים נרדפות:  
   אחוז השאילתות שבהן הרחבת מילים נרדפות שיפרה את הרלוונטיות של התוצאות.
5. יעילות האינדקס  
   זמן ממוצע לעדכון אינדקס בעת הוספת/עדכון נתונים.
6. יציבות המערכת:  
   אחוז הזמן שבו המערכת זמינה ללא קריסות.
7. חוויית משתמש:  
   ציון שביעות רצון ממשתמשים לגבי קלות השימוש בממשק החיפוש.

2. ארכיטקטורת המערכת

2.1 תרשים ומאפיינים מרכזיים (מעודכן)

**שכבת המצגת (Presentation Layer - UI Layer)**

* **רכיבים:**
  + אפליקציית Dash (app.layout): מיישמת ממשקי משתמש לחיפוש, צ'אטבוט והצגת סטטיסטיקות.
  + עמודים מותאמים לפעילויות שונות: חיפוש, ניהול, צ'אטבוט וסטטיסטיקות.
* **קוד רלוונטי:**
  + הגדרות של search\_page, manager\_page, chatbot\_page, statistics\_page.

**שכבת היישום (Application Layer - Service Layer)**

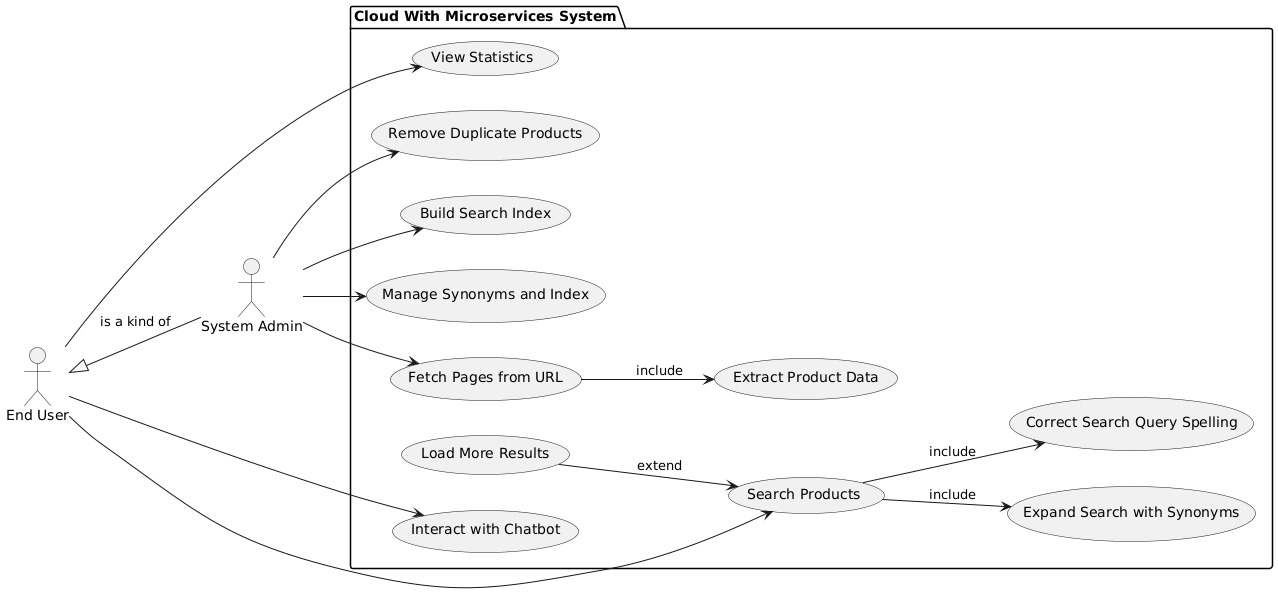
* **רכיבים:**
  + פונקציות Callback ב-Dash: מטפלות באינטראקציות המשתמש, בניהול מצב, ובהפעלת חישובים (לדוגמה, חיפוש, תגובות צ'אטבוט).
* **קוד רלוונטי:**
  + הגדרות של @app.callback עבור עמודים ופונקציונליות שונות.

**שכבת הלוגיקה העסקית (Business Logic Layer - Domain Layer)**

* **רכיבים:**
  + פונקציות לאיסוף נתונים, ניתוחם ועיבודם.
  + לוגיקת חיפוש ובניית אינדקס.
  + הרחבת שאילתות ותיקון אוטומטי של שאילתות.
* **קוד רלוונטי:**
  + פונקציות כמו: fetch\_pages, extract\_data, build\_index, search\_index, expand\_query\_with\_synonyms, autocorrect\_query.

**שכבת גישה לנתונים (Data Access Layer - Persistence Layer)**

* **רכיבים:**
  + אינטגרציה עם Firebase לאחסון ושליפת נתונים.
* **קוד רלוונטי:**
  + יוזמה ושימוש ב-firebase.FirebaseApplication דרך FBconn.post.

2.2 Use case מעודכן של פונקציונליות המערכת.  


2.3 דרישות פונקציונליות

.1 המערכת תבצע עיבוד של תוכן דפי HTML ותוציא פרטי מוצר

.2 המערכת תבנה אינדקס המבוסס על כותרות המוצרים.

.3 המערכת תאפשר אחזור מהיר של נתונים על בסיס מונחי חיפוש.

.4 המערכת תרחיב שאילתות חיפוש על ידי שימוש במילים נרדפות.

.5 המערכת תכלול מנגנון לתיקון אוטומטי של שגיאות כתיב בשאילתות חיפוש.

.6 המערכת תאפשר ניהול מונחים ומילים נרדפות באינדקס.

.7 המערכת תאפשר הסרה של מזהי מסמכים (DocIDs) לפי הצורך.

.8 המערכת תציג נתונים סטטיסטיים על המידע המאונדקס

.9 המערכת תספק צ'אטבוט אינטראקטיבי למענה על שאלות המשתמש.

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

דרישות לא פונקציונליות:

1.Performance:האפליקציה תספק זמן תגובה מהיר לפעולות חיפוש והשוואה, באמצעות מנגנוני קאשינג מתקדמים וניהול יעיל של בקשות.

2.Usability:האפליקציה תהיה בעלת ממשק משתמש אינטואיטיבי, מותאם למשתמשים שאינם טכניים, באמצעות עיצוב ידידותי ושפה פשוטה.

3. Scalability: האפליקציה תתמוך במשתמשים מרובים ובבקשות רבות בו-זמנית, באמצעות שימוש בשירותים מבוססי ענן שיאפשרו התמודדות עם גידול בביקוש.

4.Maintainability :האפליקציה תאפשר תחזוקה ושדרוגים בקלות, באמצעות תיעוד קוד ברור,וכלי ניטור ודיווח על באגים.

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

**ציון ה-SUS שהתקבל משאלון ה-SUS על המערכת שלנו הינו: 86  
-**מעיד על כך שהמערכת נתפסת כקלה לשימוש, והמשתמשים ככל הנראה חווים שביעות רצון גבוהה מהשימוש בה, לכן ראינו לנכון להמשיך לחזק את הפיצ'רים הקיימים שכן הפידבק היה חיובי (אך לא מושלם ולכן חיזקנו).

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

**תיק תחזוקה**

**קבצים ואובייקטים מרכזיים**

1. **הקובץ הראשי**:
   * **שם**: cloud\_with\_microservices.py
   * **מטרה**: מימוש מערכת לאיסוף נתונים באמצעות זחילת רשת, עיבוד נתונים, וניהול שאילתות באמצעות מנוע חיפוש המבוסס על ממשק משתמש.
2. **מחלקות עיקריות**:
   * **PageFetcher**:
     + אחראית על הורדת תכני HTML מדפי רשת לפי כתובת בסיס.
   * **DataExtractor**:
     + חולצת נתונים מתוך קובצי HTML, כולל פרטי מוצרים כמו שם, מחיר ותמונה.
   * **IndexBuilder**:
     + בונה אינדקס הפוך שמאפשר חיפוש יעיל במאגר הנתונים.
   * **QueryProcessor**:
     + מעבד שאילתות משתמש, מרחיב אותן באמצעות מילים נרדפות ומבצע חיפושים באינדקס.
   * **FirebaseService**:
     + מספק פונקציות להעלאת נתונים למסד נתונים מבוסס Firebase.
3. **מרכיבי אפליקציית Dash**:
   * כולל דפים שונים כמו דף חיפוש, דף ניהול, צ'טבוט ודף סטטיסטיקות, באמצעות Dash.

**תיעוד פונקציות**

**1. PageFetcher**

* **\_\_init\_\_(base\_url)**: Initialize the PageFetcher with a base URL template.  
  Parameters: base\_url (str): The URL template where the page number will be inserted.
* **fetch\_pages(start, end)**: Fetches pages from start to end and returns their parsed HTML content as a dictionary, where each key is in the format 'page\_X' and the value is the BeautifulSoup object. Parameters: start (int): Starting page number (default is 1). end (int): Ending page number (default is 33). Returns: A dictionary of page contents (key: 'page\_X', value: BeautifulSoup object).

**2. DataExtractor**

* **extract\_data(page)**: Extracts product details from a given page's HTML content.  
  Parameters: page (BeautifulSoup): The parsed HTML content of a page.  
  Returns: **List**: A list of dictionaries, each containing product details such as title, URL, price, image URL, and rating.
* **remove\_exact\_duplicates(products):** Removes duplicate products from a list based on title and URL.  
  Parameters: products (list): A list of dictionaries, where each dictionary contains product details (e.g., title, URL).  
  Returns: **List**: A list of unique product dictionaries, removing any duplicates with the same title and URL.

**3. IndexBuilder**

* **\_\_init\_\_(self)**: Initializes the IndexBuilder with a PorterStemmer to handle word stemming during indexing.
* **remove\_stop\_words(index)**: Removes common stop words (e.g., "a", "the", "and") from the index.  
  Parameters: index (dict): The original index to process.  
  Returns: A filtered index without stop words.
* **apply\_stemming(index)**: Applies stemming to the words in the index to reduce them to their root forms.  
  Parameters: index (dict): The original index to process.  
  Returns: A new index with stemmed words as keys.
* **build\_index(database)**: Constructs an index from a list of product records.  
  Parameters: database (list): A list of products, where each product is a dictionary containing  
  Returns: A list of dictionaries, each containing a term (word), associated document IDs, and product details.

**4. QueryProcessor**

* **\_\_init\_\_(index, synonyms)**: Initializes the QueryProcessor with an index (terms and product details) and a set of synonyms.
* **expand\_query\_with\_synonyms(query)**: Expands the input query by adding synonyms for each word.  
  Parameters: query (str): The query to expand.  
  Returns: A list of expanded query words.
* **remove\_stop\_words(self, query\_words)**: Removes common stop words (e.g., "a", "the", "and") from the query.  
  Parameters: query\_words (list) or str: A list of words or a string to process.  
  Returns: A list of query words with stop words removed.
* **search\_index(query)**: Processes the query (removes stop words, expands with synonyms, stems), then searches the index for matching products.Parameters: query (str): The query to search.  
  Returns: A dictionary of matching products, sorted by the number of term matches.
* **autocorrect\_query(query, index\_terms)**: Autocorrects the query by finding the closest matching terms from the index using difflib.  
  Parameters: query (str): The query to autocorrect. index\_terms (list): List of terms from the index for matching.  
  Returns: A corrected query string with words replaced by the closest matches.

**5. FirebaseService**

* **\_\_init\_\_(url)**: Initializes the FirebaseService with the provided Firebase URL to establish a connection.
* **upload\_data(path, data)**: Uploads data to the specified Firebase path.  
  Parameters: path (str): The path in the Firebase database where the data will be stored. data (dict): The data to upload to Firebase.  
  Returns: The result of the post request (response from Firebase).

**6. ממשקי Dash**

* **search\_page**: ממשק החיפוש.
* **manager\_page**: ממשק לניהול מונחים, מילים נרדפות ועדכוני אינדקס.
* **statistics\_page**: מציג סטטיסטיקות חזותיות על מונחים באינדקס.
* **chatbot\_page**: ממשק צ'טבוט לממשק משתמש.

**7. Callbacks של Dash**

* מטפלים באינטראקציות משתמש, כמו חיפוש, עדכון נתוני אינדקס, ניהול מילים נרדפות ואינטראקציות עם הצ'טבוט.

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

**תיק למשתמש:**

**הסבר כללי על המערכת:**

מערכת מיועדת למשוך נתונים מהאתר "Alibaba", לעבד אותם ולספק ממשק משתמש אינטראקטיבי מבוסס Dash בGoogle Colab, המערכת מציעה חוויית משתמש אינטואיטיבית הכוללת חיפוש מותאם אישית, עיבוד נתונים, הצגת סטטיסטיקות וניהול אינדקסים. הפלטפורמה תומכת במעברים בין מסכים בצורה חלקה ומאפשרת עבודה יעילה ונוחה.

**פירוט מסכים:**

**1. מסך חיפוש – Search Page**:

תיאור מסך:

מסך זה מאפשר למשתמש להזין שאילתת חיפוש בתיבת חיפוש.  
- שימוש באינדקס פנימי המאפשר חיפוש חכם ומותאם אישית למשתמש.  
- התאמה אוטומטית של השאילתה במקרה של שגיאות הקלדה (autocorrect).מעברים בין מסכים:  
- לחיצה על כפתור Search תעביר למסך תוצאות החיפוש.  
-לחיצה על Statistic Page תעביר למסך הסטטיסטיקות.

-לחיצה על Chat Bot תעביר למסך הצ'אט בוט.  
- לחיצה על Manager Page תעביר למסך ניהול אינדקסים.

**2. מסך תוצאות החיפוש – Results Page:**תיאור מסך:  
מסך זה מציג את תוצאות החיפוש לפי השאילתה שהוזנה.  
- מוצגת רשימה הכוללת: כותרת המוצר, מחיר, דירוג, קישור למוצר באתר Alibaba ותמונה.  
- כפתור Load More שיטען תוצאות נוספות.מעברים בין מסכים:

- לחיצה על Search Pageתעביר למסך החיפוש.  
-לחיצה על Statistic Page תעביר למסך הסטטיסטיקות.

-לחיצה על Chat Bot תעביר למסך הצ'אט בוט.  
- לחיצה על Manager Page תעביר למסך ניהול אינדקסים.

**3. מסך סטטיסטיקות - Statistic Page:**תיאור מסך:  
מסך זה מציג גרף הממחיש את הקשר בין מחיר המוצר לדירוג שלו.  
מעברים בין מסכים:  
- לחיצה על כפתור Back to Search Page תעביר למסך החיפוש.

- לחיצה על כפתור Back To Manager Page תעביר למסך ניהול אינדקסים.

**4. מסך צ'אט בוט – Chat Bot:**תיאור מסך:  
מסך המאפשר למשתמש לשאול שאלות ולקבל תשובות מצ'אט בוט המבוסס על הנתונים שנאספו.  
- המערכת תומכת בשאילתות שונות כמו חיפוש מידע כללי על מוצרים או סטטיסטיקות שנאספו.

מעברים בין מסכים:

- לחיצה על Search Pageתעביר למסך החיפוש.  
-לחיצה על Statistic Page תעביר למסך הסטטיסטיקות.

-לחיצה על Chat Bot תעביר למסך הצ'אט בוט.  
- לחיצה על Manager Page תעביר למסך ניהול אינדקסים.  
- לחיצה על כפתור Send תעביר תשלח לצ'אט שאלה שנרשמה.

**5. מסך ניהול אינדקסים: Manager Page -**תיאור מסך:  
מסך המיועד לניהול אינדקסים פנימיים של המערכת, כולל אפשרות הוספה, מחיקה ועדכון של אינדקסים.  
מעברים בין מסכים:  
- לחיצה על כפתור Back to Search Page תעביר למסך החיפוש.

**1. מסך חיפוש – Search Page**:

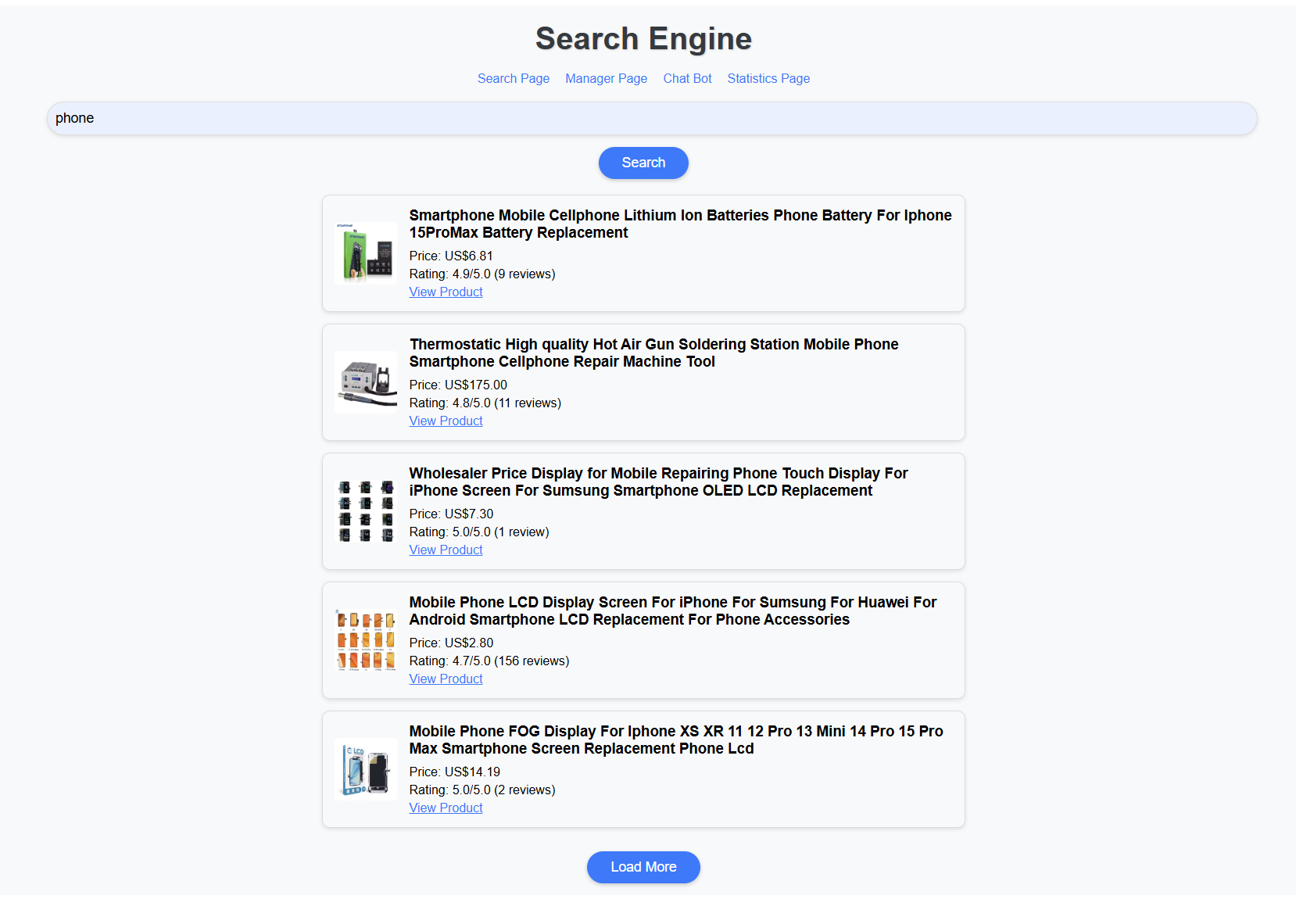
A screenshot of a search engine

Description automatically generated

A screenshot of a computer

Description automatically generated

**2. מסך תוצאות החיפוש – Results Page:**



תמונה שמכילה טקסט, צילום מסך, גופן, קו

התיאור נוצר באופן אוטומטי

**3. מסך סטטיסטיקות - Statistic Page:**

A screen shot of a graph

Description automatically generated

תמונה שמכילה טקסט, צילום מסך, גופן, קו

התיאור נוצר באופן אוטומטי

**4. מסך צ'אט בוט – Chat Bot:**

תמונה שמכילה טקסט, צילום מסך, גופן, קו

התיאור נוצר באופן אוטומטי

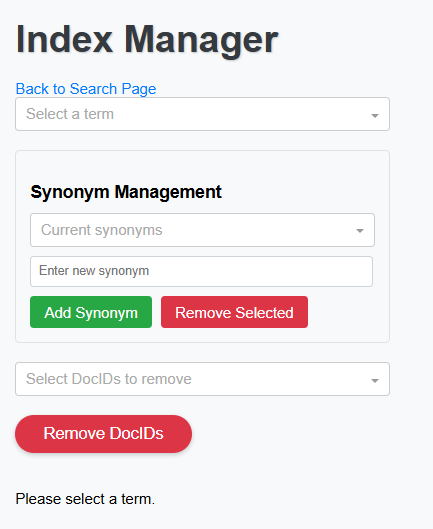
A screenshot of a chatbot

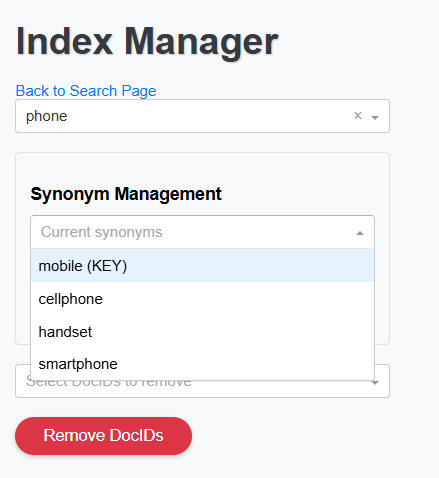
Description automatically generated

תמונה שמכילה טקסט, צילום מסך, גופן, קו

התיאור נוצר באופן אוטומטי

**5. מסך ניהול אינדקסים: Manager Page -**





**שגיאות אפשריות:**

**1. מחיקת אינדקסים:**

- מחיקת אינדקס שהוא KEY במילון המילים הנרדפות תגרום למחיקת כל המילים הנרדפות הקשורות אליו.

לדוגמה: אם נמחק את המילה mobileשמוגדרת כ KEYבמילון, כל המילים הנרדפות של mobile כמו phoneימחקו גם הן. **2. אינטראקציה עם צ'אט בוט:**

ייתכן שמשתמש יזין שאלה שאינה נתמכת או שאינה מחזירה תוצאה רלוונטית.  
3. בפונקציה **apply\_stemming:**

השימוש בפונקציה זו עלול להמיר מילים לצורה סטנדרטית באופן שגוי או לא צפוי, מה שיגרום לכך שבמסך ניהול האינדקסים, מילה מסוימת לא תימצאבשורת החיפוש:  
Select a term.לדוגמה: אם המשתמש מחפש את המילה ,mobile ייתכן שהפונקציה תמיר אותה לmobil וכך לא יתקבלו תוצאות של מילים נרדפות עבור מילת החיפוש המקורית.

1. שקיפות אלגוריתמית – כיצד הבהרתם למשתמשים את האלגוריתמים והנתונים הנאספים בקוד שלכם?

**שקיפות אלגוריתמית:**  
  
**הודעות מערכת**: הצגת הסברים על תיקון והרחבת שאילתות, למשל " האם התכוונת ל smartphone ? "

**ניהול מילים נרדפות**: ממשק לעריכת מילון המילים הנרדפות לצפייה ושליטה בתוצאות החיפוש

**תיעוד נתונים**: הצגת תוצאות מורחבות ומותאמות באופן ברור למשתמש.

**סטטיסטיקות חזותיות**: עמוד המציג מונחים נפוצים ו- Cloud Word לתובנות על עיבוד המידע.

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

אתגרים שעלו במהלך העבודה וכיצד התמודדנו:

* שימוש ב-GUI דרך Google Colab לא היה נוח: במהלך העבודה, נתקלנו בקושי לשלב ממשק משתמש גרפי (GUI) בצורה נוחה בתוך Google Colab, שאינה מותאמת לתמיכה מלאה בווידג'טים אינטראקטיביים.

פתרון: השתמשנו בDash -, פלטפורמה מתקדמת המאפשרת יצירת ממשקים גרפיים אינטראקטיביים בצורה אינטואיטיבית, ותומכת גם בריצה בתוך Google Colab. DASH סיפקה פונקציות מוכנות רבות לUI מה שהפך את העבודה על ממשק המשתמש ליעילה ונוחה יותר, תוך שימוש מלא במגבלות וביכולות של Colab.

* חסימת זחלנים על ידי אליבאבא :ניסיונות לגשת לתוכן האתר של Alibaba באמצעות זחלנים (web scrapers) נחסמו על ידי מנגנוני האבטחה של האתר.   
  פתרון: במקום לנסות לעקוף את המנגנונים, בחרנו להוריד את קבצי ה-HTML של הדפים באופן ידני ולעבוד עליהם באופן מקומי. שיטה זו אפשרה לנו לבצע ניתוח של המידע הרצוי מבלי להפר את כללי השימוש של האתר.

9. בשבוע 11 בקורס תציגו את המערכת שלכם.

ההצגה תכלול הרצת המערכת/סרטון, וכן מצגת המכילה את סעיפים 1-8 (כ-15 דקות) , אשר תוצג בשיעור האחרון, בה יוצגו הממצאים העיקריים.המצגת תאוחסן בתיקיית גיט של הקבוצה. ניתן לעצב אותה כפוסטר.

כל חברי הצוות חייבים להשתתף במצגת, וכן יש להקפיד על מצגת אסתטית ונקייה משגיאות איות.

עליכם להתייחס למצגות בסשן שלכם. ההצגה ומילוי המשוב הם **חובה לכל חברי הצוות על מנת לקבל ציון.**

לאחר ההצגה: תקבלו באופן אנונימי את המשובים של חבריכם, וכן את המשוב שלנו.

יש להגיש את הטבלה הבאה , תוך התיחסות למשובים שקיבלתם:

|  |  |  |
| --- | --- | --- |
| הערת משוב | האם לדעתכם יש צורך בשינוי במערכת בעקבות ההערה? | נימוק |
| עיצוב המערכת אפשר לעצב יותר יפה/צבעוני | כן | הוספנו את הלוגו של הקבוצה שלנו למסך הבית |
| הוספת עיצוב קטן בנביגציה בין דפים. במקום טקסט כחול, כפתור בסיסי | לא | אנחנו בכוונה עשינו את העיצוב כך שהוא מזכיר גם את העיצוב המקורי של גוגל בתור מחווה אליהם |
| הייתי מוסיף עוד פונקציונליות אולי, אפשרות לעשות את המערכת יותר personalized | לא | לדעתנו עמדנו בדרישות הפרויקט וישנן מספיק פיצרים ופונקציונליות במסכים השונים |
| להוסיף גרפים | לא | לא רואים לנכון להוסיף גרפים נוספים שאין להם משמעות פיזית למשתמש |

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.

הערות שקיבלנו בcode review:

|  |  |  |
| --- | --- | --- |
| **הערה** | **תגובה** | **האם בכוונת הצוות לבצע שינוי בעקבות ההערה? יש לפרט. אם כן, מה השינוי. אם לא, מדוע** |
|  | הקוד מבצע את החיפוש באופן יעיל ומהיר בזכות פנייה לאינדקס ומוציא תוצאות חיפוש מתאימות, הצגת נתונים במסך אדמין מהמסד מתבצעת באופן איטי. | אין לנו מה לעשות עם שיפור זמן בהצגת הנתונים |
|  | ניתן לבצע חלוקה ברורה, קיים חלק אחד גדול וקשה להבין מה קורה בו. | נבצע חלוקה ברורה |
|  | מציג תוצאות עבור stop words ולא מתעלם מהם. | נתקן את המילות מהשאילתה לפני חיפוש השאילתה |
|  | דייב אמר שימושי אך לא עורר רצון להשתמש בו. | מקובל |
|  | מתועד חלקית. | בסדר, נוסיף תיעוד |
|  | הקוד לא מובן, קיימים הרבה שורות. ניתן לשפר את קריאת הקוד על ידי תיעוד. | נתעד. |

שינויים שהוספנו בעקבות הקוד ריוויו:

הוספנו חלוקה לבלוקים בתוך הקולאב.

הוספנו שבחיפוש QUERY נתעלם מ stop words גם במהלך התיקון מילים.

בנוגע לjson במסך המנהל אף על פי שכל הJSON הוא cached בקוד.

כאשר מריצים את זה על מחשב פרטי ולא על הקולאב זה רץ הרבה יותר מהר.

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

**תשובה**: לא השתמשנו בכלי AI יותר מדי, בעיקר לדברים קטנים של פייתון. והשתמשנו בgit hub copilot בתוך הpycharm שיעזור יחד עם dash של הui שיש לנו בפרויקט.

הדו"ח יוגש לאחר סיום הסמסטר.

הנחיות:

1. יש להגיש את התרגיל בצוותים שנקבעו.
2. חובת הצגה בקורס על כל הסטודנטים בקבוצה. סטודנט אשר לא יציג, לא יקבל ציון בקורס.
3. שימו לב כי כל העבודות חייבות להיות שונות זו מזו. עבודות שייראו דומות ייפסלו ויינתן עליהן ציון 0.

בהצלחה!