**מבוא למחשוב ענן – הגשת פרויקט – צוות Wolf**

מגישים: דניאל פלדמן – 208134700

ארתור צ'רני – 206466500

איליה לזרב – 212177943

שגיא יוספוב – 315030155

אביב רז - 207940875

קישור למחברת גוגל קולאב: [Colab Notebook](https://colab.research.google.com/drive/1wPwVO5elH74WM5AfwxQOAjjOwx6G68Pc?usp=sharing)

קישור לתיקיית הגיט: [Git Folder](https://github.com/SagiYosofov/WOLF)

**1. מהות המוצר ומרכיביו:**

1.1. מודולים ופיצ'רים מעניינים:

המודולים שיש לנו הם מודולים של דף חיפוש,דף צ'טבוט, דף מנהל, דף סטטיסטיקות, דף מידע ודירוג, כל דף פועל עצמאית מהדפים האחרים ואחראי על הפעולות שעליו לבצע, בנוסף קיים מיקרו שירות הבודק אם לינק מסויים הוא תחת איסור של robots.txt של האתר של מייקרוסופט אז'ור, ומיקרו שירותים נוספים לבניית האינדקס וביצוע סקרייפינג לאתר.

הפיצ'רים שהפרויקט שלנו כולל הם חיפוש באתר אז'ור,שיחה עם צ'טבוט, ניהול האינדקס ע"י הוספה/מחיקה/עריכה של מילה ורשימת הלינקים שלה,צפייה בסטטיסטיקות, דירוג המערכת.

1.2. microservices שנעשה בהם שימוש:

א. שירות המיקרו הראשון בו השתמשנו הוא שירות האינדקס מתרגול 7, לצורך יצירת האינדקס ורשימת המסמכים. היתרון בשימוש בו היה שלא היה צורך להמציא את הגלגל מחדש ולכתוב קוד שיעבור מילה מילה וימפה את המילים לדפים.

ב. שירות המיקרו השני בו השתמשנו הוא מיקרו שירות בשם RobotsChecker שכתבנו בעצמנו – השירות מקבל base url, בדוגמא שלנו "https://azure.microsoft.com" ומוסיף לו את הסיומת "/robots.txt", מוריד את הקובץ ומפענח אותו בעזרת ספריית urllib.robotparser, ולאחר מכן נותן את האפשרות לבדוק האם ניתן לבצע סקרייפ ללינק מסוים והאם לינק מסוים עדיין שייך לאתר שכתובתו היא הbase url.

היתרון בשימוש בו הוא שכעת יש לנו מיקרו שירות גנרי שנוכל להשתמש בו לכל פרויקט בו מצטרך לבצע scraping לאתר מסוים ללא כל חשש שיחסמו אותנו בגלל ניסיון לגשת לכתובת אסורה לשימוש ושבמקום לחשוב איך להכניס את שורות הקוד הרלוונטיות למחלקה בה השתמשנו להורדת תוכן האתר, כל מה שהיינו צריכים לעשות הוא ליצור מופע של RobotsChecker ולקרוא ל2 מתודות על כל לינק שרצינו לבדוק.

ג. מיקרו שירות שלישי הוא DatabaseService אשר משמש אותנו לתקשורת עם מסד הנתונים בFirebase לצורך העלאה והורדה של נתונים, חשבנו על הרעיון רק לקראת סוף הפרויקט ולכן בחרנו להשתמש בו רק במספר חלקים בפרויקט ולא בכל מחלקה.

היתרון בשימוש בו הוא שניתן לחסוך כתיבה מחדש של פקודות להעלאה והורדה של קבצים, ובעתיד נוכל רק להחליף את הקישור למסד הנתונים למסד נתונים אחר בפיירבייס ולהשתמש בו שוב. בנוסף חוסך בכוח חישובי מכיוון שמוגדר כסינגלתון ולכן בעצם לא ניצור עוד ועוד מופעים של החיבור למסד.

1.3. KPI מרכזיים:

א. זמן תגובה – לדעתנו תוכנה שמגיבה במהירות היא תוכנה יעילה וטובה למשתמש, ניסינו בכל שלב לבצע כל פעולה בדרך הכי יעילה כדי לחסוך בזמן חישוב ולראות למשתמש נתונים כמעט בזמן אמת.

כדי לעמוד ביעד זה, מדדנו את זמן התגובה של המערכת על ידי שימוש בטיים סטמפים בכל אינטראקציה קריטית, כגון מרגע לחיצת המשתמש על כפתור ועד להצגת התוצאות על המסך. נתוני זמן התגובה הודפסו אך ורק לצרכים טכניים ונגישים אך ורק למפתחים (נמדדו בשלב הפיתוח ולאחר נמחקו מהמערכת בשלב release).



A computer screen shot of a computer code

Description automatically generated

יעדינו הוא פחות מ2 שניות והצלחנו לעשות זאת בשנייה אחת.

ב. שיעור שגיאות – פעלנו קשות כדי לוודא שהמשתמש יקבל את התוצאות המדויקות ביותר והרלוונטיות ביותר לחיפוש שלו (ראו אתגר מס' 2) ולכן שיעור השגיאות באפליקציה הוא מדד להצלחה שלנו.

מכיוון שאין לנו הכלים לנתח את תוכן הדף, בחרנו למדוד את שיעור השגיאות בעזרת דירוג של המשתמש, בלשונית מידע ודירוג, נתנו את האופציה למשתמש לשלוח לנו דירוג מספרי של עד כמה תוצאות החיפוש תאמו לשאילתה שלו, 10 הוא 100% ו1 הם 0%.

שיעור השגיאות הכללי הוא הממוצע של כל הדירוגים של הלקוחות.

היעד שהצבנו לעצמנו הוא 80% ומעלה.

ג. שביעות רצון הלקוח - שביעות רצון הלקוח עמדה במרכז העשייה שלנו. חוויית המשתמש הייתה לנגד עינינו מהרגע הראשון, והנחתה אותנו בכל שלבי הפיתוח והעיצוב. עבורנו, אם המשתמש לא נהנה מהשימוש באפליקציה זה סימן שלא עמדנו בסטנדרטים שהצבנו לעצמנו. רצון זה שלנו בא ליידי ביטוי מעיצוב המסכים שהחלטנו לעשות בצורה מינימליסטית שתחסוך מן המשתמש עומס קוגניטיבי לא נחוץ, פשטות השימוש כמו שבמסך החיפוש יש רק תיבת טקסט וכפתור אחד, ועד הרספונסיביות של עדכון מיידי של טבלת הלינקים במסך המנהל לאחר כל שינוי.

את שביעות רצון הלקוח בחרנו למדוד באותו אופן בו מדדנו את שיעור השגיאות. ע"י דירוג המשתמש.

יעדינו הוא שהממוצע יהיה 8 ומעלה.

**2. ארכיטקטורת המערכת:**

2.1. תרשים ומאפיינים מרכזיים (מעודכן):

תרשים:

תמונה שמכילה טקסט, צילום מסך, תרשים, קו

התיאור נוצר באופן אוטומטי

מאפיינים:

UI: ממשק מבוסס כרטיסיות המאפשר מעבר בין חלקי המערכת.

Web Scraper: מטפל בסריקת האתר ועיבוד התוכן של כל דף.

Management Methods: אחראיות על טיפול בפעולות המשתמש על המילים באינדקס, כגון הוספה, מחיקה ועריכת כמות הופעות בלינק של מילה.

ChatBot Methods: אחראיות על טיפול בשאילתות שנשלחות אל הצ'ט בוט, קבלת תשובות מGemini והחזרתן למשתמש

Statistics Methods: מטפל ביצירת סטטיסטיקות על בסיס נתוני השימוש במערכת, כגון המילים והשאילתות הנפוצות ביותר.

Search Methods: אחראיות על פעולות החיפוש במערכת, כולל טיפול בשאילתות, איתור תוצאות וחישוב נתונים רלוונטים.

Index Service: אחראי על בניית האינדקס על ידי מעבר וביצוע פעולות על התוכן המעובד שמתקבל מה Web Scraper.

Robot Checker Service: מטפל בבדיקת לינקים כדי לוודא שאין עליהם איסור תחת robots.txt של האתר ושהם שייכים לאתר של אז'ור ולא אחד אחר.

Firebase Service: אחראי על תקשורת עם מסד הנתונים וביצוע פעולות של העלאה והורדה ממנו.

2.2. תרשים Use case מעודכן של פונקציונליות המערכת:

תמונה שמכילה טקסט, צילום מסך, תרשים, קו

התיאור נוצר באופן אוטומטי

2.3. דרישות פונקצינליות:

א. המערכת תאפשר חיפוש.

ב. המערכת תאפשר קיום שיחה עם צ'אט בוט.

ג. המערכת תאפשר צפייה בגרפים של סטטיסטיקות שונות.

ד. המערכת תאפשר ביצוע שינויים באינדקס ההפוך (הוספה, מחיקה ועריכה).

ה. המערכת תאפשר לשלוח משוב למפתחי האפליקציה.

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

|  |  |  |
| --- | --- | --- |
| # | דרישה | קטגוריה |
| 1. | המערכת תספק למשתמשים שקיפות מלאה בנוגע לפעולת האלגוריתמים ולנתונים הנאספים | Auditability |
| 2. | המערכת חייבת להיות תואמת לדפדפנים מודרניים (Chrome, Firefox, Safari, Edge) בשתי הגרסאות האחרונות שלהם | Compatibility |
| 3. | המערכת תכלול תיעוד קוד מלא, מפורט וברור, כדי לתמוך בצוותי הפיתוח והתחזוקה | Maintainability |
| 4. | מנגנון החיפוש יספק למשתמש משוב מיידי, כגון הצגת אינדיקטור עיבוד (לדוגמה, אנימציית טעינה), כדי להראות שהמערכת מבצעת את החיפוש. | Usability |
| 5. | במקרה של שגיאה או אי יכולת לענות, המערכת תספק הודעת שגיאה ברורה | Usability |

**4. ביקורת עמיתים אשר ניתנה במהלך הצגת הפרויקט (שבוע 9), וכיצד התמודדתם איתה.**

ביקורת:

|  |  |  |
| --- | --- | --- |
| הערת משוב | האם התבצע שינוי באפליקציה בעקבות ההערה? | נימוק |
| למסך מנהל להוסיף אופציה להוסיף מילה לאינדקס | כן, הוספנו את האופציה למנהל להזין מילה, להוסיף את הurl שהיא נמצאת בו באתר, ואת מספר המופעים של המילה בurl. | אכן הסכמנו שזה חיוני שתהיה האופציה, תיתן יותר גמישות במערכת ולכן הוספנו אותה. |
| להוסיף צבע לרקע | לא | החלטנו על עיצוב מינימליסטי ופשוט בהשראת מנועי חיפוש גדולים כמו גוגל לדוגמא. |
| אין צורך להציג למשתמש ranking | לא | אנחנו ממיינים את הurls שמוצגים למשתמש לפי הranking שלהם (הכי גובה מופיע ראשון וכן הלאה). אנו רוצים לוודא שכל המשתמשים יזהו בקלות באילו דפים יש יותר סיכוי למצוא את מבוקשם ולכן לדעתנו ציון ה rankחשוב. |
| להוסיף לוגין למנהל | לא | לוגין למנהל אינו חלק מהדרישות ולכן העדפנו להשקיע בחלקים העיקריים כמו לוודא שהפרויקט עובד וכמו שצריך. |
| הוספת מסך בית | לא | מבחינתנו מסך הבית הוא מסך החיפוש, כאשר המשתמש מפעיל לראשונה את המערכת זה המסך הראשון שמוצג לו (בדומה למנועי חיפוש כמו גוגל). |
| אומרים שיש stop words שמופיעים באינדקס. נתנו לנו דוגמא : your"". צריך לדאוג לstop words יותר טובים. | כן | זיהינו את השגיאה והרחבנו את מערך הstop words בהתאם. |
| לשפר בוט | כן | שיפרנו את הצ'ט בוט בעזרת שימוש בAPI של GEMINI. |

התייחסות לציון הSUS:

A table with numbers and letters

Description automatically generated \*הנתונים המוצגים הם לאחר החסרת אחד מתגובת המשתמש בשאלות האי זוגיות והפחתת תגובת המשתמש מ5 בשאלות הזוגיות כפי שנלמד בהרצאה 8 של הקורס ממשק אדם מחשב (הערכת ממשקים)

\*\* הפעולות שהתבצעו הן סכימה והכפלה ב2.5 עבור כל שורה ולבסוף ממוצע של העמודה האחרונה (הציון הסופי בצד ימין למטה, הציון מכל משתמש בצד ימין בשורה המתאימה לו: Q# מסמן שאלה, U# מסמן משתמש)

מכיוון שתוצאה מעל 68 נחשבת מעל הממוצע, אנו יודעים כי ציון הSUS שלנו שעומד על 87.5 גבוה בהרבה מהממוצע ולכן נוכל להסיק כי המערכת שלנו שימושית ויעילה במיוחד.

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

יצירת אינדקס:

|  |  |
| --- | --- |
| Explanation | Class Name |
| Responsible for parsing a given site’s robots.txt file and checking if given links are in the site’s domain and if they’re allowed to scrape from. (Uses the library robotparser from urllib) | RobotsChecker |
| Responsible for fetching Microsoft Azure’s pages content, stemming it and cleaning it of meaningless words like stopwords and numbers. | AzureSearchEngine |
| In charge of creating the inverted index, iterates through the pages content, for each word saves all the links it appears in and the amount of times it appears in them. | IndexService |
| Responsible for communication with Firebase database, uploading and downloading data. | DatabaseService |

|  |  |
| --- | --- |
| Explanation | Method Name |
| Fetches the robots txt and parses it | load\_robots\_txt |
| Checks if a given link is allowed to be scraped | Is\_allowed |
| Checks if a given link is in microsoft azure’s domain | is\_in\_site |
| Stems a given word using PorterStemmer | stem\_word |
| Transverses all the links in the site, scraping processing and saving the content for future use. | fetch\_azure\_pages |
| Receives a string and clears it from all substrings that are not alphabet letters, turns it to lowercase and removes stopwords. | clean\_text |
| Recreates the index by fetching the pages and adding all the documents to the index. | restore\_index |
| Recieves a document’s data, creates a unique id for it, runs through all of it’s words, checking the count, updating it’s frequency and total count. | Add\_document |
| Uploads a given user ratings dictionary | Upload\_rating |
| Sorts the index by sorting each word’s url list. | Sort\_index |
| Uploads all urls and their total amount of words as separate entries in the database. | Upload\_doc\_word\_count |
| Uploads the index in a loop, so that each word is an entry | Upload\_index |
| Gets and returns the index from the database | Download\_index |

**דף מנהל:**

|  |  |
| --- | --- |
| Explanation | Function Name |
| Receives a message and it’s type (both as strings) and displays it with the right color accordingly on the message output. | display\_message |
| Renders and re-renders the page’s html code, rebuilding the table and displaying all buttons. | refresh\_html |
| Removes a given link locally and, refreshes the page and then saves the changes to firebase.  In case there are no longer links in the selected word, removes the word completely. | remove\_link |
| Receives a term, link and the terms appearance count in that link and activates all buttons related to editing the frequency, basically allowing the user to edit the term’s frequency in that link | edit\_link |
| Removes a word entirely from the local index and from firebase, then refreshes the page. | Remove\_word |
| Updates the term’s frequency in the edited link, updating the total count aswell, refreshing the page and saving to firebase. | on\_save\_edit\_btn\_click |
| Uploads the selected term and it’s doc list to firebase, updating the existing entry or uploading it as new. | save\_to\_firebase |
| Activated when the user selects a word in the combo box, refreshes the page with the newly selected word’s data | on\_word\_select |
| Adds a word to the local index with the values given in the add word text boxes, handles all types of situations such as the case in which the word is already in the index, only updating it’s url list and count, saves the new word to firebase, adds it to the dropbox and refreshes the page with the new word’s data. | Add\_word |

**דף מידע ומשוב משתמש:**

|  |  |
| --- | --- |
| Function Name | Explanation |
| on\_submit\_clicked | Activated when the submit button is clicked, uploads the selected values for user reviews to the database using the database service. |

**דף חיפוש:**

|  |  |
| --- | --- |
| **Method/Component** | **Explanation** |
| ***global\_url\_count\_map*** | A global dictionary that stores the total number of words for each URL. It is populated by **URL\_Count\_mapping()** and used to calculate relevance scores for search results. |
| ***URL\_Count\_mapping()*** | Fetches the total word count for each URL from the Firebase database under the /Links path. It saves the overall amount of words from each web page into global\_url\_count\_map for ranking search results. |
| ***add\_single\_word(word)*** | Tracks how often users search for a specific word by updating its count in the Firebase database under the /Words path. If the word already exists, it increments the count; if not, it creates a new entry with a count of 1 (indicating the first search). This data is later used for search statistics and analytics. |
| ***analyze\_query(text)*** | Splits the user's search query into individual words and processes each word using add\_single\_word(). This ensures that every word in the query is counted in the Firebase database, tracking how often users search for specific terms. |
| ***update\_query\_count\_in\_db(query)*** | Tracks how often a specific query is searched by updating its count in the Firebase database under the /Queries path. If the query is new, it initializes the count to 1; if it already exists, it increments the count. This data is used for search statistics and analytics. |
| ***stem\_query\_words(query)*** | Processes a query by reducing each word to its root form (stemming) using the PorterStemmer. This ensures that variations of the same word (e.g., "running" and "run") are treated as identical for searching. |
| ***search(stemmed\_query\_words, index)*** | Checks if each stemmed word from the query exists in the search index. If a word is found in the index, it retrieves all associated data (e.g., links, word counts within those links) and adds it to the results. This method returns a dictionary containing the search results for each word. |
| ***get\_intersection\_of\_urls(query\_data)*** | Finds URLs that are common to **all words** in the query. It iterates through the search results for each word, retrieves the associated URLs from the DocIDs field, and performs a **set intersection** to identify URLs that appear in every word's result. If no common URLs exist, it returns an empty set. |
| ***run\_search\_page()*** | Renders the search interface, including the search input, button, and results container. It also attaches event listeners to handle user interactions, such as button clicks and "Enter" key presses. |
| ***process\_search(query)*** | Orchestrates the entire search process. It takes a user query, stems the words, searches the index, finds common URLs, calculates relevance scores, and displays the results in a table. It also updates query and word statistics in Firebase for analytics. |

**דף צ'ט בוט:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Method/Function** |  |  | | --- | |  | | | **Explanation** |  | | --- | --- |  |  | | --- | |  | |
| |  |  |  | | --- | --- | --- | | run\_ChatBot() |  |  | | Sets up and displays the chatbot interface. Integrates a Gemini Generative Model to handle user input and generate bot responses dynamically. |
| |  | | --- | | handle\_message(message) | | Processes user messages. Displays the user input and generates a response  using the Gemini model. Updates the chatbot interface with the bot's response. |
| |  |  | | --- | --- | | on\_send\_clicked(b) |  | | Callback function for the "Send" button. Invokes handle\_message with the user's input and clears the input field after sending. |
| |  |  | | --- | --- | | text\_input.on\_submit() |  | | Sets up an event listener for the Enter key to trigger the same functionality as the "Send" button. |

דף סטטיסטיקות:

|  |  |
| --- | --- |
| **Method/Component** | **Explanation** |
| **get\_count(item)** | Extracts the 'count' value from a dictionary-like item. Used as a key function for sorting data by count. |
| **real\_time\_data()** | Fetches and processes data from Firebase. Prepares top-10 lists for words, queries, and common words in Azure index. |
| **run\_pie\_chart()** | Displays a bar chart of the selected dataset. |
| **run\_seaborn\_histogram()** | Displays a histogram showing the distribution of counts in the selected dataset. |
| **run\_statistics()** | Main function to handle interactive buttons (Plot, Pie Chart, Histogram, Refresh) and orchestrate the visualization display. |
| **Dropdown** | Provides an interactive way for users to select between data types (Words, Queries, Index). |
| **Buttons** | - Plot: Displays a bar chart.  - Pie Chart: Displays a pie chart.  - Histogram: Displays a histogram.  - Refresh: Updates the current graph with refreshed data. |

**6. תיק למשתמש:**

הסבר כללי על המערכת:

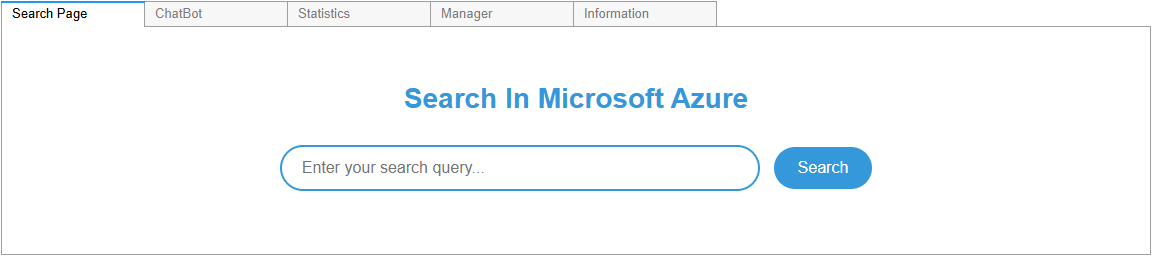
המערכת שלנו היא מנוע חיפוש שמיועד במיוחד לאתר Microsoft Azure. בדומה למנועי חיפוש מוכרים כמו גוגל, היא מאפשרת למשתמשים לחפש מידע שהם צריכים, אבל במקום לחפש בכל האינטרנט, היא מתמקדת רק בתוכן שנמצא באתר הזה. המטרה היא לעזור למשתמשים למצוא מידע במהירות ובקלות, על ידי הכוונה למאמרים ולדפים הכי רלוונטיים באתר לפי מילות החיפוש שלהם.

האפליקציה מבוססת על מאגר מידע שאספנו מכל האתר של Microsoft Azure ומשתמשת באלגוריתם חכם שיעזור למקד את תוצאות החיפוש בצורה מדויקת. בנוסף, שילבנו צ'אט עם בינה מלאכותית (LLM) שמתכתב בשפה פשוטה וברורה.

בנוסף, יש גם דף סטטיסטיקה שמציג את מילות החיפוש הכי פופולריות והכי שימושיות, כדי לעזור למשתמשים להבין מה עוד אפשר לחפש ולגלות מידע חדש בקלות.

פירוט מסכים

1. מסך חיפוש מידע – במסך החיפוש נוכל נוכל להזין מילות חיפוש ולקבל תוצאות ממוקדות מהאתר של מייקרוסופט אז'ור.



1. מסך צ'אט בינה מלאכותית – במסך הצ'ט בוט נוכל בעצם לקיים שיחה עם צ'ט בוט חכם המספק הסברים המלצות והכוונה.

A screenshot of a computer

Description automatically generated

1. מסך סטטיסטיקות – במסך הסטטיסטיקות נוכל בעצם לראות סטטיסטיקות מעניינות על האתר והמערכת, במסך זה נוכל לבחור סוג של נתונים להצגה, לצפות בחיפושים הפופולריים והרלוונטיים ביותר כמעט בזמן אמת.

A screenshot of a graph

Description automatically generated

1. מסך ניהול המערכת – מסך זה הוא מסך ניהול האינדקס ההפוך שלנו, המסך מאפשר הוספת מונח חדש, מחיקת מונח, מחיקת קישור מהרשימה של המונח, עריכת כמות מופעים של מונח בקישור, הוספת קישור למונח עם כמות הופעות מוגדרת.

A screenshot of a computer

Description automatically generated

1. מסך אינפורמציה על האפליקציה – זה למעשה מסך השקיפות האלגוריתמית שלנו, המסך מכיל מידע על איך אלגוריתם הדירוג שלנו עובד וגילוי נאות על כך שאנחנו שומרים את החיפושים כדי להציגם בסטטיסטיקות. בנוסף ניתן לדרג את חווית השימוש באפליקציה.

A screenshot of a computer

Description automatically generated

הסבר על מעבר בין מסכים:

המעבר בין המסכים מתבצע בעזרת הלשוניות שלמעלה, רק צריך לבחור בלשונית של הדף המתאים וכך נגיע אליו:



טעויות אפשריות:

במסכי חיפוש וצ'טבוט:

חיפוש ארוך ומורכב: הכנסת ביטוי חיפוש ארוך או מסורבל יכולה לבלבל את המנוע ולהחזיר תוצאות שאינן מדויקות.

שימוש במונחים לא רלוונטיים: הכנסת ביטוי שפחות רלוונטי לאתר טכנולוגי כמו "גלידת שוקולד" לא יציג תוצאות.

שימוש בסימני פיסוק: הכנסת ביטוי עם סימני פיסוק תבלבל את המנוע ולא תציג תוצאות.

שימוש בשפה שגויה: המערכת עובדת בשפה האנגלית, כל ניסיון להזין חיפוש שאינו באנגלית לא יציג תוצאות.

לחיצה על כפתור שליחה/חיפוש ללא הזנת טקסט: במקרה זה המערכת תדאג להציג הודעה מתאימה.

במסך מנהל:

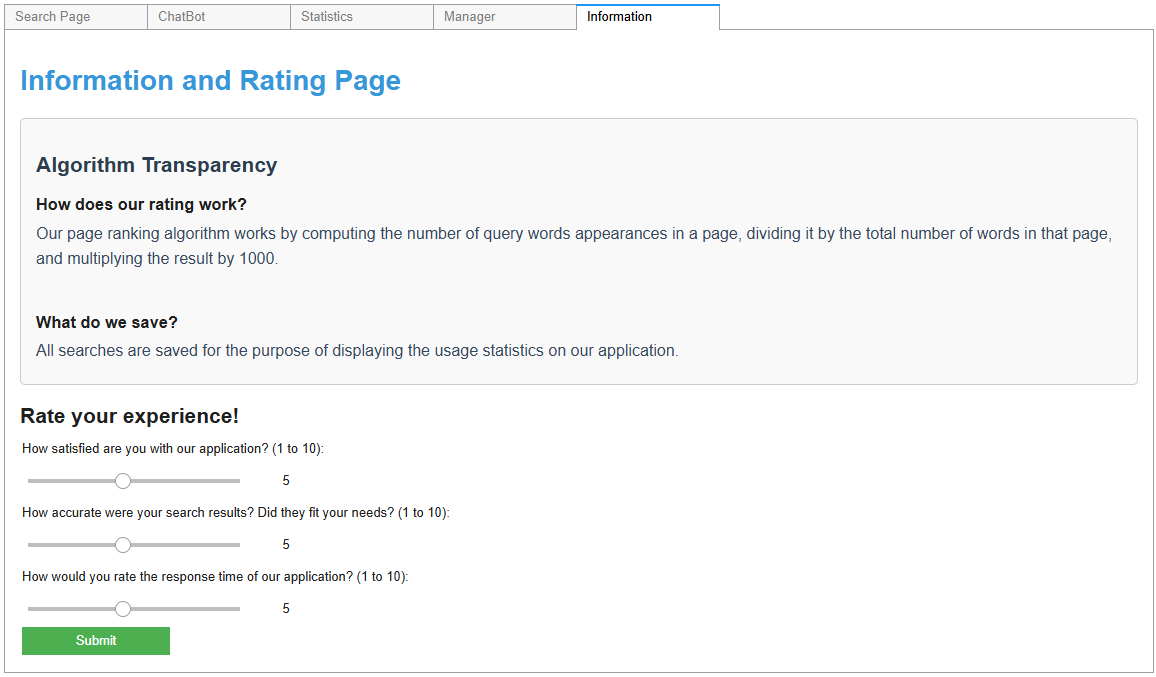
השארת שדות ריקים שנחוצים לפעולות: במידה ונרצה להוסיף מילה ולא נמלא את השדות הרלוונטים, המערכת תציג הודעה מתאימה.

הזנת קלט לא חוקי: במידה ונזין קלט לא חוקי כמו כמות הופעות קטנה או שווה ל0 או לינק מפורמט לא תקין, המערכת תודיע על כך עם הודעה מתאימה.

7. שקיפות אלגוריתמית:

בלשונית המידע, ניתן לראות את אופן פעולת האלגוריתם, ואת אופן דירוג הדפים.

המידע שנאסף מהמשתמש הוא אנונימי, השאילתות נשמרות במערכת, ונאגרות לצורך יצירה ותחזוק הסטטיסיקה, שירות שחשוב לנו לספק למשתמשים.



8. אתגרים שעלו במהלך העבודה, וכיצד התמודדתם איתם.

האתגר הראשון שהתמודדנו איתו היו במהלך בניית האינדקס, הוא שבאתר MS Azure היו הרבה מאוד מילים והאינדקס היה ממש גדול, לא יכולנו להעלות אותו בבת אחת למסד, לכן תחילה פירקנו אותו למספר חתיכות וניסנו לצמצם את המספר החתיכות כמה שאפשר כדי לחסוך בגישות למסד, לאחר מכן בצד החיפוש הצוות העדיף להשתמש באופן שונה במסד, ולשמור כל מילה כרשומה, לכן החלטנו שלעבור על כל מילה באינדקס ולהעלות אותה עם רשימת הלינקים הממוינת שלה לפי מספר הופעות.

אתגר נוסף שהתמודדנו איתו היה שבאינדקס היה חוסר דיוק בין כמות ההופעות לכמות הפעמים שניתן למצוא את המילה בדף, איתרנו את מקור הבעיה בכך שבעצם היו כפילויות של לינקים כאשר אחד הסתיים עם סיומת "/" והאחר ללא, ומבחינת הקוד היו אלו אותם לינקים שגרמו לעדכון של כמות ההופעות לכל לינק פעמיים, כלומר אם מילה מסוימת הופיעה בדף מסוים 200 פעמים, קיבלנו כי המילה הופיעה 400 פעם בכל אחד מן הדפים (עם "/" ובלי). התמודדנו עם כך שאיתרנו את החלק בקוד בו יצרנו את האינדקס ודאגנו לוודא שלא יהיו הכפילויות האלו.

אתגר שלישי היה שתחילה האינדקס היה מלא בהרבה מאוד מילים חסרות כל ערך עם סימנים מרובים ומספרים, אותן סיננו בעזרת ביטוי רגולרי שייתן לנו מילים רק שהן א"ב ולא סימנים או מספרים שלא מועילים.

9. טבלת משוב:

|  |  |  |
| --- | --- | --- |
| הערת משוב | האם לדעתכם יש צורך בשינוי במערכת בעקבות ההערה? | נימוק |
| בתוצאות חיפוש היה יכול להיות מועיל/להיראות טוב יותר בעיניי אם היה גם שם האתר בנוסף ללינק (כמו בגוגל) | לא | לדעתינו אין צורך בזה, כל הלינקים שאנחנו מציגים בחיפוש הם מהאתר של מיקרוסופט אז'ור, לכן בנוסף ללינק נציג בעצם את אותו אתר כל הזמן. |
| רקע עם צבעים | לא | החלטנו שלהמשיך עם העיצוב המינימלסיטי בהשראת מנוע החיפוש גוגל, שגם לו אין רקע צבעוני. |
| קשה לראות את הגרף ארכיטקטורה | לא | לא ראינו קשר בין ההערה למערכת. |
| להוסיף מנהל ולא רק משתמש | לא | כפי שתיארנו עוד בביקורת העמיתים, לא היה זה חלק מהדרישות ובחרנו להתמקד בדברים המרכזיים של הפרויקט. |

Code Review:

|  |  |  |
| --- | --- | --- |
| **הערה** | **תגובה** | **האם בכוונת הצוות לבצע שינוי בעקבות ההערה? יש לפרט. אם כן, מה השינוי. אם לא, מדוע** |
| פשטות:  ניתן להוסיף תיעוד | מסכים, נתקן | כן, בכוונתינו להוסיף עוד הערות ותיעוד בכל החלקים בקוד. |
| מימוש:  כן, הפונקציונליות תקינה והמערכת עקבית. | תודה, עבדנו על זה קשה. | לא, דאגנו כבר מראש לוודא שהפונקציונליות תקינה בעזרת הבדיקות המתאימות לכך וגם שהמערכת עקבית. |
| מודלריות :כן הקוד מחולק לקבצים. | מסכים | לא, הקוד כבר מחולק למיקרו שירותים כפי שתכננו מראש. |
| יעילות :לא מצאנו חריגות בביצועי המערכת. | מסכים | לא, תכננו מראש לחסוך בפעולות מיותרות בזמן הריצה כדי לוודא יעילות. |
| באגים :לא היו מקרים כאלו. | שמחים לשמוע | לא, ביצענו בדיקות רבות בתרגיל בית 3 כדי לוודא שלא יהיו תקלות. |
| טיפול בשגיאות :יש התראות על שגיאות של משתמש | שמחים לשמוע שהמערכת עובדת כראוי | לא, בעזרת בדיקות הקבלה כבר דאגנו לחשוב על כל המצבים שלדעתנו מצריכים הודעה מתאימה ולתפוס את כל החריגות שעלולות להקריס את המערכת. |

10. מקורות:

מצגת תירגול 7 – מיקרושירותים

Microsoft Azure - <https://azure.microsoft.com/en-us/>

Ipywidgets - <https://ipywidgets.readthedocs.io/en/stable/>

ChatGPT prompts:

“How can I create a loading animation with HTML for a google colab project?”

“Can you explain to me how display() function works and how it relates to Output() function? I have a problem where certain HTML elements don’t appear on my screen”