מבוא למחשוב ענן  
תרגיל בית 3

****

**Repository:** [לחצו כאן](https://github.com/tomerot/cloud-computing-kakadoo.git)

**Notebook: [לחצו כאן](https://colab.research.google.com/drive/1r2BvHRp2aEzVaO2E_hffEYuPjA-Mhqxn)**

סרטון**:** [**לחצו כאן**](https://www.youtube.com/watch?v=C0nFFuhSl3c) **(מצורף גם קובץ** mp4**)**

|  |  |
| --- | --- |
| **Team Kakadoo** | |
| **שם** | **ת.ז** |
| עופר אלזרע | 315824409 |
| תומר רוטמן | 316317023 |
| אורן עפרוני | 208741934 |
| גד עזריאל | 209146927 |
| אלכס גרומן | 320648918 |
| שלומי זריהן | 206877698 |

1 – הגדרת תפקידים, הקצאת משימות, בדיקות קבלה

|  |  |  |
| --- | --- | --- |
| שם חבר הצוות | משימות שהוקצו | הושלמו |
| **עופר אלזרע (Scrum Master)** | הקצאת המשימות לאיטרציה | ✓ |
| קביעת פגישות שבועיות | ✓ |
| תיק מתכנת | ✓ |
| **אורן עפרוני (Frontend Developer)** | פיתוח צ'אט-בוט מבוסס Gemini (יחד עם גד) | ✓ |
| **אלכס גרומן (QA)** | כתיבת בדיקות קבלה וביצוען | ✓ |
| **תומר רוטמן (UI)** | יצירת סרטון | ✓ |
| תיק משתמש | ✓ |
| **גד עזריאל (מהנדס המערכת  ו- Backend Developer)** | פיתוח צ'אט-בוט מבוסס Gemini (יחד עם אורן) | ✓ |
| **שלומי זריהן (Product Manager)** | כתיבת הסברים על ה- KPI ושקיפות אלגוריתמית | ✓ |

בדיקות קבלה

\*לכלל בדיקות הקבלה אנו יוצאים מכלל הנחה כי המערכת הופעלה בהצלחה

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **תיאור** | **תנאים מקדימים** | **תוצאה רצויה** | **הערות** |
| **1** | * לחיצה על כפתור Admin Page * הזנת סיסמא: 123456 * לחיצה על Enter | משתמש נמצא בעמוד הבית | משתמש הועבר בהצלחה לעמוד מנהל |  |
| **2** | * פתיחת Dropdown של המושגים * בחירת מושג * לחיצה על Display Term Information * לחיצה על Select All * לחיצה על Delete Selected URLs | * משתמש התחבר בתור מנהל * מושגי האינדקס נשלפו בהצלחה ל- Dropdown * התחברות ל- DB בוצעה בהצלחה | * הטבלה מוסתרת * המושג נמחק  מה- DB * ה- Dropdown מתעדכן בהתאם |  |
| **3** | * פתיחת Dropdown של המושגים * בחירת מושג * לחיצה על Display Term Information * סימון URL ספציפי * לחיצה על Delete Selected URLs | * משתמש התחבר בתור מנהל * מושגי האינדקס נשלפו בהצלחה ל- Dropdown * התחברות ל- DB בוצעה בהצלחה | * הטבלה מתעדכנת * ה- URL תחת המושג נמחק מה- DB |  |
| **4** | * פתיחת Dropdown של המושגים * בחירת מושג * לחיצה על Display Term Information * לחיצה על Delete Current Term | * משתמש התחבר בתור מנהל * מושגי האינדקס נשלפו בהצלחה ל- Dropdown * התחברות ל- DB בוצעה בהצלחה | * הטבלה מוסתרת * המושג נמחק  מה- DB * ה- Dropdown מתעדכן בהתאם |  |
| **5** | לחיצה על Recreate Index | * משתמש התחבר בתור מנהל * התחברות ל- DB בוצעה בהצלחה | * המושגים ב- DB מתעדכנים בהתאם * ה- Dropdown מתעדכן בהתאם |  |
| **6** | * לחיצה על כפתור Statistics * לחיצה על Tab “Most Common Words” * לחיצה על Tab “Least Common Words” * לחיצה על Tab “Most Common Documents” | * משתמש נמצא בעמוד הבית * התחברות ל- DB בוצעה בהצלחה | * כניסה לדף Statistics בוצע בהצלחה * Tab “Random Words” מוצג עם 5 מונחים אקראיים * מעבר בין  TAB-ים מבוצע בהצלחה ומוצג המידע הרלוונטי |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **תיאור** | **תנאים מקדימים** | **תוצאה רצויה** | **הערות** |
| **7** | * לחיצה על כפתור Statistics * לחיצה על כפתור Random Again | * משתמש נמצא בעמוד הבית * התחברות ל- DB בוצעה בהצלחה | * כניסה לדף Statistics בוצע בהצלחה * Tab “Random Words” מוצג עם 5 מונחים אקראיים * 5 מונחים אקראיים חדשים הוצגו |  |
| **8** | * לחיצה על כפתור AWS Chatbot * הזנת "What is aws?" * לחיצה על כפתור Ask | * משתמש נמצא בעמוד הבית * חיבור מוצלח ל- API של Google Generative AI | * מוצגת חלונית של תשובה * הצ'אטבוט יחזיר תשובה קצרה ורלוונטית לשאלה |  |
| **9** | * לחיצה על כפתור AWS Chatbot * הזנת "Tell me about Azure" * לחיצה על כפתור Ask | * משתמש נמצא בעמוד הבית * חיבור מוצלח ל- API של Google Generative AI | * מוצגת חלונית של תשובה * הצ'אטבוט יחזיר " This chatbot only answers questions about AWS and its services. Please ask an AWS-related question" | כל שאלה שאינה רלוונטית ל- AWS תקבל תשובה קבועה זו |
| **10** | * לחיצה על כפתור AWS Chatbot * לחיצה על כפתור Ask | * משתמש נמצא בעמוד הבית * חיבור מוצלח ל- API של Google Generative AI | * מוצגת הודעה “Please enter a question" |  |
| **11** | * לחיצה על כפתור Run Query * לחיצה על כפתור AWS Fun Fact * לחיצה נוספת על כפתור AWS Fun Fact | משתמש נמצא בעמוד הבית | * מוצג Fun Fact קצר על AWS * בלחיצה נוספת מוצג Fun Fact אחר |  |
| **12** | * לחיצה על כפתור Run Query * לחיצה על כפתור Search | * משתמש נמצא בעמוד הבית * התחברות ל- DB בוצעה בהצלחה | * מוצגת הודעה “Please enter a query first!” |  |
| **13** | * לחיצה על כפתור Run Query * הזנת "Auto Scaling" * לחיצה על כפתור Search * לחיצה על קישור "AWS Auto Scaling Documentation" | * משתמש נמצא בעמוד הבית * התחברות ל- DB בוצעה בהצלחה | * מוצג Query Results עם התוצאות הרלוונטיות * הדפים מסודרים לפי הדירוג שלהם * בלחיצה על הקישור נפתח חלונית לדף הרלוונטי |  |
| **14** | * לחיצה על כפתור Run Query * הזנת "Kishkush Balabush" * לחיצה על כפתור Search | * משתמש נמצא בעמוד הבית * התחברות ל- DB בוצעה בהצלחה | * מוצג Query Results עם הודעה “No results found" | החיפוש שהוזן לא מכיל מושגים מהאינדקס ולכן אין תוצאות רלוונטיות |

2 – בניית המערכת

1. **שימוש ב- Microservices**

במערכת שלנו נעשה שימוש נרחב במספר Microservices, והיתרונות המרכזיים בשימוש בהם כוללים מודולריות גבוהה של הקוד, מה שמוביל למבנה קוד יעיל יותר והפחתת קוד חוזר. פירוק הקוד למיקרו-שירותים מקל על הבנת אופן הפעולה של כל שירות בנפרד, ובכך גם תורם להבנת המערכת כולה. בנוסף, השימוש במיקרו-שירותים מפחית את התלות בין מקטעי קוד שונים ובין צוותים שעובדים על חלקים שונים של המערכת, מה שמאפשר זמן פיתוח מהיר יותר. יתרה מכך, כאשר הקוד מחולק למיקרו-שירותים נפרדים, קל יותר לזהות ולתקן טעויות ובאגים באופן יעיל.

מיקרו-שירותים בהם נעשה שימוש במערכת:

1. **Crawler Microservice**  
   **תיאור:** זחלן שמחלץ דפי HTML סטטיים מ- Domain מסוים אשר מועבר אליו.   
   בפרויקט מועבר אליו ה- Domain של AWS והוא מוגדר לחלץ כ- 200 דפים.  
   **יתרון שימוש:** פועל כיחידה עצמאית, ניתן לעדכן או להתאים אותו לצרכים ספציפיים. במטלה שלנו, התאמנו אותו לאתר AWS, אך ניתן להפעילו גם על אתרים אחרים לצורך איסוף נתונים ממקורות שונים או ניתוח בפורמטים מגוונים, מבלי להשפיע על יתר המערכת.
2. **Index Creator Microservice**

**תיאור:** מיקרו-שירות שאחראי על יצירת האינדקס. מאגד מידע טקסטואלי מתוך דפי HTML, ומבצע עליו עיבוד להפקת האינדקס. בפרויקט מוגדרים stop words מותאמים ל- AWS. **יתרון שימוש:** היכולת להתמקד במשימה ייעודית של עיבוד וארגון מידע טקסטואלי בצורה מבודדת. הגישה הזו מאפשרת התאמה קלה לצרכים משתנים, כמו שימוש ברשימות stop words מותאמות, ומבטיחה גמישות ושדרוגים מהירים מבלי להשפיע על יתר חלקי המערכת.

1. **Data Fetcher Microservice**

**תיאור**: מיקרו-שירות לשליפת מידע מה- Database.  
**יתרון שימוש**: גמישות בשינויים במבנה הנתונים או טכנולוגיית ה-Database, שיפור ביצועי שליפה במיקרו-שירות, ושימוש חוזר באותו שירות עבור רכיבי מערכת שונים, תוך שמירה על אחידות וניהול יעיל של גישה לנתונים.

1. **Administration Microservice**

**תיאור**: מיקרו-שירות לניהול האינדקס.  
**יתרון שימוש**: הגישה המוגבלת אליו מאפשרת רק למורשים לבצע שינויים בנתונים, ובכך משפרת את אבטחת המידע.

1. **Query Microservice**

**תיאור**: מיקרו-שירות לעיבוד שאילתה ושליפת התוצאות הרלוונטיות מהאינדקס.  
**יתרון שימוש**: מאפשר שליפת תוצאות מהאינדקס במהירות ובצורה מדויקת, תוך הפרדת תהליך עיבוד השאילתה משאר חלקי המערכת לשיפור ביצועים ותחזוקה.

1. **Statistics Microservice**

**תיאור**: מיקרו-שירות לביצוע מניפולציות להפקת סטטיסטיקות וכן ליצירת גרפים בצורה גנרית.  
**יתרון שימוש**: בזכות המבנה הגנרי שלו, מאפשר Reuse לגרפים שונים, מה שמאפשר יצירה מהירה של גרפים חדשים ללא צורך בהוספה של כמות קוד משמעותית.

1. **Key Performance Indicators רלוונטיים לפרויקט:**

**Performance** - לאחר בנייה ראשונית של מנוע החיפוש, זמן עיבוד השאילתא ארך זמן רב (4-5 שניות), ביצענו אופטימיזציה באופן שמירת מידע ב- Database, כעת זמן עיבוד השאילתא אורך כ- 1-1.5 שניות.   
זמן תגובה וקצב עיבוד של מנוע החיפוש הם מדדים קריטיים לשמירה על ביצועי מערכת יעילים.  
  
**User Experience** - בפרויקט שלנו המערכת אינה פתוחה לקהל הרחב, לכן בחרנו להישען על שאלוני SUS וביקורת עמיתים לקבלת דירוג חווית המשתמש. דירוג זה השפיע על עיצוב המערכת והשפיע על יעילותה ונוחות השימוש בה.  
  
**Security & Compliance** - בפרויקט שלנו ניתן למדוד את איכות האבטחה ע"י כלים כמו NB Defense, כלי אשר בוחן היבטים מהותיים באבטחה של הפרויקט, ומסוגל לנתח קבצי Jupyter, לאחר ניתוח הפרויקט הוא נותן מידע עבור פרצות אבטחה, תלויות בעייתיות וכדומה. בנוסף הוא נותן ציון אבטחה כללי.

1. **שקיפות אלגוריתמית**:

בפרויקט שלנו, תהליך דירוג התוצאות מתבצע מאחורי הקלעים, ואיננו חושפים את כל שלבי החישוב למשתמש. עם זאת, מוצגות בפני המשתמש הדרגות (Ranks) של התוצאות, כך שיוכל להבין את מהות סדר הצגתן ולזהות את רלוונטיות התוצאות ביחס לשאילתא שלו.

1. **אתגרים שעלו במהלך העבודה, וכיצד התמודדנו איתם**:

* ביצירת הזחלן רצינו להרחיב את מרחב החיפוש שלו, כלומר אל Domain-ים נוספים שהם תחת אתר AWS כמו AWS Documentation ו- re:Post, מכיוון שהם מכילים מידע רב ערך. עם זאת, גילינו שהדפים ב- Domain-ים אלו נוצרים דינאמית, וספריית BeautifulSoup מוגבלת ליצירת soups של דפי HTML סטטיים. לצורך פתרון הבעיה, ניסינו להשתמש בספריות שלא נלמדו בתרגולים, אך ללא הצלחה, שהביאה לויתור על היכולת.
* ביצירת האינדקס לראשונה קיבלנו הרבה מאוד מושגים שמהווים stop words והביאו ל- DB יותר מדי גדול. מדובר בכמות גדולה מאוד של מילים ולכן הסרה שלהם באופן ידני בלתי אפשרית, אבל הכרחית. לכן, יצרנו קובץ JSON של האינדקס, ושלחנו ל- ChatGPT יחד עם קובץ של stop words שהיה בידינו, וביקשנו ממנו למצוא את כל ה- stop words האלה, כך שנוכל להוסיפן בקוד לרשימת ה- stop words שאנחנו מסירים ביצירת האינדקס.
* בכתיבת אלגוריתם החיפוש הראשוני, הדירוג של לינק היה סכום המופעים של מושגי השאילתה בלינק. בשאילתות מסויימות זה יצר מצב שהדפים שהוצגו לא תמיד סודרו בצורה מספיק טובה עבור המשתמש, ולכן כדי להתגבר על זה, נתנו חשיבות גבוהה בדירוג גם להופעת מושגים מהשאילתה ב- URL עצמו.
* הזרקות CSS אשר משפיעות אחת על השניה בדפים נפרדים, ובהתאם יוצרות בעיות בדפים  
  (לדוגמא גלישת נתונים מטבלה עם גלילה כאשר כמות נתונים גדולה מידי) - בדקנו את הזרקות ה- CSS הרלוונטיות וחיפשנו אילו מאפייני Layout עלולים ליצור 'חפיפה', איפה שהיה ניתן מחקנו מאפיינים, אחרת הוספנו הגדרות רחבות יותר שמנעו חפיפה.
* תחימת הקוד לפי מיקרו שירותים וחלקים מודולריים - לאחר בניית המבנה הכללי של הקוד, ביצענו חלוקה מסודרת של מיקרו-שירותים, חלקי CSS\HTML ועוד מקטעים מופרדים באופן לוגי. ראשית החלוקה יצרה בעיות מסוימות, אך כל בעיה סודרה והקוד עובד באופן חלק ומודולרי.
* בחירת Widget מתאימים ליצירת תוכן ברור נוח לשימוש ואדפטיבי לגודל מסך, לדוגמא במסך סטטיסטיקות רצינו מבנה כלשהו שנוכל להציג בו בצורה ברורה את 3 הגרפים שבחרנו, תוך מעבר פשוט ונוח בין הגרפים. לצורך כך בחרנו ב- Widget.Tab.
* בדיקת סיסמא עבור משתמש מסוג Admin, כאשר הסיסמא מובנת בקוד יצרה לנו מספר פעמים בעיות לא צפויות(למשל הכרזה כי סיסמא אינה נכונה על אף שהיא כן), לאחר בדיקה גילינו כי מהירות קליטת קלט  
  ב- Widget.Text לעיתים יכולה להיות איטית, כתוצאה מכך באותם פעמים בהם הזנו את הסיסמא במהירות ולחצנו על כניסה, הסיסמא שהתקבלה הייתה חתוכה עד לערך שהספיק להיקלט. לכן הוסף לקוד מנגנון השהייה מאוד קצר אך מספיק כדי שבעיה זו לא תוכל לקרות גם אם המשתמש מדביק את הקלט ומיידית לוחץ על כניסה.

3 – סגירת הפרויקט

**תיק מתכנת**

תיק מתכנת זה משמש כמדריך עבור מפתחים העובדים על מנוע החיפוש שלנו, אשר רץ בסביבת עבודה של  
Google Colab. מטרת המדריך היא להקל על המפתחים להבין את מבנה המערכת ולספק כלים לשימוש נכון ויעיל בקוד, תוך שמירה על סטנדרטים גבוהים של פיתוח ועקביות בפרויקט.   
המדריך מפרט את ארכיטקטורת המערכת, ספריות עיקריות בהן נעשה שימוש, את מבנה הקוד, את תבניות העיצוב בהן השתמשנו, את השימוש ב- Firebase ואת השימוש ב- ipywidgets לצורך יצירת ממשק משתמש.

1. **סקירת מערכת כללית**

ארכיטקטורת מערכת: היישום נכתב בשפת התכנות Python, נעשה שימוש במערכת ב- Firebase לאחסון ושליפת נתונים (NoSQL Database), רכיבי ממשק המשתמש מיובאים מספריית ipywidgets, בעזרתם יצרנו רכיבי ממשק משתמש אינטראקטיביים בתוך סביבת העבודה של Google Colab (מחברות Jupyter).

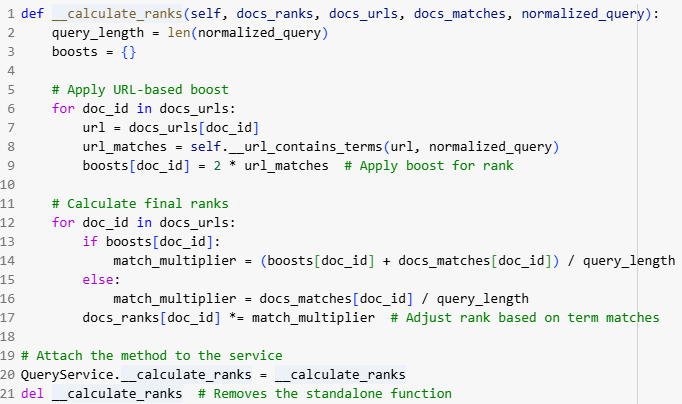
ספריות עיקריות במערכת:

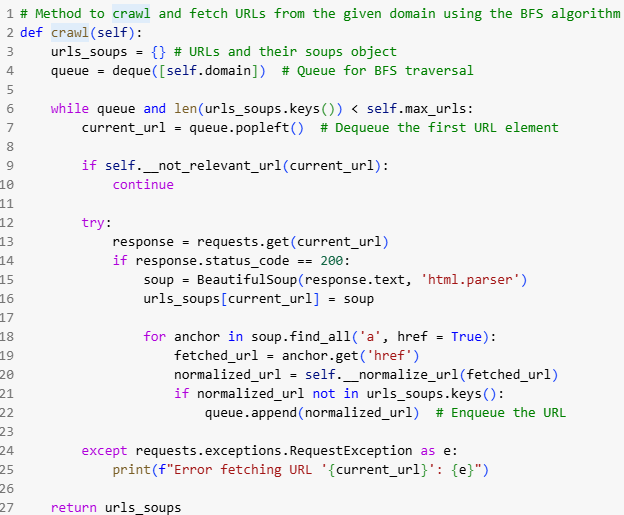
1. **ipywidgets** - רכיבי ממשק משתמש אינטראקטיביים במחברות Jupyter.
2. **IPython.display** - הצגת רכיבי HTML ותכנים גרפיים במחברות Jupyter.
3. **matplotlib** - יצירת גרפים ודיאגרמות להצגת נתונים בצורה ויזואלית בעמוד סטטיסטיקות.
4. **firebase** - ספרייה לתקשורת עם בסיס הנתונים.
5. **nltk** - ספרייה לעיבוד שפה טבעית, כולל ניתוח ושימוש במילון WordNet.
6. **re** - ביצוע התאמות וחיפושים באמצעות ביטויים רגולריים.
7. **requests** - שליחת בקשות HTTP ותקשורת עם אתרי אינטרנט.
8. **BeautifulSoup** - ניתוח וקבלת תוכן HTML מאתרים, משומש ברכיב הזחלן, וגם ביצירת האינדקס.
9. **urllib.parse** - שימש לניתוח וביצוע מניפולציות על כתובות URL.
10. **collections** – בוצע שימוש במחסנית עבור הזחלן, ובמונה לניהול יעיל של נתונים בדף סטטיסטיקות.
11. **random** - משמש לבחירות אקראיות, יצירת מספרים אקראיים, שימושים סטטיסטיים פשוטים.
12. **numpy** - ביצוע חישובים מתמטיים וניתוחים יעילים במערכת.
13. **google.generativeai** - שימוש בצ'ט בוט של ג'מיני.

קבצים מרכזיים: בפרויקט שלנו אנו משתמשים ב- Google Colab, המאפשר סביבת עבודה שיתופית מבוססת ענן למחברות Jupyter. הקובץ העיקרי - **Kakadoo.ipynb**, מרכז בתוכו את הפרויקט כולו, את לוגיקת היישום, את רכיבי הממשק, ואת התקשורת עם בסיס הנתונים.

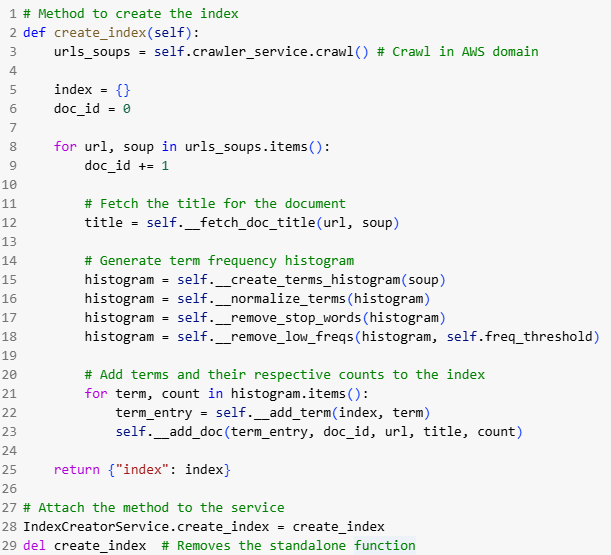
1. **קטעי קוד\פונקציות מרכזיות(ע"פ סדר כרונולוגי)**

היישום שלנו מחולק לפי קטעי קוד עיקריים, בכל אחד נפרט את קטעי הקוד ופונקציות עיקריות.

1. **ספריות והתקנות (Libraries and Installations) -** אחראי על יבוא / התקנת ספריות רלוונטיות.
2. **מיקרו-שירות זחלן (Crawler Microservice) -** מיקרו-שירות של זחלן שמחלץ דפי HTML סטטיים  
   מ- Domain מסוים אשר מועבר אליו. בפרויקט מועבר אליו ה- Domain של AWS והוא מוגדר לחלץ כ- 200 דפים.  
   **פונקציות עיקריות:**  
   **is\_within\_domain** – בדיקה שהאתר עדיין חלק מה - Domain  
   **contains\_non\_english\_codes** – בדיקה שכתובת האתר מכילה קוד של שפה שהיא אינה אנגלית (כך שנוכל לדלג על כתובת זו).  
   **contains\_non\_relevant\_keywords** – בדיקה שכתובת האתר מכילה מילות מפתח שלא רלוונטיות (כך שנוכל לדלג על כתובת זו).  
   **non\_relevant\_url -** שימוש בשלושת המתודות לעיל, על מנת לפסול כתובת במהלך הזחילה.  
   **normalize\_url** – מחיקה של Query Parameters ו- Fragments מכתובות במהלך הזחילה (כדי להימנע משכפול כתובות).  
   **crawl** – פעולת הזחילה שמתבצעת באלגוריתם BFS.
3. **מיקרו-שירות ליצירת אינדקס (Index Creator Microservice) -** מיקרו-שירות שאחראי על יצירת האינדקס. מאגד מידע טקסטואלי מתוך דפי HTML, ומבצע עליו עיבוד להפקת האינדקס. בפרויקט מקבל stop words מותאמים ל- AWS.  
   **פונקציות עיקריות:**  
   **create\_terms\_histogram** – יצירת היסטוגרמה המכילה את תדירות הופעות של מילים בדף HTML מסויים.  
   **remove\_stop\_words** – הסרת מילים נפוצות שאינן מוסיפות ערך משמעותי להיסטוגרמה.  
   **normalize\_terms** – המרה של מילים בהיסטוגרמה לצורתן הבסיסית, ע"י שימוש ב- apply\_lemmatization שמבצע למטיזציה למושג מסויים, כדי למנוע כפילויות.  
   **remove\_low\_freqs** – הסרה של מושגים מהיסטוגרמה שהופיעו בתדירות נמוכה מהרף המוגדר   
   (ברירת מחדל מוגדרת ל- 7), כדי לצמצם את האינדקס למילים יותר משמעותיות.  
   **add\_term** – יצירת רשומה חדשה עבור מילה באינדקס אם היא אינה קיימת.  
   **add\_doc** – הוספת מידע על מסמך מסויים לרשומת מילה באינדקס.  
   **fetch\_doc\_title** – שליפת כותרת הדף (נשמר כחלק מהמידע על מסמך מסויים באינדקס).  
   **create\_index** – יצירת האינדקס ע"י הפעלת שירות הזחלן לחילוץ הדפים, ושימוש במתודות המתוארות לעיל.
4. **מיקרו-שירות לשליפת נתונים (Data Fetcher Microserice) -** מיקרו-שירות לשליפת מידע מה- Database.  
   **פונקציות עיקריות:  
   fetch\_index** – שליפה של כל האינדקס.  
   **fetch\_term\_docs** – שליפה של כל נתוני המסמכים עבור מושג מסוים.
5. **מיקרו-שירות לניהול (Administration Microservice) -** מיקרו-שירות לניהול האינדקס.  
   **פונקציות עיקריות:**  
   **delete\_index** – מחיקת האינדקס הקיים מה- Database.  
   **recreate\_index** – שימוש בשירות יצירת האינדקס לצורך יצירת אינדקס מחדש (או לראשונה).  
   **fetch\_terms** – חילוץ כלל המושגים הקיימים באינדקס.  
   **fetch\_urls** – חילוץ הלינקים תחת מושג מסויים באינדקס.  
   **delete\_docs** – מחיקה של לינקים ספציפיים תחת מושג מסוים באינדקס.  
   **delete\_term** – מחיקה של מושג מסויים מהאינדקס.
6. **מיקרו-שירות לשאילתות (Query Microservice) -** מיקרו-שירות לעיבוד שאילתה ושליפת התוצאות הרלוונטיות מהאינדקס.  
   **פונקציות עיקריות:**  
   **normalize\_query** – נרמול של השאילתה ע"י הפעלת apply\_lemmatization והמרה ל- lowercase של כל מושג בשאילתה.  
   **url\_conatins\_terms** – בדיקה האם מושגי השאילתה נמצאים גם בכתובת (משפר את הדירוג).  
   **update\_maps** – עדכון של אינפורמציית העיבוד על מסמכים.  
   **calculate\_ranks** – חישוב הדרגות של המסמכים שנשלפו.  
   **fetch\_results** – מיון מסמכים לפי הדרגות ושליחתם.  
   **process\_query** – עיבוד השאילתה. מבצע שימוש במתודות המתוארות לעיל.
7. **מיקרו-שירות לסטטיסטיקות (Statistics Microservice) -** מיקרו-שירות לביצוע מניפולציות להפקת סטטיסטיקות וכן ליצירת גרפים בצורה גנרית.  
   **פונקציות עיקריות:**  
   **get\_most\_common\_words** – מבצע מניפולציה על אינדקס כדי להפיק את המושגים הנפוצים ביותר.  
   **get\_least\_common\_words** – מבצע מניפולציה על אינדקס כדי להפיק את המושגים הכי פחות נפוצים.  
   **get\_random\_words** – מבצע מניפולציה על אינדקס ומחזיר מספר הפעמים אשר מושגים רנדומליים בשימוש.  
   **get\_common\_docs** – מבצע מניפולציה על האינדקס כדי להפיק את המסמכים שבהם יש הכי הרבה מושגים.
8. **הגדרות עיצוב ויצירת דפים / קומפוננטות כלליות (HTML & CSS) -** קטע קוד זה אחראי על הזרקות CSS ועל יצירת קומפוננטות הקשורות לממשק המשתמש, קטע הקוד מרכז את כל מאפייני UI הניתנים להפרדה מהחלקים הלוגיים של המערכת.  
   **פונקציות עיקריות:**  
   **inject\_css** - מזריקה קוד CSS מותאם אישית לתוך המחברת כדי לעצב אלמנטים באופן מותאם אישית.  
   **create\_header** - יוצרת כותרת עיצובית ליישום, כולל לוגו, כותרת רקע ויישור תוכן.  
   **create\_ResultView\_page** - מחזירה רכיב HTML שמספק עיצוב לתצוגת טבלה עם   
   (גלילה ושורות מעוצבות) של תוצאות חיפוש מונחים במנוע חיפוש.  
   **create admin page** - יוצרת עיצוב מותאם לדף מנהל (דף ניהול אינדקס).  
   **create\_blur\_divider** - יוצרת רכיב עיצוב של מפריד מטושטש שמוסיף חלוקה ויזואלית בין חלקים של הדף.  
   **create\_statistics\_chart\_page** - מספקת עיצוב לטאבים ולגרפים סטטיסטיים ושמירה על גמישות התוכן.
9. **דף ראשי (Main View GUI) -** קטע קוד זה אחראי על יצירת כפתורים בדף הבית, לוגיקת המעברים בין דף הבית לדפים הבאים, יצירת עיצוב בסיס לכל היישום.  
   **פונקציות עיקריות:**  
   **create\_navigation\_buttons** - יוצרת שני סטים של כפתורי ניווט, האחד לדף הניהול והשני להפעלת שאילתה ולמעבר לסטטיסטיקות, עם עיצוב מותאם אישית.  
   **create\_photo\_container** - יוצר רכיב HTML שמציג תמונה בעיצוב רספונסיבי, כולל מיקום ועיצוב מותאם.  
   **BaseView** (מחלקה) - מחלקת בסיס לכל התצוגות שמאגדת רכיבים ותמונה ליצירת ממשק משתמש רספונסיבי וממוקד.  
   **MainDashboardView** (מחלקה) - מנהלת את הדף הראשי, כולל הצגת כפתורים, לוגו ותצוגות שונות באופן דינמי. כולל הגדרת צבע רקע לכל הדפים הפנימיים.
10. **דף מנהל לניהול אינדקס (Admin Page GUI) -** קטע קוד זה הינו מחלקה אשר מייצגת את תצוגת דף מנהל במערכת. מטרתה לספק ממשק אינטראקטיבי לניהול מושגים (Terms) וכתובות URL במאגר הנתונים שלנו, כולל הצגת מידע, מחיקה ועדכון נתונים. המחלקה משלבת פונקציונליות לעדכון טבלאות נתונים, הפעלת שאילתות, מחיקת פריטים, והחזרת משתמשים לדף הראשי.  
    **פונקציות עיקריות:**  
    **\_\_init\_\_** - מגדירה ומאתחלת את דף הניהול, כולל רשימות נפתחות, כפתורים, הודעות שגיאה ורכיבי טבלאות.  
    **initialize\_checkbox\_container** - מאתחלת את רכיב תיבות הסימון (Checkboxes) עבור הטבלה עם תיבות בחירה כלליות וממשק גלילה.  
    **update\_table\_data** - מעדכנת את תוכן הטבלה לפי הנתונים הנוכחיים ומייצרת תיבות סימון לכל פריט.  
    **on\_select\_all\_change** - מנהלת את בחירת תיבות הסימון כאשר משתמש בוחר או מבטל "בחר הכל".  
    **on\_search** - מטפלת בלחיצה על כפתור 'חיפוש' להצגת מידע על המונח הנבחר מעדכנת טבלה עם נתונים חדשים.  
    **on\_delete url** - מטפלת במחיקת כתובות URL שנבחרו מתיבות הסימון ומעדכנת את הטבלה בהתאם.  
    **on\_delete\_term** - מוחקת את המונח הנוכחי ומעדכנת את רשימת המונחים לאחר המחיקה ואת הטבלה המוצגת.  
     **on\_recreate\_index**- מטפלת בבקשה ליצירת אינדקס מחדש, כולל הצגת הודעות מצב ועדכון רשימת המונחים.  
    **on\_return** - מחזירה את המשתמש לדף הראשי עם הכפתורים המרכזיים.
11. **דף הרצת שאילתא (Enter Query GUI) -** קטע קוד זה היינו מחלקה המייצגת את תצוגת דף השאילתות, המאפשרת למשתמשים להקליד שאילתא, לבצע חיפוש, וליהנות ממידע "Fun Fact" על AWS. המחלקה כוללת ממשק אינטראקטיבי עם שדות טקסט, כפתורים והודעות שגיאה.  
    **פונקציות עיקריות:**  
    **\_\_init\_\_** - מאתחלת את התצוגה, כולל שדה להזנת שאילתא, כפתורים לחיפוש, "AWS Fun Fact", חזרה לדף הראשי, יחד עם עיצוב ומבנה דף אינטראקטיבי.  
    **on\_return\_main\_clicked** - מטפלת בלחיצה על כפתור חזרה לדף הראשי ומחזירה משתמש לתפריט הראשי.  
    **on\_search\_clicked** - מטפלת בלחיצה על כפתור החיפוש. אם השאילתה ריקה, מוצגת הודעת שגיאה.   
    אחרת, מוצג דף התוצאות עם שאילתת המשתמש.  
    **on\_fun\_fact\_clicked** - מציגה עובדה אקראית ומעניינת על AWS מתוך רשימת עובדות מוגדרת מראש.
12. **דף תוצאת חיפוש (Query Results GUI) -** קטע קוד זה היינו מחלקה אשר מציגה את תוצאות החיפוש של השאילתה שהוזנה על ידי המשתמש. היא מציגה את השאילתה המבוקשת, תוצאות בדירוג מסודר (כולל קישורים), או הודעה שאין תוצאות, ומספקת כפתור חזרה לדף החיפוש.  
    **פונקציות עיקריות:**  
    **\_\_init\_\_** - אתחול דף התוצאות, מציגה שאילתה, טבלה של תוצאות מדורגות(מציגה הודעה מתאימה אם אין).  
    **on\_return\_to\_query** - מטפלת בלחיצה על כפתור חזרה לדף השאילתות (EnterQueryView).
13. **דף צ'ט בוט (AWS Chatbot GUI) -** קטע קוד זה מורכב מפונקציית בדיקת רלוונטיות של שאילתה, אתחול מילות מפתח, אתחול מודל, ומחלקה אשר מייצגת תצוגה לממשק משתמש המאפשר למשתמשים לשאול שאלות על שירותי AWS ולקבל תשובות מצ'אט בוט אינטראקטיבי. הממשק כולל שדה קלט לשאלה, כפתור לשליחה, ותצוגת תשובה.  
    **פונקציות עיקריות:**  
    **ask\_aws\_chatbot** - פונקציה המקבלת שאלה על AWS, מוודאת שהיא קשורה לשירותי AWS, ושולחת אותה למודל(Generative AI) לקבלת תשובה. במידה והשאלה אינה קשורה ל-AWS, מחזירה הודעה מתאימה.  
    **\_\_init\_\_** - מאתחלת את התצוגה של ה- AWS Chatbot, כולל שדה להזנת שאלה, כפתור לשליחה, תווית להצגת תשובה, כפתור חזרה לדף הראשי.  
    **on\_chat\_button\_clicked** - מטפלת בלחיצה על כפתור השאלה. אם הקלט ריק, מוצגת הודעת שגיאה.   
    אחרת, השאלה נשלחת לצ'אטבוט, והתשובה מוצגת למשתמש.  
    **on\_return\_clicked** - מטפלת בלחיצה על כפתור חזרה לדף הראשי ומחזירה את המשתמש לתפריט הראשי של לוח המחוונים.
14. **דף סטטיסטיקות (Statistics GUI) -** קטע קוד זה היינו מחלקה אשר מציגה נתונים סטטיסטיים על אינדקס המערכת, כולל מונחים נפוצים, מונחים נדירים, מונחים אקראיים. היא מאפשרת הצגה אינטראקטיבית באמצעות כרטיסיות (Tabs) ודיאגרמות, כולל אפשרות רענון מילים אקראית וכפתור חזרה לעמוד הראשי.  
    **פונקציות עיקריות:**  
    **\_\_init\_\_** - מאתחלת את תצוגת הנתונים הסטטיסטיים, כולל טאבים המציגות סטטיסטיקות שונות, יחד עם כפתור חזרה לעמוד הראשי.  
    **create\_most\_common\_tab** - יוצרת כרטיסייה להצגת מילים הנפוצות ביותר עם דיאגרמת עמודות.  
    **create\_least\_common\_tab** - יוצרת כרטיסייה להצגת מילים הנדירות ביותר עם דיאגרמת עמודות והסבר.  
    **create\_random\_tab** - יוצרת כרטיסייה להצגת מילים אקראיות עם דיאגרמת עמודות וכפתור לרענון התוצאות.  
    **create\_most\_common\_docs\_tab** - יוצרת כרטיסייה להצגת מסמכים נפוצים ביותר באינדקס עם דיאגרמת עוגה.  
    **generate\_most\_common\_chart** - יוצרת דיאגרמת עמודות של המילים הנפוצות ביותר על בסיס הנתונים.  
    **generate\_least\_common\_chart** - יוצרת דיאגרמת עמודות של המילים הנדירות ביותר על בסיס הנתונים.  
    **generate\_random\_chart** - יוצרת דיאגרמת עמודות של מילים אקראיות שנבחרות באופן דינמי.  
    **generate\_common\_docs\_chart** - יוצרת דיאגרמת עוגה המציגה את המסמכים הנפוצים ביותר, עם פירוט על תדירות ההופעה.  
    **refresh\_random\_chart** - מרעננת את דיאגרמת המילים האקראיות בלחיצת כפתור.  
    **on\_return** - מטפלת בלחיצה על כפתור חזרה לעמוד הראשי ומחזירה את המשתמש לתפריט הראשי.
15. **בקר שליטה ראשי של המערכת (Controllers) -** קטע קוד מורכב מפונקציה ליצירת לוגיקה עבור כפתורי מסך ראשי(display\_main\_dashboard) ומחלקה אשר מנהלת את תצוגת לוח המחוונים של המערכת.   
    DashboardController - אחראית לחיבור כפתורים לפונקציונליות המתאימה, ניהול מעבר בין עמודים, ואבטחת גישה לדף הניהול באמצעות אימות סיסמה.   
    **פונקציות עיקריות:**  
    **display\_main\_dashboard** - אחראית על הלוגיקה של הדף הראשי. יוצרת את כל הרכיבים הנדרשים, כולל כותרת, כפתורי ניווט, רכיב תמונה, ומחברת אותם יחד עם בקר שליטה ראשי. בסיום, היא מציגה את הדף הראשי עם ה-CSS המותאם.  
    **\_\_init\_\_** - מאתחלת את בקר שליטה ראשי, מקבלת את הדף עצמו, כפתורי הניווט, ותמונה, ומגדירה את הקישורים בין כפתורים לפונקציות.  
    **setup\_callbacks** - מחברת את הלחיצות על הכפתורים לפונקציות המתאימות, כגון מעבר לדף הניהול, דף השאילתות, ודף הסטטיסטיקות, שומרת על מעקב המעבר בין דפים.  
    **handle\_admin\_page** - מנהלת את הגישה לדף הניהול, כולל קבלת סיסמה ובדיקה אם היא נכונה. במידה והסיסמה נכונה, המשתמש עובר לדף הניהול.  
    **handle\_enter\_query** - מטפלת בלחיצה על כפתור מעבר לדף השאילתות ופותחת את EnterQueryView (תצוגת דף חיפוש).   
    **handle\_statistics** - מטפלת בלחיצה על כפתור הסטטיסטיקות ופותחת את תצוגת StatisticsView.
16. **קטע קוד ראשי (Main) -** קטע הקוד מגדיר את הקשרים בין רכיבי המערכת, מחבר את המיקרו-שירותים לבסיס הנתונים, ומזניק את תצוגת דף ראשי.  
    במקטע קוד זה אין פונקציות, אלא יש קריאה ליצירת אובייקטים למיקרו שירותים שנשתמש בהם בדפים שונים של המערכת.
17. **קטעי קוד מעניינים:**

  
**calculate\_ranks** – במתודה זו דרגות המסמכים   
הראשוניות שהתקבלו על בסיס סכום מספר הפעמים  
שכל מושג מופיע במסמך, מעודכנים לדירוג הסופי.   
הדירוג הסופי נקבע ע"י אורך השאילתה ומספר המושגים  
מהשאילתה שמופיעים במסמך עצמו (כך שאם פחות  
מושגים מהשאילתה מופיעים במסמך עצמו, דירוגו קטן),  
ובנוסף מספר המושגים מהשאילתה שמופיעים בכתובת  
ה- URL (כך שאם המושגים מופיעים בכתובת ה- URL  
עצמה דירוגו גדל משמעותית)

**crawl** – פעולת הזחילה שמתבצעת   
באלגוריתם BFS. שולפת תוכן HTML (soup)  
מ-URL-ים רלוונטיים, ושומרת אותם במילון.   
בכניסה לכל לינק נשמרים הרפרנסים להמשך פעולת  
הזחלן. התהליך נעצר כשממלאים את המילון  
במספר URL-ים מוגדר מראש או שהתור מתרוקן.



**create\_index** – יצירת האינדקס של מונחים   
מתוך תוכן שנסרק, על ידי חילוץ מונחים,   
נורמליזציה, והסרת מילים חסרות משמעות   
ותדירות נמוכה. האינדקס משייך מונחים   
למסמכים עם פרטים כמו URL, כותרת   
ותדירות הופעה.

**תיק משתמש**

1. **מבוא**

מדריך זה נועד להסביר את **אופן השימוש ביישום Kakadoo Engine**, על כל מרכיביו, כדי למצות את הפונקציות שלו בצורה מיטבית. יישום זה מהווה **מנוע חיפוש עבור איתור קישורים מאתר AWS**, לצד כלים נוספים כמו:

* **פיצ'ר Fun Fact** – מספק עובדות מעניינות בצורה רנדומלית על שירותי AWS.
* **צ'אטבוט מבוסס AI** – מספק מענה לשאלות בנושאי שירותי AWS, תוך שימוש במידע מעודכן ובתשובות מותאמות לצרכי המשתמש.
* **שירות סטטיסטיקות** – מציג נתונים סטטיסטיים על האינדקס (מאגר ה- URLs שמתוכו מתבצעת שליפת תוצאות השאילתות באופן יעיל).
* **שירות אדמיניסטרציה** – ממשק ייעודי עבור מנהלי המערכת לניהול האינדקס, הכולל צפייה במושגים הקיימים באינדקס, מחיקה של מושגים, מחיקת URLs המשוייכים למושגים, ויצירת אינדקס מחדש תוך הפעלת זחלן (שירות לשליפת קישורים) על אתר AWS.

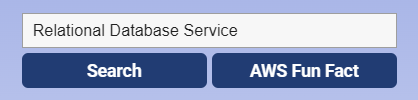
1. **השימוש במערכת**
   1. **הרצת שאילתה לחיפוש קישורים**
      1. על מנת להריץ שאילתה, תחילה יש ללחוץ על "Run Query" בעמוד הבית. אם אינכם בעמוד הבית יש לבצע את סעיף 2.1.6 בעמוד בו נמצאים.



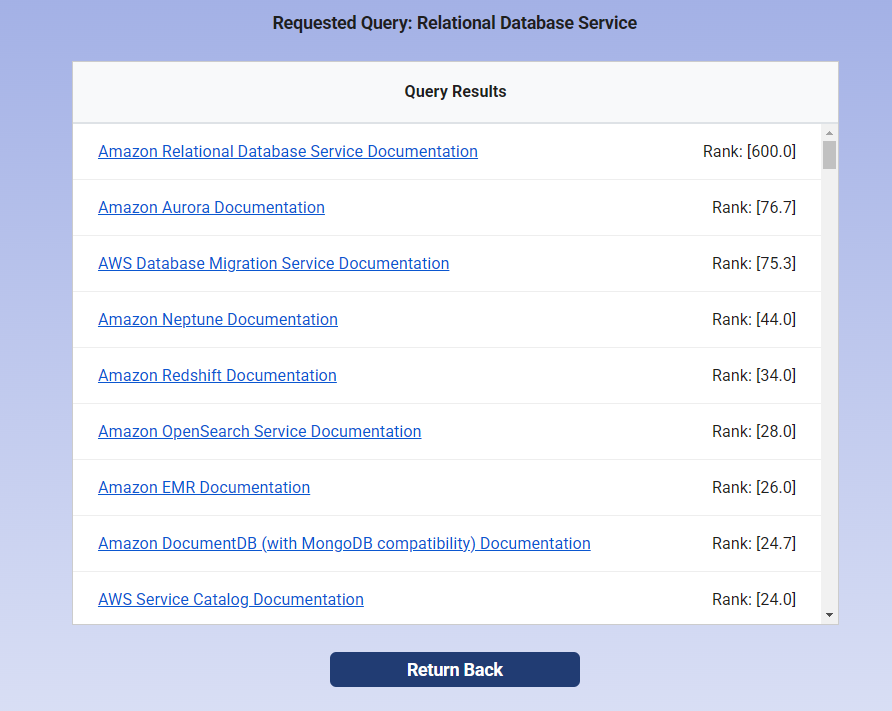
* + 1. בעמוד שמוצג, יש להזין את השאילתה הרצויה בתיבת החיפוש. השאילתה חייבת להיות באנגלית. ניתן להזין את השאילתה באותיות קטנות, גדולות, או שילוב שלהן – התוצאה תהיה זהה. בניסיון להריץ שאילתה ריקה תתקבל שגיאה.



* + 1. לאחר הזנת השאילתה, יש ללחוץ על כפתור "Search".



* + 1. יש להמתין כ- 2 שניות לעיבוד השאילתה.
    2. לאחר העיבוד, יוצגו הקישורים הרלוונטיים, מסודרים מהדירוג הגבוה ביותר לנמוך ביותר. הדירוג מייצג את מידת הרלוונטיות של כל קישור לשאילתה שהוזנה. הוא מחושב על סמך פרמטרים של מספר המושגים המופיעים בקישור, כמות הופעות מושגי השאילתה בתוכן הקישור, וגם בכתובת הקישור עצמה. לכן, הקפידו להזין שאילתה תמציתית ומדוייקת, בנושאי AWS, על מנת לקבל את התוצאות הטובות ביותר.



אם לא התקבלו תוצאות, תוצג ההודעה “No results found”. במקרה שרוצים להריץ שאילתה נוספתיש ללחוץ על כפתור "Return Back" (שבתחתית הדף כמו תמונה לעיל), על מנת לחזור לדף הרצת שאילתה, ולבצע את פעולות 2.1.2-2.1.4. אם מעוניינים לסיים, יש ללחוץ על כפתור “Return Back” ולהמשיך לסעיף הבא.

* + 1. על מנת לסיים את השימוש בכלי, יש ללחוץ על “Return to Main Page” שבתחתית הדף.
  1. **הרצת פיצ'ר Fun Fact**
     1. יש ללחוץ על "Run Query" בעמוד הבית, כפי שמתואר בסעיף 2.1.1.
     2. יש ללחוץ על "AWS Fun Fact".



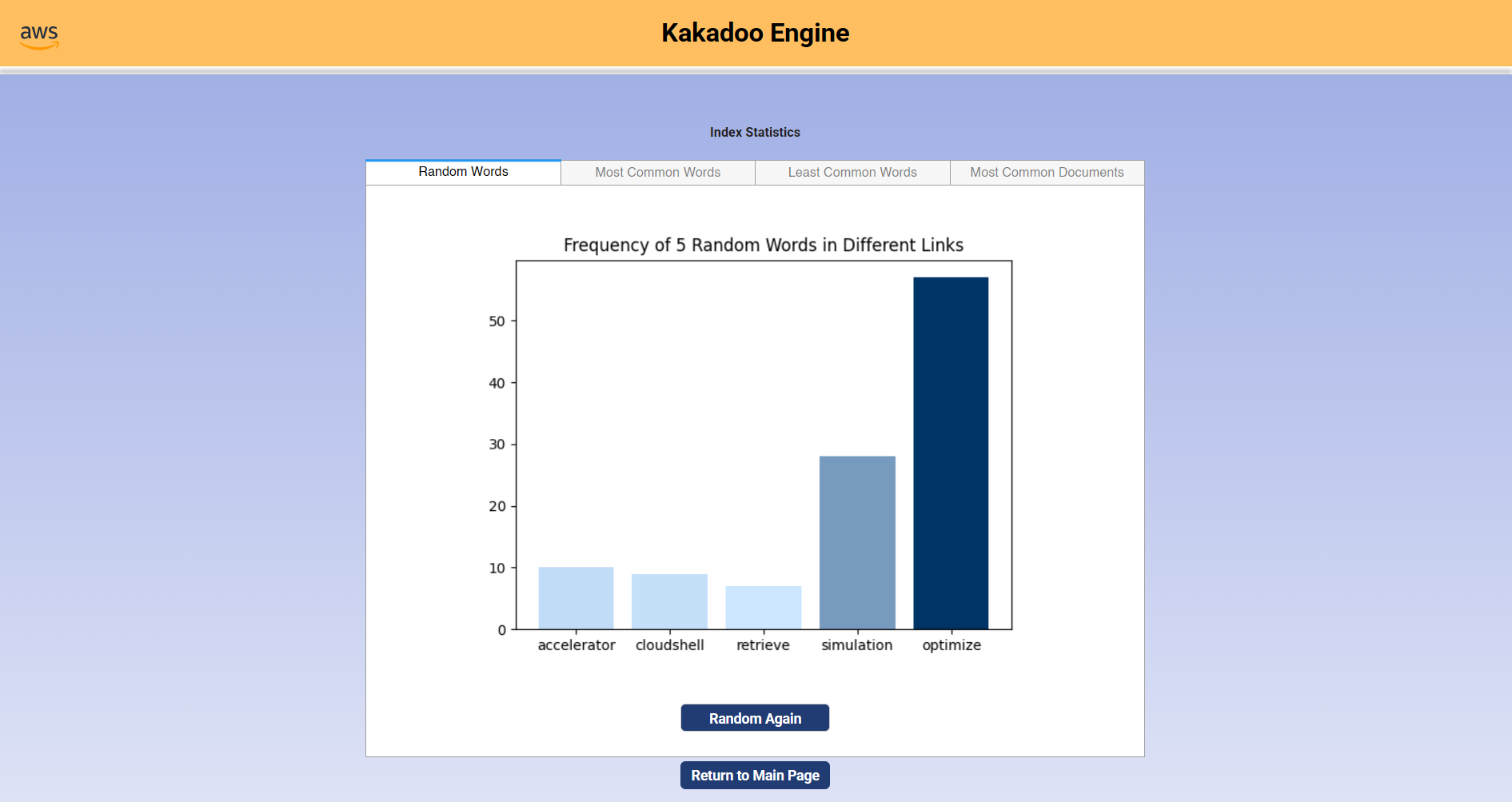
* + 1. תוצג עובדה מעניינת באופן רנדומלי. על מנת להציג עובדה מעניינת אחרת, יש לחזור על סעיף 2.2.2.



* 1. **שימוש בשירות סטטיסטיקות**
     1. על מנת לפתוח את שירות הסטטיסטיקות, יש ללחוץ על “Statistics” בעמוד הבית. אם אינכם בעמוד הבית יש לבצע את סעיף 2.1.6 בעמוד בו נמצאים.



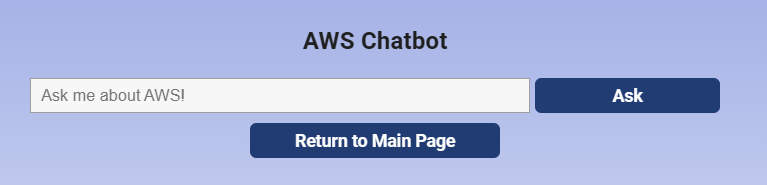
* + 1. על מנת לעבור בין גרפים, יש ללחוץ על ה- TAB הרצוי.



* + 1. עבור גרפים שנתוניהם מוצגים רנדומלית, ניתן ללחוץ על כפתור “Random Again”, שנמצא מתחת לגרף כפי שנתן לראות בתמונה לעיל, כדי להציג נתונים רנדומליים אחרים.
    2. בסיום השימוש בשירות, יש לבצע את סעיף 2.1.6 על מנת לחזור לדף הבית.
  1. **שימוש בצ'אטבוט**
     1. על מנת לפתוח את הצ'אטבוט, יש ללחוץ על “AWS Chatbot” בעמוד הבית. אם אינכם בעמוד הבית יש לבצע את סעיף 2.1.6 בעמוד בו נמצאים.



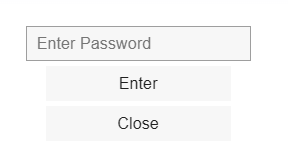
* + 1. בעמוד שמוצג, יש להזין שאלה בנושא AWS בתיבה. שאלה בנושאים אחרים תענה בדרישה להזין שאלה שכן קשורה ל- AWS.



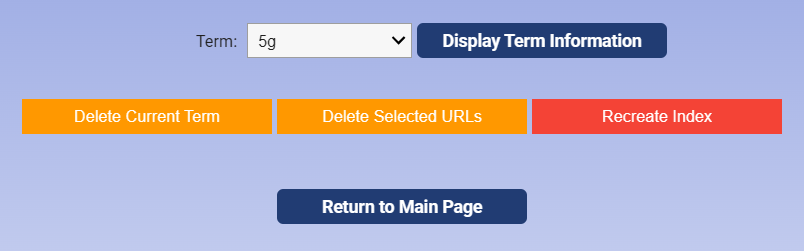
* + 1. יש ללחוץ על כפתור “Ask” כפי שניתן לראות בתמונה לעיל.
    2. תוצג תשובה מתומצתת בנושא הנדרש.
    3. בסיום השימוש בשירות, יש לבצע את סעיף 2.1.6 על מנת לחזור לדף הבית.
  1. **שימוש בשירות אדמיניסטרציה**
     1. על מנת להיכנס לשירות, יש ללחוץ תחילה על כפתור “Admin Page” בעמוד הבית. אם אינכם בעמוד הבית יש לבצע את סעיף 2.1.6 בעמוד בו נמצאים.



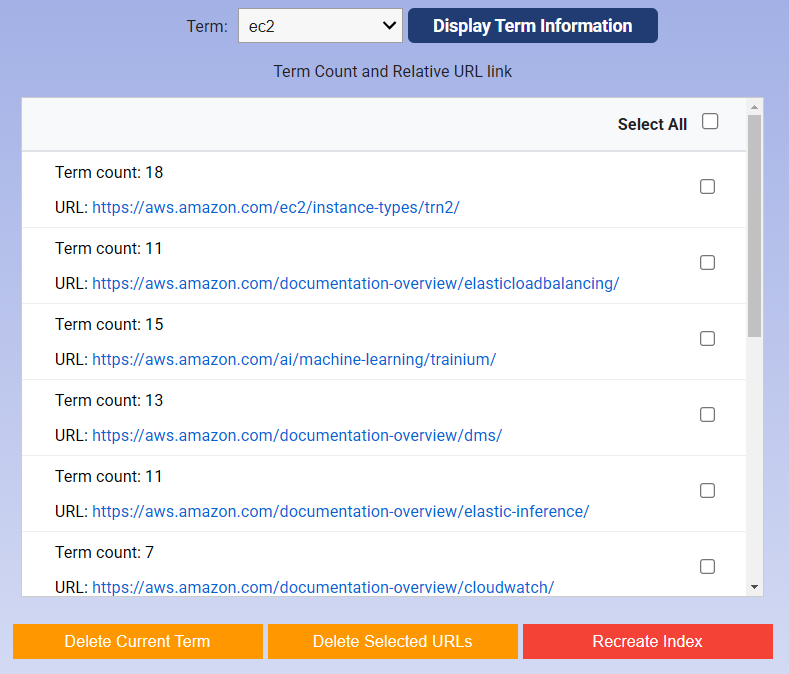
* + 1. תוצג דרישה להזנת סיסמת ה- Admin. יש להזין “123456” בתיבת הטקסט.



* + 1. יש ללחוץ על כפתור “Enter” כפי שניתן לראות בתמונה לעיל. אם ברצונכם לוותר על השימוש בשירות, ניתן ללחוץ על כפתור “Close”.
    2. ניתן לצפות ברשימת המושגים שנמצאים באינדקס על ידי לחיצה על ה- Dropdown:

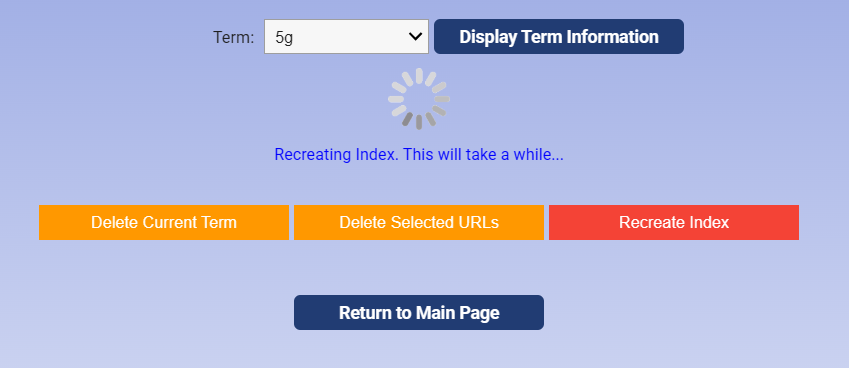


* + 1. על מנת לצפות ברשימת הקישורים שבהם המושג נמצא, יש ללחוץ על “Display Term Information” כפי שניתן לראות בתמונה לעיל. יוצגו כל ה- URLs ומספר הפעמים שבהם המושג מופיע.

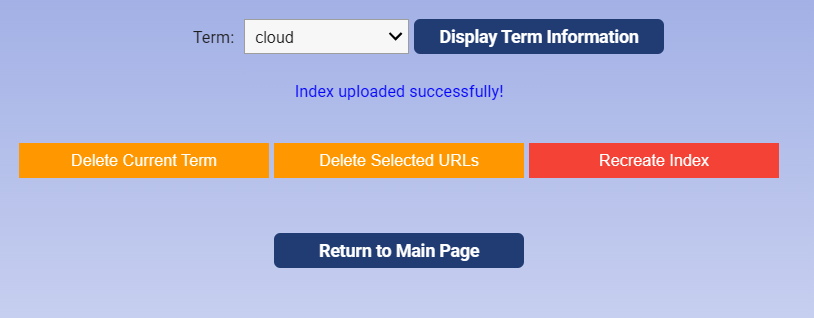


**אזהרה – הפעולות הבאות לא ניתנות לביטול לאחר ביצוען**

* + 1. על מנת למחוק את האינדקס הקיים, ולהפעיל זחלן שרץ על אתר AWS מנת ליצור אינדקס חדש  
       ב- Database, יש ללחוץ על כפתור “Recreate Index” כפי שניתן לראות בתמונה לעיל. יש להמתין  
       כ- 2 דקות עד לסיום הפעולה.



* + 1. בסיום יצירת האינדקס, אם העדכון התבצע בהצלחה תתקבל ההודעה “Index uploaded successfully!”. אחרת, תתקבל ההודעה “Error uploading index” בצירוף הסיבה לשגיאה. במקרה שהתקבלה שגיאה, יש לוודא שהחיבור לאינטרנט יציב ולאחר מכן יש לבצע את פעולה 2.5.6 מחדש.



* + 1. על מנת למחוק את המושג וכל הקישורים שתחתיו, יש ללחוץ על “Delete Current Term”.
    2. על מנת למחוק קישורים תחת המושג, יש לסמנם ע"י ה- Checkbox המקושר אליהם וללחוץ על כפתור “Delete Selected URLs” כפי שניתן לראות בתמונה לעיל. אם ברצונכם להסיר את רוב הקישורים תחת אותו מושג, ניתן לקצר תהליכים על ידי לחיצה על “Select All” Checkbox ולהסיר את סימון  
       ה- Checkbox מהקישורים שאינכם מעוניינים למחוק.
    3. בסיום השימוש בשירות, יש לבצע את סעיף 2.1.6 על מנת לחזור לדף הבית.

שקיפות אלגוריתמית בתיק המשתמש

* בהסבר על שירות החיפוש, הובהר למשתמשים מדוע עליהם לכתוב שאילתה תמציתית ומדוייקת בנושאי AWS, על סמך הסבר כללי של הפרמטרים שנלקחים בחשבון לחישוב הדירוג.
* בהסבר על שירות האדמיניסטרציה, הובהר למשתמשים שיצירת האינדקס מצריכה הפעלת זחלן על אתר AWS, פעולה שלוקחת זמן, ועל כן אפשר להבין מדוע תהליך זה אורך כ- 2 דקות (פעולה שאורכת הכי הרבה זמן ביחס לפעולות אחרות במערכת).