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

**תרגיל בית 2** – **עבודה בצוותי העבודה**

מועד הגשה: 12.1.25

המשימה בתרגיל זה: בניית אינדקס למנוע החיפוש, ובניית מסכים מרכזיים במנוע החיפוש

שימו לב: למטלה זו שלושה חלקים

*חלק ראשון : בניית אינדקס (40 נקודות)*

יש למנות מהנדס.ת מערכת בכל צוות, אשר יהיה אחראי על הגדרת הדרישות ההנדסיות, ועל הממשק מול החומרה.

מהנדס המערכת – עדן קנטור

*בכל צוות על כל אחד לבחור אחד מהתפקידים הבאים (יש להחליף מתרגיל בית 1 ) (10 נקודות)*

*scrum master -מרכז את העבודה*

*,frontend developer – פיתוח החלק האחראי על הצגה ללקוח.*

*,backend developer – פיתוח מסד הנתונים והעבודה מולו.*

*product manager – ייצוג הלקוח בצוות (בהתאם לחשיבה העיצובית שבוצעה).*

*UI – עיצוב הממשק*

*QA – בדיקות התוכנה*

נא לרשום את שם הסטודנט.ית בתרגיל זה. על מהנדס.ת המערכת לכתוב כיצד נעשתה חלוקת העבודה מול הצוות, מה היו המשימות של כל חבר צוות, האם היה ממשק בין חברי הצוות, והאם המשימות מולאו:

|  |  |  |
| --- | --- | --- |
| **איטרציה 1** | | |
| **שם חבר הצוות ותפקיד בתרגיל זה** | **משימות שהוקצו** | **משימות שהושלמו** |
| שי ליברמן – backend | מסך אדמין | מסך אדמין |
| יניב שתיל - UI | נגן פיצ'ר, עיצוב המערכת והלוגו | נגן פיצ'ר, עיצוב המערכת והלוגו |
| עדן קנטור – scrum master | מסך חיפוש שאילתא ופלט שאילתא | מסך חיפוש שאילתא ופלט שאילתא |
| דני לבובסקי - frontend | מסך סטטיסטיקות, עיצוב המסכים | מסך סטטיסטיקות, עיצוב המסכים |
| מירנא אבו חליל – product manager | הגדרת מדדים להצלחה, ארכיטקטורה | הגדרת מדדים להצלחה, ארכיטקטורה |
| דיאנא חוג'יראת - QA | כללי הזהב של שניידרמן, חישוב SUS, בדיקות של תקינות המערכת | כללי הזהב של שניידרמן, חישוב SUS, בדיקות של תקינות המערכת |

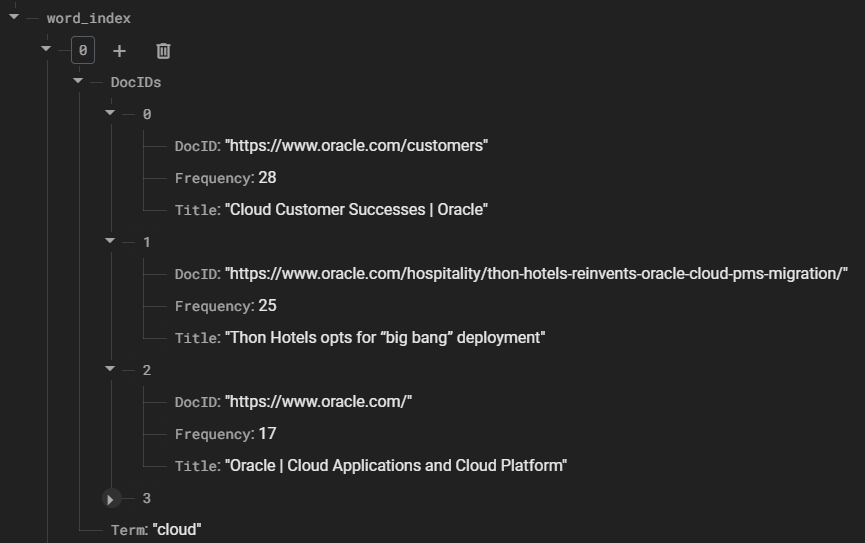
בניית אינדקס (30 נקודות)

באיטרציה זו עליכם לבנות את מסד הנתונים שמכיל את האינדקס של המילים המשמעותיות באתר אותו אתם חוקרים.

מבנה האינדקס צריך להיות אחיד לכל הקבוצות , ולכלול **לפחות** את השדות הבאים  **( אין לשנות את שמות השדות!)**:

|  |  |
| --- | --- |
| שם השדה | הסבר |
| term | term |
| DocIDs | רשימת קישורים לדפים המכילים את ה- term, ממוספרים לפי בחירתכם |

ממשו את האינדקס בקולאב.

***מבנה האינדקס ב-****FireBase* ***(להמחשה):  
  
***

1. **יש להגיש את הטבלה הבאה , תוך התייחסות למושבים שקיבלתם (10 נקודות):**

|  |  |  |
| --- | --- | --- |
| הערת משוב | האם התבצע שינוי באפליקציה בעקבות ההערה? | נימוק |
| הוספת chatbot | לא | לא הייתה דרישה לתרגיל הנוכחי, יתווסף בהמשך. |
| הוספת delete term | כן | נוסף כפתור במסך האדמין למחיקה. |
| צבעי המערכת יהיו יותר בולטים | לא | לאחר התייעצות הוחלט להשאיר את הצבעים, העדפה אישית. |
| הלינקים בעדכון term להפריד אותם | לא | נבצע את השינוי בגרסה עתידית. |
| הגרפים יציגו יותר מנושא אחד | כן | הוספנו הצגה של אינדקסים שמופיעים בהכי הרבה קישורים. |

1. **התייחסו ל -8 כללי הזהב של שניידרמן (הוצגו בתרגול). כיצד המערכת שלכם מבטאת אותם? (10 נקודות)**
   * + 1. חתירה לעקביות:

המערכת שומרת על עיצוב ואינטראקציה עקביים לאורך כל הממשק המשתמש. הדבר בא לידי ביטוי במספר אופנים:

- **עיצוב ויזואלי עקבי:** שימוש בסכמת צבעים אחידה הכוללת רקע כהה, טקסט בהיר וכפתורי פעולה בצבע כתום ("Badger Search", "Back to Search") לאורך כל המסכים.

- **מיתוג אחיד:** לוגו הגירית עם זכוכית המגדלת מופיע באופן עקבי ומשמש כסמל מזהה של המערכת.

- **מבנה תוצאות:** הצגת תוצאות החיפוש נשמרת באותו פורמט, עם הצגה עקבית של מידע על תדירות ההופעה ("Frequency") לכל תוצאה.

- **ניווט:** מיקום עקבי של כפתורי ניווט וחזרה, המאפשר למשתמש להתמצא בקלות בכל מסך במערכת.

* + - 1. לאפשר שימוש בקיצורי דרך:

המערכת מציעה מספר דרכים לגישה מהירה ויעילה למידע ופונקציונליות:

- **גישה מהירה למסכים מרכזיים:** כפתורי גישה ישירה ל"Statistics Page" ו"Admin Page" מהמסך הראשי.

- **חזרה מהירה:** כפתור "Back to Search" נגיש מכל מסך במערכת, מאפשר חזרה מהירה לחיפוש חדש.

- **ממשק חיפוש נגיש:** שדה חיפוש מרכזי עם אפשרות בחירת סוג חיפוש בוליאני (AND/ OR), המאפשר גישה מהירה למידע המבוקש לפי העדפה לסוג חיפוש.

* + - 1. הצע משוב אינפורמטיבי:

המערכת מספקת משוב ברור ומיידי למשתמש בכל שלב:

- **תוצאות חיפוש:** הצגה מיידית של מספר התוצאות שנמצאו למשל ("Your search appeared 71 times in the following links").

- **סטטיסטיקות מפורטות:** הצגת תדירות ("Frequency") לכל תוצאת חיפוש, מאפשרת למשתמש להבין את הרלוונטיות של כל תוצאה.

- **תצוגה גרפית:** גרף סטטיסטי המציג באופן ויזואלי את תדירות המונחים, עם אפשרות לבחירת סוג התצוגה (Bars/Line/Doughnut).

* + - 1. הפחת את עומס הזיכרון לטווח קצר:

המערכת מעוצבת כך שתפחית את הצורך בזכירת מידע:

- **הצגת הקשר:** תצוגה ברורה של השאילתא הנוכחית ("Results for: Cloud") בראש דף התוצאות.

- **ארגון מידע:** תצוגת תוצאות מאורגנת היטב עם מידע על תדירות ההופעה לכל תוצאה, כאשר כל לינק בתוצאת החיפוש מוצג ככותרת עניינית.

- **ויזואליזציה:** מידע סטטיסטי מוצג בצורה גרפית נוחה עם אפשרויות תצוגה שונות, המקלות על הבנת המידע ללא צורך בזכירת נתונים מספריים.

* + - 1. עיצוב דיאלוגים ברורים:

המערכת משתמשת בעיצוב ברור ואינטואיטיבי:

- **חיפוש מרכזי:** שדה חיפוש בולט עם לוגו הגירית וזכוכית המגדלת, המסמל באופן ברור את פונקציית החיפוש.

- **ניווט פשוט:** סרגל ניווט המורכב מכפתורים ברורים בראש העמוד המאפשר גישה למסכים השונים.

- **תצוגת סטטיסטיקות:** מסך סטטיסטיקות מאורגן היטב עם אפשרויות תצוגה ברורות ואינטואיטיביות.

* + - 1. אפשר היפוך קל של פעולות:

המערכת מאפשרת למשתמש לחזור ולשנות את פעולותיו בקלות:

- **ניווט חזרה:** כפתור "Back to Search" זמין בכל מסך, מאפשר חזרה מהירה לחיפוש חדש.

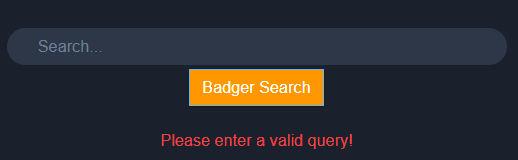
- **גמישות בתצוגה:** אפשרות לשנות את צורת התצוגה בסטטיסטיקות בין Bars, Line, ו-Doughnut.

- **עדכון חיפוש:** אפשרות קלה לביצוע חיפוש חדש או שינוי פרמטרים של החיפוש הנוכחי.  
- **בדף ה Admin :** נוכל לעדכן את האינדקסים, אך מבלי להקיש על עדכון לא ישמר וגם נוכל לבטל את השמירה דרך שינוי חוזר – נוכל לשנות כמה שבא לנו ואפילו אם נשנה למידע שהיה מקודם.

* + - 1. מנע שגיאות:

כאשר יש כישלון בחיפוש המערכת מציגה הודעה ברורה למשתמש על מה שהכישלון היה עם הסבר אם יש אפשרות לתקן.

כאשר לא מזינים מלל בשדה החיפוש המערכת מתריעה על שגיאה – ובכך המשתמש יבין שהשדה ריק.



* + - 1. מוקד השליטה אצל המשתמש :

המערכת מעניקה למשתמש שליטה מלאה על חווית השימוש:

- בחירת תצוגה: אפשרות לבחור בין סוגי תצוגה שונים בסטטיסטיקות (Bars/Line/Doughnut).

- **גמישות בחיפוש:** שליטה בסוג החיפוש (AND/ OR) ואפשרות לביצוע חיפושים חוזרים.

- **ניווט חופשי**: גישה חופשית בין המסכים השונים (חיפוש, סטטיסטיקות, מסך אדמין עבור מנהל).

-**חזרה לחיפוש:** אפשרות לחזור למסך החיפוש בכל שלב באמצעות כפתור "Back to Search".

1. **יש לרשום את ציון ה SUS של המערכת שלכם.מה מעיד הציון?(10 נקודות)**

קיבלנו 24 משובים – 2400

סכום המשובים- 2030

ציון סופי 84.5

(ציון מעל 68 נחשב מעל לממוצע כלומר המערכת שלנו טובה ונוחה בשימוש)

1. **הגדירו 3 מדדים להצלחת המערכת (הרצאה 3).(10 נקודות)**

**System availability (זמינות המערכת):**

* **הגדרה**: אחוז הזמן שבו המערכת פעילה וזמינה לשימוש, כלומר, המערכת לא נתקלת בתקלות או בעיות שמונעות את השימוש בה.

**מטרה**: לשמור על זמינות גבוהה של המערכת.

**Throughput (קיבולת/ קצב ביצוע):**

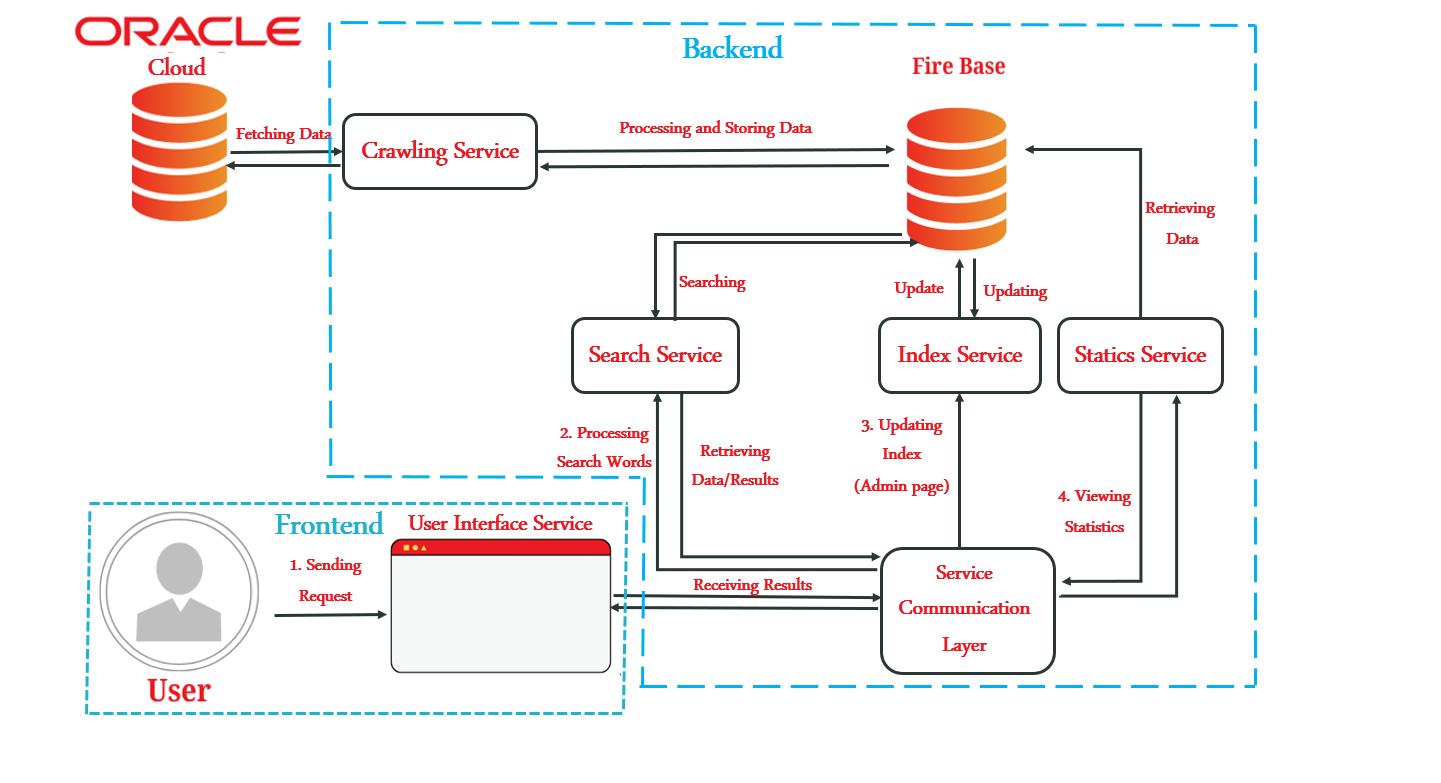
* **הגדרה**: כמות השאילתות או הבקשות שהמערכת יכולה לעבד בפרק זמן נתון, כלומר כמה חיפושים או פניות למנוע החיפוש המערכת יכולה לנהל בכל שנייה, דקה או שעה.

**מטרה**: להבטיח שהמערכת יכולה להתמודד עם כמות גדולה של חיפושים מבלי לפגוע בביצועים.

**Scalability (יכולת סקלביליות):**

* **הגדרה**: היכולת של המערכת להתמודד עם גידול בכמות המשתמשים, הנתונים או השאילתות מבלי להיתקל בירידה משמעותית בביצועים.

**מטרה**: המערכת צריכה להתרחב בקלות על מנת לתמוך במספר גדול יותר של משתמשים או יותר נתונים מבלי לפגוע במהירות החיפושים או בזמינות המערכת.

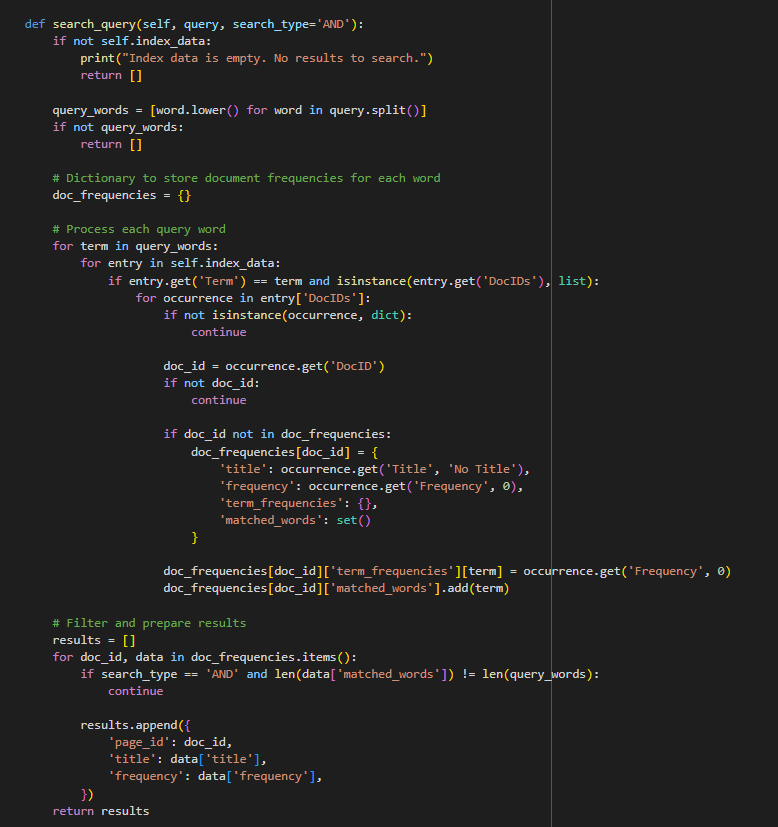
1. **** **הציגו דיאגרמת ארכיטקטורה של המערכת שלכם. הסבירו באיזה סוג ארכיטקטורה השתמשתם (הרצאה 7), ופרטו את חלקי הקוד המתייחסים לכל חלק בארכיטקטורה.(10 נקודות)**

בחרנו בסוג ארכיטקטורה : Microservices , אשר כל Service נוכל להפריד אותו ולהפעיל אותו לבד / אם משהו השתבש לא יפגע בשני.

**שירות חיפוש (Search Service)**

* שירות החיפוש אחראי על קבלת שאילתות מהמשתמש, ביצוע חיפושים במאגר נתונים (או אינדקסים) והחזרת התוצאות, השירות הזה פועל בנפרד ואינו תלוי בשירותים אחרים.

**החלק הרלוונטי בקוד:**



הפונקציה מבצעת חיפוש במאגר הנתונים/אינדקס, כדי להחזיר תוצאות חיפוש בהתאם.

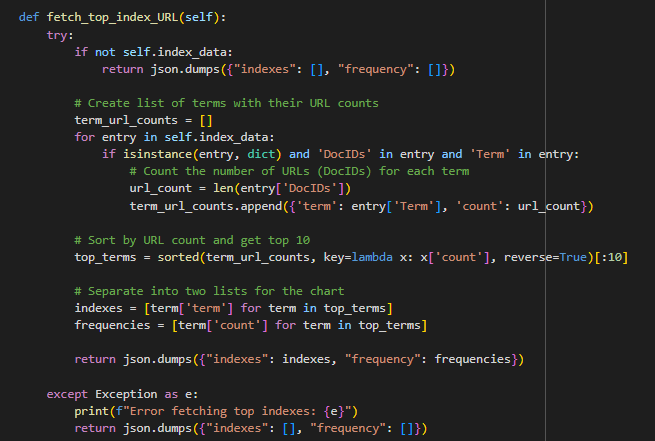
כל חיפוש מתבצע בנפרד ואינו תלוי בשירותים אחרים.

**שירות סטטיסטיקות (Statistics Service)**

החלק שאחראי על איסוף ועיבוד סטטיסטיקות.

* השירות הזה אחראי על הצגת נתונים סטטיסטיים של האינדקסים, כמו חיפושי המילים הנפוצות ביותר, כמות הקישורים לכל אינדקס.
* שירות זה מתפקד עצמאית ויודע להפיק דוחות וסטטיסטיקות על פי דרישות המשתמש.
* הוא פועל בנפרד ושומר מידע סטטיסטי על כל חיפוש.

**החלק הרלוונטי בקוד :**

  
  
  
הפונקציה אחראית להחזרת נתונים סטטיסטיים לגבי אינדקסים.

הסטטיסטיקות מבוססות על נתונים שנאספים ממספר שירותים ונתונים. כל שינוי בסטטיסטיקות מתעדכן באופן עצמאי ונשלח כתגובה לשירות הסטטיסטיקות.

**שירות ניהול אינדקסים (Index Service)**

השירות הזה אחראי על עדכון וניהול של האינדקסים, כולל הוספת קישורים או מחיקת קישורים לאינדקס ספציפי, ומחיקת אינדקסים.

שירות זה אחראי על ניהול האינדקסים. כל שינוי במידע דורש אינטראקציה עם שירות זה.

**החלק הרלוונטי בקוד:**

  
  
תמונה שמכילה טקסט, צילום מסך

התיאור נוצר באופן אוטומטי

הפונקציה update\_index מעדכנת /מוחקת קישורים של אינדקס ספציפי במערכת.  
הפונקציה delete\_index מוחקת אינדקס ספציפי מהמערכת.

**ממשק משתמש (User Interface Service)**

ממשק המשתמש מציג את הנתונים למשתמש וקולט ממנו את הקלט, אך הוא לא מבצע את העיבוד או החישוב עצמו. הוא מבצע קריאות לשירותי הלוגיקה דרך **Service Communication Layer** ומציג את התוצאות.(הממשק המשתמש מבצע את הצגת המידע למשתמש בצורה ויזואלית).  
בקוד -השירות **SearchUI** אחראית להפעיל את המסכים השונים.

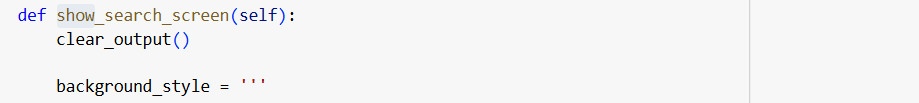
**חלקי הקוד הרלוונטיים:**

הממשק משתמש ב-ipywidgets או ספריות אחרות (כמו HTML) כדי ליצור ממשק אינטראקטיבי למשתמש. המשתמש מזין שאילתה והמערכת שולחת אותה ל Service Communication Layer.



* כל דף שמציג למשתמש את תוצאות החיפושים, הסטטיסטיקות או המנהל

כמו show\_search\_screen, show\_stats\_page, show\_admin\_page.



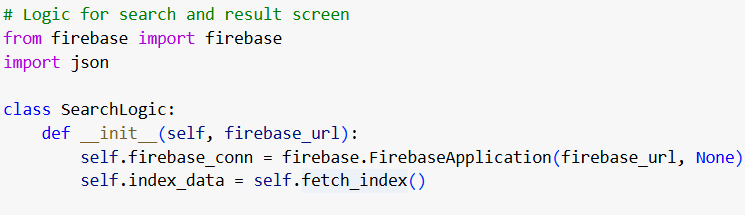
**Service Communication Layer**

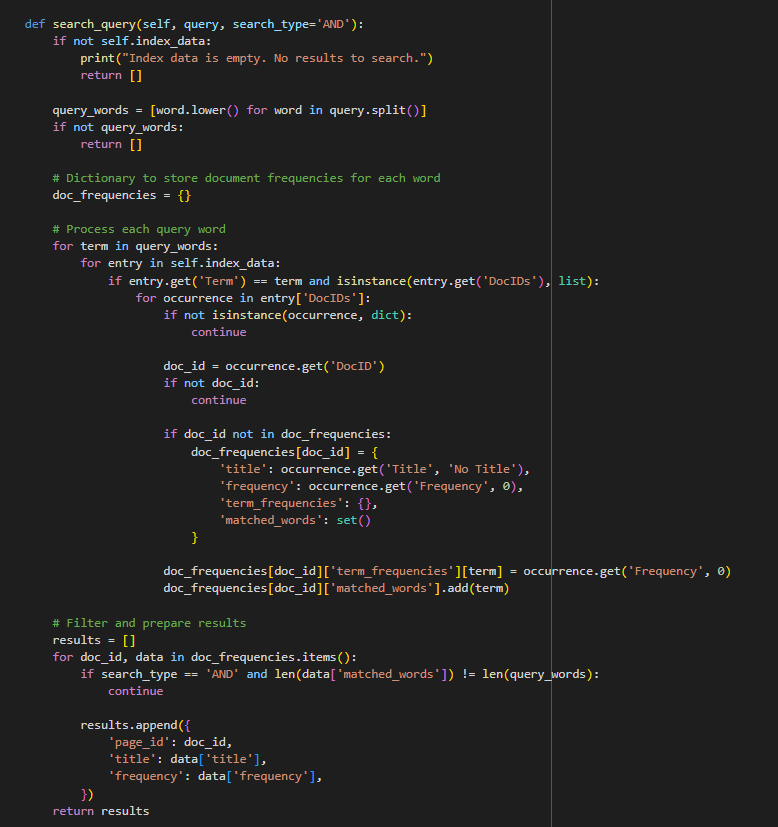
, Service Communication Layer מתפקדת כגשר בין User Interface Service לשירותים השונים, ומבצעת את הפעולות שבקשות המשתמש דורשות. היא מקבלת את הבקשות מממשק המשתמש, מפנה אותן לשירותים המתאימים.

כאשר השירות מחזיר תוצאות (כמו בתוצאה של חיפוש), Service Communication Layer מקבלת את התוצאה ומעבירה אותה חזרה לממשק, כך שהממשק יוכל להציג את התוצאה למשתמש.

בקוד -השירות**SearchLogic** אחראית להפעיל אותם ולהחזיר את התוצאה לממשק

**לדוגמה :**





הוא מבצע את הקריאה המתאימה לשירות הלוגיקה כמו search\_query(self,query) ב**SearchLogic** ומחזיר את התוצאה לממשק - show\_search\_screen.

*חלק שלישי : פיצ'ר לבחירתכם* (10 נקודות)

הוסיפו פיצ'ר מעניין למערכת, אשר לא נדרש מכם, לבחירתכם.שימו לב - הכוונה לפיצ'ר פונקציונאלי (ולא עיצובי, או שימוש בשרת כדי להעלות לענן את האתר).

כתבו מספר משפטים להסבר התוספת, וציינו היכן בקוד הוא ממומש. כמו כן הסבירו כיצד הוא מתבטא בחלק המוצג למנהל.

הוספנו למערכת פיצ'ר של נגן שירים המאפשר לנגן קבצי MP3 ישירות מקישורים באינטרנט.   
בפרויקט שלנו, השירים נשמרו במאגר GitHub Repository. פיצ'ר זה נבחר מתוך מטרה לשפר את חוויית המשתמש ולהעניק ערך מוסף למערכת.

מימוש הפיצ'ר מבוצע בשני חלקים מרכזיים:  
  
**במסך האדמין - "Music Player Management":**  
  
המנהל יכול לנהל את רשימת השירים: להוסיף קישורים, לעדכן את הרשימה הקיימת, או להסיר שירים.

פונקציה load\_songs\_list טוענת את רשימת השירים מקובץ JSON. אם הקובץ אינו קיים, נוצרת רשימת שירים ראשונית.

פונקציה save\_songs\_list שומרת את הרשימה המעודכנת בקובץ JSON, המאפשר למערכת לשמור על עקביות ועדכניות.

פונקציה create\_music\_tab מציגה ממשק נוח לניהול השירים, כולל שדות קלט לכיוון הקישורים ובדיקות תקינות (כגון ווידוא שקובצי MP3 וקישורים מ-GitHub).

**בנגן השירים עצמו:**נגן המוזיקה מאפשר למשתמשים להאזין לשירים דרך ממשק מעוצב.

פונקציה load\_songs טוענת את רשימת השירים בקובץ JSON. אם הרשימה ריקה או הקובץ פגום, מופיעה הודעה מתאימה. השירים מתעדכנים אוטומטית במקרה של שינוי על ידי המנהל.

פונקציה create\_mini\_player יוצרת את ממשק הנגן, כולל כפתורי שליטה (הפעלה, עצירה, דילוג על שירים, חזרה על שיר). כמו כן, הממשק מציג את שם השיר שמתנגן ואת התקדמות השמעתו. **סיכום:**

פיצ'ר זה משדרג את המערכת בכך שהוא מאפשר למנהל שליטה גמישה על תוכן המוזיקה, ומציע חוויית שימוש ידידותית ומותאמת אישית למשתמשים.   
המערכת שומרת על עדכניות ונוחות שימוש באמצעות שילוב ממשקים אינטראקטיביים וניהול נתונים יעיל.

קישור למחברת ב- Collab - <https://colab.research.google.com/drive/1Fe2OLHA6tG9G5VNz7tNGu7g9uTGM-xtR#scrollTo=gsPZHUTM0PyM>

קישור ל-GitHub - <https://github.com/muffindonor/cloud_computing_project.git>

**הוראות הגשה:**

1.ש להגיש במודל קובץ זיפ הכולל קובץ וורד ובו מענה לשאלות, וקישור ל- notebook ובו הקוד שלכם (יש לוודא שהקישור פומבי ונגיש). **אין לבצע שינויים במחברת לאחר ההגשה!**

**2.** הקוד צריך לרוץ במלואו מהמחברת בלבד. לא יתקבלו הגשות הכוללות הרצה באתר חיצוני (בפרט slack), או צורך להעלות קבצים למחברת על מנת שתרוץ. הגשות כאלו יקבלו ציון אפס על מרכיב הקוד

3.יש להגיש את התרגיל בצוותים, בתיקיית ה –GIT שלכם (צרפו קישור), וכן בתיקייית התרגיל ב moodle.כותרתו של הקובץ תהיה HW2\_TEAMNAME

4. שימו לב כי כל העבודות חייבות להיות שונות זו מזו. עבודות שייראו דומות ייפסלו ויינתן עליהן ציון 0.

בהצלחה!