**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

קישור למחברת: <https://colab.research.google.com/drive/1eyWxa6l8drIVKr5AAdWI0ZNjXfPyUGEh?usp=sharing>

קישור ל-github: <https://github.com/ShlomiFridman/PhoenixProject2025>

*חלק ראשון : בניית אינדקס (40 נקודות)*

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

*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** |
| *כפיר אמויאל (UI)* | *Design* | *Design* |
| *שלומי פרידמן (scrum master)* | *Firebase, Crawler, and Index services* | *Firebase, Crawler, and Index services* |
| *שחר ברנסון (backend developer)* | *Search form, result handler* | *Search form, result handler* |
| *עומר גולדשטיין (frontend developer)* | *Query service* | *Query service* |
| *ישראל אוחיון (product manager)* | *Graphs* | *Graphs* |
| *ולדי טריטנר (QA)* | *Testing* | *Testing* |

בניית אינדקס (30 נקודות)

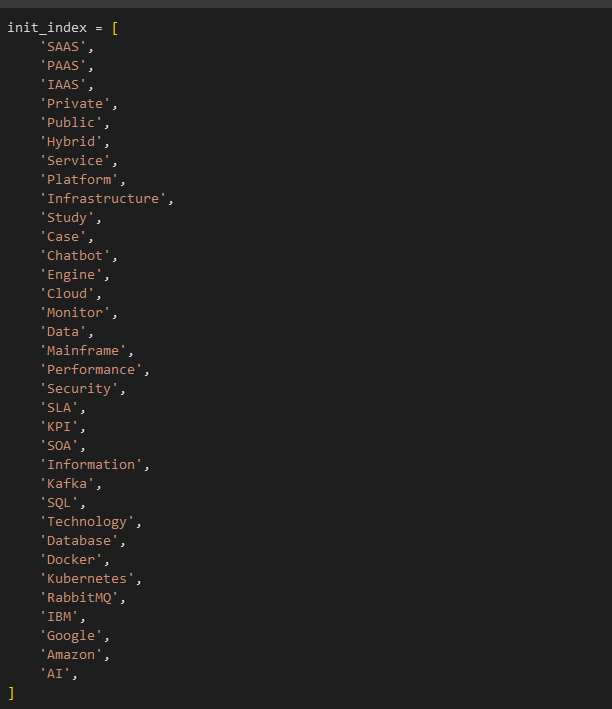
באיטרציה זו עליכם לבנות את מסד הנתונים שמכיל את האינדקס של המילים המשמעותיות באתר אותו אתם חוקרים.

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.

המילים שהוכנסו לאינדקס (לפני Stemming ועיבוד)



*חלק שני: בניית מסכים להצגה בכיתה (50 נקודות)*

בחלק זה תכינו 4 מסכים, אותם תציגו לחבריכם בפעילות שתתבצע בכיתה.

המסכים צריכים לכלול (לפחות):

מסך מנהל לעריכת אינדקס, מסך הזנת שאילתא, מסך תוצאות שאילתא, מסך סטטיסטיקות מעניינות.

בשלב זה נדרש לממש במלואם את בניית מסך המנהל ומסך הזנת השאילתא. ניתן את התוצאות להציג כרגע עם data מהעמוד הראשי של האתר שאתם עובדים עימו.  
את מסך המנהל ומסך הסטטיסטיקות עליכם לממש בצורה בסיסית. עם זאת מומלץ לממש ככל הניתן גם את ניהול האינדקס וסריקת האתר , כפי שלמדתם בתרגול 6 😊

כמובן ,שכל תוספת שתחליטו עליה, תוביל להערכה גבוהה יותר של המשימה.

בשבוע ההרצאות של 6-9.1.25 תציגו את המערכות שבניתם. המפגש יתנהל במתכונת סטודיו – כל צוות מגיעה **במלואו** לאחד המועדים עם לפטופ, כל הצותים מציגים במקביל. הסטודנטים מסתובבים בין הצוותים, **מתנסים** במערכת,וממלאים משוב .

ההצגה ומילוי המשוב הם **חובה.**

אנא השתבצו בהקדם לאחת מקבוצות ההרצאה על מנת לוודא שקיים איזון בין הצוותים:

<https://docs.google.com/spreadsheets/d/11Lyni94IKg1O5Py2t3mIL1QhITCnJCMV_-MB_Ey73NE/edit?gid=0#gid=0>

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

1. יש להגיש את הטבלה הבאה , תוך התיחסות למשובים שקיבלתם (10 נקודות):

|  |  |  |
| --- | --- | --- |
| **הערת משוב** | **האם התבצע שינוי באפליקציה בעקבות ההערה?** | **נימוק** |
| לשפר את העיצוב | כן | למרות שבדרישות שלנו יש דרישה לעיצוב מינימליסטי נשפר את העיצוב (כגון צבעים, כפתורים וכו') עבור המוצר הסופי. |
| זמן תפעול מהיר יותר \ יותר פירוט | כן | נוסיף טקסט שלמשתמש יהיה יותר ברור איך המערכת עובדת. |
| הכנסת כמות דפים לקרולר כפרמטר | לא | כבר יש פיצ'ר להוספת דף חדש במסך עריכת האינדקס. |
| הצגת האינדקסים במערכת בפלט החיפוש | לא | אנו לא רוצים לחשוף את האינדקס שלנו במסך החיפוש, על מנת שלא יהיה עומס קוגנטיבי על המשתמש, המשתמש יוכל לראות את האינדקס במסך עריכת האינדקס, ולקבל מידע עליו שם ובמסכי סטטיסטיקה. |
| זמני ריצת החיפוש, אולי כדאי לייעל את החיפוש מראש ולשמור בDB | לא | כבר מוצג למשתמש זמן החיפוש במסך התוצאות, ואנו לא רוצים לשמור את התוצאות בבסיס נתונים כדי למנוע עומס (לאחר בדיקה ה-Firebase לא אוהב את זה) בנוסף האינדקס יוכל להתעדכן בהמשך (בין במילים או בדפים חדשים \ נוספים) ולכן אנו מעדיפים שעבור כל שאילתה יהיה חיפוש חדש ועדכני |
| יש לשפר את האינדקס חיפוש | כן | נוסיף מילים חדשות לאינדקס |
| הגדל הכתב | לא | בדפדפן יש אפשרות מובנת להגדלת הכתב, לכן אין צורך לעשות את זה במערכת עצמה לאחר בדיקה עם חברי הצוות וסטודנטים אחרים, נמצא שגודל הכתב הנוכחי נוח וקל לעין |
| הגרף עם המספרים היה קצת מסורבל ללא הסבר מפורמט מהאחראי של האתר | כן | נוסיף הסבר לגרפים |

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

* Consistency – העיצוב של המערכת עקבי, עם דגש על עיצוב מינימליסטי וצבעים קלים על העין.
* Visibility – אנו מציגים את התוצאות החיפוש למשתמש בצורה ברורה, עם מספור של התוצאות על מנת שיידע איזו תוצאה קיבלה מקום ראשון.
  + בהמשך נוסיף גם פירוט נוסיף שהמשתמש יבין למה היא קיבלה את המיקום שלה.
  + לאחר קבלת המשוב הבנו שהגרפים שיש לנו כרגע במערכת לא כל כך ברורים ולכן כשלו בכלל ה-visibility, נרצה להוסיף להם פירוט, ושיהיו יותר ידידותיים למשתמש במוצר הסופי.
* Feedback – עבור כל פעולה שהמשתמש עושה במערכת (בין אם זה חיפוש, עריכת אינדקס וכו') הוא מקבל משוב מהערכת כגון תוצאות חיפוש, הודעה שהאינדקס עודכן, וכו'.
* Affordance – מוצג למשתמש תפריט במסך עריכת האינדקס עם טקסט שמנחה אותו מה אפשר לעשות ואיך. במסך החיפוש גם מוצג טקסט שמתאר לו שצריך שהוא צריך להכניס קלט לתיבת החיפוש, וכפתורים עם טקסט בכל אחד שמדריך אותו מה הם עושים.
* Simplicity – עיצבנו את המערכת עם עיצוב מינימליסטי, מסך תוצאות ברור עם עד 10 תוצאות בעמוד על מנת למנוע עומס קוגניטיבי על המשתמש.
* Error Prevention – כאשר מתבצעת פעולה בעמוד עריכת האינדקס אנו "נועלים" את הכפתורים והקלט עד שהפעולה מסתיימת, בשביל למנוע שגיאות. ואם המשתמש מכניס קלט לא תקין אנו מציגים לו הודעה בהתאם.
* Flexibility – פעולת החיפוש תומכת בחיפוש באופרטורים ללוגים (OR ו-AND) מה שמאפשר למשתמש גמישות בחיפוש, וה-chatbot הנוכחי שמציג מידע על מושגים תומך כמה pattern-ים שונים עובר בקשת הסבר על מושג.
* Aesthetic and Minimalist Design – אנו משתמשים במעט צבעים, ובעיקר בצבעים בהירים. עיצוב המסך חיפוש והתוצאות הוא מינימלי על להקל על המשתמש. והתוצאות מחולקות לעמודים על מנת שלא יהיה עליו עומס של מידע.

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

לקחנו את ממוצע התוצאות במשוב ה-SUS מהסדנא (את התוצאות של השאלות עבור משוב שלילי הפכנו), הסכום של הממוצעים יצא 35.86. ונחשב ציון:

*הציון של המערכת שלנו לפי המשוב מהסדנא הוא 72 מתוך מאה.*

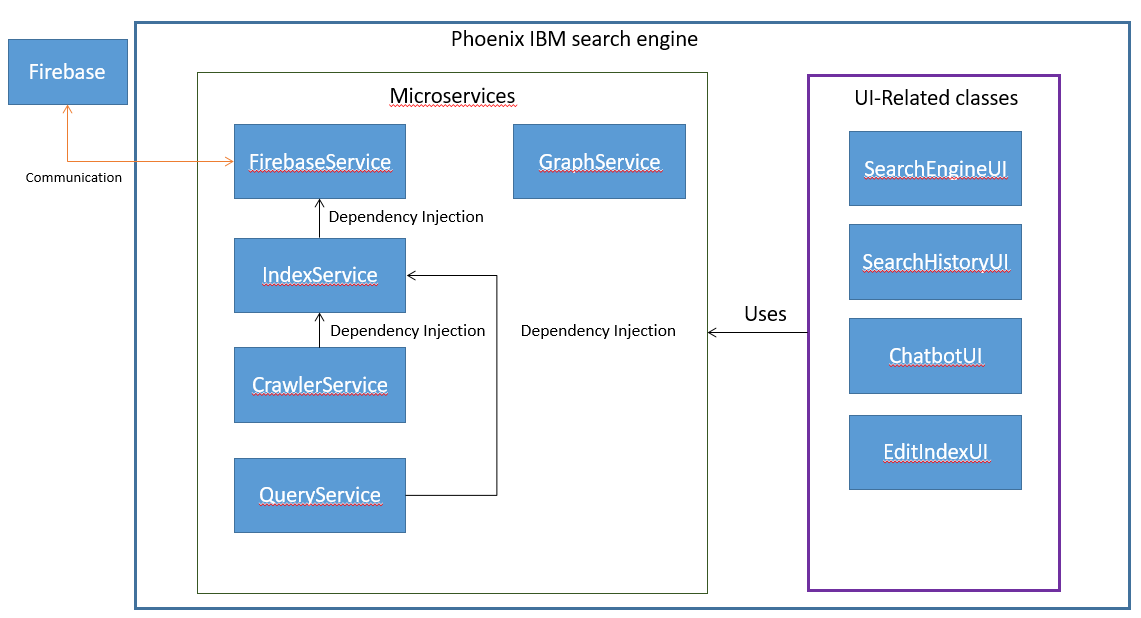
* המשך בעמוד הבא.

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

מדדי KPI לפרויקט:

* User Satisfaction
  + שהמשתמש יהיה מרוצה מתוצאת החיפוש, כלומר שהתוצאות חיפוש יכילו תוצאות נכונות.
  + מדידה: שתהיה לפחות תוצאה אחד נכונה \ רלוונטית ב-5 התוצאות הראשונות.
* Performance
  + שהמשתמש לא ימתין יותר מדי זמן לתוצאת החיפוש.
  + מדידה: אחרי לחיצה על כפתור החיפוש לא יעברו יותר מ-3 שניות עד שתוצג תוצאה כלשהי.
* Scalability & Elasticity
  + שהמערכת תהיה מודולרית, וקלה להרחבה.
  + מדידה: פיתוח מודולרי, שכל פונקציה ומחלקה תהיה עצמאית, כלומר בלי תלות במשתנים גלובלים.

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



השתמשנו בארכיטקטורה של microservices, לקחנו את הפונקציונליות העיקרית ויצרנו מהן מחלקות.

* + כל הקוד של הפונקציונליות ה-"כבדה" בערכת כגון של האינדקס, ה-crawler ושל השאילתות. הופרד זה מזה ויצרנו מהן service-ים עם תלויות מסוג ה-dependency-injections. ובכך יצרנו פתוח יותר מודולרי במערכת.
  + כל הפונקציונליות שקשורה למסכים הועברה למחלקות שמופרדות זו מזו הנעזרות בשירותים של המערכת.

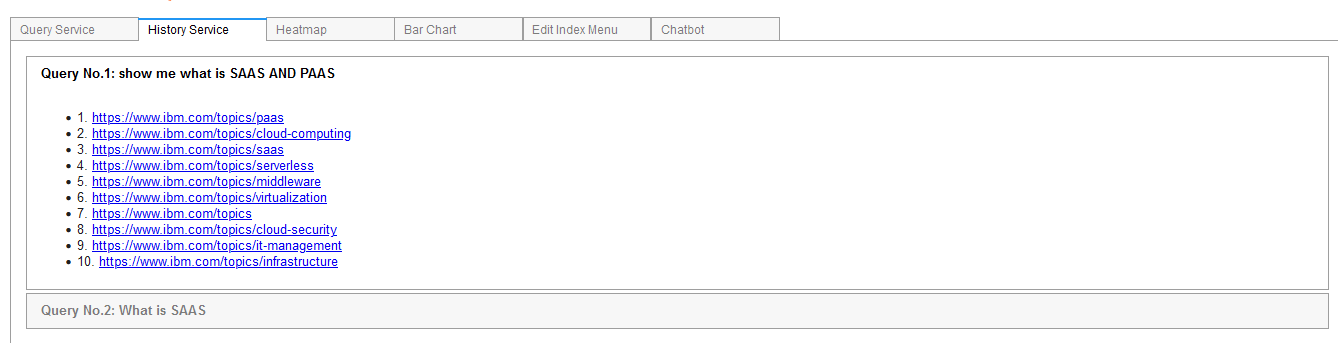
*חלק שלישי : פיצ'ר לבחירתכם* (10 נקודות)

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).

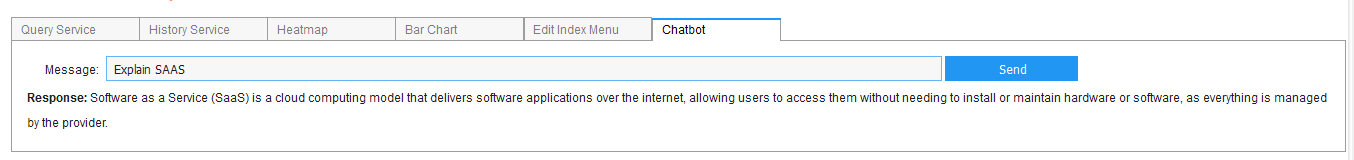
כתבו מספר משפטים להסבר התוספת, וציינו היכן בקוד הוא ממומש. כמו כן הסבירו כיצד הוא מתבטא בחלק המוצג למנהל.

הפיצ'רים שהוספנו:

* שמירת והצגת ההיסטורית חיפוש למשתמש – אנו נותנים למשתמש אפשרות לראות את ההיסטורית חיפוש שלו, עם התוצאות שהתקבלו עבור אותה שאילתה.
  + ההצגה של ההיסטוריה ממומשת על ידי SearchHistoryUI שמציג אותה ועל ידי QueryService ששומר אותה. ניתן לראות את ההיסטוריה ב-tab של History Service:



* Chatbot – מימשנו chatbot פשוט שיכול לתת למשתמש הסבר על מושגים הקשורים לנושא ה-cloud. כרגע ה-chatbot מכיר רק כמות קטנה של patter-ים, אבל השאיפה שלנו למוצר הסופי הוא chatbot יותר חכם עם שילוב של בינה מלכותית חיצונית (כגון genai).
  + המימוש של ה-chatbot הוא ב-ChatbotUI (כרגע על ידי חבילת nltk) וניתן לראות אותו ב-tab של Chatbot



**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!**

[**https://colab.research.google.com/drive/1eyWxa6l8drIVKr5AAdWI0ZNjXfPyUGEh?usp=sharing**](https://colab.research.google.com/drive/1eyWxa6l8drIVKr5AAdWI0ZNjXfPyUGEh?usp=sharing)

**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

<https://github.com/ShlomiFridman/PhoenixProject2025>

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

בהצלחה!