**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.

*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*

|  |  |  |
| --- | --- | --- |
| איטרציה 1 | | |
| שם חבר הצוות ותפקיד בתרגיל זה | משימות שהוקצו | משימות שהושלמו |
| אלכס סגל – scrum master | Building backend (chatbot + microservices), integration, code review | Building backend (chatbot + microservices), integration, code review |
| דוד קוטלר - frontend | Building frontend (Admin screen), integration | Building frontend (Admin screen), integration |
| ניר פרוימוביץ' - backend | Building backend (index+ chatbot + microservices), integration | Building backend (index+ chatbot + microservices), integration |
| אליו פיקולי – product manager | Building frontend, architecture, SUS, review, integration | Building frontend, architecture, SUS, review, integration |
| דייב רויטברג - UI | Building frontend (Main + statistics + Admin screens), integration | Building frontend (Main + statistics + Admin screens), integration |
| מקסים לזרנקו - QA | Building frontend (Main screen), integration | Building frontend (Main screen), integration |

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.

*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 |
| עיצוב | כן | עיצוב מחדש של המסכים |
| חלוקה נכונה ואיחוד כל המסכים לדף אחד | כן | פיצלנו את חלקי המערכת לBackend וFrontend מובהקים |
| ממשק משתמש | כן | ממשק המשתמש עוצב מחדש |
| אדמיןGUI | כן | נבנה |
| הוספת פיצ׳רים | כן | מומש פיצ׳ר |
| הוספת לוגו | כן | הלוגו היה קיים אך הייתה בעיה בהצגתו, הבעיה נפתרה |
| שיפור סטטיסטיקות והצגות המידע | כן | שופרה נראות המסכים |
| ותבנית הפיירבייסDBשיפור ה | כן | הוספת עדיפות לחיפוש במסד נתונים על פני הפעלת זחלן |

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

כללי הזהב של שניידרמן:

* 1. חתירה לעקביות – המסכים השונים שלנו מעוצבים באותו אופן כדי לשמור על עקביות בנראות ובנוחות הצפייה של המשתמש במסכים.
  2. לאפשר שימוש בקיצורי דרך – כרגע המערכת תומכת בחיפוש מקוצר: אחרי חיפוש ראשוני המערכת שומרת רשימה מקוצרת של החיפושים האחרונים ומציגה למשתמש.
  3. הצע משוב אינפורמטיבי – קיים גלגל חשיבה מגניב המראה בזמן טעינת הלינקים את סטטוס טעינתם.
  4. עיצוב דיאלוגים ברורים – עיצוב של שורת חיפוש ועיצוב של מסך אדמין.
  5. מנע משגיאות – אין. (בעתיד הוספת פונקציונליות שבה אם המשתמש עשה איזושהי טעות בהכנסת נתונים המערכת מתריעה על כך בתוספת איזה נתונים מנע מהמשתמש להתקדם)
  6. אפשר היפוך קל של פעולות – קיימים כפתורי חזרה בין המסכים השונים.
  7. מוקד השליטה אצל המשתמש – המשתמש יכול לחפש בכל רגע (ממסך החיפוש וגם מסך התוצאות) terms חדשים כרצונו.
  8. הפחתת עומס הזיכרון לטווח קצר – קיים תקציר מתחת לכל לינק המתאר בקצרה את הנאמר בכתבה.

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

|  |
| --- |
| **50** |
| **70** |
| **95** |
| **50** |
| **82.5** |
| **100** |
| **60** |
| **70** |
| **100** |
| **85** |
| **65** |
| **87.5** |
| **85** |
| **42.5** |
| **87.5** |
| **52.5** |
| **95** |
| **75** |
| **97.5** |
| **85** |
| **100** |
| **92.5** |
| **82.5** |
| **87.5** |

**ממוצע = 79.0625**

לפי טבלת הציונים, ציון בטווח 70-84 יקרא ציון טוב, כלומר המערכת קלה לשימוש אך ניתן לשפר אותה בעקבות המידע מהמשתמשים.

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

**א. זמינות המערכת:**  
המערכת תהיה זמינה למשתמשים כל עוד שירות הענן עליו אנו מסתמכים (כגון Google Cloud או שירותי Firebase) פעיל וללא תקלות. במקרים נדירים של השבתה של שירות הענן, ייתכן שזמינות המערכת תושפע בהתאם. עם זאת, אנו שואפים לשמור על זמינות מקסימלית על ידי שימוש במערכות ניטור והתראות כדי לזהות ולטפל בתקלות במהירות.

**ב. זמן תגובה:**  
המערכת תוכננה כך שתדע לנתב שאילתות בצורה חכמה ולהשתמש במידע המאוחסן במסד הנתונים המקומי או לשלוח את הזחלן (crawler) לחיפוש מידע חדש. זמן התגובה הממוצע בפעולות רגילות צפוי להיות מהיר, בין 1-3 שניות, עבור שאילתות קיימות במסד הנתונים.  
עם זאת, במקרה שבו יש צורך בהפעלת הזחלן לחיפוש מידע חדש, זמן התגובה עשוי להתארך, ויגיע עד 10 שניות, בהתאם למורכבות החיפוש ולמצב רשת האינטרנט.

**ג. עיכוב:**  
העיכוב במערכת מושפע ישירות מהעיכוב בשירות הענן עליו אנו מסתמכים. במקרה של Firebase, ייתכן עיכוב ממוצע של 50-300 מילישניות, בהתאם למיקום הגיאוגרפי של המשתמש ולמצב השרתים.  
למרות זאת, המערכת נבנתה כדי למזער עיכובים אלו באמצעות שימוש בקאשינג (caching) מקומי ובפריסה של שרתים ברחבי העולם כדי לשפר את הביצועים עבור משתמשים באזורים שונים.

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

Frontend

Backend

Indexer

Check terms

Add/Delete/View terms

Statistics page

Search page

Admin page

Chatbot back

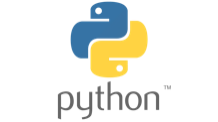
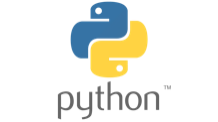
Sentence splitter

Initialize search engine

About Us page

Gemini Shark Bot







BeautifulSoup

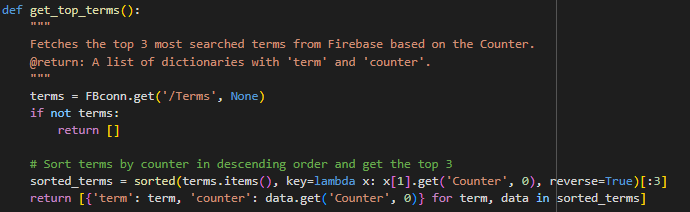
ipywidgets

*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.

פיצ'ר שלנו מציג על מסך חיפוש 3 המילים הכי נפוצות ומאפשר ליוזר לחפש את המילים האלא בלחיצה עליהם.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

**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!**

קישור למחברת:  
  
<https://colab.research.google.com/drive/1f5UAqpDNYXJbFwUnW_VbIEhH_0FWzJq3?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

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

בהצלחה!