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

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

מועד הגשה: 12.1.25

קישור לGoogle Notebook  
<https://colab.research.google.com/drive/1SPOVQE4s7Atnvl_oRxLWYqmsfHc15GSq?usp=sharing>  
קישור לGITHUB

<https://github.com/TRiZKy/CloudAccessBear>

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*

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** |
| מעיין עציוני  מהנדסת מערכת  scrum master | מדדים להצלחת המערכת, מימוש המערכת, חלוקת המשימות | מדדים להצלחת המערכת, מימוש המערכת |
| רוני ברין  frontend developer | בחירת פיצ'ר נוסף, עיצוב מסכים ונגישות משתמש. | בחירת פיצ'ר נוסף, עיצוב מסכים ונגישות משתמש. |
| טל תורגמן  backend developer | מימוש FireBase | מימוש FireBase |
| ינון לוי  product manager | ייצוג החשיבה עיצובית שהוגדרה וייצוג חשיבת הלקוח במסכים | ייצוג החשיבה עיצובית שהוגדרה וייצוג חשיבת הלקוח במסכים |
| פבלו דובצ'ינסקי  QA | בחירת פיצ'ר נוסף, בדיקות המערכת | בחירת פיצ'ר נוסף, בדיקות המערכת |
| שי ביסטריצקי  UI | מימוש חשיבה עיצובית במערכת | מימוש חשיבה עיצובית במערכת |

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 |
| אינטראקטיביות וחווית משתמש | לא | הממשק בנוי על Dash שמאפשר חווית משתמש נוחה ואינטראקטיבית, עם שימוש ב-Callbacks לתגובות מיידיות. |
| דרך הצגת המידע | לא | פונקציה כמוgenerate\_result\_cards יוצרת תצוגה מובנית ומסודרת לתוצאות החיפוש. |
| עיצוב מזמין יותר | לא | הממשק מעוצב עם אלמנטים בסיסיים אך מסודרים היטב, כמו כפתורים ותפריטים פשוטים. |
| אופציה לראות את רשימת האינדקסים | כן | הוסף מסך בו ניתן לראות את רשימת האינדקסים. |

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

**8 כללי הזהב של שניידרמן:**

**עקביות:**

המערכת משתמשת בתבניות עיצוביות עקביות בממשקים, כמו טופסי חיפוש, כפתורים ומסכים. התפריטים והקישורים נשארים במקומם בכל עמודי המערכת.

**פעולות קצרות ופשוטות:**

חיפוש ושימוש במערכת מתבצעים בזריזות. לדוגמה, הפונקציה search\_index מאפשרת חיפוש ממוקד ומהיר של מונחים.

**משוב מיידי:**

יש משוב ברור בכל שלב. לדוגמה, תוצאות חיפוש מופיעות מייד לאחר ביצוע השאילתה.

**מניעת טעויות:**

המערכת כוללת מנגנונים למניעת טעויות, כמו בדיקת תקינות של קלטים (למשל שימוש בפונקציה autocorrect\_query לתיקון טעויות הקלדה).

**תמיכה בתיקון טעויות:**

המערכת מספקת אפשרות לתקן שגיאות באמצעות הצעת תיקון אוטומטי לחיפושים שגויים.

**שליטה גמישה למשתמש:**

המערכת מתאימה למשתמשים ברמות ידע שונות. לדוגמה, ממשק המשתמש כולל כפתורי חיפוש והצעות מתקדמות.

**צמצום עומס קוגניטיבי:**

המערכת מציגה למשתמש רק את המידע הרלוונטי בכל שלב, לדוגמה, עמוד תוצאות חיפוש שמציג מידע בצורה ממוקדת ונקייה.

**תחושת סיפוק למשתמש:**

ממשק משתמש ידידותי, עיצוב מודרני ומשוב חיובי תורמים לחוויית שימוש נעימה.

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

**ציון ה-SUS שהתקבל משאלון ה-SUS על המערכת שלנו הינו: 86  
-**מעיד על כך שהמערכת נתפסת כקלה לשימוש, והמשתמשים ככל הנראה חווים שביעות רצון גבוהה מהשימוש בה.

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

**מדדים להצלחת המערכת:**

* **חוויית משתמש מיטבית  
  זמני תגובה מהירים:** המערכת מגיבה לשאילתות חיפוש ותצוגת תוצאות בזמן קצר, גם בעומסים גבוהים.  
  **ממשק משתמש ידידותי:** המשתמשים יכולים לבצע חיפושים ולנווט במערכת בקלות, ללא צורך בהסברים או הדרכה**.**
* **יכולת עיבוד נתונים יעילה  
  תמיכה בנתונים רב-שכבתיים:** המערכת מסוגלת להתמודד עם קבצי JSON מורכבים ולהציגם בצורה ברורה למשתמש.  
  **עדכון אינדקס אוטומטי:** המערכת מעדכנת את האינדקס בזמן אמת לאחר הוספת נתונים חדשים, כך שתוצאות החיפוש תמיד מעודכנות.
* **תוצאות חיפוש מדויקות ורלוונטיות**

**דיוק גבוה:** התוצאות שמוצגות למשתמש עונות במדויק על השאילתה, כולל הרחבות למילים נרדפות ודומות.

**רלוונטיות מותאמת:** תוצאות החיפוש מסודרות על פי מידת התאמתן לשאילתה, עם דגש על התאמות המבוססות על מילים מורחבות.

1. הציגו דיאגרמת ארכיטקטורה של המערכת שלכם. הסבירו באיזה סוג ארכיטקטורה השתמשתם (הרצאה 7), ופרטו את חלקי הקוד ההמתיחסים לכל חלק בארכיטקטורה.(10 נקודות)  
     
   **המערכת משתמשת בתבנית ארכיטקטורה שכבתית (Layered Architecture Pattern)**.

**שכבות המערכת:**

**שכבת המצגת (Presentation Layer - UI Layer)**

* **רכיבים:**
  + אפליקציית Dash (app.layout): מיישמת ממשקי משתמש לחיפוש, צ'אטבוט והצגת סטטיסטיקות.
  + עמודים מותאמים לפעילויות שונות: חיפוש, ניהול, צ'אטבוט וסטטיסטיקות.
* **קוד רלוונטי:**
  + הגדרות של search\_page, manager\_page, chatbot\_page, statistics\_page.

**שכבת היישום (Application Layer - Service Layer)**

* **רכיבים:**
  + פונקציות Callback ב-Dash: מטפלות באינטראקציות המשתמש, בניהול מצב, ובהפעלת חישובים (לדוגמה, חיפוש, תגובות צ'אטבוט).
* **קוד רלוונטי:**
  + הגדרות של @app.callback עבור עמודים ופונקציונליות שונות.

**שכבת הלוגיקה העסקית (Business Logic Layer - Domain Layer)**

* **רכיבים:**
  + פונקציות לאיסוף נתונים, ניתוחם ועיבודם.
  + לוגיקת חיפוש ובניית אינדקס.
  + הרחבת שאילתות ותיקון אוטומטי של שאילתות.
* **קוד רלוונטי:**
  + פונקציות כמו: fetch\_pages, extract\_data, build\_index, search\_index, expand\_query\_with\_synonyms, autocorrect\_query.

**שכבת גישה לנתונים (Data Access Layer - Persistence Layer)**

* **רכיבים:**
  + אינטגרציה עם Firebase לאחסון ושליפת נתונים.
* **קוד רלוונטי:**
  + יוזמה ושימוש ב-firebase.FirebaseApplication דרך FBconn.post.

*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 שהוספנו:**

**זיהוי מילים נרדפות בחיפוש**

הוספנו למערכת פיצ'ר שמרחיב שאילתות חיפוש באמצעות מילים נרדפות ודומות. כך, משתמש יכול לחפש "mobile" ולקבל תוצאות הכוללות "smartphone", "cellphone", ועוד. הפיצ'ר משפר אתחוויית החיפוש על ידי התאמת התוצאות גם אם המשתמש לא השתמש במילים המדויקות.   
הפיצ'ר ממומש על ידי הפונקציה **expand\_query\_with\_synonyms** שמרחיבה את השאילתה על בסיס מילון של מילים נרדפות ומשולב בפונקציית החיפוש **search\_index** שתרחיב את החיפוש למוצרים התואמים למילים הנרדפות, כך שמשתמש יקבל את מירב התוצאות המתאימות לחיפושו.בממשק ניהול האינדקס, המנהל יכול לצפות, לערוך ולהוסיף מילים נרדפות, מה שמאפשר לו לשלוט בהשפעת הפיצ'ר על תוצאות החיפוש.

**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!