מבוא למחשוב ענן - סמסטר אביב התשפ"ה

תרגיל בית 2 -– עבודה בצוותי העבודה

מועד הגשה: 28.5.25

המשימה בתרגיל זה: בניית מסכים מרכזיים במערכת

שימו לב: למטלה זו שלושה חלקים

קישור לגיט: <https://github.com/tomerroll/CloudTeam>

חלק ראשון : בניית מערכת (10 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* - ניהול העבודה, ביצוע בקרה על סטטוס משימות וקצב התקדמות.

*,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 |
| טום ביטון – scrum master | ארגון ישיבות צוות, מעקב אחרי קצב ההתקדמות, חלוקת משימות בין חברי הקבוצה. | * בוצעה פגישת עבודה וחלוקת משימות בין חברי הצוות. * בוצעה פגישה נוספת של סטטוס עבודה והצגת פערים |
| ארד הרוש – product manager | ניתוח משובים , בדיקת התאמה ללקוח. | * ניתוח משובים ותיקון האתר בהתאם. |
| יהלי רפפורט – frontend developer | בניית מסכים, הצגת נתונים מהשרת. | * בנייה של 4 מסכים * הצגת נתונים מהשרת |
| לידור בן חמו – backend developer | בניית מסד נתונים, מימוש אינדקס למילים משמעותיות. | * בנייה של מסד נתונים * מימוש אינדקס + הוספה של שדות רלוונטיים |
| תומר רול - UI | עיצוב מסכי ממשק משתמש, הדרכת הfrontend לעיצוב על פי עקרונות העיצוב. | * עיצוב ארבעת מסכי המשתמש * בקרה לפי עקרונות העיצוב שנלמדו |

בניית אינדקס (20 נקודות)

באיטרציה זו עליכם לבנות את מסד הנתונים שמכיל את האינדקס של המילים המשמעותיות באתר [mqtt.org](http://mqtt.org/), הפרוטוקול איתו אנו עובדים לקבלת מידע מהחיישנים.

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 |

משימות:

1. ממשו את האינדקס בקולאב.  
 2.רשמו בצורה מפורשת את רשימת ה stop words שבחרתם, ונמקו מדוע בחרתם במילים אלו.  
 3. ציינו האם השתמשתם ב stem/lemmatization לצורך בניית האינדקס.

תשובות:  
1.

Colab:

[Index-HW2.ipynb](https://colab.research.google.com/drive/1_fGyc3n3bC27qcu6TgpCM9UXs3Ax0G_D?usp=sharing)

DataBase Json:

<https://cloudhm2-default-rtdb.firebaseio.com/mqtt_index.json>

2.

['the', 'is', 'at', 'which', 'on', 'and', 'a', 'an', 'to', 'for', 'of',

'with', 'by', 'from', 'in', 'this', 'that', 'it', 'as', 'be', 'are',

'was', 'were', 'has', 'have', 'had', 'but', 'not', 'or', 'if', 'then',

'else', 'so', 'than', 'too', 'very', 'can', 'will', 'just', 'into',

'about', 'also', 'more', 'some', 'no', 'nor', 'any', 'all', 'both',

'each', 'other', 'when', 'where', 'how', 'why', 'who', 'because']

**מילות קישור (Conjunctions):** מילים שמחברות בין חלקי משפט, כמו *and*, *or*, *but*, *if*, *then*, *else*.

**מילות יחס (Prepositions):** מילים המציינות קשרים בין מילים או מושגים כמו *at*, *on*, *for*, *with*, *by*, *from*, *in*, *into*, *about*.

**כינויים (Pronouns):** מילים שמחליפות שמות עצם, כמו *it*, *this*, *that*, *who*.

**פעלי עזר (Auxiliary verbs):** פעלים שמסייעים לבטא זמן, מצב או אפשרות, כמו *is*, *are*, *was*, *were*, *be*, *has*, *have*, *had*, *can*, *will*, *just*.

**מילות שאלה (Interrogatives):** מילים המשמשות לשאילת שאלות, כמו *who*, *how*, *why*, *when*, *where*.

**מילות תואר וכמות (Determiners/Quantifiers):** מילים שמציינות כמויות או מיקומים, כמו *some*, *all*, *both*, *each*, *other*, *more*, *no*, *any*.

**מילות הסבר והרחבה (Discourse markers):** מילים שמחברות בין רעיונות או מוסיפות הדגשה, כמו *because*, *so*, *than*, *too*, *very*, *also*.

המילים האלה נפוצות מאוד בטקסטים והסרתן מסייעת להתמקד במילים המשמעותיות בלבד.

3. לצורך בניית האינדקס בחרנו להשתמש ב-lemmatization מכיוון שהוא מאפשר לזהות את השורש הדקדוקי התקני של המילה כפי שהיא מופיעה במילון, ולא להסיר סיומות ללא הבחנה כמו ב-stemming. בשלב של עיבוד המילים, השתמשנו במחלקה WordNetLemmatizer של NLTK, שהיא כלי לביצוע lemmatization.

כמו כן, העדפנו שלא להשתמש ב-stem מכיוון שהוא מבצע חיתוך גס ללא התייחסות להבנה סמנטית או לשורש הדקדוקי של המילה, דבר שעלול ליצור מילים ללא משמעות לאחר החיתוך. לדוגמא: incom ← incoming .

חלק שני: בניית מסכים להצגה בכיתה (50 נקודות)

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

המסכים צריכים לכלול (לפחות) (20 נקודות):

מסך מנהל למערכת, מסך שאילתא למנוע החיפוש (הזנה והצגת תוצאות), מסך סטטיסטיקות מעניינות.

בשלב זה נדרש לממש במלואם את בניית מסך המנהל ומסך השאילתא. ניתן את התוצאות להציג כרגע עם data חלקי.מומלץ לממש ככל הניתן גם את מנוע החיפוש , כפי שלמדתם בתרגול 6 😊

את מסך הסטטיסטיקות עליכם לממש בצורה בסיסית. עם זאת כמובן ,שכל תוספת שתחליטו עליה, תוביל להערכה גבוהה יותר של המשימה.

Colab:

<https://colab.research.google.com/drive/1g3Cy3LrFPTIjXRMcl0UxI5n3wp3cYIZm?usp=sharing>

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

1. התייחסו ל -8 כללי הזהב של שניידרמן (הוצגו בתרגול). כיצד המערכת שלכם מבטאת אותם? (5 נקודות)

### **יישום 8 כללי הזהב של שניידרמן במערכת שלנו**

1. **עקביות** – שימוש בעיצובים ותהליכים אחידים בכל המערכת.  
    *דוגמה:* כל כפתורי הניווט והפעולה (כגון כפתור החזרה או החיפוש) ממוקמים באותו מקום ועיצוב בכל המסכים.
2. **קיצורי דרך למשתמשים מנוסים** – המערכת פשוטה ומציעה למשתמש פעולות פשוטות ומהירות, כך שאין צורך ממשי לאפשר קיצורי דרך למשתמשים מנוסים.
3. **משוב לפעולות המשתמש** – המערכת מודיעה מיד על הצלחה או כישלון.  
    *דוגמה:* הודעה שמוצגת כשהחיפוש נכשל או כאשר המערכת מציגה את תוצאות החיפוש בצורה מיידית.
4. **מניעת שגיאות** – תכנון שמונע פעולות שגויות מראש.  
    *דוגמה:* המערכת לא מאפשרת לבצע חיפוש ריק – ישנה דרישה להזין מונח לפני שניתן ללחוץ על כפתור החיפוש. כאשר החיפוש ריק - הכפתור אינו לחיץ.
5. **דיאלוגים לסיום תהליך** – הצגת הודעות שמסמנות למשתמש שתהליך הושלם.  
    *דוגמה:* לאחר הצגת תוצאות חיפוש או טעינת נתונים, מוצגות תוצאות בצורה ברורה ומסודרת.
6. **התאוששות משגיאות** – לאפשר למשתמש לחזור או לתקן בקלות.  
    *דוגמה:* כפתור חזרה לדשבורד הראשי שמאפשר לצאת בקלות מכל מסך אחר במערכת.
7. **שליטה למשתמש** – לאפשר למשתמש שליטה חופשית על פעולותיו.  
    *דוגמה:* המשתמש יכול לנווט בין המסכים ולבחור את המידע שהוא רוצה לראות ללא מגבלות.
8. **הפחתת עומס זיכרון** – המערכת עוזרת למשתמש ולא מחייבת לזכור פרטים.  
    *דוגמה:* תיבת החיפוש מציעה מילים רלוונטיות בהתאם למה שהמשתמש מתחיל להקליד.
9. יש להגיש את הטבלה הבאה , תוך התיחסות למשובים שקיבלתם (5 נקודות):

|  |  |  |  |
| --- | --- | --- | --- |
| נושא כללי | הערת משוב | האם התבצע שינוי באפליקציה בעקבות ההערה? | נימוק |
| גרפים | 1. לשפר את הגרף של הטמפרטורה  2. גרפים לא ברורים אין הצגה ברורה של הנתונים  3. גרפים לא ברורים אין הצגה ברורה של הנתונים  4. צריך להוסיף נותנים לגרפים  5. עיצוב וערכים בגרפים יותר מונגשים | 1. כן 2. כן 3. כן 4. כן 5. כן | 1. הגרף הציג נתונים בצורה שאינה מובנת למשתמש.  2. הגרפים אכן לא היו ברורים ולכן תיקנו אותם.  3. הגרפים אכן לא היו ברורים ולכן תיקנו אותם.  4. גרפים צריכים לתת מידע ונתונים למשתמש ולכן שיפרנו.  5.הגרפים צריכים להיות מובנים ונעימים למשתמש, לכן תיקנו. |
| משחקיות | 1. לממש את המשחקים  2. צריך לממש גם פונקציונליות של משחקיות שחסרה  3. להוסיף לוח ניקוד | 1. כן 2. כן 3. כן | שלושת ההערות נוגעות לנושא המשחקיות בין העובדים שעדיין לא מימשנו בשלב הסדנא. המשחקיות היא מרכיב משמעותי בפרויקט ולכן כמובן שלאחר הסדנא מימשנו את הפיצ'ר הזה. |
| עיצוב | 1. צבעים  2. הצגת נתונים בעיצוב  3. עיצוב בעמוד חיפוש  4. הצבעים במערכת אדום עם לבן לא רואים את הכתב  5 להגדיל כתב.  6.להוסיף בתצוגת admin dashboard הפרדה יותר מובהקת בין הנתונים בבועות (אולי בעזרת צבעים יותר מפרידים)  7. הצבעים של הui  8. להוסיף כותרת ותיאור לתוצאות חיפוש  9. לשפר את העיצוב, להוסיף אולי צבע | 1. לא 2. לא 3. לא 4. כן 5. כן 6. כן 7. לא 8. כן 9. לא | 1. הערה כללית מדי.  2. לא הבנו את כוונת ההערה.  3. הערה כללית מדי.  4. הערה עניינית ורלוונטית, פנינו אל הלקוח ולכן תיקנו.  5. הערה עניינית ורלוונטית, פנינו אל הלקוח ולכן תיקנו.  6. הערה עניינית ורלוונטית, פנינו אל הלקוח ולכן תיקנו.  7. הערה כללית מדי.  8. הערה נכונה, אך זה קיים כבר במערכת.  9. הערה כללית מדי. |
| פונקציונליות חיפוש | 1. להוסיף שיהיה אפשר לחפש 2 מילים | 1. כן | 1. פונקציונליות חשובה שהיא הבסיס של מנוע חיפוש, לכן מימשנו. |
| פיצ'ר | 1. לשפר את הניהול עובדים  2. להוסיף בטבלת עובדים אלמנטים | 1. לא | 1. הערה כללית מדי.  2. |

1. יש לרשום את ציון ה SUS של המערכת שלכם.מה מעיד הציון?(5 נקודות)

הציון הממוצע של שאלון ה־SUS שקיבלנו הוא 91.2, שזה נחשב לציון גבוה. לפי הסקאלה של השאלון, זה אומר שהשמישות של המערכת היא ברמה מצוינת. המשתמשים הרגישו שהמערכת קלה לשימוש, ברורה ולא מסובכת. זה ציון שמראה כי חוויית המשתמש מאוד טובה, ושלא היו הרבה בעיות או בלבול בשימוש.

1. הגדירו 3 מדדים להצלחת המערכת (הרצאה 3).(5 נקודות)

**שאלה 4:**  
על מנת להעריך את הצלחת המערכת שפותחה, הגרדנו שלושה מדדים מרכזיים אשר ישקפו את איכותה, תפקודה והערכה למשתמשים:

1. זמינות המערכת (Availability) – מדד זה בוחן את רמת הנגישות והזמינות של המערכת לאורך זמן. מערכת יציבה, הפועלת באופן רציף ללא השבתות בלתי צפויות, תיחשב למוצלחת יותר.
2. שביעות רצון המשתמשים (User Satisfaction) – נמדדת באמצעות משוב מהמשתמשים על חוויית השימוש במערכת. לשם כך ניתן להיעזר בשאלונים תקניים כגון System Usability Scale (SUS), המעניקים ציון מספרי לחוויית המשתמש הכללית.
3. ביצועי המערכת (Performance) – מדד זה בוחן את היכולת של המערכת להגיב במהירות ולספק למשתמשים חוויית שימוש זורמת וללא עיכובים. המדד מתייחס לפרמטרים כמו זמן טעינת נתונים, מהירות הצגת המידע על המסך בתצורה גרפית ויכולת עיבוד נתונים בצורה רציפה ואמינה, גם כאשר מדובר בכמויות מידע גדולות או בשינויים תכופים מצד המשתמש. מערכת עם ביצועים גבוהים מאפשרת למשתמשים לקבל מידע עדכני בצורה מהירה ויעילה, ללא המתנה מיותרת.
4. הציגו דיאגרמת ארכיטקטורה של המערכת שלכם. הסבירו באיזה סוג ארכיטקטורה השתמשתם (הרצאה 7), ופרטו את חלקי הקוד ההמתיחסים לכל חלק בארכיטקטורה.(10 נקודות)

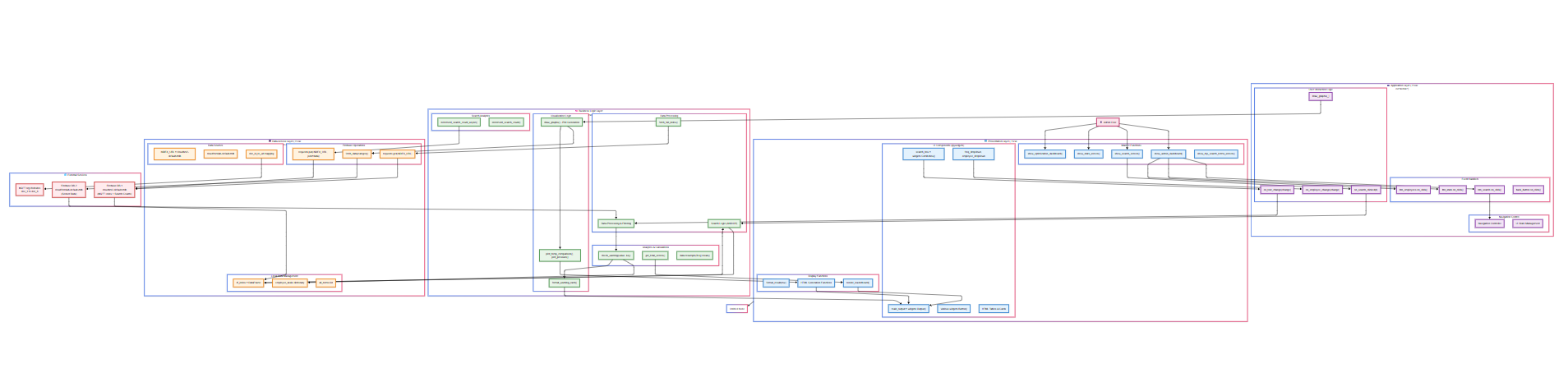
**שאלה 5:**

1. Presentation Layer (שכבת הצגה)
   * אחראית על הממשק הגרפי למשתמש.  
     מימוש בדמות פונקציות כמו show\_admin\_dashboard(), show\_search\_screen(), show\_stats\_screen() שמייצרות תצוגות עם ipywidgets.
   * הכפתורים, התוויות, הטבלאות, והגרפים מוצגים כאן.
2. Application Layer (שכבת יישום/שליטה)
   * מטפלת בניווט בין המסכים ושליטה בפעולות המשתמש.
   * דוגמה: btn\_search.on\_click(), btn\_stats.on\_click(), back\_button.on\_click()  
      מנהלות את המעבר בין הפונקציות ואת פעולות המשתמש.
3. Business Logic Layer (שכבת לוגיקה עסקית)
   * מבצעת את עיבוד הנתונים, חישובי נוכחות, סינונים, או חישוב מדדים.  
     דוגמה: fetch\_full\_index(), calculate\_general\_attendance(), check\_warning(), draw\_graphs().
4. Data Access Layer (שכבת גישה לנתונים)  
   מנהלת את הגישה למסדי נתונים מרוחקים, למשל Firebase.  
   דוגמה: requests.get(INDEX\_URL) ו- fetch\_data(category).

### למה בחרנו בארכיטקטורה זו?

* כי היא ברורה, מודולרית ומאפשרת תחזוקה נוחה.
* מאפשרת הפרדה בין ממשק המשתמש לבין עיבוד הנתונים והגישה למקורות המידע.
* מתאימה ליישומים מבוססי ווב ואינטראקטיביים כמו שלך.

[קישור לדיאגרמה](https://www.mermaidchart.com/play?utm_source=mermaid_js&utm_medium=button&utm_campaign=try_playground#pako:eNqVWOtuG8cVfpWBDBQUKikiZcmy0ARYXiWY5LLcpe2gLgbD3SG11XKH2YtlNsgbFG1juEWDJohbtA3iH_3dX30NPkbzCD0zszt7G1Iw-YPL851z5syZc5v98sBhLj24Olj47N65JWGM7ParAEXJfBmS9S1_GPCH01-9OhgRL0CzG9RPAif2WBC9Ovg18KafrmFdA1N0y-4xcVdegF0S3c4ZCd3GYYnT6hnTzjWe3WTsESWhc4sjJ6Q0qDLbhm0VeWMSR3pWc2IXGNk69lbebwm3dKcpE3NSEInZOrMlpuGqtgwNXJ1vmiAPbumw1ZoFNIgj1PDWm3vPXdI4Kq83Mm7G2JzZILECb2KWxOskRp-ilPvEFAS9w9rmS26pNHDO3hTEYO05A1JFsD2zbXNsgdRzEnosiZRAO4ljVnVgd2pOuuYLIbAI6RfYDdnaZffBEaKrtc82lCpSSdA22sMel7q2R0Nkk7lPI_Qz1AGHR_t91wKhrhetfbLZEVd9czoyuMMWLFyRGIc0Svy4EbL7svHT3rjbmwJfCEvREPuUwI_u0LmReNAbZ_YOaEBDESc1E3ZYfQaiP33_53_-7z9_QBMwCA5dyg_Jhoaosf1x-9ft19sPaPtu-6_t2-27sgEqqeqkZp3U2mPIYzCk9xqWR9ckcH0aln3XtsdYBg_wzeMgje4TyAjH95y7arhwdp5vGTfPtX3MvdEkZc3iYze70XmGQYbzE-cOz0UA1rl3bPSc51gEvr0JIDWJI73Nlp5TLgGF_bJsu3IF6jbmh1Vmu_fSxp1rYzzoSYmYvokxFMJgSRvypyYDmy6JqNTYI9adGi_wYGpMri3cMYZDkHRDco_FJqMGfmD3F8A_Jq-9pYyyDgvikPmlFcbGc9yxp0MtJwRGraj28GgwsmXpsuCgKRqRgCzpCqJpvzVPRPT__t_IWK_Bszsi_xv4_rj90_a7T-DnW_jzl11p8LhOOq-TLvZYdMmrCIkJJCNzaBR5wbJcQ3q2CIkFjSEcFonvYw-KRLVcdg3bwJOp2amrg2rW93wIvarqtDYPzcENF7NExMnIRA1j3P3EnD5wuE9BzAiIv4k9R1ZN30l8Uq-EL4zp-GY8gODrdZ6BkHNLIZHuSRiAUY3XxE_oEbqjm_KmbNM2htjqmFNRoaH6Q5-LiQ_NjUHpalSrqGWMJkMe2i544AQ4CMQ3bfB-cHiyouThdshnhedelBA_bb-aRC0kRCUXDtExmvgsLpTlcssemjbuz8YdLgiZB9uBDMQOdF_ocRFvakdI0NdgfJSEtLJF5UZj2rXyvpL6ETvFhrFri838qNXRlRYxrM_HHdwxZ2OeYl4AgwRPLFWRWAJ_SLQJHH2v3y_5oH0tkaI_vEftBKIX3JBGZC1Lv4UMfQvJ-cP2Hdr-ffs3ePhm-0Gfp5d10lNNB9N1teY-Y3k37XshnZOIInOdHnrZn4OejW-gwb8UDf6LhEbQmSCWG4KIZ9NhZbCbafj5ZKX4j9BvIFg-5VF-WC8WmNcCVTE4UwMKHV2ycPOQ7x9n1cNiSQjlo6RcLQ9M6hkGOcdniXu7ah27dEFgvjkOY3deCYyxZU5TUcG-CNnKJ_PdIl2zg0fGZALRzpOMOdhzIflxEvpoRdbrvJbt2gtvu0PmEB-JHdU6RLpOX_naXcjKClviEv2QrGiJF7rn0Py818O2YT3j6afaZ0yiuwi5nujtJNyU82k4hGY9HXEJAuVbzOfI96IHelXzQmTC2__KDRiOI5KhmgZvRbN6x_Ph_fYD5MR7IHynT4PmmYam6WHN8312ySb6uz-i3hvYC9QQZNHwtVeNl367WcyObhs1fzEPP9OGCwcao1_aNgxJ_Ah-jtIS1eE1o3IT6bdbFcUtpVgXWEK5RYOIhcKVlXsNrIqtG1u0GP7nhIVL9ILOIy-mEZfl0ddEMUP84aJ0aPCZWWJ0_-n7r_-BDH55RHzYAy50fPyZuFpCY1T3Rv6cXgvhUV77pB7ByUXyyRc41FybPkMMSvZcI5dRw2MJhFtXhhamxXS57MKUceSzoWQomME5siFNgmYJKw4SCi-smHoinVCy9ct)



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.

**תשובה:**

התוספת שמימשנו היא פיצ'ר להצגת 10 המונחים המחופשים ביותר, בפורמט טבלה מעוצבת ונוחה לקריאה.

### הסבר על התוספת:

* מטרת הפיצ'ר היא לתת למנהל מבט מהיר ואינטואיטיבי על אילו מונחים חיפשו המשתמשים הכי הרבה.
* הפיצ'ר משפר את חוויית הניהול והבקרה, בכך שהוא מספק מידע שימושי לקבלת החלטות — למשל, אילו נושאים מעניינים את המשתמשים ביותר, או אילו מונחים כדאי להרחיב עליהם במערכת.

הפיצ'ר של הצגת 10 המונחים המחופשים ביותר ממומש בתוך הפונקציה show\_top\_search\_terms\_screen, שם מתבצעת שליפת נתוני החיפושים מ-Firebase, מיון הנתונים, ובניית טבלת HTML מעוצבת להצגה.

עבור המנהל, הפיצ’ר מתבטא במסך ייעודי המציג טבלה ברורה ומעוצבת של עשרת המונחים המחופשים ביותר, הכוללת את שם המונח וכמות החיפושים שלו, ומאפשרת הבנה מהירה של תחומי העניין המרכזיים של המשתמשים.

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!