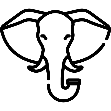
**מבוא למחשוב ענן - סמסטר חורף התשפ"ה**

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

**Group Elephant**



מועד הגשה: 12.1.25

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

שימו לב: למטלה זו שלושה חלקים

|  |  |
| --- | --- |
| **שם חבר צוות** | **ת.ז** |
| **מוחמד ג'אבר** | **212023782** |
| **אמל קנדיל** | **322727470** |
| **אסראר מגיס** | **208247262** |
| **נור דבאח** | **213911530** |
| **ראידה ח'ורי** | **314811530** |
| **בשאר חוסרי** | 212931679 |

*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** |
| נור דבאח (מהנדסת מערכת) - *scrum master* | ארגון ישיבות צוות.  ניהול לו"ז הפרויקט.  *בדיקת התקדמות העבודה.* | הכל |
| **אמל קנדיל -***frontend developer* | פיתוח ממשק משתמש עבור דף. | הכל |
| מוחמד ג'אבר - *backend developer* | פיתוח API עבור המערכת.  יישום מנגנון אימות משתמשים. | הכל |
| אסראר מגיס - *product manager* | איסוף דרישות לקוח. | הכל |
| ראידה חורי - UI | עיצוב חוויית משתמש. | הכל |
| בשאר חוסרי - QA | ביצוע בדיקות ידני במערכת ודיווח על תקלות ובאגים לתיקון | הכל |

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 |
| admins | מידע על המנהלים (שם משתמש וסיסמה) |
| most\_favorited | עוקב אחרי ספירת הפעמים שהמונחים סימנו כמועדפים על ידי משתמשים |
| most\_summarized | עוקב אחרי ספירת הפעמים שהמונחים מסוכמים |
| users | מנהל מידע הקשור למשתמש, כולל נתוני התחברות, התקדמות הלמידה והיסטוריית חידון. |

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(4) | האם התבצע שינוי באפליקציה בעקבות ההערה? | נימוק |
| מסך אדמין שיהיה יותר מסודר | כן | בוצע סידור מחדש של מסך האדמין, כולל ארגון מחדש של אלמנטים ושיפור מבנה תצוגת הנתונים. |
| עיצוב | לא | עיצוב האפליקציה נשאר כפי שהיה, שכן הוא תואם את מטרות המערכת . |
| יתמקדו במשהו אחד | לא | האפליקציה מיועדת לתת מענה לכמה צרכים במקביל, ולכן הוחלט לשמר את הרב-תכליתיות שלה. |
| לא תמיד עבד הכפתור התחברות | כן | באג בכפתור התחברות תוקן, ועבר בדיקות לוודא את פעילותו התקינה בכל המצבים. |
| מהירות | לא | המערכת נחשבת מהירה מספיק עבור המשתמשים ולכן לא בוצעו שינויים |

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

**1. Strive for Consistency (חתירה לעקביות):**המערכת שומרת על עקביות בעיצוב ובמבנה:

* כל המסכים משתמשים באותו סגנון עיצובי, גופנים, וצבעים.
* התפריטים העליונים (כמו Dashboard, Search) מסודרים באותו מיקום בכל המסכים.
* כפתורי פעולה, כמו **Login**, **Sign Up**, ו-**Favorite**, מוצגים בפורמט אחיד.

**2. Enable Frequent Users to Use Shortcuts (לאפשר קיצורי דרך למשתמשים מנוסים):**

* **Personalized Suggestions** בחיפוש מאפשרים למשתמשים מנוסים לגשת למונחים פופולריים או רלוונטיים במהירות.
* התפריטים העליונים מספקים גישה מהירה לפונקציות מרכזיות כמו סטטיסטיקות, היסטוריית חיפושים, והוספת מונחים.
* **Favorites** המועדפים של המשתמש מוצגות במסך הראשי dashboard לגישה מהירה.

**3. Offer Informative Feedback (משוב אינפורמטיבי):**

* בעת חיפוש, המשתמש מקבל תוצאות עם פרטים מלאים, כמו שם המונח, אחוזי התאמה, וקישורים.
* כפתורי פעולה כמו **Favorite** משנים צבע בעת לחיצה, מה שמבהיר שהפעולה הושלמה בהצלחה.
* בעת טעינת נתונים, מוצג עיגול מסתובב (Spinner) שנותן אינדיקציה לכך שהמערכת מעבדת את הבקשה.

**4. Design Dialog to Yield Closure (דיאלוגים שמניבים סגירה):**

* לאחר סיום חיפוש, כל התוצאות מוצגות בצורה מאורגנת ומלאה, מה שמספק למשתמש תחושת סיום.
* לאחר הוספת מונח במסך המנהל, כפתור **Refresh Terms** מאפשר לראות שהמונח נוסף בהצלחה.

**5. Offer Simple Error Handling (מניעת טעויות):**

* הודעות שגיאה ידידותיות מוצגות כאשר המשתמש מזין נתונים שגויים (לדוגמה, שם משתמש או סיסמה לא נכונים).
* המערכת בנויה כך שכפתורים ואפשרויות שאינם רלוונטיים למצב הנוכחי של המשתמש מוסתרים או מבוטלים.

**6. Permit Easy Reversal of Actions (אפשרות לבטל פעולה בקלות):**

* לחיצה נוספת על **Favorite** מבטלת את הסימון כמועדף.
* כפתור **Back** מאפשר חזרה למסכים קודמים.
* **Header** המוצג בכל המסכים מאפשר חזרה ומעברים מכל מקום לכל מקום אחר.

**7. Support Internal Locus of Control (תחושת שליטה פנימית):**

* המערכת מגיבה רק לפעולות יזומות של המשתמש, כמו חיפוש, סימון מועדף, התחלת חידון, סיכום לינקים.
* המשתמש שולט באופן מלא בפעולות דרך תפריטים וכפתורים ברורים.

**8. Reduce Short-Term Memory Load (הפחתת עומס זיכרון):**

* המידע מוצג בצורה ברורה עם גרפים ונתונים חזותיים (לדוגמה, גרף נתוני חיפוש וסטטיסטיקות).
* הצעות מותאמות אישית בחיפוש מפחיתות את הצורך לזכור מונחים שנחפשו בעבר.

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

עבור השאלות האי-זוגיות (1, 3, 5, 7, 9), יש להפחית 1 מהציון הניתן (ציון-1).

עבור השאלות הזוגיות (2, 4, 6, 8, 10), יש להפחית את הציון הנתון מ-5 (5-ציון).

נחשב את התוצאות של כל השאלות ונחבר אותן יחד.

נכפיל את הסכום ב-2.5 כדי לקבל את הציון הסופי, מתוך 100.

סה"כ: **80**

**הציון מעיד:**

המערכת מעידה על רמת שימושיות גבוהה מאוד. המשתמשים מדווחים שהיא נוחה, אינטואיטיבית, ונעימה לשימוש. ציון ה-SUS של המערכת שלנו ממוקם גבוה על סולם השימושיות, ולעיתים קרובות משתייך לקטגוריה העליונה של "Excellent" במודל פרשנות של SUS. המערכת קלה להבנה ושימוש, ומשקפת חוויית משתמש חיובית במיוחד.

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

* **) Usabilityשמישות):**

הגדרה: רמת הקלות והנוחות של המשתמשים בהפעלת המערכת וביצוע פעולות.

איך זה מתבטא בקוד:

ממשק אינטראקטיבי ופשוט.

פידבק מיידי על פעולות המשתמש.

פונקציות נלוות (כמו סיכום ומועדפים) שחוסכות זמן למשתמשים.

* **Maintainability (תחזוקה):**

הגדרה: היכולת לשנות, לתקן, או לשדרג את המערכת בקלות – בין אם מדובר בתיקון באגים, שיפור ביצועים או התאמה לסביבה חדשה

איך זה מתבטא בקוד:

הקוד מחולק ליחידות ברורות (מודולים) – Model, View, Controller (MVC).

כל מחלקה אחראית לחלק אחר במערכת:

ElephantModel – טיפול בנתונים והתחברות לפיירבייס.

ElephantView – בניית הממשק הגרפי עם ipywidgets ו HTML.

ElephantController – ניהול הזרימה בין הנתונים לתצוגה.

* **Scalability (סילומיות):**

הגדרה: היכולת של המערכת להתרחב ולהתמודד עם עלייה בכמות המשתמשים או בעומסים – מבלי לפגוע בביצועים.

איך זה מתבטא בקוד:

המערכת משתמשת בFirebase – שירות ענן שמאפשר גידול במשאבים לפי הצורך (ללא צורך בניהול שרתים פיזיים).

המערכת בנויה מ microservices מה שמאפשר התרחבות בצורה מאוד קלה.

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

A screenshot of a computer

Description automatically generated

**הארכיטקטורה שבחרנו בה** MVC **:** עם מיקרו-שירותים

החלטנו לבנות את מנוע החיפוש שלנו בארכיטקטורה מסוג MVC, כיוון שמצאנו אותה כמתאימה ביותר לפרויקט שלנו. ארכיטקטורה זו מאפשרת הפרדה ברורה בין שכבות המידע (מודל), התצוגה (View) והשליטה (Controller), מה שתורם לניהול קל של קוד, שימוש חוזר בו ויכולת תחזוקה גבוהה יותר.

לאחר שביצענו את החלוקה הראשונית, יצרנו מיקרו-שירותים קטנים בכל אחת מהשכבות. שירותים אלו מאפשרים:

גמישות ויכולת הרחבה לכל חלק של המערכת בנפרד.

אפשרות לעבודה עצמאית של צוותים שונים על חלקי המערכת.

הקטנת השפעת תקלות על המערכת כולה.

חלקי הארכיטקטורה במערכת שלנו:

:Model השכבה האחראית על הלוגיקה העסקית של המערכת והתקשורת עם בסיס הנתונים. במערכת שלנו, השירותים Stats Service, Userauth Service ,QuizHistory Service ו Term Service הם חלק מהשכבה הזו. הם מטפלים בעיבוד נתונים ושמירתם בזמן אמת בבסיס הנתונים של Firebase, יש לנו גם את ה Word ו Term Mappers שהם עובדים על המונחים והשאליתות במערכת, Search engine service היא החלק שמבצע את עיבוד הנתונים והחיפוש המלא ומחזירה את התוצאות, Fuzzy Search matcher מתקן הטעויות בשאילתת המשתמש אם קיימות כאלו ומציע דברים קרובים לזה, ה URL Summarizer מסכם ומחזיר טקסט מסוכם לפי URL וה Elephant Model מטפל בפעולות כלליות ומשותפות בין כמה רכיבים ומחבר את הכל ביחד.

View : שכבת התצוגה אחראית על הצגת המידע למשתמשים. רכיב ElephantView מייצג את השכבה הזו ומאפשר ממשק משתמש ברור ונוח, מימשנו את הכל ביחד כי זה היה יותר מימושי ועובד בצורה יותר חלוקה להחליף מסכים מאותה cell של קוד.

:Controllerשכבת השליטה אחראית על ניהול קלט המשתמש וקישור בין המודל לתצוגה. לדוגמה AuthController מטפל באימות משתמשים, SearchController and Quiz Services מטפל בשאילתות חיפוש ויצירת חידונים, AdminController מנהל פעולות אדמיניסטרטיביות, UserController מנהל פעולות המשתמש ו ElephantController מטפל בפעולות כלליות ומשותפות בין כמה רכיבים ומחבר את הכל ביחד.

הדיאגרמה המצורפת ממחישה את החלוקה הברורה בין השכבות והחיבור ביניהן, תוך שימוש ב MVC ומיקרו-שירותים.

*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 | תיאור |
| Summarizer | URL summarizer שמקבל כתובת URL וקורא את פסקאות ה-<p> בכתובת זו, מדרג כל משפט שקרא על פי תדירות מילים, אורך ומיקום, ומחזיר את המשפטים המובילים שמצא. השתמשנו בזה כשירות מיקרו שנקרא URLSummarizer, כדי לאפשר למשתמש לסכם קישור מתוצאות החיפוש ולדעת על מה מדובר בכתובת ה-URL. הוא מופעל כאשר המשתמש לוחץ על "סכם" בתוצאת חיפוש. עבור המנהל, הוא יכול לראות את הנושאים הכי מסוכמים במהלך החיפושים, מה שעוזר לזהות אילו נושאים מעניינים את המשתמשים ואילו נושאים הם אוהבים לקרוא עליהם. |
| Quiz System | מערכת חידונים המאפשרת יצירת שאלות חידון לנושאי חיפוש על ידי חיפוש מילות מפתח כגון "הוא", "הם", "פירושו", "מייצג..." בטקסטים מכתובת URL. מילים אלו עוזרות לנו לזהות אילו משפטים בטקסט יכולים לשמש כשאלה. כשהמשפטים המתאימים נמצאים, הם נשמרים ומנותחים לפי המילים שהם מכילים כדי לדעת איזה סוג שאלה מתאים להם (נכון/לא נכון, בחירה מרובה, או השלמת החסר). המטרה הייתה להפוך את המנוע לאינטראקטיבי ומהנה יותר ולעזור למשתמש לבדוק את הידע שלו על המידע שזה עתה למד. היא מופעלת כאשר המשתמש לוחץ על "התחל חידון" בתוצאת חיפוש מסוימת, אין קשר בין תכונה זו למנהל. |
| Favorites | מועדפים מאפשרים למשתמשים להוסיף קישורים מועדפים ללוח המחוונים שלהם. כך הם יכולים ליצור קיצורי דרך לאתרים מועדפים ולשמור אותם. התכונה מופעלת כאשר המשתמש לוחץ על כפתור המועדפים מתוצאת חיפוש. הכפתור מתעדכן ומוסיף/מסיר את האתר מ-Firebase, וכפתור בלוח המחוונים מתעדכן בהתאם. בלחיצה על הכפתור, ה-URL המתאים נפתח בטאב חדש. המנהל יכול לראות אילו נושאים הם הכי מועדפים, מה שעוזר לו להבין אילו נושאים המשתמשים מעדיפים. |
| Personalized Suggestions | הצעות מותאמות אישית עוזרות למשתמש לבצע חיפושים מהירים יותר ולהקל על השימוש במנוע שלנו. הצעות אלו מבוססות על חיפושים קודמים של המשתמש ומציעות חיפושים שעוזרים לחסוך זמן וליצור קיצורי דרך. ההצעות מוצגות מתחת לשורת החיפוש. כאשר לוחצים עליהן, שם הנושא מופיע בשורת החיפוש ומבוצע חיפוש. אין קשר בין תכונה זו למנהל. |
| Term Mapper | מיפוי מונחים שממפה מילים נרדפות למילה מסוימת, כדי למנוע בעיות חיפוש כאשר משתמש מחפש נושא בשם שונה ממה שיש בבסיס הנתונים. כך נמנעים מתוצאות ריקות ומספקים חוויית משתמש טובה יותר. תכונה זו משווה את קלט המשתמש למילים ממופות בבסיס הנתונים ומחליפה את הקלט במונח המתאים. אין קשר בין תכונה זו למנהל. |
| Word Mapper | מיפוי מילים שנועד לשפר את תוצאות דירוג האלגוריתם. המיפוי מופעל כאשר המנהל מוסיף מונח חדש וכתובת URL. לפני שליחת הנתונים לדירוג, המיפוי מוצא מילים קרובות ומחפש אותן, לדוגמה, עבור המונח "פיתוח", המערכת מחפשת [פיתוח, מפתח, תכנות...]. התכונה שיפרה את תוצאות הדירוג משמעותית. |
| FuzzySearchMatcher | מערכת לתיקון אוטומטי/האם התכוונת שמפעילה בדיקה בהתבסס על מיפוי מונחים, עד כמה קלט המשתמש קרוב למונח במפה. אם לא נמצאו תוצאות חיפוש, המערכת מופעלת ובודקת את המרחק של Levenshtein (כמה שלבים נדרשים לשנות את הקלט למונח ממופה). אם המרחק הוא 3 שינויים או פחות, מוצגות הצעות מתחת ל"האם התכוונת". בלחיצה על הצעה, המשתמש מועבר לתוצאות חיפוש של המונח. תכונה זו משפרת את חוויית המשתמש והופכת את המנוע לידידותי יותר. אין קשר בין תכונה זו למנהל. |

**Notebook:** [**https://colab.research.google.com/drive/1bqJXrWllmWwmyxXrVl\_CUjjuO3qI34di#scrollTo=3ouCsYKR6gO8**](https://colab.research.google.com/drive/1bqJXrWllmWwmyxXrVl_CUjjuO3qI34di#scrollTo=3ouCsYKR6gO8)

|  |  |
| --- | --- |
| **Username = aa**  **Password = a** | **Admins data:** |
| **Username = cc**  **Password = c**  **Username = amal**  **Password = 123**  **Username = mohammed**  **Password = 123**  **Username = test1**  **Password = 1** | **Users data:** |

**Git Repository:** [**https://github.com/Elephant-Team/Cloud\_Elephant\_Team\_Project**](https://github.com/Elephant-Team/Cloud_Elephant_Team_Project)

**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!