**מחשוב ענן- תרגיל בית 3**

|  |  |
| --- | --- |
|  |  |

**Link to the note-book Collab:** [**https://colab.research.google.com/drive/1X1kn8J14qp9ve640WG8Hp\_kRfaYPiOGB?usp=sharing**](https://colab.research.google.com/drive/1X1kn8J14qp9ve640WG8Hp_kRfaYPiOGB?usp=sharing)

**Link to GitHub:**

[**https://github.com/NivCohen1/HW3\_Sloth-.git**](https://github.com/NivCohen1/HW3_Sloth-.git)

****

|  |  |  |  |
| --- | --- | --- | --- |
| **שם חבר הצוות** | **משימות שהוקצו** | **משימות שהושלמו** | **בדיקת קבלה (acceptance test) – האם הבדיקה עברה?** |
| **אלכס ברדנסקי – backend developer** | **תיק מתכנת** | **הכל** | * **בדיקת הקוד** |
| **יאסר סעדי -**  **Product manager** | **הצגת סטטיסטיקה** | **הכל** | * **בדיקת נתוני הגרף** |
| **ניב כהן – מהנדס מערכת**  **Scrum master** | **Microservice**  **הכנת סרטון** | **הכל** | * **בדיקת ארגון הקוד** |
| **עדן עג'אינה -**  **QA** | **chatbot** | **הכל** | * **בדיקת את צ'אט-בוט בהתאם לשאלה הנתונה** * **בדיקת שאלות שאמורות להחזיר כלום** |
| **בר לייבוביץ -**  **Frontend developer** | **תיק משתמש** | **הכל** | * **בדיקת ה GUI** |
| **ניל אדרבר -**  **UL** | **לענות על השאלות** | **הכל** | * **בדיקת שתשובות תואמות לפרויקט .** |

**חלק 2 :**

**שאלה 1:**

הפיצ'ר הנוסף שבחרנו לממש : -Auto completeהשלמת חיפוש לפי תדירות החיפושים.  
הרעיון הוא להציג למשתמשים בעת חיפוש את התוצאות הנפוצות ביותר (שהיו בשימוש ביותר פעמים) קודם בתור הצעה להשלמת מילה בחיפוש, וזאת על מנת למקד את החיפוש ולספק להם תוצאות רלוונטיות יותר.

הסבר על הפיצ'ר:

הפיצ'ר מבצע סינון של התוצאות שהופיעו ביותר פעמים בחיפושים, כך שהמשתמשים רואים קודם את המונחים/תוצאות שהיו יותר פופולריות. הפיצ'ר מתבצע על ידי השגת ההיסטוריה של החיפושים ממסד הנתונים (Firebase) ומיון התוצאות לפי כמות הפעמים שחיפשו.

הפיצ'ר ממומש בשני חלקים:

1. עדכון היסטוריית החיפושים - כאשר משתמש מבצע חיפוש, נשמרת ההיסטוריה עם כמות הפעמים שנעשה שימוש ב- . Term
2. סינון התוצאות על פי תדירות החיפושים - כאשר מחפשים מונח, מתבצע מיון של התוצאות לפי תדירות החיפושים במערכת.

הפיצ'ר הזה מספק אינדיקציה למנהל על המונחים המובילים במערכת, ועוזר במיקוד המשאבים או בהבנת ההתנהגות של המשתמשים.

**2.**

בפרויקט השתמשנו Microservices עיקריים:

1. **FirebaseService**:  
   אחראי על תקשורת עם מסד הנתונים Firebase.  
   השירות מבצע פעולות שונות כגון:
   * הוספת כתובת URL חדשה למונח מסוים.
   * עדכון היסטוריית חיפושים של המשתמשים.
   * אחזור נתונים ותוצאות חיפוש מהמסד.

**יתרון:**  
השירות מבודד את כל פעולות התקשורת עם מסד הנתונים, מה שמאפשר ניהול נוח, מודולריות ושדרוג קל בעתיד, בלי להשפיע על שאר רכיבי המערכת.

1. **SearchService**:  
   אחראי על עיבוד והכנה של שאילתות חיפוש.  
   השירות מבצע:
   * ניתוח המונח המבוקש.
   * הכנה ואופטימיזציה של השאילתה למסד הנתונים.
   * שילוב נתונים רלוונטיים מהיסטוריית החיפושים לצורך שיפור התוצאות.

**יתרון:**  
השירות מבטיח הפרדה ברורה בין לוגיקת החיפוש לעבודה עם מסד הנתונים, משפר את הדיוק והמהירות של שאילתות החיפוש, ומאפשר גמישות במימוש אלגוריתמים שונים בעתיד.

**סיכום היתרונות בשימוש ב-Microservices:**

* **מודולריות**: מאפשרים להפריד בין תחומי אחריות של רכיבי המערכת, מה שמפשט את הניהול והתחזוקה.
* **סקלאביליות**: כל שירות יכול לגדול בנפרד לפי דרישות הביצועים שלו.
* **פיתוח עצמאי**: ניתן לפתח ולעדכן כל שירות בנפרד בלי להשפיע על שאר המערכת.
* **נוחות Debugging**: כל תקלה מוגבלת לשירות הספציפי, מה שמקל על זיהוי ופתרון בעיות

**3.צ'אטבוט במערכת:**

מימשנו צ'אטבוט אשר עונה על שאלות הקשורות למערכת, בהתבסס על המידע שנמצא ב-DB שלנו. הצ'אטבוט שפיתחנו הוא בצורת Hard Coded, ומספק מענה לשאלות בתחום ה-Cloud של Huawei. הצ'אטבוט מותאם לשאילתות על נושאים כמו סוגי שירותי הענן של Huawei, טכנולוגיות שונות ופתרונות בתחום ה-Cloud. המענה מבוצע בצורה אינטליגנטית על פי מילת מפתח שהמשתמש שואל, והצ'אטבוט בודק את השאלה ומספק תשובה מדויקת מתוך מאגר הנתונים שהוזן לו מראש.

**4. מהם ה -KPI הרלוונטיים בפרויקט שלכם? הסבירו?**

**User Experience:**

בנינו ממשק משתמש (Frontend) אינטואיטיבי שמאפשר הכנסת שאילתה בתיבת חיפוש בצורה ברורה. התוצאות מוצגות עם הדגשה של הלינק, כך שמשתמשים יכולים להבין במהירות את הרלוונטיות של התוצאות. הממשק עוצב בצורה נקייה, עם טקסט מודגש ושדות קלט ברורים.

בנוסף אנו משתמשים בשאלוני SUS וביקורת עמיתים על מנת לקבל דירוג של חווית המשתמש.

**זמן תגובה - Performance:**

טיפלנו במדד זה ע"י אופטימיזציה במסד הנתונים, מילון Firebase נבנה כך שהנתונים מסודרים לפי מילות מפתח, דבר המאפשר שליפה מהירה של נתונים.  
ע"י שימוש ב Lower case, stop words, stemming קיצצנו מילים מיותרות, איחדנו מילים מאותו השורש.

**5. שקיפות אלגוריתמית**  
  
אלגוריתם מנוע החיפוש שלנו אינו נגיש למשתמשים, אך הסברנו כיצד נבחרו תוצאות החיפוש. כאשר מתבצע חיפוש במנוע, התוצאות המוחזרות מהאינדקס מסודרות לפי סדר עדיפויות המבוסס על מספר המופעים של המילה בכל URL

במידה והשאילתה מורכבת ממספר מילים, האלגוריתם מחשב את סכום המופעים של כל מילה בביטוי עבור כל דף ומסדר את הדפים בהתאם לעדיפות זו.

בזמן הצגת תוצאות החיפוש למשתמש, אנו מציגים את הציון של כל דף ומוצג למשתמש כ: Total Appearances

ציון זה מאפשר למשתמש להבין כיצד התבצעה בחירת הדפים בתוצאות החיפוש

**.6 אתגרים שעלו במהלך העבודה**

1. חוסר תיאום בין חברי הצוות:
   * קבענו מפגשים קבועים, חילקנו משימות ברורות וביצענו סקירות התקדמות שבועיות כדי לשמור על תיאום בין כל חברי הצוות.
2. בעיות בתיאום של רכיבים שונים:
   * תכננו שלבי אינטגרציה הדרגתיים עם בדיקות בכל שלב ויצרנו ממשקים ברורים ומסודרים בין הרכיבים.
3. מגבלת מכסת נתונים ב-DB החינמי:
   * ביצענו אופטימיזציה באמצעות stop words, stemming, המרת מילים ל-lowercase ומניעת כפילויות. בנוסף, צמצמנו את העומס על ה-crawler על ידי הגבלת מספר האתרים הסרוקים.

באמצעות אופטימיזציה ותיאום טוב יותר, הצלחנו להתגבר על האתגרים ולהשיג תוצאות טובות יותר.

**\*\*חלק שלישי של התרגיל בית בקבצים נפרדים.\*\***