מבוא למחשוב ענן  
**תרגיל בית 2**

****

**Repository:** [לחצו כאן](https://github.com/tomerot/cloud-computing-kakadoo.git)

**Notebook**: [לחצו כאן](https://colab.research.google.com/drive/1GWQ76raqFSYuI9tnaXtPH2N7WesmIP5D?usp=sharing)

|  |  |
| --- | --- |
| **Team Kakadoo** | |
| **שם** | **ת.ז** |
| עופר אלזרע | 315824409 |
| תומר רוטמן | 316317023 |
| אורן עפרוני | 208741934 |
| גד עזריאל | 209146927 |
| אלכס גרומן | 320648918 |
| שלומי זריהן | 206877698 |

**1 - הקצאת המשימות לאיטרציה 1**

|  |  |  |
| --- | --- | --- |
| **שם חבר הצוות** | **משימות שהוקצו** | **הושלמו** |
| **תומר רוטמן (מהנדס המערכת  ו- Backend Developer)** | תכנון מבנה ה- DB ויצירה שלו | ✓ |
| פיתוח מיקרו-שירותים: שירות זחלן, שירות ליצירת אינדקס, שירות לשליפת מידע, שירות לאדמיניסטרציה ושירות שאילתות | ✓ |
| אינטגרציה עם עופר | ✓ |
| **אורן עפרוני (Scrum Master)** | הקצאת המשימות לאיטרציה | ✓ |
| קביעת פגישות שבועיות | ✓ |
| פיתוח פיצ'ר (יחד עם גד) | ✓ |
| **אלכס גרומן (Product Manager)** | עיבוד הפידבק מהסדנה ב- 6.1 (חישוב ציון SUS) ומענה על שאלות חוקי שניידרמן | ✓ |
| הוספת שליפות לצורך סטטיסטיקות לשירות שליפת המידע, פיתוח מיקרו-שירות לסטטיסטיקות ו- GUI לסטטיסטיקות (יחד עם שלומי) | ✓ |
| **עופר אלזרע (Frontend Developer)** | פיתוח הלוגיקה של צד המשתמש תוך שימוש ב- Fake Data | ✓ |
| אינטגרציה עם תומר | ✓ |
| **גד עזריאל (UI)** | שימוש בפלאגינים להמרת הבסיס העיצובי של האתר מקובץ  ה- FIGMA שפיתחנו בתרגיל בית 1 לקבצי HTML ו- CSS והטמעת הקבצים ב- Collab ע"י שימוש ב- HTML Widgets | ✓ |
| עיצוב Widgets | ✓ |
| פיתוח פיצ'ר (יחד עם אורן) | ✓ |
| **שלומי זריהן (QA)** | בדיקת הפונקציונליות של כלל השירותים תוך מתן דגש על מקרי קצה | ✓ |
| תיקון באגים שנמצאו ועדכון חברי הצוות | ✓ |
| הוספת שליפות לצורך סטטיסטיקות לשירות שליפת המידע, פיתוח מיקרו-שירות לסטטיסטיקות ו- GUI לסטטיסטיקות (יחד עם אלכס) | ✓ |

**2 – בניית מסכים**

1. משובים:

|  |  |  |
| --- | --- | --- |
| **הערת משוב** | **האם התבצע שינוי באפליקציה בעקבות ההערה** | **נימוק** |
| לשנות את העיצוב של הגרפים | שונה | שינינו את הגרפים כך שהצבעים שלהם יותאמו לצבעי האתר, והגוון משתנה בין מידע "חזק יותר" למידע "חזק פחות", כך שהגרפים לא בצבעים הסטנדרטיים |
| להוסיף פיצ'רים מורכבים | לא שונה | מעבר למגבלות הזמן, אנחנו רוצים לשמור על המערכת פשוטה |
| כתב יותר גדול | שונה | כעת הפונט של כל הכפתורים גדול יותר, מודגש, ובצבע לבן על גבי צבעים כהים כדי לשמור על ניגודיות |
| עיצוב | שונה | זהו משוב כללי מדי, אבל העיצוב שונה מזערית בעקבות המשובים האחרים. בפרט, הוספנו Hover Effect לכפתורים |
| באחד החיפושים התוצאות גלשו מעבר לטבלה המוגדרת | שונה | מדובר בבאג שקרה לנו לראשונה במהלך ההצגה. לאחר ניסוי וטעייה, התברר ש- CSS שהוזרק בדרך מסויימת לדף הסטטיסטיקות, גרם אחר כך לבאג בדף של ה- Admin שבו רשימת ה- URLs גלשה מעבר לגודל של הטבלה. שונתה דרך ההזרקה |
| זמן שליפת נתונים | שונה | כדי לשפר את זמן השליפה, במקום לחלץ את ה- Titles של ה- URLs בזמן הרצת השאילתה, כבר ביצירת האינדקס אנחנו שומרים Title לכל URL |
| לפשט את הממשק | שונה | זהו משוב כללי מדי, אבל סידרנו את הכפתורים השונים בצורה יותר נוחה לעין, הוספנו יותר מ- placeholder אחד להסביר כיצד להריץ שאילתה, ושינינו שמות לדברים מסויימים שנראו לנו אולי פחות ברורים |
| להוסיף צ'אט בוט | לא שונה | יתבצע לקראת ההגשה של תרגיל בית 3 |

הערה: היו מספר משובים בהם הוצע שנוסיף דברים שכבר היו במערכת כמו פיצ'ר ומשתמש אדמיניסטרטיבי לניהול האינדקס, ככל הנראה מי שמילא את המשובים הללו פספס את היכולות. כמו כן, איחדנו מספר משובים חוזרים.

1. דוגמאות לכך שהמערכת שלנו מבטאת את שמונת החוקים של שניידרמן:

חתירה לעקביות – הכפתורים ברובם באותם הצבעים ובכל דף המיקום של כפתור החזרה לעמוד הקודם תמיד נמצא למטה באמצע.

שימוש בקיצורי דרך – משתמש אדמיניסטרטיבי יכול לסמן את כל ה- URLs תחת term מסויים (לצורך מחיקה) בלחיצה אחת על Checkbox.

משוב אינפורמטיבי – בהזנת שאילתה לחיפוש, אם אין תוצאות מוצג למשתמש "No results found", כך שהוא יכול להבין שהרצת השאילתה הסתיימה (אין עוד למה לחכות).

דיאלוגים ברורים – בהזנת סיסמה שגויה למשתמש האדמיניסטרטיבי, מתקבלת ההודעה  
"Incorrect Password. Please try again", כך שברור שעליו להזין מחדש את הסיסמה.

מנע שגיאות – משתמשים לא יכולים להזין שאילתה ריקה לחיפוש (דבר שעלול להוות טריגר לשגיאה בריצת התוכנית או להביא לצריכה מיותרת של מסד הנתונים). המשתמש יקבל התראה על כך שהוא נדרש להזין טקסט והאתר לא יבצע את החיפוש.

אפשר היפוך קל של פעולות – משתמש אדמיניסטרטיבי יכול תמיד לעשות Recreate לאינדקס בלחיצת כפתור. כך, גם אם הוא מחק terms מסויימים שלבסוף הוא כן מעוניין בהם, הוא יוכל להחזיר אותם בקלות.

מוקד השליטה אצל המשתמש – משתמש יכול לכתוב שאילתה באופן חופשי ללא הגבלה על המילים, כמותם או שימוש  
ב- Capital Letters.

הפחת את עומס הזיכרון לטווח קצר – תוצאות שאילתת חיפוש מוצגות ככותרות העמודים ולא כ- URL. זה מאפשר למשתמש לזכור קישור שניגש אליו בעבר מבלי לזכור את כתובתו.

1. כדי לחשב את ציון ה- SUS עלינו:

* להחסיר 1 מהניקוד עבור שאלות אי-זוגיות
* להחסיר את הניקוד מ- 5 עבור שאלות זוגיות
* לסכום את הניקוד המתוקן של כל 10 השאלות עבור כל משוב שהתקבל
* להכפיל את הסכום ב- 2.5 עבור כל משוב שהתקבל
* לחשב את ממוצע הסכומים מכל המשובים

ציון ה- SUS שהתקבל: 85.4

בממוצע, ציון ה- SUS של מערכות הוא 68. הציון שקיבלנו הוא הרבה מעבר לממוצע, מה שמעיד על שביעות רצון של המשתמשים ועל כך שהם מוצאים במערכת שלנו שימושית, אינטואיטיבית ויעילה.

1. מדדים להצלחת המערכת:

Response Time – משך הזמן לקבל תגובה מהמערכת (למשל ביצירת שאילתת חיפוש) צריך להיות קצר ככל הניתן. משפיע על שביעות הרצון של המשתמש מהמערכת.

Security – המערכת נדרשת להבטיח הגנה על מידע רגיש (כמו האינדקס) מאנשים שאינם מורשים. פגיעה במידע רגיש היא פגיעה במערכת.

System Availability – זמינות המערכת היא דרישה בסיסית, שכן חוסר זמינות הופך אותה ללא רלוונטית עבור המשתמשים. מדובר במדד קריטי להבטחת שימושיות.

1. בפרויקט השתמשנו בארכיטקטורת Microservices. להלן דיאגרמה:



הסבר על הארכיטקטורה:

CrawlerService

מיקרו-שירות של זחלן שמחלץ דפי HTML סטטיים מ- Domain מסויים שמועבר אליו. בפרויקט מועבר אליו ה- Domain של AWS והוא מוגדר לחלץ כ- 200 דפים.

**is\_within\_domain** – בדיקה שהאתר עדיין חלק מה- Domain.

**contains\_non\_english\_codes** – בדיקה שכתובת האתר מכילה קוד של שפה שהיא אינה אנגלית (כך שנוכל לדלג על כתובת זו).

**contains\_non\_relevant\_keywords** – בדיקה שכתובת האתר מכילה מילות מפתח שלא רלוונטיות (כך שנוכל לדלג על כתובת זו).

**non\_relevant\_url** – שימוש בשלושת המתודות לעיל, על מנת לפסול כתובת במהלך הזחילה.

**normalize\_url** – מחיקה של Query Parameters ו- Fragments מכתובות במהלך הזחילה (כדי להימנע משכפול כתובות).

**crawl** – פעולת הזחילה שמתבצעת באלגוריתם BFS.

IndexCreatorService

מיקרו-שירות שאחראי על יצירת האינדקס. מאגד מידע טקסטואלי מתוך דפי HTML, ומבצע עליו עיבוד להפקת האינדקס. בפרויקט מקבל stop words מותאמים ל- AWS.

**create\_terms\_histogram** – יצירת היסטוגרמה שמכילה את תדירות ההופעות של מילים בתוך דף HTML מסויים.

**remove\_stop\_words** – הסרת מילים נפוצות שאינן מוסיפות ערך משמעותי להיסטוגרמה.

**normalize\_terms** – המרה של מילים בהיסטוגרמה לצורתן הבסיסית, ע"י שימוש ב- apply\_lemmatization שמבצע למטיזציה למושג מסויים, כדי למנוע כפילויות.

**remove\_low\_freqs** – הסרה של מושגים מההיסטוגרמה שהופיעו בתדירות נמוכה מהרף המוגדר (מוגדר בברירת מחדל  
ל- 7), כדי לצמצם את האינדקס למילים יותר משמעותיות.

**add\_term** – יצירת רשומה חדשה עבור מילה באינדקס אם היא אינה קיימת.

**add\_doc** – הוספת מידע על מסמך מסויים לרשומת מילה באינדקס.

**fetch\_doc\_title** – שליפת כותרת הדף (נשמר כחלק מהמידע על מסמך מסויים).

**create\_index** – יצירת האינדקס על ידי הפעלת שירות הזחלן לחילוץ הדפים, ושימוש במתודות המתוארות לעיל.

AdministrationService

מיקרו-שירות לניהול האינדקס.

**delete\_index** – מחיקת האינדקס הקיים מה- Database.

**recreate\_index** – שימוש בשירות יצירת האינדקס לצורך יצירת אינדקס מחדש (או לראשונה).

**fetch\_terms** – חילוץ כלל המושגים הקיימים באינדקס.

**fetch\_urls** – חילוץ הלינקים תחת מושג מסויים באינדקס.

**delete\_docs** – מחיקה של לינקים ספציפיים תחת מושג מסויים באינדקס.

**delete\_term** – מחיקה של מושג מסויים מהאינדקס.

DataFetcherService

מיקרו-שירות לשליפת מידע מה- Database.

**fetch\_index** – שליפה של כל האינדקס.

**fetch\_term\_docs** – שליפה של כל נתוני המסמכים עבור מושג מסויים.

StatisticsService

מיקרו-שירות לביצוע מניפולציות להפקת סטטיסטיקות וכן ליצירת גרפים בצורה גנרית.

**get\_most\_common\_words** – מבצע מניפולציה על האינדקס כדי להפיק את המושגים הנפוצים ביותר.

**get\_least\_common\_words** – מבצע מניפולציה על האינדקס כדי להפיק את המושגים הכי פחות נפוצים.

**get\_random\_words** – מבצע מניפולציה על האינדקס כדי להפיק את מספר הפעמים שמושגים רנדומליים בשימוש.

**get\_common\_docs** – מבצע מניפולציה על האינדקס כדי להפיק את המסמכים שבהם יש הכי הרבה מושגים.

QueryService

מיקרו-שירות לעיבוד שאילתה ושליפת התוצאות הרלוונטיות מהאינדקס.

**normalize\_query** – נרמול של השאילתה על ידי הפעלת apply\_lemmatization והמרה ל- lower case של כל מושג בשאילתה.

**url\_conatins\_terms** – בדיקה האם מושגי השאילתה נמצאים גם בכתובת (משפר את הדירוג).

**update\_maps** – עדכון של אינפורמציית העיבוד על מסמכים.

**calculate\_ranks** – חישוב הדרגות של המסמכים שנשלפו.

**fetch\_results** – מיון המסמכים לפי הדרגות ושליחתם.

**process\_query** – עיבוד השאילתה. מבצע שימוש במתודות המתוארות לעיל.

**3 – פיצ'ר**

בעמוד הרצת שאילתה, הוספנו פיצ'ר בשם "AWS Fun Fact" המציג בלחיצת כפתור באופן אקראי עובדה מעניינת על שירותי הענן של אמאזון. בכל לחיצה המשתמשים מקבלים עובדה אחרת. הפיצ'ר ממומש בקוד ה- GUI של עמוד הרצת השאילתה, ואין כרגע קישוריות לעמוד המנהל. אם יתאפשר במגבלות הזמן, קיימת האופציה להוסיף את העובדות לבסיס הנתונים ולאפשר למנהל שליטה על הוספה ומחיקה של עובדות אך בשלב זה הדבר אינו מתוכנן.