Scloud - report



מגישים: צוות Tiger

אליאור מליק 204361505  
אופיר שחף 314764283  
גיא בנבו 316139310  
ג'וליה שבאט 322240151  
מתן נחמוכה 318185956

יובל הילאי 318609237

חלק א':

1. מהות המוצר ומרכיביו – מודולים ופיצ'רים מעניינים.

מהות המוצר היא מנוע חיפוש עבור מונחי ושירותי ענן מתוך דפי האתר Alibaba cloud .

למערכת יש מספר מודולים ורכיבים:

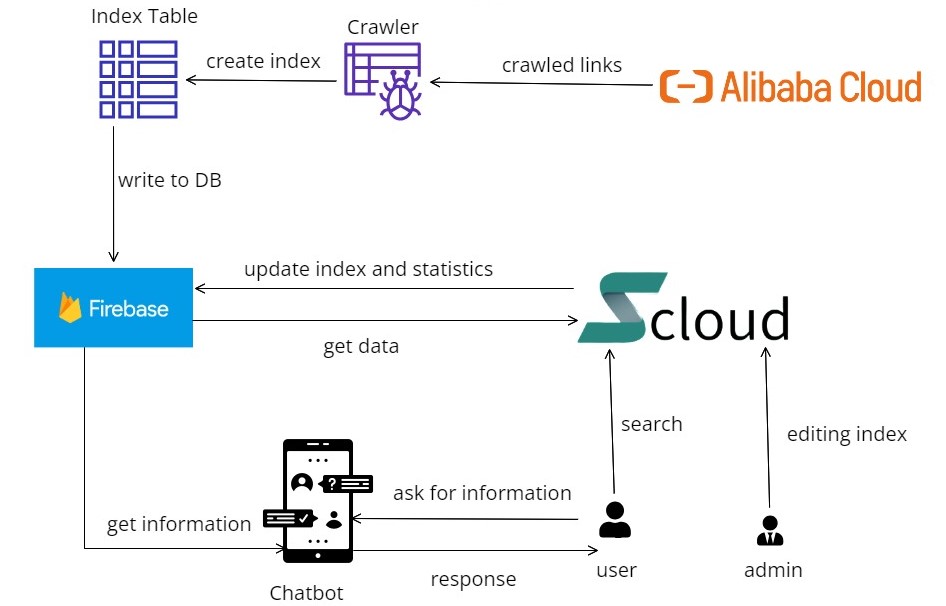
* מודול הcrawler- מודול זה אחראי על סריקת דפי אתר Alibaba Cloud ובניית האינדקס
* מסד נתונים בפיירבייס- משמש לאחסון האינדקס והדפים שנסרקו ומידע רלוונטי עבור הסטטיסטיקות
* מנוע החיפוש Scloud- מאפשר למשתמשים לבצע חיפוש של דפים באינדקס ולבצע פעולות נוספות כגון צפייה בסטטיסטיקה, עריכת אינדקס(אדמין), שימוש בצ'טבוט וכו'
* צ'טבוט- מודול זה מקבל קלט מהמשתמש ובהתאם לpatterns שהוגדרו מראש מחזיר תשובה וקישור רלוונטי למשתמש או שמבצע חיפוש עבור קלט שלא הוגדר בתבניות. התשובה המוחזרת מוצגת בממשק באתר מנוע החיפוש Scloud.

למערכת שלנו יש כמה פיצ'רים מעניינים עיקריים:

* בנוסף לחיפוש במלל חופשי, בדף הבית של המנוע החיפוש שלנו מופיעים מספר מונחים חשובים במחשוב ענן(כגון saas) שניתן לחפש אותם בלחיצת כפתור.
* בדף הבית מופיעים 5 החיפושים האחרונים של המשתמש והוא יכול לבצע חיפוש חוזר בלחיצת כפתור
* צ'טבוט- הצ'טבוט שלנו מאפשר לשאול שאלות כגון "what is saas" או לכתוב מונח לחיפוש, אם יש תוצאה הצ'טבוט ייתן את הקישור הכי רלוונטי עם תיאור קצר של הדף.

1. ארכיטקטורת המערכת – תרשים ומאפיינים מרכזיים.

תרשים ארכיטקטורת המערכת:



מאפיינים מרכזים:

* scalability: השימוש במסד נתונים של פיירבייס מאפשר לנו להגדיל את האחסון במידת הצורך, ללא הוספת חומרה פיזית אצלנו.
* maintainability: לצ'אטבוט שלנו יש ממשק בJavaScript שמשתמש בpatterns שהגדרנו בקוד בפייתון, אך בקלות ניתן לשנות אותו כך שיעבוד עם api אחרים כדוג' api של chatgpt.
* Portability: ניתן בקלות להעביר את המערכת כולה לריצה בסביבות אחרות. את הקרולר יהיה ניתן להריץ גם בסביבות נוספות הדומות לcolab שמסוגלות להריץ קוד בפייתון שדורש הרבה חישוב. בנוסף, כיוון שאת מנוע החיפוש scloud פיתחנו בJS ניתן בקלות להעבירו להרצה בדפדפן, עם שינוי קטן בממשק הJS של הצ'טבוט כך שיעבוד עם api שנתמך בJS כמו הapi של chatgpt, במקום בpatterns שהגדרנו בקוד פייתון.

1. מאפייני SOA מרכזיים של הפרויקט.

**Reusability** – שימוש באותו קטע קוד במקומות שונים למטרות שונות. בפרויקט שלנו אנחנו משתמשים בפונקציות לשליפת דפים המכילים מונח מסויים במסכים שונים בקוד ועבור מטרות- עבור מסך עריכת האינדקס, החיפוש והצ'טבוט.

**Interoperability:** שירותים שונים המעורבים בפעולת החיפוש צריכים להיות ניתנים להפעלה הדדית, ולאפשר תקשורת וחילופי נתונים חלקים. בזמן ביצוע פעולת החיפוש יש תקשורת וחילופי נתונים בין שירותי מנוע החיפוש לבין שירותי מסד הנתונים.

**Composability:** שירותים מפרקים בעיה גדולה לבעיה קטנה. בקוד שלנו שירות החיפוש מחולק למספר שירותים קטנים יותר- שירותים שעושים עיבוד למחרוזת השאילתא, שירות שאחראי על משיכת הדפים הרלוונטים לשאילתא, שירות שממיין את התוצאות ושירות שמעדכן את תצוגת התוצאות.

1. דרישות לא פונקציונליות - אתגרים איתם הפרויקט מתמודד (מסווגים לפי קישור WIKIPEDIA).

תוצאות החיפוש יוחזרו מהר למשתמש - Response time.

סדר התוצאות שיוצגו ייקבע לפי קטגוריה, כותרת, תיאור -usability .

החיפוש יתבצע לפי מלל חופשי, בחירה מתוך חיפושים חשובים או מתוך חיפושים אחרונים -usability .

עריכת האינדקס מתבצעת על ידי המנהל ומשתמשים לא יכולים לעשות זאת – security

החיפוש של הצ'אטבוט יתבצע באותה צורה שמתבצע החיפוש בשורת החיפוש הראשית – Reusability

1. תיק תחזוקה – תיעוד של כל הקבצים והאוביקטים המרכזיים, ותיעוד קצר של כל פונקציה בקוד.
2. **תיק מתכנת:**

**קבצים מרכזיים:**

1. קובץ HTML – קובץ זה מכיל את המבנה הכללי של המסך שמשמש בסיס לכל מסכי האתר. קובץ זה מכיל גם קישורים חשובים לסקריפטים המאפשרים חיבור למסד הנתונים שלנו בפיירבייס, בניית גרפים במסך הסטטיסטיקה וקישורים לתמונות ולוגואים שמשתמשים בהם בתצוגה של האתר ובכפתורים השונים.

בנוסף, בקובץ זה יש לנו div שמכיל את הצ'אטבוט של האתר וניתן לגשת אליו מכל המסכים.

1. קובץ CSS– קובץ זה מכיל את חוקי העיצוב של האלמנטים השונים באתר, בין אם נבנים דינמית או סטטיים
2. קובץ JS – קובץ זה מכיל את כל הלוגיקה והפונקציונליות של מנוע החיפוש, כולל תקשורת עם הפיירבייס לצורך חיפוש, עדכון ושליפה של דפים מהאינדקס, החלפת תוכן הhtml בצורה דינמית, בניית גרפי סטטיסטיקה, התחברות והתנתקות של האדמין, דפדוף בין מושגים באינדקס מסך המנהל וגם הגדרת הממשק של הצ'טבוט והתקשורת שלו עם הpatterns שכתובות בפייתון.
3. chatbot- חלק זה כתוב בפייתון ומגדיר את התבניות של שאלות שהצ'טבוט יכול לקבל ולענות עליהן. הקלט מתקבל דרך הממשק במנוע החיפוש והתשובה של הצ'טבוט מוחזרת אליו, אם השאלה/שאילתא לא מוגדרת בpatterns מתבצע חיפוש באמצעות הקוד JS.
4. crawler- בקובץ זה מוגדר הזחלן שסורק את דפי האתר alibabacloud.com לצורך בניית האינדקס. בנוסף מוגדרות פונ' להוצאת שמות ותיאורי הדפים לצורך הצגת התוצאות במנוע החיפוש, פונ' לסריקת תוכן הדפים(מונחים) לצורך בניית אינדקס ופונ' שונות לעיבוד מחרוזות כגון stemming.

בנוסף ישנם מספר ספריות חשובות שאנחנו משתמשים בהם בפרויקט ועושים להן import בתחילת הפרויקט כגון:nltk, beautifulsoup4, requests, firebase, ipython ועוד

**מסד נתונים:**

-במהלך הפרויקט בנינו מסד נתונים בfirebase . במסד הנתונים אנו שומרים את כל המידע הנדרש למנוע החיפוש במספר טבלאות.

הטבלה המרכזית שלנו היא index : האינדקס של מנוע החיפוש שבנוי בפורמט של inverse index כך שכל רשומה מורכבת ממונח(term) ו -DocIDsרשימת מזהי הדפים שמכילים את המונח.

טבלה נוספת היא טבלת pages המכילה את הדפים שהקרולר סרק, עבור כל דף שומרים את הלינק, שם הדף ותיאור הדף. מפתחות הרשומות בטבלה זו משמשות כמזהים של הדפים באינדקס.

בנוסף יש לנו טבלה נוספת עבור שמירת סטטיסטיקות מעניינות:

search\_index- מכילה את כל שאילתות החיפוש שבוצעו באתר ואת מספר הפעמים שבוצעו.

פונקציות מרכזיות:

JS:

homepageView- פונ' האחראית על אתחול מסך דף הבית

botView- אחראית על תצוגת הצ'טבוט של האתר

handleChat- אחראית על תפעול הצ'אט של הצ'אטבוט וקבלת קלט מהשתמש.

generateResponse- אחראית על יצירת תשובה מתאימה לקלט המשתמש בצ'אטבוט.

מתקשרת עם קוד הפייתון של הצ'טבוט שמכיל את הpatterns שהגדרנו ובמקרה הצורך מבצעת חיפוש באינדקס.

firebase.initializeApp- משמש ליצירת חיבור של קוד הjs לפיירבייס.

search- משמשת לביצוע חיפוש והצגת התוצאות

getCategoryForLink- משמש לסיווג לינקים לקטגוריות לפי מילות מפתח שהגדרנו ולפי הurl

sortSearchResults- משמש למיון תוצאות החיפוש עם עדיפות לקטגוריות מסוימות ובהתאם לתיאור ולשם הדף

createSearchResultElements- משמש ליצירת רשימה של תוצאות חיפוש להצגה

updateSearchCountInFirebase- פונ' האחראית על עדכון מונה מס' החיפושים של שאילתא מסוימת בDB לצורך סטטיסטיקה.

saveRecentSearch- משמש לשמירת עד 5 חיפושים אחרונים בlocalstorage לצורך הצגה בדף הבית.

updateRecentSearches- אחראית על עדכון 5 החיפושים האחרונים בתצוגת דף הבית

isLoggedIn- משמשת לבדיקה האם האדמין מחובר למערכת

setLoggedIn- משמשת לעדכון התחברות האדמין בlocalstorage

logout- משמש לניתוק המשתמש(אדמין) מהמערכת.

adminViewContent- משמש לטעינת מסך עדכון האינדקס של האדמין.

fetchIndexData- משמשת לשליפת האינדקס של מנוע החיפוש ממסד הנתונים בפיירבייס.

singleIndex- תפקידה להבטיח שהאינדקס נטען מקומית פעם אחת בלבד בתחילת ההרצה.

fetchPagesData- משמשת למשיכת רשימת הדפים מהDB.

singlepagesList- תפקידה להבטיח שרשימת הדפים נטענת מקומית פעם אחת בלבד בתחילת ההרצה.

fetchAndIntersectPages- פונ' זו מקבלת שאילתא ומחזירה את כל הדפים שמכילים את כל המילים בשאילתא.

fetchTermPages- פונ' זו מקבלת מונח(מילה בודדת) ומחזירה את כל הדפים שמכילים אותו.

createButtons- פונ' זו משמשת ליצירת כפתורי המונחים במסך עריכת האינדקס שבלחיצה עליהם מאפשרים לצפות בקישורים של המונחים השונים באינדקס.

createLinksList- משמשת ליצירת רשימת קישורים עבור מונח במסך עריכת האינדקס.

deleteLink- משמשת למחיקת קישור ממונח באינדקס. אם לא נשארו קישורים במונח לאחר המחיקה- גם המונח יימחק.

deleteTerm- משמשת למחיקת מונח ממסך עריכת האינדקס.

removeEntryFromIndex- מוחקת מונח מהאינדקס בDB.

displayLinks- משמשת להצגת קישורים של מונח במסך עריכת האינדקס

prevPage, nextPage- הפונקציות האלו משמשות לדפדוף בין המונחים במסך עריכת האינדקס.

fetchSearchIndexData- משמש לשליפת טבלת החיפושים search\_index ממסד הנתונים.

displaySearchIndexTable- משמש להצגת טבלת החיפושים במסך הסטטיסטיקות

createChart- משמש ליצירת תרשימים של כמויות החיפושים עבור מסך הסטטיסטיקה

linksCounter- משמש לחישוב מספר הקישורים עבור כל מונח.

updateTop10Terms- משמשת לחישוב 10 המונחים שמופיעים בהכי הרבה דפים והצגת 10 המונחים האלה בתרשים במסך הסטטיסטיקה.

createGraph(top10Terms)- משמש ליצירת התרשים של 10 המונחים עם הכי הרבה קישורים.

goToStatsPage- אחראית על המעבר ותפעול מסך הסטטיסטיקה. הפונ' טוענת את כל הנתונים והתרשימים שמוצגים במסך.

python:

**בקובץ crawler**:

crawl- זהו הזחלן שלנו שמשמש לאיסוף דפים באתר alibabacloud עד לעומק מסוים.

links\_enrichment- משמש לאיסוף מידע חיוני מהדפים שנסרקו- תיאור ושמות הדפים.

create\_index- משמש ליצירת אינדקס מהדפים שנסרקו.

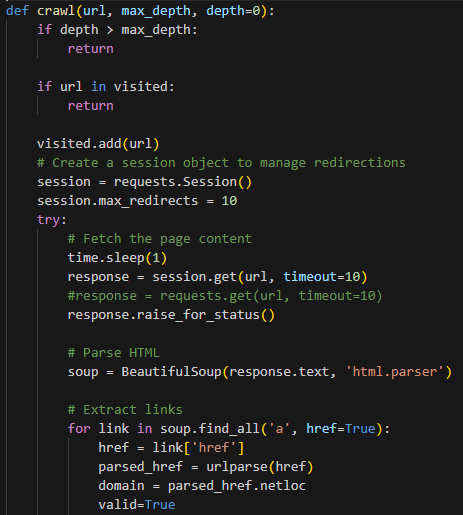
**chatbot**:

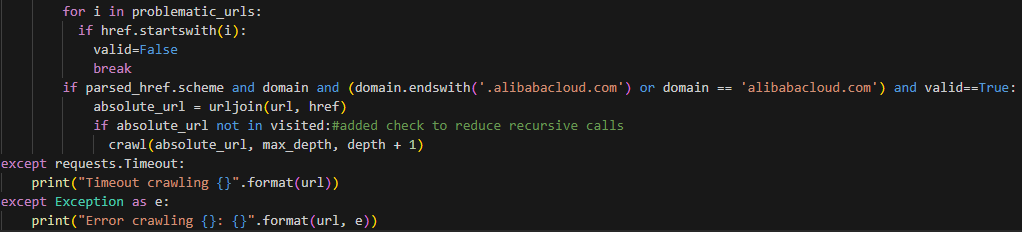
Chat- משמשת ליצירת מופע של הchatbot עם הpatterns שהגדרנו.

**קטעי קוד/תבניות עיצוב:**

1. קטע קוד מרכזי שלנו זה פונ' crawl שנמצאת בקובץ crawler.

פונ' זו שימשה אותנו לסריקת דפים באתר . alibabacloud הסריקה היא עד עומק שאנחנו מגדירים, כאשר שומרים סט של כתובות שביקרנו. מכל דף אנו מחלצים את הקישורים שנמצאים בו ועבור כל קישור דלא ביקרנו בו עדיין, אנו מבצעים קריאה רקורסיבית להמשך הסריקה.

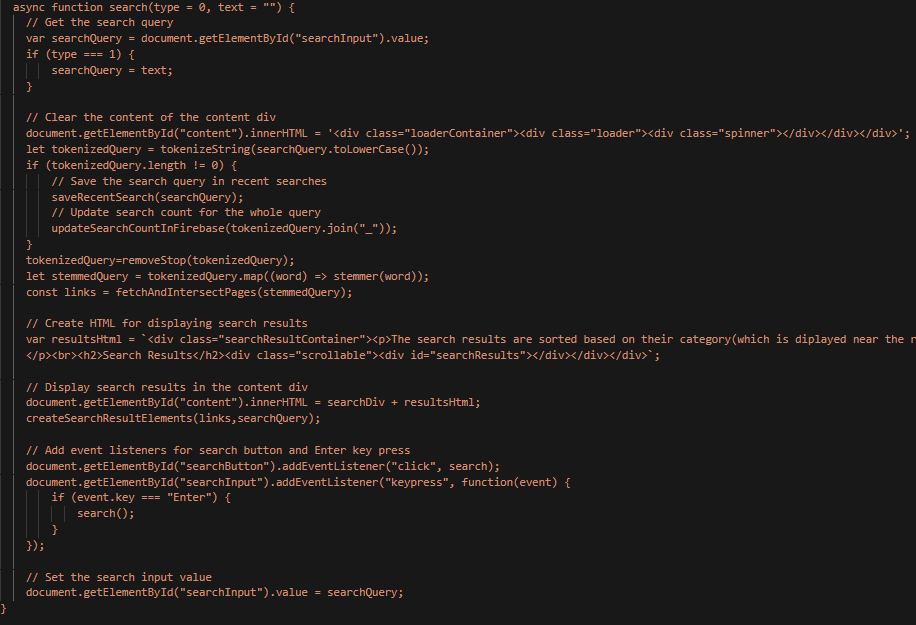




1. קטע קוד מרכזי היא פונ' החיפוש search שנמצאת בקובץ הJS

פונ' זו אחראית על תפעול ביצוע החיפוש של דפים באינדקס עבור שאילתה.

התוצאות שמתקבלות הן של דפים המכילים את כל המילים בשאילתא וממוינים בהתאם לקטגוריות ורלוונטיות השמות והתיאורים של הדפים לשאילתא.

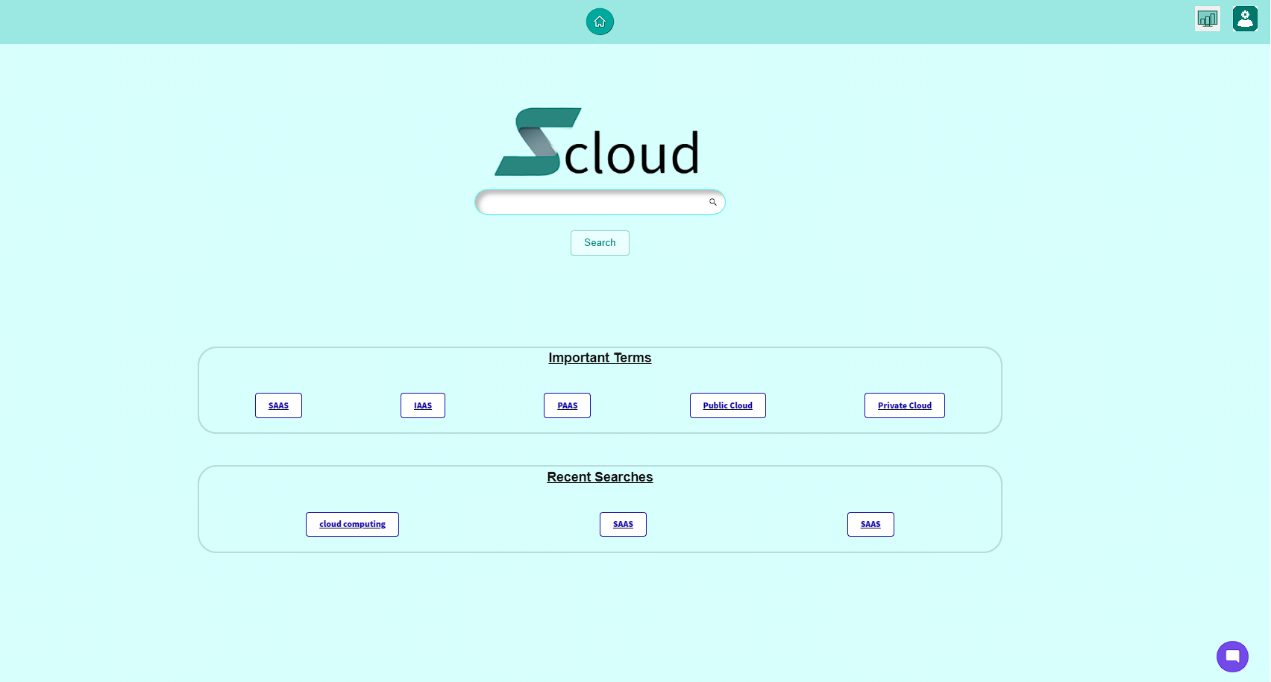


**תיק משתמש:**

מנוע החיפוש שלנו Scloud מאפשר לחפש דפים מתוך האתר alibabacloud.com שבחלקם יש מידע על מושגים ומוצרי ענן שונים של החברה.

ניתן לבצע עוד מספר פעולות בנוסף לחיפוש כגון צפייה בסטטיסטיקות מעניינות, עריכת אינדקס, שימוש בצ'אטבוט ועוד.

Homepage:

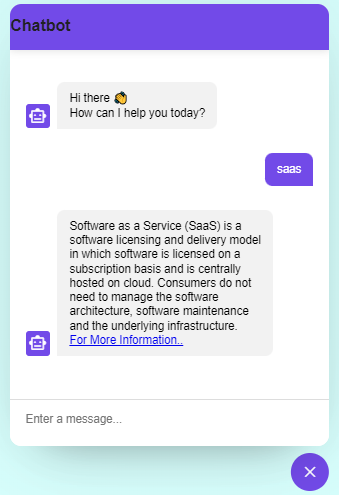
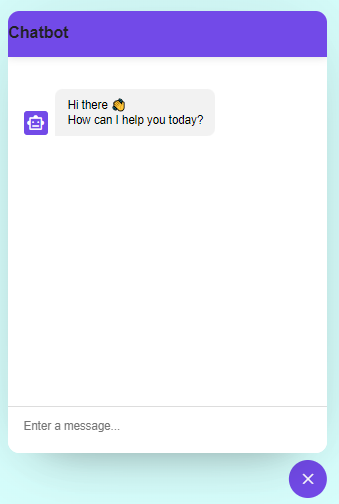


זהו המסך הראשי, מכאן ניתן לעבור לכל השאר המסכים, לבצע חיפוש ולגשת לצ'אטבוט.

החיפוש יכול להתבצע ע"י הזנת שאילתא בשורת החיפוש ולחיצה על search"" או על אנטר.

בנוסף, למשתמש מוצגים מושגים חשובים וחמשת החיפושים האחרונים שלו שהוא יכול לחפש באמצעות לחיצה עליהם.

הצ'אטבוט: ניתן לגשת לצ'אטבוט באמצעות לחיצה על הכפתור  שמופיע בכל המסכים ואז הוא נפתח



המשתמש יכול להזין שאילתא/שאלה בשורת הקלט, ללחוץ על  ולקבל מידע וקישור רלוונטי לשאילתא.

אם אין מידע על השאילתא שהוכנסה הchatbot יראה הודעת שגיאה בצ'אט.

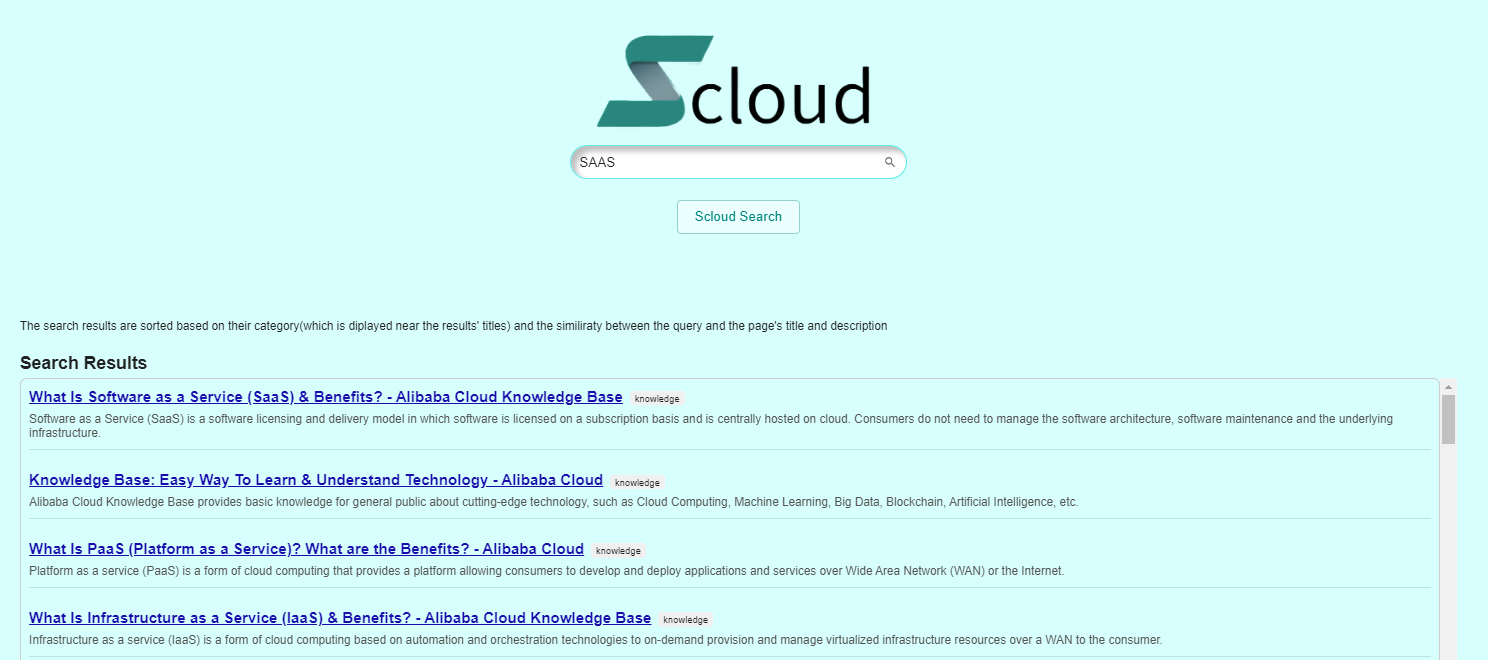
לחיצה על כפתור X תסגור את הצ'אט.

מסך תוצאות החיפוש:

במסך זה ניתן לראות את תוצאות החיפוש עבור שאילתא, כאשר עבור כל תוצאה מוצגים השם, הקטגוריה והתיאור של הדף.

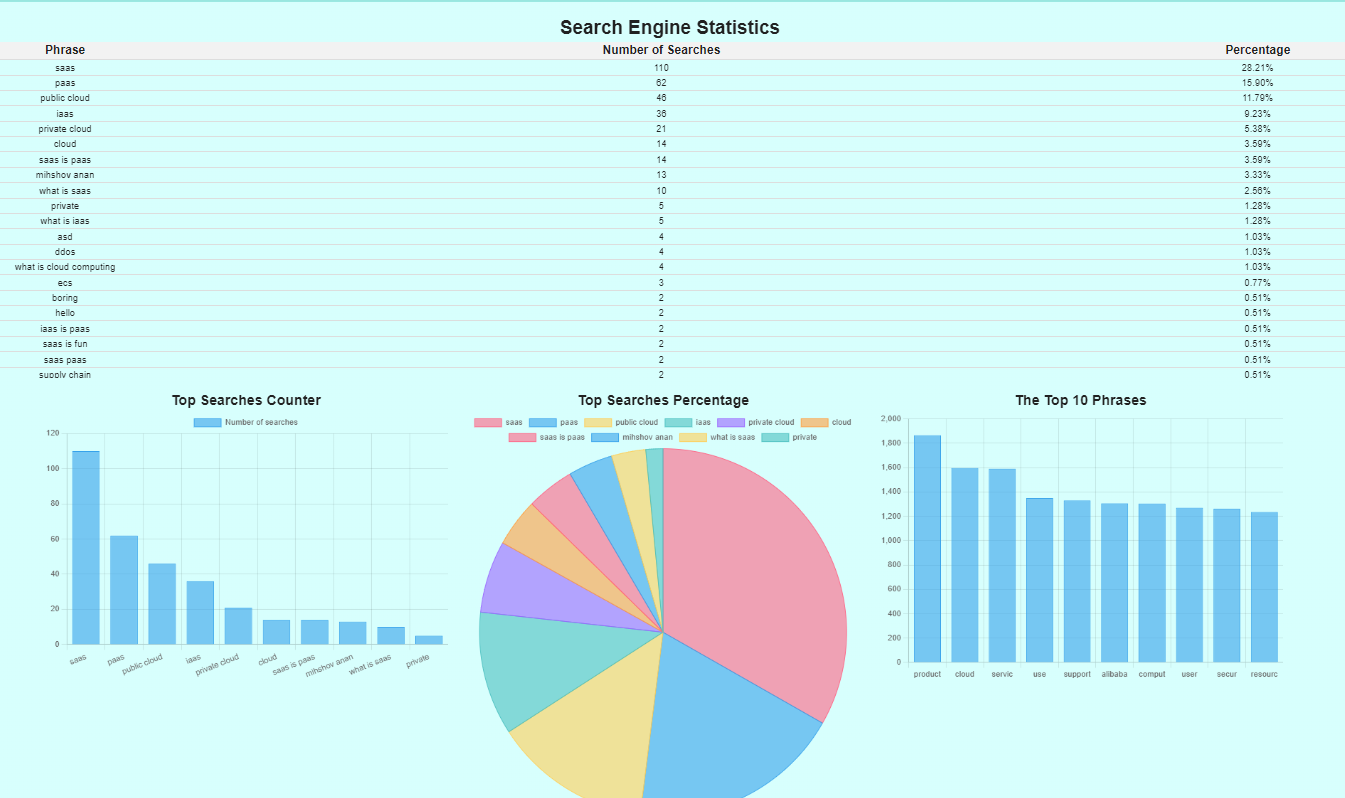
במסך זה ניתן לבצע חיפוש נוסף באמצעות שורת החיפוש.

אם המשתמש הכניס שאילתא עבורה אין תוצאות/ שאילתא לא תקינה(למשל רק רווחים) רשימת התוצאות שתוצג תהיה ריקה.



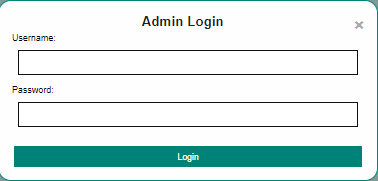
מסך הסטטיסטיקה:

באמצעות לחיצה על  ניתן לעבור למסך הסטטיסטיקה שבו ניתן לראות נתונים ותרשימים שונים כמו שאילתות החיפוש שבוצעו ומספר הפעמים שבוצעו, המונחים עם המספר הגדול ביותר של לינקים וכו'



מסך עריכת אינדקס(מסך מנהל)+התחברות אדמין:

ניתן להגיע למסך המנהל באמצעות לחיצה על .

אם האדמין לא מחובר ייפתח לו טופס התחברות בחלונית על המסך הנוכחי:

לחיצה על x תסגור את הטופס.

אם המשתמש יקליד סיסמה/שם לא נכון וילחץ על Login תקפוץ לו הודעה שמתריעה על כך. אחרת, אם יקליד את שם המשתמש והסיסמה נכונים (שם משתמש:admin, סיסמה: password123) וילחץ על login יועבר למסך עריכת האינדקס.

עריכת האינדקס-



בהתחלה בצד שמאל של המסך יוצגו לאדמין מושגים שונים מהאינדקס שיוכל לדפדף ביניהם באמצעות לחיצה על next וprev. אם ינסה להתקדם/לחזור אחורה כאשר הוא בדף האחרון או הראשון בהתאמה תקפוץ לו הודעה מתאימה.

בלחיצה על אחד מהמונחים תופיע רשימה של הקישורים של המונח במרכז המסך.

לחיצה על delete תמחק מהמושג את הקישור ולחיצה על delete all תמחק את המושג כולו.

לחיצה על  ממסך זה תפתח חלונית עם אפשרות להתנתקות באמצעות לחיצה על logout והאדמין ינותב לדף הבית

מכל מסך ניתן לחזור לדף הבית באמצעות לחיצה על .

גם המעברים למסך המנהל ומסך הסטטיסטיקה יכולים להתבצע מכל מסך באמצעות הכפתורים בtoolbar בחלק העליון של המסך.

1. ביקורת עמיתים אשר ניתנה במהלך הצגת הפרויקט, וכיצד התמודדתם איתה.

ההערות אליהן התייחסנו הן ההערות שקבלנו מחברי הכיתה בזמן שצפו בפרויקט שלנו ובנוסף קבלנו ההערות נוספות מהמשובים שנתנו במהלך הצגת הפרויקט לקראת סיום ביצוע חלק 2 של הפרויקט. חלק קטן מהתיקונים הספקנו לבצע לפני הגשת חלק 2 ואת שאר התיקונים בצענו בחלק השלישי של הפרויקט.

|  |  |
| --- | --- |
| הערת משוב | כיצד התמודדנו עם הביקורת |
| הגואי לא נעים לעין | רוב האנשים אמרו שזה כן היה נעים לעין, אבל שינינו קצת את הגוון שיהיה יותר נוח לעין |
| הצבע הירוק שברקע המסך הראשי פחות נעים לעין | שינינו קצת את הגוון שיהיה יותר נעים לעין |
| כפתור הצטבוט שיהיה יותר ברור איפה ללחוץ | הזזנו את הכפתור טיפה יותר למרכז שיראו אותו |
| עריכת אינדקס ללא האפשרות לחפש במסך האדמין היא קשה | לא שינינו, כי אנחנו מניחים שפעולת העריכה נעשית בתדירות נמוכה. |
| עיצוב הUI הוספת צבעים שונים למערכת ועיצוב הכפתורים | העדפנו להישאר עם עיצוב מינימלי על מנת להקל על המשתמש. |
| אולי לשפר עיצוב במסך סטטיסטיקות | שינינו קצת את הגודל של הסטטיסטיקות כדי שיהיה יותר נוח לצפייה |

1. אתגרים אשר עלו במהלך העבודה, וכיצד התמודדתם איתם.   
   יש להתייחס גם למטלת design thinking אשר ביצעתם בתרגיל בית יבש 1, ולבדיקת code review אשר מופיע מטה, בחלק ב.

* עבודת בסביבת google colab. בסביבה זו קשה לחלק את העבודה כיוון שלא ניתן לעבוד במקביל על אותו הקובץ. כדי להתמודד עם האתגר את רוב המימוש בצענו בסביבת עבודה אחרת ורק כאשר שלבנו פונקציונליות בצענו זאת ב - google colab.
* בתהליך design thinking בתרגיל הבית הראשון נתקלנו בקושי של מציאת רעיונות, שיפורים ופתרונות למנוע החיפוש כתוצאה משל הכרנו אותו. כדי להתמודד עם הבעיה נכנסנו אליו ובצענו בו חיפושים שונים כדי לנסות להבין איך הוא עובד ומה אנו רוצים לשנות. בנוסף השתמשנו גם במנועי חיפוש נוספים במטרה לראות כיצד תהליך החיפוש בהם מתבצע והאם ניתן לשלב חלקים מהתהליך במנוע החיפוש שלנו.
* חוסר ידע כיצד מתבצע החיפוש מנוע חיפוש. נתקלנו בקושי בפיתוח שיטת החיפוש של מנוע החיפוש שלנו מהסיבה שהידע שלנו בתהליך זה הוא לא מספיק. כדי להתגבר על קושי זה הגענו להרצאות ובצענו את התרגולים בנוסף ללמידה עצמית ושיתוף מידע בין חברי הצוות בנושא זה.
* זמן הרצה של חיפוש. כאשר הגדלנו את עומק החיפוש זמן ההרצה של הפעולה גדל באופן משמעותי. כדי להתמודד עם בעיה זו ניסינו להקדיש זמן רב לתכנון שיטת החיפוש בעומק קטן ולייעול שלה לפני שהתחלנו להגדיל את עומק החיפוש.
* אתגר מרכזי שעלה במהלך בדיקת הcode review זה אתגר של ניהול זמנים- היינו צריכים לעשות תרגיל זה לאחר סיום הסמסטר ובמהלך תקופת המבחנים הלחוצה והיה לנו גם קושי במציאת זמן להיפגש כקבוצה עבור ביצוע החלק הקבוצתי. התגברנו על אתגר זה בכך שתיאמנו מראש לבצע את התרגיל לאחר מועד הא' האחרון.

1. שקיפות אלגוריתמית – כיצד הבהרתם למשתמשים את האלגוריתמים והנתונים הנאספים בקוד שלכם?

שקיפות של אלגוריתם החיפוש – מעל תצוגת תוצאות החיפוש מופיעה פסקה המסבירה למשתמש את סדר התוצאות ולפי מה הוא נקבע.

שקיפות הנתונים – את השקיפות בנתונים יכול המשתמש לראות בעזרת הגרפים השונים שמציגים סטטיסטיקות עבור נתוני החיפוש. הסטטיסטיקה פתוחה לכולם.

1. קישור למחברת הגוגל קולאב (יש לוודא שהיא פומבית) ולתיקיית הגיט של הצוות

קישור לקולאב: <https://colab.research.google.com/drive/1rQEMVxe5N4duTzX5ytdx3IB8gXDpujyn?usp=sharing>

קישור לגיטהאב(הקובץ נמצא בתיקייה (Final Project:

<https://github.com/guybanbo/Cloud-computing-exercises.git>

חלק ב':

|  |  |
| --- | --- |
| ***צוות בודק***  **tiger** | ***צוות נבדק*** alligators |

חלק ראשון- עבודה קבוצתית

הבדיקות:

**1.Login System Acceptance Test**

**Admin Input**: Enter admin credentials (username and password).

Expected Output: The system should grant access and redirect to the index page, displaying edit/add/delete buttons to modify the index list, indicating admin-level permissions.

**User Input:** Enter regular user credentials (username and password).

Expected Output: The system grants access and redirects to the index page without showing edit/add/delete buttons, indicating user-level permissions and restricting the ability to modify the index list.

**Error Input:** Enter incorrect username or password.

Expected Output: The system displays an "Incorrect info" error message and does not grant access to the index page, prompting the user to try logging in again.

**2. Search History Function Acceptance Test**

**View Search History:** A user navigates to the search page after performing several searches.

Expected Output: The search page displays a list of previous searches that the user can view and select from.

**3. Chatbot Assistance Acceptance Test**

**Queries Chatbot for Help:** A user types a question into the chatbot interface, asking for assistance with navigating the website.

Expected Output: The chatbot provides a helpful and accurate response that guides the user or solves the user's query.

**4. Information Extraction Automation Acceptance Test**

**BackEnd:** The backend system runs the developed or integrated tool/script to extract information from the target site.

Expected Output: The system successfully extracts the required information and stores it for use on the website, demonstrating efficiency and accuracy.

**5. Dynamic Content Loading Acceptance Test**

**User Interacts with Dynamically Updated Content:** A user visits a page that includes dynamic content (e.g., news updates).

Expected Output: The page loads content dynamically without needing a page refresh, showing up-to-date information and enhancing user interaction. each acceptance test:-

6. **Viewing Statistics page Acceptance Test**: A user visits the statistics page.

expected output: the statistics are displayed correctly

7. **Searching a query Acceptance Test:**

a user performs several searches

expected output: a list of relevant links is displayed to the user

|  |  |
| --- | --- |
| test | Result |
| Admin login | Passed- the admin logged in successfully |
| Regular user login | Passed- the user logged in successfully |
| incorrect username or password | Passed- an error message was displayed |
| View Search History | Passed- after searching a term, the search history was updated and the user was able to view it |
| Queries Chatbot for Help | Passed- after entering a query to the chatbot, relevant information was provided to the user |
| Information Extraction Automation | Passed- The system successfully extracts the required information and stores it for use on the website |
| Dynamic Content Loading Acceptance | Passed- The page loads content dynamically without needing a page refresh |
| Viewing Statistics page | Passed- the statistics are displayed correctly |
| Searching a query | Passed- a list of relevant links is displayed to the user after searching a query |

כיוון שכל הטסטים שהוגדרו עברו בהצלחה, ניתן להסיק כי הקוד עמד בכל היעדים.

***חלק שני: עבודה אישית***

שם הסטודנט:\_\_\_גיא שלמה בנבו\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| **קריטריון** | **הסבר** |
| מימוש | כן, הקוד עונה על מטרת הפרויקט- הוא מאפשר לבצע חיפוש, לנהל את האינדקס, לשאול את הצ'אטבוט שאלות ולראות סטטיסטיקות רלוונטיות. |
| יעילות | לדעתי אין צורך לייעל את הקוד, הכל עבד בצורה מהירה |
| פשטות | לא, לדעתי הקוד הוא יחסית פשוט ומובן |
| מודולריות | כן, לדעתי הקוד מספיק מודולרי- יש חלוקה לפונקציות ולמחלקות, והמחברת מחולקת למספר תאים |
| תבניות עיצוב | לא הבחנתי בשימוש בתבניות עיצוב בקוד |
| OOP | כן, יש שימוש במחלקות. |
| באגים וטעויות | לא נתקלתי בבאגים בהרצת הקוד |
| טיפול בשגיאות | יש התראות על שגיאות משתמש, למשל בעת ההתחברות לאתר מופיעה הודעה אם השם משתמש/סיסמא לא נכונים.אין שימוש במנגנון תפיסת שגיאות בקוד. הבדיקה והדיבוג נוחים כיוון שיש חלוקה של הקוד למספר תאים ושימוש במחלקות. הודעות המשתמש ברורות וידידותיות. |
| בדיקות | לדעתי כדאי להוסיף בדיקות הקשורות למסך האינדקס. |
| שימושיות | הממשק שימושי וידידותי וגורם לי לרצות לחזור להשתמש בו. |
| תיעוד | כן, הקוד מתועד |
| אתיקה | אין בעיות של אתיקה בקוד |
| אבטחה | כן, שמות המשתמשים והסיסמאות גלויים כבר במסך הראשי של ההתחברות |
| ביצועים | לדעתי שינוי עתידי בקוד לא יפגע בביצועים |
| ביצועים | לדעתי לא ניתן לשפר את הביצועים |
| קריאות - readability | הקוד קריא ומובן בקלות |
| זמן תגובה- כמה זמן לוקח לפעולות להתבצע ולהחזיר תגובה למשתמש? | הקוד מגיב מהיר וביעילות, כאשר משתמש מבצע פעולות שונות כולל חיפוש הוא מקבל תגובה בזמן מהיר |
| שקיפות אלגוריתמית- האם מוסבר למשתמש איך אלגוריתם החיפוש עובד? | כן, יש מעל תוצאות חיפוש משפט המסביר לפי מה מויינו התוצאות. |

**שאלות פתוחות**:

1. *כיצד הרגשת במהלך החוויה של מעבר על הקוד?*

*הרגשתי במהלך החוויה של מעבר על הקוד שהצוות שכתב את הקוד השקיע בו רבות וניסה לבנות מנוע חיפוש שייתן למשתמשים חווית שימוש טובה*

1. *האם לדעתך ניתן לשפר תרגיל זה?*

*לדעתי היה ניתן לשפר תרגיל אם היו נותנים דוגמאות לתשובות אפשריות עבור כל קריטריון שמופיע בטבלה בחלק האישי כדי שיהיה אפשר להבין בצורה קלה יותר את התרגיל.*

1. *הערות נוספות*

אין לי הערות נוספות.

שם הסטודנט: אליאור מליק

|  |  |
| --- | --- |
| **קריטריון** | **הסבר** |
| מימוש | כן, הקוד מבצע את הנדרש, ניתן לבצע חיפוש שאילתות ולקבל עבורם דפים הקשורים באתר vmware, צפייה בסטטיסטיקות, קבלת מידע מהצ'אט בוט עבור מונחים הקשורים למחשוב ענן, עריכת וצפייה בדפי האינדקס. |
| יעילות | הקוד יעיל ועובד בצורה מהירה, הקבוצה בחרה לממש את הקוד כולו בpython שעובד בצורה מעולה בגוגל קולאב. |
| פשטות | הקוד כתוב בצורה פשוטה וקלה להבנה. |
| מודולריות | כן, הקוד מחולק למחלקות כמו התחברות,צ'אט בוט שמאפשר עבודה מודולרית, בנוסף החלקים הנוספים של הקוד מחולקים לפונקציות. |
| תבניות עיצוב | לא הבחנתי כלל בשימוש בתבניות עיצוב בקוד |
| OOP | כן, יש שימוש במחלקות כמו התחברות,צ'אט בוט. |
| באגים וטעויות | לא נתקלתי בבאגים בהרצת הקוד |
| טיפול בשגיאות | יש התראה על שגיאות משתמש, לדוגמא כאשר מנסים להתחבר עם שם משתמש או סיסמא לא נכונים מופיעה הודעת שגיאה מתאימה למשתמש. אין שימוש במנגון try-catch בקוד, הבדיקה של הקוד נוחה מכיוון שהקוד מחולק למחלקות,פונקציות ותאים קוד שונים,הודעות המשתמש ברורות ומצביעות על הכוונה\שגיאה בצורה טובה. |
| בדיקות | ניתן להוסיף בדיקות הקשורות למסך האינדקס, בדיקה הקשורה לעריכת האינדקס. |
| שימושיות | כן, הממשק שימושי ומאפשר שימוש וניווט בצורה נוחה, ומעורר רצון להמשיך להשתמש במערכת. |
| תיעוד | כן, הקוד מתועד ועבור כל פונקציה ומחלקה יש הסבר מה היא עושה. |
| אתיקה | אין בעיות של אתיקה בקוד |
| אבטחה | כן, שמות המשתמשים והסיסמאות גלויים כבר במסך הראשי של ההתחברות |
| ביצועים | לא, הקוד מחולק למחלקות ופונקציות מה שמאפשר שינויים עתידיים בצורה יותר מבוקרת, בנוסף כתוב כולו בפייתון מה שמאפשר ביצועים טובים יותר בגוגל קולאב. |
| ביצועים | לא, הביצועים הנוכחיים של המערכת הם טובים והמערכת תגובתית עם זמן תגובה נמוך. |
| קריאות - readability | הקוד קריא וניתן להבנה, חלוקת הקוד לפונקציות ומחלקות מאפשרות קריאות טובה יותר של הקוד. |
| מדרגיות – האם המערכת תתאים לצמיחה במידע ועומס? | לדעתי המערכת עלולה להגיב לא טוב כאשר נגדיל את המידע שבו המערכת משתמשת מכיוון שהמערכת רצה על גוגל קולאב שהיא בסביבת דפדפן ועלול להיווצר חיוות משתמש לא טובה כאשר הדפדפן יצטרך לעבד כמות גדולה של מידע. |
| ניידות – האם הקוד יכול לרוץ בקלות בסביבות עבודה שונות? | כן, הקוד כולו כתוב בפייתון, לכן ניתן להריץ אותו גם לוקלאית או בכל שירות ענן אחר שתומך בפייתון. |

**שאלות פתוחות**:

1. *כיצד הרגשת במהלך החוויה של מעבר על הקוד?  
   במהלך המעבר על הקוד ראיתי שנעשתה השקעה בכתיבת הקוד, הייתה גם חשיבה על העיצוב הכללי של המערכת כמו מעבר בין טאבים שמאפשר גישה מהירה לקטוגוריות שונות באתר.*
2. *האם לדעתך ניתן לשפר תרגיל זה?  
   לדעתי לא, התרגיל מסכם בצורה טובה את סקירת הקוד ומאפשר גם להוסיף חלק אישי שאינו קיים בבדיקה על מנת לכלול כמה שיותר קטגוריות בבדיקה.*
3. *הערות נוספות*

אין לי הערות נוספות.

שם הסטודנט:\_\_\_ג'וליה שבאט\_\_\_

|  |  |
| --- | --- |
| **קריטריון** | **הסבר** |
| מימוש | כן, הקוד מבצע כל מה שנדרש בהנחיות הפרויקט |
| יעילות | הקוד יעיל מבחנתי ולא רואה שצריך לשפר |
| פשטות | לא, הקוד פשוט ומובן ואין קטעי קוד מסובכים |
| מודולריות | הקוד מסודר ומודולרי. יש חלוקה ברורה לפונקציות ולמחלקות, והמחברת מחולקת לתאים בצורה מסודרת ונוחה לעבודה. |
| תבניות עיצוב | לא הצלחתי לראות שימוש בתבניות עיצוב בקוד. |
| OOP | כן |
| באגים וטעויות | במהלך כל ההרצות שעשיתי(הרבה) לא נתקלתי בבאגים או בטעויות שהשפיעו על תפקוד הקוד. |
| טיפול בשגיאות | קיימות התראות נוחות על שגיאות משתמש. אין בשימוש בקוד מנגנון אוטומטי לתיקון שגיאות, אך ההודעות ניתנות לקריאה בקלות ומובנות. |
| בדיקות | כן, ניתן ליצור בדיקות שבודקות את יכולת המערכת לזהות, לתפקד ולתקן שגיאות שונות שעלולות להתרחש בזמן ריצה. |
| שימושיות | הממשק נראה מאוד שימושי ונוח לשימוש אפילו למשתמש פעם ראשונה משתמש בו ולא יודע כלום עליו קודם. |
| תיעוד | הקוד מצויד בתיעוד מספיק שמסייע להבנת הפונקציונליות והמבנה שלו. |
| אתיקה | לא נתקלתי בשום דבר שעלול לפגוע באתיקה של הקוד. |
| אבטחה | כן, שמות המשתמשים והסיסמאות גלויים(של האדמין ומשתמש רגיל)במסך ההתחברות. |
| ביצועים | אני לא רואה צורך מיוחד בשיפור ביצועי הקוד במצבו הנוכחי. |
| קריאות - readability | הקוד נראה מסודר וקל לקריאה. |
| Scalability | הקוד יכול להתקדם באופן טוב אם יש צורך להרחיב את הדאטה או להוסיף משתמשים נוספים. |
| Robustness | הקוד מתמודד באופן טוב עם קלטים לא תקינים ומספק הודעות שגיאה ברורות וגדולות למשתמש. |

**שאלות פתוחות**:

1. *כיצד הרגשת במהלך החוויה של מעבר על הקוד?*

*במהלך חוויתי קריאה נעימה ויעילה של הקוד, שהייתה מובנית ומאורגנת היטב. מדובר במשימה שמסרתה לי תובנות חדשות ומעוררת עניין לעתיד, ומספקת פלטפורמה נהדרת ליישום רעיונות ויכולות בעתיד, בראש ובראשונה בעבודה עם גוגל קולאב.*

1. *האם לדעתך ניתן לשפר תרגיל זה?*

*לדעתי היה ניתן לשפר תרגיל לו הוסיפו דוגמאות איך התשובות נראות ולתת לנו דוגמה איך אנחנו נתן משוב מה צריך להדגיש כשבודקים פרויקט ואיך להסתכל על השאלות והתשובות.*

1. *הערות נוספות*

*לא, אין לי.*

שם הסטודנט: יובל הילאי

|  |  |
| --- | --- |
| **קריטריון** | **הסבר** |
| מימוש | כן, הקוד מבצע את הנדרש: יש עמוד חיפוש עמוד אינדקס סטטיסטיקות וצאט בוט |
| יעילות | כן, ניתן לשפר את יעילות הקוד, ולעשות אותו יותר מהיר. |
| פשטות | לא ראיתי משהו שאפשר לפשט |
| מודולריות | הקוד כן מודולרי כי יש חלוקה של פונקציות |

|  |  |
| --- | --- |
| תבניות עיצוב | ניתן אולי לפשט את העיצוב ואת המהירות של החיפוש |
| OOP | יש שימוש במחלקות. |
| באגים וטעויות | אין בבאגים בהרצת הקוד |
| טיפול בשגיאות | הקוד יודע לטפל בשגיאות הודעות המשתמש ברורות וידידותיות. |
| בדיקות | כדאי להוסיף בדיקות הקשורות למסך האינדקס. |
| שימושיות | הממשק שימושי וידידותי וגורם לרצות לחזור להשתמש בו. |
| תיעוד | הקוד מובן בקלות אך ניתן לתעד יותר - מתועד חלקית |
| אתיקה | אין הפרה באתיקה |
| אבטחה | כן, שמות המשתמשים והסיסמאות גלויים כבר במסך הראשי של ההתחברות |
| ביצועים - גמישות | לפי דעתי הוספה של דבר לא יהווה פגיעה בעתיד בקוד בשל החלוקה לפונקציות |
| ביצועים | מאמין שאפשר לשפר את הסיבוכיות של חלק מהפונקציות על מנת לייעל תהליכים ולשפר מהירות |
| - קריאות readability | הקוד מובן בקלות אך ניתן לתעד יותר |
| מהירות וזמן תגובה | מבחינת מהירות – החיפוש עובד יחסית איטי |
| סיבוכיות |  |
|  | מבחינת סיבוכיות – יש מקומות שהקוד רץ על o(n^2) אולי ניתן להפחית. |

שאלותפתוחות:

1. -הרגשתי שהקוד מבצע את הנדרש אבל מבחינת הממשק פחות ידידותי למשתמש, ולא נראה טוב בעין כל כך.
2. -לא ניתן לשפר את התרגיל.
3. הערות נוספות – אין

שם הסטודנט: מתן נחמוכה 318185956

|  |  |
| --- | --- |
| קריטריון | הסבר |
| מימוש | סך הכל הקוד הנבדק מבצע את כל מה שצריך, בין אם מדובר בפעולת חיפוש שאילתות, צפייה סטטיסטיקות, צ'אט בו, אופציות אדמין. |
| יעילות | אין צורך לבצע שינויים, הקוד מסודר ופשוט. בנוסף הקוד כתוב בפייתון, עובד מעולה בשילוב עם הקולאב. |
| פשטות | הקוד מסודר ובעל הירככיה ברורה, דבר שעוזר להבנה שלו ולפשטות. |
| מודולריות | הקוד מודולרי, בין אם במחברת אשר יש חלוקה לתאים שונים ובין אם חלוקה בקוד לפונקציות שונות, אשר יוצרות "רכיבים" שונים. |
| תבניות עיצוב | לא מצאתי שימוש. |
| OOP | יש שימוש במחלקות כחלק מהעקרונות OOP |
| באגים וטעויות | בהרצת הקוד ומעבר לא מצאתי באגים וטעויות. |
| טיפול בשגיאות | יש הודעת שגיאה במסך ההתחברות כאשר מתחברים באמצעות פרטי התחברות לא נכונים.  אין try-catch.  יש שימוש נכון בהודעות שגיאה. |
| בדיקות | ניתן להוסיף בדיקות נוספות, לדוגמא במסך האינדקס. |
| שימושיות | הממשק נוח, אך פשוט. למרות הפשטות מעורר רצון להמשיך להשתמש במערכת. |
| תיעוד | הקוד מתועד בצורה טובה וברור מה המטרה של כל פונקציה. |
| אתיקה | אין בעיות אתיקה. |
| אבטחה | שמות המשתמשים והסיסמאות מופיעים במסך הראשי של ההתחברות |
| ביצועים | הקוד מסודר והיררכי, דבר שתורם לעבודה קלה בעתיד עם שינויים. |
| ביצועים | המערכת מציגה ביצועים טובים עם תגובה מהירה לפעולות המשתמש. |
| קריאות - readability | הקוד מסודר ומתועד, דבר שמקל משמעותית על הקריאות שלו. |
| נוחות שימוש | המערכת בעלת UI מסודר ומינימלי, דבר שעוזר למקד את המשתמש באופציות הקיימות ומאפשר שימוש אינטואיטיבי בהן. |
| Quality | איכות הקוד באופן כללי טובה, יש תיעוד וסדר בקוד, חלוקה לפונקציות והירככיה. |

**שאלות פתוחות**:

1. *כיצד הרגשת במהלך החוויה של מעבר על הקוד?  
   במהלך המעבר על הקוד ראיתי שנעשתה השקעה בכתיבת הקוד, בין אם בתיעוד או בסידור שלו. הייתה גם חשיבה על העיצוב הכללי של המערכת כמו מעבר בין קטגוריות בסרגל מהיר שמאפשר גישה מהירה לקטגוריות שונות באתר. כן חושב שהיה מקום ליותר עיצוב שכולל HTML.*
2. *האם לדעתך ניתן לשפר תרגיל זה?  
   לדעתי, התרגיל מסכם בצורה טובה מאוד את הCODE REVIEW ומאפשר הוספת חלק אישי לבדיקה כדי לכלול כמה שיותר קטגוריות.*
3. *הערות נוספות*

סך הכל נהניתי לעבור על פרויקט של קבוצה שונה וללמוד ממנו עוד גישה שונה מהגישה של הקבוצה שלנו לבנית המערכת, לדוגמא נושא העיצוב שפה בוצע בעזרת WIDGETS בשונה מאיתנו.

**שאלות פתוחות**:

1. *כיצד הרגשת במהלך החוויה של מעבר על הקוד?  
   במהלך המעבר על הקוד ראיתי שנעשתה השקעה בכתיבת הקוד, בין אם בתיעוד או בסידור שלו. הייתה גם חשיבה על העיצוב הכללי של המערכת כמו מעבר בין קטגוריות בסרגל מהיר שמאפשר גישה מהירה לקטגוריות שונות באתר. כן חושב שהיה מקום ליותר עיצוב שכולל HTML.*
2. *האם לדעתך ניתן לשפר תרגיל זה?  
   לדעתי, התרגיל מסכם בצורה טובה מאוד את הCODE REVIEW ומאפשר הוספת חלק אישי לבדיקה כדי לכלול כמה שיותר קטגוריות.*
3. *הערות נוספות*

סך הכל נהניתי לעבור על פרויקט של קבוצה שונה וללמוד ממנו עוד גישה שונה מהגישה של הקבוצה שלנו לבנית המערכת, לדוגמא נושא העיצוב שפה בוצע בעזרת WIDGETS בשונה מאיתנו.

שם הסטודנט: אופיר שחף

|  |  |
| --- | --- |
| **קריטריון** | **הסבר** |
| מימוש | הקוד מבצע את המשימות שנתנו לפי הנחיות הפרויקט. ניתן לבצע חיפוש לראות סטטיסטיקות לערוך עמודי אינדקס ולדבר עם צ'אטבוט. |
| יעילות | מהרצת התוכנית ומעבר על הקוד נראה שהוא כתוב בצורה יעילה ולכן עובד כך. |
| פשטות | הקוד מחולק למחלקות ולכן הוא פשוט וקל להבנה. |
| מודולריות | נראה שהקוד מחולק לפונקציות ומחלקות בצורה טובה וחכמה מה שמאפשר לקוד עבודה מודולרית בצורה טובה. |
| תבניות עיצוב | לא מצאתי בקוד תבנית עיצוב שבאה לידי ביטוי באופן מספיק בולט לכן קשה קצת להתייחס לאופן המימוש. |
| OOP | האם ישנו שימוש נכון בעקרונות OOP? האם ניתן לשפר?  הקוד מורכב ממחלקות שמקושרות ביניהן בצורה טובה ונכונה לפי עקרונות OOP |
| באגים וטעויות | בהרצת הקוד לא ראיתי תקלות או באגים שהופיעו לי במהלך ההרצה |
| טיפול בשגיאות | ישנה התראת שגיאה בהתחברות לא נכונה אך לא הבחנתי במנגנון תפיסת שגיאות בקוד. כתוצאה מחלוקת הקוד למחלקות בצורה נכונה, בדיקת הקוד הייתה נוחה לביצוע. הודעות המשתמש אכן ברורות וידידותיות. |
| בדיקות | האם יש בדיקות שניתן להוסיף?  לא עולות לי בדיקות ספציפיות נוספות שאפשר לבצע בקוד. אולי במסך האינדקס ששם יש הרבה פונקציונליות כדאי לבצע בדיקה נוספת. |
| שימושיות | הממשק אינו מסובך להפעלה והפשטות המינימלית גורמת לחווית שימוש טובה ונוחה ומעוררת רצון להמשיך להשתמש במנוע החיפוש. |
| תיעוד | הקוד מתועד וההסבר על הפונקציות השונות מובן היטב |
| אתיקה | ממעבר על הקוד נראה שאין הפרה של כללי האתיקה השונים. |
| אבטחה | במסך ההתחברות ניתן לראות בצורה גלויה שמות משתמש וסיסמאות. |
| ביצועים | האם לדעתך שינוי עתידי בקוד יכול לגרום לפגיעה בביצועים?  כתוצאה מחלוקת הקוד בצורה טובה לפונקציות נפרדות הוא גמיש ושינויים עתידיים שייעשו בו אינם יגרמו לפגיעה בביצועים של הפונקציות האחרות. |
| ביצועים | לדעתי כתיבת הקוד כולו בפיתון נותנת למערכת רמת ביצועים גבוה מאוד. |
| קריאות - readability | הקוד מחולק למחלקות ופונקציות בצורה טובה וברורה והפונקציונליות של כל קטע מוסברת היטב ולכן הוא מובן וקל לקריאה. |
| מימוש/קריאות – שם פונקציות ושמות משתנים | שמות המשתנים והפונקציות תקינים נכונים ובשילוב תיעוד הפונקציה עוזרים מאוד ותורמים להבנת הפונקציות השונות. |
| מימוש – גודל וסיבוכיות פונקציות | רוב הפונקציות בקוד אינן גדולות במיוחד והחלוקה ביניהן מאוד ברורה. הפונקציות אינן מסובכות ובשילובה תיעוד גם מאוד קלות להבנה. |

**שאלות פתוחות:**

1. *כיצד הרגשת במהלך החוויה של מעבר על הקוד?*

*בתחילת המשימה חששתי מאוד לעבור על קוד שאני לא מכיר למרות שאני יודע פחות או יותר מה הוא עושה. תוך כדי המעבר על הקוד הרגשתי בטחון בהבנה של מה הוא עושה ומאוד התעניינתי לראות איך קבוצה אחרונה כתבה קוד שעושה פעולות זהות לקוד שלנו אבל כתוב בצורה מאוד שונה. הרגשתי שהמשימה חשובה ותורמת לי הקרבה.*

1. *האם לדעתך ניתן לשפר תרגיל זה?*

*לדעתי תרגיל זה נותן לנו לחוות את תהליך בדיקת הקוד בצורה טובה מאוד. הלמידה העצמאית של תהליך בדיקת הקוד ומה בדיוק משמעות הקריטריונים מאוד מאתגרת לדעתי ואלי כדי שנבצע את תהליך הבדיקה בצורה מיטבית אפשר בנוסף לקישור בתחילת המשימה המכיל הסברים בלבד להוסיף דוגמה עם קטעי קוד כדי שההבנה תהיה מדויקת יותר או סרטון שמסביר את הכוונה.*

1. *הערות נוספות:*

*אין לי הערות נוספות מלבד ההצעה לשיפור בסעיף הקודם.*