מסמך פרויקט

קבוצת Shark

מבוא למחשוב ענן

1. מהות המוצר ומרכיביו:

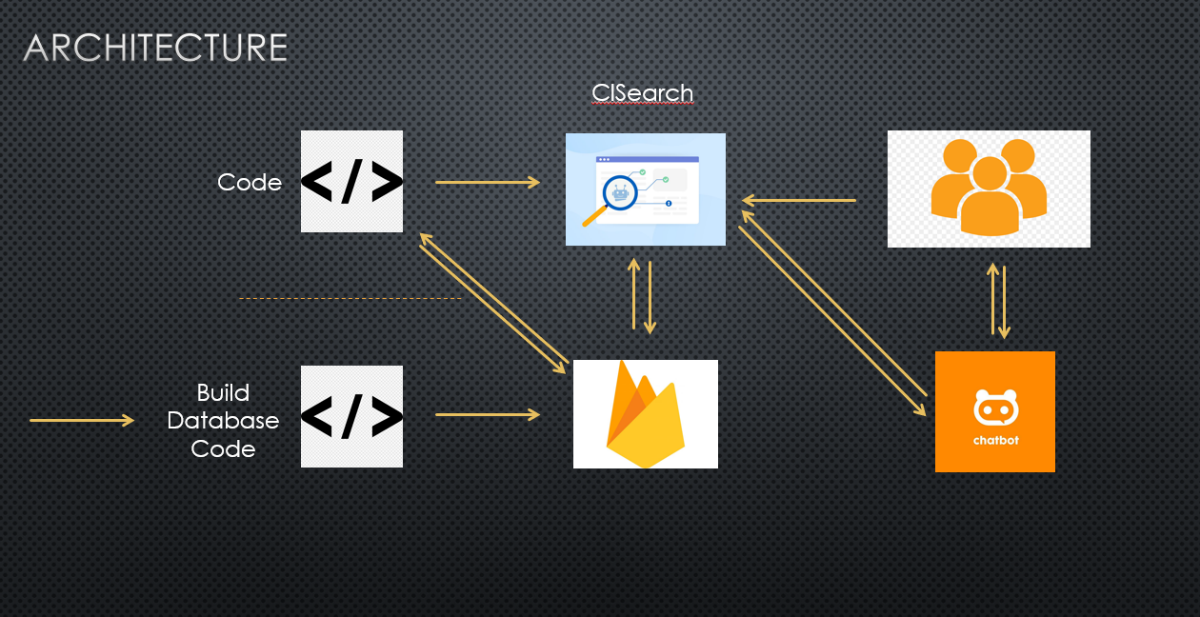
מדובר במנוע חיפוש שמתבסס על IBM Cloud, שמאפשר למשתמש לחפש דברים שמעניינים אותו, והוא מקבל תוצאות עם פירוט קטן למהות האתר וכמה הוא חשוב (יש חישוב של חשיבות האתר).

משתמש יכול להירשם/לכנס למערכת, כך שלמשתמש יש יכולת לגשת לדברים נוספים כמו chatbot, ויש להם סטטיסטיקות משלהם.

כל משתמש יכול לערוך את האינדקס שלו (יש אינדקס אחר לכל משתמש – זה תלוי בחיפושים שהמשתמש מבצע), האינדקס מאותחל לכל המשתמשים כאותו אינדקס – מסתמך על מילים חשובות ומשמעותיות באתר, ואחר כך האינדקס מתעכדן עם כל חיפוש שהמשתמש עושה.

מסך הסטטיסטיקה לגולשים שאין להם חשבון מראה את הלינקים המשתמעותיים באתר, בנוסף לכך כל משתמש יש לו דף סטטיסטקה משלו שמראה משך זמן ההתחברות שלו למערכת ב 7 ימים האחרונים, ומראה גם כמה פעמים נכנס לקישורים שונים.

1. ארכיטקטורת המערכת:



1. מאפייני SOA מעניינים של הפרויקט:

נדבר על כמה מאפייני SOA שיש לנו בפרויקט:

* Flexibility: שירותי SOA נועדו להיות ניתנים לשימוש חוזר במגוון יישומים ותהליכים עסקיים. זה מקדם גמישות בכך שהוא מאפשר למערכות שונות למנף אותו שירות למטרטת שונות.
* Abstraction: שירות SOA מסתיר מהצרכן את הטכנולוגיה הבסיסית שלו, המשתמש לא צריך לדעת או לראות מה קורה מאחורי הקלעים, אכפת לו לקבל מידע מדויק ומהיר.
* Autonomy: לשירותים יש שליטה על ההיגיון שהם מכילים, מנקודת מבט של לקוח אין צורך לדעת על היישום שלהם.
* Composability: שירותי SOA יכולים להיות מורכבים ומתוזמרים כדי ליצור יישומים חדשים, זה מאפשר גמישות בהתאמה לצרכים העסקיים המשתנים על ידי שילוב שירותים קיימים בדרכים חדשות.

הפרויקט שלנו מחולק לקבצים ונעזר בשירותי ענן שונים:

* ClSearch.ipynb: הקוד המהותי שלנו, זה הקוד של המערכת שלנו.
* DB.ipynb: הקוד שבונה את מאגר המידע שלנו (database).
* Firebase Firestore: מסד נתונים של מסמכי NoSQL שנבנה עבור קנה מידה אוטומטי, ביצועים גבוהים וקלות פיתוח יישומים.
* Google Colab: סביבה מבוססת ענן שמאפשרת לחברי הצוות לכתוב והלריץ קוד בפייתון ומאפשרת לזימון מחברות אחרות/קבצים אחרים ושונים.

1. דרישות לא פונקציונאליות:
   1. ביצועים: המערכת צריכה לתת תגובה מהירה שלא תעלה על 2 שניות כדי לא לעכב את המשתמש.
   2. סקלביליות: המערכת צריכה להתמודד עם מספר משתמשים גדול בו זמנית ללא השפעה על הביצועים.
   3. אבטחה: המערכת תממש אבטחה כדי להגן על פרטי הלקוח והנתונים שלו.
   4. נגישות: המערכת תבטיח שהממשק יהיה נגיש וידידותי למשתמש.
   5. זמינות: להבטיח זמינות של 99.9% של הפלטפורמה.
2. תיק תחזוקה:

* להלן פירוט הקבצים המרכזיים:

Clsearch.ipynb: קובץ הקוד המהותי של המערכת, זה הקוד שמכיל את העיצוב, והלוגיקה של המערכת

DB.ipynb: הקובץ שבונה את מאגר הנתונים שלנו (בונה את ה database).

cloudproject-1bbde-firebase-adminsdk-849hb-47666b58c8.json: הקובץ שמייצר את התקשורת עם Firestore Database.

* להלן תיעוד הפונקציות:

query\_word: הפונקציה הזו מאחזרת מידע הקשור למילה ספציפית מאוסף מסד הנתונים, מחפשת מידע התואם לשאילתה הנתונה, ומחזירה את הנתונים המתאימים.

create\_index: הפונקציה הזו מאתחלת אינדקס עם קבוצה מוגדרת מראש של מילים. תהליך אתחול זה מסייע בהפחתת מורכבות הזמן של החיפוש על ידי מתן נקודת התחלה לאינדקס.

update\_index: פונקציה זו אחראית לעדכון האינדקס במילים חדשות ובקישורים הרלוונטיים שלהן. אם משתמש מקליד מילה שאינה קיימת באינדקס, פונקציה זו מוסיפה את המילה זו מוסיפה את המילה יחד עם הקישורים המשויכים לה.

login: פונקציה זו מאמתת את שם המשתמש והסיסמה שסופקו כנגד נתוני משתמש מאוחסנים במסד הנתונים. הוא בודק אם המשתמש קיים ואם הסיסמה שסופקה תואמת לסיסמה המואחסנת. בנוסף, מנהלת את מצב ההתחברות ומעדכנת מידע רלוונטי לאחר כניסה מוצלחת.

logout: פונקציה זו מטפלת בתהליך הוצאת המשתמשים מהמערכת. מעדכנת את סטטוס המשתמשים כדי לציין שבוצעה יציאה, בנוסף היא מתעדת את זמן ההתנתקות ומחשבת משך הזמן שבו שהה המשתמש במערכת.

signup: פונקציה זו מקלה על תהליך ההרשמה עבור משתמשים חדשים. מטפלת ביצירת חשבונות משתמשים, אימות נתוני קלט ושמירת מידע משתמש במסד הנתונים, בנוסף הפונקציה מאתחלת את נתוני הפעלה של המשתמשים למעקב אחר פעילות המשתמש.

calculate\_rank: פונקציה זו מחשבת את הדירוג עבור כל עמוד בהתבסס על ספירת עמופעים של מילה מסוימת. הוא עושה טרנפורמציה לוגרתמית על ספירת ההתרחשויות ולאחר מכן מחשבת את הדירוג באמצעות ערכים שעברו טרנפורמציה.

count\_query\_occurrences: פונקציה זו מקבלת את התוכן של קישור נתון וסופרת את המופעים של מילים משאילתה שביצע המשתמש.

get\_documents\_id: פונקציה זו מאחזרת רשימה של מזהי מסמכים מאוסף שצוין במסד הנתונים.

display\_user\_times: פונקציה זו מאחזרת ומציגה נתוני זמן משתמש ממסד הנתונים עבור 7 הימים האחרונים. משרטטת את זמן השהיית המשתמש במהלך השבוע האחרון וגם משרטטת ספירות קישורים להדמיה טובה יותר.

count\_links: פונקציה זו מאחזרת מסמכים ממסד הנתונים וסופרת את המופעים של כל קישור קיים.

get\_user\_id: פונקציה זו מאחזרת את מזהה המשתמש של המשתמש המחובר כעת על ידי שאילתה במסד הנתונים של Firestore עבור משתמשים עם 'isLoggedIn' מוגדר ל-1.

calculate\_total\_senconds: פונקציה זו מחשבת את הזמן הכולל של משתמש בין אירועי התחברות ליציאה ומעדכנת את מסד הנתונים של Firestore בנתוני הזמן המצטברים.

get\_logout\_times: פונקציה זו מאחזרת זמני יציאה עבור משתמש ספציפי ממסד הנתונים.

get\_login\_times: פונקציה זו מאחזרת זמני כניסה עבור משתמש ספציפי ממסד הנתונים.

generate\_user\_id: פונקציה זו יוצרת מזהה משתמש ייחודי עבור משתמשים חדשים במערכת.

create\_line\_plot\_with\_makers: פונקציה זו יוצרת עלילת קווים עם סמנים המייצגים את הזמן שבו משתמש משתמש בתאריכים שונים.

display\_statistics: פונקציה זו יוצרת תרשים עוגה המציג את מופעי הקישורים שנוספו לאינדקס.

handle\_new\_link: פונקציה זו מנהלת הוספת קישור חדש לאינדקס המשויך למילה שנבחרה.

handle\_new\_item: פונקציה זו מנהלת הוספת מילה חדשה והקישור המשויך לה לאינדקס.

remove\_word\_from\_firestore: פונקציה זו מוחקת מסמך המתאים למילה נתונה ממסד הנתונים.

extract\_index\_data\_from\_firestore: פונקציה זו מאחזרת נתוני אינדקס מ-firestore, מחלצת מילים והקישורים הקשורים אליהן.

generate\_index\_table\_rows: פונקציה זו יוצרת שורות טבלת HTML עבור כל מילה באינדקס יחד עם הקישורים הייחודיים המשויכים לה.

fetch\_page: פונקציה זו מאחזרת את תוכן ה-HTML והקישורים מכתובת אתר שצוינה.

remove\_stop\_words: פונקציה זו מבטלת מילות עצירה משאילתה נתונה, ומשפרת את הרלוונטיות והדיוק של תוצאות החיפוש.

fetch\_page\_iterative: פונקציה זו מאחזרת קישורים של דפי אינטרנט באופן איטרטיבי, עד לעומק שצוין, תוך הימנעות מתלות מחזורית ומבטיחה שכל קישור חוקי ובטוח לביקור.

build\_index: פונקציה זו בונה אינדקס הפוך מאוסף של דפי אינטרנט. הוא מתייחס למילים רישיות ואותיות קטנות כשוות ערך ואינו כולל תווים שאינם מילים כדי ליצור אינדקס מנורמל.

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

המשוב מקבוצות אחרות על הפרויקט שלנו היה חיובי, חלק מהקבוצות אהב את מה שעשינו והעריך את העבודה שלנו. ממה שהבנו מהתגובות זה שהעיצוב שלנו לא היה הכי יפה (היה אפשר להיות יותר טוב), אבל מה שחשוב זה שהוא נוח והמשתמש לא צריך להתאמץ כדי להבין את הממשק ולתקשר איתו.

המשובים שקיבלנו עבור אלגוריתם החיפוש היו מחזקים, חלק מהמשובים התייחס לזה שזמן מתן התוצאות היה קצת איטי, שזה נכון אגב, וזה נובע מגודל מסד הנתונים שלנו, יש לנו מסד נתונים מאוד ענק, וזה מעכב את זמן מתן בתגובות, נקודה שאפשר לעבוד עליה ולשפר אותה.

היה תגובות על זה שאין לנו Dark mode במערכת, וזה גם נכון, לא הספקנו להוסיף את האפשרות הזאת למערכת.

כל הנקודות האלה אפשר לשפר בעתיד, לגבי זמן התגובה אפשר לקרוא ולחפש איך מזרזים חיפושים במסדי נתונים אדירים, אולי לשלב חוטים במערכת שלנו שכל חוט משתלט ומחפש באיזור מסויים במסד הנתונים, וחוט אחר יחפש באיזור אחר, וכך זמן החיפוש יהיה יותר מהיר ואז תגובה יותר מהירה למשתמש.

אפשרות ה Dark mode היא פשוטה ואפשר להוסיף אותה למערכת בקלות.

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

* אחד האתגרים שפגשנו זה עמידה בזמני הגשה, היה לנו סמסטר מאוד עמוס (כל חברי הצוות), והיה לנו קשה לשלב בין הדברים, לכך יש דברים שבאו בחשבון של דברים אחרים, אפילו הגשנו אחת המטלות באיחור. השתדלנו לחלק עבודה הוגנת בין חברי הצוות כדי לעמוד בזמנים שהוקצאו לנו.
* הבנת מה צריך לעשות בעבודות במהלך הסמסטר, היה לנו קשה להבין את העבודות במהלך הסמסטר, הניסוח של עבודות היה חסר ויש דברים שאינם מוסברים היטב. כדי לפתור את הבעיה הזו ולהמשיך בעבודה, ביקשנו עזרה מסטודנטים אחרים, או מצוות הקורס.
* עיצוב המערכת ובניית הלוגיקה, זה גם אתגר שצריך לזכור, ברגע שיש 4 מוחות שחושבים, אז יהיה קשה לשלב ולאחד את המחשבות להסכים על דבר אחד שבסוף נלך לפיו. כל אחד יש לו רעיון לעיצוב, יש לו רעיון למימוש פונקציונאליות מסויימת. תמיד היינו מסתכלים על כל הצצדים ומנסים להגיע להסכמה לפי הרוב.

1. שקיפות אלגוריתמית – כיצד הבהרתם למשתמשים את האלגורתמים והתנונים הנאספים בקוד שלכם?

הצגת סטטיסטיקות מראה שקיפות אלגורתמית, וגם הצגת הניקוד של דף מוסים זה מראה שקיפות אלגורתמית למשל, יש גם הסבר לדירוג התוצאות (למה הדף הזה ראשון וכו...).

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

ClSearch (Google Colab): <https://drive.google.com/file/d/1aAoqQg2QGWDzBGQeq2UG37cR6IFT1Fdb/view?usp=sharing>

DB (Google Colab): <https://drive.google.com/file/d/1bFzFgHeKJbRARnAog_CvBuIGi3GtUIR2/view?usp=sharing>

Git: <https://github.com/KhalidSwaid/CloudCourse.git>

1. מקורות. יש לצטט את המקורות השונים מהם אתם לוקחים את הנתונים השונים:

מצגות ותרגולי הקורס

<https://chat.openai.com/>

<https://www.google.com/>

<https://www.ibm.com/cloud>

חלק ב:

***בדיקות קבלה - Acceptance Test:***

|  |  |
| --- | --- |
| Passed/Failed | Acceptance Test |
| 1. Pass✅ 2. Pass✅ 3. Pass✅ | 1) Verify that all planned features of CISearch are implemented and functional.  2) Test the search functionality with a diverse set of queries to ensure that the results are relevant and accurate.  3) Ensure that the search algorithm do not adversely affect system performance. |
| 1. Pass✅ 2. Pass✅ 3. Pass✅ 4. Pass✅ | 4) Demonstrate that the new search algorithms have been integrated into CISearch.  5) Evaluate the effectiveness of the algorithms by conducting searches with complex queries  6) Measure the performance (speed and resource utilization) of the new algorithms to ensure they do not degrade the system’s overall performance.  7) Collect and analyze user feedback on search relevance and accuracy to confirm improvements. |
| 1. Pass✅ 2. Pass✅ | 8) Verify team alignment by reviewing meeting minutes and ensuring that all action items are addressed promptly.  9) Measure the impact of these activities on project progress and team morale through surveys or feedback sessions. |
| 1. Pass✅ 2. Pass✅ 3. Pass✅ 4. Pass✅ | 10) Compile and categorize user feedback on the UI/UX.  11) Implement changes based on prioritized feedback.  12) Conduct usability testing with a subset of users to validate that the changes address the feedback effectively.  13) Assess user satisfaction through surveys or direct feedback after the improvements have been made. |
| 1. Pass✅ 2. Pass✅ | 16) Benchmark the current backend performance and scalability.  17) Simulate high-traffic scenarios to validate scalability improvements. |
| 1. Pass✅ 2. Pass✅ 3. Pass✅ | 18) Verify the implementation of new search filters and their functionality across different search queries.  19) Ensure that result categorization works accurately with various types of search queries, reflecting the relevant categories for the search results.  20) Conduct user testing to gather feedback on the usefulness and intuitiveness of the new filters and categorization. |
| 1. Pass✅ 2. Pass✅ | 21) Document any discrepancies or issues and ensure they are resolved.  22) Validate the fixes and re-test to ensure no regression in functionality. |
| 1. Pass✅ 2. Pass✅ 3. Pass✅ 4. Pass✅ | 23) Ensure that the project timeline is detailed, with clear milestones for each key deliverable.  24) Verify that the timeline is realistic and accounts for dependencies between tasks.  25) Confirm that all stakeholders have reviewed and agreed upon the timeline and milestones.  26) Testing all the user-interface, buttons, screens, tabs, etc. |

חלק אישי:

הקבוצה הנבדקת: Salamandra

שם הסטודנט: ח'אלד סוויד 324264530

|  |  |
| --- | --- |
| קריטיון | הסבר |
| מימוש | הקוד מבצע את הנדרש ואפילו מבצע יותר ממה שנדרש. |
| יעילות | הקוד יעיל, מחולק לקבצים , אין כמעט שחזור פונקציות, ניתן לשפר את מהירות החיפוש. |
| פשטות | קוד פשוט וקריא, ניתן להבנה |
| מודולריות | הקוד מחולק נכון, השייכות של קטעי קוד מובנת, תיעוד לפונקציות ולקטעי קוד חשובים. |
| תבנית עיצוב | לא היה צורך בשימוש בתבניות עיצוב כל כך, אפשר בלי. |
| OOP | המימוש של ה chatbot זה נחשב ל OOP. |
| באגים וטעויות | הקוד עובד כצפוי, כל כפתור עושה את העבודה שלו ואת הפונקציונאליות שלו נכונה, ממבט ראשון על הקוד ועל המערכת נראה שאין באגים. |
| טיפול בשגיאות | יש התראה על שגיאות משתמש, התראה על הכנסת שם משתמש וסיסמה לא תקינים, או במידה שהמשתמש אינו קיים במערכת.  עדכון אינדקס (הוספה/מחיקה) עובד טוב ויש התראות על דברים שהמשתמש יכול לעשות, למשל לוחץ על הוספת DocId ללא לחיצה על term, קופצת התראה למשתמש. |
| בדיקות | אין צורך בהוספת עוד בדיקות. |
| שימושיות | המערכת מאוד שימושית, מעוררת השראה לשימוש, מושכת את המשתמש. |
| תיעוד | קיים תיעוד בקוד, תיעוד של פונקציות, תיעוד של קטעי קוד חשובים, אפילו יש תיעוד לקטעי קוד רגילים גם. |
| אתיקה | מבחינת אתיקה, אין שימוש וניצול בנתוני המשתמש חוץ מהצורך להתחברות למערכת. |
| אבטחה | יש אבחטה עד רמה מסוימת, לא מדובר באבטחה נגד תקיפות, אבל נגיד רק המשתמש יכול לדעת את הסיסמה שלו, זה גם אבטחה. |
| ביצועים | לא אמורה להיות בעיה בעת הוספת קטעי קוד בעתיד, יכול להיות מצב שיש לעדכן פונצקיות, אבל זה לא בעיה. |
| ביצועים | תמיד אפשר לשפר את הביצועים בדרך או אחרת. |
| קריאות-readability | הקוד מובן, גם שימוש במשתנים מובן וברור, בנוסף יש תיעוד לכל דבר בקוד. |
| GUI | העיצוב מטורף, יש השקעה מאוד רצינית בעיצוב, נוח לעין ומושך |
| Compatibility | המערכת עובדת תקין לסביבות שונות |

שאלות פתוחות:

1. המעבר על הקוד היה מאתגר, מעניין לראות דרך חשיבה של קבוצה אחרת, והביצוע של הדברים.
2. כן ניתן לשפר, אולי היה אפשר לשלב את החלק האישי בזוגות, ואז לעשות חלוקת דיעות.

שם הסטודנט: נבי מרעי 316197490

|  |  |
| --- | --- |
| קריטיון | הסבר |
| מימוש | האם הקוד מבצע את הנדרש?  כן, הקוד מבצע את כל הנדרש וכן הוא פועל כמו שצריך |
| יעילות | האם ניתן לשפר את יעילות הקוד?  לפי דעתי הקוד מספיק יעיל, חישובים פשוטים ואין שכפול קוד. |
| פשטות | האם ניתן לפשט את הקוד?  ניתן לפשט את קוד יותר, קיים שימוש במשתנים גלובאליים כדי להעביר מידע בדרך אחרת. |
| מודולריות | האם הקוד מודולרי מספיק? (חלוקה תקינה לקבצים, פונקציות ומחלקות)  כן, חלוקה נכונה וקלה להבנה, כל פונקציה אחראית על פונקציונאליות מסוימת של המערכת. יש חלקים בקוד שמכילים קריאות ל -HTML וכלן עדיין ניתן לחלק בצורה יותר טובה. |
| תבנית עיצוב | האם ישנו מימוש ברור ונכון של תבניות עיצוב?  כן יש מופע אחד של חיבור למסד נתונים (Singleton). |
| OOP | האם ישנו שימוש נכון בעקרונות OOP? האם ניתן לשפר?  בקוד שלהם אין יישום מפורש של עקרונות תכנות מונחה עצמים OOP. |
| באגים וטעויות | האם ישנם מקרים בהם הקוד לא מתנהג כצפוי? אילו?  הכל עובד כצפוי, לכל לחיצה מקבלים את התגובה שמצפים אליה, לא הצלחתי למצוא באגים או מקרים שבהם הקוד לא עובד כראוי. |
| טיפול בשגיאות | האם ישנה התראה על שגיאות משתמש? האם ישנו שימוש במנגנון תפיסת שגיאות? האם הבדיקה ודיבוג הקוד נוחים? האם הודעות המשתמש ברורות וידידותיות למשתמש?  בכל מקום שבו אפשר לקבל שגיאה בקוד מוקמת הודעה מתאימה שיכולה להופיע על המסך כדי להזהיר את המשתמש. |
| בדיקות | האם יש בדיקות שניתן להוסיף?  לא |
| שימושיות | האם הממשק שימושי? (usability)? האם המשחק מעורר רצון להמשיך במנוע?  כן, הצצשק יפה ומעוצב וקל לתפעול ואכן מעורר רצון להמשיך. |
| תיעוד | האם הממשק מתועד?  הממשק מתועד בצורה מעולה, מעל קטעי קוד יש הסבר ברור לגבי מה הוא עושה ואיך.  יש כמה שורות שהייתי מתעד שיהיה יותר ברור. |
| אתיקה | האם יש שימוש בנתוני משתמש אשר מפרים פרטיותו? האם יש ניצול של התנהגות או חולשה אנושית? האם ישנו algorithmic bias כנגד קבוצה או פרטים מסוימים?  אין שימוש |
| אבטחה | האם ישנו מידע אבטחה (שמות משתמש, ססמאות) גלויים?  במעבר שלי אנו רואה שיש אבטחה, רק המשתמש עצמו יכול לראות את סיסמתו. |
| ביצועים | האם לדעתך שינוי עתידי בקוד יכול לגרום לפגיעה בביצועים?  לפי מה שאני רואה הוספת קטעי קוד ושינוי לא יאטו את הביצועים אך תלוי מה קטע הקוד שמוסיפים ומה גודלו. |
| ביצועים | האם ניתן לשפר לדעתך את הביצועים?  תוצאות החיפוש וסטטיסטיקות מופיעות בצורה מהירה ומעבר בין מסכים גם מתבצע בצורה מהירה לכן לדעתי אין מה לשפר. |
| קריאות-readability | האם הקוד מובן בקלות? אילו חלקים בקוד היו לא ברורים הוברך? כיצד ניתן לשפר את קריאת הקוד?  קוד מובן ופשוט, יש אפשרות לתיעוד נוסף שיהיה יותר ברור. |
| GUI | האם העיצוב נראה טוב לעיין?  עיצוב יפה מאוד ובחירת צבע ירוק שהוא קשור ל NVIDIA גם מוסיף חוויה יפה. |
| Compatibility | האם האתר עובד אותו דבר לסביבות שונות?  כן, הבעיה היחידה היא גודל ורזולוציה של מסכים שונים שיתכך אי-התאמה. |

שאלות פתוחות:

1. התמודדות עם הקוד היתה מאתגרת, להתבונן כיצד קבוצה אחרת ביצעה את המשימות הייתה מעניינת. זה לא היה פשוט לעבור על הקוד, אך זה בהחלט מלמד.
2. יש מקום לשיפור, היו שאלות שחזרו על עצמן וניתן להתמקד יותר בקטע הקוד.

שם הסטודנט: ספא ח'יר 322829102

|  |  |
| --- | --- |
| **קריטריון** | **הסבר** |
| מימוש | האם הקוד מבצע את הנדרש?  **בהתאם לניתוח, הקוד תואם את הדרישות שהוצגות, הקוד מבצע את הפונקציות המוגדרות לו.** |
| יעילות | האם ניתן לשפר את יעילות הקוד?  **הקוד נראה יעיל יחסית, אך ייתכן שקיימות דרכים נוספות לייעל אותו.** |
| פשטות | האם ניתן לפשט את הקוד?  **הקוד נראה ברור יחסית, אך ייתכן שניתן לפשט אותו עוד יותר, שימוש במשתנים גלובליים גורם לקוד להיראות מורכב יותר.** |
| מודולריות | האם הקוד מודולרי מספיק? (חלוקה תקינה לקבצים, פונקציות ומחלקות)  **הקוד נראה מודולרי יחסית, עם חלוקה נכונה לקבצים, פונקציות ומחלקות, עם זאת, ישנם חלקים בקוד שניתן לחלק בצורה טובה יותר.** |
| תבניות עיצוב | האם ישנו מימוש ברור ונכון של תבניות עיצוב?  **כן, ניתן לזהות שימוש נכון בתבנית עיצוב Singleton עבור חיבור למסד נתונים.** |
| OOP | האם ישנו שימוש נכון בעקרונות OOP? האם ניתן לשפר?  **במעבר על הקוד אין יישום ברור של עקרונות OOP.** |
| באגים וטעויות | האם ישנם מקרים בהם הקוד לא מתנהג כצפוי? אילו?  **הקוד נראה מתפקד בצורה תקינה ולא נמצאו מקרים בהם הוא לא מתנהג כצפוי, כל הפונקציות זמינות ומתפקדות בהתאם לציפיות, ולא זוהו באגים או התנהגויות לא צפויות.** |
| טיפול בשגיאות | האם ישנה התראה על שגיאות משתמש?האם ישנו שימוש במנגנון תפיסת שגיאות?האם הבדיקה ודיבוג הקוד נוחים?האם הודעות המשתמש ברורות וידידותיות למשתמש?  **כן, מנגנון התראה על שגיאות משתמש מוטמע בקוד, בכל מקרה שבו עלולה להתעורר שגיאה, מוצגת הודעה מתאימה למשתמש.** |
| בדיקות | האם יש בדיקות שניתן להוסיף?  **לא, כיסינו את כל המקרים שאפשר לבדוק.** |
| שימושיות | האם הממשק שימושי? (usability)? האם המשחק מעורר רצון להמשיך במנוע?  **ממשק המשתמש מעוצב בצורה נהדרת וקל מאוד לשימוש.** |
| תיעוד | האם הממשק מתועד?  **ממשק המשתמש מתועד בצורה מקיפה וברורה, התיעוד כולל הסברים מפורטים לגבי כל פונקציה ואלמנט בממשק, אך ייתכן שיש צורך בשיפור נוסף.** |
| אתיקה | האם יש שימוש בנתוני משתמש אשר מפרים פרטיותו?האם יש ניצול של התנהגות או חולשה אנושית?האם ישנו algorithmic bias כנגד קבוצה או פרטים מסוימים?  **לא נעשה שימוש בנתוני משתמש באופן שעלול להפר את פרטיותם, הקוד אינו מנצל התנהגות או חולשות אנושיות.** |
| אבטחה | האם ישנו מידע אבטחה (שמות משתמש, ססמאות) גלויים?  **מידע אבטחה, כגון שמות משתמש וסיסמאות, אינו גלוי. רק המשתמש עצמו יכול לראות את סיסמתו, תוך שמירה על רמת אבטחה גבוהה.** |
| ביצועים | האם לדעתך שינוי עתידי בקוד יכול לגרום לפגיעה בביצועים?  **הקוד תוכנן בצורה יעילה וניתן להרחיב אותו ללא פגיעה בביצועים.** |
| ביצועים | האם ניתן לשפר לדעתך את הביצועים?  **הביצועים הנוכחיים של המערכת נראים טובים מאוד, זמני הטעינה של תוצאות החיפוש והסטטיסטיקות קצרים ו** **המעבר בין מסכים חלק ומהיר.** |
| קריאות - readability | האם הקוד מובן בקלות?אילו חלקים בקוד היו לא ברורים עבורך?כיצד ניתן לשפר את קריאות הקוד?  **הקוד נראה ברובו ברור ומובן, המבנה שלו פשוט וקליל לניתוח, השימוש בשמות משתנים ומשתנים משמעותי תורם להבנה.** |
| GUI | האם העיצוב נראה טוב לעיין?  **העיצוב נראה אטרקטיבי ומזמין. השימוש בצבע ירוק והשילוב בין המיתוג של NVIDIA ושם הפרויקט שלהם ויוצר תחושה של אחידות. הפריסה מאוזנת ונעימה לעין.** |
| Compatibility | האם האתר עובד אותו דבר לסביבות שונות?  **כן, רק הבעיה היחידה היא אי התאמה מוחלטת של מסכים בגדלים שונים.** |

שאלות פתוחות:

1. התמודדתי עם אתגרים בעת סקירת הקוד, היה מרתק לראות איך קבוצה אחרת התמודדה עם המשימות. דרכו עבור הקוד לא הייתה פשוטה, אך היא הועילה בלמידה איך להיות בודק קוד טוב יותר, מכיוון שהיא הציגה נקודת מבט חדשה.
2. קיים פוטנציאל לשיפור, חלק מהשאלות נראו כמחזוריות, וכדאי היה להתמקד יותר בקטעי הקוד עצמם, אולי אף לבצע את התרגיל בזוגות מקבוצות שונות.
3. הקורס היה מהנה עבורי, אולי כדאי להעמיק יותר בנושאי חיפוש וגישות לו.

שם סטודנט: ליאן עאמר 323068700

|  |  |
| --- | --- |
| **קריטריון** | **הסבר** |
| מימוש | האם הקוד מבצע את הנדרש?  **הקוד תואם את הדרישות הנדרשות ומצליח להפעיל את הפונקציות שהוגדרו עבורו באופן תקין.** |
| יעילות | האם ניתן לשפר את יעילות הקוד?  **הקוד פועל בצורה יעילה ברוב המקרים, אך ייתכן שיש מקום לייעול נוסף.** |
| פשטות | האם ניתן לפשט את הקוד?  **הקוד מובן למרביתו, אך קיימת אפשרות להפוך אותו לפשוט יותר, ובפרט, שימוש במשתנים גלובליים יכול להקשות על ההבנה.** |
| מודולריות | האם הקוד מודולרי מספיק? (חלוקה תקינה לקבצים, פונקציות ומחלקות)  **הקוד מחולק היטב לקבצים, פונקציות ומחלקות, אך ייתכנו חלקים שניתן למקד ולשפר את חלוקתם.** |
| תבניות עיצוב | האם ישנו מימוש ברור ונכון של תבניות עיצוב?  **ישנו שימוש נכון ויעיל בתבנית עיצוב Singleton לניהול חיבור למסד נתונים.** |
| OOP | האם ישנו שימוש נכון בעקרונות OOP? האם ניתן לשפר?  **הקוד אינו מיישם באופן ברור את כל עקרונות OOP, יש מקום לשיפור בתחום זה.** |
| באגים וטעויות | האם ישנם מקרים בהם הקוד לא מתנהג כצפוי? אילו?  **לא נמצאו באגים או טעויות; הפונקציות פועלות כצפוי ואין התנהגויות לא צפויות.** |
| טיפול בשגיאות | האם ישנה התראה על שגיאות משתמש?האם ישנו שימוש במנגנון תפיסת שגיאות?האם הבדיקה ודיבוג הקוד נוחים?האם הודעות המשתמש ברורות וידידותיות למשתמש?  **הקוד מטפל בשגיאות משתמש ומציג התראות ברורות וידידותיות, מה שמקל על הבדיקה והדיבוג.** |
| בדיקות | האם יש בדיקות שניתן להוסיף?  **התחום מכוסה היטב בבדיקות, ולא נראה שיש צורך בהוספת בדיקות נוספות.** |
| שימושיות | האם הממשק שימושי? (usability)? האם המשחק מעורר רצון להמשיך במנוע?  **הממשק נוח וידידותי למשתמש, מה שמעודד שימוש נוסף והתמקדות במשחק.** |
| תיעוד | האם הממשק מתועד?  **הממשק והקוד מתועדים היטב, אך ייתכן שיש מקום לשיפור והרחבה של התיעוד.** |
| אתיקה | האם יש שימוש בנתוני משתמש אשר מפרים פרטיותו?האם יש ניצול של התנהגות או חולשה אנושית?האם ישנו algorithmic bias כנגד קבוצה או פרטים מסוימים?  **הקוד מכבד את פרטיות המשתמש ולא מנצל חולשות אנושיות, כמו כן אין שימוש בנתונים שעלול להפר את הפרטיות.** |
| אבטחה | האם ישנו מידע אבטחה (שמות משתמש, ססמאות) גלויים?  **אין חשיפה של מידע אבטחה כגון שמות משתמש וסיסמאות, והמערכת מאובטחת היטב.** |
| ביצועים | האם לדעתך שינוי עתידי בקוד יכול לגרום לפגיעה בביצועים?  **הקוד תוכנן בצורה יעילה וניתן להרחיב אותו ללא פגיעה בביצועים.** |
| ביצועים | האם ניתן לשפר לדעתך את הביצועים?  **המערכת פועלת בצורה יעילה, ונראה שניתן להרחיב את הקוד ללא פגיעה בביצועים.** |
| קריאות - readability | האם הקוד מובן בקלות?אילו חלקים בקוד היו לא ברורים עבורך?כיצד ניתן לשפר את קריאות הקוד?  **הקוד ברובו קריא ומובן, עם שימוש בשמות משמעותיים ומבנה ברור.** |
| GUI | האם העיצוב נראה טוב לעיין?  **העיצוב נעים לעין עם שימוש בצבעים הרמוניים ופריסה מאוזנת.** |
| Compatibility | האם האתר עובד אותו דבר לסביבות שונות?  **האתר פועל באופן דומה בסביבות שונות, עם בעיה אפשרית של אי-התאמה בגדלים שונים של מסכים.** |

שאלות פתוחות:

1. במהלך הסקירה, נתקלתי במספר אתגרים ומצאתי את עצמי מרותק לאופן שבו קבוצות אחרות פתרו את המשימה. לא תמיד היה קל לנתח את הקוד, אך התהליך סיפק לי הזדמנות יקרת ערך ללמוד ולהבין כיצד לפקח ולבחון קוד בעין ביקורתית, מה שהעניק לי פרספקטיבה חדשה על התהליך.
2. יש מרחב לשיפורים בתרגיל; מסוימות מהשאלות הופיעו לעיתים קרובות מדי ולא הוסיפו ערך. נראה לי שתהליך העבודה על קטעי קוד ספציפיים יכול להיות מועיל יותר אם יתבצע בשיתוף פעולה בין חברי קבוצות שונות.
3. נהניתי מאוד מההשתלמות; חשוב לי להדגיש שהוספת תוכן נוסף על אסטרטגיות וכלים לחיפוש יכולה להעשיר את החוויה למשתתפים.