מבוא למחשוב ענן - סמסטר חורף התשפ"ה

תרגיל בית 3- עבודה בצוותי העבודה

מועד הגשה: 26.1.25

המשימה בתרגיל זה: סיום בניית המערכת, ביצוע קוד ריוויו לצוות אחר

איטרצית פיתוח מס 2

שימו לב: למטלה זו שלושה חלקים

קישור ל-colab: <https://drive.google.com/file/d/1jr0vu-Wt5ezfiHlkt-T8EDpim2iGSxX7/view?usp=sharing>

קישור ל-github: <https://github.com/ShlomiFridman/PhoenixProject2025>

חלק ראשון.עליכם להגדיר את בעלי התפקידים לאיטרציה זו (יש להחליף מתרגיל בית 2) יש לרשום מי מהנדס.ת המערכת, האחראי.ת בתרגיל זה.

|  |  |  |  |
| --- | --- | --- | --- |
| **שם חבר הצוות** | **משימות שהוקצו** | **משימות שהושלמו** | בדיקת קבלה (acceptance test) – האם הבדיקה עברה? |
| *כפיר אמויאל (product manager)* | בדיקת דרישות | בדיקת דרישות | כן |
| *שלומי פרידמן (backend developer)* | שיפור ה-chatbot | שיפור ה-chatbot | כן |
| *שחר ברנסון (frontend developer)* | תיק מתכנת | תיק מתכנת  תיק משתמש | כן |
| *עומר גולדשטיין*  *(מהנדס מערכת SCUM Master)* | מצגת, סרטון | מצגת, סרטון | כן |
| *ישראל אוחיון (QA)* | בדיקות קבלה | בדיקות קבלה | כן |
| *ולדי טריטנר (UI)* | תיק משתמש | - | - |

* מסמך הבדיקות מצורף בקובץ acceptanceTest.docx

חלק שני: בניית המערכת (80 נקודות)

עליכם לסיים את בניית המערכת.

בפרט, יש לוודא קיום של כלל האלמנטים הבאים:

1. כל הפיצ'רים שהתחלתם לממש בתרגיל בית 2 - יש לוודא שכל המסכים תקינים ועובדים כראוי, בקולאב,ולעבוד עם firebase לצורך שמירת נתונים .אין להשאיר קישורים או כפתורים למסכים שלא קיימים.

יש לוודא שניתן להריץ את הפרויקט מהמחברת בלבד , ללא כל צורך בהטענה של קבצים נוספים או יציאה למערכת חיצונית (כגון FLASK). (30 נקודות)

1. שימוש ב -microservices (תרגול 7)- עליכם לעשות שימוש לפחות בשניים. פרטו באילו שירותים עשיתם שימוש, ומה היתרון בשימוש בשירותים אלו.(10 נקודות)

השירותים שממושו:

1. FirebaseService – השירות שדרכו נעשת כל התקשורת עם ה-Firebase. הייתרון בשירות זה הוא שבגלל שכל התקשורת נעשת בשירות איחד, יותר קל לזהות ופתור באגים שקשורים ל-DB, ואם יש צורך בשליחת\שליפת נתונים כל מה שצריך הוא לקרוא לשירות.
2. IndexService – השירות שמנהל את האינדקס במערכת, את המילים, הקישורים שלהן, וכמות המופעים בכל קישור. הייתרון בשירות זה הוא שכל הלוגיקה של ניהול האינדקס נמצאת במקום אחד, ואם פונקציה מסויימת רוצה לבצע עדכון כלשהו או לגשת לאינדקס העדכני אז ניתן לעשות זאת דרך השירות.
3. CrawlerService – השירות שמבצע Crawling לפי עמוד בודד או על "עץ" של כתובת. הייתרון של שירות זה הוא שהשירות עושה לוגיקה נוספת "מאחורי הקליים" כגון בדיקת ה-robot, לקיחת כתובות נוספות מהעמוד וכו'.
4. QueryService – השירות שאחראי על מתן תשובה לשאילתות, ודירוג עמודים. הייתרון של שירות זה הוא שהוא עובד בנפרד משאר השירותים, הוא עדיין צריך שירות של index אבל הוא מקבל אותו על ידי dependency injection מה שמאפשר פיתוח מודולרי עם תלויות חלשות יותר. הייתרון של שירות זה הוא שלאחר האתחול על מנת לקבל תשובה צריך לשלוח רק את ה-query מבלי לבצע קריאות אחרות.
5. GraphService – השירות שאחראי על בניית הגרפים במערכת (heatmap ו-bar), הוא מקבל dependency injection של שירות אינדקס ולפיו בונה את הגרפים ומציג אותם ב-output שהוגדר לו באתחול. הייתרון של שירות זה הוא שאם רוצים לעדכן את הגרפים כל מה שצריך הוא לקרוא לו שוב, והוא יבצע את העדכון ואוטומטית יציג אותם במסכים.
6. צטבוט שישולב במערכת (כפי שלמדתם בתרגול 8), הכולל מידע אשר מגיע מה - DB שלכם, ועונה בצורה אינטליגנטית לשאלות המשתמש. (15 נקודות).

* שילבנו במערכת צטבוט שעובד עם Genai, כל הודעה שמגיע אליו הוא עוברת עיבוד כך שהוא יענה להודעה בתור צטבוט של אתר IBM.

1. מהם ה -KPI הרלוונטיים בפרויקט שלכם? הסבירו? (10 נקודות)

**מדדי KPI לפרויקט**:

- **User Satisfaction**

o שהמשתמש יהיה מרוצה מתוצאת החיפוש, כלומר שהתוצאות חיפוש יכילו תוצאותנכונות.

o מדידה: שתהיה לפחות תוצאה אחד נכונה \ רלוונטית ב-5 התוצאות הראשונות.

- **Performance**

o שהמשתמש לא ימתין יותר מדי זמן לתוצאת החיפוש.

o מדידה: אחרי לחיצה על כפתור החיפוש לא יעברו יותר מ-3 שניות עד שתוצג תוצאה כלשהי.

- **Scalability & Elasticity**

o שהמערכת תהיה מודולרית, וקלה להרחבה.

o מדידה: פיתוח מודולרי, שכל פונקציה ומחלקה תהיה עצמאית, כלומר בלי תלות במשתנים גלובלים.

1. שקיפות אלגוריתמית – כיצד הבהרתם למשתמשים את האלגוריתמים והנתונים הנאספים בקוד שלכם? (5 נקודות)

* אנו מציגים למשתמש את אופן דירוג העמודים בתוצאות חיפוש, את הדירוג של כל עמוד, כמה עמודים נמצאו, וכמה זמן לקח לבצע את החיפוש. בנוסף במערכת יש את הגרף של bar שמראה את כמות המופעים של מילות מפתח באתר של IBM, וגרף heatmap שמראה כמה url-ים משותפים יש בין כל שני מילות מפתח, על מנת לתת למשתמש כמה שיותר מידע כדי שיוכל לבנות query יותר יעיל.

1. אתגרים שעלו במהלך העבודה , וכיצד התמודדתם איתם (לדוגמא- DB גדול מדי, יצירת ויזואליזציה, מבנה הקוד וכו) (10 נקודות)

* בניית האינדקס: בהתחלה רצינו שהאינדקס יבנה באופן דינאמי במהלך תהליך ה-crawling, אבל ראינו שהוא בנה אותו עם יותר מדי מילים (למרות שהוצאנו את ה-stopwords) שלא היה ניתן לשמור אותו ב-firebase. אז הוחלט שנגדיר את האינדקס הראשוני עם מונחים מהקורס ומהאתר של IBM ונבנה את האינדקס רק עבור אותן מילים. ובהמשך במידת הצורך ניתן להוסיף מילים חדשות על ידי תפריט האינדקס.
* תהליך ה-Crawling: בהתחלה עשינו crawling על כ-200 עמודים, אבל ראינו שאנו שלא מגיעים לעמודים שציפינו, כגון עמודים שמכילים מידע על מונחים. אז שיפרנו את ה-CrawlerService על מנת שיתעלם מעמודים שאינם רלוונתים לנו, כגון עמודי הורדות, עמודי mp3 וכו'.
* קוד מודולרי: לא היה לנו ניסיון רב בפייטון, ולכן בהתחלה רשמנו את כל הפעולות העיקריות בפונצקיות משלהן, מה שגרם לעבודה מבולגנת כאשר פונקציה אחת הייתה תלוייה בשנייה.

לאחר מכן הוחלט שנלמד עצמאית את החלק של Class-ים בפייטון ובנינו את השירותים עם כמות תלויות מינימלית (ואיפה שיש אנו משתמשים ב-dependency injection). כתוצאה מכך יצא לנו מבנה יותר מסודר של המערכת, עם שימוש בשירותים, ובלי קוד "גלובלי" בסקריפט אלה רק קריאה לפונצקיית ה-main.

* המשך בעמוד הבא.

חלק שלישי : סגירת הפרויקט \_( 20 נקודות)

בחלק זה תכינו מסמכים המתארים את הפרויקט שלכם:

1. יש לבנות תיק למתכנת הכולל את שמות כל הקבצים המרכזיים, פונקציות מרכזיות, קטעי קוד/תבניות עיצוב מעניינים שהשתמשתם בהם.

* מצורף במסמך וורד בשם DeveloperKit.docx
* הסבר נוסף נמצא גם עמוד 2 של קובץ זה.

1. יש לבנות תיק למשתמש , הכולל הסבר כללי על המערכת , פירוט מסכים, מעברים בין מסכים והסבר על טעויות אפשרויות.

* מצורף במסמך וורד בשם UserKit.docx

1. התייחסו בתיק המשתמש לשקיפות אלגוריתמית – כיצד הבהרתם למשתמשים את האלגוריתמים והנתונים הנאספים בקוד שלכם?

* מצורף במסמך וורד בשם UserKit.docx

1. יש להכין סרטון קצר של 30-60 שניות, המתאר את השימוש במערכת. הסרטון משמש כ – elevator pitch למערכת שלכם, כלומר יש לכלול בו הסבר מקצועי ועם זאת שיווקי , המדגיש את האלמנטים המיוחדים של המערכת שבניתם. יש להגיש את הסרטון בפורמט mp4.

הוראות הגשה:

1. יש להגיש את התרגיל בצוותים, בתיקיית ה –GIT שלכם, וכן בתיקייית התרגיל ב moodle. חובה לכלול קישור לתיקיית הגיט בקובץ במוודל.

2. יש להגיש במוודל קובץ זיפ הכולל קובץ וורד ובו מענה לשאלות, וקישור ל- notebook ובו הקוד שלכם (יש לוודא שהקישור פומבי ונגיש).

3. כותרתו של הקובץ תהיה HW3\_TEAMNAME

4. שימו לב כי כל העבודות חייבות להיות שונות זו מזו.אנו מריצות תוכנה לבדיקת עבודות זהות. עבודות שייראו דומות ייפסלו ויינתן עליהן ציון 0.

בהצלחה!