**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

**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? |
| עבד + וואדיע | מעבר על ה microservices , KPI , CHATBOT , הגשה סופית | הכל | 100% |
| יוסף + אדם + ראני | מעבד על מחברת קוד + בניית תיק מתכנת | הכל | 100% |
| גורג + אמיר | בדיקת שימושיות ממשק + בניית תיק משתמש | הכל | 100% |
|  |  |  |  |
|  |  |  |  |

***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)

בוצע , עברנו על כל דפים הממשק , מבנה האינדקס בפיירבייס , לא קיימים קישורים חצוניים שמחברת ה כולאב מסתמכת עליהם .

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

שירות אינדקס :

**מיקרוסרביס זחלן (Crawler Microservice)**

* אחראי: איסוף וחילוץ תוכן מדפי אינטרנט
* יתרון: מאפשר סריקה מקבילית ועדכון דינמי של תוכן

**מיקרוסרביס עיבוד טקסט (Text Processing Microservice)**

* אחראי: ניתוח טקסט ויצירת אינדקס
* יתרון: מאפשר עיבוד מהיר ויעיל של טקסט בנפרד מהזחילה

שירותי שאילתה :

**מיקרוסרביס אחזור (Retrieval Microservice)**

* אחראי: שליפת נתונים מהאינדקס
* יתרון: מאפשר חיפוש מהיר ומבוזר

**מיקרוסרביס דירוג (Ranking Microservice)**

* אחראי: דירוג ומיון תוצאות החיפוש
* יתרון: מאפשר התאמה אישית של אלגוריתם הדירוג

שירותי מנהל :

**מיקרוסרביס אימות (Authentication Microservice)**

* אחראי: אימות משתמשים מנהלים
* יתרון: אבטחה משופרת וניהול הרשאות מרכזי

**מיקרוסרביס ניהול אינדקס (Index Management Microservice)**

* אחראי: הוספה והסרה של מונחים מהאינדקס
* יתרון: גמישות בעדכון ותחזוקת האינדק

1. צטבוט שישולב במערכת (כפי שלמדתם בתרגול 8), הכולל מידע אשר מגיע מה - DB שלכם, ועונה בצורה אינטליגנטית לשאלות המשתמש. (15 נקודות).

הסבר הפיצ'ר: הוספנו למנוע החיפוש Chatbot, של Chatbase.co ,שמאפשר למשתמשים לשאול שאלות ולקבל תשובות בזמן אמת. הפיצ'ר נועד לספק מענה מיידי לשאלות בנוגע למטרה של הממשק כלומר (שאלות רלוונטיות ל Google Cloud Engine ) , ולשפר את חוויית המשתמש על ידי מתן פתרונות לשאלות נפוצות.

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

* **אפקטיביות (דיוק ורלוונטיות) Effectiveness**

מנוע החיפוש מדרג התוצאות (כתובות URL) לפי תדירות המילים ב Query, ומוודא שהתוצאות הרלוונטיות מדורגות לפי נוסחת RANKING , שהיא סכום תדירויות המילים בשאילתה לקישור הנתון . אין אחוז דיוק ספציפי, אך הדירוג מבוסס על המופע הגבוה ביותר של המילה.

* **יעילות  Efficiency**

מנוע החיפוש מעבד את השאילתות ומציג תוצאות בתוך 0.75–1 שניה, התצוגה של התוצאות היא באופן מיידי כי אין לנו מספר עצום של תוצאות .

* **איכות (עקביות ואמינות) Quality**

הפלטפורמה פועלת ב-Google Colab ומטפלת בשגיאות עם הודעות ברורות (למשל, חיפוש ללא מילת מפתח או התחברות לא נכונה של מנהל), ומספקת פעולה חלקה ללא צורך בתחזוקת שרת.

* **חדשנות (תכונות ושדרוגים חדשים) Innovation**

התכונות המרכזיות הושלמו במהלך הסמסטר, ושדרוגים עתידיים כוללים תכונות כמו הצעות נכונות, גרפים סטטיסטיים מפורטים יותר, והגבלת תחום הצ'אט-בוט לנושא הרלוונטי לשיפור החוויה המקצועית.

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

**גרף כמו זה שהצגת קל להבנה בגלל כמה סיבות מרכזיות:**

1. **ויזואליזציה ברורה ופשוטה:**
   * עמודות בצבע אחיד ומסודר מאפשרות להשוות בקלות את הערכים אחד לשני.
   * אין עומס של מידע נוסף שעלול להקשות על הבנת המידע המרכזי.
2. **תוויות ברורות:**
   * צירי ה-X וה-Y מתויגים בצורה ברורה עם שמות כמו "Terms" ו-"Frequency", כך שהמשתמש מבין מיד מה הגרף מציג.
3. **כותרת שמסבירה את המידע:**
   * כותרת הגרף "Most Frequent Terms in the Index" מספקת הקשר ברור למידע שמוצג.
4. **סידור היררכי:**
   * המידע מסודר לפי גובה עמודות, המאפשר לראות באופן מיידי איזה מונח הוא השכיח ביותר.
5. **עיצוב נקי:**
   * עיצוב מינימליסטי עם רקע בהיר ופונט ברור עוזר למשתמש להתמקד בנתונים עצמם בלי הסחות דעת

**צ'אט-בוט** : זה מספק מענה ממוקד לנושאים ספציפיים בצורה מקצועית, תוך שמירה על פרטיות המשתמש והימנעות מגישה לנתונים אישיים או היסטוריית חיפוש. הוא מתאפיין בתשובות ברורות ומנומסות שמסבירות את יכולותיו בצורה ידידותית, ויוצר אינטראקציה חכמה על ידי שאילת שאלות המותאמות לצרכי המשתמש. בנוסף, עיצובו הפשוט והנגיש מאפשר לכל משתמש להשתמש בו בקלות וביעילות.

**הצבעים** : בממשק יוצרים חוויית משתמש נעימה וברורה בזכות שילוב רקע בהיר שמדגיש ניקיון ונותן פוקוס לאלמנטים החשובים, שימוש בכחול לכפתורים וקישורים אינטראקטיביים שמקרינים אמון, ירוק שמוסיף ידידותיות ותחושה טבעית כחלק מהמותג, ושחור לאייקון הצ'אט שמושך תשומת לב לשירות. השילוב הזה שומר על איזון בין מקצועיות לפשטות.

1. אתגרים שעלו במהלך העבודה , וכיצד התמודדתם איתם (לדוגמא- DB גדול מדי, יצירת ויזואליזציה, מבנה הקוד וכו) (10 נקודות)
2. **יצירת ויזואליזציה**
   * **אתגר**: הוספת ויזואליזציות למערכת, כמו גרפים וסטטיסטיקות, דרשה כלים שלא היו חלק מהממשק הבסיסי של Google Colab.
   * **פתרון**: השתמשנו ב-Chart.js ובספריות JavaScript ליצירת גרפים דינמיים, ושילבנו אותם ב-GUI באמצעות HTML ו-JavaScript.
3. **פערי ידע בין חברי הצוות**
   * **אתגר**: לכל אחד מחברי הצוות היו רמות ידע שונות, דבר שיצר קושי בחלוקת משימות והבנה משותפת של הבעיות.
   * **פתרון**: קיימנו פגישות תיאום, ייעדנו זמן ללמידה משותפת, ופרסנו את המשימות לפי תחומי הידע והחוזקות של כל אחד.
4. **תיאום זמן עבודה**
   * **אתגר**: הצוות כלל סטודנטים עם לוחות זמנים עמוסים, מה שהקשה על קביעת זמן עבודה משותף.
   * **פתרון**: השתמשנו בכלי ניהול כמו Google Calendar ותכננו פגישות קצרות בזמנים קבועים. עבדנו בספרינטים קטנים כדי לוודא שכל אחד מתקדם לפי יכולתו.
5. **חוסר Debugger ב-Google Colab**
   * **אתגר**: Google Colab אינו מספק debugger כמו בסביבות פיתוח אחרות, מה שהקשה על פתרון בעיות בקוד מורכב.
   * **פתרון**: השתמשנו בפקודות הדפסה (print) לניטור משתנים ושלבי קוד. בנוסף, הרצנו חלקים מהקוד בסביבות מקומיות עם תמיכה ב-debugging.

***חלק שלישי : סגירת הפרויקט \_( 20 נקודות)***

בחלק זה תכינו מסמכים המתארים את הפרויקט שלכם:

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

**תיק למתכנת Turtle Search**

**תיאור כללי של הפרויקט**

Turtle Search הוא מנוע חיפוש מבוסס Firebase המאפשר חיפוש, ניתוח סטטיסטיקות וניהול מונחים באמצעות ממשק אינטראקטיבי. הפרויקט משלב שימוש בממשקי API של Firebase, ממשקי משתמש מעוצבים ב-HTML ו-JavaScript, וכלים להצגת נתונים סטטיסטיים בצורה גרפית.

**מטרות הפרויקט:**

* ניהול נתונים במערכת מבוססת Firebase.
* מתן אפשרות חיפוש יעילה.
* הצגת סטטיסטיקות על נתונים במערכת.
* ממשק ידידותי למשתמש לניהול אדמיניסטרטיבי.

**טכנולוגיות:**

* **שפות:** Python, HTML, CSS, JavaScript.
* **בסיס נתונים:** Firebase Real-Time Database.
* **ספריות נוספות:** Chart.js להצגת נתונים, IPython להדמיות.

**שמות כל הקבצים המרכזיים**

| **תיאור** |
| --- |
| קובץ הרצה ראשי, כולל הגדרות Firebase וניהול חיפושים. |
| עמוד הבית של המערכת, כולל ממשק החיפוש והניהול. |
| עיצוב אלמנטים גרפיים בעמודי HTML. |
| הגדרות התחברות ל-Firebase. |
| לוגיקה להצגת נתונים סטטיסטיים וגרפים. |

**פונקציות מרכזיות**

1. **פונקציה authenticateUser:**

**תיאור:** מבצעת אימות משתמש באמצעות Firebase.

function authenticateUser(email, password) {

firebase.auth().signInWithEmailAndPassword(email, password)

.then((userCredential) => {

console.log('User signed in:', userCredential.user);

})

.catch((error) => {

console.error('Authentication failed:', error.message);

});

}

1. **פונקציה handle\_statistics:**

**תיאור:** אוספת נתונים סטטיסטיים מבסיס הנתונים ומציגה אותם בגרפים אינטראקטיביים.

from IPython.display import display, Javascript

def handle\_statistics():

firebase\_url = "https://your-firebase-database.firebaseio.com/"

response = requests.get(f"{firebase\_url}/index.json")

index\_data = response.json()

# עיבוד הנתונים

# הצגת סטטיסטיקות באמצעות Chart.js

1. **פונקציה addTerm:**

**תיאור:** מוסיפה מונח חדש לבסיס הנתונים דרך ממשק האדמין.

function addTerm(term, url, count) {

firebase.database().ref('terms').push({

term: term,

url: url,

count: count

});

alert('Term added successfully!');

}

**קטעי עיצוב מעניינים**

* 1. **עיצוב כפתורים ב-CSS**

button {

background-color: #63b3ed;

color: white;

padding: 10px 20px;

border: none;

border-radius: 20px;

cursor: pointer;

transition: background-color 0.3s ease;

}

button:hover {

background-color: #4299e1;

}

* 1. **עיצוב תיבת חיפוש**

input[type="text"] {

width: 80%;

padding: 12px;

border: 2px solid #e2e8f0;

border-radius: 25px;

font-size: 16px;

outline: none;

}

**שיפורים עתידיים:**

* הוספת תמיכה בחיפושים מרובים בבת אחת.
* שיפור אלגוריתם החיפוש להתאמה אישית של תוצאות.
* שילוב למידת מכונה לניתוח סטטיסטי מתקדם.
* הוספת תמיכה במספר שפות בממשק המשתמש.

**סיכום:**

Turtle Searchהוא פרויקט המשלב טכנולוגיות עדכניות במטרה לספק פתרון חיפוש יעיל וקל לשימוש. עם ממשק ידידותי, ניתוח נתונים אינטראקטיבי ופונקציונליות מתקדמת, הפרויקט מהווה בסיס מעולה להמשך פיתוח.

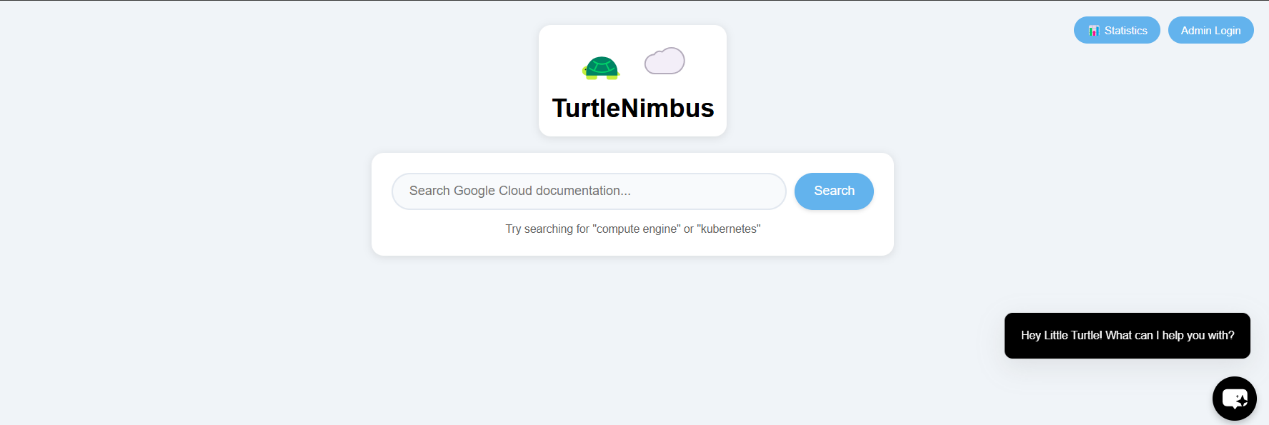
**תיק משתמש: מערכת TurtleNimbus**

**הסבר כללי על המערכת:**

מערכת TurtleNimbus נועדה לאפשר למשתמשים לחפש מידע מתוך מסמכים ותיעוד של Google Cloud בצורה נוחה ויעילה. המערכת כוללת ממשק חיפוש ידידותי, מערכת סטטיסטיקות להצגת נתונים על התכנים במאגר, ולוח ניהול למנהלי המערכת. בנוסף, המערכת כוללת עוזר אישי וירטואלי המסייע למשתמשים בשאלות ותקלות.

**פירוט המסכים:**

**מסך הבית**

* **מטרת המסך**: לאפשר חיפוש ראשוני של תוכן מתוך מאגר התיעוד.
* **מרכיבים מרכזיים**:
  + שדה חיפוש להזנת מונחי החיפוש.
  + כפתור "Search" לביצוע החיפוש.
  + עיצוב לוגו "TurtleNimbus".
  + כפתורים לכניסה למסך הסטטיסטיקות ולוח הניהול (Admin Login).

**מסך תוצאות חיפוש**

* **מטרת המסך**: להציג את התוצאות הרלוונטיות למונח החיפוש.
* **מרכיבים מרכזיים**:
  + רשימת תוצאות עם כותרת, קישור וציון רלוונטיות לכל פריט.
  + אפשרות לדפדף בין עמודי תוצאות באמצעות כפתורי "Next" ו-"Previous".
  + עוזר וירטואלי המופיע בפינה התחתונה של המסך.

A screenshot of a chat

Description automatically generated

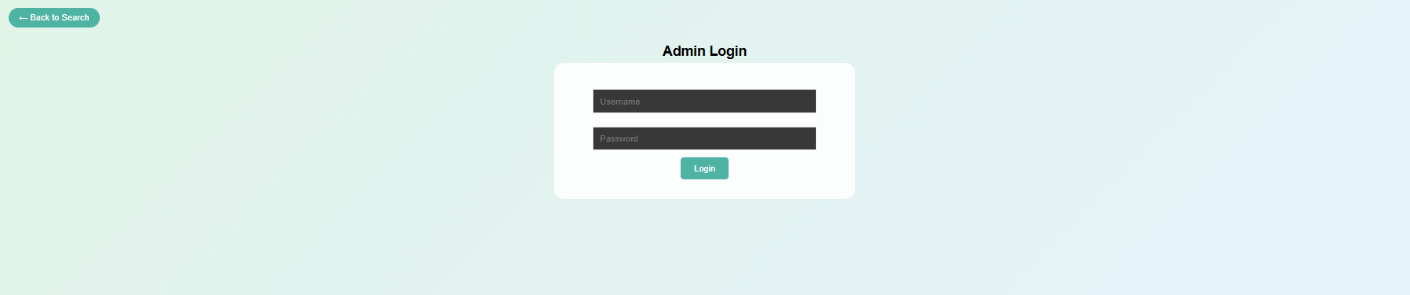
**מסך סטטיסטיקות**

* **מטרת המסך**: להציג מידע סטטיסטי על התכנים במערכת.
* **מרכיבים מרכזיים**:
  + מספר המילים הכולל במאגר.
  + מספר המונחים הייחודיים.
  + מספר ה-URLs במאגר.
  + גרף המציג את המונחים הנפוצים ביותר במאגר.
  + A screenshot of a graph

    Description automatically generatedכפתור חזרה לחיפוש.

**מסך כניסת מנהל (Admin Login)**

* **מטרת המסך**: לאמת את זהות מנהל המערכת.
* **מרכיבים מרכזיים**:
  + שדה להזנת שם משתמש.
  + שדה להזנת סיסמה.
* כפתור Login לביצוע כניסה.



**מסך לוח ניהול (Admin Dashboard)**

* **מטרת המסך**: לאפשר למנהלי המערכת להוסיף, להסיר ולנהל מונחים ו-URLs במערכת.
* **מרכיבים מרכזיים**:
  + טופס להוספת מונחים חדשים (כולל URL).
  + טופס להסרת מונחים קיימים.
  + טופס להסרת URL ממונח קיים.
  + כפתור Logout ליציאה מהממשק.

A screenshot of a computer

Description automatically generated

**מעברים בין מסכים:**

* **מעבר ממסך הבית למסך תוצאות החיפוש**: מתבצע על ידי לחיצה על כפתור "Search" לאחר הזנת מונח.
* **מעבר ממסך הבית ללוח הניהול**: מתבצע על ידי לחיצה על כפתור "Admin Login" ולאחר מכן אימות זהות במסך הכניסה.
* **מעבר ממסך הבית למסך הסטטיסטיקות**: מתבצע על ידי לחיצה על כפתור "Statistics".
* **מעבר מלוח הניהול למסך הבית**: מתבצע על ידי לחיצה על כפתור Logout.
* **מעבר בין עמודי תוצאות החיפוש**: מתבצע על ידי שימוש בכפתורי "Next" ו-"Previous".

**טעויות אפשריות והתמודדות:**

* **פרטי מנהל שגויים**: אם שם המשתמש או הסיסמה אינם נכונים במסך הכניסה, תוצג הודעה: "שם משתמש או סיסמה שגויים".
* **מונח לא נמצא**: אם אין תוצאות לחיפוש, תוצג הודעה: "לא נמצאו תוצאות עבור המונח שהוזן".
* **שדה ריק בטפסי לוח הניהול**: אם המשתמש מנסה להוסיף או להסיר מונח/URL ללא מילוי כל השדות, תוצג הודעה: "אנא מלא את כל השדות הנדרשים".

**סיכום**

מערכת TurtleNimbus נבנתה כדי לשפר את חוויית החיפוש והניהול של מסמכים. התיק נועד לסייע למשתמשים להכיר את תכונות המערכת ולהתמודד עם טעויות נפוצות בצורה יעילה.

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.

קישור מחברת עדכני + הרשאה לערוך :

<https://colab.research.google.com/drive/1ddXuZuVv7JVxbGMmJ908LNi5mBH8j3mG?usp=sharing>

קישור GitHub : <https://github.com/wadeaT/Cloud-Computing-Course>

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!