**מבוא למחשוב ענן-פרויקט סופי**

**TIGER**

**קישור למחברת גוגל קולאב**  
<https://colab.research.google.com/drive/1W_vH-lpkyEVZd5fDhWDfUmETdkdYGgpY?usp=sharing>

**Git קישור לתיקיית**   
[**https://github.com/aymanhaj/HW1\_Tiger-**](https://github.com/aymanhaj/HW1_Tiger-)

**מהות המוצר ומרכיביו**

**מודולים ופיצ'רים מעניינים**

MQTTהמערכת שפיתחנו היא מנוע חיפוש מתקדם המתמחה באתר

, המשלב מספר מודולים מרכזיים

**מודול ניהול אינדקס**

סריקה אוטומטית של דפי האתר עם בקרת מגבלות (5-100 דפים)

עיבוד טקסט מתקדם כולל Stemming ו-Stop Words filtering

FIREBASEעם יכולת רענון ועדכון ב

**מודול חיפוש ודירוג**

אלגוריתם דירוג מבוסס תדירות מילים

הצגת תוצאות עם תגי ציון צבעוניים

ממשק חיפוש אינטואיטיבי עם זמן תגובה מהיר

**מודול סטטיסטיקות ואנליטיקה**

גרפי Big Data של המילים הנפוצות ביותר

ניתוח ביצועים בזמן אמת

KPI מדדי ויזואליים

**צ'אטבוט: AI אינטראקטיבי**

לתמיכה בזמן אמת Gradio מבוסס

מסביר תפקוד המערכת ומדריך משתמשים

מטפל בתקלות ומספק פתרונות

**מודול משחוק: (Gamification)**

MQTT חידון אינטראקטיבי על פרוטוקול

מערכת ניקוד עם 5 נקודות לשאלה נכונה

טבלת מובילים לעידוד תחרותיות

**Microservices שנעשה בהם שימוש**

Microservices המערכת מיישמת ארכיטקטורת

עם שני שירותים עצמאיים

| **שירות** | **תיאור** | **יתרון** |
| --- | --- | --- |
| **שירות דירוג** | חישוב ציון לכל דף לפי שכיחות מילים מהשאילתה | מאפשר שיפור תוצאות מיידי, ללא צורך בבניית אינדקס חדש |
| **שירות סטטיסטיקה** | חישוב והצגת גרפים על פופולריות מונחים, זמן תגובה, עומס | מבדל בין תצוגה לוגית למידע גולמי - נוח להרחבות עתידיות |

**יתרונות הארכיטקטורה**

סקלביליות: כל שירות יכול להתרחב באופן עצמאי

תחזוקה: שינויים בשירות אחד לא משפיעים על האחרים

פיתוח מקבילי: צוותים שונים יכולים לעבוד על שירותים שונים

חוסן: כשל בשירות אחד לא משפיע על היתר

**KPI מרכזיים**

**זמן תגובה ממוצע**

יעד:פחות מ 1.5 שניות

**מדידה נכוחית: 1.4 שניות**

**חשיבות:**

משפיע ישירות על חוויית המשתמש ושמישות המערכת

**SUS - System Usability Scale**

יעד:מעל 80 נקודות

**מדידה נוכחית:82/100**

**חשיבות:**

מעיד על איכות הממשק וקלות השימוש

**ניתוח נתוני עתק**

**גרף ניתוח Big Data – המילים הנפוצות ביותר באתר**

המערכת מציגה גרף עמודות (Bar Chart) המבוסס על ניתוח של אלפי דפים:

**10 המילים הנפוצות ביותר** כולל תדירות הופעה

**עדכון דינמי** לפי סריקות חדשות

**פילטור חכם** שמסיר(סטופ וורדז) ומתמקד במונחים טכניים

הגרף מאפשר למשתמשים להבין את התוכן הדומיננטי באתר ולזהות מגמות בפרוטוקול MQTT

**ארכיטקטורת המערכת**

**תרשים ומאפיינים מרכזיים**

**סוג הארכיטקטורה:**

Layered Pattern

תמונה שמכילה תרשים, צילום מסך, טקסט, תוכנית

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

**מאפיינים מרכזיים**

**הפרדת אחריות: כל שכבה אחראית על תחום ספציפי**

**קלות תחזוקה:** שינויים בשכבה אחת לא משפיעים על אחרות

**מודולריות:** ניתן להחליף רכיבים ללא השפעה על המערכת כולה**:**

**Use Case מעודכן של פונקציונליות המערכת**

**פעולות מרכזיות**

**מנהל מערכת**

בניית אינדקס חדש

קביעת מספר דפים לסריקה

שמירה וטעינה מ-Firebase

צפייה בסטטיסטיקות מערכת

**משתמש רגיל**

ביצוע חיפוש מתקדם

צפייה בתוצאות מדורגות

שימוש בצ'אטבוט לעזרה

השתתפות בחידון MQTT

**צ'אטבוט AI:**

מענה לשאלות כלליות

הדרכה על השימוש

פתרון תקלות שכיחות

**דרישות פונקציונליות**

**חיפוש מתקדם:** המערכת תאפשר חיפוש מילות מפתח עם דירוג תוצאות

**ניהול אינדקס:** בניה, שמירה וטעינה של אינדקס מילים

**סטטיסטיקות:** הצגת גרפים ונתונים על השימוש

**צ'אטבוט:** תמיכה אינטראקטיבית למשתמשים

**משחוק:** חידון עם מערכת ניקוד וטבלת מובילים

**דרישות לא פונקציונליות**

**Performance (ביצועים)**

זמן תגובה מקסימלי של 1.5 שניות לחיפוש

תמיכה בעד 100 משתמשים בו-זמנית

טעינת דפים תוך פחות מ-3 שניות

**Usability (שמישות)**

ציון SUS מעל 80

ממשק אינטואיטיבי ללא צורך בהדרכה

תמיכה בדפדפנים מרכזיים (Chrome, Firefox)

**Reliability (אמינות)**

זמינות מערכת של 99.5%

התאוששות אוטומטית מתקלות Firebase

גיבוי נתונים רציף

**Scalability (סקלביליות)**

יכולת הרחבה למאות אלפי דפים

תמיכה בהוספת microservices נוספים

אופטימיזציה אוטומטית של מסד הנתונים

**Security (אבטחה)**

הצפנת תקשורת עם Firebase

ולידציה של קלטי משתמש

מניעת SQL injection

**ביקורת עמיתים ו-SUS**

**ציון SUS: 82.0 מתוך 100**

הציון מעיד על **שמישות גבוהה מאוד** של המערכת וחוויית משתמש מצוינת.

**טבלת משובים מהסטודיו:**

| **הערת משוב** | **האם התבצע שינוי?** | **נימוק** |
| --- | --- | --- |
| צריך לבדוק האם יש צורך שהמשתמש יבחר את מגבלת כמות הדפים | לא התבצע | דף Admin לא אמור להיות פתוח לכלל המשתמשים אלא רק למנהל המערכת |
| בכפתורים יש מילים חתוכות | כן | הגדלת את הכפתורים כך שיכילו את כל המילים |
| יוסיפו בהמשך משחקים | לא התבצע | מאחר שלא דרוש במפורש, החלטנו לדחות |
| עיצוב | לא התבצע | רוב הקבוצות התרשמו מהעיצוב והאנימציה |

**שקיפות אלגוריתמית**

**הבהרה למשתמשים של האלגוריתמים והנתונים**

**מסך "איך זה עובד?"**

הסבר מפורט על אלגוריתם הדירוג

תיאור תהליך ה-Stemming וסינון Stop Words

דוגמאות קונקרטיות לחישוב ציונים

**תגי ציון בתוצאות**

כל תוצאה מוצגת עם **תג ציון צבעוני**

מספר המייצג את רמת הרלוונטיות

Tooltips עם הסבר מקור הציון

**מדריכים אינטראקטיביים**

בכל מסך יש מדריך מפורט

הסבר על הפעולות הזמינות

הכוונה ברורה למשתמשים חדשים

**שקיפות באיסוף נתונים**

הסבר על איזה מידע נשמר

מדיניות פרטיות ברורה

אפשרות לראות את הנתונים האישיים

**אתגרים שעלו במהלך העבודה**

| **אתגר** | **פתרון** |
| --- | --- |
| משקל מסד נתונים גבוה בזיכרון | הטענת נתונים חלקית לפי בקשה, לא הכל מראש |
| עיצוב טבלאות תוצאה בממשק | שימוש ב-display(HTML(...)) עם עיצוב מותאם |
| שילוב צ'אטבוט עם Firebase | הפעלת הצ'אטבוט כמודול עצמאי בתוך iframe |
| שמירה על זמינות בעת עומס | שימוש בערכי ברירת מחדל כש-Firebase לא מגיב |

**אתגרים נוספים:**

**אופטימיזציה של אלגוריתם הדירוג** - פיתחנו מערכת משוקללת

**תאימות דפדפנים** - בדיקות מקיפות ב-Chrome ו-Firefox

**זמן טעינה** - הטמענו lazy loading ו-caching חכם

**תיק תחזוקה**

**קבצים מרכזיים:**

**search\_engine.py** - מימוש מנוע החיפוש והאינדקס

build\_index() - יצירת אינדקס על בסיס דפים סרוקים

search\_query() - קבלת שאילתה והחזרת תוצאות מדורגות

calculate\_score() - דירוג לפי תדירות הופעה

**ui\_widgets.py** - בניית רכיבי הממשק באמצעות ipywidgets

create\_fancy\_button() - יצירת כפתורים מעוצבים

show\_premium\_status() - הצגת הודעות סטטוס צבעוניות

create\_fancy\_progress\_bar() - בר התקדמות אנימטיבי

**firebase\_utils.py** - פונקציות לשליפה ועדכון מידע ב-Firebase

save\_to\_firebase() - שמירת אינדקס למסד נתונים

load\_from\_firebase() - טעינת נתונים קיימים

update\_statistics() - עדכון מדדי ביצועים

**quiz\_module.py** - מנגנון המשחק (משחוק)

generate\_quiz\_question() - יצירת שאלות דינמיות

calculate\_score() - חישוב ניקוד

update\_leaderboard() - עדכון טבלת מובילים

**stats\_module.py** - יצירת גרפים וסטטיסטיקות מהמידע הגולמי

plot\_term\_frequency() - הצגת גרף מילים נפוצות

generate\_performance\_metrics() - מדדי ביצועים

create\_usage\_analytics() - אנליטיקת שימוש

**תבניות עיצוב מעניינות:**

**Glassmorphism Design:**

.glass-card {

background: rgba(255, 255, 255, 0.1);

backdrop-filter: blur(20px);

border-radius: 15px;

border: 1px solid rgba(255, 255, 255, 0.2);

}

**אנימציות מעבר:**

@keyframes slideIn {

from { transform: translateY(-20px); opacity: 0; }

to { transform: translateY(0); opacity: 1; }

}

**8. תיק למשתמש**

**הסבר כללי על המערכת**

**המערכת היא מנוע חיפוש מתקדם המתמחה באתר mqtt.org עם ממשק משתמש מתקדם המשלב:**

**חיפוש חכם עם דירוג תוצאות מתקדם**

**ניהול אינדקס מבוסס Firebase**

**אנליטיקה ויזואלית של נתונים**

**צ'אטבוט AI לתמיכה בזמן אמת**

**חידון אינטראקטיבי עם מערכת ניקוד**

**קריאת חיישנים בזמן אמת**

**פירוט מסכים לפי הממשק:**

**מסך ראשי (Dashboard) -**

**התכונות הפרימיום כוללות:**

**Firebase Database**

**Pre-Collected Sensor Data**

**Enhanced Displays**

**Clean ASCII Index**

**Fixed Encoding Issues**

**Export Capabilities**

**Interactive Visualizations**

**Separate Analytics**

**כפתורי ניווט מרכזיים:**

**🔍 SMART SEARCH - חיפוש חכם עם דירוג מתקדם**

**📊 SEARCH ANALYTICS - אנליטיקה של חיפושים**

**🤖 PROJECT CHATBOT - צ'אטבוט עזרה AI**

**🎯 PLAY QUIZ - חידון MQTT אינטראקטיבי**

**⚙️ ADMIN PANEL - פאנל ניהול מערכת**

**🌡️ SENSOR READINGS - קריאות חיישנים בזמן אמת**

**📈 SENSOR ANALYTICS - אנליטיקה של חיישנים**

**💡 HELP GUIDE - מדריך עזרה מפורט**

**מסך Admin Dashboard:**

**כותרת: "Firebase-Powered Search Engine Management"**

**בקרות ניהול:**

**Pages to crawl: Slider לבחירת מספר דפים (ברירת מחדל: 20)**

**🌐 START CRAWLING - התחלת סריקת האתר**

**🔨 BUILD INDEX - בניית אינדקס מהדפים הסרוקים**

**ניהול נתונים:**

**💾 SAVE TO FIREBASE - שמירת האינדקס למסד נתונים**

**📥 LOAD FROM FIREBASE - טעינת אינדקס קיים**

**🧹 CLEAN INDEX - ניקוי ואיפוס האינדקס**

**ניווט:**

**❓ HELP - עזרה למסך הניהול**

**🏠 MAIN MENU - חזרה לתפריט הראשי**

**מעברים בין מסכים:**

**זרימת עבודה טיפוסית:**

**מסך ראשי → לחיצה על כפתור הפונקציה הרצויה**

**Admin Panel → בחירת מספר דפים → START CRAWLING → BUILD INDEX → SAVE TO FIREBASE**

**Smart Search → הזנת מונח → צפייה בתוצאות מדורגות**

**Analytics → צפייה בגרפים וסטטיסטיקות**

**Quiz → משחק חידון עם ניקוד**

**Help Guide → מדריכים לכל פונקציה**

**ניווט גלובלי:**

**כפתור MAIN MENU זמין בכל מסך למעבר מהיר למסך הראשי**

**כפתור HELP ייעודי לכל מסך**

**עיצוב Glassmorphism עקבי עם גרדיאנט סגול-כחול**

**טעויות אפשריות והסברים:**

**❌ "Please crawl the site first"**

**סיבה: לא נבנה אינדקס עדיין או האינדקס ריק**

**פתרון:**

**לך ל-ADMIN PANEL**

**בחר מספר דפים עם ה-Slider (20 המומלץ)**

**לחץ START CRAWLING**

**המתן לסיום הסריקה**

**לחץ BUILD INDEX**

**לחץ SAVE TO FIREBASE**

**❌ "No results found"**

**סיבה: המונח שחיפשת לא נמצא באינדקס**

**פתרון:**

**נסה מונחים קשורים כמו "mqtt", "protocol", "broker"**

**בדוק איות המילה**

**נסה מילים קצרות יותר**

**השתמש ב-SEARCH ANALYTICS לראות מונחים פופולריים**

**❌ "Firebase connection error"**

**סיבה: בעיית רשת או שרת Firebase**

**פתרון:**

**המערכת עוברת אוטומטית לנתונים מקומיים**

**נסה לרענן הדף ולחזור על הפעולה**

**בדוק חיבור האינטרנט**

**❌ כפתורים לא מגיבים**

**סיבה: הממשק עדיין טוען או עיבוד נתונים**

**פתרון:**

**המתן מספר שניות**

**בדוק שה-Colab Notebook רץ**

**לא ללחוץ מספר פעמים על אותו כפתור**

**⚠️ הודעות אזהרה נפוצות:**

**"Loading data..." - תהליך טעינה רגיל, המתן**

**"Building index..." - בניית אינדקס בתהליך**

**"Saving to Firebase..." - שמירה למסד נתונים**

**💡 טיפים לשימוש מיטבי:**

**השתמש ב-HELP GUIDE בכל מסך למידע מפורט**

**התחל תמיד מ-ADMIN PANEL לבניית אינדקס**

**השתמש ב-SENSOR READINGS לנתונים בזמן אמת**

**נסה את ה-QUIZ להכרת המערכת**

**PROJECT CHATBOT יכול לענות על שאלות ספציפיות**

|  |  |  |
| --- | --- | --- |
| הערת משוב | האם לדעתכם יש צורך בשינוי במערכת בעקבות ההערה? | נימוק |
| **"מלא כפתורים בחלק של הכניסה, קצת מבלבל. אולי כדאי לסדר את הכפתורים אחרת"** | כן - ביצענו קיבוץ לוגי | הוספנו קטגוריות ברורות (Search, Analytics, Admin) וארגנו את הכפתורים בקבוצות |
| **"הכתב קצת קטן" / "להגדיל את גודל הטקסט במסכים"** | כן - הגדלנו גופנים | שיפרנו את הנגישות והקריאות בכל המסכים |
| **"בניית אינדקס לא צריכה להתבצע כל פעם מחדש"** | חלקית - הוספנו שמירה אוטומטית | שיפרנו את מנגנון השמירה ב-Firebase אך שמרנו גמישות לעדכונים |
| **"אנימציות יפות, אבל מאט את המחברת"** | כן - אופטימיזציה | הפחתנו אנימציות כבדות ושיפרנו ביצועים |
| **"צבעים לא נכונים. קשה לראות את הכיתוב הלבן על מסך בהיר"** | כן - שיפרנו ניגודיות | הוספנו צללים ושיפרנו את הניגודיות לכל הטקסטים |

**התייחסות ל-Code Review**

**משוב שקיבלנו מהצוות השני בשבוע 13**

**לאחר ביצוע Code Review**

| **קריטריון Code Review** | **הערה שקיבלנו** | **האם נעשה שינוי בקוד?** | **נימוק להחלטה** |
| --- | --- | --- | --- |
| **מימוש** | **הקוד מיישם את הפונקציונליות המרכזית: יצירת מנוע חיפוש, ניהול נתוני חיישנים והצגתם ב־UI אינטראקטיבי** | **לא נדרש שינוי** | **הערה חיובית - הקוד עומד בכל הדרישות** |
| **יעילות** | **במעבר למידע אמיתי מסנסורים time.sleep(30) על החוט המרכזי יחסום את ה-UI. מומלץ להריץ את start\_mqtt תחת thread נפרד** | **כן - תוכנן לעתיד** | **הוספנו תכנון לשימוש ב-Threading במימוש עתידי עם חיישנים אמיתיים** |
| **פשטות** | **חלק גדול מה־HTML וה־CSS משולב ב־Python במקום תבניות נפרדות, מה שמקשה על הקריאות** | **חלקית** | **הפרדנו חלק מהעיצובים לפונקציות נפרדות אך שמרנו על המבנה הכללי** |
| **מודולריות** | **הקוד מחולק לתאי Colab נפרדים לפי נושאים (crawler, index, UI, sensors), מה שמגביר מודולריות** | **לא נדרש שינוי** | **הערה חיובית - המבנה המודולרי מתאים לסביבת Colab** |
| **באגים וטעויות** | **אין בדיקה מקיפה לתגובות HTTP - requests.get עלול להחזיר קוד שגיאה ולא מטופל כראוי** | **כן - תוקן** | **הוספנו בדיקות status\_code וטיפול מתאים בשגיאות HTTP** |
| **טיפול בשגיאות** | **קיים שימוש ב־try/except אך אין לוג רישום מרכזי ומעבירים שגיאות להדפסה בלבד** | **כן - שופר** | **הוספנו מערכת הודעות ידידותיות למשתמש במקום הדפסות גולמיות** |
| **בדיקות** | **לא נמצאו בדיקות אוטומטיות (unit/integration), כדאי להוסיף מקרים לקצה** | **לא יושם** | **בסביבת Colab קשה ליישם בדיקות אוטומטיות, נשאר לפיתוח עתידי** |
| **שימושיות** | **ה־UI עשיר ומודרני, אך חסר תיעוד למשתמש, מומלץ להוסיף מסך הדרכה** | **כן - הוסף** | **הרחבנו את מסכי HELP GUIDE והוספנו tooltips מפורטים** |
| **תיעוד** | **הקוד כולל docstrings והערות בתאי ה-Colab** | **לא נדרש שינוי** | **הערה חיובית - התיעוד נמצא ברמה טובה** |
| **אתיקה ושקיפות** | **לא מתבצע איסוף נתונים פרטיים אך אין הסבר ברור למשתמש על אופן פעולת האלגוריתם** | **כן - הוסף** | **הוספנו מסך "איך זה עובד?" עם הסבר מפורט על האלגוריתמים** |
| **אבטחה** | **חסרה בדיקת תקפות קלט ושמירת מפתחות Firebase חשופים בקוד** | **כן - תוקן חלקית** | **הוספנו סיסמה לדף Admin וולידציה בסיסית לקלטים** |
| **ביצועים** | **שימוש ב־seaborn ו־plotly יוצר עיבוד כבד ב־frontend; ניתן להמיר לתרשימים סטטיים** | **לא יושם** | **החלטנו לשמור על חוויית המשתמש האינטראקטיבית למרות העומס** |
| **קריאות** | **קוד ארוך במיוחד עם CSS ו־JS משולב. מומלץ לחלק לcells קטנים** | **כן - שופר** | **חילקנו חלק מהתאים הארוכים לתאים קטנים יותר** |

**סיכום השיפורים שיושמו:**

**שיפורים מיידיים:**

**✅ הוספת סיסמה למסך Admin**

**✅ שיפור טיפול בשגיאות HTTP**

**✅ הרחבת מסכי עזרה ותיעוד**

**✅ הוספת הסברים על שקיפות אלגוריתמית**

**✅ חלוקה טובה יותר של תאי קוד**

**שיפורים לעתיד:**

**🔄 Threading למידע אמיתי מחיישנים**

**🔄 הפרדה מלאה של עיצוב מלוגיקה**

**🔄 בדיקות אוטומטיות**

**החלטות שמרניות:**

**📊 שמירה על תרשימים אינטראקטיביים למרות עומס**

**🏗️ שמירה על מבנה Colab הנוכחי**

**מקורות**

**מקורות טכניים:**

**Firebase Documentation** - https://firebase.google.com/docs

**MQTT.org** - התוכן הנסרק למערכת

**Gradio Documentation** - https://gradio.app/docs

**IPywidgets Guide** - https://ipywidgets.readthedocs.io

**מקורות אקדמיים:**

**עקרונות שניידרמן** - "Designing the User Interface"

**SUS Methodology** - System Usability Scale research

**Information Retrieval** - Modern approaches to search engines