**צוות SLOTH**

**קישור למחברת הגוגל קולאב**

<https://colab.research.google.com/drive/1XnUkvo-NkC6eK1nhABqlGszM5qk09MWw>

**קישור לתיקיית הגיט** של הצוות

<https://github.com/reuven-itzhakov/Cloud-Computing>

1. מהות המוצר ומרכיביו

***1.*1 מודולים ופיצ'רים מעניינים**

**עוזר ה-AI – זיכרון שיחה מובנה:**

העוזר שומר את הקשר השיחתי באמצעות שליחה חוזרת של היסטוריית הצ'אט בכל אינטראקציה עם המשתמש. תכונה זו מאפשרת ל-AI "לזכור" את תוכן השיחה ולהגיב בהתאם להקשר המצטבר.

**גרפים מתעדכנים בזמן אמת:**

השתמשנו בספריות Widgets ו-Matplotlib להצגת גרפים דינמיים המתעדכנים בזמן אמת בלחיצה על כפתור Play. פונקציית ההצגה של Widgets קיבלה כפרמטר את פונקציית העדכון, אשר אחראית לרענן את תוכן הגרף בזמן הריצה.

**1.2 microservices שנעשה בהם שימוש**

שירות ה DataManager

אחראי על שליפה של נתוני חיישנים. הנתונים יכולים להגיע ממקורות שונים:

* MQTT - הוא פרוטוקול לתקשורת עם חיישנים בזמן אמת.
* נתונים מדומים (Fake Data) - נתונים שנוצרים לצורכי בדיקות או הדגמות.

השירות מספק ממשק אחיד לקבלת נתונים - כך שהמערכת כולה יכולה להשתמש בנתונים בלי לדעת מאיפה הם באמת הגיעו.

**יתרונות השימוש ב DataManager**

* הפשטה (Abstraction) - השירות מסתיר את המורכבות של חיבור למקורות שונים או נתונים מדומים. שאר המערכת מדברת עם ה DataManager בלי צורך לדעת מהו מקור הנתונים.
* גמישות (Flexibility) - אפשר בקלות להחליף בין מצב של נתונים אמיתיים לנתונים מדומים באמצעות מתודה שנקראת set\_mode מאוד שימושי במצבים של פיתוח, בדיקות או הדגמות כשאין גישה לנתונים אמיתיים.

**שירות ה Indexer**

נועד לבצע סריקה של אתר אינטרנט, לחלץ ממנו תוכן טקסטואלי, לעבד אותו ולבנות אינדקס הפוך שמאוחסן במסד הנתונים Firebase.

**תפקוד**

* סריקת אתרים - השירות מתחיל מכתובת URL ראשי) ועוקב אחרי קישורים פנימיים באתר, עד לעומק מסוים, כדי לאסוף תוכן מדפי האתר.
* עיבוד טקסט - בכל עמוד שנאסף, מחלצים את הטקסט, ממירים את המילים לרותיות קטנות, מסירים מילות עצירה ולבסוף מבצעים stemming.
* בניית אינדקס הפוך - נבנה מבנה נתונים שבו כל מילה ייחודית מצביעה על רשימת עמודים שבהם היא מופיעה, כולל כמה פעמים היא מופיעה בכל עמוד.
* שמירה במסד הנתונים - האינדקס המאובזר מאוחסן במסד הנתונים כדי לשמור עליו בצורה קבועה ונגישה לחיפושים.

יתרונות השימוש ב Indexer

* חיפוש יעיל -האינדקס ההפוך מאפשר לשלוף במהירות עמודים שמכילים מילה מסוימת – בלי צורך לסרוק את כל המסמכים בכל פעם מחדש.
* סקיילביליות - אחסון האינדקס במסד הנתונים מאפשר להתמודד עם נפחים גדולים יותר של נתונים, וגם לתמוך בשימוש מרובה במקביל.
* שימוש חוזר - כל הלוגיקה של הסריקה והאינדוקס מרוכזת במחלקת Indexer - אפשר בקלות להשתמש בה שוב כדי לאנדקס אתרים נוספים.

**1.3 KPI מרכזיים**

הגדרנו מספר KPI רלוונטיים, שמאפשרים להעריך את דיוק המערכת, יעילותה, זמינות הנתונים, ומעורבות המשתמשים. להלן המדדים המרכזיים:

Outlier Detection Accuray1 - דיוק בזיהוי חריגות. אחוז הנתונים שזוהו כחריגים באופן נכון, מתוך כלל הנתונים שנבדקו. מטרת המערכת היא להתריע על חריגות ולכן חשוב לוודא שהמערכת יודעת להבחין בין ערכים תקינים לבין ערכים שאינם סבירים נמדוד באמצעות השוואה בין ערכים ידועים מראש כחריגים לבין אלו שהמערכת סימנה בפועל.

2. Sensor Data Latency -זמן תגובה להצגת נתונים. הזמן שעובר בין קליטת הנתון מהחיישן לבין הצגתו על ה LIVE GRAPH. המערכת מיועדת לעבודה בזמן אמת ולכן זמן תגובה קצר חיוני לשימוש תקין. נמדוד זאת באמצעות מעקב אחרי timestamps והשוואת זמני שליחה מול זמני הצגה.

3. System Uptime - זמן זמינות של המערכת. אחוז הזמן שבו המערכת הייתה פעילה וזמינה להצגת נתונים ולעיבודם. מדד המעיד על יציבות המערכת. נעריך זאת לפי רישום זמני עבודה, השוואה בין זמן כולל לבין זמני נפילה או שגיאות קריטיות. 4. Statistics Processing Tim - זמן חישוב סטטיסטיקות, זאת אומרת כמה זמן לוקח למערכת לחשב ממוצעים, להציג גרפים או להפיק ניתוחים סטטיסטיים עבור החיישנים. מה שמשפיע על חוויית המשתמש, בעיקר כשמדובר על מערכות שעובדות בזמן אמת. נעמוד זאת בעזרת חותמות זמן בתחילת וסיום כל פעולת עיבוד.

**1.4 ניתוח נתוני עתק**

מערכת ניתוח נתוני חיישנים בשיטת MapReduce

מערכות איסוף נתונים מחיישנים מייצרות באופן רציף כמויות גדולות של מידע גולמי, הכולל ערכים נמדדים (כגון טמפרטורה, לחות, לחץ וכו') יחד עם חותמות זמן (timestamps). על מנת לנתח ולהבין מידע זה בצורה יעילה, נדרש לבצע עליו תהליכי עיבוד, מיון וניתוח סטטיסטי.

אחת השיטות הנפוצות והיעילות לעיבוד כמויות כאלה של נתונים היא שיטת MapReduce. זוהי גישה המורכבת משני שלבים עיקריים:

* Map - בשלב זה הנתונים הגולמיים נאספים וממופים למבנה אחיד. כל ערך (למשל מדידת טמפרטורה) מקושר למזהה זמן (timestamp), מה שמאפשר לעקוב אחרי כמה פעמים כל ערך הופיע.
* Reduce - בשלב זה מתבצעת קיבוץ של נתונים דומים: כל ערך ממופה לכמות הפעמים שהוא הופיע. כך ניתן לבנות מבנה שמייצג את ההתפלגות של ערכים לאורך זמן עבור כל חיישן.

לאחר עיבוד הנתונים, ניתן לבצע עליהם ניתוחים סטטיסטיים חשובים כמו חישוב ערך ממוצע, ערך מינימלי ומקסימלי, זיהוי הערך השכיח ביותר ועוד. ניתוחים אלה משמשים להסקת תובנות על מצב הסביבה, תקלות במערכת, אנומליות או זיהוי דפוסים לאורך זמן.

המערכת שבנויה בקוד מתבססת על עקרונות אלו. הנתונים נשלפים ממסד נתונים (Firebase) שבו נשמרו קריאות של חיישנים, עוברים תהליך מיפוי וצימצום, ולאחר מכן מוצגים בצורה מרוכזת ומובנת. מעבר לכך, המערכת מייצרת היסטוגרמה חזותית של ההתפלגות – כלי חשוב להבנת התנהגות ערכי החיישנים.

בסופו של תהליך, מתקבל מערך של טבלאות וסיכומים סטטיסטיים עבור כל חיישן, לצד תצוגה גרפית הממחישה את פיזור הערכים.

תמונה שמכילה צילום מסך, תרשים, צבעוני, ריבוע

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

**מה מוצג בגרף**

* ציר X - הערכים שנמדדו על ידי הסנסורים – לדוגמה: טמפרטורה, לחות, לחץ או מרחק.
* ציר Y - מספר הפעמים שערך מסוים הופיע.
* צבעים שונים - כל צבע מייצג סנסור אחר.

**תובנות מהגרף**

* **טווחי פעילות של סנסורים שונים**
* Humidity(Outdoor), Humidity(Indoor) מופיעים רבות בטווחים הגבוהים (50-60), כלומר נמדדו ערכים גבוהים יחסית.
* Temperature(Outdoor) ,Temperature(Indoor) מתרכזים בטווח 30-50 - טווח אופייני לטמפרטורות סביבתיות.
* Distance(Indoor) נמדד בטווח רחב יחסית (20-55), מה שעשוי להצביע על שינויים רבים בסביבה (כמו תנועה).
* **ערכים דומיננטיים**
* סביב 55-60 נראית קפיצה גדולה בכמות המדידות, בעיקר מצד Humidity(Outdoor), Humidity(Indoor) - אולי מדובר בתקופה או סביבה עם לחות גבוהה.
* באזור 25-35 בולטים Pressure(Indoor), Distance(Indoor) - כנראה סנסורים שפועלים בעיקר בטווחים אלה.

**טבלאות**

תמונה שמכילה טקסט, צילום מסך, גופן, קו

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

**תובנות Humidity**

הלחות הממוצעת זהה בפנים ובחוץ (53.84).

ערך הלחות הנפוץ ביותר כמעט זהה בין Indoor ל-Outdoor.

טווחי הלחות דומים, עם סטייה קלה למעלה בתוך המבנה.

תמונה שמכילה טקסט, צילום מסך, גופן, קו

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

**תובנות Pressure**

ממוצע הלחץ זהה בפנים ובחוץ – 46.15.

הערך הנפוץ שונה: 57.31 בחוץ לעומת 51.05 בפנים.

Indoor מציג טווח רחב מעט יותר, אך קרוב מאוד.

תמונה שמכילה טקסט, צילום מסך, גופן, קו

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

**תובנות Temperature**

הטמפרטורה הממוצעת זהה בין חוץ לפנים (29.98), אבל הערך הנפוץ שונה:

בחוץ: סביב 30.83 מעלות - אולי חשיפה ישירה לשמש.

בפנים: 28.14 קריר יותר, אולי עם מיזוג.

Indoor מציג ערכים מעט יותר קיצוניים (גם נמוך יותר וגם גבוה יותר)

תמונה שמכילה טקסט, צילום מסך, גופן, קו

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

**תובנות Dlight**

טווח עוצמת האור רחב יחסית, בין 28.65 ל־60.78.

הערך הנפוץ (45.02) קרוב מאוד לממוצע (45.35) – זה מרמז על התפלגות יציבה.

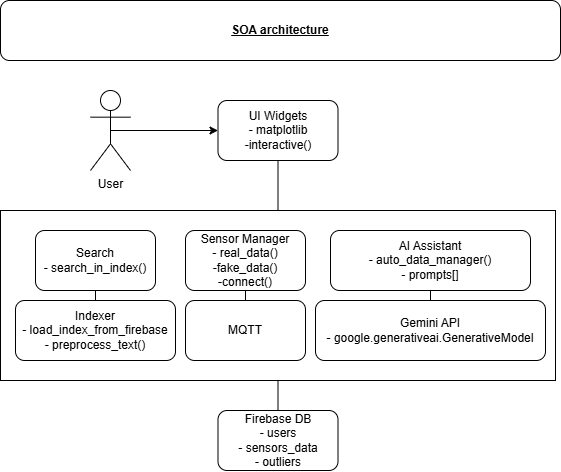
**תובנות Distance**

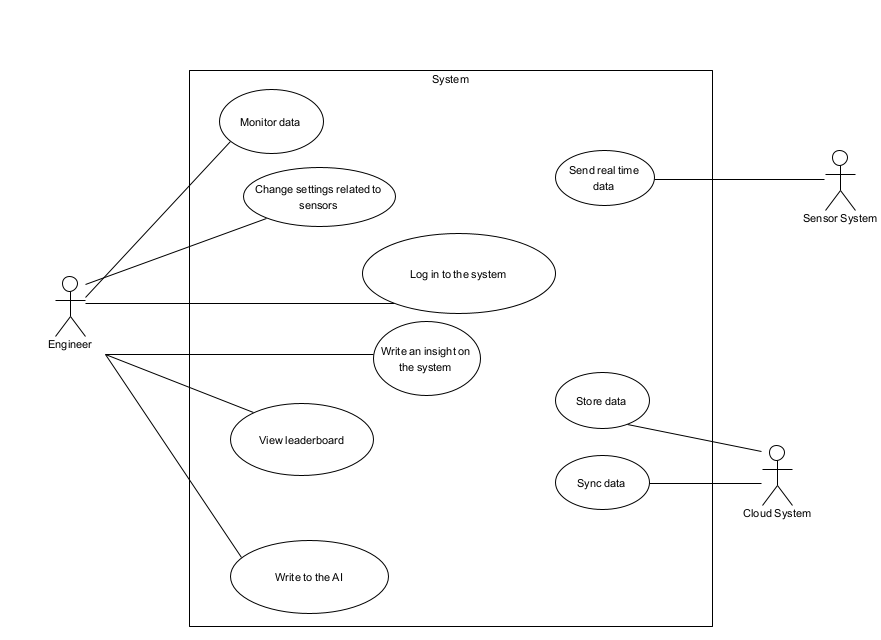
טווח גדול של ערכי מרחק – עשוי להעיד על פעילות דינמית.

הערך הנפוץ (40.87) גבוה מהממוצע (35.22).

2. ארכיטקטורת המערכת

2.1 תרשים ומאפיינים מרכזיים (מעודכן)



**2.2 Use case מעודכן של פונקציונליות המערכת.**

**2.3 דרישות פונקציונליות**

* המערכת תאפשר ניטור בזמן אמת של תהליכי הייצור.
* המערכת תאפשר הצגת נתונים בצורה ויזואלית.
* המערכת תאפשר הגדרת משימות יומיות לשיפור פרמטרים בתהליך הייצור.
* המערכת תאפשר תמריצים ותגמולים על שיפורים בתהליך הייצור.
* המערכת תאפשר השוואת ביצועים מול עמיתים.

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

* המערכת תעבוד בתוך סביבת קולאב, ללא התקנת תוכנות נוספות. (compatibility)
* המערכת תתמוך בקליטה והצגה של נתונים מ-8 חיישנים בו זמנית. (capacity)
* המערכת תתמוך בנתוני זמן אמת של שנייה לחיישן. (capacity)
* כל סטייה מערך סף שהוגדר ב-Admin תסומן בצורה גרפית ברורה. (reliability)
* כל גרף יוצג עם כותרת ברורה ויחידות מידה. (usability)

4. ביקורת עמיתים אשר ניתנה במהלך הסטודיו (שבוע 9) , וכיצד התמודדתם איתה. נא לחשב גם את ציון SUS ולהתייחס אליו.

|  |  |  |
| --- | --- | --- |
| הערת משוב | האם התבצע שינוי באפליקציה בעקבות ההערה? | נימוק |
| עיצוב | כן | אומנם הדגש היה לרוב על הפונקציונליות אך החלטנו בסופו של דבר גם להוסיף עיצוב ידידותי למשתמש |
| צריך להתחבר לדאטה אמיתי, אבל חוץ מזה הכל מעולה | לא | במהלך ההצגה הוחלט להראות את הfake data אך חשוב לציין כי מימוש של real data אכן קיים במערכת, וההתממשקות אליו זהה לfake data ולא דורשת שינוי רב במערכת |
| אפשרות לצפות בחריגות לפי תקופות של חודש, שבוע וכו' | לא | המידע אשר אנו מציגים הוא על פי זמן ספציפי נתון ולכן משוב זה אינו רלוונטי |
| פונט קטן יותר | לא | לאחר ניסיון של שינוי הפונט גילינו כי זה פחות ידידותי למשתמש |
| יותר הבהרה בכפתורים וערכים | כן | אכן כדאי להסביר למשתמש על כפתור ופעולתו |
| חיפוש לא עובד בצורה טובה מופיעים קישורים כפולים | כן | תיקנו בהתאם |
| מקסימום ומינימום של החיישנים צריכים להיות מודעים לפני קבלת מידע מהחיישנים , וזה על מנת לדעת אם יש חריגות בנתונים המתקבלים מחיישנים | כן | כעת ניתן לצפות בערכים הקודמים לפני השינוי עצמו |

תמונה שמכילה טקסט, צילום מסך, מספר, מקביל

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

ציון סופי 88.2432 מעיד על שמישות מעולה.  
מדובר במערכת או מוצר שמתקבלים בצורה מאוד חיובית על ידי המשתמשים. הם מרגישים שהם יכולים להשתמש במערכת בצורה אינטואיטיבית ויעילה, ואין הרבה בעיות שמונעות מהם לבצע את המשימות שלהם בצורה קלה.

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

בכל מסך במערכת שילבנו הדפסות שמסבירות למשתמש מה מתבצע ומאיפה נלקחים הנתונים, כולל פירוט על תהליכי העיבוד. לדוגמה, במסך Raw Data (מסך 2) ובמסך Statistics (מסך 3) מוצגים גרפים של נתוני החיישנים, כאשר ערכים חורגים מודגשים – בין אם כנקודות אדומות או כנקודות מעל/מתחת לגבולות שהוגדרו. הגבולות עצמם מסומנים בקווים אדומים מקווקווים, ומבוססים על ערכי מינימום ומקסימום שהוגדרו במסך המנהל.

**6. אתגרים אשר עלו במהלך העבודה, וכיצד התמודדתם איתם.**

אתגרים שעלו במהלך העבודה וכיצד התמודדנו איתם:

במהלך פיתוח המערכת נתקלנו במספר אתגרים טכניים ולוגיים. כל אתגר חייב אותנו לחשוב בצורה יצירתית, לכתוב קוד גמיש ולהבטיח שהמערכת תישאר יציבה גם בתנאים משתנים. האתגרים המרכזיים והפתרונות שנקטנו:

1. חוסר זמינות של נתוני חיישנים אמיתיים  
    לעיתים קרובות לא קיבלנו נתונים אמיתיים מהחיישנים, בשל תקלות תקשורת או זמינות נמוכה של השירות מצד הספק (בעיה חיצונית שלא תלויה בנו). איך התמודדנו? יצרנו מנגנון של Fake Data כאשר לא התקבלו נתונים אמיתיים, המערכת ייצרה נתונים על בסיס ערכים אחרונים שהתקבלו, תוך שמירה על עקביות סטטיסטית. כך שמרנו על חוויית משתמש רציפה וגרפים עדכניים.
2. הצורך בעבודה עם זרימת נתונים בזמן אמת (Live Data)  
    היה אתגר לוודא שהמערכת תעדכן את הגרפים והסטטיסטיקות באופן חי, בלי השהיות וללא קריסות. השתמשנו במבני נתונים יעילים ומחזורי עדכון חכמים (interval updates), כך שהמערכת יודעת כל כמה שניות למשוך נתונים חדשים ולהציגם בצורה חלקה. בנוסף, יצרנו מנגנון המאפשר לשלוט ב-"Delta Time" – כלומר כמה זמן אחורה מוצג בגרף – כדי לשפר את הביצועים.
3. יצירת ויזואליזציה ברורה ומובנת  
    בהתחלה, הגרפים לא היו מספיק ברורים – קשה היה להבין מהם אילו ערכים חריגים. הוספנו סימון ויזואלי לחריגות (Outliers) באמצעות נקודות אדומות, הסברים בטאב הניהול, והגדרות של מינימום ומקסימום לכל חיישן. כך שיפרנו את ניתוח הנתונים בצורה ויזואלית נגישה.
4. מבנה קוד שגדל והפך למורכב  
    ככל שהמערכת התרחבה – עם יותר טאבים, גרפים ואלגוריתמים – הקוד הפך למסורבל. ארגנו את הקוד בצורה מודולרית – כל פונקציה אחראית על חלק מסוים במערכת (לדוגמה: get\_raw\_data(), show\_average(), create\_login\_widget() וכו’), והפרדנו בין הלוגיקה של הצגת הנתונים ללוגיקה של העיבוד. בנוסף, הוספנו הערות ותיעוד פנימי כדי לשמור על קריאות.
5. התמודדות עם נתונים גדולים לאורך זמן  
    כאשר הגרפים הציגו טווחי זמן ארוכים מדי, הביצועים ירדו. איך התמודדנו? אפשרנו למשתמש לבחור את טווח הזמן של התצוגה באמצעות Delta Time, מה שמאפשר שליטה על כמות הנתונים המעובדים בכל רגע, ומשפר את תגובתיות הממשק.

לסיכום כל אתגר שעלה חייב אותנו להתאים את הפתרון בצורה חכמה, לחשוב מראש על מקרים קיצוניים ולשמור על איזון בין חוויית משתמש, ביצועים ודיוק. ההתמודדות עם האתגרים חיזקה את היכולת שלנו לעבוד בצוות, לכתוב קוד איכותי ולפתור בעיות אמיתיות מהשטח.

**7 . תיק תחזוקה – תיאור של כל הקבצים והאובייקטים המרכזיים, ותיעוד קצר של כל פונקציה בקוד.**

קבצים: כל הקוד נמצא בקובץ אחד Sloth.py, מכיל את: הממשק הגרפי, קוד צד שרת, מיקרו שירותים, UI.

**פונקציות מרכזיות:**

1. Indexer - מיקרו שירות לאינדוקס תכנים מאתר אינטרנט (במקרה שלנו mqtt.org)

crawl - סורק את כל הלינקים מהלינק המרכזי.

store\_index - שומר את האינדקס ב-Firebase.

scrape - סורק מילים ומסנן לפי שכיחות.

2. DataManager - מנהל נתוני החיישנים.

get - מחזיר קריאת חיישנים (אמיתית במידה והחיישנים פעילים / פייק במידה והחיישנים לא פעילים)

connect - מתחבר ל-MQTT לקבלת נתונים בזמן אמת (במידה והחיישנים פעילים).

auto\_data, real\_data, fake\_data - מייצרים נתוני חיישנים במצבים שונים.

3.ממשק משתמש - UI

get\_raw\_data - מציג גרף נתונים חיים עם תצוגת חריגות (במידה והחיישנים פעילים).

check\_outliers\_batch - מזהה ערכים חורגים לפי ערכי קיצון.

save\_delta\_time, change\_valid\_values - מאפשרים למשתמש לבחור מתוך רשימת חיישנים את החיישן הרצוי , ולהגדיר מינ/מקס או את הפרש הזמן.

create\_graph\_viewer - גרף סטטיסטי לפי חיישן.

create\_login\_widget - לוגין למערכת משחוק.

show\_leaderboard - הצגת טבלת ניקוד משתמשים.

show\_ai\_assistant - תיבת שיחה עם עוזר AI.

4. search - מפעיל את מנוע החיפוש

load\_index\_from\_firebase - טוענת את האינדקס ההפוך ממסד הנתונים Firebase.

preprocess\_text - מנקה, מנרמל ומבצע stemming על טקסט לקראת חיפוש.

search\_in\_index - מבצע את פעולת החיפוש עצמה על גבי האינדקס הטעון, ומחזיר רשימת תוצאות מדורגת לפי התאמה.

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

המערכת שפותחה היא דשבורד ייעודי למהנדסים, המאפשר צפייה בערכי חיישנים ובנתונים נוספים.  
 המערכת כוללת 7 מסכים, כמפורט להלן:

**מסך 1Administrator**   
במסך זה מוצג מצב החיישנים בזמן אמת, כלומר ברגע הרצת הקוד:

* חיישן שאינו זמין יוצג בצבע אדום – "לא זמין".
* חיישן זמין יוצג בצבע ירוק – "זמין".

מתחת למצב החיישנים קיים כפתור Refreshהמאפשר לבצע ריענון על מנת לבדוק שוב את זמינות החיישנים ולזהות שינויים במצבם (למשל, חיישנים שהפכו זמינים יוצגו כעת בירוק).

בנוסף, ניתן במסך זה לשנות את ערכי הסף התקינים של החיישנים:

* יש לבחור תחילה את החיישן הרצוי.
* להזין את הערכים החדשים.
* ולאחר מכן ללחוץ על כפתור Save לצורך שמירת השינויים.

בנוסף, קיימת אפשרות לשנות את פרק הזמן עבור הצגת הנתונים בגרף הלייב (הנמצא במסך 2Raw Data ) לכל חיישן בנפרד.  
 לדוגמה, אם בוחרים את החיישן Humidity (Outdoor) ומזינים את הערך 40 – אזי בגרף הלייב שבמסך השני, פרק הזמן של הנתונים שיוצגו יהיה 40 שניות.

כמו כן, קיימת אופציה לצפות בערכים החורגים של כל חיישן.  
 ערכים אלו מגיעים מהדגימות שנשמרו במסד הנתונים ומוגדרים כ"חריגים" כאשר הם חורגים מגבולות הערכים התקינים שהוגדרו קודם לכן.  
 שימו לב: כאשר מתבצע שינוי בערכים התקינים, מתבצע גם עדכון אוטומטי של הערכים החורגים בהתאם.

**מסך 2 Raw Data**   
במסך זה ניתן לבחור חיישן ולאחר מכן ללחוץ על כפתור החץ, כדי לצפות בגרף לייב של ערכי החיישן הנבחר.  
 יש לשים לב:

* אם החיישן נגיש וזמין, יוצגו ערכי אמת מקריאותיו בזמן אמת.
* אם החיישן אינו זמין, יוצגו ערכים פקטיביים המיוצרים על בסיס הקריאה האחרונה ממנו.

משך הזמן המוצג בגרף הוא לפי הערך שהוגדר עבור אותו חיישן במסך הראשון.  
 בגרף מוצגים:

* ממוצע הנתונים בפרק הזמן האחרון.
* וכן ערכים חורגים, בהתאם להגדרות שנקבעו במסך הראשון.

**מסך 3Statistics**   
במסך זה ניתן לצפות בממוצע ערכי חיישן שנדגמו ונשמרו במסד הנתונים.  
 לשם כך, יש לבחור את החיישן הרצוי.

בנוסף, ניתן להציג גרף סטטיסטי של ערכי החיישן:

* הגרף מציג את כל הערכים ההיסטוריים שנדגמו ונשמרו במסד הנתונים.
* כולל סימון ערכים חורגים, המסומנים באמצעות קווים אדומים מקווקווים לפי גבולות הערכים התקינים.

קיימים שני כפתורים נוספים:

* Show All Sensorsמציג את כל הגרפים של כלל החיישנים.
* Hide All Graphsמסיר את כל הגרפים מהתצוגה.

בשל ריבוי הנתונים והעומס שכרוך בהצגת גרפים מרובים, הוספנו מנגנון מניעת טעויות:

* נחסמת האפשרות ללחוץ שוב על כפתורים מסוימים עד לסיום טעינת הגרפים במלואם.
* בנוסף, מוצגת הודעה מתאימה למשתמש.

**מסך 4 Search**   
מסך החיפוש נבנה בהתאם לדרישות שהוגדרו בתרגיל הבית.  
 המסך מאפשר לבצע חיפוש באתר [https://mqtt.org](https://mqtt.org/) ולהציג:

* קישורים רלוונטיים מתוך האתר.
* וכן את מספר המופעים של מילת החיפוש בכל דף תוצאה.

**מסך 5 Login**   
במסך זה המשתמש יכול להתחבר למערכת באמצעות הזנת שם משתמש וסיסמה.  
 לאחר מכן יש ללחוץ על כפתור ההתחברות.

לאחר ההתחברות, ניתנת למשתמש האפשרות להוסיף תובנה.  
 תובנות אלו יוצגו במסך ה Leaderboard והן מקנות נקודות למשתמש.  
 על מנת להתחבר יש לראות בתחילת הקובץ שם משתמש וסיסמה של המשתמשים הרשומים במערכת.

**מסך 6 Leaderboard**   
במסך זה מוצגת טבלה המכילה את כל המשתמשים הרשומים במערכת.  
 לכל משתמש מוצגים:

* מספר הנקודות שצבר (כאשר כל תובנה מקנה נקודה אחת)
* ורשימת התובנות שהוא הזין.

הטבלה ממוינת לפי מספר הנקודות, בסדר יורד.

כדי לצפות בשינויים שהתרחשו בזמן אמת (למשל, הוספת תובנה חדשה למשתמש בשםEden ) יש ללחוץ על כפתור Refresh לצורך רענון הטבלה.

**מסך 7 AI Assistant**   
במסך זה מופיע צ’אטבוט חכם – העוזר הווירטואלי של המערכת.

יכולות הצ’אטבוט כוללות:

* זיכרון של השיחה.
* ידע נרחב על המערכת כולה.
* יכולת סיוע למשתמשים בהבנת הפעולות השונות ובשימוש מושכל בכלי המערכת.
* וכן גישה לערכי החיישנים בזמן אמת, המאפשרת הסקת מסקנות וניתוח עומק תוך כדי השיחה.

**מניעת טעויות משתמש**  
 במהלך פיתוח המערכת הושם דגש רב על שמישות, נוחות ומניעת שגיאות:

* לכל רכיב נוסף הסבר מילולי, המבהיר את תפקידו ואת ההשפעה שיש לשינוי בו על רכיבים אחרים.
* במסך הסטטיסטיקה, מאחר ועיבוד כמויות גדולות של מידע עלול להימשך זמן, ננקטו צעדים למניעת טעויות:
  + חסימת זמנית של כפתורים עד להשלמת הטעינה.
  + הצגת הודעה מתאימה למשתמש על הפעולה המתבצעת.

9. בשבוע 11 בקורס תציגו את המערכת שלכם.

ההצגה תכלול הרצת המערכת/סרטון, וכן הצגה של הקוד והאלמנטים המרכזיים מסעיפים 1-6 (טקסט בלבד, אין צורך במצגת).

כל חברי הצוות חייבים להשתתף בהצגה, וכן יש להקפיד על הצגה ברורה ומתומצתת.

עליכם להתייחס להצגות בסשן שלכם, ולמלא משוב לפחות ל -5 צוותים ביום בו הצגתם. ההצגה ומילוי המשוב הם **חובה לכל חברי הצוות על מנת לקבל ציון.**

לאחר ההצגה:

תקבלו באופן אנונימי את המשובים של חבריכם, וכן את המשוב שלנו.

יש להגיש את הטבלה הבאה , תוך התיחסות למשובים שקיבלתם:

|  |  |  |
| --- | --- | --- |
| הערת משוב | האם לדעתכם יש צורך בשינוי במערכת בעקבות ההערה? | נימוק |
| הגרפים קצת קטנים אפשר להגדיל אותם אבל הפונקציונליות שלהם בסדר גמור | לא | הוחלט שכאשר כל הגרפים מוצגים יחד, הם יוצגו בגודל מוקטן יחסית, כדי להקל על השוואה בין גרפים של חיישנים שונים ולאפשר צפייה בכל הגרפים במקביל. כאשר מוצג גרף של חיישן בודד, הוא מוצג בגודל גדול יותר במכוון, על מנת לספק חוויית צפייה נוחה וממוקדת |
| הגרפים ברורים וקלים להבנה אך גם מעט חסר עיצוב | לא | לא התמקדנו בעיצוב המערכת, מאחר שלא הייתה לכך דרישה. העיצוב שנבחר הוא בסיסי, כאשר עיקר הדגש הושם על הפונקציונליות של המערכת. |
| בהתחברות להוסיף היררכיה ברורה לדף  בלוח נקודות יש יותר מידי מידע | לא | מסך ההתחברות נועד אך ורק לצורך התחברות כמשתמש לצורך הוספת תובנות, וזאת כדי לדמות תהליך התחברות בסיסי. בלוח הנקודות מוצג דירוג המשתמשים – הדבר מצוין במפורש, ובנוסף מדובר בהתנהגות טבעית שבה המשתמש בראש הטבלה הוא בעל הניקוד המרבי |
| המערכת לא מעוצבת מספיק | לא | בשל היעדר דרישה לעיצוב גרפי, המערכת אינה מעוצבת באופן מתקדם. עם זאת, היא פשוטה ונוחה לשימוש, ומכילה הסברים ברורים בכל מסך, מה שמקל על המשתמשים בהתמצאות ובתפעול המערכת |

11. התייחסות למשוב שקיבלתם בשבוע 13 – code review – יש להגיש את המשוב שקיבלתם מהצוות השני בחלק הקבוצתי , ולכתוב התייחסות מתאימה

|  |  |  |
| --- | --- | --- |
| קריטריון | **הסבר** | **התייחסות** |
| מימוש | האם הקוד מבצע את הנדרש? פרטו.  תשובה: כן, הקוד מבצע את הכי מינימום שנדרש , כלומר נטו פונקציונאלי | תגובה: מרוצים מההערה  שינוי: לא |
| יעילות | האם ניתן לשפר את יעילות הקוד?כיצד? תשובה: לא, מבחינה פונקציונלית הקוד פשוט ויעיל ככל האפשר | תגובה: מסכימים  שינוי: לא |
| פשטות | האם ניתן לפשט את הקוד?כיצד?  תשובה: לא , הקוד מאוד פשוט | תגובה: ״תודה״ מסכימים  שינוי: אין |
| מודולריות | האם הקוד מודולרי מספיק? (חלוקה תקינה לקבצים, פונקציות ומחלקות)  תשובה: כן החלוקה ברורה בתוך כל cell אך רק צריך לתת לcells שמות משמעותיים, למשל בעזרת @title לדפדוף יותר מהיר בקוד בקולאב | תגובה: לא למדנו את זה  שינוי: לא משנים בהמשך כי לא נלמד במסגרת הקורס לפי טענתיהם |
| באגים וטעויות | האם ישנם מקרים בהם הקוד לא מתנהג כצפוי? אילו? תשובה: לא, אין באגים, הכל עובד תקין | תגובה: מסכימים  שינוי: לא |
| 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? תשובה: טיפול בשגיאות הוא טוב ונראה במסך ה LOGIN כאשר מזינים קלט שגוי, וגם print statements נמצאים לדיבוג בקוד | תגובה: מסכימים  שינוי: לא |
| בדיקות | האם יש בדיקות שניתן להוסיף? תשובה: לא שידוע לנו | תגובה: מסכימים  שינוי: לא |
| שימושיות | האם הממשק שימושי? (usability)? האם המערכת מעוררת רצון להמשיך להשתמש בה?  תשובה: לא , חוסר עיצוב , חוסר אינטראקציה עם המערכת לא מעודדת המשך להשתמש בה באף אופן | תגובה: לא היה חייב לעשות עיצוב מיוחד כלשהוא ולכן השתמשנו רק בwidgets ו matplotlib  שינוי: לא |
| תיעוד | האם הקוד מתועד?  תשובה: הקוד מתועד היטב . | תגובה: מסכימים  שינוי: לא |
| אתיקה ושקיפות | האם יש שימוש בנתוני משתמש אשר מפרים פרטיותו?האם ישנו הסבר למשתמש כיצד האלגוריתם מתנהג? האם ישנו algorithmic bias כנגד קבוצה או פרטים מסוימים?  תשובה: יש שקיפות מילולית בממשק לכל tab מה יש ומציג למשתמש, אין algorithmic bias הכל מנותח לרוב הנתונים | תגובה: מסכימים  שינוי: לא |
| אבטחה | האם ישנו מידע אבטחה (שמות משתמש, ססמאות) גלויים?  תשובה: הכל נמצא במסד הנתונים וזה מצויין ! | תגובה: מסכימים  שינוי: לא |
| ביצועים | האם שינוי עתידי בקוד יכול לגרום לפגיעה בביצועים?כיצד ניתן לשפר?  תשובה: בגדול הכל בסדר אבל הנתונים של החיישנים מצריכים זמן רב בכדי לטעון | תגובה: מסכימים  שינוי: לא, כמות הנתונים היא גדולה ולכן לוקח זמן רב לטעינה |
| קריאות - readability | האם הקוד מובן בקלות?אילו חלקים בקוד היו לא ברורים עבורך?כיצד ניתן לשפר את קריאות הקוד?  תשובה: כן הקוד קריא , אפשר להוסיף section לכל cell | תגובה: מסכימים  שינוי: לא |

13. מקורות. יש לצטט את המקומות מהם אתם לוקחים את הנתונים השונים. יש לכלול פרומפטים לכלי AI, במידה והשתמשתם בהם.

השתמשנו בעיקר ב chat GPT ונתנו לו את הפרומפטים הבאים:

"איך ליצור טבלה ללא שימור בHTML בקולאב"

"איך לשנות צבעים בגרפים של np.plot"

"מדוע השליפה ממסד הנתונים כל כך איטית והאם עדיף לשלוף את כל הנתונים בפעם אחת ולעשות להם את העיבוד בפייתון?"

[Google.com:](https://www.google.com/)

[average israel temperature](https://www.britannica.com/place/Israel/Climate):

“Average annual temperatures vary throughout Israel based on elevation and location, with the coastal areas adjacent to the Mediterranean Sea having milder temperatures—ranging from about 84 °F (29 °C) in August to about 61 °F (16 °C) in January”

[average atmospheric pressure](https://www.noaa.gov/jetstream/atmosphere/air-pressure#:~:text=remains%20the%20same.-,The%20standard%20pressure%20at%20sea%2Dlevel%20is%201013.25%20in%20both,the%20atmosphere%20decreases%20with%20height.):

The standard pressure at sea-level is 1013.25 in both millibars (mb) and hectopascal (hPa). The number of molecules in the atmosphere decreases with height.

[average humidity:](https://www.climate.top/israel/tel-aviv/humidity/)

| [**Jan**](https://www.climate.top/israel/tel-aviv/january/) | [**Feb**](https://www.climate.top/israel/tel-aviv/february/) | [**Mar**](https://www.climate.top/israel/tel-aviv/march/) | [**Apr**](https://www.climate.top/israel/tel-aviv/april/) | [**May**](https://www.climate.top/israel/tel-aviv/may/) | [**Jun**](https://www.climate.top/israel/tel-aviv/june/) | [**Jul**](https://www.climate.top/israel/tel-aviv/july/) | [**Aug**](https://www.climate.top/israel/tel-aviv/august/) | [**Sep**](https://www.climate.top/israel/tel-aviv/september/) | [**Oct**](https://www.climate.top/israel/tel-aviv/october/) | [**Nov**](https://www.climate.top/israel/tel-aviv/november/) | [**Dec**](https://www.climate.top/israel/tel-aviv/december/) |  | **Annual** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| relative humidity icon | Relative Humidity (%) | 72 | 75 | 74 | 71 | 68 | 68 | 70 | 70 | 69 | 67 | 69 | 73 |  | 70.5 |