תרגיל בית 3 – Unicorn

קישור למחברת: <https://colab.research.google.com/drive/18fgHvLeaDzJPNGH-kZnDVnLQcYmR4iD4?usp=sharing>

קישור לגיט: <https://github.com/guyshargo/Unicloud.git>

קישור לסרטון: <https://drive.google.com/file/d/16nogYCXwgi1YnEGp55YPjI5wV2Q4yZVG/view?usp=sharing>

מהנדס מערכת: סתו אברהם

| שם חבר הצוות | משימות שהוקצו | משימות שהושלמו | **Acceptance test –** בדיקת קבלה |
| --- | --- | --- | --- |
| שני פחימה | שקיפות אלגוריתמית, מסך big data, יצירת אינדקס, יצירת התראות על חריגות. | שקיפות אלגוריתמית, מסך big data, יצירת אינדקס, יצירת התראות על חריגות. | 1. הכנסת ערך חריג יוצרת רשומת ‎/outliers ומציגה כרטיס חריגה אדום.  2. לחיצה על Run Big-Data Analysis מריצה את צינור Spark ומציגה גרף תקין.  3. קריאה ל-crawl() מוסיפה ≥ 50 דפים לאינדקס וחיפוש “mqtt” מחזיר תוצאה.  4. בעת הרצת האינדקס , האינדקס נשמר בRAM של האינדקס סרוויס ולא בדאטה בייס. |
| ירדן נחום  207965401 | צאטבוט ומסך task manager , תיק משתמש | צאטבוט ומסך task manager , תיק משתמש | **צ'אט בוט**   * ניתן להזין שאלה ולקבל תשובה ע"י לחיצה על "Send" או Enter. * הצאט יחזיר תשובה עבור שאלות על המערכת עצמה או על מושגים מ mqtt * לחיצה על "Clear Chat" מנקה את ההיסטוריה.   **Task Manager**  **טעינת משימות** בעת כניסה לטאב, הפונקציה refresh\_task\_table מופעלת ומציגה את כל המשימות מהמסד נתונים בטבלה  * כאשר משימה נמצאת בסטטוס "Done", מוצג לצידה כפתור Confirm בתוך טבלה   **פילטרים משימה**   * כאשר המשתמש משנה ערך באחד מתיבות הסינון filter\_status filter\_priority, או filter\_assignee, הפונקציה refresh\_task\_table מופעלת מחדש, ומציגה בטבלה רק את המשימות שתואמות את תנאי הסינון  **עריכת משימה**  * כאשר המשתמש בוחר מזהה משימה מתפריט הפונקציה on\_task\_select change נטענת ומציגה את פרטי המשימה בטופס העריכה, הכולל השדות מזהה משימה כותרת, סטטוס, עדיפות והעובד שהמשימה שייכת לו, תאריך תחילת משימה וסיום * בלחיצה על כפתור " Update Task" מפעילה את הפונקציה update\_task, המעדכנת את המשימה במסד נתונים, ולאחר מכן מפעילה מחדש את refresh\_task\_table לרענון התצוגה.   **מחיקת משימה**   * כאשר לוחצים על כפתור " Delete Task", מופעלת הפונקציה delete\_task, המציגה תיבת אישור מסוג עם האפשרויות "Cancel" ו־"Yes, Delete". * אם המשתמש בוחר "Yes, Delete", הפונקציה handle\_confirm מבצעת מחיקה של המשימה ממסד הנתונים ומרעננת את הטבלה עם refresh\_task\_table, ומאפסת את טופס ההוספה |
| גיא זמיר | microservices ,מסך big data, יצירת אינדקס, יצירת התראות על חריגות. | microservices ,מסך big data, יצירת אינדקס, יצירת התראות על חריגות. | 1.בעת הרצת האינדקס , האינדקס נשמר בRAM של האינדקס סרוויס ולא בדאטה בייס.  2. לחיצה על Run Big-Data Analysis מריצה את צינור Spark ובמידה ולא מצליחה ליצר גרף מודיעה על כך בהודעת UI.  3.result\_service.format\_results מחזיר קישורים וpage rank לדפים המתאימים לשאילתה של המשתמש. |
| עומרי שפיצר  208275313 | מסך כניסה ויציאה למשתמש, משימות עובד, חיישנים של עובד, דף סטטיסטיקות | מסך כניסה ויציאה למשתמש, משימות עובד, חיישנים של עובד, דף סטטיסטיקות | **מסך כניסה ויציאה:**   * בעת מילוי פרטי משתמש (סיסמה ושם משתמש נכונים) המשתמש נכנס בהצלחה לאפליקציה. * בלחיצה על logout המשתמש יוצא מהמערכת בהצלחה. * בעת מילוי פרטי משתמש (סיסמה ושם משתמש לא נכונים) המשתמש מקבל הודעת שגיאה.   **מסך משימות לעובד:**   * בלחיצה על פילטורי ה-drop-down, המערכת תסנן את משימות העובד על פיהן. * בלחיצה על כפתור העריכה, סטטוס משימת העובד תעודכן בהתאם. * כאשר לעובד אין משימות, תוצג הודעה בהתאם.   **מסך חיישני העובד:**   * המערכת מראה את החיישנים המוצגים לעובד, שהמנהל הגדיר. * כאשר אין החיישנים שהוגדרו כפעילים לעובד, תצד הודעה בהתאם.   **מסך הסטטיסטיקות:**   * בלחיצת כפתורי סינון החיישנים, יוגדרו החיישנים שנבחרו בגרפי הסטטיסטיקה. * בלחיצת כפתור יצירת הגרפים, המערכת תציג את הגרפים על פי החיישנים שנבחרו. * כאשר תאריכי הסינון לא תואמים לתאריכי הקצה של החיישנים, תוצג הודעת שגיאה. * בלחיצת כפתור Clear Graphs, תוצג למשתמש דף ריק מגרפים. * בלחיצת כפתור Export To Excel, תתבצע הורדה של קובץ אקסל עם ערכי החיישנים שנבחרו. |
| סתו אברהם  209155381 | מסך חיישנים למנהל ועובד, שמירת הנתונים של החיישנים, KPI, נקודות לשיפור ושימור | מסך חיישנים למנהל ועובד, שמירת הנתונים של החיישנים, KPI, נקודות לשיפור ושימור | **מסך חיישנים:** 1. בלחיצה על כפתור Toggle isShow במסך המנהל, הערך מתעדכן ב־Firebase ומוגדר אם החיישן יוצג לעובד. בעת טעינת הגרפים של החיישנים, המערכת תטען את מצב הisShow מFireBase, כך שהנתונים יישמרו בכל הרצה של המערכת.  2. בעת שינוי ערך isShow לחיישן, מסך העובדים מתעדכן ומציג רק את החיישנים שסומנו כ־true.  3. בלחיצה על כפתור Refresh, המערכת מתחברת זמנית ל־MQTT, טוענת נתונים עדכניים או מדמה ערכים במקרה של ניתוק, ומעדכנת את הגרפים בהתאם.  4. בעת קבלה של נתון חדש (אמת או מדומה) המערכת שומרת את הערך בFireBase על פי ערכו. המערכת שומרת גם את הערך של הנתון האחרון.  5. כאשר ממוצע 6 הערכים האחרונים מוגדר כחריג, צבע הגרף ישתנה מירוק לאדום כדי לסמל חריגה באופן ויזואלי. |
| גיא שרגורודסקי  206372914 | אלמנט משחקי (חנות הטבות), leaderboard, מסך task manager, תיק משתמש, סרטון. | אלמנט משחקי (חנות הטבות), leaderboard, מסך task manager, תיק משתמש, סרטון. | **Task Manager****אישור משימה שהושלמה**.  * לחיצה על הכפתור confirm ליד משימה, מוסיפה לעובד שאליו שייכת המשימה את מספר הנקודות שמוגדר בשדה Score של המשימה והמשימה נמחקת ממסד הנתונים, ותצוגת הטבלה מתרעננת על ידי קריאה ל־refresh\_task\_table. לאחר העדכון מוצגת הודעת אישור   **Store** **טעינת פריטים מהחנות** בעת טעינת הטאב, הפונקציה get\_sorted\_store\_items קוראת את נתוני הפריטים ממסד הנתונים ומחזירה רשימה ממוינת לפי הסדר שלהם במסד נתונים  **בעת קניית פריט מהחנות:**  בעת לחיצה על פריט מהתיבה item\_dropdown תוצג תיבת אישור רכישה עם שם הפריט, עלותו, והודעת שאלה ושני כפתורים yes ו cancel  בעת לחיצה על כפתור אישור "yes" מחיר המתנה יורד מהניקוד של המשתמש ומתעדכן ב firebase הניקוד החדש והיסטוריית הרכישות שלו, ומוצגת למשתמש הודעת הצלחה **טבלת דירוג עובדים (Leaderboard)** **בעת טעינת טאב החנות, טבלת הדירוג מציגה את העובדים**   * רשימת העובדים נלקחת ממסד הנתונים ומוצגים בטבלה על פי שדה score מהגבוה לנמוך.  **עדכון טבלת הדירוג לאחר רכישה** **בעת ביצוע רכישה מהחנות:** לאחר ביצוע הרכישה:  הפונקציה build\_leaderboard\_html נטענת מחדש ומציגה את הנתוני העובדים העדכניים. |

1. **בפרויקט השתמשנו בשלושה microservices:**

Index Service:אחראי לבניית אינדקס מהמילים במסמכים ושמירתו.

Query Service:מקבל את שאילתת החיפוש של המשתמש ומחזיר את האתרים התואמים לפי הIndex Service.

Result Service: ממיר את התוצאה שמתקבלת מה- Query Service לתוצאה מלאה עם מהתוכן המוצגת למשתמש.

היתרון בעבודה עם Microservices הוא הפרדה ברורה בין שלבי התהליך, מה שמקל על איתור בעיות, מאפשר תחזוקה והרחבה של כל חלק במערכת בנפרד, ותומך בארכיטקטורה מודולרית.

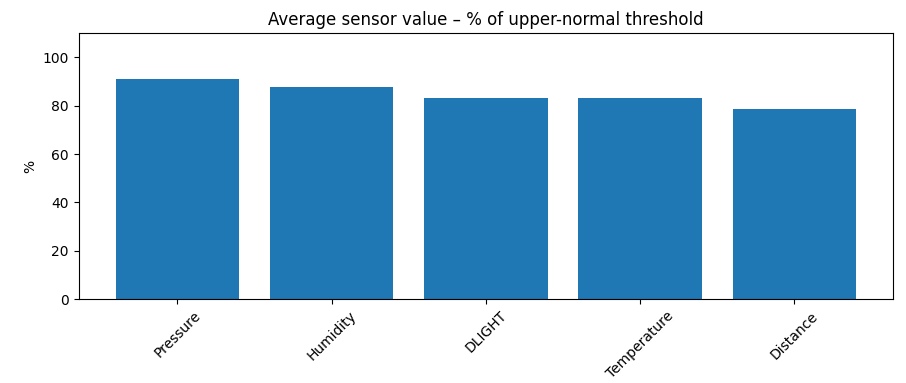
בנוסף, האינדקס מאוחסן בזיכרון (RAM) בתוך Index Service במקום שליפות ממסד נתונים, מה שמאפשר זמן תגובה מהיר במיוחד ומייעל את ביצועי המערכת בזמן אמת.

1. **KPI הרלוונטיים בפרויקט שלכם?**
2. **Respone Time\Latency**זמן התגובה של המערכת מתואר ע"י הזמן הממוצע שלוקח מרגע שליחת הבקשה על ידי המשתמש ועד לקבלת תגובה מהמערכת. הזמינות מאפשרת שליטה ותמרון בזמן אמת במערכת, מעקב אחר תקלות ומניעת אובדן או אי דיוק בנתונים. ככל שזמן התגובה או הטעינה יהיו קצרים יותר, כך תעלה היכולת למנוע אובדן של נתונים או אי דיוק.
3. **Reliability**אמינות המערכת מתוארת על ידי היכולת של המערכת לספק תוצאות אמינות ונכונות באופן עקבי, ללא תלות בתחזוק המערכת לאורך זמן.  
   אמינות המערכת נמדדת כאשר ניטור הנתונים מהמערכת משקף באופן אמין את הנעשה בפועל.
4. **User Satisfaction Score**חוויות המשתמש מתוארת ע”י ביצוע ומעקב אחר המשימות היומיות של עובדי המערכת. ככל שהעובדים יבצעו את משימותיהם כך תוכל המערכת לעקוב באופן אמין ומדוייק יותר אחר הספק העובדים.
5. **Mean Time To Recovery**זמן ההתאוששות מתואר על ידי הזמן הממוצע שלוקח למערכת מהרגע שהיא זיהתה כשל ועד להתאוששות. במערכת שמטרתה לעקוב אחר ניטור חיישנים, בכל כשל כגון ניתוק מMQTT עלול לגרור אובדן של נתונים. ככל שזמן ההתאוששות קצר יותר, כך המערכת תספק נתונים אמינים יותר.

4. **שקיפות אלגוריתמית**

* **Dashboard** - בכל כרטיס חיישן מוצגים יחידת המידה של החיישן וערך נוכחי. צבע הרקע של החיישן מתחלף לירוק/אדום בהתאם לבדיקת החריגה- ירוק אין חריגה, אדום יש חריגה. בנוסף אם המשתמש רוצה לשאול "מהם ערכי החריגה של החיישנים?" הוא יכול להפנות את השאלה לצ'אט ולקבל שקיפות על המידע.
* **חיפוש** - אחרי כל שאילתה אנו מציגים לכל תוצאה את מספר ההופעות המדויק של מילות המפתח ואת דירוג Page Rank המחושב. בכך המשתמש רואה במפורש את הקריטריונים שיצרו את סדר התוצאות .
* **סטטיסטיקה** - בדף מוצגים גרפי מידע על החיישנים הנבחרים, בנושאים: ממוצע ערכי, אחוזי שינוי מהממוצע הכללי ושוני ערכי מהערך הראשון. בנוסף, כל חיישן צבוע בצבע ייחודי, כדי לאפשר זיהוי מהיר. בכפתור “Export to Excel” אפשר להוריד גיליון ובו כל ערכי הגרפים הנבחרים, השמורים בעמודות נפרדות, כך שהמשתמש יכול לאמת את הנוסחאות בעצמו.
* **חיישנים**- מעל כל גרף מצויין שמו של החיישן ואת הערכים שלו בגרף. כל חיישן צבוע בצבע המצביע על חריגה בממוצע הכללי של הסנסור המוצג.

1. **מידע עתק:** עבור כל היסטוריית הנתונים של כל חיישן, חושב ממוצע לכל חיישן, והממוצע נורמל לאחוזים על מנת להציג את הממוצעים של כל החיישנים באותו הגרף:



1. **אתגרים שעלו במהלך העבודה , וכיצד התמודדתם איתם**

**סביבת עבודה**העבודה במחברת אחת בGoogle Colab היוותה אתגר בעבודה בקבוצה גדולה. נתקלנו בקושי בתיאום גרסאות, דריסת משתנים ורציפות בעבודה. כדי להתמודד עם האתגר התחלקנו לצוותי עבודה, והקפדנו לשמור על תקשורת בעבודה על המחברת כדי לא לאבד מידע וזמן יקר.

**ממשק Google Colab**במהלך העבודה נוכחנו לגלות כי שימוש בלולאות אינסופיות ופקודת sleep חוסמים את ממשק הUI במערכת, וכך גורמים לתקלה בפונקציונליות. מאחר שהחיבור לMQTT מתבצע באמצעות לולאה אינסופית מצאנו דרכים יצירתיות לשמר את הפונקציונליות של המערכת באמצעות כפתור לרענון הנתונים, וכך לשמר את קבלת הנתונים בזמן אמת.

**שימוש בBig Data עם Apache Sprak**   
שימוש בSpark האט את ביצועי המערכת והקשה על עבודה רציפה ואיתור ליקויים. לאחר חשיבה קבוצתית ומעקב אחר התוכנית הצלחנו לפענח את הבעיה שהייתה בעומס נתונים בFireBase. לאחר שהחלטנו על כמות הנתונים שיש לשמור והפחתנו אותה משמעותית הצלחנו לפתור את הבעיה.

8. **תיק מתכנת:**

### **מיקרו-סרביסים (Microservices)**

1. :IndexService

מנהל את האינדקס בזיכרון, שומר מסמכים ומייצר מיפוי של מילה ומזהי מסמכים.  
פונקציות עיקריות:  
 • add\_document – מוסיף מסמך חדש ומעדכן את האינדקס.  
 • search\_word – מחזיר מזהי מסמכים שמכילים את המילה.  
 • get\_document – מחזיר את תוכן המסמך לפי מזהה.

2. QueryService

יוצר שאילתות חיפוש ומבצע חיתוך תוצאות (AND בין מילים).

פונקציה עיקרית:  
 • create\_query – מחפש מסמכים לכל מונח ומחזיר רק את ההצטלבות.

3. ResultService

אחראי על עיצוב תוצאות החיפוש.

פונקציה עיקרית:  
 • format\_results – מחזיר תוצאה מעוצבת לכל מסמך.

**Search Page**

4. search\_records: מנקה את השאילתה, עושה stemming, ושולח ל־QueryService.  
הQueryService מדרג את התוצאות לפי מספר הופעות המילים במסמך.

5. get\_suggestions: מחזירה מילים שמתחילות באותו prefix מהאינדקס.

6.common\_prefix\_len: מחזירה את אורך ההתאמה ההתחלתית בין שתי מילים – משמשת לדירוג הצעות.

**MQTT וחיישנים**

7. sensor\_dashboard\_loop:

לולאת רקע שמתבצעת כל 10 שניות:  
 • מאזינה ל־MQTT, ואם ההתחברות נכשלה או אין נתונים – מדמה ערכים.  
 • מציגה את הגרפים לפי הערכים אחרונים.

8. handle\_sensor\_value:

מעדכנת ערך חיישן ב Firebase:

latest, history, outliers.

קובעת אם הערך חורג מהטווח התקין ומעדכנת את הצבע.

9. simulate\_data:

מדמה ערכים ריאליסטיים לפי סוג החיישן כשאין MQTT פעיל.

10. create\_sensor\_card

יוצרת כרטיס גרפי עבור כל חיישן במסך הניהול הכולל:

כותרת עם שם החיישן ויחידות המדידה, גרף ערכים חיים המתעדכן בזמן אמת באמצעות רענון, כפתור Toggle שמאפשר למנהל לקבוע האם החיישן יוצג גם במסך העובדים.

הערך נשמר ב־Firebase תחת isShow, כך שהוא נשמר בין רענונים.

11. plot\_ sensor

מציגה גרף של ערכי חיישן:

ציר X מייצג את השעות (HH:MM:SS) של הדגימות.

צבע הגרף משתנה בהתאם לתקינות הערכים (ירוק – תקין, אדום – חריג).

מתבצע טעינה מחדש תוך שימוש ב־matplotlib.

12. refresh\_handler  
 מאפשר רענון ידני של הנתונים בלחיצת כפתור:

מפעיל את לולאת ה־MQTT לפרק זמן קצר (כ־6 שניות), מציג למשתמש את זמן הרענון האחרון.

טוען מחדש את הגרפים בהתאם לנתונים האחרונים שנקלטו.

13. on\_message  
  
 פונקציית Callback של MQTT:

מפענחת את תוכן ההודעה שהתקבלה.

מזהה את מיקום החיישן (indoor/outdoor).

שולחת את הערכים לפונקציה handle\_sensor\_value לצורך שמירה והצגה.

14. unpack\_values  
 מפענחת את רשימת הערכים של כל חיישן עבור תצוגת הגרף:

מפרידה בין טיימסטאמפים לבין ערכים.

מציגה רק את השעה לצורך קריאות טובה יותר בגרף.

**עיבוד נתוני Big Data**

10. dump\_history\_to\_csv:

שומר את ההיסטוריה של כל החיישנים בקובץ CSV מקומי.

11. run\_spark\_pipeline:

מריץ Apache Spark על הקובץ, מחשב ממוצעים לפי סוג חיישן.

מחזיר DataFrame עם אחוז מהממוצע ביחס לסף העליון התקין.

12. plot\_bigdata\_charts:

מצייר גרף עמודות שמראה איזה חיישנים במצב חריג.

**סטטיסטיקה וגרפים**

13. average\_by\_x\_values:

מחשב את הממוצע של כל מקטע בעל X ערכים, עבור Y מקטעים, בהינתן התאריכים והערכים של החיישנים.

14. calculate\_absolute\_changes:

מחשב את אחוז השינוי בכל מקטע מהממוצע הכללי של החיישן.

15. calculate\_normalized\_changes:

מחשב את אחוז השינוי של כל מקטע יחסית לערך הראשון בערכים.

16. plot\_diff\_heatmap:

מציג Heatmap של הבדלים מנורמלים בין שני סוגי חיישנים (למשל טמפ' ולחות).

**ניהול משימות :**

17.get\_next\_task\_id:

מחולל מזהה משימה חדש בפורמט T001, T002 וכו'.

סורק את מזהי המשימות הקיימים במסד הנתונים.

מוצא את המספר הפנוי הקטן ביותר ויוצר מזהה חדש.

refresh\_task\_table.18:

טוען את טבלת המשימות ומציג רק את המשימות התואמות לפי סינון (סטטוס, עדיפות, עובד).

כולל יצירת כפתור אישור (Confirm) למשימות שבוצעו.

מתרענן אוטומטית כאשר ערכי הסינון משתנים.

update\_task .19:

עדכון פרטי משימה קיימת לאחר עריכתה.

שולח את הערכים המעודכנים ל-Firebase.

מרענן את הטבלה כדי לשקף את העדכון.

delete\_task .20:

מאפשר למשתמש למחוק משימה קיימת לאחר אישור סופי. כולל מנגנון לווידוא מחיקה.

save\_task .21:

מוסיף משימה חדשה למערכת.

לוקח id למשימה חדשה מהפונקציה get\_next\_task\_id().

שומר למסד הנתונים את כלל הפרטים : assigned to, description, score, due date, start date and status שהוזנו בטופס למשימות.

22. get\_workers:

מייבאת את כל העובדים במערכת דרך ה-Firebase. מאפשר ייחוס משימות לעובדים קיימים במערכת.

**Store**

23. load\_store\_tab:

טוען את טאב החנות:  
 • מציג פריטים זמינים לפי הניקוד של המשתמש.  
 • מאפשר לרכוש פריטים עם נקודות.  
 • כולל דירוג עובדים לפי ניקוד (leaderboard).

24. build\_store\_table\_html:

בונה טבלת HTML של פריטי חנות – עם צבעים ואייקונים.

פריטים לא זמינים מוצגים באפור אוטומטית.

**התחברות (Login)**

25. login\_button\_clicked:

מאמת את פרטי המשתמש על פי המייל והסיסמה שנבחרו, כולל בדיקת תקינות בסיסית. אם המשתמש נמצא, הוא עובר למסך הראשי לפי סוג המשתמש (מנהל או עובד). נשמר מידע של המשתמש במשתנה גלובלי לצורך המשך עבודה בטאבים אחרים.

26. show\_tabs:

מגדיר את הממשק הגרפי של טאב של סך המסכים במערכת.

27. logout\_button\_clicked:

מנתק את המשתמש הנוכחי, מאפס את המידע השמור במשתנה הגלובלי ומחזיר למסך המשתמש הלא מחובר.

### **לוח בקרה (Dashboard)**

28. load\_dashboard\_tab:

מציג את לוח הבקרה הראשי לעובד או למנהל בהתאם להרשאות. כולל בתוכו סטטיסטיקות כלליות, ויזואליזציה של חיישנים, עדכוני משימות וסטטוס חנות.

29. create\_latest\_sensors:

יוצר תיבה גרפית עם ערכים עדכניים לכל החיישנים הפנימיים והחיצוניים. משמש גם בדשבורד וגם בטאבים אחרים להצגת מצב חיישנים בזמן אמת.

30. create\_outliers\_alert:

מציג התראות חכמות כאשר חיישנים חורגים מערכים תקינים.

מוודא שהמשתמש רואה חריגות בצורה בולטת בלוח הבקרה.

31. create\_manage\_sensors:

מאפשר למנהל לצפות, לנהל ולבצע פעולות אדמיניסטרטיביות על כל החיישנים (כיבוי, סימולציה, חיבור מחדש).

כולל טבלת חיישנים עם סטטוס עדכני ותפריטים אינטראקטיביים.

build\_prompt .32:

בונה תבנית Prompt הכוללת תיאור מלא של המערכת בהתאם לשאלה שהמשתמש שואל.

מבנה ה-Prompt כולל הסברים על כל הטאבים (Dashboard, Sensors, Tasks וכו')

מוסיף את שאלת המשתמש בסוף המחרוזת לצורך הפקת תשובה מותאמת להקשר.

createChatBot .33:

יוצר את ממשק העוזר הווירטואלי

כולל שדה קלט, כפתור שליחה, וכפתור ניקוי.

מציג את השאלה של המשתמש והתשובה מהמודל בגופן שונה ותחום.

משתמש בפונקציה build\_prompt כדי לבנות את הקלט למודל השפה.

משתמש בפונקציה to\_markdown להצגת התשובה בפורמט ברור.

1. **תיק למשתמש:**

**הסבר כללי על המערכת:**

מערכת Unicloud היא פלטפורמה מבוססת ענן המיועדת לניטור, ניהול והצגה של נתונים מחיישנים שונים כגון טמפרטורה, לחות, לחץ, מרחק ו-DLIGHT. המידע נאסף בזמן אמת דרך פרוטוקול MQTT ונשמר בענן, כאשר המערכת מושכת ומציגה את הנתונים למשתמשים בממשק אינטראקטיבי ונוח. המערכת כוללת ממשקים ייעודיים למנהלים ולעובדים, ומאפשרת למנהל לנהל ולעקוב אחר משימות העובדים שלו, בעוד שהעובדים יכולים לתעד את התקדמותם ולעדכן את סטטוס המשימות שלהם. בנוסף, המערכת משלבת אלמנט משחקי שבו העובדים צוברים נקודות על השלמת משימות ויכולים לממש אותן בחנות הטבות פנימית, מה שתורם להגברת המוטיבציה והמעורבות בארגון. באמצעות מסכים ייעודיים להצגת סטטיסטיקות, חיפוש מידע, ניהול חיישנים וניתוח נתוני ביג דאטה, Unicloud מספקת פתרון מקיף ויעיל לניהול תהליכים בארגון חכם ומתקדם.

**פירוט מסכים:**

1. **מסך Login/Logout:**

* מאפשר התחברות למערכת באמצעות אימייל או שם משתמש וסיסמה.
* תומך בשני סוגי משתמשים: מנהל ועובד.
* כולל אפשרות להתנתק (Logout) ומעבר חזרה למסך ה- Login.

1. **מסך ראשי (Dashboard):**

המסך כולל 3 אזורים מרכזיים:

* אזור Assistant Chat Bot המאפשר מענה לשאלות על המערכת על ידי שימוש ב API של Gemini.
* אזור המציג ערכים אחרונים מכל החיישנים הפנימיים והחיצוניים המוצגים בזמן אמת.
* אזור הצגת התראות חריגות שהתקבלו מנתוני החיישנים (ערכים שנמצאים מחוץ לטווחים המוגדרים).

1. **מסך חיפוש:**

כולל שורת חיפוש המאפשרת למשתמש לחפש באינדקס שנבנה מאתר [mqtt.org](http://mqtt.org) , ועוזר במציאת תכנים רלוונטיים ודוקיומנטציה של חיישנים הקשורים לפרוטוקול MQTT. תוך כדי הקלדת הערך המבוקש, יוצגו למשתמש הצעות לערכי חיפוש דומים.

1. **מסך משימות:**

* עבור משתמש מסוג מנהל - מסך Task Manager: תוצג טבלת כל המשימות אשר ניתנו לעובדים, עם אפשרויות פילטור (לפי שם, סטטוס או עדיפות ביצוע) ועם היכולת לאשר באופן סופי משימות שהסטטוס שלהן הוא Done ובכך לספק את הניקוד שלה לעובד. בנוסף יכול המנהל ליצור משימה חדשה, להקצות לה מאפיינים כמו: נושא, ניקוד, תאריכי התחלה וסיום, תיאור קצר ולבחור עובד לו המשימה משוייכת, וגם לערוך או למחוק משימות קיימות.
* עבור משתמש מסוג עובד - מסך My Tasks: תוצג טבלת כל המשימות אשר הוקצו לו בלבד עם היכולות לעדכן את הסטטוס שלהן מ- To Do ל- In Progress או ל- Done.

1. **מסך סטטיסטיקות:**

הצגת נתונים גרפיים וסטטיסטיים של החיישנים בהתאם לטווח הערכים הנבחר על ידי המשתמש (תאריך התחלה ותאריך סיום). ניתן לבחור באילו חיישנים להציג. המידע המוצג יכלול בין היתר:

ממוצע כולל של הערכים, אחוז השינוי מהממוצע הראשון, הבדלים בין ערכי לחץ וטמפרטורה מנורמלים (indoor), הבדלים בין טמפרטורה ולחות (indoor) ואת האפשרות לייצוא את הנתונים הללו לקובץ אקסל מסודר.

1. **מסך Overall Sensors:**

הצגת נתוני חיישנים פנימיים וחיצוניים כאשר חיישנים פנימיים כוללים מידע על: טמפרטורה, לחות, לחץ ומרחק, וחיישנים חיצוניים כוללים מידע על: טמפרטורה, לחות ו DLIGHT.

כל חיישן מוצג עם יחידות מידע רלוונטיות המוגדרות מראש. צבע גרף החיישן משתנה בהתאם לקליטת ערכים חריגים: אם התקבלו ערכים מחוץ לטווח שהוגדר - אדום, אחרת - ירוק.

* מנהל יוכל לבחור אילו חיישנים יוצגו לעובדים בממשק שלהם, מסך כל החיישנים הקיימים במערכת.
* ניתן לרענן את נתוני החיישנים באמצעות כפתור "Refresh".
* מבוסס על פרוטוקול MQTT.

1. **מסך Big Data:**

מציג ניתוח גרפי מתקדם של מידע שנאסף מהחיישנים לצורך הפקת תובנות מעמיקות יותר.

1. **מסך My Sensors:**

מציג את החיישנים שהמנהל בחר לשתף את העובד. הגרפים המוצגים דומים לגרפים המוצגים במסך Overall Sensors. במקרה של רענון במסך החיישנים של המנהל, יעודכנו החיישנים אצל העובד.

1. **מסך My Tasks:**

הצגת המשימות של העובד המחובר. ניתן לסנן תוצאות על פי דחיפות ושלב ביצוע. כמו כן, העובד יכול לעדכן את המשימה בשלב ביצועה, אם היא לביצוע, בביצוע או בוצעה.

1. **מסך Store:**

המסך מחולק לשני חלקים עיקריים:

חנות: תכלול טבלת הטבות כאשר לכל הטבה: שם, תיאור קצר ועלות. ניתן לרכוש הטבות בשימוש נקודות המתקבלות כתוצאה מהשלמת משימות. לעובד המחובר במערכת יוצג באופן אוטומטי הניקוד הקיים שלו, ולפיכך יודגשו רק ההטבות שהוא יכול לממש, ורק מהן ניתן יהיה לבחור הטבה למימוש.

לוח ניקוד (Leaderboard): יכלול טבלה המציגה דירוג של כל עובדי החברה לפי ניקוד מצטבר.

**מעבר בין מסכים:** תחילה יוצגו שני Tabs: לוגין וחיפוש. לאחר ביצוע לוגין ייפתחו שאר ה- Tabs הרלוונטיים לפי ההרשאות של המשתמש והניווט ביניהן ייתבצע על ידי לחיצה על ה- Tab הרצוי כאשר לכל Tab כותרת מתאימה.

**טעויות אפשריות:**

1. הזנת פרטי התחברות שגויים תוביל לכשל בהתחברות למערכת.
2. אי עדכון סטטוס משימות על ידי עובדים עלולה לגרום לכך שהמנהל לא יוכל לעקוב בצורה מדויקת אחר הביצועים בשטח.
3. צפייה ושימוש בנתוני חיישנים לא מעודכנים, במידה והמשתמש לא לחץ על כפתור Refresh.
4. מנהל שלא מקפיד על סימון חיישנים כמיועדים להצגה לעובדיו, עלול לגרום לכך שחיישנים רלוונטיים לא יהיו נגישים לעובדים ויעכבו את מהלך העבודה.
5. **כיצד הבהרתם למשתמשים את האלגוריתמים והנתונים הנאספים בקוד שלכם?**

בהתייחס לדרישת השקיפות האלגוריתמית, במערכת Unicloud הקפדנו להבהיר למשתמשים כיצד מתקבלות ההחלטות, כיצד מחושבים הנתונים ומהם הקריטריונים לפעולות אוטומטיות:

במסך ה-Dashboard: המשתמשים רואים בכל חיישן את ערכו הנוכחי יחד עם יחידת המדידה המתאימה. צבע הרקע של כל כרטיס משתנה באופן מיידי לירוק או אדום על סמך בדיקת חריגה מאלגוריתם רלוונטי (המוגדר בטווחים קבועים מראש). אם למשתמש יש שאלה על ערכי הרצפה והתקרה או ההיגיון שמאחורי ההתראות – הוא יכול לשאול את צ'אט הבוט המובנה ולקבל הסבר מדויק, ובכך נוצרת שקיפות מלאה לפעולת המערכת בזמן אמת.

במסך החיפוש: לאחר כל שאילתה, מוצגות לכל תוצאה גם כמות ההופעות המדויקת של מילות החיפוש בדף, וגם ציון PageRank המשוקלל של הדף, מה שמבהיר למשתמש בצורה ישירה כיצד מתבצע הדירוג ומה תרם למיקום של כל תוצאה ברשימה.

במסך הסטטיסטיקה: המשתמשים מקבלים גרפים צבעוניים המציגים חישובים על החיישנים הנבחרים: ממוצעים, אחוזי שינוי ושונות מנתוני ייחוס. הכפתור “Export to Excel” מאפשר הורדה של הנתונים עצמם (כולל כל הערכים והחישובים), כך שהמשתמש יכול לבדוק ולאמת בעצמו את נוסחאות האלגוריתם ולוודא את אמינות החישוב.

במסך החיישנים: כל גרף כולל כותרת ברורה המציינת את סוג החיישן ויחידת המדידה, והגרף עצמו מתעדכן בצבע אדום או ירוק לפי מצב החריגה מהממוצע. גם כאן המשתמש רואה בצורה ישירה את הקשר בין ערכים לבין החריגה, ללא צורך להניח או לשער מהו הקריטריון.

באמצעות עקרונות אלה, Unicloud מקפידה על שקיפות מלאה של האלגוריתמים והחלטות המערכת, תוך הצגת מדדים כמותיים, הסברים נגישים, וצמצום אפקט "קופסה שחורה" עבור המשתמשים.