

**המחלקה להנדסת תוכנה - מבוא למחשוב ענן**

**פרויקט**

מועד הצגה: שבוע 11 - בזמן ההרצאה

מועד הגשה: **10.7.25**

**Turtles team**

עינב בן שושן – 315064881

יובל לרפלד – 318186897

מתן עמרן - 316326313

עמיר אזולאי – 206695488

חן צפיר – 319005104

קישור לתיקיית הגיט:

[https://github.com/einavbs1/Final\\_Project\\_Corelia](https://github.com/einavbs1/Final_Project_Corelia)

קישור למחברת הColab:

 [Final\\_Project\\_Turtles.ipynb](#)

## 1. מהות המוצר ומרכיביו

### 1.1. מודולים ופיצ'רים מעניינים:

- 1.1.1. פיצ'ר – מנוע חיפוש אשר מתוך שאילתת המשתמש יודע להתאים את התוצאות שהמשתמש יצרה (כמובן בעזרת פונקציות NLP).
- 1.1.2. מודל – NLP ששומש בהסרת stop word, פונקצית priority stem לדפים השונים.
- 1.1.3. פיצ'ר – widgets של python על מנת ליצור את הUI.
- 1.1.4. פיצ'ר – תצוגת הגרפים בעזרת matplotlib.
- 1.1.5. פיצ'ר – משחקיות gamification הוספת משחקיות למשתמשים שיוכלו להשלים משימות שמתעדכנות בזמן אמת.
- 1.1.6. מודל – Chatbot (Gemini API) יצרנו צאט בוט שאומן עם שאלות מראש כך שניתן לחקור אותו ולהשתמש בו על מנת לחקור על האתר שלנו ונושא הפרויקט.
- 1.1.7. פיצ'ר – security אבטחה והצפנה של הסיסמאות השמורות בDB על מנת שלא יוכלו לגשת לסיסמא לראות אותה.
- 1.1.8. פיצ'ר – לולאת מעקב אחרי שינוי נתונים בDB כך שהמידע יתעדכן בזמן אמת למנהל הADMIN לגבי משתמשים שמתחברים, מתנתקים וכן חיפושים אחרונים שמבצעים.

### 1.2. microservices שנעשה בהם שימוש:

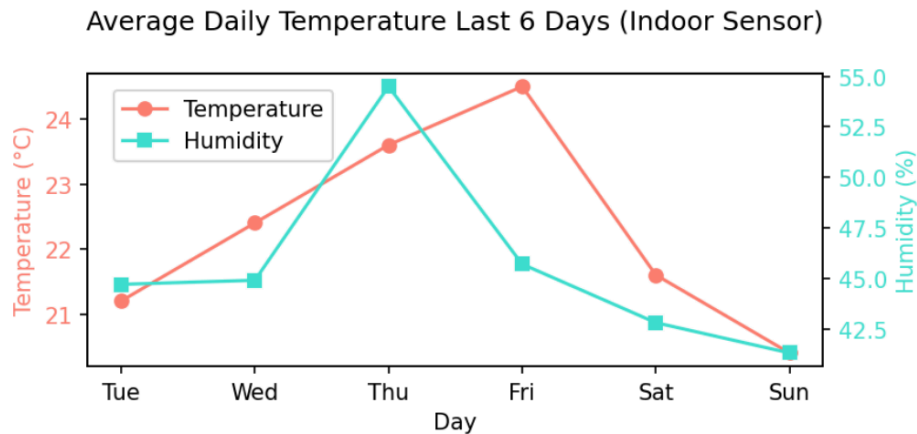
- 1.2.1. שירות אינדקס - שירות שיוצר אינדקס לאתר מסוים. בנוסף ניתן לבחור עומק, בסיום השימוש בשירות האינדקס יופיע בDB
  - 1.2.2. שירות שאילתא - בשימוש בשירות ניתן ליצור שאילתא שנעזרת באינדקס על מנת לקבל האינדקסים הרלוונטיים למילה.
- היתרון נובע מהעובדה שהשירותים עצמאיים יחסית, שכל אחד אחראי על פונקציונליות אחת ברורה. ומאפשר גמישות לשינויים ושיפורים ובנוסף מאפשר להריץ רק את השירותים עצמם במקרה הצורך.

### 1.3. KPI מרכזיים:

- בטאב Gamification אנו מציגים מדדים כמותיים של ביצועים, כמו:
  - Daily Score Table
  - מציג ניקוד יומי לכל משתמש (מבוסס על משימות שהושלמו באותו יום).
  - Monthly Score Table
  - מציג ניקוד חודשי מצטבר לכל משתמש (מבוסס על משימות שהושלמו במהלך החודש).
  - משימות מדורגות לפי קושי
  - משימות מדורגות לפי רמות קושי (easy, medium, hard) והמשתמש מקבל ניקוד בהתאם.
  - עבור משימה שהיא easy המשתמש מקבל 10 נקודות
  - עבור משימה שהיא medium המשתמש מקבל 30 נקודות
  - עבור משימה שהיא hard המשתמש מקבל 50 נקודות
- הניקוד המצטבר לפי משתמש מתעדכן בדאטה בייס (פיירבייס) ומוצג בטבלאות. המטרה היא שיפור ביצועים של משתמשים (עובדים) באמצעות משימות מדורגות. אנו מודדים, מציגים ומשווים בין עובדים על בסיס ההתקדמות שלהם (ניקוד).

#### 1.4. ניתוח נתוני עתק

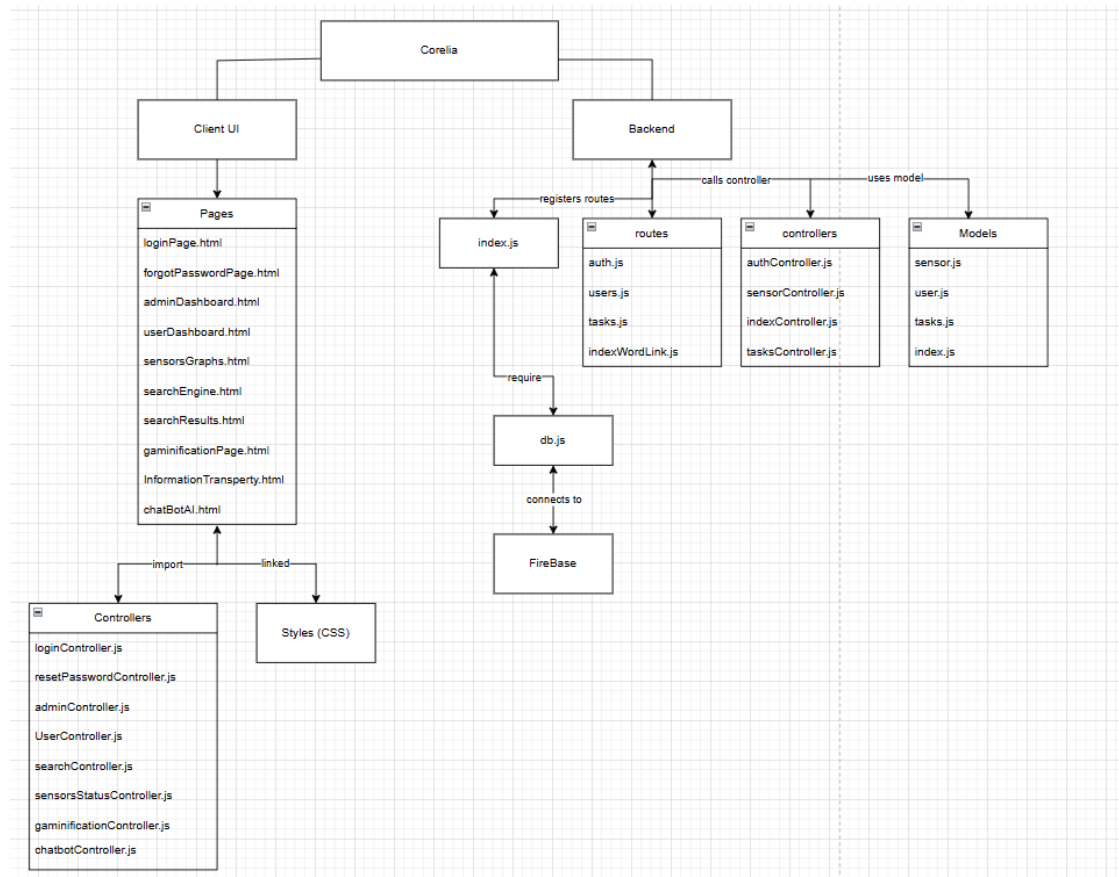
ביצענו את הניתוח על נתוני החיישן הפנימי Indoor, על מנת לטעון את הנתונים הרלוונטים השתמשנו ב-Spark, ולאחר מכן ייצרנו DataFrame עליו ביצענו פעולות groupBy לחישוב הממוצע היומי, לאחר מכן השתמשנו בנתונים ליצירת גרף המתאר את מגמות הטמפרטורה והלחות ב-6 הימים האחרונים.



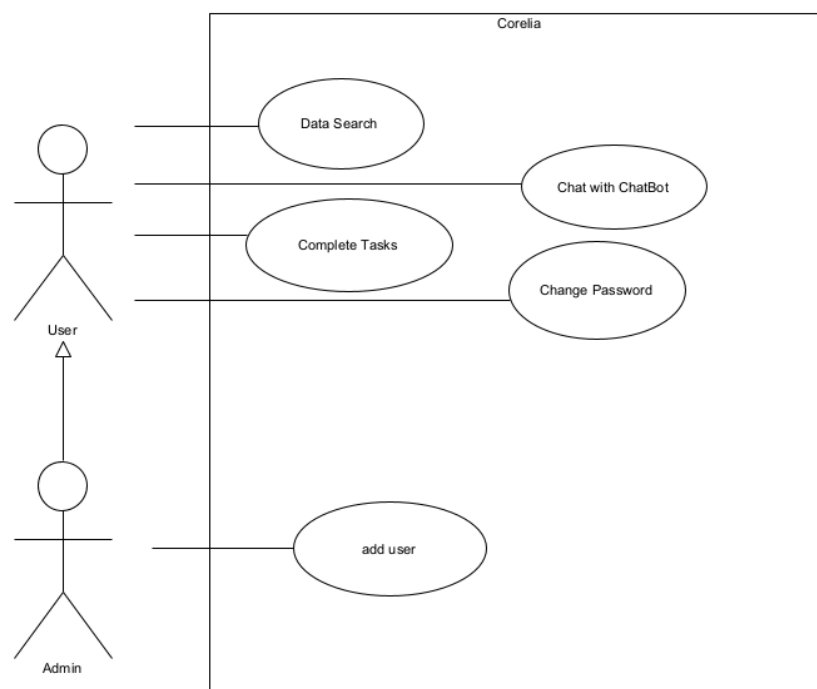
ניתוח הגרף -

- **שלישי-רביעי** : הטמפרטורה יחסית נמוכה עם כ- $21^{\circ}\text{C}$  ולחות סביב 44%–45%. למחרת עולה הטמפרטורה בכמה מעלות, והלחות נשארת בערך אותה רמה.
  - **חמישי** : הלחות קופצת ל-55% והטמפרטורה עוקבת אחריה ועולה מעל  $23^{\circ}\text{C}$ .
  - **שישי-שבת** : הלחות יורדת בצורה חדה ל-46% ביום שישי ול-42% בשבת, והטמפרטורה עולה מעט ואז יורדת חזרה ל- $21.5^{\circ}\text{C}$ . דבר שיכול לרמז על הפעלת קירור.
  - **ראשון** : הלחות נמוכה במיוחד 41% והטמפרטורה יורדת עוד קצת ל- $20^{\circ}\text{C}$  – יכול להיות בגלל קירור/אורור חזק או שהייה מועטה בחדר.
- לסיכום : בתחילת השבוע הטמפרטורה והלחות היו יציבים, עם עלייה קלה בטמפרטורה בעוד הלחות נשארה יציבה. באמצע השבוע הלחות קפצה והטמפרטורה טיפסה יחד איתה, כנראה בשל שינוי בתנאי הסביבה או ריבוי פעילות באזור. בסוף השבוע חל שינוי הפוך, כלומר הלחות והטמפרטורות ירדו, מה שמעיד על הפעלת מזגן או אורור מוגבר. ניתן להניח שהשינויים באקלים הפנימיים משקפים יותר תפעול אנושי ותפעול יזום מאשר תנודות טבעיות אקראיות.

2. ארכיטקטורת המערכת  
2.1. תרשים ומאפיינים מרכזיים:



2.2. Use case מעודן של פונקציונליות המערכת:



### 2.3. דרישות פונקציונליות:

- 2.3.1. ביצוע חיפוש מידע.
- 2.3.2. השלמת משימות.
- 2.3.3. ביצוע שיחה עם צ'אט-בוט.
- 2.3.4. שינוי סיסמא.
- 2.3.5. הוספת משתמש.

### 3. דרישות לא פונקציונליות:

- 3.1. המערכת תחבר את המשתמש תוך 3 שניות בממוצע.
- 3.2. המערכת תציג את תוצאות החיפוש תוך 5 שניות בממוצע.
- 3.3. כל סיסמאות המשתמשים יאוחסנו באופן מוצפן.
- 3.4. הקוד יורץ ללא שום תלות חיצונית.
- 3.5. המערכת תבדוק ותשווה סיסמאות בהשוואה לערך המוצפן בלבד ולא תשמור סיסמאות לא מוצפנות.

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

נימוק	האם בוצע שינוי במערכת	אלמנטים שיש לשפר במערכת
בשני המשובים לא צוינו בעיות /הצעות שיפור, לכן אין שינויים נדרשים. המערכת תואמת את ציפיות המשתמשים.	-----	-----

### חישוב שאלוני sus :

1. הייתי רוצה להשתמש במערכת זו לעיתים קרובות.
2. מצאתי כי המערכת מסובכת ללא סיבה.
3. חשבתי שהמערכת קלה לשימוש.
4. אצדקק לתמיכת איש טכני כדי שאוכל להשתמש במערכת זו.
5. מצאתי כי הפונקציות השונות של המערכת היו מתואמות היטב.
6. חשבתי כי היה יותר מידי חוסר עקביות במערכת זו.
7. לדעתי רוב האנשים יהיו מסוגלים ללמוד להשתמש במערכת זו בקלות.
8. מצאתי כי המערכת מאוד מסורבלת לשימוש.
9. חשתי ביטחון רב כאשר השתמשתי במערכת זו.
10. עלי ללמוד הרבה דברים לפני שאוכל להשתמש במערכת זו.

משוב	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	sus_score
משוב 1	5	1	5	1	5	1	5	1	5	1	100
משוב 2	5	1	5	1	5	1	5	1	5	1	100

ציון SUS יהיו 100 עקב המשובים שקיבלנו.

## 5. פירוט שקיפות אלגוריתמית:

במערכת הוקצה טאב ייעודי המבהיר למשתמשים את אופן פעולתם של האלגוריתמים והנתונים שנאספים. בטאב זה מוצגים הסברים על כל רכיב במערכת, כולל:

- אופן חישוב הניקוד
- שיטות אינדוקס המידע
- אופן הצגת הגרפים
- סוגי הנתונים הנשמרים במערכת

בנוסף, המערכת מבהירה כי לא נשמרים נתונים אישיים וכל עיבוד המידע נעשה באופן שקוף ומשמש לצרכים לימודיים בלבד.

## 6. אתגרים אשר עלו במהלך העבודה, וכיצד התמודדתם איתם:

6.1. **קושי ביצירת ויזואליזציה שתרוץ בקולאב:** בהתחלה ניסינו להשתמש בפלאסק, אך לאחר שיחה עם נעמי במהלך קבלת המשימות, גילינו שיש דרישה לא להריץ את היישום בצורה זו. היא המליצה לנו על גרדיו, אך גילינו שגם כאן יש קשיים. בסופו של דבר, החלטנו להשתמש בטאבים כפי שלמדנו בתרגול, והפתרון הזה עובד מצוין.

6.2. **קושי בהבאת מידע מהחיישנים:** במהלך העבודה נתקלנו בקשיים בהבאת המידע מהחיישנים, מה שגרם למבנה הקוד להשתנות באופן תדיר. החיישנים קרסו לעיתים קרובות עד שנכנסנו למצב שבו לא ניתן היה לחלץ מידע דרך הברוקר. לפיכך, החלטנו להפיק פייק דאטה המבוסס על הנתונים הראשוניים שהצלחנו לחלץ מהחיישנים, והוספנו נתונים נוספים שיצרנו בעצמנו על מנת להציג גרפים סטטיסטיים.

6.3. **קושי בביצוע איפוס אוטומטי לגיימיפיקציה:** היה אתגר במימוש האיפוס האוטומטי של המשימות מדי יום ושמירת ההיסטוריה. לדוגמה, משימות שהושלמו אתמול לא נמחקות או נשמרות פעמיים. מצאנו פתרון בכתיבת פונקציה שמבצעת איפוס יומי, מעבירה את המשימות שהושלמו לארכיון לפי תאריך ומורידה אותן מה־dropdown בעת ביצוע משימה כלשהי.

6.4. **אתגר ביצירת ה-API לצ'אט:** במהלך יצירת ה-API לצ'אט נתקלנו בקשיים, במיוחד במימוש API של ג'ימיני. למרות שהצלחנו ליצור את ה-API כפי שלמדנו בתרגול 8, כל פעם נתקלנו בבעיות בקבלת המידע. לאחר שימוש בצ'אט, הצלחנו להבין את הדרך הנכונה ליצירת API תקין.

6.5. **קושי במימוש האינדקס:** חוינו קושי בניהול האינדקס, שהיה מאוחסן ב-DB ונדרש להיות מנוהל בשני טיפוסים שונים – מצד אחד לפי הצורך של Firebase, ומצד שני לפי הצורך של התוכנית שרצה בפיתוח. זה דרש עבודה רבה על מנת לסנכרן את המבנים השונים.

6.6. **אתגר בניהול קוד ארוך:** ניהול קוד ארוך היה אתגר מתמשך. מדובר בקושי גנרי, אך הוא לווה אותנו בכל הגשה והיה משמעותי בתהליך העבודה.

6.7. **קושי בעבודה עם נתוני פייק:** במקום למשוך את הנתונים ישירות מ-Firebase, עבדנו עם נתונים פיקטיביים מבוססי נתונים ראשוניים שהצלחנו לחלץ מהחיישנים. כדי לשלב אותם, היינו

צריכים לקרוא לפונקציות בתוך המחברת, מה שגרם ליותר מורכבות ואריכות בקוד.

6.8. **קושי בתקשורת עקב היעדרות חברי הצוות:** שלושה מתוך חמישה חברי הצוות היו במילואים, ולכן כל אחד היה אחראי על משימותיו. מדי פעם הייתה בעיית זמינות, דבר שהוביל לנתקים בתקשורת, אך כמובן זה היה מובן. לפעמים לא היה ברור מראש מי מבצע כל תפקיד, מה שגרם לכך שחלק מהמשימות נשארו פתוחות עד לרגע האחרון.

6.9. **קושי בהתחברות למערכת ובניהול הרשאות:** אתגר בהטמעה של אלגוריתם התחברות והתנתקות למערכת, כך שכל משתמש יקבל גישה בהתאם להרשאותיו. וסיווג של כל משתמש לדוגמא אדמין יופנה למסך אדמין ומשתמש רגיל למסך משתמש.

6.10. **הבנת ושיפור הקוד של חבר בקבוצה:** חלק מהקושי היה בהבנת הקוד שנכתב על ידי אחד מחברי הצוות, שכולל פונקציות מורכבות שלא תמיד היו ברורות. לאחר שבחנו את הקוד לעומק, הצלחנו להבין את הכוונה מאחוריו, וביצענו שיפורים ושינויים שהפכו את הפונקציות לאפקטיביות יותר, תוך התאמתם לצרכים החדשים של המערכת.

תיק תחזוקה:

תיק למתכנת – Corelia

נבנה ע"י: turtles group

גירסה: 1.3

תאריך שחרור הגירסה: 24/06/2025

חברי הצוות:

עינב בן שושן – 315064881

יובל לרפלד – 318186897

מתן עמרן – 316326313

עמיר אזולאי – 206695488

חן צפיר – 319005104

♦ קבצים עיקריים:

• מחברת הפרוייקט – HW3\_Turtles\_Corelia

♦ ספריות עיקריות:

- **FireBase** - חיבור למסד נתונים בזמן אמת.
- **lpywidgets** - יצירת ממשק משתמש אינטראקטיבי במחברת.
- **Sendgrid** - לשליחת מיילים (כולל איפוס סיסמה).
- **google.generativeai** - שילוב עם Gemini לצ'אט בוט.
- **matplotlib, pandas, numpy** - לניתוח והצגת נתונים.

♦ הרשאות משתמשים:

• -User

o איפוס סיסמא.

o אפשרות לשחק בטאב המשחקיות וצפייה בניקוד.

## • -Admin

- o יצירת משתמש חדש למערכת.
- o צפייה בכל נתוני המשתמשים.
- o צפייה בכל המילים שנמצאים בDB.

## • Admin יורש הרשאות User

♦ פונקציות מרכזיות:

שם הפונקציה	תיאור הפונקציה
add_user()	הוספת משתמש חדש ל-Firebase.
add_index_to_firebase()	שמירת אינדקס מילים/נתונים ב-Firebase
create_temp_humidity_chart()	יצירת גרף טמפרטורה ולחות לאורך זמן.
create_daylight_hours_chart()	יצירת גרף שעות אור מול פעילות המשתמש.
calculateDailyScore()	חישוב ניקוד יומי למשתמש.
DisplayDailyRanks()	הצגת דירוג יומי לכלל המשתמשים.
complete_task()	סימון משימה כהושלמה ועדכון המערכת.
monitor_loop()	ניטור משתמשים מחוברים בזמן אמת.
apply_stemming()	עיבוד טקסט לקיצוץ מילים לשורש (Stemming).
create_index_inspector()	יצירת ממשק לצפייה באינדקס המילים.
password_reset_email()	שליחת מייל איפוס סיסמה למשתמש.
update_tabs()	עדכון ממשק הטאבים בהתאם להרשאות המשתמש
fetch_users_from_firebase()	שליפת רשימת משתמשים מה DB.
handle_login()	ניהול תהליך התחברות משתמש (תוך הסתכלות על הרשאות).
handle_logout()	ניהול תהליך התנתקות משתמש והסתרת כל הטאבים.
on_click_search()	הפעלת חיפוש לפי טקסט שהוזן ע"י המשתמש.
saveUserSearchQuery()	שמירת שאילתת החיפוש של המשתמש ב-Firebase (למעקב אחרי חיפושים אחרונים).
on_click_lucky()	חיפוש "I'm Feeling Lucky": פתיחת התוצאה המדורגת הראשונה.
prioritize_pages()	דירוג העמודים בחיפוש לפי רמת רלוונטיות.



## ♦ תבניות מיוחדות:

- **ממשק משתמש אינטראקטיבי עם widgets -**
  - שימוש ב- widgets.Tab () לבניית טאבים נוחים למעבר בין הדפים השונים שיש למערכת להציע.
  - שימוש ב- widgets.HTML () ליצירת טבלאות ודוחות.
  - פונקציות אשר: מסתירות את כל הטאבים במקרה שלא מחוברים, מציגות את הטאבים במקרה של חיבור וסיווג על פי משתמש שמחובר.
- **טבלת ניקוד (LeaderBoard) -**
  - תבנית להצגת דירוג יומי / חודשי:
    - DisplayDailyRanks ()
    - DisplayMonthlyRanks ()
- **גרפים סטטיסטיים עם Matplotlib -**
  - גרף טמפרטורה ולחות לאורך זמן - create\_temp\_humidity\_chart ()
  - גרף שעות אור - create\_daylight\_hours\_chart ()
- **כפתור "Feeling Lucky" -**
  - כפתור ייעודי שמוביל אוטומטית לתוצאה עם הדירוג הגבוה ביותר - on\_click\_lucky ()
- **דירוג עמודים בתוצאות חיפוש (Prioritization) -**
  - מנגנון פנימי לקביעת סדר התוצאות בחיפוש - prioritize\_pages ()
- **שליחת מייל רספונסיבי לאיפוס סיסמה -**
  - תבנית מייל מעוצבת שנשלחת אוטומטית בעת איפוס סיסמה שמשתמשת בפונקציה שמייצרת סיסמא רנדומלית (כמובן לאחר שבודקת שפרטי המשתמש נכונים ואומתו) - password\_reset\_email ()
- **Real-time User Monitoring -**
  - לולאת רקע (thread) לניטור משתמשים מחוברים ולעדכון טבלת האדמין - monitor\_loop\_ ()
- **מנוע חיפוש חכם עם Stemming -**
  - עיבוד מילים לשורש (stemming) לשיפור תוצאות החיפוש –
    - apply\_stemming ()
    - get\_search\_results ()

## ♦ קטעי קוד מעניינים:

## • שליחת סיסמה חדשה במייל-

```
def send_password_reset_email(to_email, new_password):
    global SENDGRID_API_KEY, forgotPage_error_message
    try:
        sg = sendgrid.SendGridAPIClient(api_key=SENDGRID_API_KEY)
        message = Mail(
            from_email = "einavbsl@gmail.com",
            to_emails=to_email,
            subject='Reset Your Password',
            plain_text_content=f'''
                Hello,

                We received a request to reset your password.

                This is your new password:
                {new_password}

                Best regards,
                Corelia Turtles Group
            ''')

        response = sg.send(message)
        forgotPage_error_message.value = "An email has been sent to your email address
with the new password."+"\\n Please check your inbox and your SPAM folder."
    except Exception as e:
        forgotPage_error_message.value = "Failed to send email. Please try again later."
```

## הצגת הטאבים הרלוונטיים:

```
def resetTabs():
    global tabs, currUserisAdmin
    currPage = tabs.selected_index if hasattr(tabs, 'selected_index') else 0
    tabs.children=[tab0_content,tab1_content,tab2_content,tab3_content, tab5_content,
tab6_content]
    if currUserisAdmin == None :
        tabs.set_title(0, 'Login Page')
    elif currUserisAdmin:
        tabs.set_title(0, 'Admin Dashboard')
    else:
        tabs.set_title(0, 'User Dashboard')
    tabs.set_title(1, 'Graphs')
    tabs.set_title(2, 'Gamification')
    tabs.set_title(3, 'Search')
    tabs.set_title(4, 'Informations Transportation')
    tabs.set_title(5, 'Chatbot Assistant')
    tabs.selected_index = currPage
```

7. תיק למשתמש:

**Corelia – תיק למשתמש**

**נבנה ע"י:** turtles group.

**גירסה:** 1.3

**תאריך שחרור הגירסה:** 24/06/2025

**חברי הצוות:**

עינב בן שושן – 315064881

יובל לרפלד – 318186897

מתן עמרן – 316326313

עמיר אזולאי – 206695488

חן צפיר – 319005104

**♦ הסבר כללי על המערכת:**

מערכת **Corelia** היא מערכת אינטראקטיבית לניהול ובקרה על חיישנים במפעל, המיועדת הן לעובדים והן למנהלים.

המערכת מאפשרת ניהול יומיומי של פעילות העובדים, מעקב אחר השימוש במידע מהחיישנים, ומתן כלים לשיפור הלמידה והמעורבות בעזרת מרכיבי משחקיות (Gamification). במערכת משולב מנגנון **תחרויות יומיות וחודשיות** בין העובדים, שמטרתו לעודד את העובדים להשלים משימות יומיות ולצבור ניקוד. העובדים המתמידים ביותר יתברגו בטבלת **הדירוג** ויתמודדו על תואר עובד החודש.

למנהלים ניתנת אפשרות:

- לנהל את רשימת העובדים
- להוסיף ולהסיר עובדים
- לעקוב בזמן אמת אחר מצב ההתחברות של כל עובד
- לצפות בחיפושים האחרונים שבוצעו ע"י העובדים
- לנתח את השימוש במערכת בעזרת לוחות בקרה ייעודיים

המערכת כוללת גם **צ'אט בוט לימודי** המשלב מידע אודות עולם המכשור והענף. העובדים יכולים לפנות לצ'אט בוט בכל שלב ולהעמיק את הידע שלהם בנושאים שונים, כחלק ממדיניות ההעשרה והלמידה של הארגון.

במהלך השימוש במערכת, כל עובד יכול לעבור בקלות בין הטאבים והתכנים השונים המותאמים להרשאותיו — כולל:

- משימות יומיות
- דירוגים
- חיפוש מידע
- אינדקס מילים
- צ'אט בוט לימודי

המערכת בנויה כך שתספק **חוויית משתמש פשוטה ונוחה** גם לעובדים חדשים במערכת, עם דגש על חווית שימוש ידידותית ותמיכה מלאה בעברית.

#### ♦ מעברים בין מסכים:

- **מסך ההתחברות –**
  - מסך איפוס סיסמא.
  - הצגת כל הטאבים (כמנהל).
  - הצגת כל הטאבים (כמשתמש רגיל).
- **מסך ייעודי למשתמש –**
  - יציאה לדף ההתחברות.
- **מסך איפוס סיסמא –**
  - יציאה לדף ההתחברות.
- **כפתור “Feeling Lucky” במנוע החיפוש –**
  - פתיחת התוצאה המתאימה ביותר בכרטיסיה חדשה.
- **כפתור “Search” במנוע החיפוש –**
  - מסך תוצאות החיפוש.

**כל שאר המעברים בין הדפים מתבצעות דרך הכפתורים הפיזיים בממשק.**

♦ פירוט מסכי האתר:

שם המסך	תיאור מילולי קצר
Login	במסך זה ניתן להתחבר למערכת באמצעות שם משתמש וסיסמה (עובד רגיל או מנהל).
Forgot my password	במסך זה ניתן לאפס סיסמה. לאחר מילוי הפרטים, תישלח סיסמה חדשה למייל.
Admin Dashboard	מסך ייעודי למנהלים. מאפשר לבצע את כל פעולות הניהול (ניהול עובדים, ניטור, צפייה בחיפוש וכו').
User Dashboard	מסך אישי לעובד רגיל. מאפשר לבצע פעולות בסיסיות, לצפות במשימות, ניקוד וחיפוש.
Graphs	מסך המציג גרפים חשובים לעובד: תקינות סביבת העבודה, מידע עדכני מהחיישנים, גרפים היסטוריים (ימים/שעות אחרונות).
Gamification	מסך המשחקיות — כל עובד (או מנהל) יכול להשתתף בתחרות עובד החודש, לבצע משימות יומיות ולצבור ניקוד.
Search	מסך חיפוש: מאפשר חיפוש מידע על תקלות או נושאים שונים במערכת. כולל: <ol style="list-style-type: none"> <li>1. כפתור חיפוש רגיל — להצגת דף תוצאות*.</li> <li>2. כפתור I'm Feeling Lucky — מעבר ישיר לתוצאה הראשונה בדף חדש (מקל על חווית החיפוש ומקצר אותה).</li> </ol>
Results*	מסך הצגת תוצאות החיפוש, ממוינות לפי רלוונטיות לשאלתה.
Information Transparency	מסך מידע — מסביר על כל דף באתר, ומשקף למשתמש את הנתונים השונים.
Chatbot Assistant	צ'אט בוט אינטראקטיבי — מאפשר לשאול שאלות ולחקור נושאים מתחום פלטפורמות המחשוב בענף.

♦ פירוט שקיפות אלגוריתמית:

במערכת הוקצה **טאב ייעודי** המבהיר למשתמשים את אופן פעולתם של האלגוריתמים והנתונים שנאספים. בטאב זה מוצגים הסברים על כל רכיב במערכת, כולל:

- אופן חישוב הניקוד
- שיטות אינדוקס המידע
- אופן הצגת הגרפים
- סוגי הנתונים הנשמרים במערכת

בנוסף, המערכת מבהירה כי **לא נשמרים נתונים אישיים** וכל עיבוד המידע נעשה באופן שקוף ומשמש לצרכים לימודיים בלבד.

♦ אפשרויות שגיאה ופתרונות:

שם המסך + שגיאות	פתרון השגיאה
<b>Login:</b> <b>Incorrect username or password. Please try again.</b>	1. הזנת שם משתמש וסיסמא נכונים. 2. איפוס סיסמא.
<b>Forgot my password:</b> 1. Please enter both username and email. 2. Username or email is incorrect. Please try again.	1. הזנת כל הפרטים המבוקשים. 2. הכנסת פרטים נכונים – לפנות למנהל במקרה שלא יודעים את הפרטים.
<b>Admin Dashboard:</b> 1. Add user: a. Please fill in all fields b. Email already exists. Please choose a different email. c. Username already exists. Please choose a different username. 2. Reset Password: a. Passwords do not match.	1. הזמנת פרטים תקינים של משתמש חדש. במסך המנהל ניתן לראות את המשתמשים והאימיילים של כל המשתמשים במקרה שיש חשד של כפל נתונים. 2. הכנסת סיסמאות זהות ב2 החלונות.
<b>User Dashboard</b> 1. Reset Password: a. Passwords do not match.	1. הכנסת סיסמאות זהות ב2 החלונות.
<b>Search:</b> 1. No results found	1. שינוי שאילתה לחיפוש על מנת לנסות למצוא תוצאות מתאימות.

8. המשובים אותם קיבלנו:

A	B	C	D	E	F
הדיווח בברכת נכונות	ארכיון המסך (מנהל, חלקה לשימוש) ברורה ונכונה	הפירוט המעניינים ביותר לדעתי הם...	דעתי על הגרפים שהוצגו למנהל	אלמנטים שמונחי במערכת	ולשאר/למסך במערכת
5	5	5	5	5	5
5	5	5	5	5	5

כל המשובים שלנו היו טובים.

לכן אין מה לשנות במערכת.

הערות משנה	האם לדעתכם יש צורך בשינוי במערכת בעקבות ההערה?	ייחוס
------------	--	-------

9. התייחסות למשובי code review :

הערה	תגובה	האם בכוונת הצוות לבצע שינוי בעקבות ההערה? יש לפרט. אם כן, מה השינוי. אם לא, מדוע
פשטות	הצוות מסכים שהממשק ברור, אך הבין שחלק מהפונקציות מעט עמוסות מבחינה לוגית.	כן – בכוונתנו לפצל חלק מהפונקציות הארוכות (כמו <code>assign_task</code> ) לפונקציות עזר נפרדות.
מימוש	הצוות ציין שהוא השקיע הרבה בחלק הוויזואלי, אך היה פחות ממוקד באבטחה או תחזוקה.	כן – בכוונתנו להוסיף אימות משתמשים מתקדם יותר ולהפריד לוגיקה לקבצים נפרדים. (ביצענו שליחת סיסמה במייל)
מודלריות	הצוות מודע לכך שהכל מרוכז במחברת אחת לצורך ההגשה, אבל הסכים שזה מקשה על תחזוקה.	לא, ההוראות של הצוות החינוכי היו להגיש הכל כמחברת אחת.
יעילות	הצוות ציין שבוצעו בדיקות עומס, והמערכת מגיבה מהר. אין בעיות זמן ריצה.	לא נדרש שינוי – המערכת מגיבה מהר, ואין צורך באופטימיזציה בשלב זה.
באגים	לא נצפו באגים קשים במהלך הבדיקה. הצוות שמח לשמוע זאת.	לא נדרש שינוי כי אין באגים
טיפול בשגיאות	הצוות הודה שחלק מהפונקציות רגישות לשגיאות (למשל: חיבור ל-Firebase, קבצי חוץ).	כן – הוספנו <code>try/except</code> והודעות שגיאה ידידותיות למשתמש בממשק ולא רק בקונסול