

המחלקה להנדסת תוכנה - מבוא למחשב ענן

פרויקט

Turtles team:

יעקב בן שושן - 315064881 , יובל לרפלד - 318186897 , מתן עמרן - 316326313
עמיר אזולאי - 206695488 , חן צפיר - 319005104

https://github.com/einavbs1/Final_Project_Corelia
[Final Project Turtles.ipynb](#) :Colab

Username: admin1 Password: admin1
Username: user1 Password: user1

1.1. מודולים ופיצ'רים מעוניינים:

- 1.1.1. פיצ'ר - מנוע חיפוש אשר מター שאלות המשתמש יודיע להתאים את התוצאות שהמשתמש ירצה (כਮובן בעזרת פונקציות NLP).
- 1.1.2. מודל - NLP ששומש בהסרת word stop, פונקציית stem וpriority לדפים השונים.
- 1.1.3. פיצ'ר - widgets של python על מנת ליצור את הAI .
- 1.1.4. פיצ'ר - תצוגת הגרפים בעזרת matplotlib .
- 1.1.5. פיצ'ר - משחקי gamification הוסף משחקים למשתמשים שיוכלו להשלים משימות שמתעדכנות בזמן אמיתי.
- 1.1.6. מודל - Gemini API Chatbot יצרנו צאט בוט שאומן עם שאלות מראש קר שנitin לחקור אותו ולהשתמש בו על מנת לחקור על האתר שלנו וನושא הפרויקט.
- 1.1.7. פיצ'ר - security אבטחה והצפנה של הסיסמות השומרות בDB על מנת שלא יוכל לגשת לסיסמה לראות אותה.
- 1.1.8. פיצ'ר - לולאת מ.ukב אחריו שינוי נתונים בDB כך שהמידע יתעדכן בזמן אמיתי למנהל ADMIN לגבי משתמשים שמחברים, מתנתקים וכן חיפושים אחרים שמבצעים.

1.2. microservices שנעשה בהם שימוש:

- 1.2.1. שירות אינדקס - שירות שיצור אינדקס לאתר מסוים. בנוסף ניתן לבחור עמוק, בסיום השימוש בשירות האינדקס יופיע בDB
- 1.2.2. שירות שאלתה - שימוש בשירות ניתן ליצור שאלת שאלת אינדקס על מנת לקבל האינדקסים הרלוונטיים למילה. היתרון נובع מהעובדת שהשירותים עצמאיים יחסית, שכן אחד אחראי על פונקציונליות אחת ברורה. ומאפשר גמישות לשינויים ושיפורים ובנוסף אפשר להריץ רק את השירותים עצם במקרה הצורך.

1.3.1. איצות ודיוק הדיווח בזמן אמת (Real-Time Reporting Accuracy KPI)

האתגר: המערכת מבוססת על שינויים שמספקים נתונים בזמן אמת, אך חשוב לוודא שהנתונים שמגיעים לכל משתמש מוצגים באופן מדויק ומהימן. כל עיונות בדיווח יכול להוביל להחלטות שגויות מצד המשתמשים, במיוחד כשמדבר במידע קריטי כמו מצב הציד או תנאי המבידה.

מדידה: המדידה תבצע על ידי השוואת התוצאות שמתקבלות מהשינויים לבין הסטנדרטים שהוגדרו מראש (למשל, המקום המקורי של המערכת או מדידות הדינמיות). כל טעות במדידה שהסת�性 שלה מעלה ערך מסוים תיחס כטעונה תיקון.

תוצאה רצiosa: המטרה היא להבטיח שתוצאות הדיווח יהיו מדויקות ב-98%, והסת�性 מדידה ידנית לא תעבור $\pm 0.3^{\circ}$ מעלות צלייז. כל קראיה שטטיתתה גבוהה יותר תדרוש פעולה מיידית.

1.3.2. עקבות בהזנת נתונים לאורק זמן (Data Consistency KPI)

האתגר: אחת הבעיות המרכזיות במערכות נתונים היא חוסר עקבות בהזנה ובביצוע נתונים לאורק זמן. אם נתונים החישנים לא מזונים בצורה אחידה או שמערכת הדיווח לא יציבה, זה יפגע בהבנה הכללת של מצב המערכת.

מדידה: המדידה תבצע על ידי ניטור רציף של רמת השינויים בנתונים במהלך תקופות זמן שונות (לדוגמה, במשך שעה או יומם). נמדד את השיעור שבו הנתונים משתנים בצורה לא צפופה או יוצאים מהתווך הצפוי.

תוצאה רצiosa: המטרה היא לשמר על עקבות של לפחות 95% בכל המדידות. זה אומר שהשינויים במדדים לא יהיו חדים או לא צפויים, והם צריכים להיות בטוחו שנקבע מראש.

1.3.3. זמינות המשאבים בזמן שבר (Resource Availability During Faults KPI)

האתגר: כאשר יש תקלת במערכת, המערכת צריכה להיות מסוגלת לשמור על המשאים קרייטיים זמינים למשתמשים, כדי למנוע השפעה שלילית על תהליכיים. אם הכלים הקיימים אינם זמינים, זה עשוי להוביל לפגיעה בתפקוקה.

מדידה: המדידה תבצע על ידי מעקב אחרி זמן המשאים שהיו בלתי זמינים או לא פעילים במהלך תקלת. נמדד את הזמן שבו רכיבי מערכת קרייטיים (כמו חישנים או מערכת דיווח) לא היו זמינים בשל תקלות טכניות.

תוצאה רצiosa: המטרה היא להשיג זמינות של לפחות 97% למשאים הקרייטיים בזמן שבר, ככלומר, לא יותר מ-10 דקות ללא גישה למשאים החשובים ביותר.

1.3.4. אינטראקציה עם מערכת הצ'אטבוט (Chatbot Interaction KPI)

האתגר: הצ'אטבוט נדרש לעזור למשתמשים לפתור בעיות במערכות במהירות. מدد חשוב יהיה לבדוק עד כמה הצ'אטבוט עוזר למשתמשים בפתרון בעיות תוך חווית שירות טובה. **מדידה:** המדידה תבצע על פי אחוז השיחות שהסתינו בהצלחה כאשר הц'אטבוט פתר את הבעיה ללא צורך בתגובות אנושית. אנחנו גם נמדד את זמן התגובה של הצ'אטבוט לשאלות ולבעיות.

תוצאה רצiosa: המטרה היא להבטיח ש-90% מהשיחות בצ'אטבוט יסתיימו בהצלחה, ככלומר הבעיה תיפתר על ידי הצ'אטבוט מוביל שידרש מענה של אדם אנושי. כמו כן, הזמן לתגובה של הצ'אטבוט צריך להיות פחות מ-10 שניות.

1.3.5. סyncronization משימות ושיתוף פעולה בין העובדים (Task Synchronization and Collaboration KPI)

האתגר: המערכת פועלת על בסיס שיתוף פעולה בין עובדים, והשילובות בין העובדים יכול להניע את הביצועים, אך כאשר יש חוסר סyncronization בין המשימות המבוצעות על ידי כל עובד, זה עשוי להוביל לחסיפות, חוסר בהירות ולבזבוז זמן. אם העובדים לא מודעים

למשימות שבוצעו על ידי אחרים או לא יודעים מה נותר להם לעשות, זה יכול להוביל לעיכובים ולחוסר סדר.

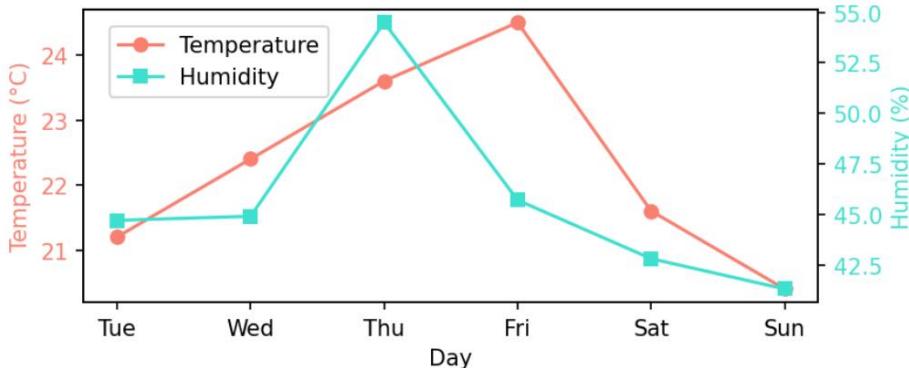
מדידה: המדידה תבוצע לפי אחוז המשימות שהזנו ונרשמו בזורה ברורה, כך שכל עובד יוכל לראות בזמן את המשימות שהושלמו על ידי אחרים ואת המשימות שנותרו להם להשלים. יש למדוד את כמות ההתראות שהתקבלו במערכת בונגע לאי-הבנות או שגיאות בשירות המשימות בין עובדים, לדוגמה, כאשר שני עובדים מבצעים את אותה המשימהambil לදעת שהוא כבר הושלמה.

תוצאה רצiosa: המטרה היא להציג לפחות לפחות 95% סyncron בין המשימות שהזנו במערכת, כך של עובד ידע בדיוק איזה משימות הושלמו על ידי אחרים ואיזה משימות נותרו לו לבצע. כמו כן, המטרה היא להבטיח שתהליכי השיתוף יגבר את המעורבות והשתתפות במשחקיות, ובכך תתרום לזריזות ולשיפור הביצועים הכלליים של הוצאות.

1.4. ניתוח נתונים עתק

ביצינו את הנתוח על נתונים החיצוני הפנימי Indoor, על מנת לטעון את הנתונים הרלוונטיים השתמשנו ב-**Spark DataFrame**, ולאחר מכן ייצרנו `groupBy` עליהם פועלות לחישוב הממוצע היומי, לאחר מכן השתמשנו נתונים ליצור גרפף המתאר את מגמות הטמפרטורה והלחות ב-6 הימים האחרונים.

Average Daily Temperature Last 6 Days (Indoor Sensor)



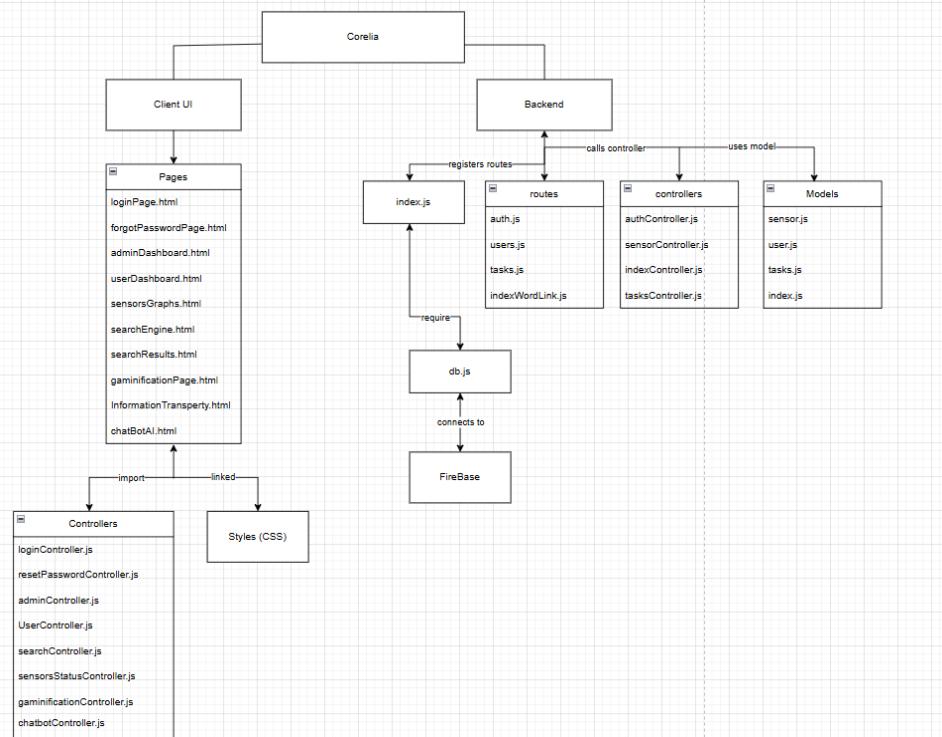
ניתוח הגרפ -

- שלישי-רביעי : הטמפרטורה יחסית נמוכה עם כ-21°C ולחות סביב 44-45%. לאחר עולה הטמפרטורה בכמה מעלות, והלחות נשארת בערך אותה רמה.
- חמישי : הלחות קפצת ל-55% והטמפרטורה עוקבת אחריה ועולה מעל -23°C.
- שישי-שבת : הלחות יורדת בצורה חדה ל-46% ביום שישי ול-42% ביום שבת, והטמפרטורה עולה מעט ואז יורדת חזרה ל-21.5°C. דבר שיכל לרמז על הפעלת קירור.
- ראשון : הלחות נמוכה במיוחד ב-41% והטמפרטורה יורדת עוד קצת ל-20°C - יכול להיות בגל קירור/אוויר חם או שהיה מועטה בחדר.
- לסיכום : בתחלת השבוע הטמפרטורה והלחות היו יציבים, עם עלייה קלה בטמפרטורה בעוד הלחות נשארה יציבה. באמצע השבוע הלחות קפצה והטמפרטורה טיפסה יחד איתה, נראית בשל שינוי בתנאי הסביבה או ריבוי פעילות באזורי. בסוף השבוע חל שינוי הפוך,

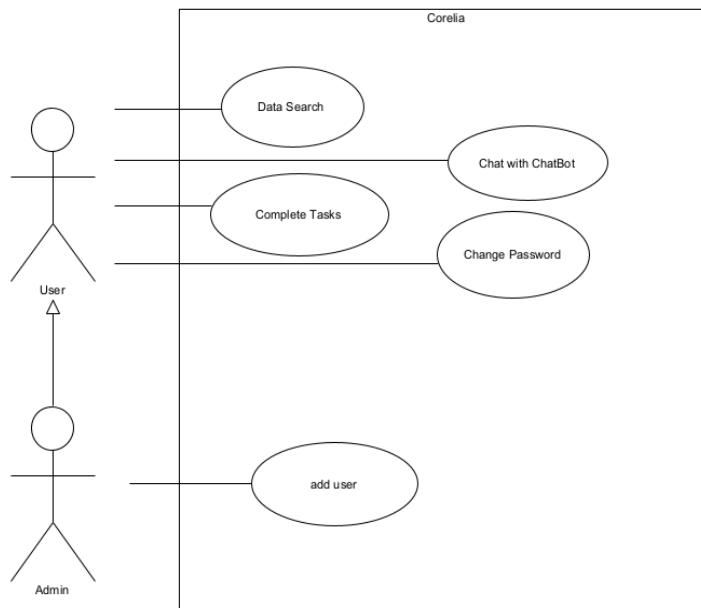
כלומר הלחות והטמפרטורות ירדו, מה שמייד על הפעלת מזגן או אורור מוגבר. ניתן להניח שהשינויים באקלים הפנימיים משקפים יותר פעולה אנושית ופעולת יוזם מאשר תנודות טבעיות אקראיות.

2. ארכיטקטורת המערכת

2.1. תרשימים ומאפיינים מרכזיים



2.2. מעדכן 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 סיסמה.
- 2.3.5. הוספה משתמש.

3. דרישות לא פונקציונליות:

- 3.1. המערכת תחבר את המשתמש תוך 3 שניות במוצע.
- 3.2. המערכת תציג את תוצאות החיפוש תוך 5 שניות במוצע.
- 3.3. כל סיסמות המשתמשים יאחסנו באופן מוצפן.
- 3.4. הקוד יירץ ללא שום תלות חיצונית.
- 3.5. המערכת תבדוק ותשווה סיסמות בהשואה לערך המוצפן בלבד ולא תשמור סיסמות לא מוצפנות.

4. ביקורת עמידים אשר ניתנה במהלך הסטודיו (שבוע 9) , כיצד התמודدتם אליה. נא לחשב גם את ציון SUS ולהתיחס אליו.

אלמנטים שיש לשפר במערכת	האם בוצע שינוי במערכת	নিম্নক
בשני המশובים לא צוינו בעיות / הצעות שיפור, لكن אין שינויים נדרשים. המערכת תואמת את ציפיות המשתמשים.		

חישוב שאלוני SUS :

1. הייתה רוצה להשתמש במערכת זו לעתים קרובות.
2. מצאת כי המערכת מסובכת ללא סיבה.
3. חשבתי שהמערכת קלה לשימוש.
4. אזדקק לתמיכה איש טכני כדי שאוכל להשתמש במערכת זו.
5. מצאת כי הפונקציות השונות של המערכת היו מתאימות היטב.
6. חשבתי כי היה יותר מיידי חוסר עקיבות במערכת זו.
7. לדעתי רוב האנשים יהיו מסוגלים ללמוד להשתמש במערכת זו בקלות.
8. מצאת כי המערכת מאוד מסובכת לשימוש.
9. חשתי ביטחון רב כאשר השתמשתי במערכת זו.
10. עלי ללמידה הרבה דברים לפני שאוכל להשתמש במערכת זו.

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

ציון SUS הינו 100 עקב המשובים שקיבלנו.

5. **פירוט שיקיפות אלגוריתמית:**

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

- אופן חישוב הניקוד
- שיטות אינדוקס המידע
- אופן הצגת הגרפים
- סוג הנתונים הנשמרנים במערכת

בנוסף, המערכת מבהירה כי לא נשמרים נתונים אישיים וכל עיבוד המידע נעשה אופן שקווי ומשמש לצרכים למודים בלבד.

6. **אתגרים אשר עלו במהלך העבודה, כיצד התמודدتם איתם:**

6.1. קושי ביצירת ויזואלייזציה שתורוץ בקורסא: בהתחלה ניסינו להשתמש בפלאסק, אך לאחר שיחה עם נעמי במהלך קבלת המשימות, גילינו שיש דרישת לא להריץ את היישום בצורה זו. היא המליצה לנו על גרדיו, אך גילינו שגם יש קשיים. בסופו של דבר, החלפנו השימוש בטאים כפוי שלמדו בתרגול, והפתרון הזה עבד מזמן.

6.2. קושי בהבאת מידע מהחישונים: במהלך העבודה נתקלנו בבעיות בהבאת המידע מהחישונים, מה שגרם לבנייה הקוד לששתנות באופן תדר. החישונים קרטסו לעיתים קרובות עד שנכנסנו במצב שבו לא ניתן היה לחץ מידע דרך הבราวקר. לפיכך, החלפנו להפיק פיקט דאטה המבוסס על הנתונים הראשוניים שהצלחנו לחוץ מהחישונים, והוספנו נתונים נוספים שייצרנו בעצמנו על מנת להציג גרפים סטטיסטיים.

6.3. קושי ביצוע איפוס אוטומטי לגימיפיקציה: היה אתגר במימוש האיפוס האוטומטי של המשימות מדי יום ושמירת ההיסטוריה. לדוגמה, משימות שהושלמו אטמול לא נמחקות או נשמרות פעמיים. מצאנו פתרון בכתיבת פונקציה שמבצעת איפוס יומי, מעבירה את המשימות שהושלמו לאריך לפי תאריך ומורידה אותן מה-chowndrop בעט ביצוע משימה כלשהי.

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

6.5. קושי במימוש האינדקס: חווינו קושי בניהול האינדקס, שהוא מאוחסן ב-DB ונדרש להיות מנוהל בשני טיפוסים שונים - מצד אחד לפי הוצר של Firebase, מצד שני לפי הצורך של התוכנית שרצה בפייתון. זה דרש עבודה רבה על מנת לסנן את המבנים השונים.

6.6. אתגר בניהול קוד ארכו: ניהול קוד ארכו היה אתגר מתמשך. מדובר בkowski גנרי, אך הוא לווה אותנו בכל הgesha והיה משמעותי במהלך העבודה.

6.7. קושי בעבודה עם נתונים פיקט: במקום Firebase, נקבעו למשוך את הנתונים שירות מ-Firebase, עבדנו עם נתונים פיקטיביים מבוססי נתונים ראשוניים שהצלחנו לחוץ מהחישונים. כדי לשלב אותם, הינו

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

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

6.9. קושי בהתחברות למערכת ובניהול הרשות: אתגר בהטמעה של אלגוריתם התחרבות והتنתקות למערכת, כך שכל משתמש יקבל גישה בהתאם להרשותו. וסיווג של כל משתמש לדוגמא אדמין יפנה 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, כולל 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 החדשניים של המערכת.

תיק תחזוקה:

תיק למתכנת - Corelia

בנייה ע"י: turtles group

גירסה: 1.3

תאריך שחרור הגירסה: 24/06/2025

חברי הצוות:

יעקב בן שושן - 315064881

יובל לרפלד - 318186897

מתן עמרן - 316326313

עמיר איזולאי - 206695488

חן צפיר - 319005104

◊ קבצים עיקריים:

- HW3_Turtles_Corelia -

◊ ספריות עיקריות:

• **FireBase** חיבור למסד נתונים בזמן אמת.

• **ipywidgets** ייצורמשק משתמש אינטראקטיבי במחברת.

• **Sendgrid** לשילוח מיילים (כולל איפוס סיסמה).

• **google.generativeai** שילוב עם Gemini לצ'אטבוט.

• **matplotlib, pandas, numpy** לניתוח והציג נתונים.

❖ הרשות משתמשים:

- **User** (פרטי דיפולט לבדיקות: username: user1 , Password: user1)
 - איפוס סיסמה.
 - אפשרות לשחק בטאב המשתמש וצפיה בኒקוד.

- **Admin** (פרטי דיפולט לבדיקות: username: admin1 , Password: admin1)
 - ייצור משתמש חדש למערכת.
 - צפיה בכל נתוני המשתמשים.
 - צפיה בכל המילים שנמצאים בDB.

• User Admin יורש הרשות Admin

❖ 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 גרפ טמפרטורה ולחות לאחר זמן.
create_daylight_hours_chart()	יצירת גרפ שעות אוור מול פעילות המשתמש.
calculateDailyScore()	חישוב ניקוד יומי למשתמש.
DisplayDailyRanks()	הציג דירוג יומי לכל המשתמשים.
complete_task()	סימון משימה כהושלה ועדכון המערכת.
monitor_loop()	ניתור משתמשים מחוברים בזמן אמת.
apply_stemming()	עיבוד טקסט לקיצוץ מילים לשורש (Stemming).
create_index_inspector()	יצירת משאך לצפייה באינדקס המילים.
password_reset_email()	שליחת מייל איפום סיסמה למשתמש.
update_tabs()	עדכון מסך הטאים בהתאם להרשות המשתמש
fetch_users_from_firebase()	שלילת רשימת משתמשים מה DB.
handle_login()	ניהול תהליך התחברות משתמש (טור הסתכלות על הרשות).
handle_logout()	ניהול תהליך התנתקות משתמש והסתרת כל הטאים.
on_click_search()	הפעלת חיפוש לפי טקסט שהוזן ע"י המשתמש.
saveUserSearchQuery()	שמירת שאלחת החיפוש של המשתמש ב-FireBase (למיעקב אחריו חיפושים אחרונים).
on_click_lucky()	חיפוש "Feeling Lucky m'a": פתיחת התוצאה המדורה הראשונה.
prioritize_pages()	דרוג העמודים בחיפוש לפי רמת רלוונטיות.

❖ **tabular data**

- **ממשק משתמש אינטראקטיבי עם widgets**
 - שימוש ב- Tab(`widgets.Tab`) לבניית טאבים נוחים למעבר בין הדפים השונים שיש למערכת להציג.
 - שימוש ב- HTML(`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 להציג דירוג יומי / חודשי:
 - `DisplayDailyRanks()`
 - `DisplayMonthlyRanks()`
- **גרפים סטטיסטיים עם Matplotlib**
 - גרפ טמפרטורה ולחות לאורך זמן - `(create_temp_humidity_chart)`
 - גרפ שעות אור - `(create_daylight_hours_chart)`
- **כפתור "Feeling Lucky"**
 - כפתור "יעוד" שמודול אוטומטי לתוכה 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 משתמש בפונקציה שמייצרת סיסמה רנדומלית (כਮון לאחר שבודקת שפטו המשמש נכונים ואומתו) - `(password_reset_email)`
- **-Real-time User Monitoring**
 - לולאת רקע (`thread`) לניטור משתמשים מחוברים ולעדכן טבלת האדים - `(monitor_loop)`
- **מנוע חיפוש חכם עם Stemming**
 - עיבוד מילים לשורש (`stemming`) לשיפור תוצאות החיפוש
 - `apply_stemming()`
 - `get_search_results()`

הטוען 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 = "einavbs1@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)


```

הציג הטאים 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 :

תיק למשתמש - Corelia

בבנה ע"י : turtles group.

גירסה: 1.3

תאריך שחרור הגירסה: 24/06/2025

חברי הצוות:

יעיבן בן שושן - 315064881

יובל לרפלד - 318186897

מתן עמרן - 316326313

עמיר איזולאי - 206695488

חן צפיר - 319005104

❖ הסבר כללי על המערכת:

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

המערכת מאפשרת ניהול יומיומי של פעילות העובדים, מוקב אחר השימוש במידע מהשינויים, ומתן

כלים לשיפור הלמידה והמעורבות בעזרת מרכיבי משחקים (Gamification).

במערכת משולב מנגנון תחרויות יומיות וחודשיות בין העובדים, שמטרתו לעודד את העובדים להשלים משימות יומיות ולצבור ניקוד.

העובדים המתמידים ביותר יתרגו בittelת הדירוג ויתמודדו על תואר עובד החודש.

למנהלים ניתנת אפשרות:

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

המערכת כוללת גם צ'אט בוט לימוד המשלב מידע אודiot עולם המחשב והענן. העובדים יכולים לפנות לצ'אט בוט בכל שלב ולהעמיק את הידע שלהם בנושאים שונים, כחלק ממדייניות העשרה והלמידה של הארגון.

במהלך השימוש במערכת, כל עובד יכול לעبور בקלות בין הטאים והתכנים השונים המותאימים להרשותו - כולל:

- **משימות יומיות**
- **דירוגים**
- **חיפוש מידע**
- **אינדקס מילימ**
- **צ'אטבוט לימודי**

המערכת בונה כרך שתספק חוותית משתמש חדש פשרה ונוחה גם לעובדים חדשים במערכת, עם דגש על חוותית שימוש יידוטית ותמייה מלאה בעברית.

❖ מעברים בין מסכינים:

• מסך התחברות:

- **מסך איפוס סיסמא.**
- **הציגת כל הטאים (כמנהל).**
- **הציגת כל הטאים (כמשתמש רגיל).**

• מסך "יעודי למשתמש":

- **יציאה לדף התחברות.**

• מסך איפוס סיסמא:

- **יציאה לדף התחברות.**

• כפתור "Feeling Lucky" במנוע החיפוש:

- **פתיחה התוצאה המתאימה ביותר בכרטיסיה חדשה.**

• כפתור "Search" במנוע החיפוש:

- **מסך תוצאות החיפוש.**

כל שאר המעברים בין הדפים מתבצעות דרך הcpters הפיזיים
בממשק.

❖ **פירוט מסכי האתר:**

שם המסך	תיאור מילולי קוצר
Login	במסך זה ניתן להתחבר למערכת באמצעות שם משתמש וסיסמה (עובד רגיל או מנהל).
Forgot my password	במסך זה ניתן לאפס סיסמה. לאחר מלאי הפרטים, תישלח סיסמה חדשה למיל.
Admin Dashboard	מסך ייעודי למנהל. מאפשר לבצע את כל פעולות ניהול (ניהול עובדים, ניטור, צפיה בחיפושים וכו').
User Dashboard	מסך אישי לעובד רגיל. מאפשר לבצע פעולות בסיסיות, לצפות במשימות, ניקוד וחיפושים.
Graphs	מסך המציג גרפים חשובים לעובד: תקינות סביבת העבודה, מידע עדכני מהחיצון, גרפים היסטוריים (ימים/שעות אחרונות).
Gamification	מסך המשחקיות: כל עובד (או מנהל) יכול להשתתף בתחרויות עובד החודש, לבצע משימות יומיומיות ולצבור ניקוד.
Search	מסך חיפוש: מאפשר חיפוש מיידי על תקלות או נושאים שונים במערכת. כולל: 1. כפתור חיפוש רגיל: להציג דף תוצאות*. 2. כפתור Feeling Lucky I'm : מעבר ישיר לתוצאה הראשונה בדף חדש (מקל על חווית החיפוש ומזכיר אותה).
Results*	מסך הצגת תוצאות החיפוש, ממוננות לפי רלוונטיות לשאלתה.
Information Transparency	מסך מידע: מסביר על כל דף באתר, ומשקף למשתמש את הנתונים השונים.
Chatbot Assistant	צ'אטבוט אינטראקטיבי: מאפשר לשאול שאלות ולחזור נושאים מתחום פלטפורמות המחשב בענין.

❖ **פירוט שקיופות אלגוריתמיות:**

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

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

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

❖ אפשרויות שגיאה ופתרונות:

פתרונות השגיאה	שם המסר + שגיאות
1. הזנת שם משתמש וסיסמה נכונים. 2. איפוא סיסמה.	Login: Incorrect username or password. Please try again.
1. הזנת כל הפרטים המבוקשים. 2. הכנסת פרטים נכונים - לפנות למנהל במקרה שלא יודעים את הפרטים.	Forgot my password: Please enter both username and email. Username or email is incorrect. .2 Please try again.
1. הזמנת פרטים תקניים של משתמש חדש. במקרה המנהל ניתן לראות את המשתמשים והאימיילים של כל המשתמשים במקרה שיש חשד של כפל נתונים. 2. הכנסת סיסמות זרות ב2 חלונות.	Admin Dashboard: Add user: .1 Please fill in all fields .a Email already exists. .b Please choose a different email. Username already exists. .c Please choose a different username. Reset Password: .2 Passwords do not match. .a
1. הכנסת סיסמות זרות ב2 חלונות.	User Dashboard Reset Password: .1 Passwords do not match. .a
1. שינוי שאלתה לחיפוש על מנת לנסות למצוא תוצאות מתאימות.	Search: No results found .1

8. המושבים אותם קיבלנו:

A	B	C	D	E	F
		התקנות החומר (גנבה, הרקע לסייעות) בוגרונה וכוכבה	המשרדים המבוקשים מילויו לשלוח. הם	המשרדים שabayit ומשרדים	ולא/or לאפקט בשרותה
5	5	5 Refreshing data, new tab for search results.	5 Refreshing data, new tab for search results.	5 Refreshing data, new tab for search results.	5 Refreshing data, new tab for search results.

כל המושבים שלנו היו טובים.
לקין מה לשנות במערכת.

הערות מושב	האם לדעתכם יש צורך בשינוי?	כמזהק
במערכת בעקבות ההערה?		

9. התיחסות למשוב : code review

הערה	תגובה	האם בכוון הוצאות לבצע שינוי בעקבות ההערה? יש לפרט. אם כן, מה השינוי. אם לא, מדוע
פשטות	הוראות מסכימים שהמשק ברור, אך הבין שחלק מהfonקציות מעט מעססות מבחינה לוגית.	כן: בכוונתנו לפצל חלק מהfonקציות הארוכות (כמו <code>assign_task</code>) לפונקציות עזר נפרדות.
IMPLEMENTATION	הוראות ציינן שהוא skuious הרבה בחלוקת הויזואלי, אך היה פחות מוקד באבטחה או תחזוקה.	כן: בכוונתנו להוסיף אימוט משתמשים מתקדם יותר ולהפריד לוגיקה לקבצים נפרדים. (ביצענו שליחת סיסמה במיל')
מודולריות	הוראות מודיע על כך שהכל מרכז במחברת אחת לצורך ההגשה, אבל הסכים זהה מקשה על תחזוקה.	לא, ההוראות של הוצאות החינוכי היו להגיש הכל במחברת אחת.
יעילות	הוראות ציינן 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 קשיים במהלך הבדיקה. הוראות שמח לשמעו זאת.	לא נדרש שינוי כי אין באגים
טיפול בשגיאות	הוראות הודה שחלק מהfonקציות רגישות לשגיאות (למשל: חיבור ל-Firebase, קובץ חזץ).	כן: הוספנו <code>try/except</code> והודעתו שגיאיה ידידותיות למשתמש במשק ולא רק בקונסול

מקורות:

1. ChatGPT
2. Gemini
3. [/https://ipywidgets.readthedocs.io](https://ipywidgets.readthedocs.io)
4. Google
5. Youtube
6. <https://www.geeksforgeeks.org/python>
7. [/https://mqtt.org](https://mqtt.org)
8. Wikipedia

:Prompts

1. "אילו מדדי KPI יכולו לסייע לנו להבין האם המערכת פועלת באופן תקין? "
2. "איך ניתן לבטל את כלות הקבצים המקומיים באתר שלנו? שהאיון יפתח ללא צורך בהתקנה
Locale של הקובץ"
3. "אילו גרפים יניתו לנו את המיטב מהנתונים הבאים של החישנים?:"
4. "איך אני יכול לקשר את האתר שלי לאינדקס שנמצא ב-FIREBASE?"
5. "איך יהיה ניתן להוציא / לשנות / לקבל נתונים מהDB שלנו? "
6. "אילו ספירות יכולו לעזור לנו כדי להעיר את נראות האתר?"