**המחלקה להנדסת תוכנה - קורס מבוא למחשוב ענן**

**סמסטר אביב התשפ"ד**

**תרגיל בית 3**- עבודה **בצוותי העבודה**

מועד הגשה:

15.8.24

גאד טאהא 206893075

ראמי טאהא 207410523

עבד אלכרים זידאן 212101943

מוחמד חטיב 212704167

חליל מנסור 322965492

אבראהים גראמנה 212186209

המשימה בתרגיל זה: סיום בניית המערכת, ביצוע קוד ריוויו לצוות אחר - איטרצית פיתוח מס 2

שימו לב: למטלה זו שלושה חלקים

קישור למחברת קולאב : <https://colab.research.google.com/drive/1vH06qNj_Rp_390fIyf5jf1Oz8ela8PAA?usp=sharing>

קישור לGITHUB :

<https://github.com/SharkZeedan/Cloud-Computing>

**חלק ראשון.**(10 נקודות).עליכם להגדיר את בעלי התפקידים לאיטרציה זו (יש להחליף מתרגיל בית 2) יש לרשום מי מהנדס.ת המערכת, האחראי.ת בתרגיל זה.

|  |  |  |  |
| --- | --- | --- | --- |
| **שם חבר הצוות** | **משימות שהוקצו** | **משימות שהושלמו** | **Acceptance test –** בדיקת קבלה |
| ראמי טאהא + גאד טאהא + כרים זידאן | וודאות שהמסכים תקינים + צטבוט + שינוי בקוד ( אם צריך) , (חלק 2) | הכל | לבדוק אם כל המסכים עובדים בצורה תקינה , ואין כפתורים למסכים שלא קיימים , ולבדוק אם בצטבוט עובד בצורה תקינה. |
| חליל מנסור | תיק מתכנת | הכל | לבדוק האם הקבצים המרכזיים והפונקציות המרכזיות הוסברו בצורה מובנה וברורה. |
| מוחמד חטיב | תיק משתמש | הכל | לבדוק אם כל הוסברו מטרת כל אחד מהמסכים ואיך להשתמש בכם , וגם אם הוסבר את המעברים בין המסכים. |
| אבראהים כראמנה | הכנת הסרטון | הכל | לבדוק האם האודיו בסרטון נשמע וברור , וגם אם ההסבר מקצועי. |
|  |  |  |  |

***חלק שני: בניית המערכת*** (70 נקודות)

עליכם לסיים את בניית המערכת.

בפרט, יש לוודא קיום של כלל האלמנטים הבאים:

1. כל הפיצ'רים שהתחלתם לממש בתרגיל בית 2 - יש לוודא שכל המסכים תקינים ועובדים כראוי, **בקולאב**,ולעבוד עם firebase לצורך שמירת נתונים .אין להשאיר קישורים או כפתורים למסכים שלא קיימים.
2. צטבוט שישולב במערכת (כפי שלמדתם בתרגול 8), הכולל מידע אשר מגיע מה - DB שלכם.
3. התייחסות לאלמנטים רלוונטיים לקובץ ה -JSON שקיבלתם ( לפי מספר הקובץ, ולפי השאלות בנספח א)
4. התייחסות למשוב שקיבלתם בשבוע 10 – code review – יש להגיש את המשוב שקיבלתם מהצוות השני בחלק הקבוצתי , ולכתוב התייחסות מתאימה:לכל הערה, יש לרשום אותה ולציין האם נעשה שינוי בקוד בעקבות זאת.

|  |  |  |
| --- | --- | --- |
| **הערה** | **תגובה** | **האם בכוונת הצוות לבצע שינוי בעקבות ההערה? יש לפרט. אם כן, מה השינוי. אם לא, מדוע** |
| ניגודיות בצבעים בתפריט הראשי | הסבירו מדוע הצבעים טובים . | What does Contradiction means to you, since the comment is related to a non functional requierment and may not explicitly be wrong for others , it’s a personal opinion,  no changes will be made. |
| כמות שעות כוללת של עבודת הסטודנטים - לשנות כך שתהיה מוצגת עבור כל תקופת זמן מצומצמת וממוקדת | מקבלים את ההערה הבונה ויבצעו שינוי בהתאם. | This feature was asked from as a requierment in HM3 ( add the total working hours of each worker, students who worked in night shifts and holidays.)  We will not entirely remove this feature but certainly discuss adding a new one that targets a smaller period of time. |

***חלק שלישי : סגירת הפרויקט \_( 20 נקודות)***

בחלק זה תכינו מסמכים המתארים את הפרויקט שלכם:

1. יש לבנות תיק למתכנת הכולל את שמות כל הקבצים המרכזיים, פונקציות מרכזיות, קטעי קוד/תבניות עיצוב מעניינים שהשתמשתם בהם.

### *פונקציות מרכזיות בקוד:*

**fetch\_page(url)**:

* + **תיאור**: שולפת את תוכן הדף מה-URL הנתון ומחזירה את התוכן כ-BeautifulSoup object.
  + **איך זה עובד**:
    - מבצעת בקשה לכתובת ה-URL.
    - אם התגובה תקינה (קוד סטטוס 200), מנתחת את התוכן באמצעות BeautifulSoup ומחזירה את התוצאה.
    - אחרת, מחזירה None.

**index\_words(soup)**:

* + **תיאור**: יוצרת אינדקס של מילים מתוך התוכן של BeautifulSoup object.
  + **איך זה עובד**:
    - שולפת את כל המילים מהטקסט בדף.
    - סופרת את תדירות כל מילה ומחזירה מילון עם המילים והכמויות שלהן.

**remove\_stop\_words(index)**:

* + **תיאור**: מסננת מילים נפוצות שאינן תורמות למשמעות (stop words) מהאינדקס.
  + **איך זה עובד**:
    - מסירה מילים נפוצות מתוך האינדקס לפי רשימה מוגדרת.
    - מחזירה את האינדקס לאחר הסינון.

**apply\_stemming(index)**:

* + **תיאור**: מבצעת צימוד (stemming) למילים באינדקס כדי לאחד צורות שונות של אותה מילה.
  + **איך זה עובד**:
    - מפעילה Porter Stemmer על כל מילה באינדקס.
    - מעדכנת את האינדקס עם מילים צמודות ומחזירה את התוצאה.

**search(query, index)**:

* + **תיאור**: מחפשת מונחים בתוך האינדקס ומחזירה את תדירותם.
  + **איך זה עובד**:
    - מבצעת צימוד על מונחי החיפוש.
    - מחפשת את המונחים בצורתם הצמודה באינדקס ומחזירה את התדירויות שלהם.

**search\_engine(url, query)**:

* + **תיאור**: מבצעת חיפוש בדף לפי שאילתת חיפוש.
  + **איך זה עובד**:
    - שולפת את תוכן הדף מה-URL.
    - יוצרת אינדקס למילים, מסננת stop words, מבצעת צימוד ומבצעת חיפוש לפי השאילתה.
    - מחזירה את התוצאות.

**find\_index(url)**:

* + **תיאור**: יוצרת אינדקס למילים בדף ומביאה אותו.
  + **איך זה עובד**:
    - שולפת את תוכן הדף מה-URL.
    - יוצרת אינדקס, מסננת stop words ומבצעת צימוד.
    - מחזירה את האינדקס.

**add\_index\_to\_firebase(url)**:

* + **תיאור**: מוסיפה אינדקס למילים ל-Firebase.
  + **איך זה עובד**:
    - מביאה אינדקס מהדף וממירה אותו לרשימה של אובייקטים.
    - שולחת את הרשימה ל-Firebase ומדפיסה את התגובה.

**create\_most\_frequent\_activity\_page()**:

* + **תיאור**: יוצרת עמוד HTML המציג גרף של חמש הפעילויות השכיחות ביותר.
  + **איך זה עובד**:
    - סופרת את התדירויות של הפעילויות מהנתונים.
    - יוצרת גרף בעזרת matplotlib, שומרת אותו כתמונה ומציגה אותו בדף HTML.

**create\_activity\_distribution\_over\_time\_page()**:

הפונקציה create\_activity\_distribution\_over\_time\_page יוצרת דף HTML המציג גרף היסטוגרמה שמתאר את התפלגות הפעילויות לאורך זמן. היא אוספת נתונים הכוללים זמנים, ממירה אותם לתאריכים, ובונה היסטוגרמה באמצעות matplotlib. לאחר מכן, היא מציגה את הגרף כקובץ תמונה בפורמט base64 בדפדפן, יחד עם כפתור לחזרה לעמוד הראשי.

**create\_on\_most\_frequent\_worker\_names\_page()**:

הפונקציה create\_on\_most\_frequent\_worker\_names\_page יוצרת דף HTML המציג טבלה של השמות השכיחים ביותר של עובדים מתוך הנתונים שהורדו מ-Firebase. היא סופרת את מספר הפעמים שכל שם מופיע, ממיינת את השמות לפי שכיחותם בסדר יורד, ומציגה את המידע בטבלה מסודרת. בנוסף, היא מוסיפה כפתור לחזרה לעמוד הראשי.

**create\_on\_recent\_activity\_details\_button\_click\_page():**

הפונקציה create\_on\_recent\_activity\_details\_button\_click\_page יוצרת דף HTML המציג טבלה עם פרטי הפעילויות האחרונות (עד 20 רשומות) מתוך הנתונים שהורדו מ-Firebase. הטבלה כוללת עמודות כמו משתמש, תיאור, מסמך, כרטיסיה, וזמן הפעילות. בנוסף, היא מוסיפה כפתור לחזרה לעמוד הראשי.

### create\_chatbot\_page():

פונקציה זו יוצרת עמוד צ'אטבוט הכולל עיצוב, תיבת טקסט להזנת שאלות, כפתור לשליחה, והוראות לשאלות שניתן לשאול את הצ'אטבוט. כל התשובות מתקבלות על ידי קריאה לפונקציה answer\_question.

### is\_night\_shift():

בודקת אם השעה שמוזנת בפורמט "%Y-%m-%d %H:%M:%S" היא חלק ממשמרת לילה (בין 20:00 ל-06:00).

### is\_weekend():

בודקת אם התאריך שמוזן הוא סוף שבוע (שבת או ראשון).

### calculate\_total\_hours():

מחשבת את מספר השעות הכולל שעובד כלשהו עבד, בהתבסס על הנתונים שנשלפו מהמסד.

### count\_special\_shifts():

מחשבת את מספר המשמרות המיוחדות (לילה, סוף שבוע) שעובד עבד ומחזירה את הנתונים במבנה מילון.

### answer\_question():

מנתחת את השאלה שהוזנה על ידי המשתמש, ומשיבה תשובה בהתאם לנתונים מהמסד, בהתבסס על מילות מפתח כמו "users", "tabs", "total hours" וכדומה.

**fetch\_data\_from\_firebase:**

פונקציה זו אחראית על קבלת הנתונים ממסד נתונים Firebase. היא משתמשת בכתובת URL שניתנת כדי לשלוח בקשת GET, ומעדכנת את המשתנה הגלובלי data עם הנתונים המתקבלים. במידה וקיימת בעיה בקבלת הנתונים (כגון בעיות אינטרנט), היא תציג הודעת שגיאה.

**create\_upload\_page:**

פונקציה זו יוצרת עמוד HTML שמאפשר העלאת קובץ JSON ל-Firebase. היא כוללת עיצוב של העמוד, יצירת אזור דראג-אנד-דרופ (Drag-and-Drop) להעלאת הקובץ, כפתור להעלאה, וקוד JavaScript שמטפל בתהליך הדראג-אנד-דרופ והעלאת הקובץ.

**upload\_json\_to\_firebase(content)**:

פונקציה זו מופעלת כאשר המשתמש מעלה קובץ JSON דרך העמוד שנוצר על ידי create\_upload\_page. היא מפרשת את הקובץ המועלה (אם הוא תקין) ומעלה אותו למסד הנתונים ב-Firebase על ידי מחיקת הנתונים הקודמים במסד ולאחר מכן העלאת הקובץ החדש. בסיום, הפונקציה מבצעת קריאה ל-fetch\_data\_from\_firebase כדי לעדכן את הנתונים ביישום.

**create\_search\_page()**:

 **מה זה עושה**: יוצר דף חיפוש שבו המשתמש יכול להזין פרמטרים כמו תיעוד, טאב, זמן התחלתי, זמן סיום, ומשתמש כדי לחפש ולסנן נתונים. התוצאות מוצגות בטבלה.

 **איך זה עובד בקוד**:

1. מגדיר CSS מותאם אישית לעיצוב הדף.
2. יוצר שדות קלט עבור הפרמטרים השונים ולחצן חיפוש.
3. כאשר נלחץ לחצן החיפוש, פונקציית search\_firebase מחפשת את הנתונים ב-Firebase בהתבסס על הפרמטרים שניתנים ומציגה את התוצאות בטבלה.

### create\_chart\_page():

 **מה זה עושה**: מציג תרשים עוגה המראה את התפלגות הפעילות לפי משתמשים.

 **איך זה עובד בקוד**:

1. אוסף את הנתונים מ-Firebase ומחשב את הפעילות לפי משתמשים.
2. יוצר תרשים עוגה עם נתוני התפלגות וממיר אותו לתמונה.
3. מציג את התרשים בתמונת HTML.

### create\_most\_frequent\_activity\_page()

 **מה זה עושה**: מציג תרשים עמודות עם עשרת הפעילויות הנפוצות ביותר מתוך הנתונים.

 **איך זה עובד בקוד**:

1. אוסף את הנתונים מ-Firebase ומחשב את תדירות הפעילויות.
2. יוצר תרשים עמודות עם עשרת הפעילויות הכי נפוצות וממיר אותו לתמונה.
3. מציג את התרשים בתמונת HTML.

### create\_activity\_distribution\_over\_time\_page()

**מה היא עושה:** יוצרת דף המציג גרף של פעילויות מחולקות לפי זמן. היא סופרת את הפעילויות לפי תאריך ומציגה אותן בגרף.

**איך היא פועלת:**

* שואבת נתונים עדכניים מ-Firebase.
* מחשבת את מספר הפעילויות לפי תאריך.
* מציגה גרף קווים המראה את מספר הפעילויות לכל תאריך.
* ממירה את הגרף לתמונה בקידוד Base64 בפורמט PNG.
* יוצרת דף HTML המראה את הגרף ומכיל כפתור "חזור לדף הראשי".

### create\_on\_most\_frequent\_worker\_names\_page()

**מה היא עושה:** יוצרת דף המראה גרף עמודות של 10 העובדים הפעילים ביותר לפי מספר הפעילויות.

**איך היא פועלת:**

* שואבת נתונים מ-Firebase.
* סופרת את מספר הפעילויות לכל עובד.
* יוצרת גרף עמודות של 10 העובדים עם הכי הרבה פעילויות.
* ממירה את הגרף לתמונה בקידוד Base64 בפורמט PNG.
* יוצרת דף HTML המראה את הגרף ומכיל כפתור "חזור לדף הראשי".

### create\_on\_recent\_activity\_details\_button\_click\_page()

**מה היא עושה:** מציגה דף עם טבלה של 20 הפעילויות האחרונות.

**איך היא פועלת:**

* שואבת נתונים עדכניים מ-Firebase.
* ממיינת את הפעילויות לפי זמן בסדר יורד ומביאה את 20 האחרונות.
* יוצרת טבלת HTML עם פרטי הפעילויות האחרונות.
* מציגה את הטבלה ומכילה כפתור "חזור לדף הראשי".

**create\_salary\_page()**:

**מה היא עושה:** מספקת טופס לחישוב והצגת השכר הכולל עבור משתמש ספציפי לפי שיעור שכר ושעות עבודה.

**איך היא פועלת:**

* מגדירה סגנון CSS לטופס ולתוצאה.
* יוצרת רכיבי טופס לקבלת קלט מהמשתמש ולחישוב.
* מחשבת את השעות הכוללות והשכר עבור המשתמש הספציפי.
* מציגה את התוצאה בטבלה ב-HTML ומכילה כפתור "חזור לדף הראשי".

### create\_working\_hours\_page():

**מה היא עושה:** יוצרת דף המציג את שעות העבודה הכוללות לכל תלמיד.

**איך היא פועלת:**

* שואבת נתונים מ-Firebase.
* מחשבת את שעות העבודה הכוללות לכל תלמיד.
* יוצרת טבלה ב-HTML המציגה את שעות העבודה של כל תלמיד.
* מציגה את הטבלה ומכילה כפתור "חזור לדף הראשי".

### create\_night\_weekend\_workers\_page():

**מה היא עושה:** מציגה דף עם רשימת עובדים שעבדו בלילה או בסופי שבוע.

**איך היא פועלת:**

* שואבת נתונים מ-Firebase.
* מזהה עובדים שעבדו בשעות לילה או בסופי שבוע.
* יוצרת טבלאות HTML המפרטות את העובדים הללו.
* מציגה את הטבלאות ומכילה כפתור "חזור לדף הראשי".

### טיפול בכפתורי לחיצה:

**מה הם עושים:** מגדירים פעולות לכפתורים שונים כדי לנווט לדפים שונים כשנלחצים.

**איך הם פועלים:**

* כל פונקציה מנקה את הפלט הנוכחי וקוראת לפונקציה המתאימה ליצירת הדף.

### show\_main\_page(b=None):

**מה היא עושה:** מציגה את הדף הראשי עם כפתורי ניווט.

**איך היא פועלת:**

* יוצרת פריסת HTML לדף הראשי.
* מציגה כפתורים לניווט לחלקים שונים.
* מציגה את הכפתורים בתצורת תיבה אנכית.

### upload\_json\_to\_firebase():

**מה היא עושה:** מטפלת בהעלאת נתוני JSON ל-Firebase.

**איך היא פועלת:**

* לא כלולה בקוד שסופק, אך רשומה כטיפול להעלאות קבצים.

### fetch\_data\_from\_firebase():

**מה היא עושה:** שואבת נתונים עדכניים מ-Firebase ומציבה אותם במשתנה גלובלי data.

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

תיק משתמש

* 1. הסבר כללי על המערכת :-

המערכת היא מערכת ניהול ומעקב אחר פעילויות משתמשים. היא מאפשרת למנהלי מערכת לצפות במידע על פעילויות המשתמשים, לחפש מידע ספציפי, ולראות ניתוחים שונים.

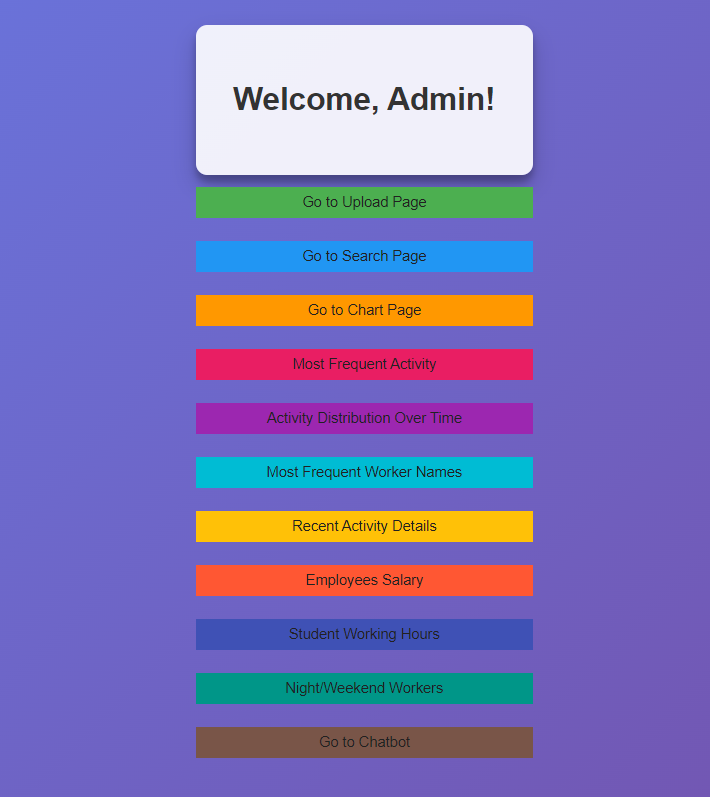
מטרת המערכת :-

* מעקב אחר פעילויות משתמשים.
* ניתוח דפוסי עבודה.
* חישוב שעות עבודה ושכר.
* הבנה של הנתונים באמצעות גרפים וטבלאות.
  1. פירוט מסכים + טעויות אפשריות:-

המערכת כוללת מספר מסכים עיקריים:

א. מסך ראשי:- Main Page

מציג תפריט עם כפתורים לגישה למסכים שונים של המערכת.



מהמסך הראשי ניתן לעבור לכל אחד מהמסכים האחרים באמצעות לחיצה על הכפתור המתאים.

ב. מסך העלאת קובץ:- Upload Page

מאפשר העלאת קובץ JSON לבסיס הנתונים.



להעלאת קובץ חדש של הנתונים אתה יכול ללחוץ על המלבן הגדול לבחור את הקובץ, או לגרור את הקובץ למלבן ולשחרר.

וודא שהעלית הקובץ הנכון, ובלחיצה על הכפתור Upload to Firebase, הקובץ יעלה למסד הנתונים.



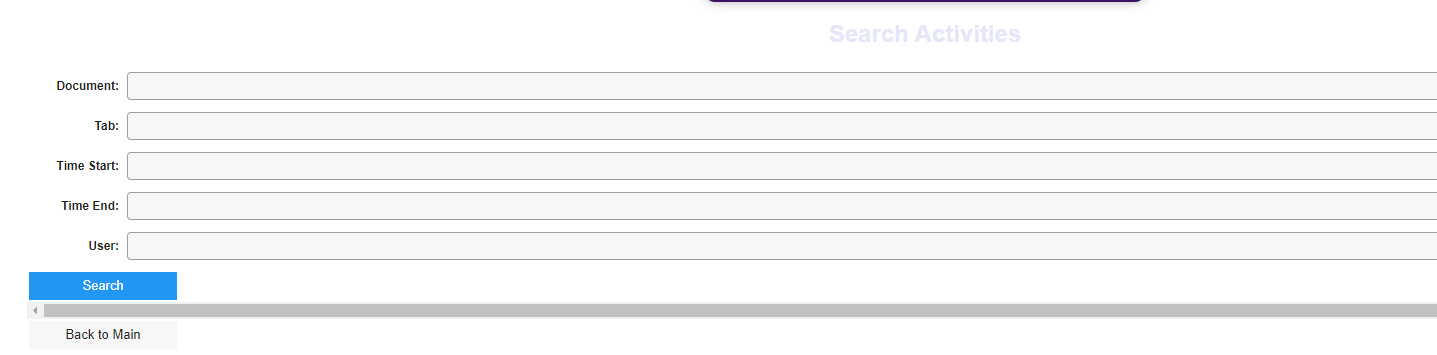
אם מופע שתי השורות האלה , אז הכול תקין ואתה יכול להמשיך.

טעויות אפשריות:-

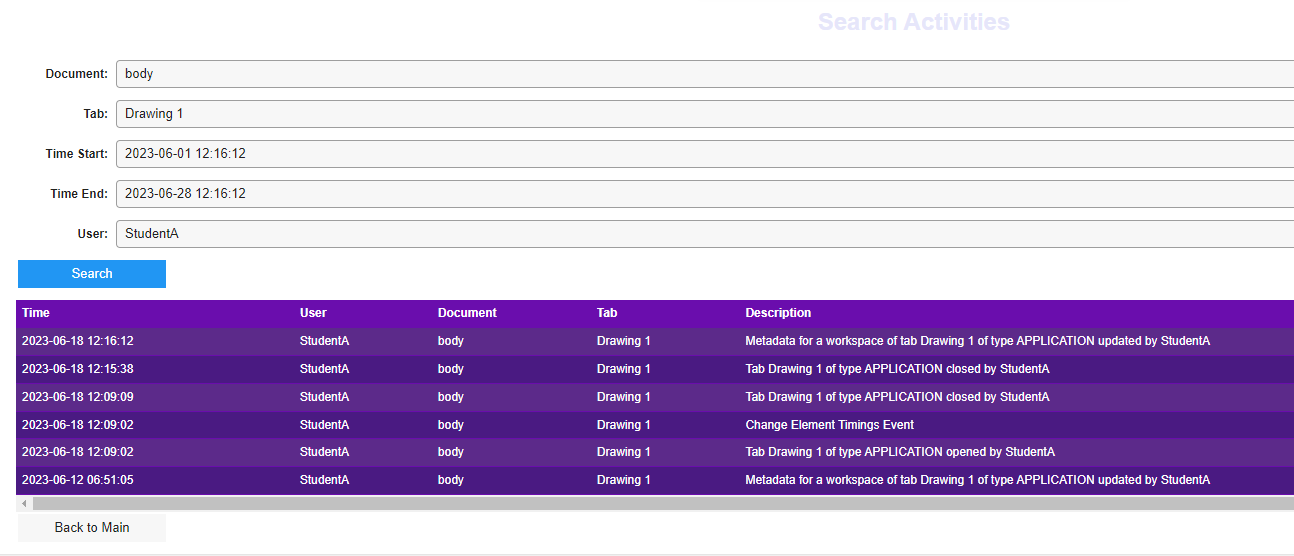
העלאת קובץ שהוא לא JSON.

ג. מסך חיפוש:- Search Page

מאפשר חיפוש פעילויות לפי פרמטרים שונים כמו משתמש, מסמך, וטווח זמנים.



להשתמש המסך הזה , צריך למלא את הנתונים שצריך לחפש עליהם, וללחוץ כל הכפתור Search , למשל:-



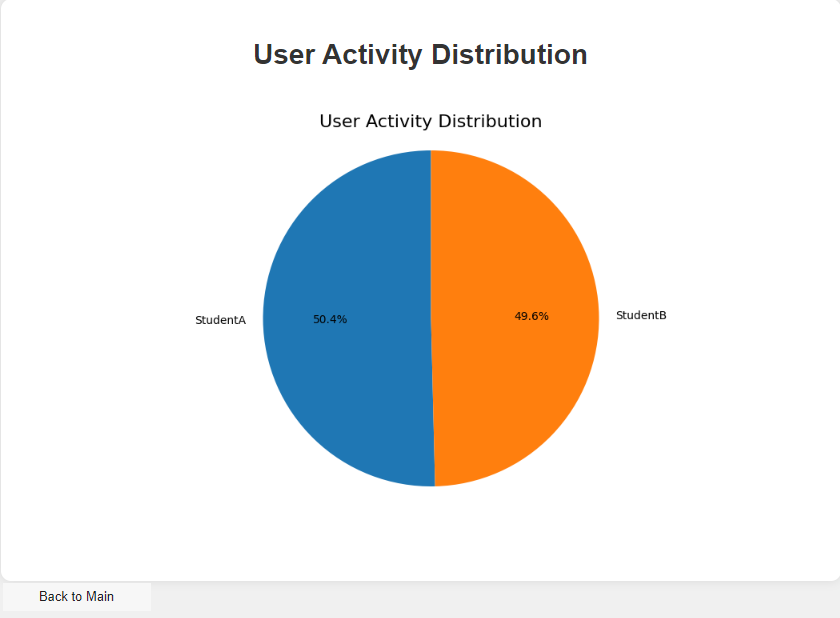
אם אין נתונים במסד הנתונים לפי השדות שהוכנסו , יופיע:- No matching data found.

אם אחד השדות ריק , אז יוצג כול הנתונים לשדה הזה , למשל אם כול השדות ריקים , אז המערכת תציג את כול הנתונים בטבלה.

ד. מסכי ניתוח וויזואליזציה:

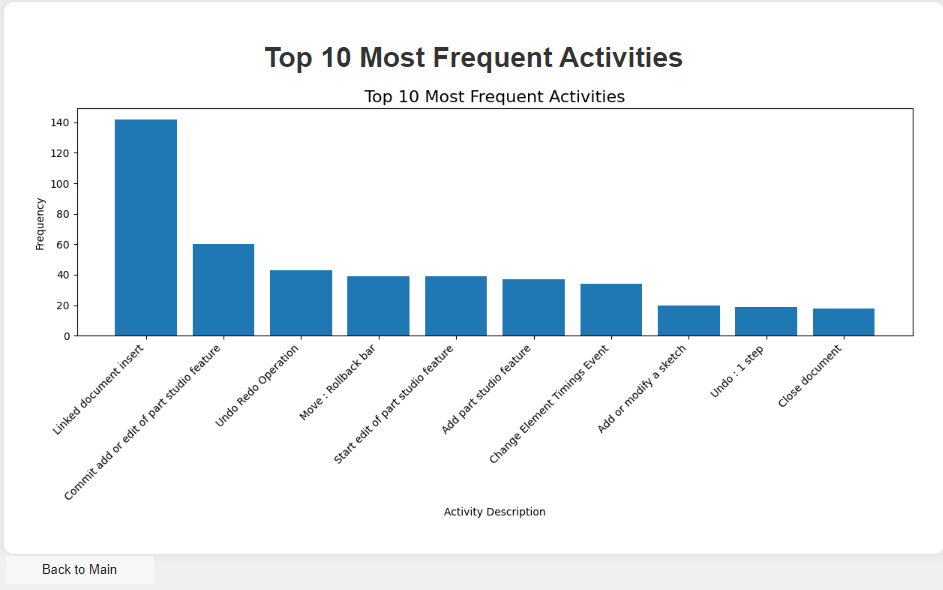
* מסך תרשים פעילות משתמשים. (Chart Page)

יוצג לכל עובד את אחוז הפעילות שהוא עשה עבור כל הפעילות במערכת.



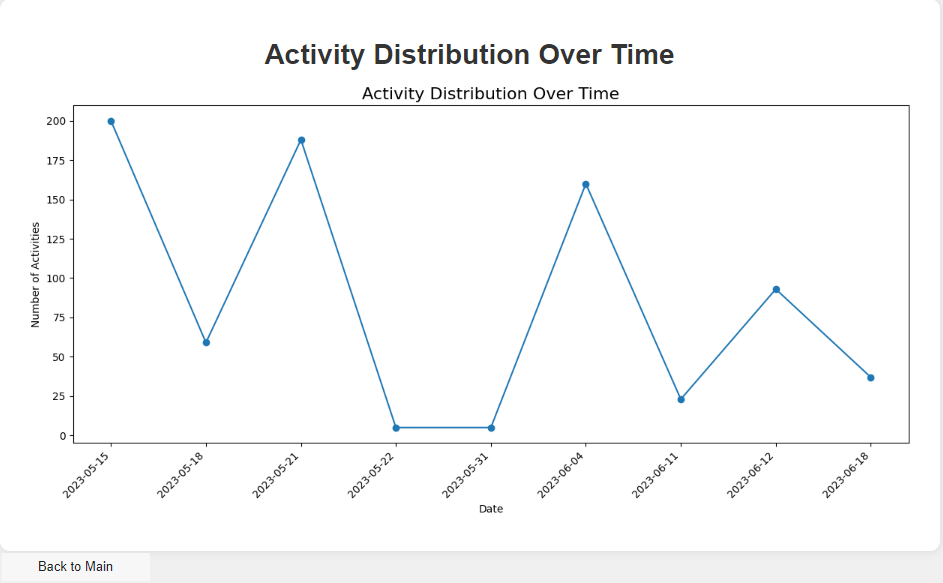
* מסך פעילויות נפוצות ביותר. (Most Frequent Activities Page)

המסך יציג את 10 פעולות הכי נפוצות. הציר X יופיע את שם הפעולה, והציר Y יופיע לכל פעולה כמה פעמים היא נעשתה.



* מסך התפלגות פעילויות לאורך זמן. (Activity Distribution Over Time Page)

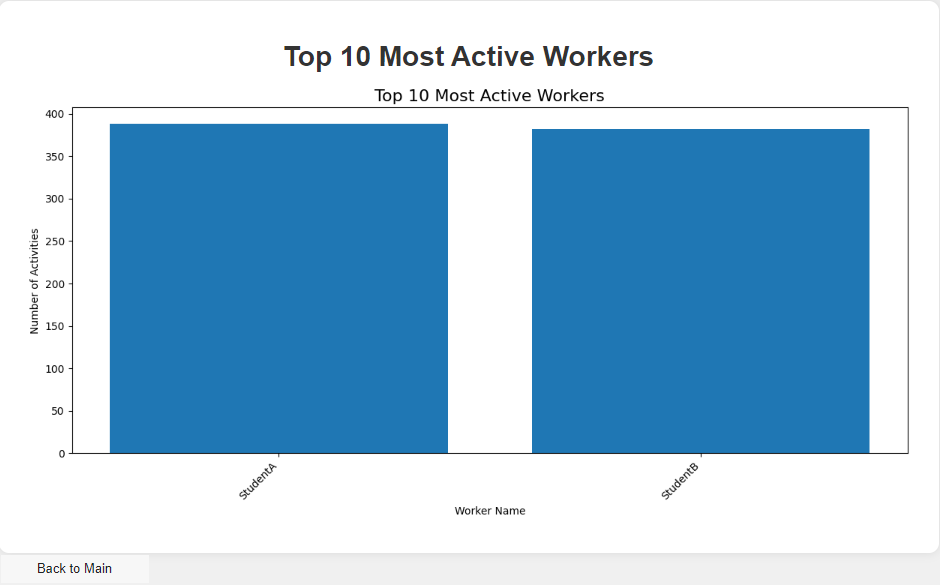
המסך מטרתו היא לספק תמונה ויזואלית של כמות הפעילויות שבוצעו בתקופות זמן שונות, כל שבציר X מציג את הזמן (תאריכים) ובציר Y מציג את מספר הפעילויות.



* מסך שמות העובדים הפעילים ביותר. (Most Frequent Worker Names Page)

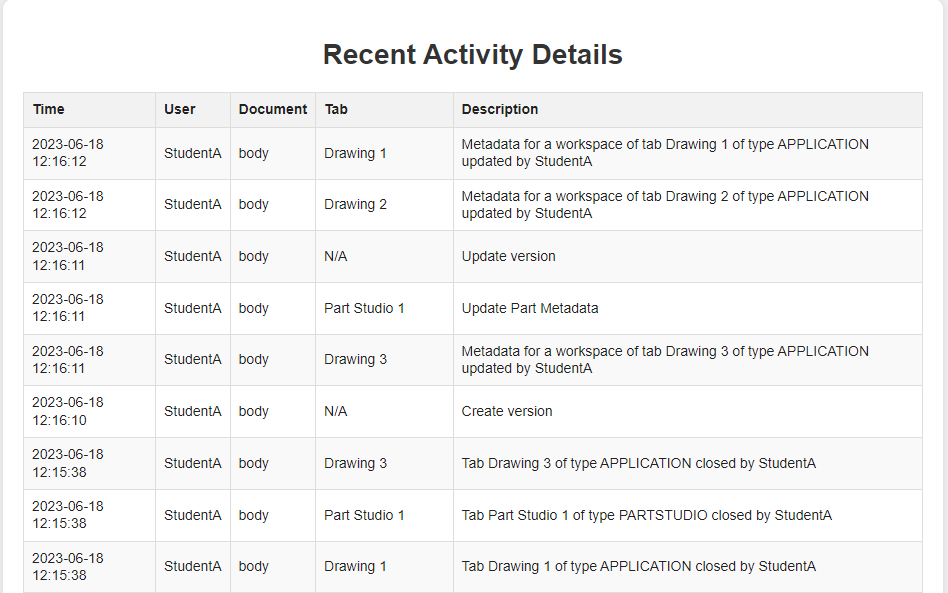
מסך זה מציג את רשימת העובדים המדורגת לפי רמת הפעילות שלהם במערכת. המטרה היא לזהות את העובדים שמשתמשים במערכת בתדירות הגבוהה ביותר.

כך שציר X מופע שם העובד והציר Y מופיע את מספר המשימות.



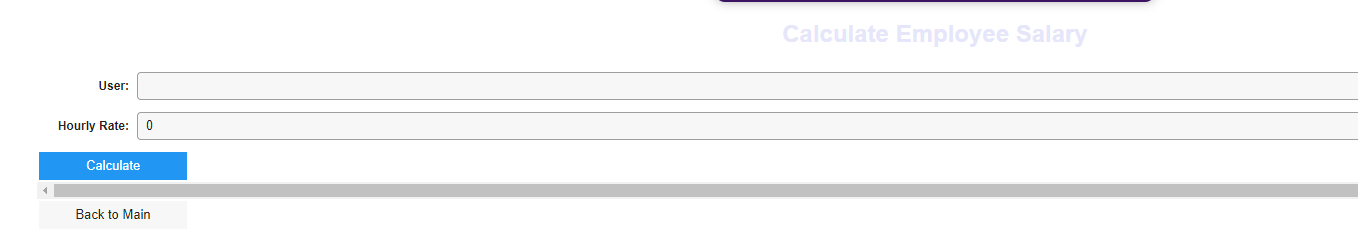
* מסך פרטי פעילות אחרונה. (Recent Activity Details Page)

מסך זה מציג את הפעילויות האחרונות שבוצעו במערכת. המטרה היא לספק תמונה מידית ועדכנית של הפעילות במערכת.

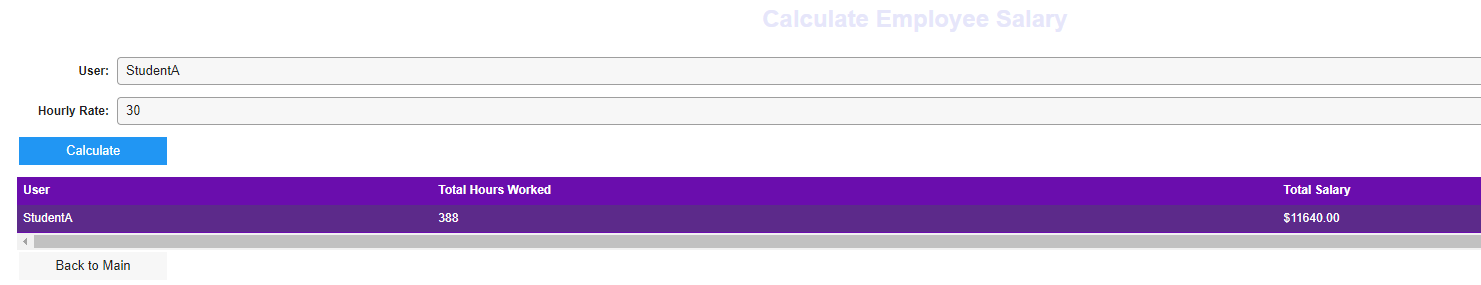


ה. מסך חישוב שכר: (Employee's Salary Page)

מסך זה מאפשר חישוב שכר העובדים על סמך שעות העבודה שלהם במערכת. המטרה היא לספק כלי יעיל לחישוב מהיר ומדויק של שכר העובדים, תוך התבססות על נתוני הפעילות שנאספו במערכת.



לחישוב את השכר , יש לשים את שם העובד והשכר שלו לשעה ולחיצה על הכפתור Calculate, ויוצג את שם העובד עם השעות שהוא עבד , והשכר הסופי שלו, למשל:-



טעות אפשרית:- להשאיר את שם העובד ריק , אז יופיע הטבלה אפסים בשעות ובשכר, ובלי שם העובד.

ו. מסך שעות עבודה: (Student Working Hour Page)

מציג את שעות העבודה של כל עובד.



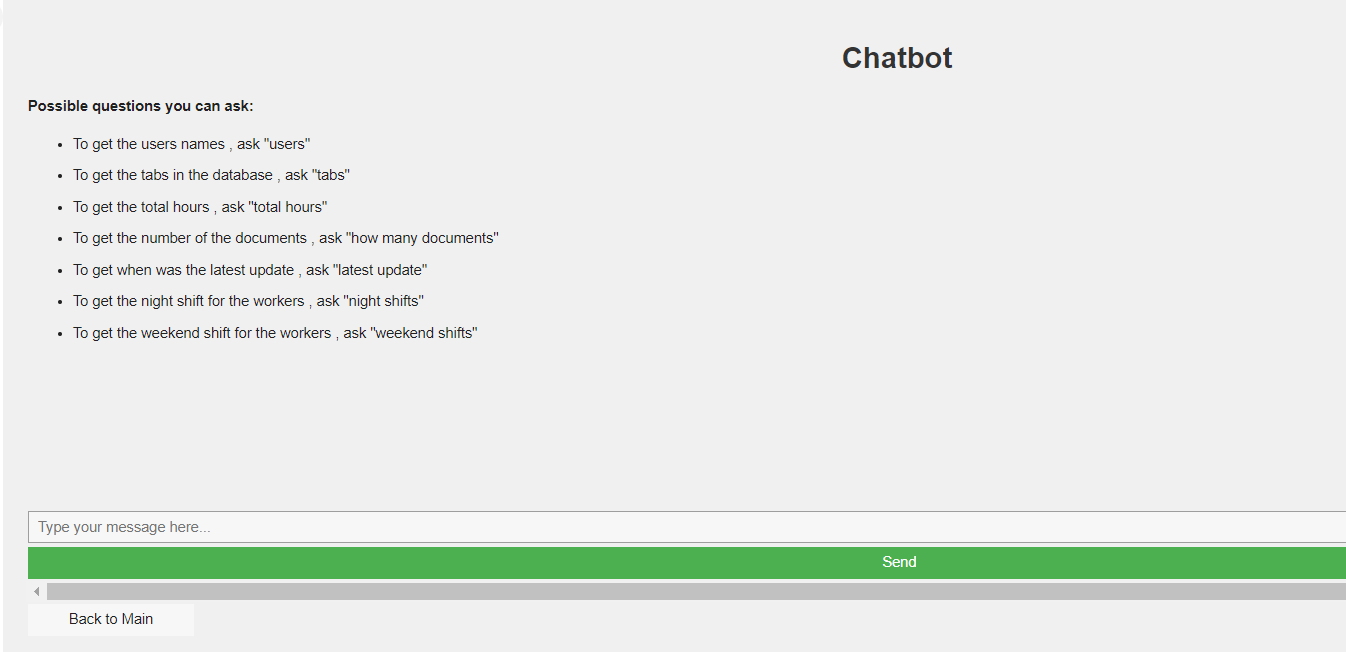
ז. מסך עובדי לילה וסופ"ש: (Night/Weekend Workers Page)

מציג רשימה של עובדים שעבדו בשעות לילה או בסופי שבוע.



ח. מסך Chatbot:- (Go to Chatbot)

מציג כמה שאלות רלוונטיות , ועונה על השאלות בהתאם.



להשתמש במסך הזה , אתה בוחר את השאלה שאתה רוצה , ולשים את השאלה במקום הרלוונטי וללחוץ Send, ויציג לך את התשובה לשאלה שלך.

3. מעברים בין מסכים:-

מהמסך הראשי ניתן לעבור לכל אחד מהמסכים האחרים באמצעות לחיצה על הכפתור המתאים.

מכל מסך ניתן לחזור למסך הראשי באמצעות כפתור Back to Main.

1. יש להכין סרטון קצר של 30-60 שניות, המתאר את השימוש במערכת. הסרטון משמש כ – elevator pitch  למערכת שלכם, כלומר יש לכלול בו הסבר מקצועי ועם זאת שיווקי , המדגיש את האלמנטים המיוחדים של המערכת שבניתם. יש להגיש את הסרטון בפורמט mp4.

הסרטון:

<https://drive.google.com/file/d/1P34_FJdr4UO9yHTxJdswJGE-zcd8aa68/view?usp=sharing>

הוראות הגשה:

1. יש להגיש את התרגיל בצוותים, בתיקיית ה –GIT שלכם, וכן בתיקייית התרגיל ב moodle. **חובה לכלול קישור לתיקיית הגיט בקובץ במוודל.**
2. יש להגיש במוודל קובץ זיפ הכולל קובץ וורד ובו מענה לשאלות, וקישור ל- notebook ובו הקוד שלכם (יש לוודא שהקישור פומבי ונגיש).
3. כותרתו של הקובץ תהיה HW3\_groupName
4. שימו לב כי כל העבודות חייבות להיות שונות זו מזו.אנו מריצות תוכנה לבדיקת עבודות זהות. עבודות שייראו דומות ייפסלו ויינתן עליהן ציון 0.

בהצלחה!

נספח א: אלמנטים שיש להתייחס אליהם לפי מספר הקובץ שקיבלתם

|  |  |
| --- | --- |
| Task | audit no. |
| find a progress pattern of the students’ assignment: how they create new tabs and add features, who is the main contributor to the task? is there a pattern in the days/hours they worked? can you find collaboration signs (e.g. working during the same hours, contributing to the same tab)? | 0 |
| find working hours of each of the students. find occurrences of working during night, weekends and holidays. | 1 |
| find the contribution of each of the users to the progress of the task. if there is one than one student, investigate his or her tasks. | 2 |
| find the contribution of each of the users to the progress of the task | 3 |
| what are the main activities of the student? are they creative (creating documents, deleting, modifying) viewing (e.g. opening and closing tabs frequently) or administrative (importing files, exporting documents etc. )? | 4 |
| how do you suggest to find the best performer? what are the criteria for “best performer” as you see it, and what are the patterns that you expect to see, | 5 |
| using this table, try to find patterns - e.g. create and delete, moving between tabs. | 6 |
| what are the conclusions we can draw from this audit file? can you say if the students are proficient in CAD? can you tell that they collaborate? is there a dominant students, or one who is in the shades? | 7 |