**מבוא למחשוב ענן**

**פרויקט**

206893075גאד טאהא

212101943כרים זידאן

207410523ראמי טאהא

מוחמד חטיב 212704167

חליל מנסור 322965492

אבראהים גראמנה 212186209

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

Link to colab: <https://colab.research.google.com/drive/1vH06qNj_Rp_390fIyf5jf1Oz8ela8PAA?usp=sharing>

סעיף1:

מהות המוצר ומרכיביו – מודולים ופיצ'רים מעניינים. יש להתיחס במיוחד ליעדים שקיבלתם בתרגיל בית 3, לפי מספר ה -JSON שהצוות שלכם עסק בו (נספח א בתרגיל בית 3).

**מהות המוצר ומרכיביו**

**1. מהות המוצר:** המוצר הוא מערכת לניהול ומעקב שעות עבודה של סטודנטים. המערכת אוספת נתונים על שעות העבודה של כל סטודנט ומביאה תובנות על עבודותיהם בשעות לא שגרתיות, כולל לילות, סופי שבוע וחגים.

**2. מרכיבי המערכת:**

**א. מודול ניהול שעות עבודה:**

* **תיאור:** מודול זה אחראי על איסוף, ניהול ושמירה של שעות העבודה של הסטודנטים.
* **פיצ'רים:**
  + **הזנת שעות עבודה:** אפשרות להזנת שעות העבודה ידנית או אוטומטית.
  + **ניהול נתונים:** שמירה ועדכון של שעות העבודה במאגר הנתונים.

**ב. מודול ניתוח שעות עבודה:**

* **תיאור:** מודול זה עוסק בניתוח נתוני שעות העבודה, כולל גילוי שעות עבודה לא שגרתיות.
* **פיצ'רים:**
  + **הפקת דוחות:** יצירת דוחות על שעות העבודה של כל סטודנט.
  + **זיהוי שעות עבודה בלילה:** איתור שעות עבודה שנעשו במהלך הלילה.
  + **זיהוי שעות עבודה בסופי שבוע וחגים:** זיהוי שעות עבודה שנעשו בסופי שבוע ובחגים.

**ג. מודול התראות ודיווחים:**

* **תיאור:** מודול זה מספק התראות ודיווחים למנהלי הפרויקטים ולסטודנטים על שעות עבודה לא שגרתיות.
* **פיצ'רים:**
  + **התראות בזמן אמת:** התראה על שעות עבודה במהלך הלילה, סופי שבוע וחגים.
  + **דיווחים מותאמים אישית:** אפשרות לשליחת דוחות מותאמים אישית למנהלי הפרויקטים ולסטודנטים.

**ד. מודול ממשק משתמש:**

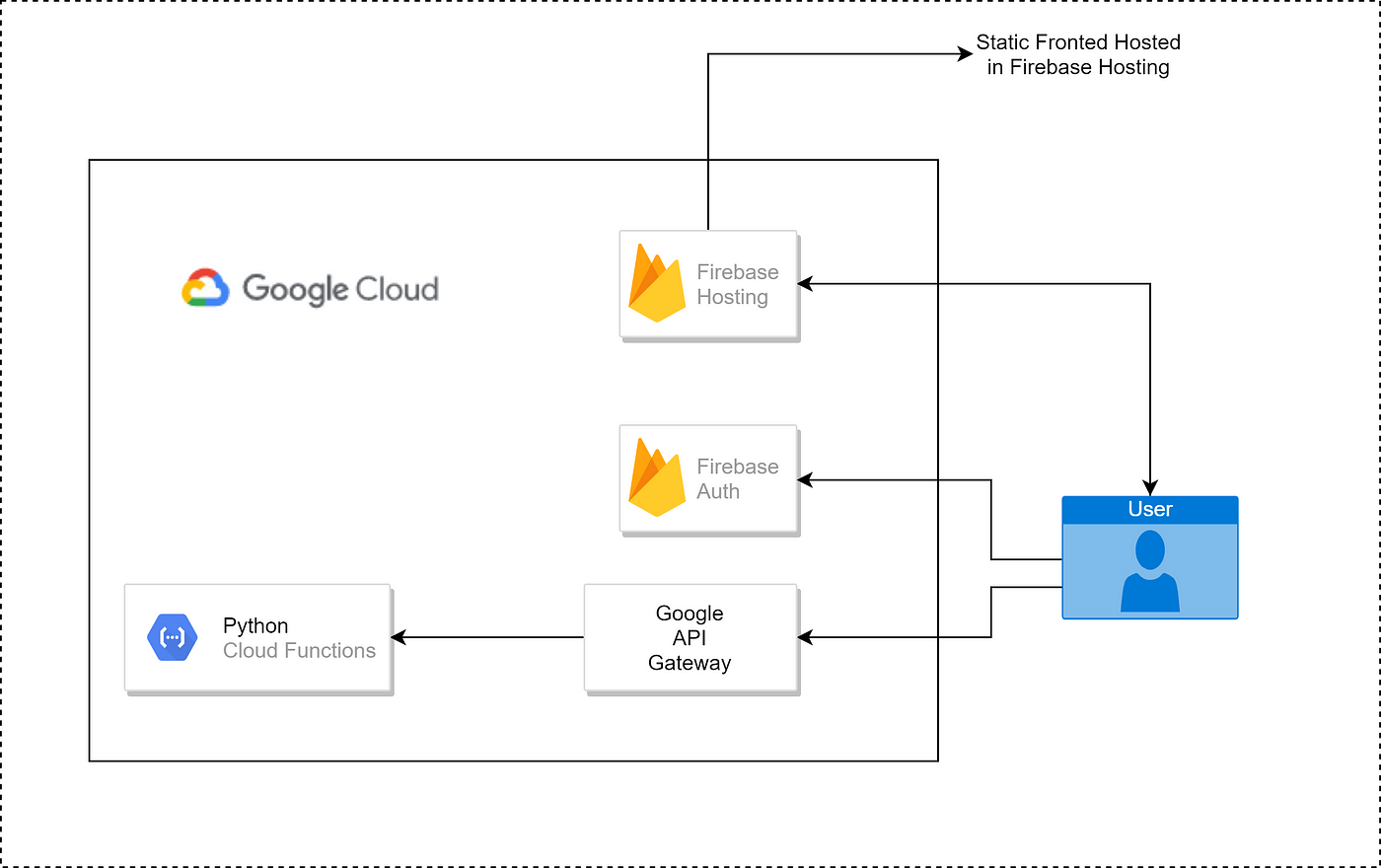
* **תיאור:** מודול זה מספק את הממשק עבור הסטודנטים ומנהלי הפרויקטים לניהול ומעקב שעות העבודה.
* **פיצ'רים:**
  + **לוח מחוונים (Dashboard):** תצוגה גרפית של שעות העבודה, כולל תרשימים וגרפים.
  + **ניהול פרופילים:** אפשרות לעדכון פרטי הסטודנט ונתוני שעות העבודה.
  + **גישה לסטטיסטיקות:** הצגת סטטיסטיקות על שעות העבודה לא שגרתיות.

**3. מודולים ופיצ'רים מעניינים:**

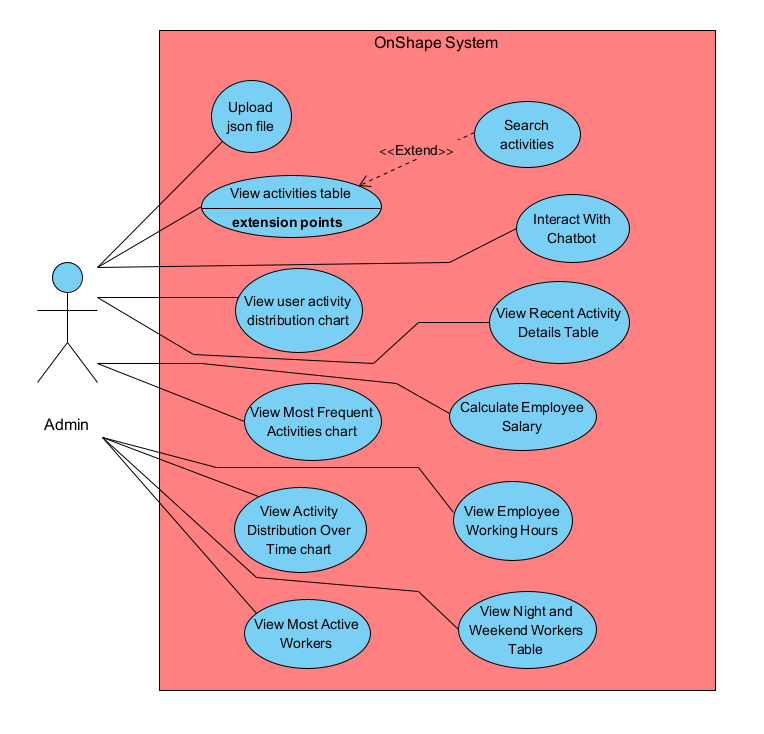
* **פיצ'ר ניתוח מתקדם:** אפשרות לנתח שעות עבודה לפי קריטריונים שונים כגון תדירות, שעות עבודה שבועיות, ועוד.
* **התראות חכמות:** יכולת לשלוח התראות מותאמות אישית על פי התנהגות העבודה של כל סטודנט.
* **ויזואליזציה של נתונים:** הצגת שעות העבודה בצורה גרפית, כגון תרשימים של שעות עבודה בלילה, סופי שבוע וחגים, המאפשרים למנהלי הפרויקטים לזהות מגמות ושינויים בקלות.

סעיף 2 :

Architecture of the program :



סעיף 3 : Use Case Diagram



סעיף 4 :

Non functional requirements :

**Usability NFRs**

1)The system provides a graphical user interface with clearly labeled buttons and forms

2)The interface is designed with a consistent color scheme and layout

3)Navigation between different functions (upload, search, charts) is straightforward

4)Data visualization is used to present information in an easily digestible format

5)The system provides clear feedback for user actions , successful uploads, search results

**Performance**: The system interacts with a Firebase database, suggesting performance considerations.

**Scalability**: The use of Firebase as a backend implies the system is designed to handle potential growth in data and users.

**Security**: Firebase authentication is used, indicating a focus on data security and user authentication.

**Maintainability**: The code is structured into functions, making it easier to maintain and update.

**Compatibility**: The use of web technologies suggests the system is designed to be compatible with various devices and browsers.

**Reliability**: Error handling is implemented in various functions, indicating a focus on system reliability.

**Portability**: The use of Python and web technologies suggests the system can be deployed on different platforms.

סעיף 5:

5. תיק תחזוקה – תיאור של כל הקבצים והאובייקטים המרכזיים, ותיעוד קצר של כל פונקציה 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.

סעיף 6 :

אחרי שבוע 7 והצגת מסכים ראשוניים למערכת שלנו ,בהמלצה של המרצה ואחרי פיידבק על מה שעשו אחרים הפשתנו את רמת העיצוב והתייחסנו לפונקצינאליות של הכפתורים , הפכנו את DASHBOARDהמערכת שלנו כך שיוזר יראה מסך ראשי אחד

שמתייחס אחרי העלאת קובץ גסון\חיבור דרך הדרייב לפעוליות שמתבצעת על ידי כפתורים

**The sum of SUS for 40 users is : 395**

**The average of all SUS is : 77.375**

סעיף 7 :

א.

היה מספר אתגרים שליוו אותנו במהלך העבודה ומהם:

1)**חילוק המשימות בין חברי הצוות היה בעייתי, מאז שאנחנו לא מכירים אחד את השני מקודם ולא יודעים את תחום החוזקה של כל אחד.**

**זה הוביל לאי תיאום ואי הסכם בין חברי הצוות ובאיזה משימה יתעסק כל אחד.**

**2)סיפוק הדרישות של עוזי: במפגש עם עוזי בהרצאה שאלנו מספר שאלות בנוגע לפרוייקט כגון,**

* איזה נתונים חשובים תרצה שיוצגו?ֿֿ

סיום משימה בזמן, לראות שיש עבודת צוות , הצגת נתוני משתמשים בצורה ויזואלית, שני דאשבורדים אחד לעובדים ואחד לבעלי ההרשאות כמו סטייק הולדר

* על מה אני יכול להסתמך בהוצאת נתונים?

תסתמך באופן עיקרי על קובץ ה JSON ואתה יכול גם להשתמש ב Onshape עצמה.

* איך אני אמור לגלות עבודת צוות?

קולאביראשן : אם אני רואה שני אנשים על אותה משימה כאשר השני משנה משהוא שהראשון עשה, onshape יש לה את האפשרות להראות עבודת צוות , ב json אני יכול לראות את מי שייצר את החלק ואם מישהוא אחר שינה בו משהוא אז כנראה שיש עבודה משותפת ולבסוף עוזי אמר שאקח שלושה ארבעה קרטריונים שמהווים עבודת צוות ולחפש אותם בין הנתונים

* איך אני יודע שעובדה מנוסה יותר מאחרים או פחות?

אם תדירות ה import של קבצי stl או עריכת extrudes היא יותר גבוהה מאחרים אז העובד באמת נותן עבודה מאז ושתי הפעולות האלה הדרך כלל מי שעושה אותן הוא מהנדס בכיר בעל נסיון

היה לנו קושי להבין את משמעות התשובות של עוזי וליישמם באופן פונקציונלי במערכת שלנו מאז שהתשובות מבחנתינו היו כלליות מידי ולא ידענו איך זה יהי רלוונטי עבור קבצי נתונים שונים והאם הם יהיו באותו מבנה ונתונים כך שנסיק אותן מסקנות.

3**) לאחרי ביצוע תהליך של divergent thinking הצטבר שיש הרבה רעיונות לחברי הצוות שחלק מהם אכן היה רלוונטי ודורש צומת לב וחלק אחר היה לא רלוונטי, והקושי היה בעצם ההחלטה על איזה רעיונות לוותר ועל איזה להשאיר ולחדד ולעבור הלאה לdivergent thinking ולשכנע את בעל הרעיון שההצעה שלו פחות רלוונטית.**

4**) לקבוע ימי ושעות עבודה על פרוייקט היה גם קושי ששווה להזכיר מאז שחלקנו עובד ולא יכול להיפגש כל יום, היה לנו קשה למצוא שעות פגישה דווקא כי גם הסמסטר מקוצר , עמוס ויש הרבה לחץ בקורסים אחרים שלא לכולנו אותם קוסים וזמנים.**

5**) קושי נוסף שגילינו אותו לאחרי שהראנו את השלד של המסך שלנו לקבוצות אחרות ולמרצה שלנו נעמי, נעמי אמרה לנו שהדף עמוס ואינו מוצג באופן טוב, על פי כך יש לשנות את תצוגת הדף והכפתורים.**

**זה לקח זמן רב להחליט על אופן הצגת הדפים ועל העיצוב.**

**6) אי תיאום בין חברי הצוות: היה אי תיאום משמעותי מה שגרם לנזק בציון שלנו מאז שחילקנו את עצמינו לצוותים כך שכל צוות אחראי על משימה מסויימת ולאחר העדכון כל צוות ישלח בדיסקורד את הגרסה המעודכנת שלו, מה שקרה הוא שהצטבר שהיה אי תיאום כך שכל צוות עבד על קובץ אחר והצוות שאמור להמשיך על הקוד המעודכן שהיה מהצוות שלפניו עבד על קובץ אחר לגמרי מה שגרם לנפילת הקוד במקומות מסויימים למרות שאצל צוות אחד הקוד רץ ללא בעיות.**

**7) קושי נוסף היה , איך נשמור את המידע במסד הנתונים, לאחרי שהתייעצנו יחד וחזרנו על התרגולים ובעזרת כלי בינה מלאכותית הגענו למסקנה שהדאטא יישמר כ ליסט כך שתיקיום האם יהיו בראשית ואז את תוכן השורות שורה שורה ממוספרים.**

ב.

|  |  |  |
| --- | --- | --- |
| **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 , |

סעיף 8:

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

**שקיפות אלגוריתמית**

**1. תיעוד והסבר על האלגוריתמים:**

* **תיעוד בקוד:** כל פונקציה בקוד שלנו מלווה בתיעוד ברור ומפורט, המפרט את מטרת הפונקציה, הקלטים שהיא מקבלת, התהליכים שהיא מבצעת, והפלטים שהיא מחזירה. התיעוד מסביר את אופן פעולתה של כל פונקציה ואילו חישובים או החלטות נעשים בה.
  + דוגמה: פונקציה לחישוב שכר מתועדת עם הסבר על נוסחת החישוב ושימושיה.
* **תיעוד למשתמשים:** בממשק המשתמש ובמסמכי עזרה, אנו מספקים הסברים על כיצד פועלים האלגוריתמים המרכזיים במערכת ואילו החלטות מתקבלות.

**2. הצגת נתונים נאספים:**

* **שקיפות בנתונים:** המשתמשים יכולים לראות את סוגי הנתונים הנאספים עליהם, כגון נתוני שימוש או פרטי פרופיל. אנו מספקים להם מידע ברור על אופן השימוש בנתונים.
  + דוגמה: במסך הגדרות הפרטיות, המשתמשים יכולים לראות את סוגי הנתונים שנאספים ולבצע שינויים בהתאם לצורך.
* **אפשרויות ניהול פרטיות:** המשתמשים יכולים לגשת לנתוניהם האישיים, לערוך או למחוק אותם בהתאם לצורך. התהליך הזה מתועד ומוסבר בצורה ברורה.

**3. מדיניות פרטיות:**

* **מדיניות פרטיות ברורה:** אנו מספקים מדיניות פרטיות מפורטת, המפרטת כיצד הנתונים נאספים, נשמרים ומשמשים. המדיניות נכתבת בשפה פשוטה ומובנת כדי להבטיח שקיפות מלאה.
* **עדכונים ושינויים:** אנו מעדכנים את המשתמשים על שינויים במדיניות הפרטיות ובתהליכי עיבוד הנתונים, ומספקים להם הסברים על השפעת השינויים.

סעיף 9:

|  |  |  |
| --- | --- | --- |
| **הערת משוב** | **האם לדעתכם יש צורך בשינוי במערכת בעקבות ההערה?** | **נימוק** |
| הבהרה לגבי איזה קובץ גייסון הם קיבלו | כן | אם המשתמשים אינם בטוחים איזה קובץ JSON הם קיבלו, זה עלול לגרום לבלבול וטעויות בעיבוד הנתונים. מתן הנחיות ברורות או סימון מתאים של הקבצים יכול לשפר את השימושיות. |
| הצבעים, יותר מדי צבעוניים והכתב מעט נבלע בצבע | לא | עיצוב המערכת מבוסס על צבעים חזקים ותוססים שמסייעים למשוך את תשומת הלב של המשתמשים ולשפר את החוויה החזותית. עם זאת, ניתן להוסיף אפשרות התאמה אישית למשתמשים שירצו לשנות את הצבעים בהתאם להעדפותיהם האישיות |
| chatbot | לא | אם אין בעיות פונקציונליות ברורות או תלונות משמעותיות מצד המשתמשים על הצ'אטבוט, אין צורך להשקיע זמן ומשאבים בשינויו. אפשר להתמקד בשיפורים באזורים אחרים שבהם המשוב היה ברור יותר ודרש שינוי. |
| יש צבעים שלא מתאימים להכל, יש גרפים וטקסט שלא נראה לעין... כמו לבן ואפור | כן | בעיות נראות עם טקסט וגרפים הן בעיות חמורות של שימושיות. הבטחת שכל האלמנטים בממשק יהיו קריאים בקלות, ללא תלות בצבעי הרקע, היא חיונית לתקשורת יעילה של המידע. |
| לא היה כל כך ברור מה הדגש של ה-JSON | כן | אם המשתמשים לא מבינים את הדגש או המטרה של הנתונים בקובץ ה-JSON, הם עלולים לפרש את הנתונים בצורה לא נכונה או להשתמש בהם באופן שגוי. הסברים ברורים או תיעוד נלווה יכולים להבטיח שהמשתמשים יבינו את ההקשר של הנתונים וכיצד להשתמש בהם בצורה נכונה. |
| כדאי להוסיף פעולות ברורות שניתן להבין מסקנות יעילות על משתמשי המערכת ממנה מגיעים הנתונים | כן. | מתן פעולות או כלים ברורים שיאפשרו למשתמשים להסיק תובנות מהנתונים הוא חיוני ליעילות המערכת. הוספת תכונות שיקלו על ניתוח נתוני המשתמשים תשפר מאוד את השימושיות של המערכת. |

סעיף 10:

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

מקורות:

1. נתוני Firebase:

- הנתונים שנעשה בהם שימוש בקוד נאספו ממסד הנתונים של Firebase.

2. כלי AI:

- השתמשנו ב-ChatGPT מבית OpenAI לעזרה ביצירת הסברים על פונקציות, טיוב הקוד, והכוונה כללית. הפרומפטים שנעשו היו בקשות להסביר איך לבנות דפים מסוימים, לכתוב פונקציות ולפרט את הפעולות שנעשות בקוד.

פרומפטים לדוגמה:

- "הסבר לי כיצד ניתן ליצור דף HTML שמציג גרף של פעילויות מחולקות לפי זמן."

- "תאר לי איך אני יכול להוסיף כפתור חזרה לדף הראשי בכל דף HTML שאני יוצר."