**דו"ח מסכם לקבוצת Bear**

[קישור למחברת קולאב](https://colab.research.google.com/drive/150S2V0nbB1XKU4fQTw7HN1hOUB89Vo4Z?usp=sharing)

[קישור לתיקיית גיט](https://github.com/KatrinFrimus/Introduction-to-cloud-computing/tree/main)

1.מהות המוצר הינו:

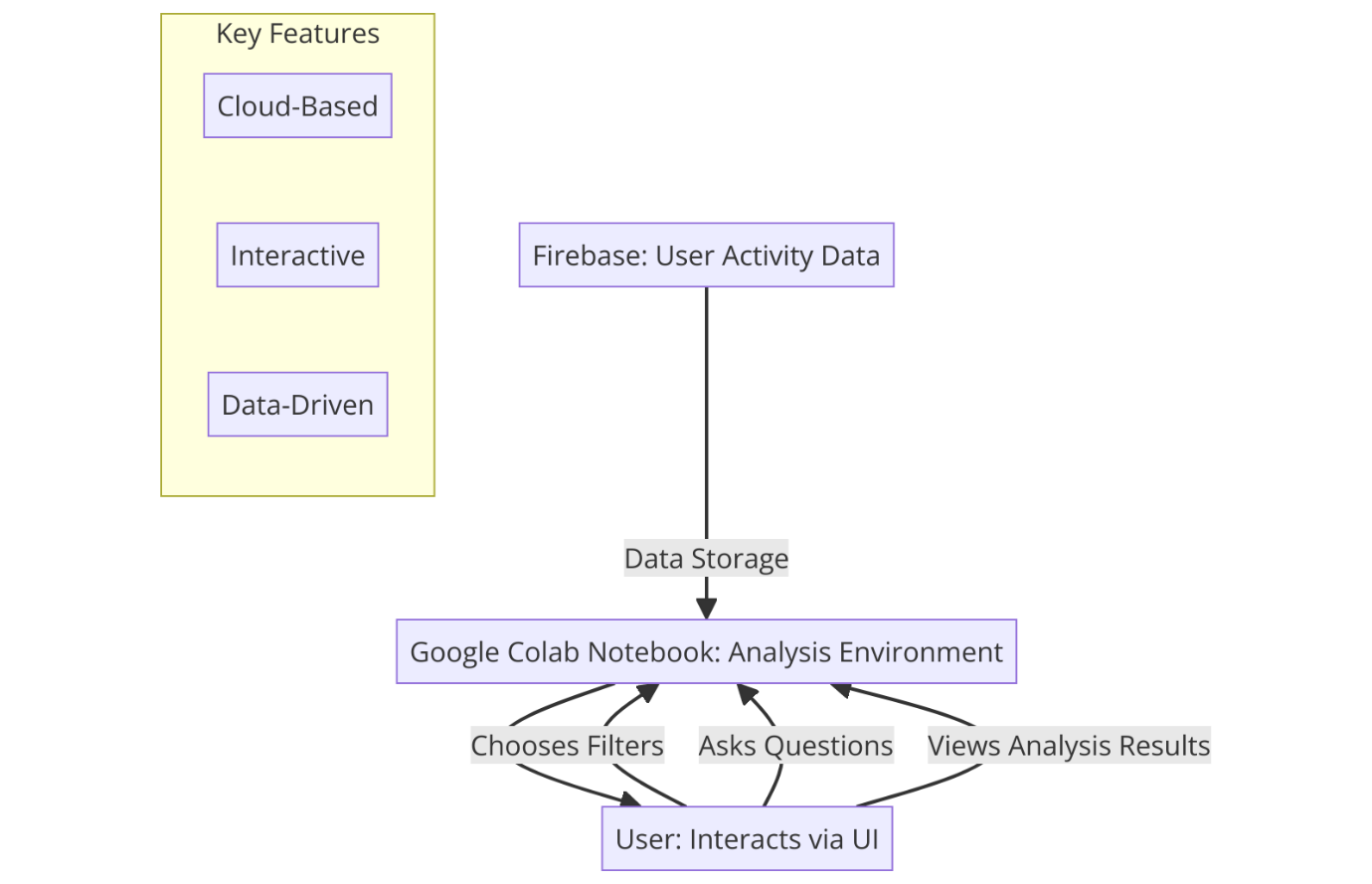
אפליקציית אנליזה לפעילות בתוכנת OnShape האפליקציה נכתבה בשפת פייתון.

**המודולים העיקריים:**

* **Firebase Integration**:המודול הזה אחראי על חיבור האפליקציה למסד הנתונים firebase משם נטענים נתוני הפעילות של משתמשי OnShape
* **:Data Processing and Visualization** מודולים אלו מאפשרים ניתוח ויזואליזציה של הנתונים, כמו יצירת גרפים ותרשימים המציגים את מספר הפעולות לפי פילטרים שונים, שעות עבודה של משתמשים ועוד.
* **Table Generation and PDF Export** :מודולים אלו מאפשרים יצירת טבלאות מנתונים מסוננים ושמירה שלהם כקובץ PDF
* **Chatbot:**מודול זה מאפשר למשתמשים לשאול שאלות על הנתונים באמצעות צ'אטבוט.

**פיצ'רים מעניינים:**

* **פילטרים:** האפליקציה מאפשרת סינון נתונים לפי משתמש, תאריך, סוג פעולה ועוד.
* **יצירת pdf:**אפשרות לשמירת גרפים וטבלאות כקובץ PDF.
* **צ'אטבוט:** ממשק אינטראקטיבי המאפשר למשתמשים לשאול שאלות על הנתונים.
* **זיהוי דפוסים:** האפליקציה תיצור גרף עבור דפוסי עבודה ופעולות שבוצעו בקובץ הניתן.
* **ניתוח שעות עבודה:** הצגת גרפים המציגים את התפלגות שעות העבודה של משתמשים.

2. 

מאפיינים מרכזיים:

**:firebase** מסד הנתונים Firebase משמש לאחסון נתוני הפעילות של משתמשי OnShape .

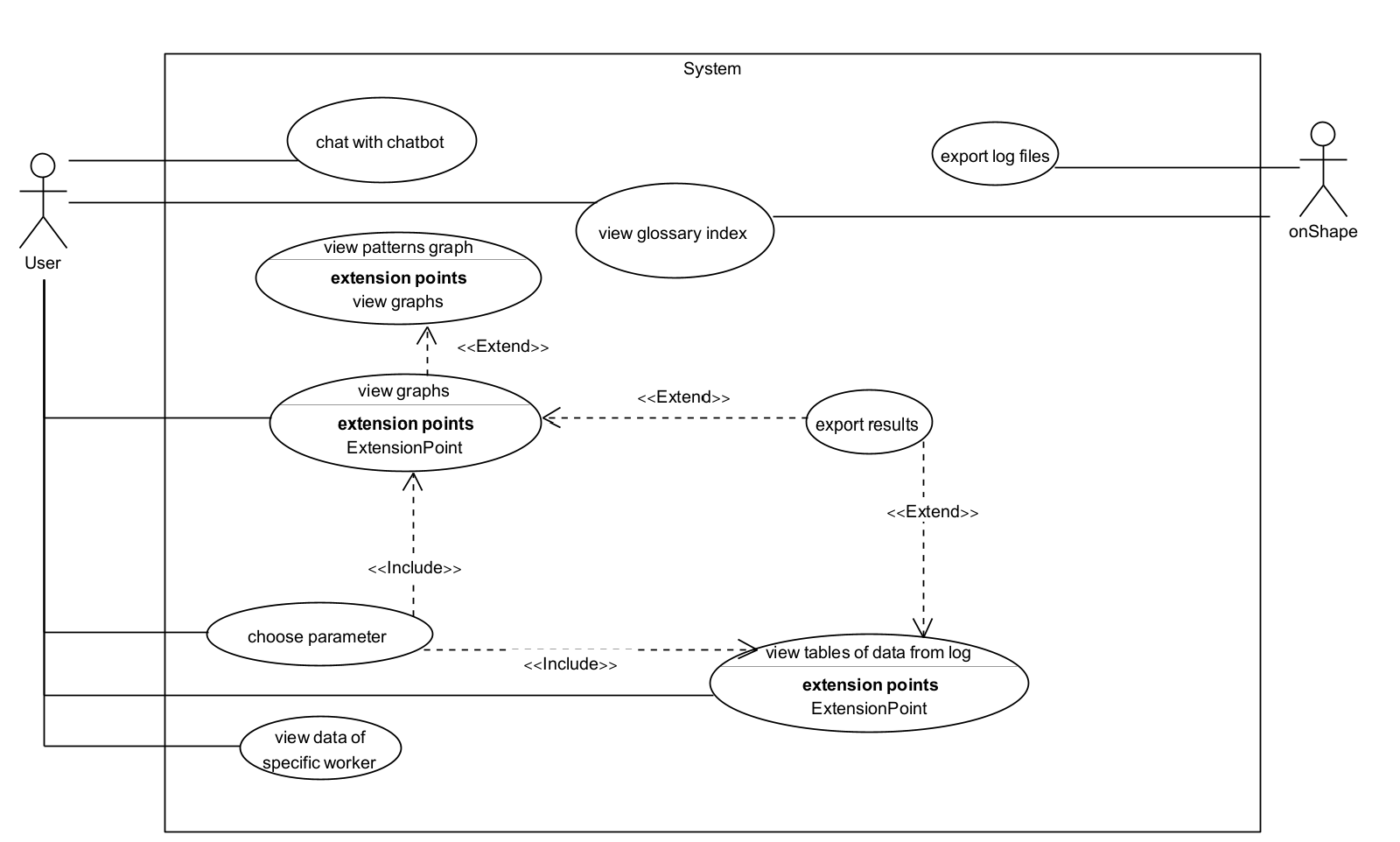
**:google colab notebook** זהו הסביבה בה רץ קוד האנליזה. הקוד כתוב ב-Python וכולל ספריות שונות לניתוח נתונים, יצירת גרפים, ממשק משתמש ועוד.

**:user**המשתמש מקיים אינטראקציה עם המערכת דרך ממשק המשתמש ב-Google Colab בוחר פילטרים, שואל שאלות בצ'אטבוט וצופה בתוצאות האנליזה.

**מאפיינים מרכזיים:**

* **:cloud base**המערכת מבוססת ענן, כך שהנתונים והאנליזה זמינים מכל מקום.
* **:interactive** ממשק המשתמש מאפשר למשתמשים לבחור פילטרים ולשאול שאלות בצורה אינטראקטיבית.
* **:data-driven** המערכת מבוססת על נתונים אמיתיים שנאספים מ-Firebaseומאפשרת תובנות מבוססות נתונים על פעילות משתמשים ב-onshape.

3.usecase



4.דרישות לא פונקציונליות:

* קלות תחזוקה**-**הקוד צריך להיות מתועד היטב, ברור ומסודר, כך שניתן יהיה לשדרג או לתקן אותו במהירות וללא קושי גם על ידי מתכנתים אחרים שלא כתבו את הקוד הראשוני. (Maintenance).
* המערכת תטפל בנתונים גדולים בצורה אמינה ויעילה, תוך שמירה על שלמות הנתונים וללא איבוד מידע, גם כאשר מתרחשות תקלות. (Data Integrity)
* המערכת תהיה זמינה כתוכנת ענן ותספק למשתמשים גישה מכל מקום ובכל זמן, גם בשעות עומס(Usability)
* יתבצע גיבוי שוטף של הנתונים כדי שיהיה שניתן לשחזר מידע במקרה של תקלה או אובדן נתונים. (Backup)
* עיבוד הנתונים מקובץ הלוג יתבצע בשימוש מינימלי במשאבי המערכת(Efficiency)

5. תיק תחזוקה-

קבצים ואובייקטים מרכזיים:

* **Notebook.ipynb-**קובץ Jupyter Notebook המכיל את כל הקוד של האפליקציה, כולל טעינת נתונים, ניתוח, ויזואליזציה וממשק משתמש.
* **-Firebase Database**מסד נתונים בענן המאחסן את נתוני הפעילות של משתמשי OnShape
* **-graph.pdf**קובץ PDF המכיל גרפים שנוצרו על ידי האפליקציה.
* **table.html, table.pdf-**קבצים המכילים טבלאות שנוצרו על ידי האפליקציה.

תיעוד קצר של פונקציות:

**-split\_time(record)** פונקציה זו מקבלת רשומה ממסד הנתונים ומפרידה את השדה 'Time' לשני

שדות: date וtime.

**create\_graph\_pdfs(button)-**פונקציה זו שומרת את הגרף הנוכחי כקובץ PDF בשם 'graph.pdf'.

**create\_pdfs\_table(button)-**פונקציה זו שומרת את הטבלה הנוכחית כקובץ PDF בשם 'table.pdf'

**build\_graph\_by\_num\_of\_operation(change)** -פונקציה זו בונה גרף המציג את מספר הפעולות לפי פילטר שנבחר על ידי המשתמש.

**fetch\_page(url)** -פונקציה זו מורידה את תוכן הדף מהכתובת שניתנה.

**-index\_words(soup)**פונקציה זו יוצרת אינדקס של מילים מתוך תוכן הדף.

**remove\_stop\_words(index)-** פונקציה זו מסירה מילים נפוצות (stop words) מהאינדקס.

**apply\_stemming(index)-** פונקציה זו מבצעת Stemming על המילים באינדקס.

**search(query, index)-** פונקציה זו מחפשת מילים באינדקס.

**search\_engine(url)-** פונקציה זו יוצרת אינדקס מילים מדף אינטרנט.

**print\_glossary()-**פונקציה זו מדפיסה מילון מונחים מאתר OnShape

**update\_value\_options(change)-**פונקציה זו מעדכנת את אפשרויות הבחירה בפילטר לפי המפתח שנבחר.

**display\_filtered\_table(change)-**פונקציה זו מציגה טבלה מסוננת לפי פילטר שנבחר.

**plot\_student\_hours(selected\_student)-**פונקציה זו מציגה גרף של שעות עבודה עבור סטודנט מסוים.

**activity\_hours\_per\_student()-**פונקציה זו מציגה גרף של דפוסי פעילות לפי שעות עבור כל סטודנט.

**overall\_activity\_by\_hours()-**פונקציה זו מציגה גרף של דפוסי פעילות לפי שעות עבור כל הסטודנטים יחד.

**detailed\_categorize\_activity(description)-**פונקציה זו מסווגת פעולות לפי קטגוריות.

**reset\_interface(b)-**פונקציה זו מאפסת את ממשק המשתמש.

**show\_widgets(b)-**פונקציה זו מציגה ווידג'טים שונים לפי בחירת המשתמש.

**chat\_bot()-**פונקציה זו מפעילה צ'אטבוט המאפשר למשתמש לשאול שאלות על הנתונים.

**-get\_students()**מחזירה רשימה של כל התלמידים שעובדים על הפרויקט.

**get\_most\_common\_actions()-** מחזירה את שתי הפעולות הנפוצות ביותר, יחד עם מספר הפעמים שבוצעו.

**get\_actions\_per\_student()-**מחזירה את מספר הפעולות שכל תלמיד ביצע.

**get\_top\_student()** -מחזירה את התלמיד שביצע את המספר הרב ביותר של פעולות.

**get\_shortest\_actions()-**מחזירה את חמש הפעולות הקצרות ביותר, יחד עם משך הזמן שלהן.

**get\_total\_time()-**מחזירה את הזמן הכולל של כל הפעולות שבוצעו.

**-get\_shared\_actions()** מחזירה את חמש הפעולות שבוצעו על ידי יותר מתלמיד אחד, יחד עם מספר הפעמים שבוצעו.

**get\_unique\_actions\_per\_student()-**מחזירה מחזירה רשימה של פעולות שבוצעו על ידי תלמיד אחד בלבד.

**get\_actions\_per\_hour()-**מחזירה את חמש השעות עם מספר הפעולות הרב ביותר.

**get\_night\_actions()-**מחזירה את מספר הפעולות שבוצעו בלילה (בין 00:00 ל-06:00).

6.ביקורת עמיתים-

|  |  |  |
| --- | --- | --- |
| הערת משוב | האם התבצע שינוי במערכת בעקבות ההערה? | נימוק |
| ההתפלגות בגרך יכלה להיות מעט יותר ברורה | כן | כמות העמודות ירדו, ע"י הוספת אפשרות לפילטר על מנת שהגרף לא יהיה עמוס |
| הצגת נתונים לגבי עובד יחיד | כן | נוספה בטבלה אפשרות לפילטר עבור סטודנט ספציפי, כך ניתן לראות את כל הנתונים לגביו. |
| צריך להיות יותר גרפים | לא | בשלב זה התבקשנו ליצר 2-3 מסכים, בהמשך יתווספו עוד דפים עם גרפים נוספים. |
| מסך login | לא | אין צורך במסך כזה במערכת הנוכחית. |

1. הציון ב SUS של המערכת הוא72.5. הציון מראה כי המערכת עונה על ציפיות נוחות שימוש גבוהות, אך עדיין יש מקום לשיפורים כדי להגיע לרמות נוחות שימוש גבוהות יותר.
2. הגדירו 3 מדדים להצלחת המערכת
   * + 1. הגרפים שמציגה המערכת צריכים להיות קלים להבנה גם למשתמשים שאינם מבינים בנתונים , המידע צריך להיות ברור באופן מיידי.
       2. המערכת צריכה להיות גמישה מספיק כדי לאפשר שינויים ושיפורים לאורך זמן.

3.הנתונים שהמערכת מציגה צריכים להיות רלוונטיים ולסייע למנהל או למשתמש לקבל החלטות שללא המערכת היה לוקח לו יותר זמן להחליט אותם בעקבות ניתוח של הקובץ לוג.

7. אתגרים אשר עלו במהלך העבודה –

1. קושי בתיאום זמנים- אנחנו קבוצה של 6 אנשים, במהלך הסמסטר נתקלנו בקושי לתאם פגישות כלל צוותיות-מילואים, עבודה וכו', לכן קבענו להיפגש אחת לשבוע ולחלק את המשימות הצוותיות לקבוצות קטנות יותר.

2.אי שימוש בpython – למדנו את השפה בסמסטר הראשון ומאז לא חזרנו אליה, היה קושי לחזור על הסינטקס של השפה.

3.עומס- שילוב של פרויקטים נוספים במהלך הסמסטר גרמו לעומס -גם בחלוקת הזמנים וגם בחלוקת המטלות.

א.**design thinking**

**divergent thinking:**

|  |  |
| --- | --- |
| רעיונות | שם |
| עבור כל עדכון של נתונים המערכת תשלח התראה למנהל הצוות. | עדכונים |
| המערכת תציג בצורה ויזואלית ע"י טבלאות וגרפים את הנתונים המתבקשים. | טבלאות |
| המערכת תשלב בתוכה בינה מלאכותית כאשר היא תעזור למנהל הצוות לנתח את הנתונים ולהציע לו הצעות לייעול, שיפור שניתן לבצע בפרויקט. | AI |
| המערכת תשמש להתקשרות בזמן אמת בין המנהל לוקחות, כאשר הלקוחות יוכלו בכל זמן לתת שיפורים ומעקב אחר העבודה המתבצעת. | פידבק בזמן אמת |
| המערכת תשלב בתוכה את האופציות לתקשורת בין מנהל הצוות לעובדים שתחתיו באמצעות צ'אט, תגובות על פעולות שבוצעו, שיחת וידאו וכו' | תקשורת בין משתמשי המערכת |
| מנהל הצוות לא יצטרך להיות מחובר באופן קבוע למערכת על מנת להסתכל או לקבל עדכונים על העבודה הנעשית | ממשק חיצוני |
| המערכת תשלח למייל של מנהל הצוות עדכונים של דברים בעלי חשיבות רבה של אותו יום. | אבני דרך |
| מנהל הצוות יוכל לשלוח עבור כל עובד משימות לאותו יום או לזמן עתידי עם תאריך יעד אותו הוא מגדיר. | משימות בזמן אמת |

**convergent thinking**

פתרון בעל הימור בטוח, : הנגשת המידע בצורה טבלאית ונוחה פר משתמש , קבלת מידע כמה מתוך הפרויקט הושלם בהצלחה .(טבלאות)

פתרון המשמעותי ביותר: ממשק אשר יאפשר לה לא לעבוד בצורה יומיומית מול המערכת , יעדכן אותה בדברים משמעותיים שקורים, הממשק יאפשר משוב אוטומטי על תפוקת העובדים.(פידבק ומשימות בזמן אמת)

פתרון משנה כללי משחק. : שילוב של בינה מלאכותית בתוך האפליקציית on shape (בינה מלאכותית)

**ב. משובים מהcode review :**

|  |  |  |
| --- | --- | --- |
| הערת משוב | האם לדעתכם יש צורך בשינוי במערכת בעקבות ההערה? | נימוק |
| יצירת pdf ללא בחירת גרף אפשרית | כן | חסמנו את האופציה לעשות זאת |
| יצירת pdf בטבלה | כן | חסמנו את האופציה לעשות לזאת |

8. שקיפות אלגוריתמית, הקוד שלנו חשוף למשתמש , מתועד ומחולק לפי פונקציות כך שאפשר להבין מה עושה כל תא בקוד, מה הקלט של פונקציה ומה הפלט, כל סוג גרף מיוצג בעזרת כפתור עם שם אינטואיטיבי ומובן מה התוצר בלחיצה כל הכפתור.

השקיפות מבטיחה שהמשתמשים יכולים לדעת אילו נתונים נאספים, כיצד הם מעובדים, ואיך נבנות התובנות שמוצגות להם.

9.משובים:

|  |  |  |
| --- | --- | --- |
| הערת משוב | האם לדעתכם יש צורך בשינוי במערכת בעקבות ההערה? | נימוק |
| גרף מסובך | לא | המערכת שלנו מכילה גם גרפים פשוטים הגרף המסובך הוא במטרה להציג דפוסי פעילות והגרף צריך להכיל המון מידע, הגרף מציג מידע מתקדם למטרות השוואה. |
| צבעים גנריים | לא | המערכת שלנו מכילה צבע אחיד הנוח לעין. |
| אפשרות לבחירה מתוך טווח זמנים רחב יותר | כן | כרגע הקובץ לוג מאוד מצומצם ולכן לא נראה כי צריך שינוי , בפיתוח העתידי וע"פ דרישה אולי נצטרך לעשות לכל גרף גם סינון לפי תקופת זמן. |

10.מקורות-

<https://matplotlib.org>

<https://www.python.org>

<https://www.chatgpt.com> תיקוני באגים-

prompt:

* Explain how to design button using widget?
* Teach me how to sort an array in python?

מצגות מהתרגולים וההרצאות