**מבוא למחשוב ענן – תיק מתכנת**

**קישור GIT :** [**LINK**](https://github.com/AvishayBR/Cloud_Group_Shablool/tree/bd21f421ba7c52b8022c03e34f8ccd3ef174eca5/HW1)



חברי הצוות :

|  |  |  |
| --- | --- | --- |
| 1. | אבישי בר | 316165984 |
| 2. | רון בנדל | 207285099 |
| 3. | תאמר עאמר | 207418583 |
| 4. | בהאלדין סויד | 207659392 |
| 5. | רביע לחאם | 209318419 |

**ארכיטקטורה**

A screenshot of a computer

Description automatically generated

**ניתוח פרויקט למתכנת**

נפרט את ניתוח הפרויקט למתכנת על פי ההיררכיה המוצגת לעיל. נרד מהרמה הגבוהה ביותר לרמה הנמוכה ביותר ועבור כל רמה נסביר את מהות הקובץ, משימתו ונקודות מרכזיות שחשוב להבין בקובץ .

**קובץ setup\_and\_run.py**

משימה: קובץ זה בעצם תפקידו לבצע את ריצת האפליקציה. על מנת לבצע את ריצת האפליקציה יש להתקין את כל הספריות שהאפליקציה נשענת עליהן וגם להגדיר את working\_directory עבור האפליקציה. לאחר שבוצעו פעולות אלה על הקובץ להתחיל את ריצת האפליקציה.

נקודות מרכזיות :

* לפני ביצוע ההתקנות יש להדיר את ה working\_directory בעזרת הפקודה os.chdir
* כלל ההתקנות הנדרשות מופיעות בקובץ ייעודי בשם requirements.txt

**קובץ requirements.txt**

משימה: תפקיד קובץ זה הוא להכיל את כל שמות הספריות אשר עליהן מתבסס הפרויקט. כל ספריה שנדרשת להתקנה תופיע בשורה נפרדת.

נקודות מרכזיות :

* במידה ומתווספות עוד ספריות לפרויקט ניתן להשתמש בפקודה pip freeze > requirement.txt אשר כותבת לקובץ את כלל הספריות הנדרשות כולל הגירסה הנדרשת להרצת הפרויקט.

**קובץ app.py**

משימה: קובץ זה בעצם מריץ את האפליקציה של הפרויקט. ההרצה של האפליקציה מתבססת על תשתית של ngrok אשר מפיקה url שדרכו ניתן לשייט באפליקציה כולה. משימות הקובץ הן: להפיק את הurl באופן תקין, לבנות sidebar שעליו יתבסס הניתוב ולהתחבר לבסיס הנתונים.

נקודות מרכזיות :

* בקוד ישנו אובייקט הנקרא sidebar המגדיר את הניתוב החדש עבור כל דף. בנוסף ישנה פונקציית callback אשר מגדירה לכל ניתוב את הדף המתאים לו.
* בקוד ישנה מחרוזת הנקראת NGROK\_AUTH\_TOKEN אשר מגדירה את הטוקן אשר מופק מהתשתית. יש להגדיר טוקן אישי ספציפי להרצה. ניתן לקבל את הטוקן מהאתר [NGROK](https://ngrok.com/)
* ההתחברות לדאטה בייס נשמרת באובייקט הנקרא FBconn. ההתחברות ייעודית ל url אשר שלח בפונקציה. במידה ומשנים את הדאטה בייס יש לשנות גם את ה url.
* בקוד ישנה קריאה לפונקציה register\_callbacks שעליה נרחיב בהמשך. פונקציה זו נותנת לפונקציות ה callbacks את האובייקט של האפליקציה והאובייקט של החיבור לדאטה בייס על מנת שיוכלו להשתמש באובייקטים אלו מקובץ ה callbacks.

**קובץ callbacks.py**

משימה: קובץ זה מכיל את הפונקציונאליות של כל הכפתורים באפליקציה. במידה וישנו כפתור שאמור לבצע איזשהו אירוע מסוים יהיה לו מימוש בקובץ זה.

נקודות מרכזיות :

* בתחילת הקובץ ישנה פונקציה def register\_callbacks(app, FBconn) שהיא בעצם מבצעת קישור בין האובייקטים של האפליקציה לבין הפונקציונאליות שלהם. בפונקציה זו אין מה לגעת והיא מוגדרת כחלק מהתשתית.
* בפונקציות הפנימיות ישנה פונקציה עבור כל כפתור או אלמנט כזה או אחר. כל פונקציה צריכה לדעת לאיזה אלמנט היא מתייחסת ולכן לפני הפונקציה אנו כותבים את הייחוס לאלמנט. לדוגמה בקוד הבא :
* @app.callback(  
   Output('document-dropdown', 'value'),  
   [Input('select-all-button', 'n\_clicks'),  
   Input('clear-all-button', 'n\_clicks')],  
   [State('document-dropdown', 'options')]  
  )

אנו מתייחסים להיכן יגיע הפלט של הפונקציה (בקוד זה הפלט יהיה לאלמנט 'document-dropdown'

אנו מתייחסים גם לאלמנט אשר מייצר את הטריגר לכניסה לפונקציה (בקוד זה הטירגר הוא לחיצה על הכפתור 'select\clear-all-button' )

אנו מתייחסים למצב כרגע של אלמנטים נוספים וזה מאפשר שימוש של תכונות אלמנטים אלו בקוד ( לעיל אנו רואים כי אנו מקבלים את האופציות האפשריות של ה 'document-dropdown' )

**קובץ data\_cache.py**

משימה: קובץ זה בעצם מממש תבנית עיצוב הנקראת Mediator Pattern. בתבנית עיצוב זו אנו יוצאים מנקודת הנחה שישנם כמה אובייקטים שונים המעוניינים לקרוא ולעדכן אובייקט מסויים. לכן אנו שומרים את האובייקט בקובץ זה וממשמשים פונקציות של עדכון האובייקט וקריאת האובייקט.

נקודות מרכזיות :

* בפונקציות של הקריאה יש להגדיר את האובייקט הנדרש כ global על מנת לא לייצר אובייקט נוסף.
* במידה ומעוניינים באובייקט נוסף יש להוסיף אותו כמשתנה גלובלי לקובץ זה ולמממש עבור פונקציות לקריאה וכתיבה.

**קובץ home\_page.py**

משימה: קובץ זה בעצם מכיל את דף הבית. דף זה הוא הדף הראשון שהמשתמש רואה שהוא פותח את האפליקציה.

נקודות מרכזיות :

* הקובץ משתמש בתמונות אשר מיובאות מהרשת.
* בקובץ ישנן אלמנטים html אשר יש להם id בשביל העיצוב השמור בקובץ styles.css בתוך תקיית assets.

**קובץ about\_page.py**

משימה: קובץ זה מכיל את הדף המנותב כאשר המשתמש לוחץ על הדף about. דף זה מכיל את מידע על האפליקציה על המפתחים שלה ופרטי יצירת קשר.

נקודות מרכזיות :

* הקובץ משתמש בתמונות אשר מיובאות מתקיית assets.
* בקובץ ישנן אלמנטים html אשר יש להם id בשביל העיצוב השמור בקובץ styles.css בתוך תקיית assets.

**קובץ indexing\_utils.py**

משימה: זה הוא קובץ עזר אשר מכיל את רוב הפונקציות הנדרשות על מנת לבצע indexing. על מנת לבצע Indexing עבור כל דף נוכל להשתמש בפונקציות הממומשות בקובץ זה.

נקודות מרכזיות :

* הפונקציה המרכזית בקובץ היא search\_engine . פונקציה זו אחראית לקרוא לשאר הפונקציה על מנת לבצע את המשימה.
* הפונקציה search\_engine מקבלת כפרמטר url כך שנוכל לעשות את הפועלה לכל דף html שרק נבחר.
* ה Indexing מתחשב ב stop\_words בהיבט האלגוריתמיקה.

**קובץ glossary\_index\_page.py**

משימה: קובץ זה מכיל את הדף המנותב כאשר המשתמש לוחץ על הדף glossary\_index. דף זה מקנה למשתמש את היכולת לצפות ב indexing של הדף glossary באפליקציה onshape

נקודות מרכזיות :

* הדף משתמש ב callback על מנת לאפשר את הפונקציונאליות של כפתור החיפוש.
* הדף מציג את התוצאות בצורה טבלאית על בסיס אלמנטית שהתשתית יצרה בעצמה.

**קובץ firebase\_utils.py**

משימה: קובץ זה הוא בעצם קובץ עזר לקשר עם הדאטה בייס. רוב הפעולות אשר נרצה לבצע בדאטה בייס יופיעו בקובץ זה.

נקודות מרכזיות :

* ישנן 3 פונקציות מרכזיות בקובץ אשר מאוד אלמנטריות בשימוש בדאטה בייס :
  + שליפה כל הקבצים ששמורים בדאטה בייס
  + שליפת קובץ ספציפי מהדאטה בייס
  + כתיבה לדאטה בייס

**קובץ setup\_page.py**

משימה: קובץ זה אחראי על דף הטעינה של הג'ייסון. קובץ זה הוא היחיד שיכול לבצע את הכתיבה של הקובץ ג'ייסון שאותו אנו רוצים לפענח לקובץ ה data\_cache.py. הדף מציע שני אופנים לטעינת קובץ ג'ייסון : טעינה ממערכת הקבצים או טעינה מהדאטה בייס. במידה והתבצעה טעינה ממערכת הקבצים הדף מאפשר גם את שמירת הקובץ בדאטה בייס על מנת לאפשר ליוזר להשתמש בקובץ זה בעתיד בצורה קלה ונוחה יותר.

נקודות מרכזיות :

* לאחר טעינת קובץ בהצלחה יוצגו למשתמש הדפים של האנליזה. פונקציונאליות זו מופיע בקובץ ה callbacks בפונקציה update\_sidebar.
* במידה והקובץ שהמשתמש מעוניין לשמור בדאטה בייס כבר שמור בדאטה בייס השמירה תיכשל.
* בכל פעם שהמשתמש ייכנס לדף זה האופציות לצפייה באנליזות הג'ייסון ייעלמו מפני שדף זה ייעודי לשינוי נתוני האנליזה.
* במעמד טעינת הג'ייסון בהצלחה מהדאטה בייס או ממערכת הקבצים הקובץ יוצג ליוזר כטבלה על מנת שיוכל לוודא באיזשהי צורה שזה הוא הקובץ שהתכוון להעלות.

**קובץ graph\_utils.py**

משימה: קובץ זה הוא בעצם קובץ עזר ליצירת גרפים. באנליזה נוצרים הרבה מאוד גרפים ועלינו למצוא את הגרפים באופן שיהיה קל לתחזוק. לכן ישנו קובץ ייעודי המכיל את הפונקציונאליות של הגרפים.

נקודות מרכזיות :

* כמוסכמה, כל פונקציה מקבלת כארגומנט דאטה פריים אשר זה מייצג את המידע שעליו צריך הגרף לבצע את האנליזה.
* הפונקציות כולן מחזירות דאטה פריימים שמהן ניתן לייצר גרף לאחר ביצוע כל האנליזות.
* ישנה הנחת עבודה שאנו מתמודדים עם קבצי ג'ייסון בפורמט מסויים ואחיד ובכך אנו מניחים כי ישנם טורים אשר חייבים להופיע בדאטה פריים. במידה והנחה זו משתנה יש לשנות את הקריאה של טורים ספציפיים בכל פונקציה בקובץ זה.

**קובץ analysis\_page.py**

משימה: זה הוא בעצם קובץ אשר אחראי על הדף של האנליזה הכללית. דף האנליזה הכללית מאפשר למשתמש לבצע פלטור לפי אלמטנטים שונים כמו לדוגמה : טווח תאריכים , מתמשים וכו'. הדף מקנה למתמש את האפשרות לראות את האנליזות כגרפים עבור המידע שהוא בחר.   
אופציה לכניסה לדף זה תופיע רק במידה וקובץ הג'ייסון נטען באופן מוצלח.

נקודות מרכזיות :

* הכפתורים בדף משתמשים בפונקציות callback על מנת לאפשר את הפונקציונאליות שלהם.
* בקריאה לדף האפליקציה דוחפת לאלמנט הדף את הדאטה של הקובץ ג'ייסון הנוכחי הטעון.
* ישנה הנחת עבודה שאנו מתמודדים עם קבצי ג'ייסון בפורמט מסויים ואחיד ובכך אנו מניחים כי ישנם טורים אשר חייבים להופיע בדאטה פריים. במידה והנחה זו משתנה יש לשנות את הקריאה של טורים ספציפיים בכל פונקציה בקובץ זה.
* במידה ומשנים id של אלמנט מסויים דבר זה משפיע על ה callback ויש לשנות זאת גם בקובץ הcallbacks.

**קובץ quality\_page.py**

משימה: זה הוא בעצם קובץ אשר אחראי על הדף של האנליזה של הסטודנטים הטובים ביותר בפרוייקט על פי הג'ייסון. כמו הדך הקודם גם דף זה מאפשר למשתמש לבצע אנליזה לפי פרמטרים שונים. הדף מקנה למשתמש את האפשרות לראות את המידע כגרף וגם מייצר פודיום שבו מסודרים הסטודנטים על פי הביצועים שלהם.   
אופציה לכניסה לדף זה תופיע רק במידה וקובץ הג'ייסון נטען באופן מוצלח.

נקודות מרכזיות :

* הכפתורים בדף משתמשים בפונקציות callback על מנת לאפשר את הפונקציונאליות שלהם.
* בקריאה לדף האפליקציה דוחפת לאלמנט הדף את הדאטה של הקובץ ג'ייסון הנוכחי הטעון.
* ישנה הנחת עבודה שאנו מתמודדים עם קבצי ג'ייסון בפורמט מסויים ואחיד ובכך אנו מניחים כי ישנם טורים אשר חייבים להופיע בדאטה פריים. במידה והנחה זו משתנה יש לשנות את הקריאה של טורים ספציפיים בכל פונקציה בקובץ זה.
* במידה ומשנים id של אלמנט מסויים דבר זה משפיע על ה callback ויש לשנות זאת גם בקובץ הcallbacks.