**מבוא למחשוב ענן - סמסטר אביב התשפ"ה**

**תרגיל בית 2** -– **עבודה בצוותי העבודה**

מועד הגשה: 28.5.25

המשימה בתרגיל זה: בניית מסכים מרכזיים במערכת

שימו לב: למטלה זו שלושה חלקים



קישור ל**Repository**:

[**https://github.com/KfirCohen-PyLab/Cloud-Computing-Elephant-team**](https://github.com/KfirCohen-PyLab/Cloud-Computing-Elephant-team)

קישור למחברת: [**https://colab.research.google.com/drive/1tvVo\_yz0CbM2AkoIV1dEnd--d2a2Hwwv?usp=sharing**](https://colab.research.google.com/drive/1tvVo_yz0CbM2AkoIV1dEnd--d2a2Hwwv?usp=sharing)

הערות:

לכניסה למערכת יש להזין שם משתמש וסיסמה של אחד מהמשתמשים ההבאים:

מנהל מערכת:

Username: admin

Password: 123

עובד כללי:

Username: user

Password: 123

*חלק ראשון : בניית מערכת (10 נקודות)*

יש למנות מהנדס.ת מערכת בכל צוות, אשר יהיה אחראי על הגדרת הדרישות ההנדסיות, ועל הממשק מול החומרה.

*בכל צוות על כל אחד לבחור אחד מהתפקידים הבאים (יש להחליף מתרגיל בית 1 ) (10 נקודות)*

*scrum master -מרכז את העבודה*

*,frontend developer – פיתוח החלק האחראי על הצגה ללקוח.*

*,backend developer – פיתוח מסד הנתונים והעבודה מולו.*

*product manager – ייצוג הלקוח בצוות (בהתאם לחשיבה העיצובית שבוצעה).*

*UI – עיצוב הממשק*

*QA – בדיקות התוכנה*

נא לרשום את שם

הסטודנט.ית בתרגיל זה. על מהנדס.ת המערכת לכתוב כיצד נעשתה חלוקת העבודה מול הצוות, מה היו המשימות של כל חבר צוות, האם היה ממשק בין חברי הצוות, והאם המשימות מולאו:

|  |  |  |
| --- | --- | --- |
| **איטרציה 1** | | |
| **שם חבר הצוות ותפקיד בתרגיל זה** | **משימות שהוקצו** | **משימות שהושלמו** |
| דניאל רוזנצבייג  UI | בניית אינדקס, מסך חיפוש, שיפור מסכים + עיצוב | בניית אינדקס, מסך חיפוש, שיפור מסכים + עיצוב |
| אליאס פרח (System Engineer)  Scrum Master | בניית אינדקס, מסך חיפוש, שיפור מסך גרפים, SUS, שניידרמן | בניית אינדקס, מסך חיפוש, שיפור מסך גרפים, SUS, שניידרמן |
| ניקיטה קונובלנקו  Product Manager | מסך גרפים, פיצ'ר | מסך גרפים, פיצ'ר |
| כפיר כהן  QA | מסך גרפים, ארכיטקטורה | מסך גרפים, ארכיטקטורה |
| עידו בן עמרה  Frontend Developer | מסך חיבור + מנהל, פיצ'ר | מסך חיבור + מנהל, פיצ'ר |
| איתן סוכצ'בסקי  Backend Developer | מסך חיבור + מנהל, היסטוריית חיפוש, דיאלוגים | מסך חיבור + מנהל, היסטוריית חיפוש, דיאלוגים |

בניית אינדקס (20 נקודות)

באיטרציה זו עליכם לבנות את מסד הנתונים שמכיל את האינדקס של המילים המשמעותיות באתר [mqtt.org](http://mqtt.org), הפרוטוקול איתו אנו עובדים לקבלת מידע מהחיישנים.

מבנה האינדקס צריך להיות אחיד לכל הקבוצות , ולכלול **לפחות** את השדות הבאים  **( אין לשנות את שמות השדות!)**:

|  |  |
| --- | --- |
| שם השדה | הסבר |
| term | term |
| DocIDs | רשימת קישורים לדפים המכילים את ה- term, ממוספרים לפי בחירתכם |

משימות:

1. ממשו את האינדקס בקולאב. ממשנו.
2. רשמו בצורה מפורשת את רשימת ה stop words שבחרתם, ונמקו מדוע בחרתם במילים אלו.

|  |  |
| --- | --- |
| מילה | סיבת בחירה |
| a/an | מילת עצירה שכמעט ולא מוסיפה משמעות. |
| the | מילת עצירה. |
| and/or | מילת עצירה. |
| in/on/at | מילת עצירה. |
| to | מילת עצירה. |
| of | מילת עצירה. |
| is | מילת עצירה. |
| with | מילת עצירה. |
| for | מילת עצירה. |
| it | מילת עצירה. |
| as | מילת עצירה. |
| that | מילת עצירה. |
| can | מילת עצירה. |
| by | מילת עצירה. |
| s | אות באנגלית ולא מילה. |
| more | מילת עצירה. |
| because | מילת עצירה. |

רוב המילים נבחרו היות שהיו מובילות באינדקס כאשר הן רק מילות עצירה/אות.

3. ציינו האם השתמשתם ב stem/lemmatization לצורך בניית האינדקס.

השתמשנו ב-stemming.

*חלק שני: בניית מסכים להצגה בכיתה (50 נקודות)*

בחלק זה תכינו 4 מסכים, אותם תציגו לחבריכם בפעילות שתתבצע בכיתה.

המסכים צריכים לכלול (לפחות) (20 נקודות):

מסך מנהל למערכת, מסך שאילתא למנוע החיפוש (הזנה והצגת תוצאות), מסך סטטיסטיקות מעניינות.

בשלב זה נדרש לממש במלואם את בניית מסך המנהל ומסך השאילתא. ניתן את התוצאות להציג כרגע עם data חלקי. מומלץ לממש ככל הניתן גם את מנוע החיפוש , כפי שלמדתם בתרגול 6 😊

את מסך הסטטיסטיקות עליכם לממש בצורה בסיסית. עם זאת כמובן ,שכל תוספת שתחליטו עליה, תוביל להערכה גבוהה יותר של המשימה.

בשבוע ההרצאות של 19-20.5.25 תציגו את המערכות שבניתם. המפגש יתנהל במתכונת סטודיו – כל צוות מגיעה **במלואו** לאחד המועדים עם לפטופ, כל הצותים מציגים במקביל. הסטודנטים מסתובבים בין הצוותים, **מתנסים** במערכת, וממלאים משוב .

ההצגה ומילוי המשוב הם **חובה.**

אנא השתבצו בהקדם לאחת מקבוצות ההרצאה על מנת לוודא שקיים איזון בין הצוותים:

https://docs.google.com/spreadsheets/d/1-grJWvFQGtMkCJ8lg9v9ZSYULBvA4ixaEskBCZglIiE/edit?gid=0#gid=0

לאחר ההצגה תקבלו באופן אנונימי את המשובים של חבריכם, וכן את המשוב שלנו. משימות:

1. התייחסו ל -8 כללי הזהב של שניידרמן (הוצגו בתרגול). כיצד המערכת שלכם מבטאת אותם? (5 נקודות).
2. **עקביות**- למשל על מחיקת משתמש מוצלחת יש הערה בירוק שמקרה על הצלחה של פעולה.
3. **קיצורי דרך עבור משתמשים מנוסים**- אין קיצורי דרך בפרויקט שלנו. כל פעולה צריך לעשות עם עכבר.
4. **משוב ברור ואינפורמטיבי** - עבור פעולת ריענון. יש אייקון של טעינה שמראה למשתמש שהמערכת לא תקועה ושהיא בטעינה.
5. **עיצוב דיאלוגים ברורים**- יש לנו דיאלוגים ברורים, עבור פעולת מחיקה, ישנה הודעה שאומרת שהמשתמש נמחק.
6. **מניעת שגיאות -** המערכת מיידעת את המשתמש במקרה שלא רשם אימייל נכון.
7. **התאוששות -**  אין לנו במערכת אלמנט של התאוששות, אי אפשר לחזור אחורה בפעולות במידה וקורת טעות.
8. **שליטה -**  על פעולות חשובות כמו מחיקת משתמש או קניית פרס במשחק המערכת שואלת האם המשתמש בטוח בהמשך הפעולה.
9. **צמצם את העומס על הזיכרון קצר-הטווח**- אין לנו במערכת אלמנט שמצריך מהמשתמש לזכור.
10. יש להגיש את הטבלה הבאה , תוך התיחסות למשובים שקיבלתם (5 נקודות):

|  |  |  |
| --- | --- | --- |
| הערת משוב | האם התבצע שינוי באפליקציה בעקבות ההערה? | נימוק |
| עיצוב, צבעים וחווית משתמש | כן | שיפרנו את העיצוב על ידי הוספת צבעים והזזנו כמה דברים שיהיה יותר יפה ונוח. |
| דף גרפים עמוס | כן | פיצלנו גרפים וטבלאות לטאבים נפרדים. |
| אין משחקים | כן | הוספנו אפשרות לקבל נקודות. |
| פיצ'ר מעניין | כן | הוספנו חנות לקופונים. |
| חיבור משתמש | לא | לא הייתה בעיה, בטעות עשינו שינוי למשתמש בזמן הסדנה ולא זכרנו. |
| לאפשר לאדמין לראות מי מחובר | לא | לא ראינו סיבה לעשות את זה וזה לא מרגיש שזה מועיל למשהו. |
| כאשר מתחברים רואים לוגין כמה פעמים | כן | תיקנו. |
| אפשרות לבחור גרפים | לא | אנחנו חושבים שיותר נוח שכל המידע מופיע פשוט מופרד בטאבים כדי שיהיה נוח לעובד למצוא דברים מהר. |
| אפשרות להסיר אדמין | כן | האופציה הוסרה. |

1. יש לרשום את ציון ה SUS של המערכת שלכם. מה מעיד הציון?(5 נקודות)

לאחר חישוב הציונים מתוך המשובים שקיבלנו, חישבנו ממוצע שלהם:

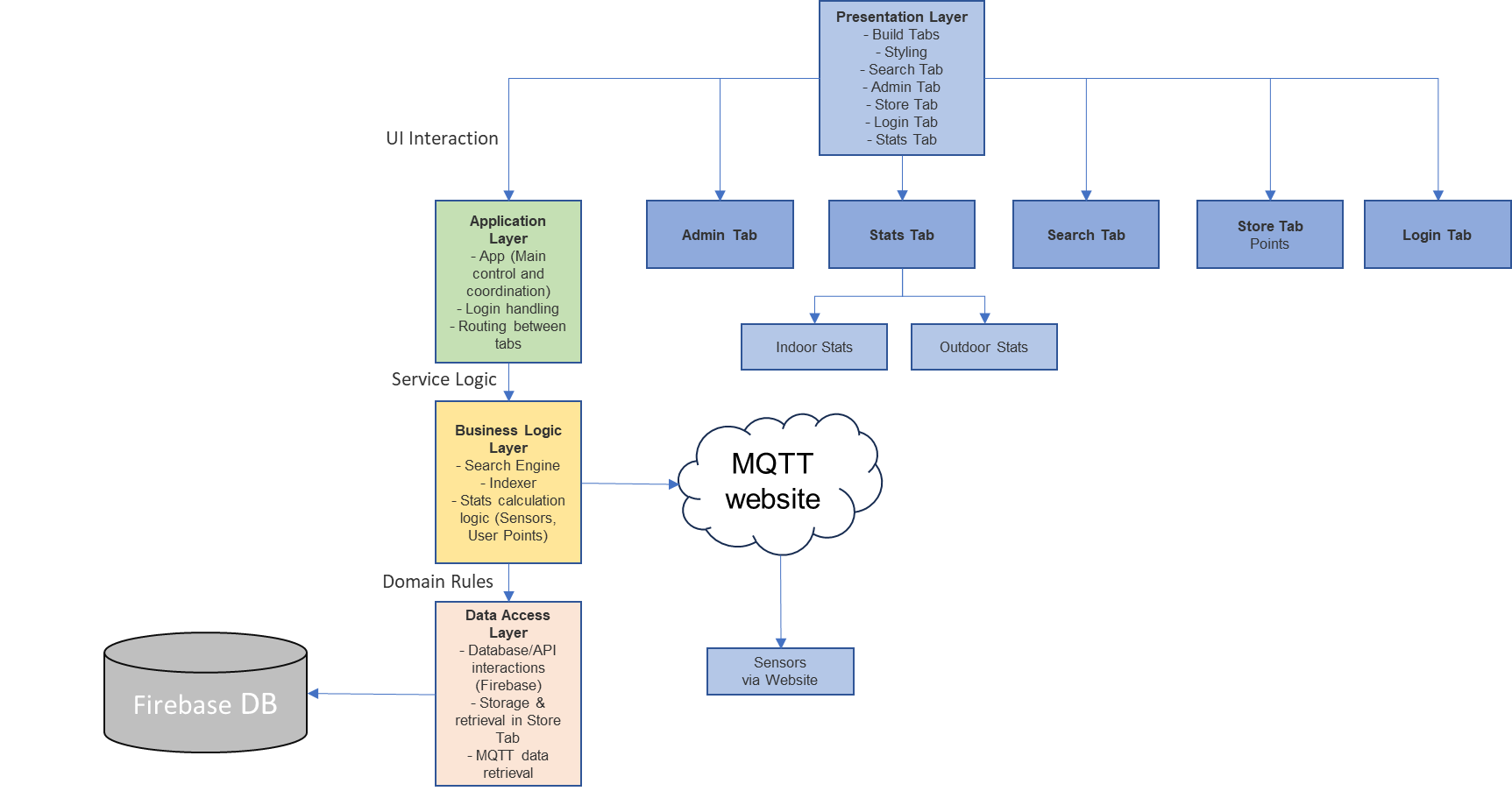
קיבלנו ציון של 87.67. הציון גבוה מאוד ומעיד שהמערכת שלנו בעלת שימושיות מציונת, אך יש מקום קטן לשיפור.

1. הגדירו 3 מדדים להצלחת המערכת (הרצאה 3).(5 נקודות)

* **Service System Availability** - היות שהמערכת הינה דשבורד לעובדים, חשוב שהמערכת תהיה זמינה כמעט תמיד (99.9%).
* **Response Time -** חשוב שזמן התגובה במערכת יהיה ככל שיותר קטן כדי שהמשתמש ירגיש טוב בעת השימוש (100 מילישניות).
* **Security -** במערכת קיימים משתמשים של עובדים וחשוב שהמערכת תהיה מאובטחת כמה שיותר.

1. הציגו דיאגרמת ארכיטקטורה של המערכת שלכם. הסבירו באיזה סוג ארכיטקטורה השתמשתם (הרצאה 7), ופרטו את חלקי הקוד ההמתיחסים לכל חלק בארכיטקטורה.(10 נקודות)

בפרויקט השתמשנו בארכיטקטורת Layered. להלן דיאגרמה:



הסבר על הארכיטקטורה:

#### מחלקה בשכבת המצגת (Presentation Layer) שכבה זו אחראית על ממשק המשתמש, כולל עיצוב הטאבים, סגנון חזותי וטיפול באינטראקציות של המשתמש עם הדשבורד, כגון חיפוש, ניהול משתמשים, רכישת מתנות וניהול סטטיסטיקות.

* **SearchTab**
  + **on\_search\_button\_clicked**: מפעילה חיפוש ומעדכנת את ההיסטוריה כאשר לוחצים על כפתור החיפוש.
  + **build\_search\_results**: מציגה תוצאות חיפוש עם כותרות ותיאורים בהתבסס על השאילתה.
  + **on\_history\_button\_clicked**: מציגה את רשימת ההיסטוריה של החיפושים כאשר לוחצים על כפתור ההיסטוריה.
  + **on\_history\_item\_clicked**: מבצעת חיפוש מחדש מתוך פריט שנבחר בהיסטוריה.
* **AdminTab**
  + **update\_users**: מרעננת את נתוני המשתמשים מ-Firebase.
  + **update\_gifts**: טוענת ומעדכנת נתוני מתנות מ-Firebase.
  + **create\_indexer\_tab**: יוצרת טאב לאינדוקס של אתר אינטרנט ל-Firebase.
  + **create\_terms\_tab**: יוצרת טאב להצגה וניהול של מונחים מאונדקסים.
  + **show\_terms**: מעדכנת את תצוגת המונחים עבור עמוד נבחר.
  + **term\_item**: יוצרת פריט ממשק משתמש עבור מונח בודד עם אפשרות מחיקה.
  + **create\_users\_tab**: מציגה את רשימת המשתמשים הנוכחית.
  + **update\_users\_tab**: מרעננת את תצוגת רשימת המשתמשים.
  + **create\_add\_user\_tab**: מספקת טופס להוספת משתמשים חדשים ל-Firebase.
  + **create\_delete\_user\_tab**: מאפשרת מחיקה של משתמש נבחר מ-Firebase.
  + **create\_edit\_user\_tab**: מאפשרת עריכה של פרטי משתמש ב-Firebase.
  + **create\_manage\_gifts\_tab**: יוצרת טאב לניהול מתנות (הוספה/עריכה/מחיקה).
  + **update\_manage\_gifts\_tab**: מרעננת את ממשק ניהול המתנות.
  + **on\_add\_gift\_clicked**: מוסיפה מתנה חדשה ל-Firebase.
  + **on\_gift\_selected**: מעדכנת שדות קלט כאשר מתנה נבחרת.
  + **on\_edit\_gift\_clicked**: מעדכנת מתנה קיימת ב-Firebase.
  + **on\_delete\_gift\_clicked**: מוחקת מתנה נבחרת מ-Firebase.
* **StoreTab**
  + **update\_gifts\_display**: מעדכנת את תצוגת המתנות הזמינות עם כפתורי קנייה.
  + **on\_buy\_clicked**: מטפלת בתהליך הרכישה, מפחיתה נקודות ומעדכנת ב-Firebase.
* **LoginTab**
  + **get\_user\_data**: שולפת נתוני משתמש מ-Firebase לפי שם משתמש.
  + **authenticate\_user**: מאמתת פרטי משתמש ומעדכנת את הניקוד אם חלף פרק הזמן הנדרש.
  + **on\_login\_clicked**: מטפלת בניסיון התחברות ומעדכנת את מצב האפליקציה.
* **StatsTab**
  + **load\_data**: שולפת ומכינה נתוני חיישנים מ-Firebase.
  + **refresh\_stats**: מרעננת ידנית את תצוגת הסטטיסטיקות.
  + **create\_plot\_base64**: ממירה גרפים של Matplotlib לקידוד base64 לשימוש ב-HTML.
  + **create\_graphs\_and\_stats**: יוצרת גרפים של סדרות זמן והתפלגות עבור נתוני חיישנים.
  + **update\_stats\_display**: מעדכנת את ממשק המשתמש עם הגרפים והטבלאות העדכניים.
* **App**
  + **load\_stats\_tabs**: טוענת נתונים עבור טאבים של סטטיסטיקות פנימיות וחיצוניות.
  + **login\_user**: מעדכנת את ממשק המשתמש והטאבים לאחר התחברות מוצלחת.
  + **logout\_user**: מנקה את סשן המשתמש ומחזירה לטאבים של אורח.
  + **guest\_tabs**: מגדירה טאבים עבור משתמשים אורחים.
  + **admin\_tabs**: מגדירה טאבים עבור משתמשים מנהלים.
  + **worker\_tabs**: מגדירה טאבים עבור משתמשים עובדים.
  + **change\_tab**: מחליפה לטאב מסוים לפי אינדקס.

**מחלקה בשכבת היישום (Application Layer)**שכבה זו מנהלת את זרימת האפליקציה, כולל אימות משתמשים, ניהול סשנים והפעלת ניווט בין טאבים בהתאם לתפקיד המשתמש (אורח, מנהל, עובד).

* **App** (פונקציות זהות לאלו שמעל).
* **LoginTab** (פונקציות זהות לאלו שמעל, כלולות כאן עבור לוגיקת אימות).

**מחלקה בשכבת הלוגיקה העסקית (Business Logic Layer)**שכבה זו מיישמת את הלוגיקה המרכזית של האפליקציה, כולל חיפוש אתרים, אינדוקס של מונחים, וחישוב סטטיסטיקות עבור נתוני חיישנים וניקוד משתמשים.

* **Indexer**
  + **crawl\_website**: סורק אתר אינטרנט באופן רקורסיבי, מאנדקס מונחים ומטא-נתונים.
  + **normalize\_url**: מנרמל כתובות URL על ידי הסרת חלקים וסלאשים מיותרים.
  + **index\_words**: מוציא וסופר מילים מטקסט של דף אינטרנט.
  + **remove\_stop\_words**: מסירה מילות עצירה נפוצות מהאינדקס.
  + **apply\_stemming**: מיישמת תהליך stemming על מילים באינדקס לעקביות.
  + **build\_index**: בונה ומאחסן את האינדקס ואת המטא-נתונים ב-Firebase.
  + **clear\_index**: מוחקת את האינדקס ואת המטא-נתונים מ-Firebase.
* **SearchEngine**
  + **load\_index**: טוענת את האינדקס ואת המטא-נתונים מ-Firebase.
  + **search**: מעבדת שאילתה ומחזירה מזהי מסמכים ממוינים לפי דירוג.
  + **get\_index\_entry**: שולפת או מוצאת את ההתאמה הקרובה ביותר למילה באינדקס.
  + **remove\_stop\_words**: מסירה מילות עצירה משאילתה.
  + **apply\_stemming**: מיישמת stemming על שאילתה.
  + **delete\_term**: מסירה מונח מהאינדקס ב-Firebase.
* **StatsTab** (פונקציות זהות לאלו שמעל, כלולות כאן עבור חישוב סטטיסטי).

#### פונקציות בשכבת הגישה לנתונים (Data Access Layer) שכבה זו מטפלת בתקשורת עם מקורות נתונים חיצוניים, כגון Firebase לאחסון ושליפת נתונים של משתמשים, מתנות, חיישנים ומונחים מאונדקסים, וכן שילוב נתונים מ-MQTT.

* **crawl\_website**, **build\_index**, **clear\_index :indexer** (עבור שליפה ואחסון נתונים).
* **load\_index :SearchEngine** (עבור שליפת נתונים).
* **AdminTab**:
* **update\_users**, **update\_gifts**, **create\_add\_user\_tab**, **create\_delete\_user\_tab**,**create\_edit\_user\_tab**, **create\_manage\_gifts\_tab**,**on\_add\_gift\_clicked**, **on\_gift\_selected**, **on\_edit\_gift\_clicked**,**on\_delete\_gift\_clicked** (עבור ניהול משתמשים ומתנות).
* **update\_gifts\_display**, **on\_buy\_clicked :StoreTab** (עבור נתוני מתנות וניקוד).
* **get\_user\_data**, **authenticate\_user :LoginTab** (עבור גישה לנתוני משתמש).
* **load\_data :StatsTab** (עבור שליפת נתוני חיישנים).

*חלק שלישי : פיצ'ר לבחירתכם* (10 נקודות)

הוסיפו פיצ'ר מעניין למערכת, אשר לא נדרש מכם, לבחירתכם. שימו לב - הכוונה לפיצ'ר פונקציונאלי (ולא עיצובי, או שימוש בשרת כדי להעלות לענן את האתר).

כתבו מספר משפטים להסבר התוספת, וציינו היכן בקוד הוא ממומש. כמו כן הסבירו כיצד הוא מתבטא בחלק המוצג למנהל.

הוספנו חנות לקניית מתנות כאשר העובד מרוויח מספיק נקודות, מתנות אלו מנוהלות על ידי מנהל המערכת שיכול להוסיף ולערוך מתנות. את הנקודות העובדים מקבלים על ידי ביצוע פעולות במערכת כגון כניסה למערכת (בחלק השלישי נוסיף משימות שבהם העובדים יוכלו להרוויח נקודות).

הפיצ'ר ממומש בכמה מקומות בקוד:

* Admin Tab - ניהול של מתנות הנמכרות בחנות
* Store Tab - החנות עצמה
* Login Tab - הענקת נקודות למשתמש בעת התחברות
* App - תצוגה של כמות נקודות שכרגע לעובד יש

החלק מתבטא במסך המנהל על ידי מסך ניהול (CRUD) למתנות המוצגות לעובדים.

**הוראות הגשה:**

1. ש להגיש במודל קובץ זיפ הכולל קובץ וורד ובו מענה לשאלות, וקישור ל- notebook ובו הקוד שלכם (יש לוודא שהקישור פומבי ונגיש). **אין לבצע שינויים במחברת לאחר ההגשה!**

**2.** הקוד צריך לרוץ במלואו מהמחברת בלבד. לא יתקבלו הגשות הכוללות הרצה באתר חיצוני (בפרט slack), או צורך להעלות קבצים למחברת על מנת שתרוץ. הגשות כאלו יקבלו ציון אפס על מרכיב הקוד.

3. יש להגיש את התרגיל בצוותים, בתיקיית ה –GIT שלכם (צרפו קישור), וכן בתיקייית התרגיל ב moodle.כותרתו של הקובץ תהיה HW2\_TEAMNAME.

4. שימו לב כי כל העבודות חייבות להיות שונות זו מזו. עבודות שייראו דומות ייפסלו ויינתן עליהן ציון.

בהצלחה!