פרויקט בהנדסת תוכנה – התמחות בהגנת סייבר   
סמל מקצוע 883589 - 5 יח"ל

**מערכת לרשימת קניות חכמה SUPER LIST**



שם התלמיד: ניב מאיר  
תעודת זהות: 213637861  
שם בית הספר ועיר: מקיף ה' אמירים ראשון לציון.  
שם המנחה: מוטי מתתיהו  
מועד השגת המסמך: 20/05/2021  
  
**תוכן עניינים**

[פרויקט בהנדסת תוכנה – התמחות בהגנת סייבר סמל מקצוע 883589 - 5 יח"ל 1](#_Toc72001247)

[מסמך ייזום 4](#_Toc72001248)

[מבוא 5](#_Toc72001249)

[מה המערכת אמורה לבצע: 5](#_Toc72001250)

[תיאור המוצר: 5](#_Toc72001251)

[אילוצים ודרישות: 5](#_Toc72001252)

[תיחום הפרויקט: 6](#_Toc72001253)

[ארכיטקטורת המערכת: 6](#_Toc72001254)

[התמונות נלקחו ממודלים תלת מימדיים של תוכנת WORD 7](#_Toc72001255)

[סביבת עבודה: 7](#_Toc72001256)

[מסמך אפיון 8](#_Toc72001257)

[פונקציונליות המערכת 8](#_Toc72001258)

[אילוצים עיקריים: 9](#_Toc72001259)

[דרישות המערכת: 9](#_Toc72001260)

[דרישות בסיס הנתונים: 11](#_Toc72001261)

[סביבת עבודה: 11](#_Toc72001262)

[פיתוח עתידי: 11](#_Toc72001263)

[מסמך עיצוב 12](#_Toc72001264)

[טבלת allproducts: 12](#_Toc72001265)

[טבלת mylist: 13](#_Toc72001266)

[טבלת users: 13](#_Toc72001267)

[פירוט השדות 13](#_Toc72001268)

[טבלת allproducts – טבלה המכילה את כל המוצרים בסופר: 13](#_Toc72001269)

[טבלת mylist – טבלה המכילה את כל המוצרים ברשימה של כל לקוח: 14](#_Toc72001270)

[טבלת users – טבלה המכילה את כל הלקוחות שנרשמו לאתר: 14](#_Toc72001271)

[תיאור המודולים 15](#_Toc72001272)

[Super maps: 15](#_Toc72001273)

[Extras: 16](#_Toc72001274)

[Flask: 16](#_Toc72001275)

[DATABASE 17](#_Toc72001276)

[Users: 17](#_Toc72001277)

[Mylist: 17](#_Toc72001278)

[Allproducts 18](#_Toc72001279)

[מודולים מיוחדים 19](#_Toc72001280)

[הגדרות תקשורת – שרת אינטרנט (HTTP): 19](#_Toc72001281)

[דיאגרמת המערכת 19](#_Toc72001282)

[אלגוריתם השרת: 19](#_Toc72001283)

[הדיאגרמה נוצרה על ידי האתר drawio 20](#_Toc72001284)

[אלגוריתם הלקוח : 20](#_Toc72001285)

[הדיאגרמה נוצרה על ידי האתר drawio 21](#_Toc72001286)

[תיאור ממשק המשתמש 22](#_Toc72001287)

[התקנה? 22](#_Toc72001288)

[ממשק המשתמש 22](#_Toc72001289)

[מדריך למשתמש 22](#_Toc72001290)

[יומן רפלקציה 23](#_Toc72001291)

[תיעוד מחקר 24](#_Toc72001292)

[GUI - בחירת המודול לשרת האינטרנט ולמידה עליו – 24](#_Toc72001293)

[רפלקציה 24](#_Toc72001294)

[ביבליוגרפיה: 25](#_Toc72001295)

# למחוק בתוכן עניינים נוצר על ידי #

מסמך ייזום

מבוא

הגדרת הבעיה:  
בעת הכנת רשימת קניות, אנו כותבים אותה לא לפי סדר, אלא מבולגנת, לפי פריט שנזכרים שצריך להוסיף לרשימה. בנוסף בעת הקנייה בסופר, לא תמיד אנחנו יודעים היכן נמצא כל מוצר שאנו מעוניינים לרכוש, מה שגורם לנו לחפש אותו בין המדפים, לעיתים ללכת הלוך וחזור בסופר על מנת למצוא מוצר ספציפי ולבזבז זמן יקר.   
  
מוטיבציה לפיתוח:

לאחר שצפיתי בהוריי מכינים רשימת קניות שבועית, ממיינים ומסדרים אותה לפי נושאים ותחומים עלה לי הרעיון למצוא פתרון שיארגן ויסדר את הקניה בסופר מראש בצורה אוטומטית, בקלות ובאופן עצמאי. לכן החלטתי להקים פרויקט שיקל על הלקוח לבצע את קניותיו באמצעות רשימת קניות מאורגנת ומסודרת מראש המכוונת את הלקוח בקניותיו בסופר.

מה המערכת אמורה לבצע:

הלקוח מכין רשימת קניות ממוחשבת. הוא יוכל להתחבר דיגיטלית לחנות שבה הוא מעוניין לבצע את קניותיו ולקבל רשימת קניות מסודרת לפי מיקום הפריטים בחנות.  
בנוסף קיימת מפה המכוונת את המשתמש בסופר לפי רשימתו, לאיזה מחלקה ומדף עליו לגשת כדי למצוא את המוצר.

תיאור המוצר:  
המטרה המרכזית של הפרויקט היא להקל על הלקוח, ולעזור לו לבצע את קניותיו ללא צורך לבקש את עזרת עובדי החנות.  
  
התרחישים שהמערכת עונה עליהם:  
חיבור דיגיטלי בין הלקוח לאתר החנות  
סידור רשימת הקניות של הלקוח לפי מקום הימצאם של הפריטים במחלקות השונות והמדפים בחנות.

אילוצים ודרישות:  
המערכת צריכה לאפשר ללקוח לבצע חיפוש ולבחור מוצרים ולתת לו את מיקומם בחנות, לסדר את הרשימה באופן אוטומטי.  
המערכת לא תציג מחירים בכדי למנוע ריגול תעשייתי בין רשתות.  
  
המערכת צריכה להיות ידידותית למשתמש, במיוחד לאנשים מבוגרים, שמתקשים בהפעלת מערכות דיגיטליות, המערכת תוצג כאתר אינטרנטי.  
  
אתגרי הפרויקט:

למידת ארכיטקטורת שרת אינטרנט, וספציפית שימוש במודול Flask.

למידת שפות התכנות HTML ו- CSS.

חקירת מסד הנתונים SQLITE.

חיבור של מספר לקוחות לשרת מבלי התנגשויות ביניהם.

שליחת מיילים באופן אוטומטי.

תיחום הפרויקט:  
ביצוע קניות מזון באמצעות מערכת דיגיטלית.  
המערכת תטפל בנושא סידור רשימות המוצרים ומיקומם בחנות, המערכת לא תתעסק במחירים.

ארכיטקטורת המערכת:

המערכת נבנתה בארכיטקטורת שרת אינטרנט. אשר תומכת במספר רב של לקוחות,  
השרת מתקשר בפרוטוקול HTTP (Hypertext Transfer Protocol), פרוטוקול תקשורת שנועד להעברת דפי HTML ואובייקטים שהם מכילים (כמו תמונות, קובצי קול וכו') ברשת האינטרנט.

שרת אינטרנט

לקוח - דפדפן אינטרנט

אינטרנט

התמונות נלקחו ממודלים תלת מימדיים של תוכנת WORD

סביבת עבודה:  
שפת התכנות: PYTHON, HTML, CSS  
סביבת העבודה: PYCHARM  
כלים נדרשים: FLASK

|  |  |
| --- | --- |
| יעד | זמן לביצוע (עד) |
| MVP | 20/11/2020 |
| לימוד FLASK | 15/12/2020 |
| סיום בסיס נתונים | 31/1/2021 |
| בניית האתר | 28/2/2021 |
| הגשת פרויקט סופי | 23/4/2021 |

מסמך אפיון

פונקציונליות המערכת

1.התחברות משתמש קיים למערכת (LOG IN)  
1.1 .אפשרות איפוס סיסמא  
1.2 קבלת קוד למייל לאישור  
2.הרשמה ראשונית למערכת  
3.מסך הבית בו ניתנת אפשרות לעבור לרשימה הקניות או לעמוד המחלקות   
3.1 תצוגה של רשימת הקניות הנוכחית של המשתמש ואפשרות להסרתם   
3.2 תצוגה של כל המחלקות הקיימות בסופר  
3.2.1 תצוגה של כל המוצרים הקיימים במלאי לפי מחלקה או לפי חיפוש ואפשרות להוסיף אותם לרשימת הקניות  
3.3 תצוגה של מסלול ההליכה בסופר לפי הרשימה של המשתמש

אילוצים עיקריים:

המערכת דורשת חיבור לאינטרנט  
המערכת תומכת במספר רב של משתמשים

דרישות המערכת:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | דרישה | קלט | תהליך | פלט | טיפול בשגיאות |
| 1 | התחברות משתמש קיים | מייל וסיסמא | העברת הנתונים לשרת האינטרנט ולבסיס הנתונים לצורך בדיקתם | הלקוח התחבר בהצלחה מעבר למסך הראשי | אם יש חוסר בפרטי המשתמש לא ניתן לעבור הלאה, במידה והמשתמש לא קיים ו/או פרטיו לא נכונים תופיע הועדת שגיאה |
| 1.1 | איפוס סיסמא | אימייל ושם משתמש | בדיקה אם השם משתמש והמייל קיימים ותואמים זה לזה, במידה והם תואמים יקבל המשתמש קוד למייל אותו יצטרך להכניס ולאחר מכן יוכל לשנות את סיסמתו | מסך איפוס סיסמא | במידה ואחד הפרטים שהוכנסו לא קיים בבסיס הנתונים או שהם לא תואמים זה לזה, תוחזר הודעת שגיאה |
| 1.2 | אישור המשתמש | הקוד שמתקבל מהמשתמש | השרת שולח למייל של הלקוח קוד אותו הוא צריך להכניס כדי להיכנס לאתר | מסך להכנסת הקוד | במידה והקוד לא נכון תוחזר הודעת שגיאה ותופיע אפשרות לקבלת קוד אחר |
| 2 | הרשמה למערכת | אימייל, שם משתמש וסיסמא | קבלת הנתונים ושמירתם בבסיס הנתונים | הלקוח נרשם בהצלחה מעבר למסך הראשי | אם יש חוסר בפרטי המשתמש לא ניתן לעבור הלאה, במידה והם שגויים גם לא ניתן לעבור הלאה |
| 3 | מסך ראשי | בחירת המסך הבא | הצגת מסך הבית | מעבר למסך הנבחר (הרשימה שלי או כל המוצרים) |  |
| 3.1 | רשימת הקניות | הסרת מוצר | לקיחת הנתונים על הרשימה של הלקוח מבסיס הנתונים | הצגת הרשימה של הלקוח |  |
| 3.2 | המחלקות בסופר | בחירת המחלקה הרצויה | לקיחת המוצרים מהמחלקה הנבחרת מבסיס הנתונים | הצגת רשימה המכילה את כל המוצרים של המחלקה שנבחרה |  |
| 3.2.1 | כל המוצרים | הוספת מוצר לפי מחלקה נבחרת או לפי חיפוש | בדיקה אם המוצר לא קיים כבר ברשימה של הלקוח והוספתו לרשימה | הצגת כל המוצרים הקיימים ואופציה להוסיף אותם |  |
| 3.3 | מפת המוצרים |  | יוצר מפה חדשה מותאמת למצב הרשימה הקיימת של המשתמש | הצגת מפה המכילה את המסלול של המשתמש בסופר |  |

דרישות בסיס הנתונים:  
  
בבסיס הנתונים ישמרו פרטי הלקוחות הרשומים, כל המוצרים הקיימים ורשימת הלקוח.

סביבת עבודה:

התוכנית נכתבה ב PYTHON 3.8   
ה- GUI נבנה בעזרת שימוש ב HTML,CSS  
השרת נבנה באמצעות FLASK   
בסיס הנתונים נבנה בעזרת SQLITE

פיתוח עתידי:

* שיפור המפה והתאמתה למפות גדולות יותר באמצעות שימוש בדיקסטרה.
* שיפור ופיתוח נראות האתר.
* פיתוח השרת לעבודה לא רק ברשת פרטית (ב- LAN).

מסמך עיצוב

כל אחת מהשדות מייצגת עמודה בכל טבלה בבסיס נתונים:

טבלת allproducts:

|  |  |
| --- | --- |
| שם השדה | סוג השדה |
| product | text |
| locationname | text |
| Locationnum | number |
| shelf | text |

טבלת mylist:

|  |  |
| --- | --- |
| שם השדה | סוג השדה |
| userid | number |
| product | text |
| locationname | text |
| locationnum | number |
| shelf | text |

טבלת users:

|  |  |
| --- | --- |
| שם השדה | סוג השדה |
| userid | number |
| email | text |
| password | text |
| username | text |

פירוט השדות

טבלת allproducts – טבלה המכילה את כל המוצרים בסופר:

* Product – שם המוצר, מוצג כאשר הלקוח מחפש את המוצרים אותם הוא רוצה להוסיף לרשימת הקניות שלו, סוג – מחרוזת.
* Locationname – שם המחלקה בה נמצא המוצר, מוצג כאשר הלקוח מחפש את המוצרים אותם הוא רוצה להוסיף לרשימתו, סוג – מחרוזת.
* Locationnum – מספר המחלקה בה נמצא המוצר, נועד לשימוש אישי של השרת, סוג – מספר שלם.
* Shelf – המדף עליו מונח המוצר בתוך המחלקה, מוצג כאשר הלקוח מחפש את המוצרים אותם הוא רוצה להוסיף לרשימתו, סוג - מחרוזת

טבלת mylist – טבלה המכילה את כל המוצרים ברשימה של כל לקוח:

* Userid – זהות הלקוח, מקשר בין בחירת המוצרים לרשימת הלקוח הרלוונטית, סוג – מספר שלם.
* Product – שם המוצר, מוצג כאשר הלקוח רוצה לשנות את רשימת הקניות שלו, סוג – מחרוזת.
* Locationname – שם המחלקה בה נמצא המוצר, עוזר למשתמש למצוא בקלות את המוצרים אותם הוא מחפש ברמת מחלקה, סוג – מחרוזת.
* Locationnum – מספר המחלקה בה נמצא המוצר, נועד לשימוש אישי של השרת, סוג – מספר שלם.
* Shelf – המדף עליו מונח המוצר בתוך המחלקה, עוזר למשתמש למצוא בקלות את המוצרים אותם הוא מחפש ברמת מדף, סוג- מחרוזת

טבלת users – טבלה המכילה את כל הלקוחות שנרשמו לאתר:

* Userid – זהות הלקוח, סוג – מחרוזת.
* Email – מייל הלקוח, מזהה ייחודי של הלקוח כלומר לא יכול להיות שני לקוחות (משתמשים) עם מייל זהה, בעזרתו ניתן לאפס את הסיסמא ולהתחבר לאתר, סוג – מחרוזת.
* Password – סיסמת הלקוח, חייבת להיות ארוכה מ8 תווים ומכילה גם מספרים וגם אותיות, בעזרתה ניתן להתחבר לאתר, סוג – מחרוזת.
* Username – שם הלקוח, מופיע בעמוד הראשי של האתר לאחר התחברות, סוג – מחרוזת.

תיאור המודולים

Super maps:

מחלקה האחראית על יצירת מפות המשתמש ועדכונם.

\_\_init\_\_ - פעולה בונה היוצרת את מיקומי הנקודות מכוונים את המשתמש במפה שלו.

Make a copy – פעולה המקבלת את שם הקובץ בעבור המשתמש ויוצרת העתק של המפה המקורית.

delete last copy – פעולה המקבלת את שם הקובץ ולפי המשתמש מוחקת את העותק האחרון שלו אם קיים.

draw arrows – פעולה המקבלת את הרשימה של המשתמש ואת שם הקובץ שלו ועורכת את המפה שלו לפי מצב רשימת הקניות הנוכחי שלו.

Extras:

מחלקה האחראית על שליחת מיילים ועל הצפנת מידע.

\_\_init\_\_ - פעולה בונה היוצרת התחברות למייל השולח  
encrypt – פעולה המקבלת טקסט ואחראית על הצפנתו והחזרתו.

Encryptiontest – פעולה המקבלת שני משתמשים, אחד מוצפן ואחד לא ובודקת אם הם זהים.  
codegenerator – פעולה האחראית על יצירת קוד אשר נשלח למשתמש.  
send\_email – פעולה המקבלת כתובת מייל ושולחת אליו הודעה המכילה את הקוד אותו הוא צריך להכניס באתר.

Flask:   
  
מודול האחראי על יצירת שרת אינטרנט מסוג FLASK

render\_template – אחראי על העברת תוכן העמוד מסוג HTML אל המשתמש.  
request – אחראי על בדיקת בקשת המשתמש, לדוגמא האם שלח בקשת GET על מנת לקבל ערך מהשרת, או שלח בקשת POST על מנת להעביר מידע אל השרת וכדומה.  
redirect – אחראי על ניתוב המשתמש בשרת על ידי שינוי כתובת האתר לפי העמוד אליו הוא פונה.

url\_for – כאשר משתמשים בפעולת redirect, יש להשתמש בפעולה זו כדי לומר שאנו מעבירים את המשתמש אל כתובת אינטרנט.

flash – פעולה האחראית על הודעות קופצות במסך המשתמש.

session – פעולה האחראית על הפרדת המשתמשים באתר.

DATABASE

Users:  
  
מחלקה המתחברת לבסיס הנתונים ואחראית על ניהול המידע על המשתמשים.  
  
\_\_INIT\_\_ - פעולה בונה, יוצרת בבסיס הנתונים את טבלת המשתמשים עם השדות: קוד המשתמש, מייל, שמו והסיסמא.

get\_user\_email – פעולה המקבלת את קוד המשתמש (ה-ID שלו) ומחזירה את המייל של המשתמש מן בסיס הנתונים.

insert\_user – פעולה המקבלת את המייל השם והסיסמא של המשתמש ומוסיפה אותו לטבלת המשתמשים.

update\_password – פעולה המקבלת את המייל של המשתמש ואת הסיסמא החדשה שלו, ומשנה את הסיסמא המקורית לסיסמא החדשה.

Mylist:

מחלקה המתחברת לבסיס הנתונים ואחראית על שמירה וניהול המידע על הרשימות שיצרו המשתמשים.  
  
\_\_INIT\_\_ - פעולה בונה, יוצרת בבסיס הנתונים את טבלת הרשימה שלי עם השדות: קוד המשתמש, שם המוצר, שם המחלקה, מספר המחלקה והמדף בו נמצא המוצר.

insert\_product – פעולה המקבלת את המידע על המוצר אותו בחר המשתמש ומכניסה רשומה חדשה לבסיס הנתונים.

product\_isexist – פעולה המקבלת את קוד המשתמש ושם מוצר ובודקת אם בעבור המשתמש המוצר המבוקש כבר קיים בטבלת הרשימה שלי.

delete\_product – פעולה המקבלת את שם המוצר למחיקה ואת קוד המשתמש ומוחקת בעבור המשתמש את המוצר אותו בחר למחוק.

get\_my\_products – פעולה המקבלת את קוד המשתמש ומחזירה את המוצרים של המשתמש בצורה ממוינת לפי המחלקות והמדפים.

Allproducts:

מחלקה המתחברת לבסיס הנתונים ואחראית על שמירת המידע על כל המוצרים בסופר.

\_\_INIT\_\_ - פעולה בונה, יוצרת בבסיס הנתונים את טבלת כל המוצרים, עם השדות: שם המוצר, שם המחלקה, מספר המחלקה והמדף בו נמצא המוצר.

insert\_product – פעולה המקבלת את המידע על המוצר ומכניסה רשומה חדשה לבסיס הנתונים.

product\_isexist – פעולה המקבלת את שם המוצר ובודקת אם המוצר המבוקש כבר קיים בטבלת כל המוצרים.

isempty – פעולה המחזירה אם טבלת כל המוצרים בבסיס הנתונים ריקה מנתונים.

get\_products – פעולה המקבלת מידע מהמשתמש ומחזירה את המוצרים אותם ביקש, לדוגמא אם חיפש מוצר מסוים רק המוצר יחזור, ואם ביקש מחלקה מסוימת כל המוצרים מן המחלקה יוחזרו.

מודולים מיוחדים

Smtplib – מודול האחראי על שליחת מיילים.

bcrypt – מודול האחראי על הצפנה.

random – מודול האחראי על הגרלות.

sqlite3 – מודול האחראי על התחברות לבסיס נתונים וכתיבה בו.

PIL – מודול האחראי על עריכת תמונות.

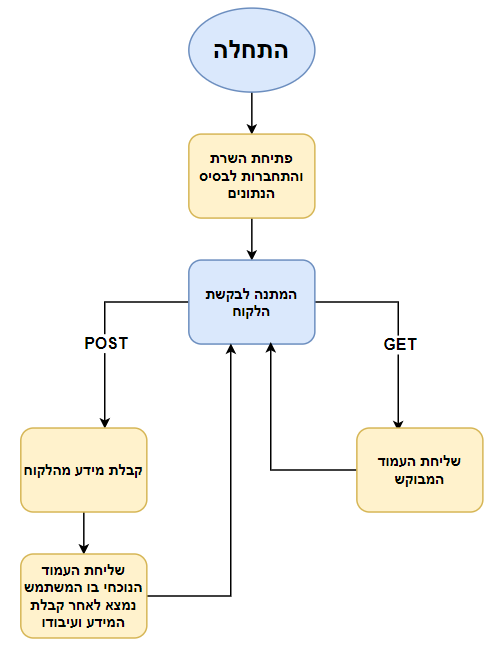
הגדרות תקשורת – שרת אינטרנט (HTTP):

שרת האינטרנט מכיל שתי בקשות, בקשת GET, ובקשת POST.

GET – בקשה בה מבקש הדפדפן מהשרת מידע, במטרה להציג אותו למשתמש.  
POST – בקשה בה מעלה המשתמש מידע אל השרת אשר מעבד את המידע ומחזיר מידע אל הדפדפן.

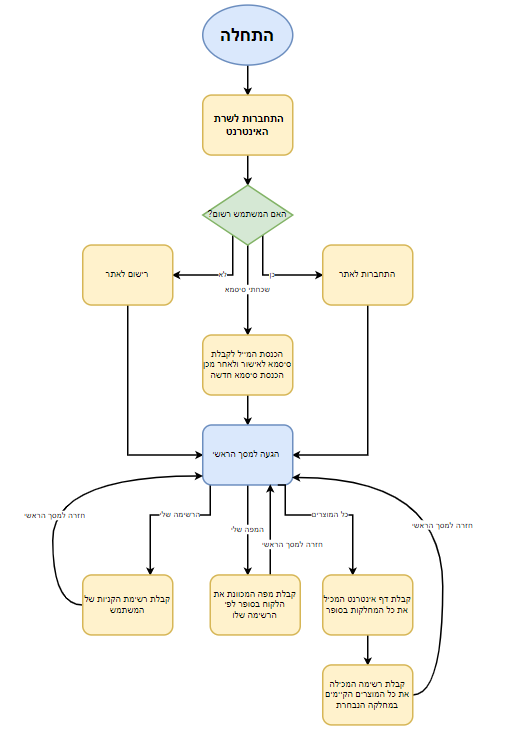
דיאגרמת המערכת

אלגוריתם השרת:



הדיאגרמה נוצרה על ידי האתר drawio

אלגוריתם הלקוח :



הדיאגרמה נוצרה על ידי האתר drawio

תיאור ממשק המשתמש

התקנה?

ממשק המשתמש

#צירוף תמונות של האתר

מדריך למשתמש

יומן רפלקציה

תיעוד מחקר

GUI - בחירת המודול לשרת האינטרנט ולמידה עליו –

במהלך חיפושיי אחר מודול לשרת אינטרנט בPYTHON נתקלתי בשני מודולים עיקריים, FLASK ו- DJANGO. לאחר מחקר על שני המודולים הגעתי למסקנה כי FLASK הינו מודול יותר קל ללמידה עצמית, ומהירה.

לאחר הבחירה בFLASK , חיפשתי מקורות מידע והסברים כדי להבין כיצד הוא עובד ממגוון מקורות כמו YOUTUBE, UDEMY וSTACKOVERFLOW.

שיתוף האתר ברשת –

תחילה היה מוגדר השרת כ- localhost ,אך השרת אמור לתת שירות למספר מחשבים ברשת המקומית, חיפשתי פתרון לכך וגיליתי כי שינוי הIP של הHOST ל- 0.0.0.0 הופך את השרת לגלוי ברשת המקומית.

גישה למספר משתמשים –

כאשר השרת פתוח למספר רב של משתמשים במקביל, הם עלולים "להפריע" האחד לשני בכך שהמידע שלהם עלול לעבור ביניהם שלא לצורך. לאחר חיפושים רבים כיצד להפריד בין הגישות במקביל של המשתמשים, מצאתי כי ישנה פונקציה הנקראת SESSION היוצרת THREAD ייעודי עבור כל משתמש.

בעיית מיון –

הלקוח מוסיף מוצרים לרשימתו בצורה רנדומלית ולא מאורגנת, לפי חוסר שהוא מודע אליו באותו הרגע וכאשר נזכר במוצר שצריך להוסיף לרשימה. הרשימה שהלקוח רואה צריכה להיות ממוינת בכל פעם שהוא צופה בה ולכן הייתי צריך לקחת את המוצרים שהלקוח בחר ולמיין אותם לפי המחלקות שלהם והמדפים בכל מחלקה.  
בהתחלה ניסיתי למיין באופן עצמאי כלומר לגשת אל הרשימה של הלקוח מבסיס הנתונים לשמור את המידע, לאחר מכן לעבור על כל הרשימה לפי מספר המחלקות ובעבור כל מעבר על מחלקה לפי מספר המדפים בה, פעולה זו יצרה התנגשות בין הגישה למידע לבין הוצאתו מהרשימה על מנת להמשיך למיין.  
התחלתי לחקור באינטרנט כיצד ניתן למיין רשימה לפי שני פרמטרים וגיליתי כי ניתן למיין בפעולה בנויה מראש של בסיס הנתונים.

התחלתי דרך בלה בלה ראיתי שזה לא טוב ואז מימשתי בעזרת הבסיס נתונים

רפלקציה

למדתי רבות במהלך העבודה על הפרויקט, חקרתי המון באינטרנט אחר מקורות ידע ללמידה אישית ומידע שימושי שיעזור לי לפתח את הקוד שלי, הבנתי שאני יכול למצוא פתרונות שונים לבעיות שונות. בעזרת אנשים באינטרנט שנתקלו בבעיות דומות קיבלתי רעיון איך לקדם את הפרויקט שלי ולהתאים את הפתרון שלהם לבעיה שלי.

בפרויקט שלי בחרתי לממש שרת אינטרנט, זהו תחום שלא נבחר לביצוע ע"י חבריי לכיתה כך שהזדקקתי לבצע יותר למידה עצמית יחד עם סיועו של המורה שלי.

זהו הפרויקט הגדול הראשון שעשיתי לגמרי בעצמי ואני גאה בתוצאה.

ביבליוגרפיה:

לימוד FLASK -   
  
<https://www.youtube.com/watch?v=zdgYw-3tzfI&list=PLkDSMtLssfHKr9gzWvxe5r-8WgEE4f-ht&index=6>

מסך כניסה (LOGIN) -

<https://www.w3schools.com/howto/tryit.asp?filename=tryhow_css_login_form>

NAV BAR –

<https://www.w3schools.com/howto/howto_js_topnav.asp>

טבלה-  
  
<https://www.w3schools.com/css/css_table.asp>

מחיקת שורה בטבלה:

<https://www.w3schools.com/jsref/met_table_deleterow.asp>

Search box:

<https://www.w3schools.com/howto/howto_css_animated_search.asp>  
  
  
search icon:  
  
<https://icon-library.com/icon/white-search-icon-png-12.html>