

פרויקט גמר לתואר הנדסאי מחשבים



מגישות: רינה אוחיון ונחמי כ"ץ

מגמת הנדסאת תוכנה

שם המנחה: גב' מ. שמעונוביץ

תאריך ההגשה : ספטמבר 19



שמות המגישות: רינה אוחיון ונחמי כ"ץ

חתימה: _____

חתימה: _____

שם המנחה: גב' מ. שמעונוביץ

חתימה: _____

שם רכזת המגמה: גב' ח. ברגמן

חתימה: _____

ספטמבר 2018

אישור הצעת הפרויקט ממה"ט

תודות

רצינו להביע את תודתנו לכל מי שסייע לנו במהלך העבודה על הפרויקט.
לגב 'שמעונוביץ', מנחת הפרויקט שהדריכה אותנו במהלך העבודה ולימדה אותנו את עקרונות השפה והטכנולוגיה שבה עבדנו.
למורה חנה ברגמן, רכזת מגמת מחשבים אשר דאגה שנקבל את כל התנאים שהיינו צריכות בשביל הפרויקט.
לחיה וקסשטוק, המנחה הצמודה שלוותה אותנו לאורך כל הפרויקט בסבלנות וכישרון רב עודדה, הדריכה ונענתה לשאלותינו בכל עת, ועזרה לנו להגיע למוצר המוגמר.
להורים שלנו שעמדו לצדינו, תמכו ועודדו אותנו לאורך הפרויקט.
לחברותינו היקרות שהתעניינו יעצו ועזרו והרבה מעבר.....
וכמובן, אין אנחנו מספיקין להודות .. לריבונו של עולם שראינו הרבה סיעתא דשמיא במהלך העבודה.

הצהרה

הצהרת התלמידים/ות

אנו רינה אוחיון ת.ז.

ונחמי כ"ץ ת.ז.

החתומים/ות מטה, מצהירים/ות בזאת כי כל הפרויקט המוגש בספר זה, הינו פרי עבודתנו בלבד על בסיס הנחייתה של המנחה חיה וקסשטוק ותוך הסתמכות על לימודינו במכללה והמידע אשר רכשנו תוך כדי העבודה.

חתימת התלמידים/ות: רינה אוחיון נחמי כ"ץ

חתימת המנחה:

תוכן העניינים

I.....	פרויקט גמר לתואר הנדסאי מחשבים	
III	אישור הצעת הפרויקט ממה"ט	
1	תודות	
2	הצהרה	
3	תוכן העניינים	
4	מבוא	1.
6	מדריך למתכנת:	2.
6	אסטרטגיות טכנולוגיות:	2.1.
7	תיאור מבנה הפרויקט:	2.2.
8	עקרונות התכנון/ הבניה/ הניתוח:	2.3.
12	תרשימים:	2.4.
16	מבנה נתונים מאוכסנים:	2.5.
19	תוכן הפרויקט:	2.6.
40	מדריך למשתמש: 3.1 הוראות כלליות לשימוש באתר:	3.
42	מסכים:	3.2.
46	סיכום ומסקנות:	4.
47	נספחים:	5.
49	ביבליוגרפיה:	6.

1. מבוא

כאשר אדם מעוניין לארגן קבוצה, בכל נושא וענין – בין קבוצות רכישה לקניית מוצר כלשהוא בהנחה קבוצתית משמעותית. ובין קבוצות חברתיות בהן הוא רוצה למצוא שותפים נוספים לפעילות מסוימת כדוגמת נסיעה לטיול, קבוצת תמיכה. - יש לו עבודה רבה. הוא צריך לדאוג לפרסום מתאים על מנת להגיע לקהלי יעד רבים - בעיתונים, לוחות קיר ומודעות רחוב. הוא זה שמתקשר עם כל המתעניינים בקבוצה, עונה לשאלות ומסביר את אותם פרטים פעמים רבות.

גם לאחר שנמצאו האנשים המתאימים לקבוצה עליו לדאוג לתאום עם כולם, לאיסוף כסף במידה וצריך, וכמובן שכל שינוי בזמן / תאריך גורר מאמץ לתאום נוסף בין כולם. כמו כן כאשר אדם מעוניין להצטרף לקבוצה, עליו לחפש את הקבוצות בכל מקום פרסום אפשרי, עליו לברר מידע עם כל הקבוצות שמצא מי מהן מתאימה לצורך שלו. כמובן שפעולות אלו דורשות עבודה וזמן רב. על מנת להקל ביצירת הקבוצה ובניהולה פתחנו אתר חברתי ידידותי למשתמש המאפשר לפתוח קבוצה ולהצטרף לקבוצה בקלות. וכן לבקש יצירת קבוצה שאינה קיימת.

ממשקים לאורח מזדמן-

האורח יכול לצפות בקבוצות הפתוחות ובפרטים שלהם, לערוך חיפוש מתקדם, ולהשוות בין הקבוצות ע"י סל קבוצות. הוא יכול גם לצפות בבקשות ליצירת קבוצה שבאתר. האורח לא יכול לפתוח קבוצה, להרשם לקבוצה קיימת ולבקש בקשה ליצירת קבוצה חדשה.

ממשקים לחבר באתר-

משתמש הרשום לאתר יכול ליהנות מיצירת קבוצה, הצטרפות לקבוצה קיימת, בקשת קבוצה חדשה.

משתמש יכול לצפות בכל ביצועיו באתר-

קבוצות שפתח – בתור מנהל הקבוצה הוא יכול לראות את מצב הרישום לקבוצה, לעדכן את הפרטים ולשלוח הודעות לנרשמים.

קבוצות שנרשם – יכול לעדכן רישום לקבוצה או לבטל רישום וכן לשלוח הודעה למנהל הקבוצה.

קבוצות שנסגרו בהצלחה – לאחר סגירת הקבוצה יכולים חברי הקבוצה לתקשר ביניהם בצ'אט דרך האתר.

האתר שלנו מאפשר למשתמש לפתוח קבוצה עם קריטריונים מפורטים - ביניהם מספר חברים נדרש ותאריך יעד, לפיהם תדע המערכת מתי לסגור את הקבוצה. בכמה קליקים וללא עלות

הוא מקבל פרסום סוחף בהיקף נרחב וניהול מושלם של הקבוצה – כולל אופציה לתקשורת קלה ונוחה עם הנרשמים לקבוצה באמצעות מייל, ולאחר סגירת הקבוצה - באמצעות צ'אט באתר, הפתוח לכלל החברים הרשומים לקבוצה זו.

גם המשתמש החפץ להירשם לקבוצות ימצא באתר שלנו קבוצות רבות בקלות וללא מאמץ עם אפשרויות חיפוש מתקדמות והשוואה נוחה ומהירה ע"י סל קבוצות, כולל קבלת פרוט של כל קבוצה וקבוצה.

משתמש שלא מצא את הקבוצה שרצה – יכול לבקש פתיחת קבוצה. בקשה זו -מלבד היותה מפורסמת באתר ויכולה להענות מצד משתמשים אחרים - היא גם ובעיקר סוכן חכם – השולח הודעה למשתמש זה עם כל קבוצה שנפתחת באותה קטגוריה / תת קטגוריה וכן מתווך בין בקשות עם קטגוריה זהה וקריטריונים דומים.

כדי לדאוג לאמינות הקבוצות והנרשמים – בכל פתיחת קבוצה או רישום והצטרפות לקבוצה יש לשלם דמי רצינות – סכום סימלי - באמצעות paypal.

2. מדריך למתכנת:

2.1. אסטרטגיות טכנולוגיות:

בכתיבת פרויקט זה שמנו דגש רב על התאמה טכנולוגית מדויקת ושאיפה לטכנולוגיות חדשניות:

צד שרת – server side כתבנו בשפת C# בפלטפורמת Asp.net Web API. תוך שימוש בטכנולוגיות נוספות כמו Entity Framework בסביבת עבודה של Visual Studio 2017. צד לקוח – client side כתבנו בשפות Typescript, Html5, CSS בפלטפורמת Angular7 בסביבת Visual Code.

בסיס הנתונים נכתב בעזרת SQL Server.

את הממשק עצמו עיצבנו בספריות אנגולר חדשניות Angular Material, Material Icon Design. העברת הנתונים בין ה-server וה-client היא ע"י אובייקטים בשפת JSON.

2.2. תיאור מבנה הפרויקט:

2.2.1. מבנה ה- Server side:

ה Solution שלנו כולל 4 פרויקטים בהתאם למודל השכבות:

• DAL

פרויקט זה מורכב ממקור נתונים-מסד הנתונים שלנו, וממערכת תוכנה Entity Framework אשר

תפקידה

לקרוא את המידע הנדרש למערכת, לשמור את העדכונים, להוסיף מידע חדש או למחוק פרטי מידע קיימים.

הפרויקט מובנה בשיטת Entity Framework DataBase First ולכן ה DB נבנה ראשון ועל סמך זה נסנו המחלקות והמאפיינים.

• Entities

שכבה זו אחראית על המרת אובייקטים ממחלקת מיקרוסופט למחלקות שלנו על מנת שנוכל להעביר אותם בין הצד שרת לצד הקלינט. לשם כך יצרנו מחלקות (Class) עבור כל מחלקה בשכבת ה DAL-. ופונקציות המרה דו כיוניות.

• BLL

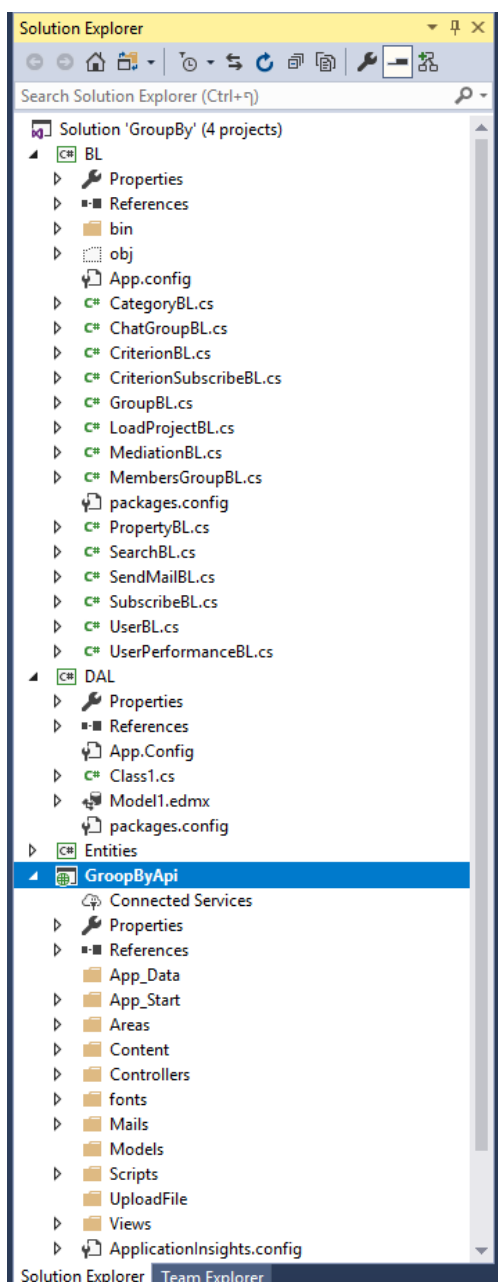
הפרויקט שאחראי על הלוגיקה העסקית של המערכת, עוסקת בעיבוד המידע, חישובים שונים ושליחת המידע לשכבת התצוגה.

בפרויקט זה נממש את הפונקציונליות של המערכת.

מסד הנתונים והממשק משתמש מתקשרים דרך השכבה הזו.

• GroupByApi

פרויקט Web-API עימו מתממשק האתר.



הוא מכיל Controllers: אלו מחלקות המכילות פונקציות המקבלות קריאות Http מפעילות פונקציות בשכבת ה BL ומחזירות את המידע הרצוי לצד הקלינט.

2.2.2. מבנה ה Client side:

מבנה ה-client מורכב משלושה אלמנטים: דפי SCSS דפי HTML ודפי TS. דפי ה-TS כוללים בתוכם פונקציות לוגיות שונות, התחברות לשרתים ולספריות Angular Material, שרותי תשלום PayPal. דפי ה-HTML מהווים את התצוגה. דפי ה-CSS אחראיים על העיצוב והמראה הכללי. ההתחברות לצד שרת ע"י קריאות Http הנמצאות בדפים המכונים Services.

2.3. עקרונות התכנון/ הבניה/ הניתוח:

2.3.1. עקרונות תיאורטיים:

Server

בצד השרת (המכונה גם ss) מתייחסים לפעולות המבוצעות על ידי השרת במערכת יחסי לקוח-שרת ברשת מחשבים. בדרך כלל, שרת היא תכנית מחשב, כגון שרת אינטרנט, שפועל על שרת מרוחק, נגיש מהחשב המקומי של משתמש או תחנת העבודה. פעילות יכולה להתבצע בצד השרת כי הם דורשים גישה למידע או פונקציונלי שאינה זמין בלקוח, או לדרוש התנהגות אופיינית שאינה אמין כאשר הוא נעשה בצד לקוח.

C#

אלגוריתמיקה בסביבת visual studio ושפת c# C# היא שפת תכנות עילית מרובת-פרדיגמות, מונחית עצמים. בעיקרה המשלבת רעיונות כמו טיפוסיות חזקה, אימפרטיביות, הצהרתיות, פונקציונליות, פרוצדורליות וגנריות. השפה פותחה על ידי מיקרוסופט בשנת 2000 כחלק מפרויקט דוט נט ותוקננה בשנים 2005-2006 על ידי ארגון התקינה Ecma כתקן ECMA-334 וארגון התקינה הבינלאומי (איזו) כתקן ISO/IEC 2006:23270.

Entity Framework:

Entity Framework Model הוא כלי חדשני ויעיל של Microsoft על מנת ליצור חיבור למסד נתונים. Entity Framework Model הוא ממפה אובייקטים מקושרים-קרי, בסיס נתונים- המאפשר למפתחי NET. לעבוד עם בסיסי נתונים טבלאיים באמצעות אובייקטים ייעודיים.

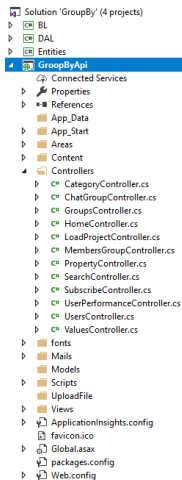
ASP.net Web API

מה זה API?

API הוא קיצור של Interface Program Application. במונחים פשוטים, זהו ממשק של פיסת תוכנה עם העולם החיצוני, הקובע איך ישתמשו בה. אלה החוקים שנקבעו עבור האינטראקציה שלה עם העולם הרחב, אלה יקבעו אילו חלקים של התוכנה יכולות לדבר עם תוכנות אחרות, ואיך היא תגיב.

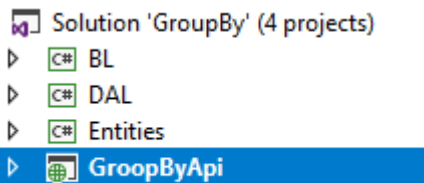
Web API מה זה?

ASP.NET Web API הוא מסגרת המקלה על בניית שירותי HTTP המגיעים למגוון רחב של לקוחות, כולל דפדפנים והתקנים ניידים. API Web .NET Framework הוא פלטפורמה אידיאלית לבניית יישומים ב .NET. בפרויקט שלנו זו שכבת GroupByApi.



הפרדת שכבות:

כל תוכנה/ אתר שנפגוש בעולם מבוססת על ארכיטקטורת שכבות הנקראת בשם Tier- Three



שלוש שכבות במבנה של UI-BL-DAL Application. זוהי תבנית עיצוב בסיסית שמגדירה הפרדת האפליקציה ל: שכבת נתונים, שכבת לוגיקה, ושכבת ממשק משתמש. לתבנית עיצוב זו יתרונות רבים:

1. תחזוקה:

ניתן להחליף או לתקן מימוש פנימי של שכבה אחת בארכיטקטורה בלי לשנות שכבה אחרת.

2. נוחות פיתוח:

אדם אחד עובד על רכיב בתכנה, אדם אחר עובד על רכיב אחר, כל עוד שהחתימות זהות ניתן לשלב כוחות ולייעל זמני פיתוח.

3. בדיקות:

תקלה כלשהיא מבודדת בכל שכבה בנפרד, לדוגמא אם לא קיבלנו רשימת נתונים לתצוגה נבדוק קודם את שלב הביניים (שכבת ה BL) אם הנתונים שם תקינים נדע בוודאות שגם ברמת שכבת ה DAL הנתונים תקינים וכל שנותר הוא לפתור את התקלה ברמת ה UI.

4. שימוש חוזר:

5. נניח שהרכיבים DAL + BL אהובים עלינו ועובדים היטב, ניתן להחליף את

פלטפורמת UI לטכנולוגיה אחרת במינימום מאמץ.

6. אבטחה:

נוח יותר להגדיר Interface API לשכבה מסוימת בלי לחשוף מבני נתונים Logic או Data שלא רלוונטי למשתמש, כמו גם חסימה בפני האקרים (רלוונטי יותר בטכנולוגיות Web) ברמות שונות.

הפרויקט של צד שרת-GroupBy מכיל את שלושת השכבות הנ"ל.

Client

בצד הלקוח מתייחסים לפעולות המבוצעות על ידי הלקוח ביחסי לקוח-שרת ברשת מחשבים. בדרך כלל, הלקוח הוא יישום מחשב, כגון דפדפן אינטרנט, שפועל על המחשב המקומי של משתמש או תחנת עבודה ומתחבר לשרת כנדרש.

פעילות יכולה להתבצע בצד הלקוח כי הם דורשים גישה למידע או פונקציונלי שנגיש בלקוח, אך לא בשרת, כי המשתמש צריך לצפות בהם או לספק קלט, או מכיוון שהשרת חסר כוח העיבוד כדי לבצע את הפעולות במועד לכל הלקוחות שהוא משרת. בנוסף, אם ניתן לבצע פעולות על ידי הלקוח, בלי לשלוח נתונים בשרת, הם יכולים לקחת פחות זמן, להשתמש בפחות רוחב פס, וכרוך בסיכון ביטחוני פחותה.

כאשר השרת משמש נתונים באופן נפוץ, למשל פי פרוטוקולי FTP או HTTP, משתמשים יכולים להיות בחירתם של מספר תוכניות לקוח (רוב הדפדפנים המודרניים יכולים לבקש ולקבל נתונים באמצעות שני פרוטוקולים אלה). במקרה של יישומים מיוחדים נוסף, מתכנתים יכולים לכתוב פרוטוקול שלהם שרת, הלקוח, והתקשורת, שניתן להשתמש בהם רק אחד עם השני. תוכניות הפועלות על המחשב המקומי של משתמש מבלי שולח או מקבלים נתונים דרך רשת אינן לקוחות נחשבים, ופעולות בצד הלקוח כך הפעולות של תוכניות כאלה לא תיחשבנה.

Angular

Angular היא תשתית תוכנה בקוד פתוח ליישומי רשת המתחזקת על ידי גוגל ועל ידי קהילה של מפתחים רבים וחברות, לצורך פתירת אתגרים רבים איתם נתקלים בפיתוח יישומי דף-יחיד. מטרתה היא פשוט הפיתוח והבדיקות של יישומים כאלו באמצעות תשתית תוכנה לארכיטקטורות צד לקוח כמו MVC או MVVM, יחד עם רכיבים בהם משתמשים בדרך כלל ביישומי אינטרנט עשירים.

ספריית Angular עובדת על ידי קריאת דף ה-HTML, שאל התגיות שבו נוספו תכונות נוספות. Angular יפרש את התכונות הללו כהנחיות לקשר את אזורי הקלט או הפלט בדף למודל שמיוצג על ידי משתני Javascript האלו ניתנים לשינוי על ידי קוד, וניתן לגשת אליהם בצורה סטטית, או בצורה דינמית בעזרת JSON.

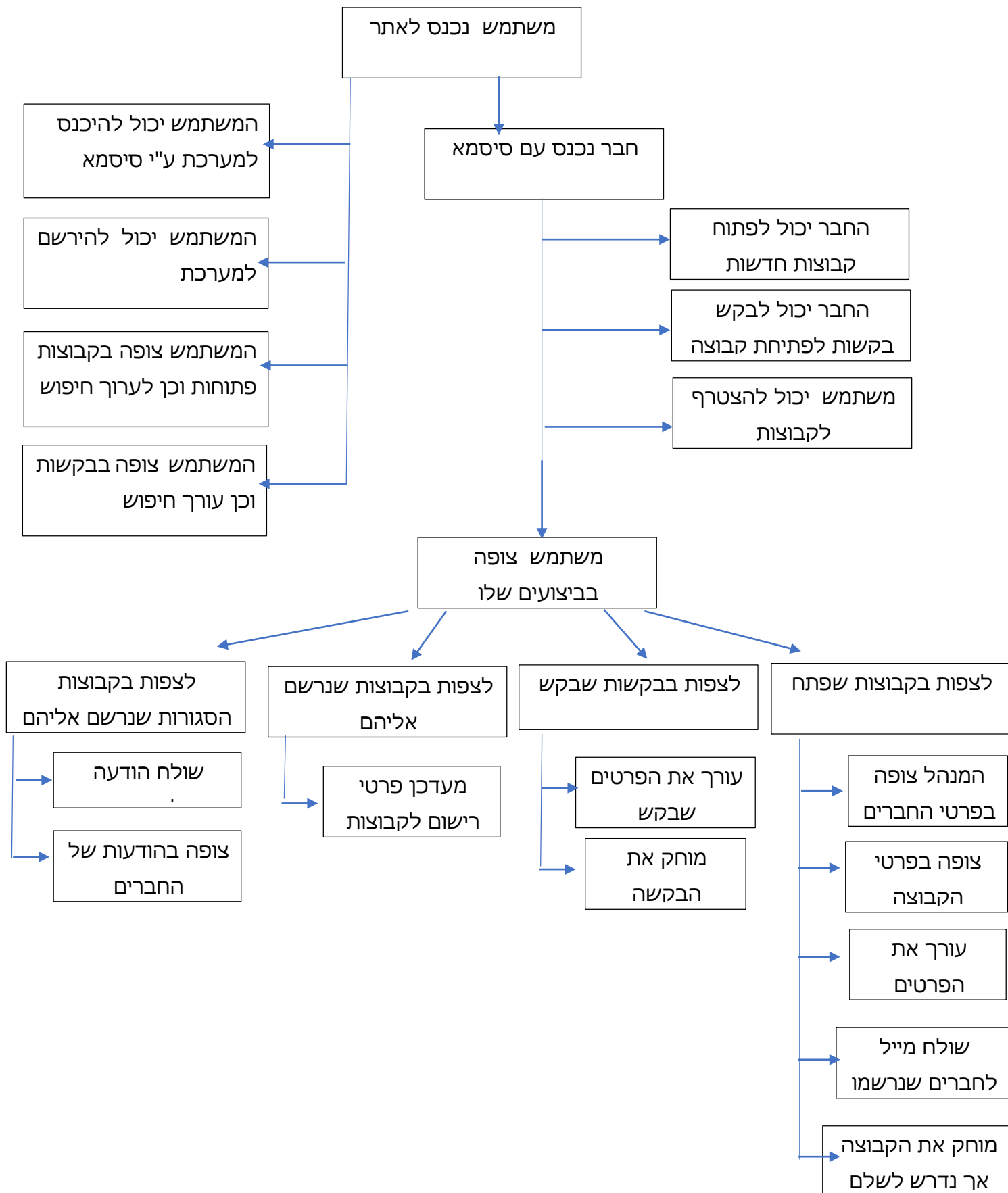
Typescript

היא שפת תכנות חנימית ומבוססת קוד פתוח המפותחת ומתוחזקת על ידי מיקרוסופט. היא מכילה את קבוצת כל פקודות ותחביר JavaScript הפופולרית, כלומר כל קוד JavaScript הוא גם קוד Typescript תקין, ומוסיפה עליה טיפוסים סטטיים ותכנות מונחה עצמים מבוסס מחלקות. בין התכונות שהיא מוסיפה: static typing, תמיכה במחלקות ותמיכה במודולים ובדקורטורים.

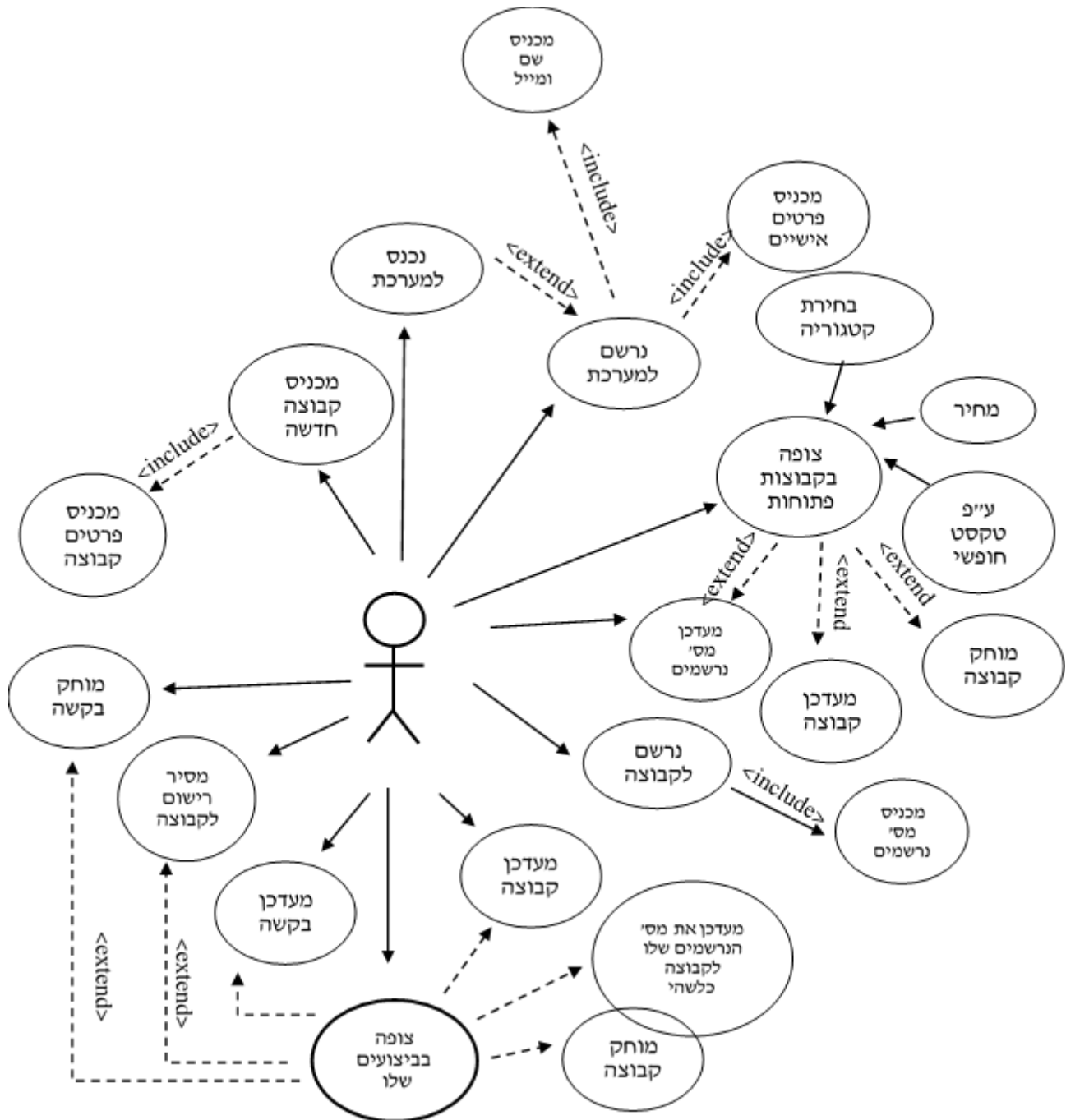
Typescript היא סופר-סט (superset) של Javascript כלומר, הקוד מבוסס על Javascript וחייב לעבור קומפילציה (תרגום) ל- Javascript כדי שהדפדפנים יבינו אותו מפני שדפדפנים מבינים Javascript בלבד. בסופו של דבר בסופו של דבר קוד Typescript מעובד לכדי קוד JavaScript ולכן ניתן לומר "באחריות" שהשפה נתמכת בכל דפדפן שתומך ב- JavaScript. קבצי Typescript יישמרו בסיומת ts.

2.4. תרשימים:

2.4.1. עץ תהליכים:

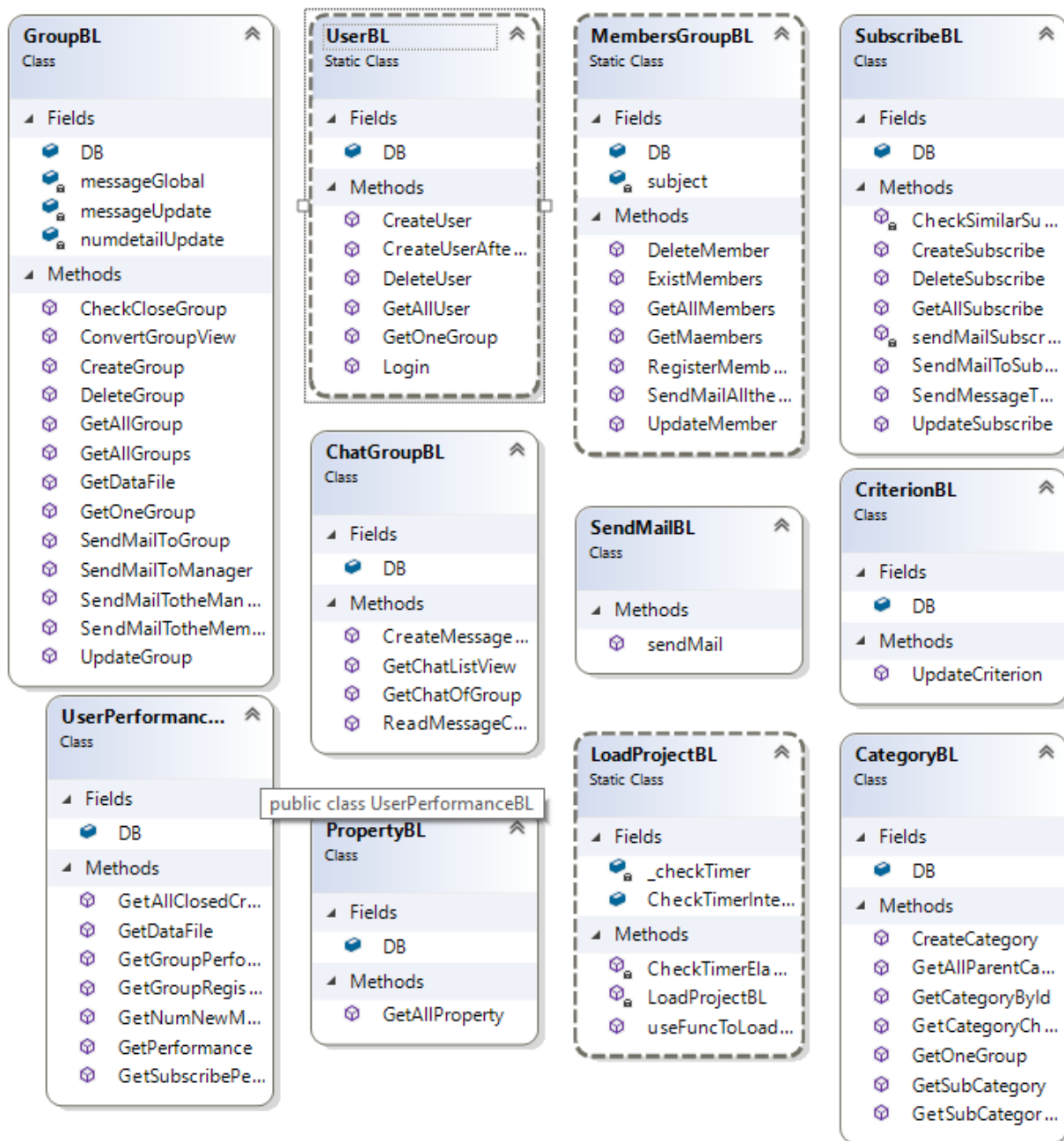


2.4.2. תרשים Uml:











תרשים מראה המחלקות: 2.4.3.

BL

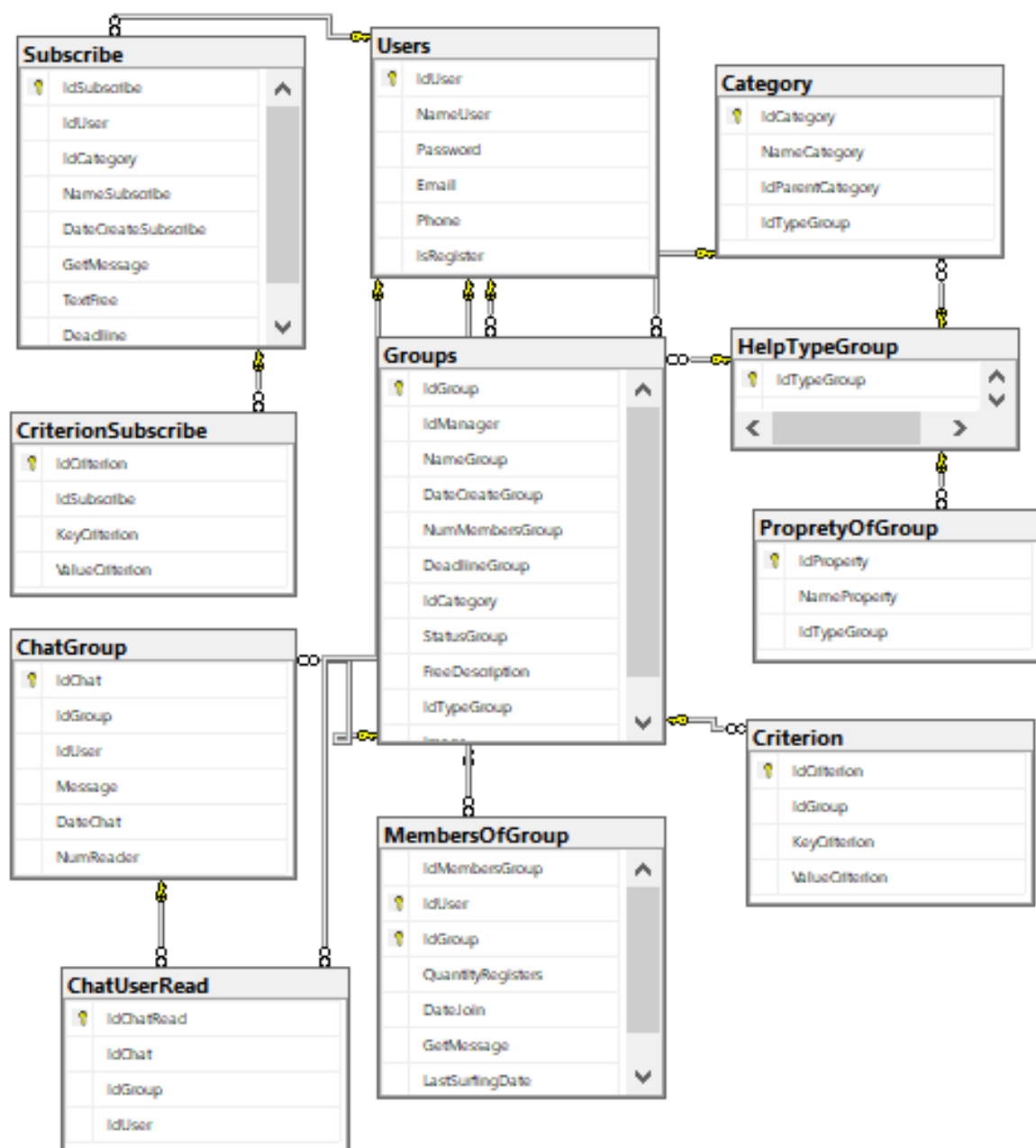


API

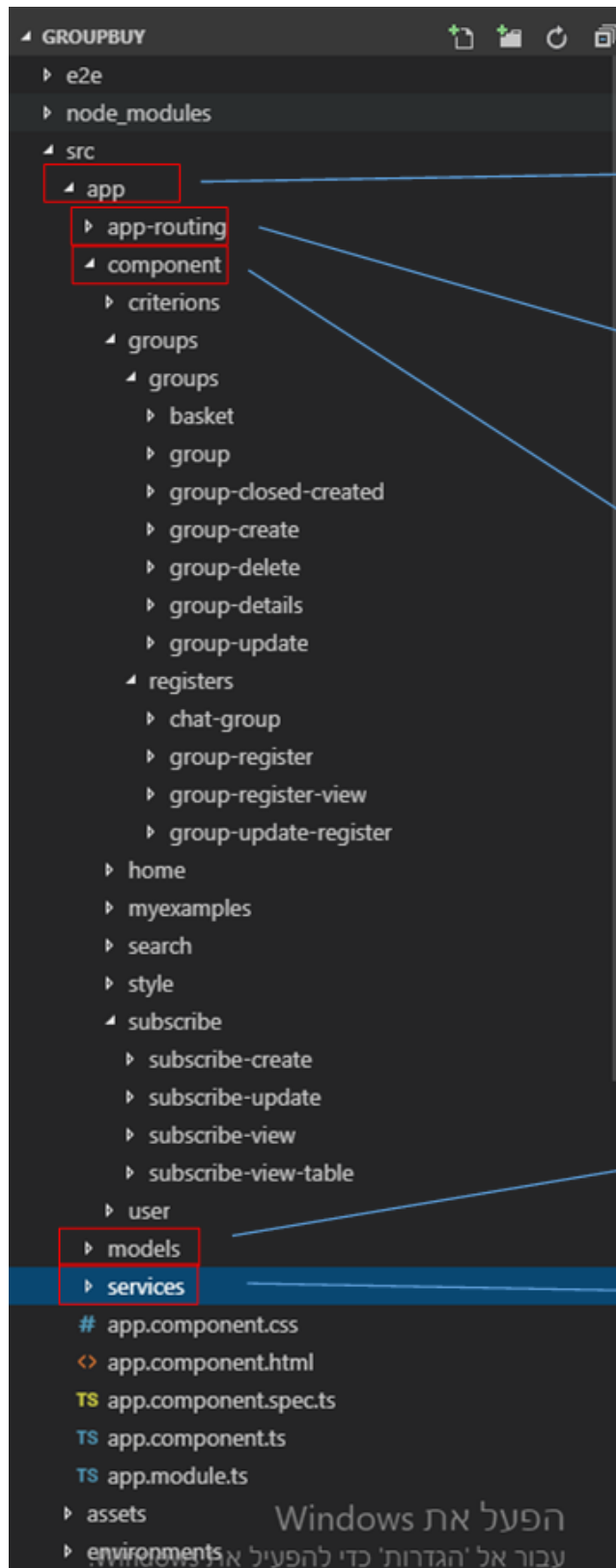
GroupsController  Class → ApiController Methods <ul style="list-style-type: none"> CreateGroup DeleteGroup GetAllGroup GetAllGroups GetGroup PutGroup SendMailTothe... 	UserPerformanc...  Class → ApiController Methods <ul style="list-style-type: none"> GetAllClosedCr... GetNumNewM... GetPerformance 	MembersGroup...  Class → ApiController Methods <ul style="list-style-type: none"> CreateMembers DeleteMember ExistMembers GetAllMaembers GetMaember SendMailAllthe... UpdateMember 	SubscribeContr...  Class → ApiController Methods <ul style="list-style-type: none"> CreateSubscribe DeleteSubscribe GetAllSubscribe PutSubscribe
UsersController  Class → ApiController Methods <ul style="list-style-type: none"> CreateUser CreateUserAfte... DeleteUser GetAllUser GetUser Login 	CategoryContro...  Class → ApiController Methods <ul style="list-style-type: none"> CreateCategory GetAllParentCa... GetCategoryByld GetCategoryCh... GetSubCategory GetSubCategor... 	ChatGroupContr...  Class → ApiController Methods <ul style="list-style-type: none"> CreateMessage... GetChatListView GetChatOfGroup ReadMessageC... 	LoadProjectCon...  Class → ApiController Methods <ul style="list-style-type: none"> SetTimer

2.5. מבנה נתונים מאוכסנים:

2.5.1. תיאור מבנה קבצי SQL:



2.5.2. מבנה קבצים ותיקיות:



בתוך תיקיה זו נמצאים הקבצים שמגדירים דברים לכל האתר, והוא הקומפוננטה הראשית שמכילה בתוכה את כל הדפים.

ניווט הלינקים באתר.

בתיקייה זו נמצאים כל דפי האתר מסודרים בתיקיות ותת תיקיות. כל דף באתר בנוי משלושה קבצים:

1. דף html
2. דף css
3. דף ts

תיקייה זו מכילה את כל המחלקות והאובייקטים.

תיקייה זו מכילה קבצים שבהם פונקציות השולחות ומקבלות קריאות http ותפקידם לקשר בין הקלינט לסרבר.

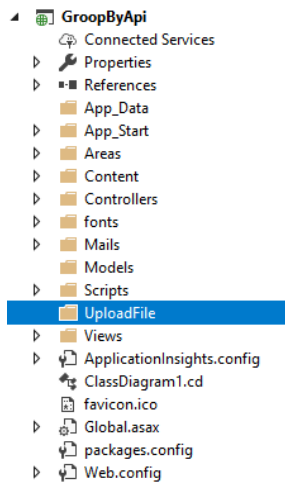
הפעל את Windows

עבור אל 'הגדרות' כדי להפעיל את Windows

צד הקלינט

2.5.3. תיקיית תמונות פרופיל הקבוצה-

בעת יצירת הקבוצה, המשתמש יכול להעלות תמונה שמביעה את הקבוצה שלו לדוגמא את המוצר שרכש או את המקום שאליו יסעו. התמונות נשמרות בתיקית UploadFile בשרת.



2.6. תוכן הפרויקט:

2.6.1. תאור המחלקות:

מחלקות ב-DAL:

Groups – (קבוצה)

מחלקה זו נותנת אפשרות להוסיף קבוצה חדשה עם כל הפרטים

```
14 references
public partial class Groups
{
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
    1 reference | 0 exceptions
    public Groups()
    {
        this.ChatGroup = new HashSet<ChatGroup>();
        this.Criterion = new HashSet<Criterion>();
        this.MembersOfGroup = new HashSet<MembersOfGroup>();
    }

    9 references | 0 exceptions
    public int IdGroup { get; set; }
    6 references | 0 exceptions
    public int IdManager { get; set; } //IdUser of the User open the Group
    7 references | 0 exceptions
    public string NameGroup { get; set; }
    2 references | 0 exceptions
    public System.DateTime DateCreateGroup { get; set; } //DateOfthe CreateGroup
    5 references | 0 exceptions
    public int NumMembersGroup { get; set; } //Num of MaxMium of members
    8 references | 0 exceptions
    public System.DateTime DeadlineGroup { get; set; } //DeadLine close the Group
    4 references | 0 exceptions
    public int IdCategory { get; set; }
    13 references | 0 exceptions
    public int StatusGroup { get; set; } //Status:0=closed,1=open,2=notSuccessful,3=deleted
    4 references | 0 exceptions
    public string FreeDescription { get; set; }
    3 references | 0 exceptions
    public int IdTypeGroup { get; set; } //Type of the Group:1=Bissusnes,2=frindlyGroup,3=Other
    2 references | 0 exceptions
    public string Image { get; set; } //Image of Group

    1 reference | 0 exceptions
    public virtual Category Category { get; set; }
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    1 reference | 0 exceptions
    public virtual ICollection<ChatGroup> ChatGroup { get; set; } //the list of the Messages that the members of this group left
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    4 references | 0 exceptions
    public virtual ICollection<Criterion> Criterion { get; set; } //the list of details about the group
    0 references | 0 exceptions
    public virtual HelpTypeGroup HelpTypeGroup { get; set; }
    4 references | 0 exceptions
    public virtual Users Users { get; set; }
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    4 references | 0 exceptions
    public virtual ICollection<MembersOfGroup> MembersOfGroup { get; set; } //the list of the members that register to this group
}
}
```

-User (משתמש)

מחלקה זו נותנת אפשרות להוסיף משתמש לאתר

```
public partial class Users
{
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
    1 reference | 0 exceptions
    public Users()
    {
        this.Groups = new HashSet<Groups>();
        this.MembersOfGroup = new HashSet<MembersOfGroup>();
        this.Subscribe = new HashSet<Subscribe>();
        this.ChatUserRead = new HashSet<ChatUserRead>();
    }

    8 references | 0 exceptions
    public int IdUser { get; set; }
    13 references | 0 exceptions
    public string NameUser { get; set; }
    4 references | 0 exceptions
    public string Password { get; set; }
    10 references | 0 exceptions
    public string Email { get; set; }
    2 references | 0 exceptions
    public string Phone { get; set; }
    5 references | 0 exceptions
    public Nullable<bool> IsRegister { get; set; } //if the new user confirm his register

    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    1 reference | 0 exceptions
    public virtual ICollection<Groups> Groups { get; set; } //all groups this user create
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    2 references | 0 exceptions
    public virtual ICollection<MembersOfGroup> MembersOfGroup { get; set; } //all groups this user register
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    1 reference | 0 exceptions
    public virtual ICollection<Subscribe> Subscribe { get; set; } // all subscribe this user ask
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    1 reference | 0 exceptions
    public virtual ICollection<ChatUserRead> ChatUserRead { get; set; } //all message chat this user read
}
```

-MembersOfGroup (משתמש רשום בקבוצה)

מחלקה שנותנת לרשום משתמש לקבוצה פתוחה

```
10 references
public partial class MembersOfGroup
{
    3 references | 0 exceptions
    public int IdMembersGroup { get; set; }
    15 references | 0 exceptions
    public int IdUser { get; set; } //the user that join to Group
    22 references | 0 exceptions
    public int IdGroup { get; set; }
    6 references | 0 exceptions
    public int QuantityRegisters { get; set; } //Quantity of the number the user
    4 references | 0 exceptions
    public System.DateTime DateJoin { get; set; } //Date of the user register to the group
    4 references | 0 exceptions
    public bool GetMessage { get; set; } //boolean if get the mail of the manager
    3 references | 0 exceptions
    public System.DateTime LastSurfingDate { get; set; } //date of the last surfing at the website

    8 references | 0 exceptions
    public virtual Groups Groups { get; set; }
    5 references | 0 exceptions
    public virtual Users Users { get; set; }
}
```

-Subscribe (בקשות לקבוצה שאינה קיימת)

מחלקה זו מאפשרת ליצור בקשה ליצירת קבוצה - ע"י משתמש אחר או תיווך חכם ע"י האתר

13 references

```
public partial class Subscribe
{
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
    1 reference | 0 exceptions
    public Subscribe()
    {
        this.CriterionSubscribe = new HashSet<CriterionSubscribe>();
    }

    8 references | 0 exceptions
    public int IdSubscribe { get; set; }
    4 references | 0 exceptions
    public int IdUser { get; set; } //the user that ask the subscribe
    7 references | 0 exceptions
    public int IdCategory { get; set; }
    5 references | 0 exceptions
    public string NameSubscribe { get; set; }
    3 references | 0 exceptions
    public System.DateTime DateCreateSubscribe { get; set; }
    3 references | 0 exceptions
    public int GetMessage { get; set; }
    3 references | 0 exceptions
    public string TextFree { get; set; } //describe the subscribe
    6 references | 0 exceptions
    public Nullable<System.DateTime> Deadline { get; set; } //date to delete the subscribe

    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    9 references | 0 exceptions
    public virtual ICollection<CriterionSubscribe> CriterionSubscribe { get; set; } //list of the criterion
    12 references | 0 exceptions
    public virtual Users Users { get; set; }
}
```

-ChatGroup (הודעות צ'אט של קבוצה)

מחלקה זו מאפשרת להוסיף הודעות חדשות לצ'אט קבוצתי בין החברים – לאחר סגירת הקבוצה

11 references

```
public partial class ChatGroup
{
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
    1 reference | 0 exceptions
    public ChatGroup()
    {
        this.ChatUserRead = new HashSet<ChatUserRead>();
    }

    6 references | 0 exceptions
    public int IdChat { get; set; }
    7 references | 0 exceptions
    public int IdGroup { get; set; }
    6 references | 0 exceptions
    public int IdUser { get; set; }
    3 references | 0 exceptions
    public string Message { get; set; }
    5 references | 0 exceptions
    public System.DateTime DateChat { get; set; }
    4 references | 0 exceptions
    public int NumReader { get; set; } //num of members of groups that read this message

    0 references | 0 exceptions
    public virtual Groups Groups { get; set; }
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    1 reference | 0 exceptions
    public virtual ICollection<ChatUserRead> ChatUserRead { get; set; }
}
```


-ChatUserRead (משתמש שקרא הודעה)

מחלקה זו מסמנת לכל משתמש איזה הודעות מבין ההודעות שהושארו בקבוצתו הוא קרא

7 references

```
public partial class ChatUserRead
```

```
{
```

1 reference | 0 exceptions

```
public int IdChatRead { get; set; }
```

3 references | 0 exceptions

```
public int IdChat { get; set; }
```

2 references | 0 exceptions

```
public int IdGroup { get; set; }
```

4 references | 0 exceptions

```
public int IdUser { get; set; }
```

```
}
```

0 references | 0 exceptions

```
public virtual ChatGroup ChatGroup { get; set; }
```

0 references | 0 exceptions

```
public virtual Users Users { get; set; }
```

```
}
```

מחלקות ב API

GroupController - (קבוצה)

```
[EnableCors(origins: "*", headers: "*", methods: "*")]
[RoutePrefix("api/Groups")]
0 references
public class GroupsController : ApiController
{
    /// <summary> לקבל את כל הקבוצות הפתוחות מקבל מזהה משתמש כדי לדעת איזה הרשאה לתת ...
    [Route("GetAllGroups/{idUser}")]
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public List<Entities.GroupView> GetAllGroup(int? IdUser)...
    [Route("GetAllGroups")]
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public List<Entities.Groups> GetAllGroups()...
    /// <summary> לקבל קבוצה אחת
    [Route("GetGroup/{id}")]
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public Entities.Groups GetGroup(int id)...
    /// <summary> מחיקת הקבוצה לא מחקה אמיתית סימון מצב
    [Route("DeleteGroup/{id}")]
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public HttpResponseMessage DeleteGroup(int id)...
    /// <summary> יצירת קבוצה חדשה
    [Route("CreateGroup")]
    [HttpPost]
    0 references | 0 requests | 0 exceptions
    async public Task<IHttpActionResult> CreateGroup()...
    /// <summary> עדכון פרטיהקבוצה
    [Route("UpdateGroup")]
    [HttpPut]
    0 references | 0 requests | 0 exceptions
    public HttpResponseMessage PutGroup([FromBody] Entities.Groups g)...
    /// <summary> שליחת מייל למנהל הקבוצה
    [Route("SendMailToManger/{idManager}/{BodyMail}")]
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public HttpResponseMessage SendMailTotheManager(int idManager, string[] BodyMail)...
}
```

(משתמש) -UserController

```

[RoutePrefix("api/Users")]
0 references
public class UsersController : ApiController
{
    /// <summary> לקבל את כל המשתמשים הרשומים באתר
    [Route("GetAllUsers")]
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public List<Entities.Users> GetAllUser()...
    /// <summary> קבלת משתמש אחד

    [Route("GetUser/{id}")]
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public Entities.Users GetUser(int id)...
    /// <summary> מחיקת משתמש

    [Route("DeleteUser/{id}")]
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public HttpResponseMessage DeleteUser(int id)...
    /// <summary> יצירת משתמש חדש

    [Route("CreateUser")]
    [HttpPost]
    0 references | 0 requests | 0 exceptions
    public HttpResponseMessage CreateUser([FromBody] Entities.Users u)...
    /// <summary> הפיכת המשתמש לפעיל לאחר אימות הרישום

    [Route("CreateUserAfterConfirm/{idUser}")]
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public HttpResponseMessage CreateUserAfterConfirm(int idUser)...
    /// <summary> כניסת המשתמש לאתר

    [Route("Login")]
    [HttpPost]
    0 references | 0 requests | 0 exceptions
    public HttpResponseMessage Login([FromBody] Entities.Users u)...
}

```

MembersOfGroupController (משתמש בקבוצה – חבר בקבוצה)

```
[RoutePrefix("api/MembersGroup")]
0 references
public class MembersGroupController : ApiController
{
    /// <summary> קבלת כל החברים הרשומים לקבוצה הזו
    [Route("GetAllMembers/{idGroup}")]
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public List<Entities.RegisterGroupView> GetAllMembers(int idGroup)...
    /// <summary> קבלת חבר אחד לקבוצה מסוימת
    [Route("GetMember/{idUser}/{idGroup}")]
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public Entities.MembersOfGroup GetMember(int idUser, int idGroup)...
    /// <summary> רישום חבר חדש
    [Route("CreateMembers")]
    [HttpPost]
    0 references | 0 requests | 0 exceptions
    public HttpResponseMessage CreateMembers([FromBody] Entities.MembersOfGroup m)...
    /// <summary> בודק האם המשתמש נרשם לקבוצה הזו
    [Route("ExistMembers/{idUser}/{idGroup}")]
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public HttpResponseMessage ExistMembers(int idUser, int idGroup)...
    /// <summary> ביטול רישום של חבר לקבוצה
    [Route("DeleteMember/{idUser}/{idGroup}")]
    [HttpDelete]
    0 references | 0 requests | 0 exceptions
    public HttpResponseMessage DeleteMember(int idUser, int idGroup)...
    /// <summary> עדכון רישום לקבוצה
    [Route("UpdateMember")]
    [HttpPut]
    0 references | 0 requests | 0 exceptions
    public HttpResponseMessage UpdateMember([FromBody] Entities.MembersOfGroup m)...
    /// <summary> שליחת הודעה לכל חברי הקבוצה
    [Route("SendMailRegisterGroup/{idGroup}/{Message}")]
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public HttpResponseMessage SendMailAlltheRegister(int idGroup, string Message)...
}
```

UserPerformanceController - (ביצועים של המשתמש באתר)

```

0 references
public class UserPerformanceController : ApiController
{
    /// <summary> קבלת כל רשימת הביצועים של המשתמש-קבוצות בקשות וכו
    [Route("GetPerformance/{idUser}")] // פעולות של המשתמש
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public Entities.PerformanceUser GetPerformance(int IdUser)...
    /// <summary> קבלת כל הקבוצות הסגורות של המשתמש
    [Route("GetAllClosedCreatedGroups/{idUser}")]
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public List<Entities.GroupView> GetAllClosedCreatedGroups(int idUser)...
    /// <summary> קבלת מספר הודעות חדשות מאז הגלישה האחרונה של משתמש זה
    [Route("GetNumNewMessage/{idUser}")]
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public int GetNumNewMessage(int idUser)...
}

```

SubscribeController - (בקשות ליצירת ותוור קבוצות)

```

0 references
public class SubscribeController : ApiController
{
    /// <summary> לקבל את כל הבקשות
    [Route("GetAllSubscribe")]
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public List<Entities.Subscribe> GetAllSubscribe()...
    /// <summary> מחיקת בקשה
    [Route("DeleteSubscribe/{id}")]
    [HttpDelete]
    0 references | 0 requests | 0 exceptions
    public HttpResponseMessage DeleteSubscribe(int id)...
    /// <summary> יצירת בקשה חדשה
    [Route("CreateSubscribe")]
    [HttpPost]
    0 references | 0 requests | 0 exceptions
    public HttpResponseMessage CreateSubscribe([FromBody] Entities.Subscribe s)...
    /// <summary> עדכון בקשה
    [Route("UpdateSubscribe")]
    [HttpPut]
    0 references | 0 requests | 0 exceptions
    public HttpResponseMessage PutSubscribe([FromBody] Entities.Subscribe s)...
}

```

ChatController - צ'אט קבוצתי לקבוצות שנסגרו)

```
[EnableCors(origins: "*", headers: "*", methods: "*")]
[RoutePrefix("api/ChatGroup")]
0 references
public class ChatGroupController : ApiController
{
    /// <summary> יצירת הודעה חדשה לקבוצה
    [Route("CreateMessageChat")]
    [HttpPost]
    0 references | 0 requests | 0 exceptions
    public HttpResponseMessage CreateMessageChat([FromBody]Entities.ChatGroup c)...
    /// <summary> קבלת רשימה של הקבוצות לפי קבוצה ומשתמש כדי לרשום איזה הודעות קראה
    [Route("GetChatOfGroup/{idGroup}/{idUser}")]
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public List<Entities.ChatGroup> GetChatOfGroup(int idGroup,int idUser)...
    /// <summary> קבלת כל ההודעות של קבוצה מסוימת
    [Route("GetChatListView/{idGroup}/{idUser}")]
    [HttpGet]
    0 references | 0 requests | 0 exceptions
    public List<Entities.ChatListView> GetChatListView(int idGroup, int idUser)...
    /// <summary> סימון למשתמש שקרא הודעה
    [Route("ReadMessageChat")]
    [HttpPost] (Alt+)
    0 references | 0 requests | 0 exceptions
    public HttpResponseMessage ReadMessageChat([FromBody] Entities.ChatUserRead chatRead)...
```

2.6.2. תאור פונקציות

פונקציות בצד שרת

1. פונקציית CheckSimilarSubscribe

פונקציה זו מקבלת בקשה חדשה שנוצרה ובודקת בכל הבקשות הקיימות באתר אם יש בקשות דומות שאפשר לתווך ביניהן. הפונקציה מוצאת את כל הבקשות מאותה קטגוריה ועם קריטריונים דומים ושולחת להם מייל עם פרטי הבקשות הדומות.

```

/// <summary>
/// פונקציה שמתווכת בין בקשות דומות, מופעלת כשנוספת בקשה חדשה ,
/// מפעילה פונקציה שליחת מייל לכל המשתמשים שנמצאו עם בקשות דומות ומציעה להם לתקשר בעצמם ולפתוח קבוצה באתר
/// </summary>
/// <param name="subscribe">הבקשה החדשה</param>
1 reference | 0 exceptions
private static void CheckSimilarSubscribe(Subscribe subscribe)
{
    bool flag;
    var AllSubscribes = DB.Subscribe.Where(p => p.IdCategory == subscribe.IdCategory && p.IdSubscribe != subscribe.IdSubscribe);
    List<DAL.Subscribe> subscribes = new List<Subscribe>();
    if (AllSubscribes != null)
    {
        foreach (var item in AllSubscribes)//עובר על הבקשות מאותו קטגוריה ומחפש בקריטריונים
        {
            flag = false;
            foreach (var crit in subscribe.CriterionSubscribe)//עובר על הקריטריונים של הבקשה החדשה
            {
                foreach (var criter in item.CriterionSubscribe)// (item) עובר על הבקשה הנוכחית של
                {
                    if (crit.KeyCriterion.Contains(criter.KeyCriterion) || crit.ValueCriterion.Contains(criter.ValueCriterion))
                    {
                        subscribes.Add(item);
                        flag = true;
                        break;//אם יש קריטריון דומה אחד כבר נוסף למערך ואין צורך לבדוק שוב
                    }
                }
            }
            if (flag)
                break;
        }
        subscribes.Add(subscribe);
        sendMailSubscribe(subscribes, "MailSubscribe.html");
    }
}

```

הפעל את Windows

2. פונקציית sendMailSubscribe

פונקציה ששולחת מייל עם כל פרטי הבקשות הדומות לכל המשתמשים שבקשו בקשות אלו.

```

/ <summary>
/ פונקציית שליחת מייל לכל המשתמשים שיש להם בקשות דומות
/ html מוצאת את המשתמשים מכניסה את הפרטים שלהם לטבלה של html
/ ואת פרטי הבקשה שלהם
/ </summary>
/ <param name="subscribes">הדומות הדומות</param>
/ <param name="path">לדף של המייל</param>
reference | 0 exceptions
ivate static void sendMailSubscribe(List<Subscribe> subscribes, string path)

string tableUserOfSubscribe = "";
string trForTable = @"<tr>
    <td>{0}</td>
    <td>{1}</td>
    <td>{2}</td>
    <td>{3}</td>
    <td>{4}</td>
</tr>";

string criterionim;
string AllCriterion = "";
foreach (var item in subscribes) // המרה של הקריטריונים ל html
{
    AllCriterion = "<p>";
    criterionim = "<b>{0}</b>: {1} ,";
    foreach (var crit in item.CriterionSubscribe)
    {
        AllCriterion += string.Format(criterionim, crit.KeyCriterion, crit.ValueCriterion);
    }
    AllCriterion += "</p>";
    tableUserOfSubscribe += string.Format(trForTable, item.Users.NameUser, item.NameSubscribe, item.DateCreateSubscribe, item.Users.Email, AllCriterion);
}

string filesDir = HttpContext.Current.Server.MapPath("~/Mails");
string FileExtension = Path.GetExtension(path);
string destFile = System.IO.Path.Combine(filesDir, path);
StreamReader SourceReader = new StreamReader(destFile);
var HtmlBody = SourceReader.ReadToEnd();
foreach (var item in subscribes)
{
    string subject = "שלום ";
    subject += item.Users.NameUser;
    subject += " |";
    subject += "ש" + subscribes.Count + " הדומות לבקשתך ";
    string messageBody = string.Format(HtmlBody, item.Users.Email, item.Users.NameUser, tableUserOfSubscribe);
    SendMailBL.SendMail(subject, messageBody, item.Users);
}

```


3. פונקציה **CheckTimer** במחלקת Timer

```

/// <summary>
/// במחלקה זו הטימר מופעל ע"י קריאה שמגיע מהקלינט בעת טעינת האתר
/// הטימר בודק האם צריך לסגור את הקבוצה לפי התאריך של סגירת הקבוצה
/// </summary>
2 references
public static class LoadProjectBL
{
    //הקצאת הטימר
    private readonly static System.Timers.Timer _checkTimer = new System.Timers.Timer();
    //הטימר יופעל בכל יום פעם אחת בלבד
    public static readonly int CheckTimerInterval = 1000 * 60*60*24;
    /// <summary>
    /// הבונה של הטימר מאתחלת לאיזה פונקציה להפעיל
    /// </summary>
    0 references | 0 exceptions
    static LoadProjectBL()
    {
        _checkTimer.Elapsed += CheckTimerElapsed;
        _checkTimer.Interval = CheckTimerInterval;
        _checkTimer.Enabled = true;
    }
    /// <summary>
    /// הפונקציה מופעלת כל יום ובודקת איזה קבוצות צריך לסמן אותם כסגורים או לשלוח הודעת התרעה שהקבוצה עומדת להיסגר
    /// </summary>
    1 reference | 0 exceptions
    static void CheckTimerElapsed(object source, ElapsedEventArgs e)
    {
        DAL.NechamiRinaGroupByEntities DB = new DAL.NechamiRinaGroupByEntities();
        //var f = group.Where(g => DateTime.Compare(g.DeadlineGroup.Date,DateTime.Today.Date)==1);
        //מכיל את כל הקבוצות שעדיין פתוחות
        var groupOpen = DB.Groups.Where(g => g.StatusGroup == 1).ToList();
        if (groupOpen != null)
        {
            עובר בלולאה על כל הקבוצות
            foreach (var item in DB.Groups.ToList())
            {
                אם התאריך יעד יעבור רק עוד שבוע אז שולח הודעה למנהל הקבוצה שהקבוצה עומדת להיסגר
                if ((int)(DateTime.Today.Date - item.DeadlineGroup.Date).TotalDays == 7)
                {
                    מפעיל פונקציה ששולחת מייל למנהל
                    GroupBL.SendMailToTheManagerBeforeDeadline(item.Users, 7);
                }
                else if ((int)(DateTime.Today.Date - item.DeadlineGroup.Date).TotalDays == 3)
                {
                    GroupBL.SendMailToTheManagerBeforeDeadline(item.Users, 3);
                }
                else if ((int)(DateTime.Today.Date - item.DeadlineGroup.Date).TotalDays == 2)
                {
                    GroupBL.SendMailToTheMembersBeforeDeadline(item.MembersOfGroup.ToList(), 2);
                }
                אם הקבוצה נסגרת היום שולח הודעה למנהל ולשאר חברי הקבוצה שהקבוצה נסגרת
                else if ((int)(DateTime.Today.Date - item.DeadlineGroup.Date).TotalDays == 0)
                {
                    GroupBL.SendMailToTheManagerBeforeDeadline(item.Users, 0);
                    GroupBL.SendMailToTheMembersBeforeDeadline(item.MembersOfGroup.ToList(), 0);
                    item.StatusGroup = 0; //הקבוצה נסגרה
                }
            }
        }
        עדכון בנתונים שהקבוצה נסגרה
        DB.SaveChanges();
    }
}

```

פונקציות בצד לקוח

בדף Group - תצוגת כל הקבוצות הפתוחות

1. פונקצייה לקבלה מהשרת של כל הקבוצות הפתוחות

פונקציה לרפרוש הקבוצות. נקראת בעת טעינה ולאחר כל שינוי `RefreshGroups()` שהתרחש בדף

```
this.servicegroup.GetAllGroups(Number(localStorage.getItem("idUser"))).subscribe((data:
GroupView[]) => {
    this.numberOfOpenGroups = data.length;
    this.AlltheOpenGroups = data;
    this.searchListData = data;
    this.DataPaginatorList = data;
    this.searchListData.forEach(res => {
        this.blob = this.dataURItoBlob(res.contentImage); // יצירת בלוב בשביל התמונה
        this.url = window.URL.createObjectURL(this.blob);
        console.log(this.searchListData);
        res['imageSrc'] = this.sanitizer.bypassSecurityTrustResourceUrl(this.url);
        res.basket=false;
    });
    this.finish=true;
    this.idUserSurfing = Number(localStorage.getItem("idUser"));
    this.ClickRadio(this.labelPosition); // כשהקבוצות נטענות לפי המיון שבחר
    this.lengthItems = this.AlltheOpenGroups.length; // אתחול paginator
    this.pageSize = 6;
    this.fillGroups(0);
}
)
```

2. פונקציות רישום לקבוצה – פותחת את רישום לאתר אם עדיין לא נרשמו

groupRegister(item: GroupView) { // רישום לקבוצה

if (!localStorage.getItem("idUser")) // אם עדיין לא נכנסו לאתר

```

    this.openDialogRegisterWebSite(item);
else
    this.openDialogRegisterGroup(item);
}

```

//פתיחת דיאלוג רישום אם המשתמש עדיין לא ביצע כניסה / רישום לאתר/

```

openDialogRegisterWebSite(item) {
    const dialogConfig = new MatDialogConfig();
    dialogConfig.disableClose = true;
    dialogConfig.autoFocus = true;
    dialogConfig.direction = "rtl";
    dialogConfig.data = {
        id: 1,
        title: 'Angular For Beginners'
    };
    const dialogRef = this.dialog.open(RegisterComponent, dialogConfig);
    dialogRef.afterClosed().subscribe(data => {
        if (data == undefined) { //אם הדיאלוג נסגר בגלל שנרשם/
            this.RefreshGroups();
            //בדיקה האם משתמש זה כבר רשום לקבוצה/
            this.servicegroup.ExistsMemberINGroup(Number(localStorage.getItem("idUser")),
item.IdGroup).subscribe(data => {
                this.openDialogRegisterGroup(item);
            }
            , error => { //אם משתמש זה כבר נרשם לקבוצה יפתח לו חלון של עריכת פרטי
הרישום כולל אפשרות מחיקה מקבוצה
                Swal.fire({
                    title: 'הנך רשום לקבוצה זו',
                    text: "האם ברצונך לערוך את פרטי הרישום?",
                    type: 'warning',
                    showCancelButton: true,
                    confirmButtonText: 'כן',
                    cancelButtonText: 'ביטול',
                    cancelButtonColor: '#d33',

```

```

        confirmButtonColor: 'green',
        showCloseButton: true
    }).then((result) => {
        if (result.value) { //רוצה לעדכן יפתח חלון עדכון
            this.UpdateopenDialogGroup(item);
        }
        else if (result.dismiss === Swal.DismissReason.cancel) {
            this.dialog.closeAll();
        }
    });
});
}
});
}

```

```

openDialogRegisterGroup(item) { //דיאלוג רישום לקבוצה מסוימת
    const dialogConfig = new MatDialogConfig();
    dialogConfig.disableClose = true;
    dialogConfig.autoFocus = true;
    dialogConfig.direction = "rtl";
    dialogConfig.data = {
        id: 2,
        title: 'Angular For Beginners',
        group: item
    };
    const dialogRef = this.dialog.open(GroupRegisterComponent, dialogConfig);
    dialogRef.afterClosed().subscribe(result => {
        if (result == undefined)
            this.RefreshGroups();
    });
}

```

בדף Create-Group יצירת קבוצה

3. פונקציות שונות – הוספת והסרת קריטריונים, בחירת קובץ

הוספת קריטריונים קבועים של האתר, הקריטריונים נוספים בהתאם `AddPropety()` `{//`
לקטגוריה

```

this.groupService.GetPropertyById(this.idType).subscribe((data: Property[]) => {
  console.log(data);
  this.g.CriterionList = [];
  this.i = 0;
  //המרה למערך של הקריטריונים
  for (let item of data) {
    this.g.CriterionList.push(new Criterion());
    this.g.CriterionList[this.i].KeyCriterion = item.NameProperty;
    this.i++;
  }
  this.length = this.g.CriterionList.length;
}, error => {
  alert("errorGetAllGroup");
});
}
//הוספת קריטריון
clickAddCritarion() {
  this.g.CriterionList.push(new Criterion());
}
//הסרת קריטריון
clickRemoveCritarion() {
  if (this.g.CriterionList.length > this.length) {
    let lengthOfSubsribreList = this.g.CriterionList.length - 1;
    this.g.CriterionList = this.g.CriterionList.splice(0, lengthOfSubsribreList);
  }
}

onSelectFile(event, file) {
  this.file = file[0];
  this.filesUpload.push(file);
}

```

```

if (event.target.files && event.target.files[0]) {
    var reader = new FileReader();

    reader.readAsDataURL(event.target.files[0]); // read file as data url

    reader.onload = (event) => { // called once readAsDataURL is completed
        this.url = event.target['result'];
    }
}
}

```

4. פונקציית יצירת קבוצה – שולחת את הקבוצה ומקבלת תשובה האם יצירת הקבוצה הצליחה

```

submitCreateGroup() {
    this.g.IdCategory = this.category.IdCategory;
    this.g.IdManager = Number(localStorage.getItem("idUser"));
    this.g.IdTypeGroup = this.idType;
    this.registrGroup.IdUser = this.g.IdManager;
    this.registrGroup.DateJoin = new Date();
    this.registrGroup.GetMessage = true;
    let list: Criterion[] = [];
    for (let i = 0; i < this.g.CriterionList.length; i++) {
        if (this.g.CriterionList[i].KeyCriterion != null &&
            this.g.CriterionList[i].ValueCriterion != null) {
            list.push(this.g.CriterionList[i]);
        }
    }
    this.g.CriterionList = list;
    this.groupService.GroupCreate(this.g, this.file, this.registrGroup).subscribe(data => {
        Swal.fire({
            type: 'success',
            title: 'הקבוצה נוצרה בהצלחה!',
            text: "מיד תקבל הודעה מפורטת במייל",
            showConfirmButton: true,

```

```
//timer: 1200
});
},
error => {
  alert("error.statusText");
});
}
```

בדף ChatGroup הודעות של קבוצה שנסגרה בהצלחה

5. פונקציה בעת טעינה - מקבלת בניתוב פרמטר - IdGroup

```
ngOnInit() {
  this.activatedRoute.params.subscribe(p => {רפרוש של הרשימות
    this.tempid = p['idGroup'] || "";
    if (this.tempid != "") {
      this.IdGroup = this.tempid;
      this.GetIdManger();
      this.GetTheListChat();רפרוש של הרשימות
      this.NumNewChat = 0;
    }
  });
}
```

6. פונקציה לקבלת כל ההודעות של הקבוצה שהתקבלה בניתוב מופעלת בעת טעינה ובכל שינוי באתר – כתיבת הודעה חדשה וסימון הודעה כהודעה שנקראה

```
GetTheListChat() {
  if (Number(localStorage.getItem("idUser"))) {אם בכלל אין מישור שנרשם
    this.chatService.GetChatListView(this.IdGroup,
    Number(localStorage.getItem("idUser"))).subscribe(
      (data: ChatListView[]) => {
        this.chatList = data;
        this.chatList = this.chatList.sort(function compare(a, b) {
          var dateA = new Date(a.DateChat);
          var dateB = new Date(b.DateChat);
          return dateB.getTime() - dateA.getTime()
        });
      });
    }
  }
```

```

});
this.AccountNewMessageForGroup(this.IdGroup);
});
this.chatService.GetAllMemberinGroupForView(this.IdGroup).subscribe((data:
RegisterGroupView[]) => {
    this.MembersofChat = data;
});
}
}

```

7. פונקציית שליחת מייל לכל חברי הקבוצה – רק למנהל הקבוצה

```

SendMailToRegisterGroup())//יכול/ מנהל רק החברים
{
    this.MailToRegisterGroup = "";
    this.serviceGroup.SendMailOftheregitergroup(this.IdGroup,
    this.MailToRegisterGroup).subscribe(data => {
        });
    }
}

```

8. פונקציה ליצירת הודעה חדשה

```

CreateMessageChat() {
    this.MessageChat.IdUser = Number(localStorage.getItem("idUser"));
    this.MessageChat.IdGroup = this.IdGroup;
    this.MessageChat.DateChat = new Date();
    this.chatService.CreateMessageChat(this.MessageChat).subscribe(
        (data: ChatGroup) => {
            //סימן הודעה שנקראה למתמש שכתב את ההודעה החדשה
            this.ReadMessageChat(data.IdChat);
        });
    }
    );
}

```

9. פונקציה לסימון קריאת הודעה למשתמש

```

ReadMessageChat(idChat: number) {
    this.chatUserRead.IdChat = idChat;
}

```



```
this.chatUserRead.IdGroup = this.IdGroup;
```

```
this.chatUserRead.IdUser = Number(localStorage.getItem("idUser"));
```

```
this.chatService.ReadMessageChat(this.chatUserRead).subscribe(data => {
```

```
    this.GetTheListChat();//רפרוש של הרשימות
```

```
});
```

```
}
```

בדף Basket

הסל נשמר ב localStorage של מחשב הקלינט, בכל טעינה של האתר מתבצעת בדיקה אם יש בסל שאצל הקלינט קבוצות שכבר לא רלוונטיות כמו קבוצות שנסגרו בהצלחה או נמחקו.

עדכון שינויים בסל נעשה באמצעות משתנה Subject

```
private subjectBasket = new Subject<any>();//משתנה שמעדכן בשינוי של הסל
getBasketObservable(): Observable<any> {
    return this.subjectBasket.asObservable();
}
```

10. פונקציות הוספה והסרה מהסל

```
addToBasket(item) { //הוספה לסל
    item.basket = true;
    if (!localStorage.getItem("basket")) { //אם עדיין אין סל
        this.basket = [];
        this.basket.push(item);
    }
    else {
        this.basket = JSON.parse(localStorage.getItem("basket"));//שליפה של הסל
        'מהלוקל סטורג'
        this.basket.push(item);
    }
    localStorage.setItem("basket", JSON.stringify(this.basket));//שמירת הסל
    'בלוקל סטורג' של מחשב הקלינט
    this.subjectBasket.next(this.basket.length);//העברה של המידע שהיה שינוי בסל
}
```

```
removeFromBasket(item) { //הסרה מהסל
    item.basket = false;
    this.basket = JSON.parse(localStorage.getItem("basket"));
    this.basket.splice(item, 1);
    if (this.basket.length == 0) //אם מחקו את כל הקבוצות מהסל אז להסיר מהלוקל
        localStorage.removeItem("basket");
    else
        localStorage.setItem("basket", JSON.stringify(this.basket));
}
```

```
this.subjectBasket.next(this.basket.length);
}
```

11. פונקציה ש"מאזינה" לשינויים בסל (כדי לעדכן את מספר הקבוצות

שמופיע בלינק באתר

(נמצאת ב constructor)

```
this.subscriptionBasket = this.func.getBasketObservable().subscribe(length
=> {
    this.NumItemBasket = length;
});
```

12. פונקציות שבודקות את הסל ואת הקבוצות בדף הנוכחי

// בודקת את הסל אם יש בו קבוצות שכבר נסגרו או נמחקו – לפי קבוצות שלא הגיעו

```
CheckBasket(groups: GroupView[] = []) {
    if (localStorage.getItem("basket")) {
        this.groupService.GetAllGroupS().subscribe((data: Groups[]) => {
            this.GroupList = data;
            let j;
            this.basket = JSON.parse(localStorage.getItem("basket"));
            for (let i = 0; i < this.basket.length; i++) {
                for (j = 0; j < this.GroupList.length; j++) {
                    if (this.basket[i].IdGroup == this.GroupList[j].IdGroup)
                        break;
                }
                if (j == this.GroupList.length) {
                    let item = this.basket[i];
                    this.basket.splice(item.IdGroup, 1);
                }
            }
            localStorage.setItem("basket", JSON.stringify(this.basket));
        });
        if (groups.length > 0)
            this.CheckGroupIfBasket(groups);
    }
}
```

// בודקת מי מהקבוצות שהגיעו נמצאת בסל וצריך שיהיה לה סמל של הסרה

```
CheckGroupIfBasket(groups: GroupView[]) {
    this.basket = JSON.parse(localStorage.getItem("basket"));
    for (let i = 0; i < groups.length; i++) {
        for (let j = 0; j < this.basket.length; j++) {
            if (groups[i].IdGroup == this.basket[j].IdGroup) {
                groups[i].basket = true;
            }
        }
    }
    this.subjectBasket.next(this.basket.length);
}
```

3. מדריך למשתמש:

3.1 הוראות כלליות לשימוש באתר:

3.1.1 מדריך לאורח:

בעת כניסת אורח לאתר יש לו מספר אפשרויות אך מעט מוגבלות: האורח יכול להכנס לדף צפיה בקבוצות הפתוחות (קבוצות שלא נסגרו לרשום) להיכנס לפרטים של כל קבוצה לערוך חיפוש נרחבים לפי שם קבוצה, קטגוריה, קריטריונים. וכן לסנן אותם לפי אחוזי הרישום ועוד. יש לו אפשרות גם ללחוץ על הוסף לסל אם ברצונו אחר כך לערוך השוואה בין קבוצות. אך אם רצונו להצטרף לקבוצה או לפתוח בעצמו קבוצה עליו להרשם לאתר. וכן אם הוא מעוניין לראות בקשות שאנשים הציעו או רוצים קבוצות שיפתחו הוא יכול להכנס לדף צפיה בקשות הוא אף יכול לערוך שם חיפוש. אך להוסיף בקשה חדשה הוא אינו יכול.

3.1.2 מדריך למשתמש:

כאשר משתמש נכנס לאתר הוא יכול לבצע הרבה פעולות:

1. הוא יכול להכנס לדף צפיה בקבוצות להיכנס לפרטים ועוד.
2. המשתמש יכול להצטרף לקבוצה עליו להכניס את מספר הנרשמים. בתור התחבובות הוא הצטרך להכניס פרטי האשראי באמצעות ה-paypal.

אם המשתמש נכנס לקבוצה שכבר רשום אליה הוא יוכל לעדכן את מספר הנרשמים וכן לצאת מהקבוצה.

אם המשתמש נכנס לקבוצה שהוא פתח אותה-בתור מנהל יש לו אפשרויות נוספות-

1. הוא יכול להכנס לדף צפיה בחברי הקבוצה לשלוח להם הודעות או לשלוח לחבר באופן פרטי.
2. הוא יכול לערוך את פרטי הקבוצה מחדש, את תאריך הסגירה לא יוכל לשנות.
3. הוא יכול למחוק את הקבוצה לפי התנאים הבאים:

אם אף אחד עדיין לא נרשם לקבוצה חוץ ממנו הוא יכול למחוק אותה.

אך אם נרשמו עוד חברים הוא מועבר לדף מחיקת קבוצה ושם ידרש לשלם קנס לחברים על סגירת הקבוצה.

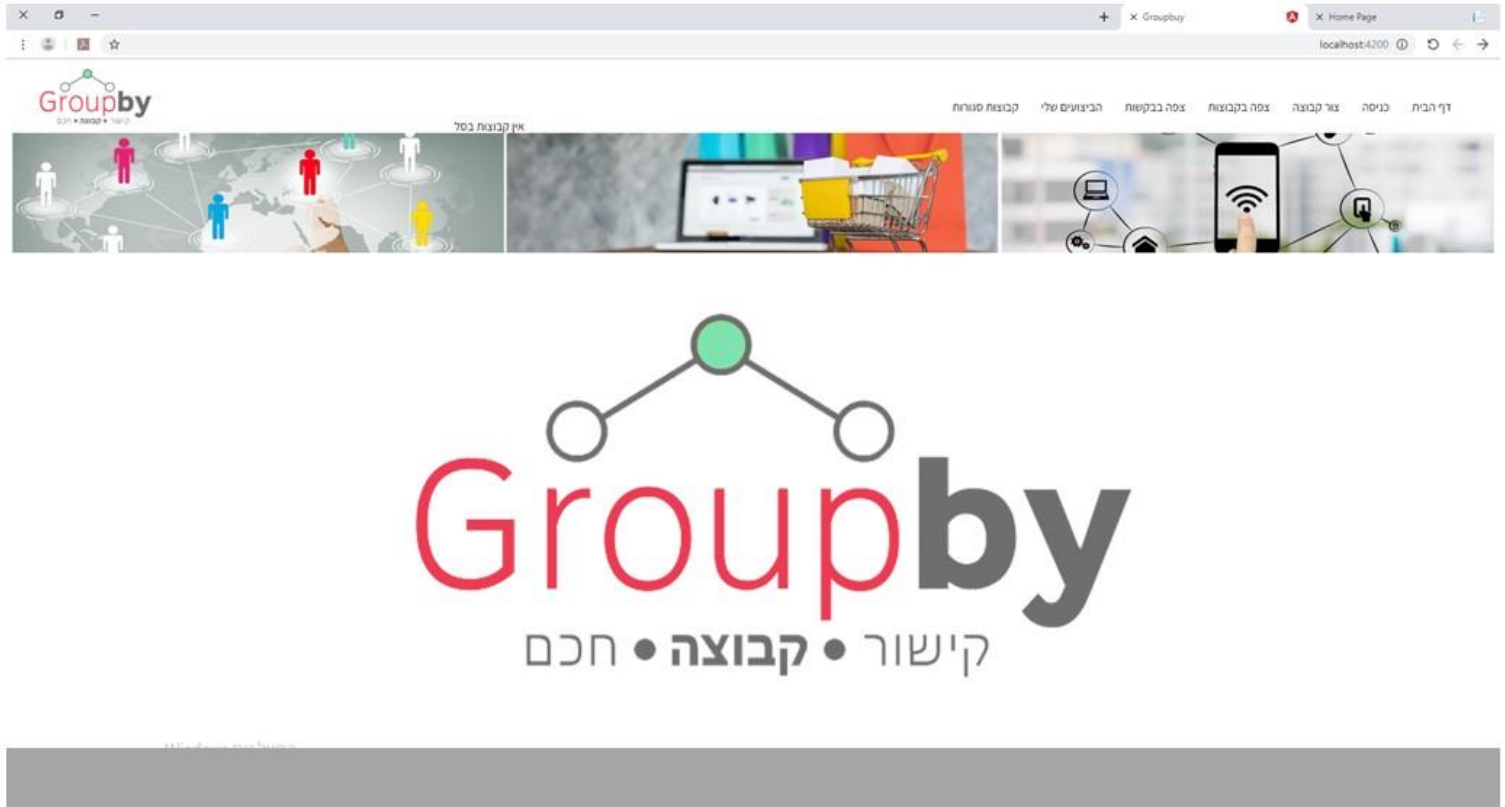
3. הוא יכול לפתוח קבוצה חדשה עליו למלא טופס על הקבוצה: שם, קטגוריה, מספר החברים הכללי, מספר הנרשמים שלו, תמונה של המוצר או של תאור הקבוצה, תאריך סגירת הקבוצה.

4. כל משתמש יכול לצפות בדף ביצועים שלו באתר: קבוצות שפתח, קבוצות שהצטרף ובקשות שבקש.

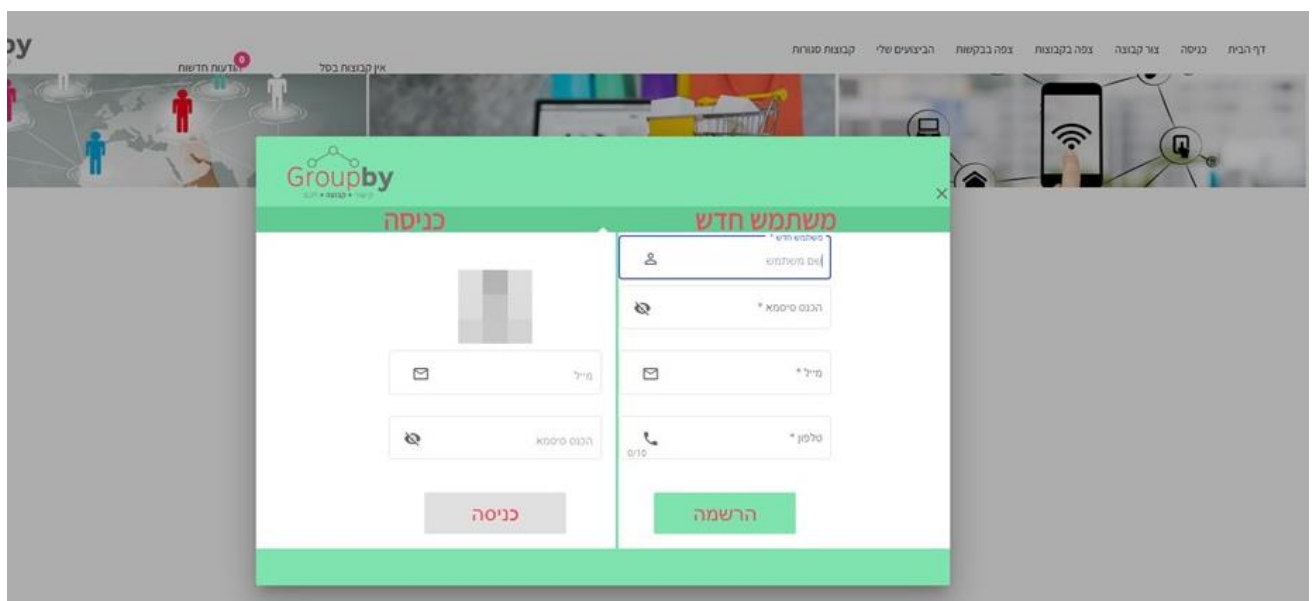
5. הוא יכול לצפות בקבוצות סגורות שהוא מצורף אליהם-שם הוא יכול לצפות בהודעות של חברי הקבוצה וכן להשאיר הודעה.

3.2. מסכים:

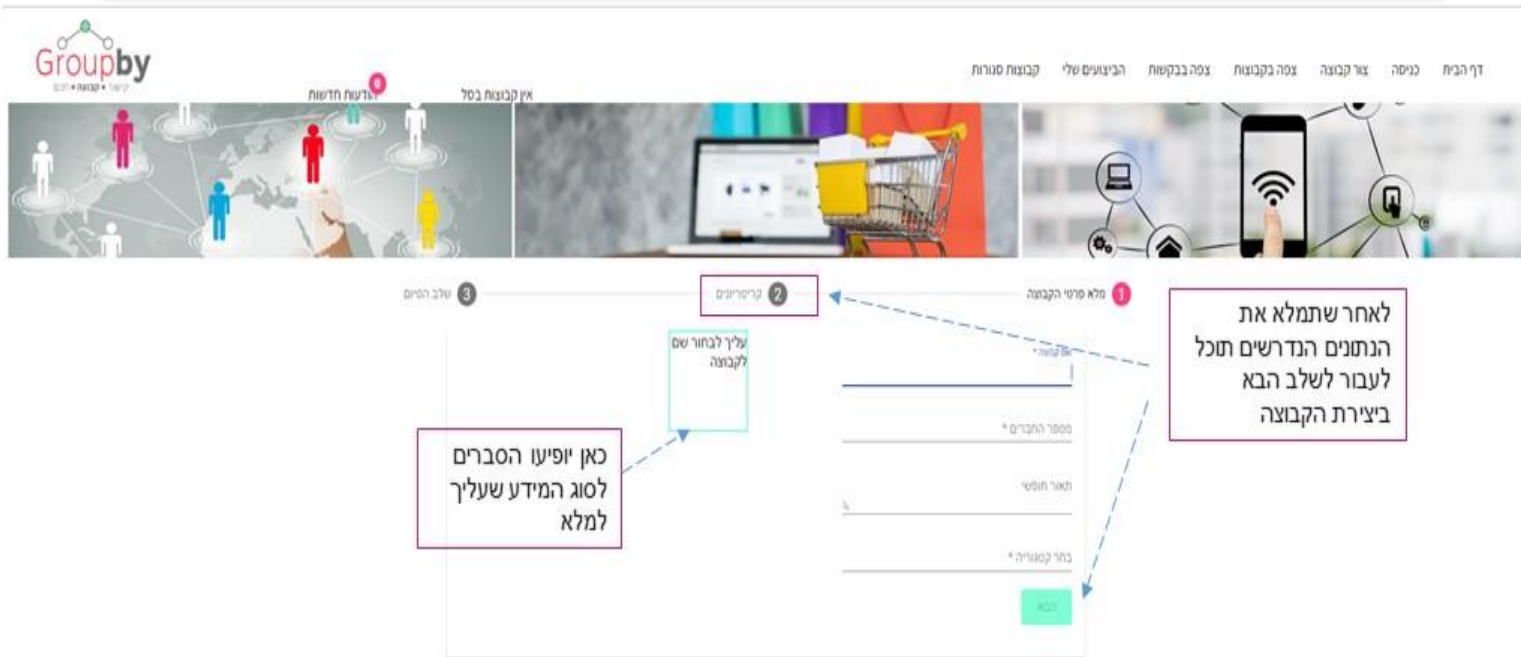
3.2.1. מסך ראשי:



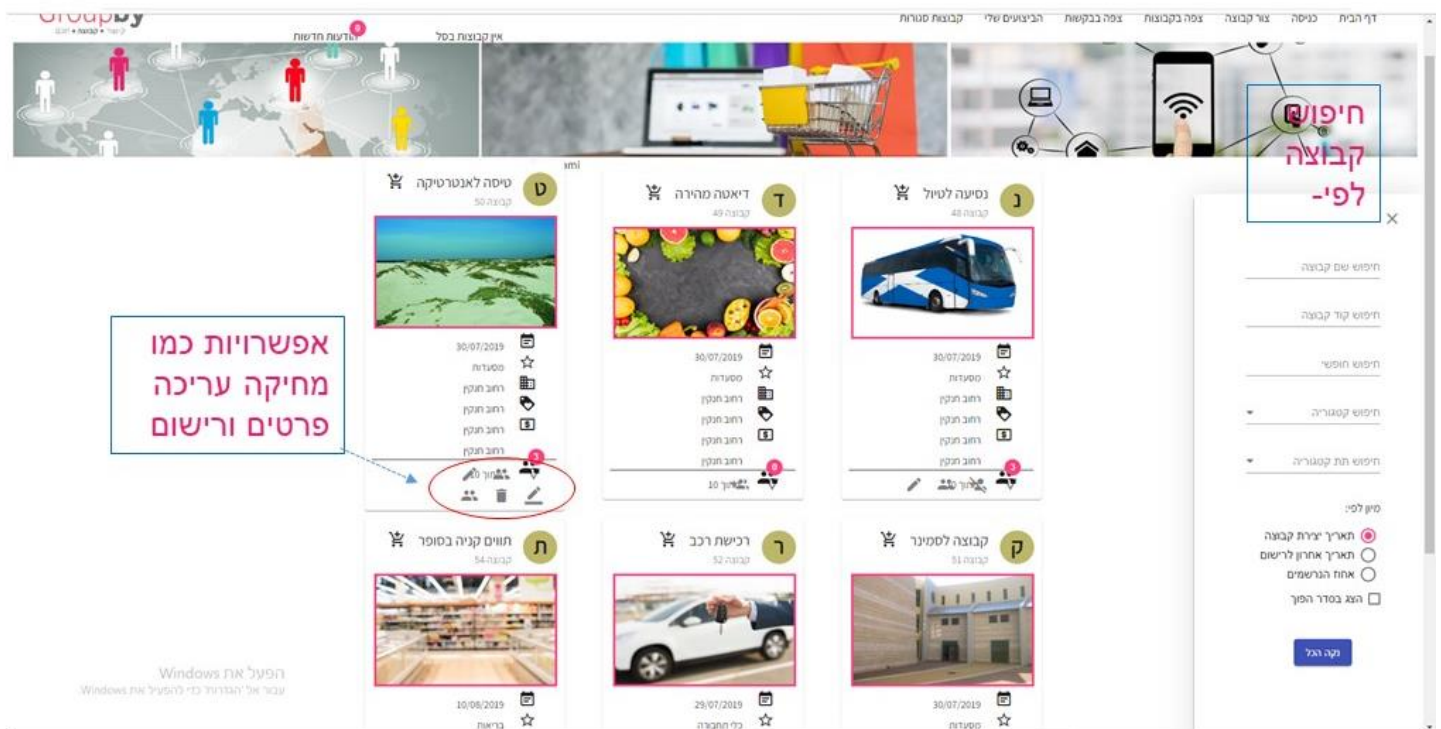
3.2.1. רישום וכניסה לאתר



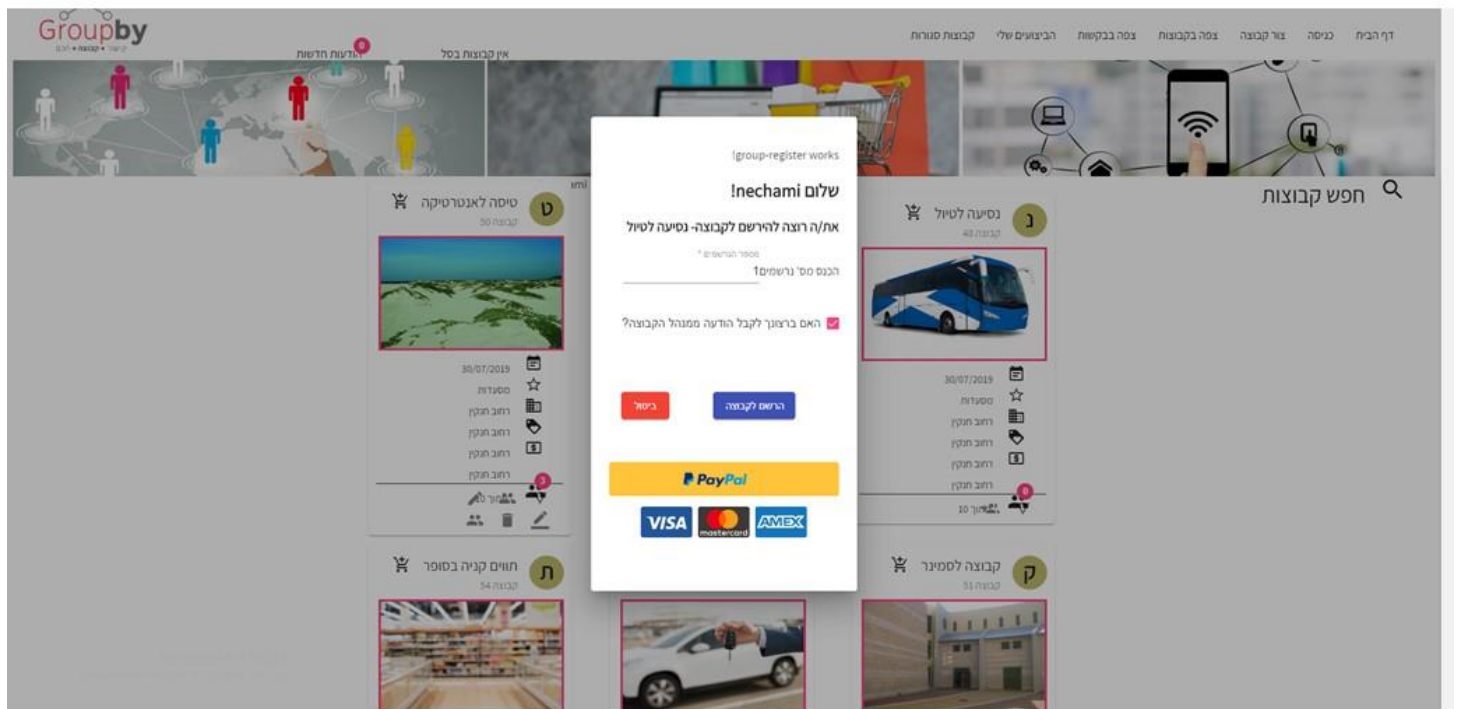
3.2.2. יצירת קבוצה חדשה



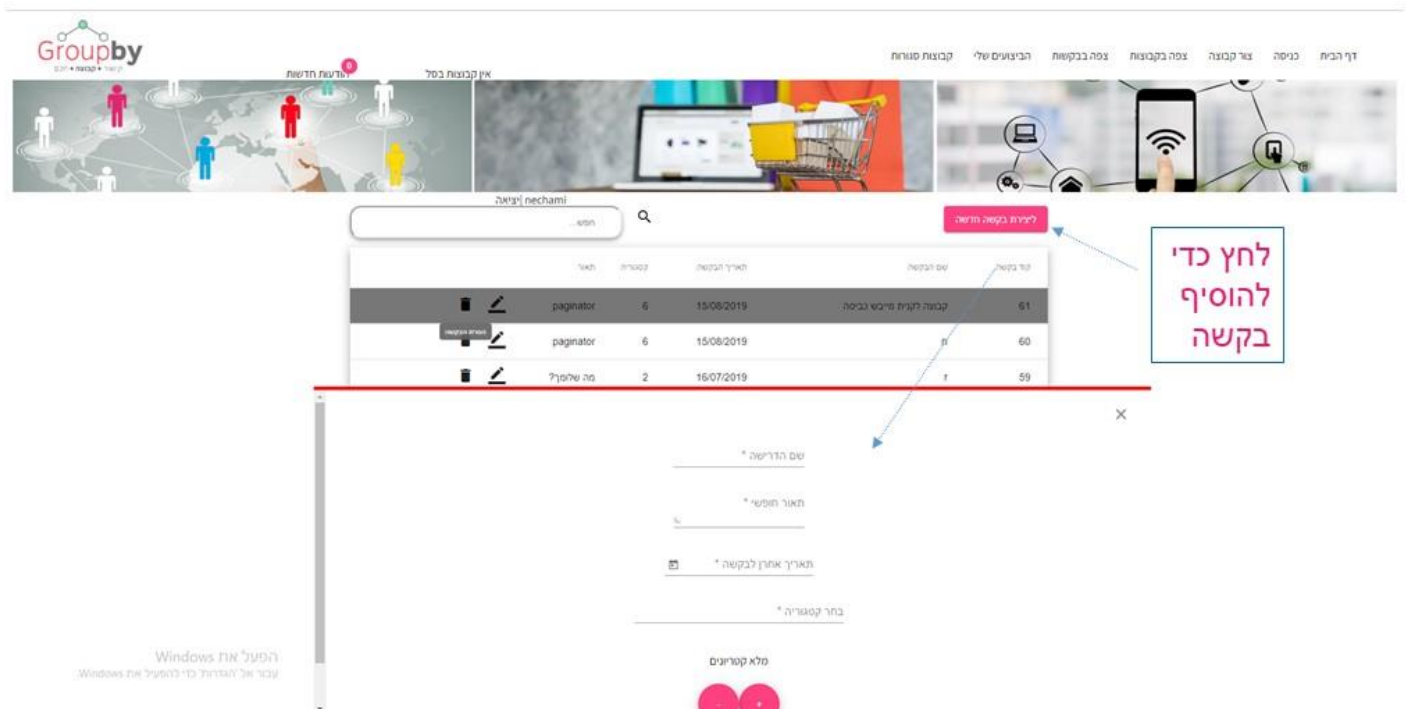
3.2.3. צפייה בקבוצות פתוחות



3.2.5. רישום לקבוצה



3.2.6. צפיה בבקשות ויצירת בקשה חדשה



3.2.7. צפיה בהודעות לקבוצות סגורות



0

מייל לחברים בקבוצה

שלח הודעה חדשה לחברים בקבוצה שלך

צור הודעה חדשה

מייל השולח הודעות לחברים

מייל השולח	הודעות לחברים	תאריך	מספר הודעות
katz6679@gmail.com	bjnhhhhhhhhhhhhhhhhh	03/09	3
katz6679@gmail.com	שלום! בואו נבדוק אם הציאט עובד!!!!!!!!!!!!	29/08	קראו את ההודעה
gmail.com@6162142	שלום! בואו נבדוק אם הציאט עובד!!!!!!!!!!!!	20/08	3
q@q	שלום! בואו נבדוק אם הציאט עובד!!!!!!!!!!!!	20/08	3
q@q	שלום! בואו נבדוק אם הציאט עובד!!!!!!!!!!!!	10/08	3

כמנהל-תוכל לשלוח מייל לחברים בקבוצה

רק למנהל-מייל לחברים ב...

מספר החברים בקבוצה שלך הוא: 2

katz6679@gmail.com

q@q

4. סיכום ומסקנות:

השקענו רבות בבניה נכונה של המערכת ועל תקשורת נכונה עם השרת, ראשית הפרויקט הקנה לנו מרחב הסתכלות ופרספקטיבה רחבה על פרויקט מושלם, משלב ההצעה ועד לתוצר המוגמר תוך דרך ארוכה הכוללת באגים ופתרונם ויצירתיות רבה על מנת להפוך את האתר ליעיל, חווייתי ומקצועי ככל האפשר. במהלך הפרויקט למדנו הרבה, ואנו מרגישות כי הוא תרם לנו רבות. תחומים שהכרנו באופן שטחי, נעשו ידידותיים בהרבה ממה שחשבנו. השקענו זמן ומאמץ מרובים בלמידת נושאים וטכנולוגיות חדשים. גילינו שמהנה גם אם לא קל ללמוד ולהכיר תחומים חדשים ולנסות דרכי חשיבה מגוונות ומפתיעות שלא חשבנו עליהם ממבט ראשון. למדנו לחשוב בהגיון וביעילות על מנת להשיג תוצאות במהירות. למדנו לעבוד עם

Angular ועם Angular Material.

ומעל הכול הפנמנו שסבלנות והתמדה-אין כמותם בכל משימה הנראית קשה עד בלתי אפשרית....

5. נספחים:

קובץ Package.json:

```
{
  "name": "groupbuy",
  "version": "0.0.0",
  "scripts": {
    "ng": "ng",
    "start": "ng serve",
    "build": "ng build",
    "test": "ng test",
    "lint": "ng lint",
    "e2e": "ng e2e"
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "^6.1.10",
    "@angular/cdk": "^7.3.7",
    "@angular/common": "^6.1.0",
    "@angular/compiler": "^6.1.0",
    "@angular/core": "^6.1.0",
    "@angular/forms": "^6.1.0",
    "@angular/http": "^6.1.0",
    "@angular/material": "^7.3.7",
    "@angular/platform-browser": "^6.1.0",
    "@angular/platform-browser-dynamic": "^6.1.0",
    "@angular/router": "^6.1.0",
    "core-js": "^2.5.4",
    "hammerjs": "^2.0.8",
    "moment": "^2.24.0",
    "ngx-paypal": "^5.0.0",
    "rxjs": "~6.2.0",
    "sweetalert2": "^8.11.7",
    "zone.js": "~0.8.26"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "~0.8.0",
    "@angular/cli": "~6.2.5",
    "@angular/compiler-cli": "^6.1.0",
    "@angular/language-service": "^6.1.0",
    "@types/jasmine": "~2.8.8",
    "@types/jasminewd2": "~2.0.3",
    "@types/node": "~8.9.4",
    "codelyzer": "~4.3.0",
    "jasmine-core": "~2.99.1",
    "jasmine-spec-reporter": "~4.2.1",
```

```
"karma": "^4.1.0",  
"karma-chrome-launcher": "~2.2.0",  
"karma-coverage-istanbul-reporter": "~2.0.1",  
"karma-jasmine": "~1.1.2",  
"karma-jasmine-html-reporter": "^0.2.2",  
"protractor": "~5.4.0",  
"ts-node": "~7.0.0",  
"tslint": "~5.11.0",  
"typescript": "~2.9.2"  
}  
}
```

6. ביבליוגרפיה:

אתרים בנושא תכנות:

- stackoverflow.com
- webmaster.org.il
- w3schools.com
- material.angular.io
- angular.io
- material.io
- github.com

