



# ספר פרויקט

שם המכללה: "מרכז החרדי להכשרה מקצועית".

מסלול: הנדסאי תוכנה – מגמת סייבר.

שם הפרויקט: "freeDelivery"

שמות הסטודנטים: הילה אבשלומוב, יעלי גולדשטיין.





## הצעה לפרויקט גמר

### מסלול : הנדסאי תוכנה

שם בוגר	ת"ז	כתובת	טלפון	שנת סיום
הילה אבשלומוב				2020
יעלי גולדשטיין				2020

שם המנחה	כתובת	טלפון	תואר	מקום עבודה
אילנה פיישר			תואר שני בפיזיקה והנדסת תוכנה, הסמכת CISO ( chief information security officer	מכון לב, ירושלים

מל המכללה: 72274

שם המכללה: המרכז החרדי להכשרה מקצועית

מסלול: הנדסאים

מגמה: הנדסת תוכנה התמחות סייבר

### 1. שם הפרויקט

### 2. רקע

#### 2.1. תיאור ורקע כללי

בפרויקט זה אנו מציגות אתר שמתווך בין אנשים שרוצים להעביר משלוחים לאיזורים מסויימים בארץ, שלא ע"י הדואר או חברת תיווך אלא ע"י אנשים פרטיים שנוסעים

הילה אבשלומוב



באופן קבוע מאיזור מגוריהם/איזור הקרוב אליהם לאזור הרצוי למשלוח ומעוניינים בהתנדבות העברת משלוחים. אדם זה יכניס את פרטיו האישיים, פרטי החבילה, האיזורים והתאריך הרצוי, המערכת תחפש נוסע ברשימת הנוסעים הקבועים המתאים לו לפי מיקום בחישוב מרחק ע"י גוגל מפת, לפי תאריך, גודל חבילה, סוג החבילה האם שביר או לא, תציג אותם למבקש המשלוח ואם הוא מאשר שאכן נראה לו מתאים המערכת תשלח בקשת משלוח עם פרטיו למייל. אם המתנדב מעוניין לבצע שליחות המבקש והשליח יתאמו ביניהם במייל, מבקש השליחות יסמן באתר שמצא שליח מתאים. יום אחרי ביצוע השליחות מבקש המשלוח מקבל הודעה במייל אם נעזר באתר שייכנס למלא משוב. המשוב עוזר למערכת לקבוע דירוג לכל מתנדב לפי כמות שליחויות שביצע.

## **2.2. מטרת המערכת:**

- \*שליחות מהירה
- \*חסכון בדמי משלוח
- \*ריכוז גדול של שליחים ליעדים רבים בארץ

## **3.סקירת מצב קיים בשוק, אלו בעיות קיימות:**

כיום ניתן לשתף דרך רשתות חברתיות ווצאפ וכו' הודעה על בקשה לשליחות, שנמצא נוסע מתאים לא תמיד ניתן לדעת אם האדם אמין או לא, לעומת זאת אצלנו יש בדיקת אימות ת"ז, השליחים שנרשמים לאתר הם כאלה המעוניינים בביצוע שליחויות, באתר שלנו יש ריכוז גדול של שליחים הקיימים בשוק, עם מידע על השעות בהם השליחים פנויים לעבודה וכך מאפשר התאמה מהירה בין מבקש המשלוח לשליח.

## **4.מה הפרויקט אמור לחדש או לשפר:**

אפשרות שליחות מהירה ומדויקת כי הנוסעים הרשומים באתר הינם קבועים ויודעים בדיוק מתי יגיעו ליעדם, אמנם השליחות היא לא עד בית הלקוח, אך יש כל מיני חבילות של אנשים פרטיים שמעוניינים שהמשלוח יגיע במהירות האפשרית וללקוח אין בעיה לאסוף את החבילה ממקום מפגש עם הנוסע.

## **5.דרישות מערכת ופונקציונליות:**

### **5.1 דרישות מערכת:**

הסביבה שבה המערכת תוטמע הינה מחשב אישי מבוסס חלונות כולל תקשורת בסיסית ברשת האינטרנט. רוחב הפס של התקשורת אמור להיות סטנדרטי- 100mbps אמור להספיק (ניתן על ידי כל ספקי שירותי האינטרנט).  
סביבות פיתוח: visual studio 2017 professional  
שפת פיתוח: C#.NET, מסד נתונים רלציוני.  
הגישה למערכת ע"י דפדפן סטנדרטי.

### **5.2 דרישות פונקציונליות:**

#### **5.2.1 דרישות עבור משתמש:**

- 5.2.1.1 כניסה/הרשמה לאתר עם שם משתמש וסיסמא
- 5.2.1.2 שליחת התראות ועדכונים למייל
- 5.2.1.3 התקשרות עם מנהל האתר
- 5.2.1.4 שליח
  - 5.2.1.4.1 אפשרות עריכת פרטי נסיעות
  - 5.2.1.5 מבקש משלוח
  - 5.2.1.5.1 אפשרות עריכת פרטי החבילה
  - 5.2.1.5.2 אפשרות סינונים לבחירת שליח רצוי לפי קריטריונים

הילל יחיאל



### 5.2.5 דרישות עבור מנהל:

- 5.2.5.1 צפיה במאגר נתוני שליחים
- 5.2.5.2 צפיה במאגר נתוני מבקשי משלוח
- 5.2.5.3 אפשרות הוספה, עדכון ומחיקה של שליח/ מבקש משלוח

### 6 בעיות צפויות במהלך הפיתוח ופתרונות אפשריים:

#### 6.1 בעיות אפשריות:

- 6.1.1 רישום רובוטים/ רישום משתמשים לא רצויים למערכת, דבר שיפגע באמינות ויעילות האתר
- 6.1.2 ביצוע התקשורת בין מבקש המשלוח לשליח

#### 6.2 פתרונות:

- 6.2.1 בעת רישום משתמש המערכת תבצע ולידציה למס' זהות שהוזן.
- 6.2.2 חשבנו לחשוף את מספר הטלפון או המייל של השליח למבקש המשלוח אבל בגלל אבטחת מידע העדפנו שהמערכת תתאם ביניהם

### 7 פתרון טכנולוגי נבחר:

#### 7.1 טופולוגית הפתרון:

המערכת מורכבת ממסד נתונים של SQL-server המכיל את כל נתוני האתר. שרת מקומי של visual studio המקושר למסד שדרכו נשלפים הנתונים ומתבצעים בו אלגוריתמים. וצד לקוח עבור המשתמשים אותו ניתן להריץ על דפדפן אינטרנט סטנדרטי.

#### 7.2 טכנולוגיות בשימוש:

Sql server, Asp.Net טכנולוגיות אלו תואמות את דרישות האתר.

#### 7.3 שפות הפיתוח:

- 7.3.1 צד שרת ייכתב בשפת C# (web api) שפה על עילית מונחית עצמים.
- 7.3.2 צד הלקוח יהיה angular8 ויעשה שימוש ב-css, javascript ועוד...

#### 7.4 תיאור הארכיטקטורה הנבחרת:

הארכיטקטורה הנבחרת היא חלוקה לארבע שכבות - 4-TIRE ARCHITECTURE. בפרויקטים המבוססים על DATABASE מומלץ להשתמש בגישה של מודל השכבות. בארכיטקטורה הזו קיימת הפרדה בין השכבות השונות בפרויקט. היתרון העיקרי בחלוקה לשכבות היא חלוקה קלה, במיוחד במערכות גדולות כך שניתן יחסית בקלות להחליף כל שכבה בלי לגעת בשכבות האחרות כאשר נרצה לעשות שינויים בתכנית.

#### 7.5 חלוקה לתכניות ומודולים:

- 7.5.1 שכבת נתונים ( DATA ACCESS LAYER-DAL) שכבה זו מורכבת ממקור נתונים- מסד הנתונים שלנו וממערכת תוכנה ENTITY FRAMEWORK אשר תפקידה לקרוא את המידע הנדרש למערכת, לשמור את הנתונים, להוסיף מידע חדש ולמחוק פרטי מידע קיימים.
- 7.5.2 שכבת הלוגיקה העסקית ( BUSINESS LOGIC-BL) השכבה שאחראית על הלוגיקה של המערכת עוסקת בעיבוד מידע וחישובים שונים ושליחת המידע לשכבת התצוגה. בשכבה זו נממש את הפונקציונאליות של המערכת.
- 7.5.3 BUSINESS ENTITIES-BE בשכבה זו קיימות רק מחלקות של אובייקטים שמקבילות למחלקות בשכבות ה- DAL לצרכי אבטחה- הגנה על שכבת ה- DAL מפני חשיפה ולצורך וולידציות שונות על מידע המגיע מהקליינט.
- 7.5.4 WEB API שכבה זו אחראית על חשיפת הפונקציונליות על מנת לאפשר תקשורת בין שרת ללקוח.

#### 7.6 סביבת השרת:

לצורך הפרויקט נשתמש בשרת מקומי. לאחר גמר הפרויקט והעברתו ללקוח, נעלה אותו לענן.

הי' א' י' א'



### 7.7 ממשק המשתמש/לקוח-GUI

שכבה זו מורכבת מדף HTML בודד ע"פ טכנולוגיית angular שמוצגת למשתמש באמצעות דפדפן

### 7.8 ממשקים למערכות אחרות API::GoogleMapsAPI

שימוש בחבילות תוכנה: CSS, Entity Framework, Visual Studio, Bootstrap  
ועם Google Maps

### 8. שימוש במבני נתונים וארגון קבצים:

#### 8.1 מבני נתונים:

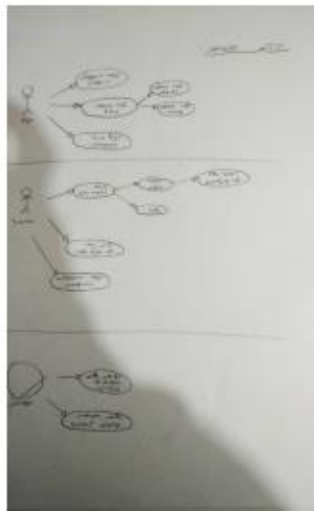
- אמצעי תחבורה (אוטובוס, רכבת, אופניים, אופנוע)
- נסיעות (מקום יציאה, מקום חזרה, שעת יציאה, שעת חזרה - אפשרות לבחירת כמה נסיעות ביום)
- נוסע קבוע (שם, סיסמא, תז, טלפון, ת. לידה, מייל, מקום יציאה, מקום חזרה, שעת יציאה, שעת חזרה, משקל מקסימלי, שביר/לא שביר, תחבורה דרך הלור(בחירה מרובה), תחבורה דרך חזור(בחירה מרובה), כמות שליחיות
- מבקש המשלוח (שם, סיסמא, טלפון, ת. לידה, מייל, מקום יציאה, מקום חזרה, שעת יציאה, שעת חזרה, שם- פירוט חבילה, משקל, שביר/לא שביר, תמונה-אופציונלי, תאריך שליחה, הערות)
- פרטי חבילה (שם- פירוט, משקל, שביר/לא שביר, תמונה-אופציונלי)

#### 8.2 שיטת האחסון:

נתוני האתר יישמרו בתוכנת SQL, פעולות על הנתונים יבוצעו ב- entity #framework C

### 9. תרשימי מערכת מרכזיים Visio

#### Use Case 9.1



### 10. תיאור המרכיב האלגוריתמי-חישובי

חישוב מרחקים בין נק' המוצא לנק' היעד של החבילה ע"י שימוש בgoogle maps

היעד יחל



## 11. תיאור/התייחסות לנושאי אבטחת מידע:

- 11.1 למשתמש רגיל, לגולש אלמוני ולמנהל יש interface שונה. ז.א. לכל דף באתר יש הגדרה למי מותר לראות את הדף הזה, ונעשה אימות על פי תפקיד משתמש
- 11.2 ל סיסמאות וכל מידע רגיש, נכתבים במאגר נתונים או אם זה בקובץ בצורה מוצפנת, כדי שאי אפשר לגנוב אותם. לכן, במידה ולמישהו יש גישה לנתונים, הוא לא יכול לקרוא אותם
- 11.3 כל input שמשתמש מקיש וזה נשלח לשרת עובד ולידציה: סוג, אורך, פורמט וגבולות, כדי למנוע התקפות כמו SQL injection, cross-site scripting ו- buffer overflow
- 11.4 בדרך הגדרת סיסמה או שינוי סיסמה לדרוש ממשתמש להגדיר סיסמה בעלת לפחות 8 תאים עם כללים לרמת החוזק של סיסמה (כמו שיהיה לפחות אות אחת גדולה וספרה אחת). במידה וסיסמה לא עומדת בדרישות הכפתור "הדכן" חייב להיות לא פעיל

## 12. משאבים הנדרשים לפרויקט:

- 12.1 מספר שעות המוקדשות לפרויקט: 700
- 12.2 חלוקת העבודה בין חברי הצוות: 350 שעות כל אחת
- 12.3 ציוד נדרש: מחשב, רשת אינטרנטית
- 12.4 תוכנות נדרשות: מערכת הפעלה windows 7 ומעלה, Microsoft.Net Framework
- 12.5 ידע חדש שנלמד לצורך ביצוע הפרויקט: התממשקות עם מערכות google maps API
- 12.6 ספרות ומקורות מידע:

<https://angular.io/>

<https://developers.google.com/maps/documentation/>

<https://www.w3schools.com/>

<https://stackoverflow.com/>

<https://www.tutorialspoint.com/>

<http://webmaster.org.il/>

## 13. תכנית עבודה ושליבים למימד הפרויקט:

- 13.1 ייזום: יוני
- 13.2 ניתוח מערכת: יוני
- 13.3 ניתוח מבנה נתונים: יוני
- 13.4 אפיון UI-UI: עד אוגוסט
- 13.5 כתיבת הלוגיקה העסקית: עד ינואר
- 13.6 כתיבת ממשק המשתמש: עד ינואר
- 13.7 עיצוב: עד פברואר
- 13.8 בדיקות התוכנה: עד פברואר

היג'י



**14. תכנון הבדיקות שיבוצעו:**

מספר בדיקה	מספר הדרישה במסמך האפיון	מקרי הבדיקה	ידנית/אוטומטית	חשיבות	הערות
1	5.2.1.1	בדיקה האם שם משתמש והסיסמה קיימים והאם מתאימים ביניהם	ידנית	גבוהה	-
2	5.2.1.2	בדיקת תקינות רישום שעות נסיעה (יציאה וחזרה)	ידנית	גבוהה	-
3	5.2.1.3	בדיקה שאכן המייל נשלח למנהל האתר- נרשם לשליח שהבקשה לעזרה התקבלה ותטופל בהקדם	ידנית	גבוהה	-
4	5.2.1.5.1	אפשרות עריכת פרטי חבילה גם לאחר שהבקשה כבר נרשמה באתר	ידנית	גבוהה	
5	5.2.1.5.2	בדיקה שאכן השליחים המוצגים תואמים את דרישות מבקש המשלוח לפי הסינונים שבחר	ידנית	גבוהה	
6	5.2.5.3	בדיקת אפשרות של מנהל הוספה/עדכון/מחיקה של שליח/מבקש משלוח	ידנית	גבוהה	

**15. בקרת גרסאות (version control):**

אין בקרת גרסאות, מאחר והפרויקט מיוצר מקומית על המחשב. במקרה ונרצה לעבוד ממחשבים שונים נעלה את הפרויקט ל git המאפשר עבודת צוות ממחשבים שונים, ומבטח את הפרויקט כיוון שקיימות גרסאות קודמות וניתן לחזור אליהן בכל עת.

הילל יחיאל









## תודות

פרויקט זה היווה עבורנו הזדמנות לבחון וליישם את הידע הנרחב והמגוון, אותו רכשנו במהלך שנות לימודינו.

חשוב לנו להביע את תודתנו והערכתנו לכל אלו אשר תמכו, סייעו ועזרו לנו לכל אורך הדרך.

ברצוננו להודות:

- לבורא עולם, שנתן לנו את הידע והכישרונות, היכולות, הכלים והרעיונות לביצוע הפרויקט.
- לרכזת המגמה עדינה שטיינברג על הדאגה וההתחשבות בתקופת הלימודים.
- לגב' מירי על העזרה בהנחיית הפרויקט לאורך הדרך, תוך דגש על הקידום המקצועי.
- לגב' נחמי על העזרה והסיוע הרב שנתנה לנו בכל דבר לו נדרשנו ועל הליווי וההכוונה אשר סיפקה לנו.
- לחברות המדהימות על ההתעניינות, הפירגון והעזרה...



## מבוא:

בזמן שחיפשנו אחר רעיון לפרויקט פרצה בישראל מגפת הקורונה אשר גררה אחריה סגר, עוצר וסגירת כלל המוסדות הלא חיוניים.

מצב זה גרם לכך שאנשים רבים נותרו ללא פרנסה וללא אפשרות להגיע לקרוביהם הנמצאים בערים הרחוקות מהם.

לכן הקו שהנחה אותנו בבחירה הושפע מכך שרצינו למצוא דרך יעילה ומעשית שבה ניתן יהיה לסייע לחברה ולעודד את ערך החסד בעמינו.

בעקבות כך בנינו את freeDelivery, אתר שמתווך בין אנשים שרוצים להעביר משלוחים לאזורים מסוימים בארץ בלי לשלם, לבין אנשים פרטיים שנוסעים מאזור הקרוב אליהם ללימודים/ עבודה באופן קבוע.

השליחים יעשו זאת בהתנדבות מלאה וכן בזמן עוצר או סגר יהיו גם כאלו שיש להם אישורי נסיעה מיוחדים למעבר חופשי בארץ.

תהליך החיפוש פשוט: הלקוח מזין את פרטיו, פרטי חבילתו והאזורים המבוקשים עם ממשק Google Maps, המערכת תבצע חיפוש במאגר השליחים ותציג את אלו העונים על הקריטריונים.

הלקוח יבחר בשליחים המתאימים לו והמערכת תשלח אליהם בקשה לשליחות ותקדם את התהליך.

היתרונות באתר זה הן שהשליחויות מתבצעות ללא תשלום וכן האתר מספק נוחות וחיסכון בזמן.

מודול נוסף הקיים במערכת הוא כניסת מנהל המאפשרת צפיה במאגרי המשתמשים עם אופציות לעדכון, מחיקה והוספה שלהם, וכן יש לו האפשרות למחוק תגובות ופידבקים שאינם ראויים.



## תוכן עניינים

### 1. הגדרת דרישות ותיאור כללי

1.1 תיאור כללי

1.1.1 מטרות המערכת

1.1.2 היקף שעות

1.1.3 מבנה וארגון

1.1.4 משימות המערכת

1.2 תיאור חומרה

1.3 תיאור תוכנת המערכת

1.4 תיאור פונקציות המערכת

### 2. ממשקים חיצוניים

### 3. ממשק אדם מכונה

3.1 כללי

3.2 תיאור מסכים

### 4. ארגון קבצים ומבנה נתונים

4.1 ארגון קבצים

4.2 מבנה נתונים

### 5. תכנון

5.1 כללי

5.2 עקרונות התכנות

5.3 תיאור אלגוריתמים

5.4 בדיקות המערכת

### 6. מה הקנה הפרויקט

### 7. ביבליוגרפיה



## הגדרת דרישות ותיאור כללי.

### 1.1 תיאור כללי

פרויקט זה נועד לתווך בין אנשים המעוניינים להעביר משלוחים לאזורים מסוימים בארץ, ע"י אנשים פרטיים שנוסעים באופן קבוע מאזור מגוריהם/ אזור הקרוב אליהם, לאזור הרצוי למשלוח ומעוניינים בהתנדבות העברת משלוחים. הלקוחות והשליחים יירשמו לאתר ע"י הזנת פרטים אישיים, בנוסף שליח יזין גם את פרטי הנסיעה. בחיפוש שליח: הלקוח יזין את פרטי השליחות ומיקום רצוי למפגש עם השליח, המערכת תבצע חיפוש המתאים ללקוח. ותשלח מיילים לשליחים המתאימים. המשתמשים יכולים בכל עת לצאת מהאתר לצמיתות - להימחק מרשימת הלקוחות/ השליחים. כששתמש נרשם/ נמחק מהאתר מקבל הודעה מתאימה למייל.

#### 1.1.1 מטרות הפרויקט:

- שליחות מהירה.
- חסכון בדמי משלוח.
- ריכוז גדול של שליחים ליעדים רבים בארץ.

#### 1.1.2 היקף שעות:

770 שעות.

#### 1.1.3 מבנה וארגון:

- ממשק למשתמש שדרכו מבצע את כל הפעולות באתר.
- ממשק למנהל שדרכו יכול לבצע שינויים במסד הנתונים.

#### 1.1.4 משימות המערכת:

- מציאת שליח מתאים ללקוח ע"י סינונים בטבלת חיפוש.
- שליחת מייל לשליחים המתאימים.
- כניסה לאתר באמצעות מייל וסיסמה.
- בכניסת מנהל: הוספה, עדכון ומחיקה של נתונים.
- בכניסת משתמש:
  - עדכון
  - מחיקה



## 1.2 תיאור חומרה

מחשב PC עם התאמה לדפדפנים Chrome, Explorer.

## 1.3 תיאור תוכנת המערכת

### 1.3.1 כללי

- שרת - מנוהל ע"י WEB API ונכתב בשפת #C, ASP.NET, גישה למסד הנתונים באמצעות DB FIRST- Entity Framework
- מסד נתונים - מסד הנתונים הוא ב- SQL SERVER.
- צד לקוח - נכתב בטכנולוגית Angular 9.

### 1.3.2 כלי התוכנה לפיתוח המערכת

כלים למפתחים: Visual Studio, Visual StudioCode, SQL Servr  
שפות פיתוח: Angular 9, CSS, HTML, TS, C#

## 1.4 תיאור פונקציות המערכת

### פונקציות בכניסת מנהל

הפונקציות הבסיסיות ממומשות בכניסת המנהל לצורך שליפת נתונים על שליחים, לקוחות ועוד.  
ובן לצורך עדכון, הוספה ומחיקה של הנתונים.

#### טבלת שליחים

עריכת משתמש קיים ומחיקת משתמש.  
פונקציות נוספות:  
edit() – עריכת משתמש קיים.  
remove() – מחיקת משתמש.

#### טבלת לקוחות

עריכת משתמש קיים ומחיקת משתמש.  
edit() – עריכת משתמש קיים.  
remove() – מחיקת משתמש.



### טבלת תגובות

למנהל יש אפשרות ניהול תגובות, משתמש מכניס תגובה למערכת, התגובה עוברת למערכת זמני עד אישור מנהל, באישור מנהל התגובה עוברת למערכת הראשי ונמצאת במערכת.

removeFormTemp() – מחיקה מהמערכת הזמני.  
removeFromMain() – מחיקה מהמערכת הראשי.  
transFromTempToMain() – העברה מהמערכת הזמני לראשי.  
transFormMainToTemp() – העברה מהמערכת הראשי לזמני.

### **פונקציות בבניסת משתמש**

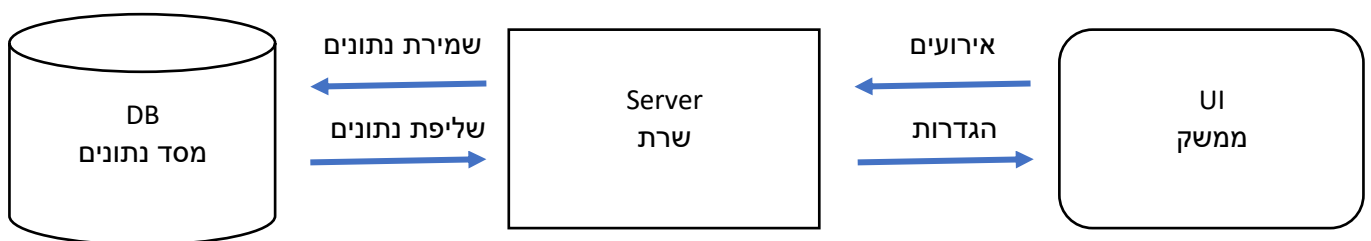
- signIn() – בודקת שהמייל והסיסמא קיימים ותואמים זה לזה.
- UpdateClient() / UpdeteDeliv() – עדכון פרטים אישיים שליח/לקוח.
- שומרת את הקוד משתמש לשימוש.
- Sure() – מחיקת עצמאית של המשתמש מה-DB.
- onSubmit() – רישום משתמש למערכת, לאחר בדיקה אם המשתמש קיים.

### **פונקציות בחיפוש שליח**

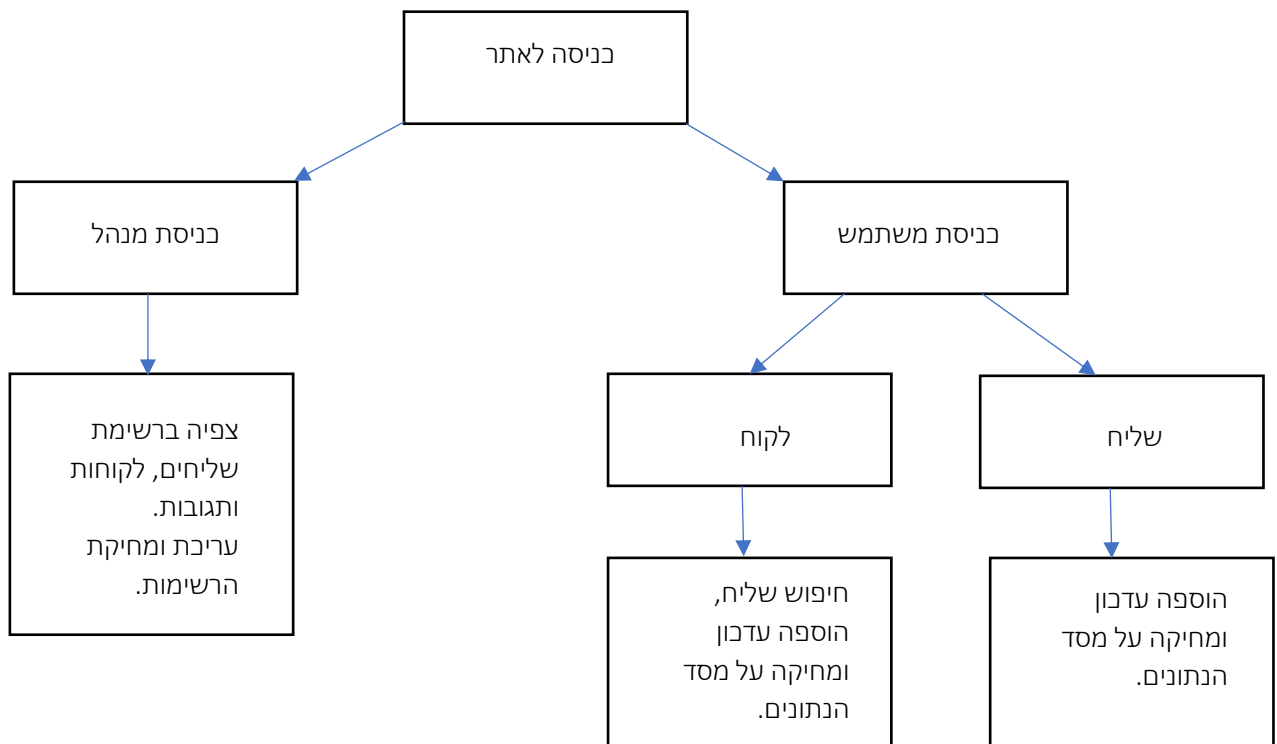
- sortAllDeliv() – חיפוש שליח לפי קריטריונים המתאימים ללקוח.

## **1.4.1 תרשימים**

תרשים זרימה בסיסי על התנהגות המערכת.



תרשים זרימה בסיסי בממשק משתמש:



## 2. ממשקים חיצוניים

התממשקות עם GoogleMapsAPI.

## 3. ממשק אדם מכונה

3.1 כללי

האתר פותח תוך שמירה על ממשק נוח וידידותי ככל שניתן על מנת לספק למשתמש חווית שימוש קלה וברורה, ורצון לחזור ולהשתמש בשירותי האתר.

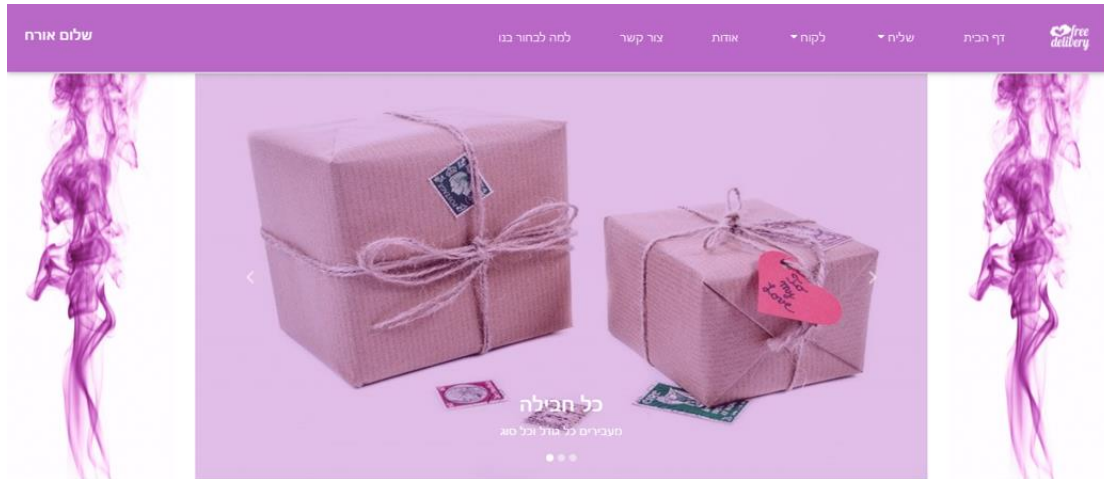




## 3.2 תיאור מסכים

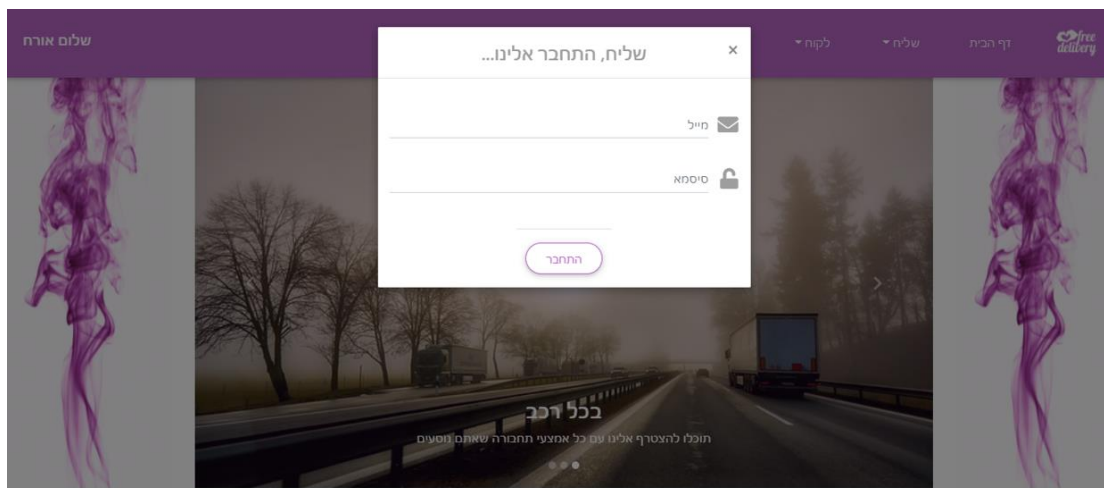
להלן צילומי המסכים השונים ותיאורם-

### ○ דף הבית



### ○ Login – מסך הכניסה

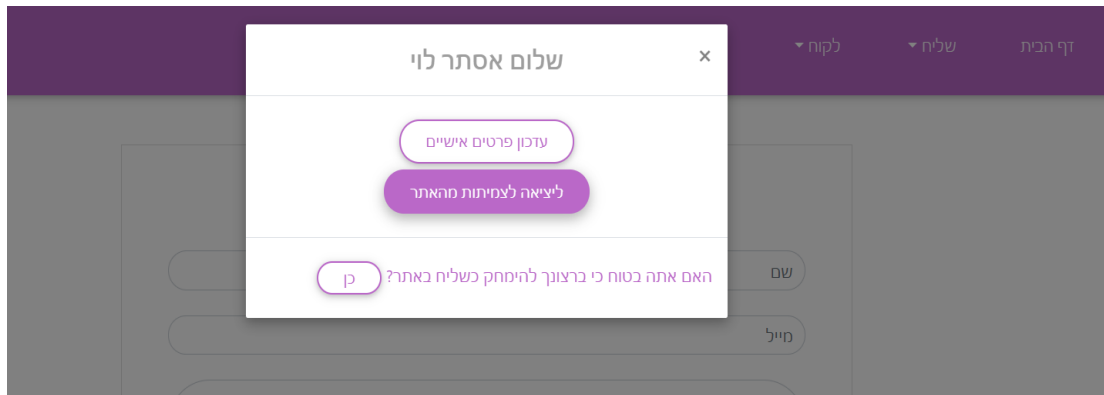
במסך הכניסה על המשתמש להתחבר למערכת באמצעות כתובת מייל וסיסמא שבחר בעת יצירת החשבון. כתובת מייל וסיסמא שמוגדרות במסד הנתונים כמנהל יועברו לדף כניסת מנהל.



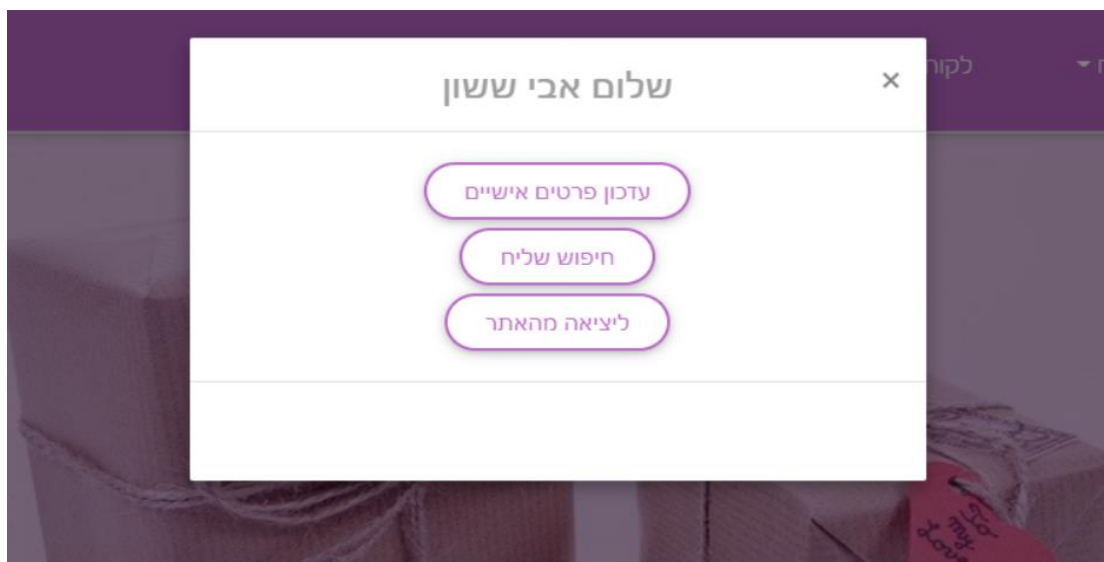




- יצאה לצמיתות מהאתר (מחיקת משתמש מה-DB)  
בלחיצה על יציאה, המשתמש יישאל האם הוא בטוח כי ברצונו להימחק כשליח/ כלקוח באתר בהתאם לכניסת המשתמש.



- ללקוח יש אפשרות גם לחפש שליח מתאים.



## ○ חיפוש שליח

בדף זה הלקוח מבצע חיפוש שליח לפי קריטריונים וסינונים המתאימים לו ע"י לחיצה על כפתור 'חיפוש שליח לפי קריטריונים' ולאחר מכן בטבלה רואה את השליחים המתאימים לו.

דף הבית

שליח

לקוח

אודות

צור קשר

למה לבחור בנו

עדכון שלוחות שבוצעה

free delivery

חיפוש שליח לפי קריטריונים

מיקום יציאה	שעת יציאה	מיקום הגעה	שעת הגעה	מרחק בק"מ	גודל חבילה	שביר / לא שביר	כמות שלוחות שביצע	שלח הצעה למייל
בית הדפוס 11	PM 12:00	גולדה מאיר 125	PM 4:02		ללא הגבלה	true	1	<input type="checkbox"/>
פינסקר 11	AM 4:00	אביעדר ילן 8	AM 5:00		עד 1 ק"ג	true	1	<input type="checkbox"/>
ליאון בלום 8	AM 9:00	פינסקר 4	AM 10:30		עד 2 ק"ג	false		<input type="checkbox"/>
פנסק 4	AM 1:00	לוינסקע 8	AM 2:00		עד 2 ק"ג	false	1	<input type="checkbox"/>
אחיסטך 10 תל אביב, ישראל	PM 2:00	הרב אלמושנינו 3	PM 3:00		ללא הגבלה	false		<input type="checkbox"/>
עמנואל הרמי, תל אביב, ישראל	AM 11:00	ארהוביץ' 11, תל אביב, ישראל	AM 12:00		ללא הגבלה	true		<input type="checkbox"/>

שלח הצעות לשליחים המסומנים

שלח הצעות לכל השליחים

## חיפוש שליח

תאריך רצוי למשלוח

מיקום יציאה  
מיקום הגעה

הגדר מרחק רצוי (ק"מ)

0 5 10

חבילה שבירה ?

בחר משקל חבילה



סגור

חפש



## פירוט חיפוש שליח לפי קריטריונים:

הקריטריונים לחיפוש הם:

תאריך רצוי למשלוח:

תאריך רצוי למשלוח 

מיקומי יציאה והגעה, שעות יציאה והגעה.  
הלקוח מזין את המיקום שממנו הוא מעוניין לשלוח את החבילה ואת המיקום שאליו הוא שולח. בנוסף מזחן את השעות יציאה והגעה.

---		מיקום יציאה	
---		מיקום הגעה	

מרחק רצוי:

בשדה זה הלקוח יכול לבחור עד איזה מרחק מעוניין ממיקום יציאתו למיקום יציאה של השליח.

הגדר מרחק רצוי (ק"מ)

0 5 10



לאחר מכן הלקוח מגדיר את פרטי החבילה שלו:

חבילה שבירה:

מסמן האם החבילה שברצונו לשלוח שבירה או לא. אם החבילה שבירה יוצגו לפניו רק שליחים שמוכנים להעביר חבילות שבירות ואם לא שבירה, יוצגו שני סוגי השליחים(שמסכימים) וחבילה שבירה ושאינם מסכימים לה)

☐ חבילה שבירה ?

משקל חבילה:

בוחר את המשקל של החבילה ומוצגים לפניו כל השליחים שמסכימים למשקל זה ולמשקל הנמוך ממנו.

בחר משקל חבילה



## ○ מסך שליחויות שבוצעו

זהו סקר שנשלח ללקוח יום לאחר בקשת השליחות שלו(פירוט קוד נמצא בהמשך).

שלום אורח

למה לבחור בנוצור קשראודותלקוחשליחדף הביתfree delivery

סקר ביצוע שליחות...

כן

לא

האם השליחות שביקשת בוצעה?

לא נמצא שליח פנוי.

נמצא שליח חיצוני.

לא היה צורך בשליחות.

אחר

שלח

© 2020 freedelivery : Hila & Yaeli

שלום אורח

למה לבחור בנוצור קשראודותלקוחשליחדף הביתfree delivery

סקר ביצוע שליחות...

כן

לא

האם השליחות שביקשת בוצעה?

שם הלקוח: Daniel cohen

תאריך: 2020-12-25

שעות הגעת השליחות:

שם השליח:

הערות:

שלח

© 2020 freedelivery : Hila & Yaeli

## ○ מסך בניסת מנהל

בניסה למסך זה מתאפשרת רק למשתמשים בעלי הרשאת מנהל. כאן יכול המנהל לבצע פעולות עדכון, הוספה, ומחיקה של נתונים מה-DB בטבלאות השונות.





ניהול תמונות

תמונות זמניות

תמונות לתצוגה

קוד ID	תמונה ID	מחיקה	העברה
7	טוב מאוד תהליך הזמנה מהיר		
8	אתר מעולה תודה רבה.		
11	מאד מאוד מרוגז!!		
12	משלוחים חנים בכל הארץ! מעולה!!!		
13	יופי של אתר		

קוד ID	תמונה ID	מחיקה	העברה
23	xx		
27	מועיל מאוד		

שלום מנהל

למה לבחור בנו

צור קשר

אודות

לקוח

שלח

דף הבית

freedelivery

שליחים

חיפוש

קוד ID	ID שליח	שם שליח	טלפון שליח	תאריך לידה	מייל שליח	כמות שליחויות	מחיקה	עריכה
13	000000000	יחזקה כוי	0545874521	09/02/1995	100hila100@gmail.com	1		
22	147852693	שלים אא	0526599753	01/01/1999	212moran@gmail.com	5		
32	147852369	yyyy yyyy	0545236985	10/10/1987	ilanaaishah@gmail.com			
33	000000018	אור קדוש	0525459874	05/11/1980	freedelivery66@gmail.com			
34	000000042	שלים אא	0526599753	06/11/1998	100hila100@gmail.com			
38	000000026	שלימוי אבי	0526599753	09/11/1985	drimoonlights@gmail.com	3		
39	000000059	xxx xxxxxxx	0526599753	12/12/1988	ilanaaishah@gmail.com			
40	000000034	רויט ממן	0526599753	05/06/1972	drimoonlights@gmail.com	5		

© 2020 freedelivery : Hila & Yaeli

אודות

הסבר אודות האתר לשם מה הוקם, ומה הרעיון עומד מאחוריו.



## freeDelivery אודות

בתקופה שבה החל נגיף הקורונה להשתולל בארץ, אנשים רבים נותרו בבתים, לא ניתן היה לנסוע ממקום למקום, מקומות ציבוריים רבים נסגרו וביניהם שירותי הדואר הרבים, וכך נוצר מצב שבו אנשים שרצו להעביר חבילות, מכתבים וכד' לקרוביהם או לכל צורך אחר לא יכלו לעשות זאת, אם מהסיבה שפרנסתם נפגעה ואם מהסיבה שלא יכלו לצאת מבתיכם.

בעקבות כל זאת החלטנו להקים את free delivery, שפועל 24 שעות ביממה באופן שבו כל אדם שקיבל אישורי נסיעה מיוחדים ברחבי הארץ יכול להירשם לאתר בתור משלוחן מתנדב וכל אדם שזקוק להעברת משלוח בין המקומות שהוא נוסע בהם יוכל להיעזר בו ולהיות בטוח שהחבילה תגיע ליעדה בזמן וביום שנקבע.

כמו כן אתר זה נועד לשמור על ערך החסד ולהראות לעולם שלמרות כל מה שקורה מסביב, עדיין אנשים מעוניינים לעזור לכל אדם באשר הוא, גם אם הוא לא מכיר וכלי לצפות לתגמול בעד מעשיו.

לסיום נדגיש כי לאחר שיעבור משבר הקורונה אתרינו יפעל ביתר יעילות כי יהיו מתנדבים רבים יותר ולא רק בודדים שקיבלו אישור מיוחד...

## יצירת קשר

משתמש באתר יכול ליצור קשר עם מנהל האתר באמצעות שליחת מייל אליו.

### צור קשר

שם

מייל

נושא

הודעה

שלח



## ארגון קבצים ומבנה נתונים.

### 4.1 ארגון קבצים

- כל רכיבי המערכת נבנו בארכיטקטורת אנגולר.
- לכל רכיב במערכת יש תיקיה נפרדת המכילה:
  - קובץ CSS – האחראי על העיצוב.
  - קובץ HTML – האחראי על המבנה הוויזואלי.
  - קובץ (TS) Typescript – האחראי על הלוגיקה והבדיקות.
- כל רכיבי המערכת מוכלים במודול ראשי המנהל אותם.

### 4.2 מבנה נתונים

מסד הנתונים מכיל את כל הטבלאות המשמשות את הפרויקט.  
כל טבלה מכילה שדה מפתח למניעת כפילות נתונים, ולשליפה קלה ומהירה של הנתונים.  
בכל טבלה השדות מוגדרות לפי סוג הטיפוס של הנתונים.  
שליפת הנתונים מתבצעת לפי השמות של השדות מתוך מסד הנתונים המנוהל במערכת SQL Server.

○ טבלת שליחים

	Column Name	Data Type	Allow Nulls
🔑	code_delivery	int	<input type="checkbox"/>
	first_name_delivery	varchar(20)	<input checked="" type="checkbox"/>
	last_name_delivery	varchar(20)	<input checked="" type="checkbox"/>
	bornDate	date	<input checked="" type="checkbox"/>
	id_delivery	varchar(10)	<input checked="" type="checkbox"/>
	password	varchar(20)	<input checked="" type="checkbox"/>
	phone	varchar(10)	<input checked="" type="checkbox"/>
	email	varchar(50)	<input checked="" type="checkbox"/>
	max_weight	float	<input checked="" type="checkbox"/>
	isBreak	bit	<input checked="" type="checkbox"/>
	amount_of_sent	int	<input checked="" type="checkbox"/>



○ טבלת לקוחות

	Column Name	Data Type	Allow Nulls
🔑	code_client	int	<input type="checkbox"/>
	first_name_client	varchar(20)	<input checked="" type="checkbox"/>
	last_name_client	varchar(20)	<input checked="" type="checkbox"/>
	password_client	varchar(20)	<input checked="" type="checkbox"/>
	phone	varchar(20)	<input checked="" type="checkbox"/>
	bornDate	date	<input checked="" type="checkbox"/>
	id_client	varchar(10)	<input checked="" type="checkbox"/>
	email	varchar(50)	<input checked="" type="checkbox"/>
	code_package	int	<input checked="" type="checkbox"/>

○ טבלת נסיעות

	Column Name	Data Type	Allow Nulls
🔑	code_travel	int	<input type="checkbox"/>
	code_delivery	int	<input checked="" type="checkbox"/>
	leaving_address	varchar(50)	<input checked="" type="checkbox"/>
	arrival_address	varchar(50)	<input checked="" type="checkbox"/>
	leaving_time	varchar(50)	<input checked="" type="checkbox"/>
	arrival_time	varchar(50)	<input checked="" type="checkbox"/>
	lat_leaving_address	decimal(8, 6)	<input checked="" type="checkbox"/>
	lng_leaving_address	decimal(8, 6)	<input checked="" type="checkbox"/>
	lat_arrival_address	decimal(8, 6)	<input checked="" type="checkbox"/>
	lng_arrival_address	decimal(8, 6)	<input checked="" type="checkbox"/>
	bus	bit	<input checked="" type="checkbox"/>
	car	bit	<input checked="" type="checkbox"/>
	bicycle	bit	<input checked="" type="checkbox"/>
	train	bit	<input checked="" type="checkbox"/>
	walk	bit	<input checked="" type="checkbox"/>
	motorcycle	bit	<input checked="" type="checkbox"/>

○ טבלת כלי תחבורה

	Column Name	Data Type	Allow Nulls
🔑	code_transport	int	<input type="checkbox"/>
	description	varchar(20)	<input checked="" type="checkbox"/>



○ טבלת פרטי חבילה

	Column Name	Data Type	Allow Nulls
🔑	code_package	int	<input type="checkbox"/>
	name_package	varchar(20)	<input checked="" type="checkbox"/>
	description_package	varchar(1000)	<input checked="" type="checkbox"/>
	weight	float	<input checked="" type="checkbox"/>
	image	varchar(100)	<input checked="" type="checkbox"/>
	isBreak	bit	<input checked="" type="checkbox"/>

○ טבלת משקל ותיאור חבילה

	Column Name	Data Type	Allow Nulls
🔑	code_weight	int	<input type="checkbox"/>
	desc_weight	nvarchar(50)	<input checked="" type="checkbox"/>

○ טבלת ארכיון שליחים

	Column Name	Data Type	Allow Nulls
🔑	code_delivery	int	<input type="checkbox"/>
	first_name_delivery	varchar(20)	<input checked="" type="checkbox"/>
	last_name_delivery	varchar(20)	<input checked="" type="checkbox"/>
	bornDate	date	<input checked="" type="checkbox"/>
	id_delivery	varchar(10)	<input checked="" type="checkbox"/>
	password	varchar(20)	<input checked="" type="checkbox"/>
	phone	varchar(10)	<input checked="" type="checkbox"/>
	email	varchar(50)	<input checked="" type="checkbox"/>
	max_weight	float	<input checked="" type="checkbox"/>
	isBreak	bit	<input checked="" type="checkbox"/>
	amount_of_sent	int	<input checked="" type="checkbox"/>



טבלת ארכיון לקוחות

	Column Name	Data Type	Allow Nulls
🔑	code_client	int	<input type="checkbox"/>
	first_name_client	varchar(20)	<input checked="" type="checkbox"/>
	last_name_client	varchar(20)	<input checked="" type="checkbox"/>
	password_client	varchar(20)	<input checked="" type="checkbox"/>
	phone	varchar(20)	<input checked="" type="checkbox"/>
	bornDate	date	<input checked="" type="checkbox"/>
	id_client	varchar(10)	<input checked="" type="checkbox"/>
	email	varchar(50)	<input checked="" type="checkbox"/>
	code_package	int	<input checked="" type="checkbox"/>

טבלת שליחויות שבוצעו

	Column Name	Data Type	Allow Nulls
🔑	code_delivered_package	int	<input type="checkbox"/>
	date_delivered_package	date	<input checked="" type="checkbox"/>
	time_delivered_package	time(7)	<input checked="" type="checkbox"/>
	id_delivaery_person	varchar(10)	<input checked="" type="checkbox"/>
	id_client	varchar(10)	<input checked="" type="checkbox"/>

טבלת תגובות משתמשים

	Column Name	Data Type	Allow Nulls
🔑	code_comment	int	<input type="checkbox"/>
	details_comment	varchar(100)	<input checked="" type="checkbox"/>

טבלת תגובות זמניות

	Column Name	Data Type	Allow Nulls
🔑	code_temporary_comme...	int	<input type="checkbox"/>
	details_temporary_com...	varchar(100)	<input checked="" type="checkbox"/>



○ טבלת מיילים

	Column Name	Data Type	Allow Nulls
🔑	emailCode	int	<input type="checkbox"/>
	mailTo	varchar(50)	<input checked="" type="checkbox"/>
	mailFrom	varchar(50)	<input checked="" type="checkbox"/>
	subject	varchar(50)	<input checked="" type="checkbox"/>
	body	varchar(200)	<input checked="" type="checkbox"/>

## 5. תכנון

### 5.1 כללי

האתר מחולק לצד שרת וצד לקוח.  
צד השרת נבנה ב-Web API בשפת C#.  
צד הלקוח נבנה בשפת typescript בטכנולוגיית Angular 9  
עם אלמנטים של HTML ועיצוב ב-CSS תוך שימוש בספריית bootstrap.

### 5.2 עקרונות התכנות

בפרויקט זה המטרה הייתה לכתוב קוד קצר ויעיל תוך שימוש מקסימאלי  
בטכנולוגיות העומדות לרשותנו בכלי התוכנה שבחרנו לפיתוח.  
הפרויקט נבנה תוך חלוקה בין ממשק המשתמש, התפקוד הלוגי של המערכת  
והתקשרות למערכות אחרות.





## 5.3 תיאור אלגוריתמים

להלן כמה קטעי קוד עיקריים:

- בעת רישום שליח/ חיפוש שליח, המשתמש יזין את אזור הנסיעה/ אזור מבוקש לו עם ממשק Google Maps. באנגולר-

```
findAddress() {
  this.mapsAPILoader.load().then(() => {

    let autocomplete = new google.maps.places.Autocomplete(this.searchElementRef.nativeElement, {
      types: ["address"],
    });

    let autocomplete1 = new google.maps.places.Autocomplete(this.searchElementRef1.nativeElement, { ...
  });

    autocomplete.addListener("place_changed", (event) => {
      this.ngZone.run(() => {
        let place: google.maps.places.PlaceResult = autocomplete.getPlace();
        let latitude = place.geometry.location.lat();
        let longitude = place.geometry.location.lng();
        // alert(latitude + " * " + longitude);
        this.travels[0].leaving_address = this.searchElementRef.nativeElement.value;
        this.travels[0].lat_leaving_address = latitude;
        this.travels[0].lng_leaving_address = longitude;
        alert(this.travels[0].leaving_address);
        alert("leaving"+this.travels[0].lat_leaving_address+" "+this.travels[0].lng_leaving_address);
      });
    });

    autocomplete1.addListener("place_changed", (event) => { ...
  });
});
}
```

- בעת חיפוש שליח הלקוח מוצא את השליח המתאימים לו. ומסמן V לאילו שליחים יישלח מייל שהם מתאימים לביצוע שליחותו. או לחלופין בוחר באפשרות לשליחת מיילים לכל השליחים. באנגולר-

```
addDeliveryEmailToEmailList($event, codeDel) {
  if ($event.target.checked === true) {
    var emailDel = new email();
    emailDel.mailFrom = "freedelivery8@gmail.com";
    this.delivService.getDelivery(codeDel).subscribe(
      success => {
        emailDel.mailTo = success.email;
        emailDel.subject = "שליחות לתאריך: " + this.mysearch.dated;
        emailDel.body = "שלום" + success.first_name_delivery + " " + success.last_name_delivery +
          "<br/>" + "נעבור לקוד/ה: " + this.cln.first_name_client + " " + this.cln.l
          ast_name_client +
          "<br/>" + "פלאפון: " + this.cln.phone +
          "<br/>" + "מייל: " + this.cln.email + "<br/>" + "freedelivery צוות";
        this.emailsDel.push(emailDel);
      });
  }
  else {
    this.delivService.getDelivery(codeDel).subscribe(
      success => {
        this.mailSearch = success.email;
        this.removeDeliveryEmailFromEmailList(this.mailSearch);
      });
  }
}

removeDeliveryEmailFromEmailList(item) {
  for (let i = 0; i < this.emailsDel.length; i++) {
    if (this.emailsDel[i].mailTo === item) {
      this.emailsDel.splice(Number(i), 1)
      break;
    }
  }
}
```



## בשרת-

```
[HttpPost]
public IActionResult SendEmailToList(int x, List<Email> userEmail)
{
    try
    {
        for (int i = 1; i < userEmail.Count; i++)
        {
            MailMessage mail = new MailMessage();
            mail.From = new MailAddress("freedelivery86@gmail.com");
            mail.Subject = userEmail[i].subject;
            mail.Body = userEmail[i].body;
            mail.To.Add(userEmail[i].mailTo);
            mail.IsBodyHtml = true;

            SmtpClient SmtpServer = new SmtpClient("smtp.gmail.com");
            SmtpServer.Port = 587;
            SmtpServer.UseDefaultCredentials = false;
            SmtpServer.EnableSsl = true;
            SmtpServer.Credentials = new System.Net.NetworkCredential("freedelivery86@gmail.com", "hilayael86");
            SmtpServer.Send(mail);
        }
        return Ok(true);
    }
    catch (Exception)
    {
        throw;
    }
}
```

- בעת ששליח מעדכן שליחות שבוצעה ע"י שליח מסוים. מתעדכן בטבלה של השליחים עוד שליחות למספר השליחויות שביצע. כך קל יותר ללקוח לבחור שליח שלא מכיר, אך רואה שביצע כבר כמה שליחויות. באנגולר-

```
onSubmit() {
    this.delivService.getDelivery(Number(this.codeDel)).subscribe(success => {
        this.myDeliv = success;
        alert(success.id_delivery);
        this.mydelPack.id_delivaery_person = success.id_delivery;
        this.delPackser.addDeleveryedPack(this.mydelPack).subscribe(succ => {
            this.openSuccessModal();
            this.router.navigate(['home']);
        })
    },
    error => {
        alert('ERROR!!');
    })
}
```



בגישה לנתוני השליח מה-client ל-server , כדי למצוא את השליח ולהוסיף לו לכמות שליחויות שביצע שליחות נוספת.

```
addDeleveriedPack(dp: deliveryPackages): Observable<deliveryPackages>{  
    return this.http.post<deliveryPackages>(this.baseUrl, dp);  
}
```

בשרת-

```
[HttpPost]  
0 references  
public IActionResult Post([FromBody] delivered_packages delivery_pack)  
{  
    try  
    {  
        var deliv_pack = db.delivered_packages.Add(delivery_pack);  
        db.SaveChanges();  
        AddAmountOfSent(delivery_pack.id_delivaery_person);  
        return Ok(deliv_pack);  
    }  
    catch  
    {  
        return InternalServerError();  
    }  
}  
[HttpPost]  
1 reference  
public IActionResult AddAmountOfSent(string idDel)  
{  
    delivery_person del = db.delivery_person.FirstOrDefault(x => x.id_delivery == idDel);  
    if (del == null)  
    {  
        return InternalServerError();  
    }  
    if (del.amount_of_sent == null)  
        del.amount_of_sent = 1;  
    else  
        del.amount_of_sent++;  
    db.SaveChanges();  
    return Ok(del);  
}
```



- בעת מחיקת שליח, נמחקות גם כל הגדרות נסיעותיו הקבועות שלו.  
בשרת- מחיקת שליח.

```

DetailsDelivController.cs* X
freeDelivery-server freeDelivery_server.Controllers.DetailsDelivController
125
126 [HttpDelete]
127 0 references
128 public IActionResult Delete(int id)
129 {
130     DeliveryArchiveController delArch = new DeliveryArchiveController();
131     //int empId = Convert.ToInt32(id);
132     delivery_person del = db.delivery_person.Find(id);
133     if (del == null)
134     {
135         return NotFound();
136     }
137     delArch.Post(del);
138     db.delivery_person.Remove(del);
139     db.SaveChanges();
140     return Ok(del);
141 }
142
143
144
145
146

```

בשרת- מחיקת הגדרת נסיעות קבועות של השליח.

```

TravelController.cs* X
freeDelivery-server freeDelivery_server.Controllers.TravelController
119
120 [HttpDelete]
121 0 references
122 public IActionResult Delete(int deliveryId, int x)
123 {
124     List<travel> t = new List<travel>();
125     t = db.travels.Where(c => c.code_delivery == deliveryId).ToList();
126     if (t == null)
127     {
128         return NotFound();
129     }
130     for (int i = 0; i < t.Count; i++)
131     {
132         db.travels.Remove(t[i]);
133         db.SaveChanges();
134     }
135     return Ok();
136 }
137
138
139

```



באנגולר-  
מחיקת שליח דרך בניסת מנהל.

```
remove(deliveryId: number) {  
  if (confirm("Are you sure you want to delete this ?")) {  
    this.delivService.deleteDelivery(deliveryId).subscribe(() => {  
      this.travelService.deleteTravelByDeliveryId(deliveryId).subscribe()  
      this.loadAllDelivery();  
    });  
  }  
}
```

מחיקת שליח דרך השליח בעצמו.

```
Sure() { //מחיקת שליח מרשימת השליחים  
  sessionStorage.removeItem('codeclient');  
  var codedel = Number(sessionStorage.getItem("codedeliv"));  
  this.delivService.deleteDelivery(codedel).subscribe(success => {  
    this.travelService.deleteTravelByDeliveryId(codedel).subscribe()  
    this.send();  
    sessionStorage.removeItem('codedeliv');  
    alert("delete!!");  
  }, error => {  
    alert("not delete!!");  
  })  
}
```

גישה לנתוני השליח מה- client ל-server באמצעות שימוש בפונקציה מתוך קובץ הservice של שליח.

```
deleteDelivery(id: number): Observable<number> {  
  const httpOptions = { headers: new HttpHeaders({ 'Content-Type': 'application/json' }) };  
  return this.http.delete<number>(this.baseUrl + '/Delete?id=' + id,  
    httpOptions);  
}
```

```
deleteTravelByDeliveryId(deliveryId: number) {  
  const httpOptions = { headers: new HttpHeaders({ 'Content-Type': 'application/json' }) };  
  return this.http.delete<number>(this.baseUrl + '/Delete?deliveryId=' + deliveryId + '&x=0',  
    httpOptions);  
}
```



- לאחר שלקוח מגיש בקשה למשלוח פרטיו נשמרים במערכת כדי שיום לאחר התאריך שביקש המערכת תשלח אליו מייל אוטומטי כדי שימלא סקר אודות ביצוע השליחות(נמצא בפירוט המסכים) ע"י שימוש ב- Quartz.
- הפונקציה נמצאת בשרת, מופעלת כל יום בשעה קבועה ובודקת האם קיים לקוח שביקש שליחות אתמול:

```

8 namespace freeDelivery_server.Models
9 {
10     2 references
11     public class Scheduler24
12     {
13         1 reference
14         public async void Start()
15         {
16             StdSchedulerFactory factory = new StdSchedulerFactory();
17
18             // get a scheduler
19             IScheduler scheduler = await factory.GetScheduler();
20             await scheduler.Start();
21
22             // define the job and tie it to our HelloJob class
23             IJobDetail job = JobBuilder.Create<SendEmailFeedback>()
24                 .WithIdentity("myJob", "group1")
25                 .Build();
26
27             ITrigger trigger = TriggerBuilder.Create()
28                 .WithDailyTimeIntervalSchedule
29                 (s1 =>
30                     s1.WithIntervalInHours(24)
31                     .OnEveryDay()
32                     .StartingDailyAt(TimeOfDay.HourAndMinuteOfDay(8, 0))
33                 )
34                 .Build();
35
36             await scheduler.ScheduleJob(job, trigger);
37         }
38     }

```

```

SendEmailFeedback.cs* X
freeDelivery-server freeDelivery_server.Models.SendEmailFeedback send()
1 reference
37 private async Task send()
38 {
39     var yesterday = DateTime.Today.AddDays(-1);
40     List<package> packages = new List<package>();
41     packages = GetPacks();
42
43     for (int i = 0; i < packages.Count; i++)
44     {
45         if (packages[i].send_date == yesterday.Date)
46         {
47             int code = packages[i].code_client.Value;
48             client myClient = new client();
49             myClient = getClient(code);
50             if (myClient != null)
51             {
52                 var myLink = "http://localhost:4200/update-package-delivery?codepackage=" + packages[i].code;
53                 string strBody3 = "<a href=" + myLink + "> רחץ כאן </a>";
54                 MailMessage mail = new MailMessage();
55                 mail.From = new MailAddress("freedelivery8@gmail.com");
56                 mail.Subject = "סקר אודות בקטת טריזיות";
57                 mail.Body = "טריז " + myClient.first_name_client + " " + myClient.last_name_client +
58                     "<br/>אנו מודים לך שבחרת להמשיך במסרוננו!<br/>" +
59                     "נשמח אם תשקיע זמן קצר לסטנה של מספר שאלות שיתנו לנו משוב על השליחים והטריזיות!<br/>" +
60                     "אנו מודים לך מראש על שיתוף הפעולה, ומחייבים לקרוא את המשוב שלך בדיוקנות!<br/>" +
61                     "דיווח!<br/>" + "בברכה" + " " + strBody3 + " " + "לכניסה לסקר!<br/>";
62
63                 mail.To.Add(myClient.email);
64                 mail.IsBodyHtml = true;
65
66                 SmtplibClient smtpServer = new SmtplibClient("smtp.gmail.com");
67                 smtpServer.Port = 587;
68                 smtpServer.UseDefaultCredentials = false;
69                 smtpServer.EnableSsl = true;
70                 smtpServer.Credentials = new System.Net.NetworkCredential("freedelivery8@gmail.com", "hlayael186");
71                 smtpServer.Send(mail);
72             }
73         }
74     }
75
76     0 references
77     public async Task Execute(IJobExecutionContext context)
78     {
79         await send();
80     }
81 }

```



## 5.4 בדיקות המערכת

ערכנו בדיקות גם בתחום הוויזואלי וגם בתחום הלוגי למניעת טעויות ונפילות. בתחום של הקוד נעשו ולידציות על שדות קלט למניעת הבנסת ערכים לא תקינים (דבר שיכול לגרם לשגיאות). לאורך כל תהליך הפיתוח בדקנו במה אפשר לשפר ולייעל כמה שיותר את האתר ולהימנע ממקרים של נפילות וטעויות.

## 6. מה הקנה הפרויקט

- ניסיון בתכנות ואפיון מערכת מהבסיס ועד לשלב הגמר.
- הבנת התקשורת בין צד שרת לצד לקוח.
- ניסיון מעשי בבניית שרת Web API.
- ניסיון מעשי בעבודה עם Entity Framework.
- ניסיון מעשי בבניית מסד נתונים ב SQL Server .
- ניסיון מעשי בבניית פרויקט בטכנולוגית Angular.
- למידה מעמיקה בכתיבת דפי HTML ושימוש ב- CSS.
- פיתוח יכולת עצמי, ניסיון בחיפוש באינטרנט ומציאת פתרונות לבעיות.
- פיתוח יכולת חשיבה לוגית.





## 7. ביבליוגרפיה

[/https://stackoverflow.com](https://stackoverflow.com)

[/https://mdbootstrap.com](https://mdbootstrap.com)

[/https://www.w3schools.com](https://www.w3schools.com)

[/https://jasonwatmore.com](https://jasonwatmore.com)

[/https://www.c-sharpcorner.com](https://www.c-sharpcorner.com)

[/https://coryryan.com](https://coryryan.com)

[/https://www.npmjs.com](https://www.npmjs.com)

[/https://angular.io](https://angular.io)

[/com.https://www.itsolutionstuff](https://www.itsolutionstuff.com)

[/angular-tutorials-https://www.positronx.io/free](https://www.positronx.io/free-angular-tutorials)

<https://medium.com/@dhormale/use-google-places-api-autocomplete-using-angular-for-resident-and-office-address-23cc33078e8>

