

DeliveryWay

ספר פרויקט

מגישה:

דבורה אטב

שם המנחה:

רחל דרבקין

הצעת פרויקט – יד הנדסת תוכנה

סמל מוסד : 716605

שם מכללה : בית יעקב פרי תואר אלעד

שם הסטודנט : דבורה גילה אטב

ת"ז הסטודנט: 323686881

שם הפרויקט : DeliveryWay

תיאור הפרויקט :

ניהול חלוקת משלוחים עבור חברת מכר מרחוק.

המערכת אוגרת נתוני הזמנות הלקוחות מרחבי העיר, ומחלקת את העבודה בצורה המיטבית בין המשלוחנים שהיא מעסיקה,

אלגוריתם חכם ויעיל יחשב מסלול קצר ואופטימלי עבור כל משלוחן, בשאיפה להותיר שליחים ללא מסלול ולהקטין את כח האדם.

וכך ההזמנות יגיעו ללקוחות במהירות וללא בזבז אנרגיות, כסף, זמן ודלק מיותר.

הגדרת הבעיה האלגוריתמית:

חלוקת יעדים לקבוצות קרובות המתאימות למספר השליחים שהוזן או קטנות ממנו,

חלוקת יעדים למספר מסלולים אופטימליים. על פי המרחקים בין היעדים, מספר השליחים, אורך יום העבודה של כל שליח, וחישוב הזמן שלוקח לחלק חבילה בודדת.

חישוב מסלול אופטימלי לכל קבוצה.

בחישוב מסלול, נצטרך להחליט כמה נקודות ביניים יהיו בחישוב, וכמה פעמים הוא ישתנה במהלך הדרך.

רקע תיאורטי בתחום הפרויקט :

קיימות כיום בשוק מספר אפליקציות המחשבות מסלול קצר ביותר עבור מספר יעדים,

אולם לא מצאנו אחת כזו המסוגלת לנהל חלוקה למספר שליחים בניצול מקסימלי של זמן המשמרת של כל אחד ואחד מהם.

היתרון של האפליקציה שלנו, שהיא מסוגלת, בנוסף לחישוב המסלול הבודד, לחלק את היעדים למספר מסלולים אופטימליים. על פי המרחקים בין היעדים, מספר השליחים, אורך יום העבודה של כל שליח, וחישוב הזמן שלוקח לחלק חבילה בודדת.

תהליכים עיקריים בפרויקט :

- המשתמש מכניס נקודות חלוקת משלוחים
- המשתמש מזין עבור כל חבילה זמן משלוח משוער או משאיר את זמן ברירת המחדל
- המשתמש מכניס את מספר השליחים ואורך יום העבודה שלהם
- המערכת מבצעת חישוב וחלוקה של מסלולים
- המערכת מציגה מסלול ייעודי עבור כל שליח

תיאור הטכנולוגיה :

צד שרת :

c#

צד לקוח:

angular

שפת תכנות בצד הלקוח:

html+tip script

מסד נתונים : mongo db

פרוטוקולי תקשורת:

http

לוחות זמנים:

מתאריך 01/12/2021 עד תאריך 01/01/2022 חקר הפרויקט וכתיבת הצעה.

מתאריך 01/01/2022 עד תאריך 01/02/2022 כתיבת מבניות הפרויקט.

מתאריך 01/02/2022 עד תאריך 01/03/2022 הקמת מסד נתונים.

מתאריך 01/04/2022 עד תאריך 01/05/2022 כתיבת האלגוריתם.

מתאריך 01/05/2022 עד תאריך 01/06/2022 הקמת ממשק משתמש.



חתימת הסטודנט :

חתימת רכז המגמה :

אישור משרד החינוך:

תקציר

ל DeliveryWay קיימות שתי פלטפורמות, אתר ואפליקציה.

אתר המיועד עבור מנהל, בעל עסק עם משלוחים להפצה.

האתר מאפשר למנהל לנהל את מאגר ההזמנות והפצתם למשלוח במהירות ויעילות מרבית.

אפליקציה המיועדת עבור שליח, האחראי על הפצת המשלוחים שהוקצו עבורו לחלוקה במסלול היעיל ביותר.

על פי דרישת העסק המספק סחורות בסדרי גודל קטנים בנפחם, בעת חישוב המסלול לא נלקח בחשבון גודל החבילה וגודל רכב השליח, מכיוון שאלו נתונים שאינם רלוונטיים עבורו יותר. כל השליחים של העסק בעלי רכב פרטי.

הן האתר והן האפליקציה נבנו תוך שימוש בטכנולוגיות מתקדמות על מנת לאפשר נוחות משתמש מהירות ויעילות והתאמה לצרכי הלקוח.

תוכן העניינים

1.	הגדרת דרישות ותיאור כללי	
1.1	תיאור כללי	
1.2	תיאור חומרת המערכת	
1.3	תיאור תוכנת המערכת	
1.4	תיאור פונקציות המערכת	
1.5	זרימת המידע במערכת	
2.	ממשקים חיצוניים	
3.	ממשק אדם ומכונה	
3.1	כללי	
3.2	תיאור מסכים	
4.	מבנה נתונים וארגון קבצים	
4.1	ארגון קבצים	
4.2	מבנה נתונים	
5.	תכנון	
5.1	כללי	
5.2	עקרונות התכנות	
5.3	תיאור אלגוריתמים	
5.4	בדיקת המערכת	
6.	מה הקנה הפרויקט	
7.	ביבליוגרפיה	

1. הגדרת דרישות ותיאור כללי

1.1 תיאור כללי

האתר הינו אתר ארגוני שמטרתו להקל על המנהל, בעל העסק בחלוקת ההזמנות לשליחים על פי אזורים ותכנון מסלול ההפצה המהיר ביותר עבור כל שליח.

וכן מתן פלטפורמה קלה ונוחה עבור השליח ללא צורך בבזבז זמן מיותר על חישוב מסלול או דיווח על חבילות שמסר ללקוח.

האתר יאפשר למנהל להוסיף הזמנה למאגר ההזמנות להפצה, להוסיף שליח למאגר השליחים, ולהפעיל מסלול.

בפלטפורמת האתר תינתן אפשרות למנהל לצפות בנתוני השליחים וכן אפשרות לסנן על פי שליחים פעילים / לא פעילים.

כמו כן המנהל יוכל לצפות כמה הזמנות קיימות במערכת ועדיין לא נשלחו להפצה וכן כמה הזמנות קיימות במערכת שנשלחו להפצה אך עדיין לא נמסרו ללקוח על מנת לקבל אינדיקציה האם "משתלם" להפעיל מסלול.

האפליקציה תאפשר לשליח לצפות בהזמנות שהוקצו עבורו למסירה על פי סדר המסלול שחושב עבורו.

שליח יוכל לנווט מתוך האפליקציה בקלות לכתובת המסירה ולאשר מסירת הזמנה.

כמו כן תינתן אפשרות לשליח לצפות בנתוניו האישיים וכן לעדכן עצמו כפעיל / לא פעיל ומספר פלאפון.

1.1.1 מטרת המערכת

- להקל על המנהל בחלוקת ההזמנות להפצה לשליחים על פי אזורים בקלות מהירות ויעילות מרבית.
- המנהל יוכל להיכנס לאתר, להכניס פרטי הזמנה, שליח, ולנהל את הפצת המשלוחים באופן חכם ויעיל.
- לאפשר לשליח הפצת ומסירת הזמנות בנוחות וקלות מרבית.

1.1.2 היקף הפרויקט

700 שעות

1.1.3 מבנה וארגון

הפרויקט מורכב משלושה חלקים:

DATABASE - שמירת נתונים של מנהל, שליחים, והזמנות

הנתונים מאוחסנים ב – MongoDB Atlas.

SERVER - שכבת DAL גישה לנתונים ושליפתם על ידי ENTITY CORE

גישה לקריאה, שמירה ועדכון של הנתונים על ידי WEB API.

CLIENT – כולל מסכים של המערכת, מימוש האתר ב ANGULAR והאפליקציה ב IONIC.

מודולים עיקריים:

הוספת שליח – מאפשר למנהל להוסיף שליח למאגר השליחים.

מאגר שליחים – מאפשר למנהל לצפות בפרטי שליחים קיימים, ולסנן על פי שליחים פעילים או לא פעילים.

הוספת הזמנה – מאפשר למנהל להוסיף הזמנה למאגר ההזמנות.

מאגר הזמנות – מאפשר למנהל לצפות במספר הזמנות הקיימות במערכת שנשלחו להפצה ועדיין לא

נמסרו ובמספר הזמנות הקיימות במערכת ועדיין לא הוקצו למסלול מסוים.

הפעלת מסלול – מאפשר למנהל להפעיל את מסלול ההפצה בקלות ללא צורך חישוב מצדו.

1.1.4 משימות המערכת

- מנהל יוכל להוסיף שליח למאגר השליחים.
- מנהל יוכל להוסיף הזמנה למאגר ההזמנות.
- שליח יוכל להירשם לאפליקציה לאחר שנוסף על ידי המנהל למאגר השליחים.
- שליח יוכל לעדכן פרטים אישיים.
- שליח יוכל לצפות בהזמנות שהוקצו עבורו להפצה על פי סדר.
- שליח יוכל לצפות בפרטי הזמנה ספציפית להפצה.
- שליח יוכל לנווט על ידי לחיצת כפתור לכתובת הלקוח.
- שליח יוכל לאשר מסירת הזמנה.

1.2 תיאור חומרת המערכת

1.2.1 כללי

להפעלת האתר נדרש מחשב עם דפדפן chrome המותקן עליו.

להפעלת האפליקציה נדרש סמארטפון,

Android – 5.0 + with chromium 60+

iOS – 13.0+

1.2.2 מרכיבי המערכת

מחשב PC עם התאמה לדפדפן CHROME

סמארטפון android או iOS עם התאמה לדפדפן CHROME + חנות אפליקציות.

1.3 תיאור תוכנת המערכת

1.3.1 כללי

בכתיבת פרויקט זה שמתי דגש רב על התאמה טכנולוגית מדויקת – שאיפה לטכנולוגיות חדשניות שיחד עם זאת לא תפגענה באיכות הפרויקט ותתאמה לדרישותיו.

שפות הפיתוח בצד השרת:

#C - שפה זו הינה שפת תכנות פופולארית שמיועדת לפיתוח כללי של מגוון אפליקציות.

התחביר והעקרונות שלה פשוטים מצד אחד, אך עשירים ביכולות מצד שני. תנאי הכרחי לשימוש במגוון טכנולוגיות מערכת חכמה והגייונית ומצד שני עשירה ביכולות מתקדמות #C הוא תנאי הכרחי לשימוש במגוון טכנולוגיות, מתוכן EDM – הפיכת מסד הנתונים למודל ישויות על ידי הפיכת טבלה למחלקה ושדה ל property ובצורה זו ניתן יהיה לשלוף/למחוק/לעדכן נתונים מהמסד נתונים.

שפות הפיתוח בצד הלקוח:

JavaScript – שפת תכנות דינמית מונחית עצמים, המותאמת לשילוב באתרי אינטרנט ורצה

על ידי דפדפן האינטרנט בצד הלקוח.

השפה מרחיבה את יכולות שפת התגיות הבסיסית HTML ומאפשרת בכך ליצור יישומי אינטרנט

מתוחכמים יותר.

5HTML – היא שפת בניית אתרים לסימון טקסט, שפת תגיות, אשר נותנת הנחיות לדפדפנים

בנוגע לאופן הצגתו של דף האינטרנט מבחינת טקסט, תמונות, טבלאות עיצוב ועוד. זוהי

שפת הקוד הבסיסית לסביבת ה WEB- עבור בניית אתרים.

5HTML כוללת כלים חדשים שבאו להקל על עבודת המפתחים.

Bootstrap - היא סביבת עבודה בקוד פתוח לצד לקוח, שמכילה אוסף של כלים ליצירת אפליקציות Web ואתרים.

Css - שפת עיצוב שמגדירה את תבנית העיצוב של מסמכי HTML. Css מטפלת בגופנים, צבעים, גבולות שורות,

גובה, רוחב, תמונת רקע, מיקום מתקדם ודברים רבים נוספים.

Angular - שפת פיתוח צד לקוח המוסיפה אבסטרקציות מעולם פיתוח Object Oriented, הכתיבה מתבצעת

ע"י כתיבת HTML שבתוכו משולבים תוספות של אנגולר, כתיבת components כדי לשלוט על ה- templates, כתיבה של services כדי לשלוט על הלוגיקה, ושילוב של components ו- services ליצירת modules.

1.3.2 כלי התוכנה לפיתוח המערכת

Visual Studio, Visual Studio Code, MongoDB Atlas, GIT, Web Api, Entity Core

שפות תכנות בשימוש-

Angular,Css,HTML5,TS,C#

1.4 תיאור מבנה הפרויקט

הארכיטקטורה הנבחרת היא חלוקה ל – 3 שכבות **3 Tier Architecture**

בפרויקטים שמבוססים על Database מומלץ להשתמש בגישה של מודל השכבות, בארכיטקטורה זו קיימת הפרדה בין השכבות השונות בפרויקט.

היתרון העיקרי בחלוקה לשכבות היא תחזוקה קלה, במיוחד במערכות גדולות, כך שניתן יחסית בקלות להחליף כל שכבה בלי לגעת בשכבות האחרות כאשר נרצה לעשות שינויים בתוכנית.

1. שכבת תצוגה (Presentation Layer)

שכבת ממשק המשתמש (UI – User Interface). שכבת התצוגה מדפי HTML באתר אינטרנט. השכבה מתקשרת בין המשתמש לבין שאר השכבות של המערכת.

2. שכבת הלוגיקה העסקית (BL – Business Logic)

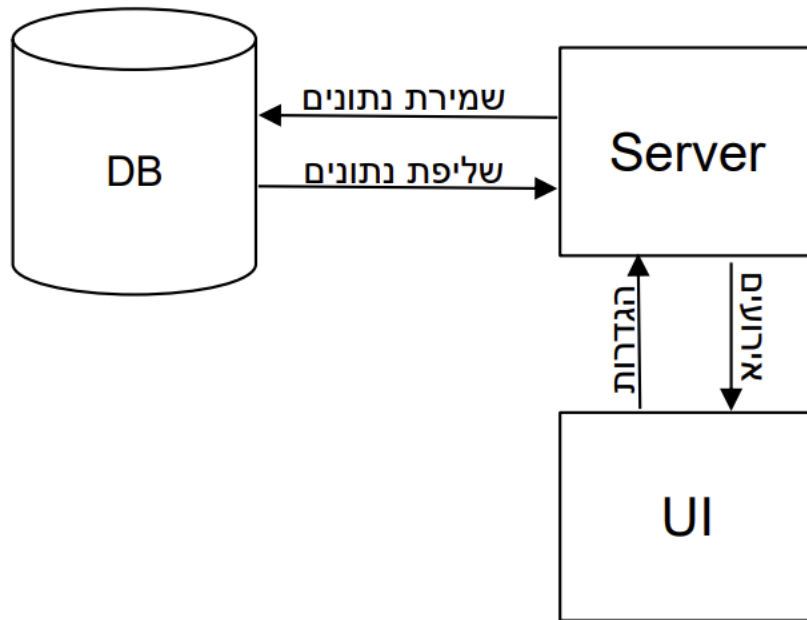
השכבה שממונה על הלוגיקה של המערכת, עוסקת בעיבוד המידע, בחישובים שונים ושליחתו לשכבת התצוגה. בשכבה זו נממש את הפונקציונאליות של המערכת.

3. שכבת הנתונים (DAL – Data Access Layer)

שכבה זו מורכבת ממקור נתונים – מסד הנתונים שלנו, וממערכת תוכנה entity framework אשר תפקידה לקרוא את המידע הנדרש למערכת, לשמור את העדכונים, ולהוסיף מידע חדש או למחוק פרטי מידע קיימים.

1.5. זרימת המידע במערכת

1.5.1 תרשים זרימה בסיסי על התנהגות המערכת



1.5.2 תרשים הטבלאות

Order
_id
deliverymanId
doneOrNot
name
email

phone
lat
lng
addressLine
deliverymanNum

deliveryMan
_id
firstName
lastName
active
email
phone
tz

2. ממשקים חיצוניים

- Google Maps API:

כאשר המנהל מוסיף הזמנה למאגר ומזין פרטי לקוח בעת הזנת כתובת ניתנת אפשרות בחירה של

כתובת מדויקת מתוך מאגר הכתובות של גוגל.

כך הכתובת נשמרת במסד הנתונים על ידי הקורדינאטות.

בנוסף לכך לצורך חישוב המסלול ישנה קריאה לשרת Google Maps על מנת לחשב את המרחק בין כל שתי כתובות.

- Firebase :

עבור Authentication, ניהול המשתמשים במערכת בצורה חכמה ובטוחה.

מלבד אפשרויות הרישום והכניסה הנוחים ניתן לשחזר סיסמא במהירות דרך המייל.

- Waze:

שליח יוכל לנווט לכתובת הלקוח בלחיצת כפתור אחת בלבד אשר תוביל אותו ישירות לאפליקציית waze

יש לציין כי אם לא תותקן אפליקציה זו אצל השליח יפתח לו הניווט בדפדפן + המלצה להורדת האפליקציה.

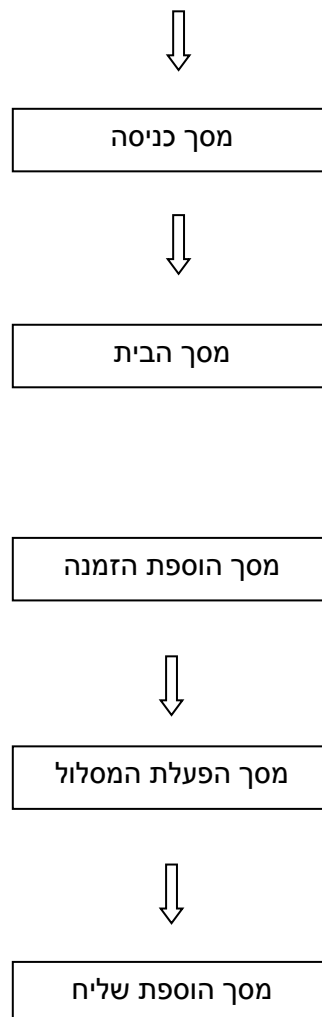
3. ממשק אדם ומכונה

3.1 כללי

הממשק של המערכת נבנה בבהירות ובפשטות מקסימליים כדי שהמשתמש ימצא מיד את מבוקשו. עיקר תשומת הלב הייתה על ארגון פשוט ונח של כל הנתונים. כמו כן, התצוגה נבנתה בהתאמה להרשאת המשתמש, כל משתמש רואה רק את הדברים הקשורים אליו.

3.2 תרשים מסכים

מסכי האתר





מסך צפייה בפרטי שליחים

מסכי האפליקציה



מסך הרשמה



מסך כניסה



מסך הבית



מסך הזמנות להפצה על פי סדר המסלול

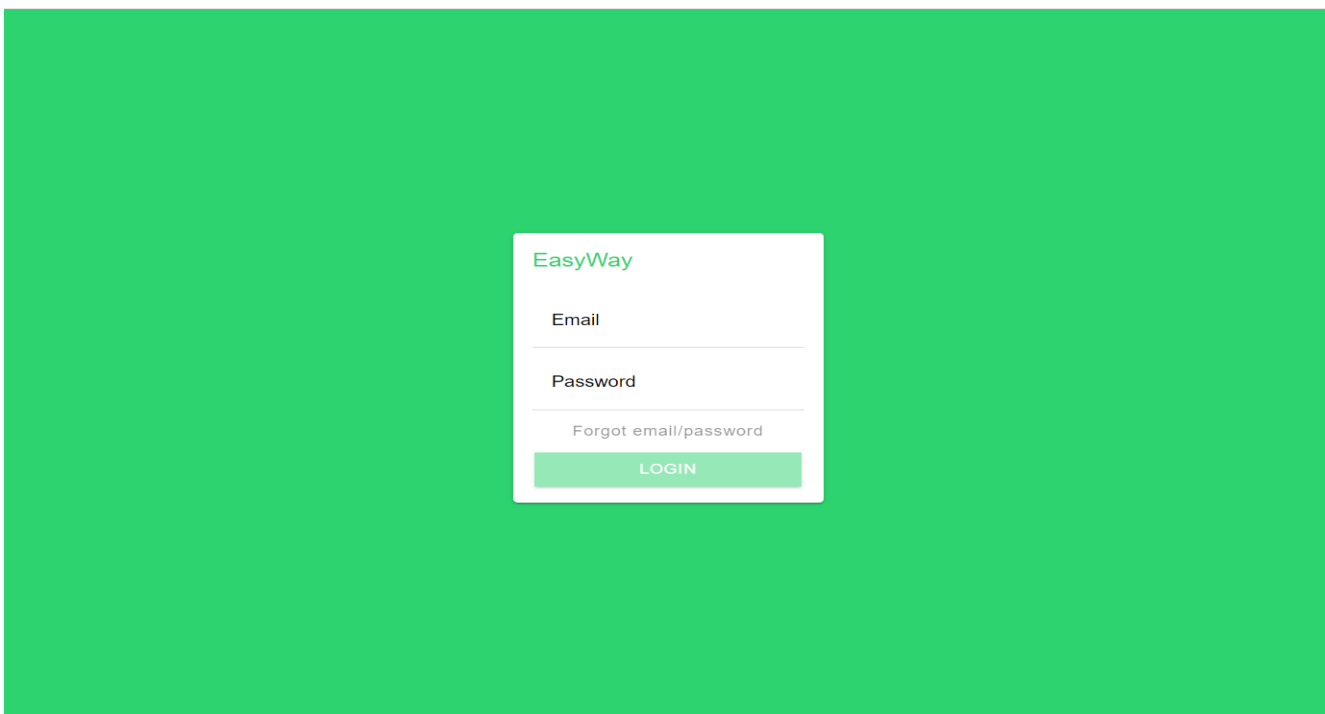
מסך הזמנה ספציפית להפצה

3.3 תיאור מסכים

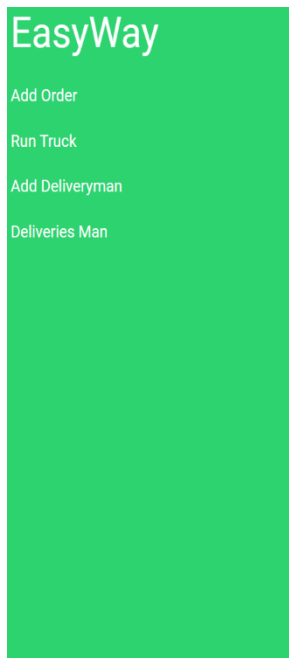
להלן צילומי המסכים:

3.3.1 מסכי האתר עבור המנהל

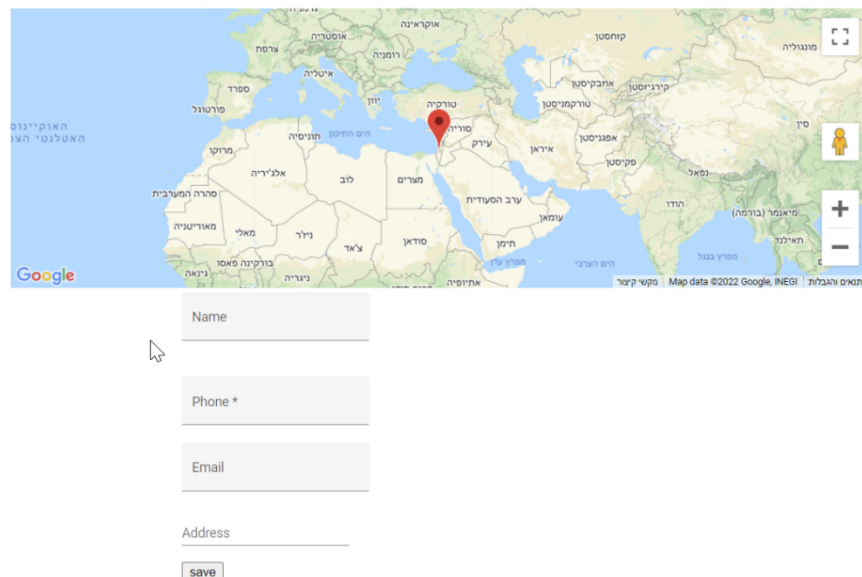
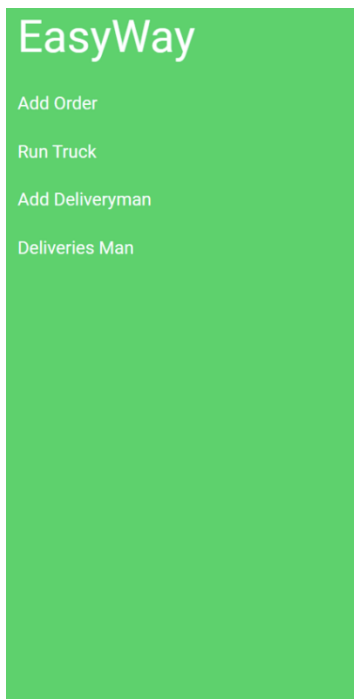
מסך כניסה לאתר



מסך הבית



מסך הוספת הזמנה



מסך הפעלת מסלול

EasyWay

Add Order

Run Truck

Add Deliveryman

Deliveries Man

There are 3 orders which haven't been delivered yet



There are 0 orders which haven't been shipped for distribution yet

Run the track

מסך הוספת שליח

EasyWay

Add Order

Run Truck

Add Deliveryman

Deliveries Man

first name *

last name *

tz *

phone *

email *

☒ active

save

מסך צפייה בפרטי שליח

EasyWay

Add Order

Run Truck

Add Deliveryman

Deliveries Man

Active deliverymen list

Deliverymen list

FirstName

LastName

Phone

Active



3.3.2 מסכי האפליקציה עבור השליח

מסך כניסה לאפליקציה

EasyWay

Email


Password

Forgot email/password

LOGIN

REGISTER

מסך הרשמה לאפליקציה

 Register

Contact

Email


Password

Confirm password


REGISTER


מסך הבית וצפייה בפרטים אישיים של השליח


☰ Easy-Way

 moshe

Personal Details


 moshe

 mosh@gmail.com

 0527777777

Status: Active

UPDATE DETAILS



מסך עדכון פרטי שליח

← Update your details

Details

Phone

Active

☐

CONFIRM

מסך ההזמנות למסירה על פי הסדר שחושב במסלול

← Deliveries for today

Order Number

62062fb5c57e48f7ec493570

Order Status

false

START

Order Number

6209582ac96a71ed0283ec98

Order Status

false

Order Number

620d3f55da6a7e17858fff6a

Order Status

false

מסך צפייה בפרטי לקוח של הזמנה ספציפית להפצה עם כפתור ניווט ל waze ואישור מסירת הזמנה.

← Order

Customer Details

yair

055-6770923

y@gmail.com

GO !

CONFIRM DELIVERY

4. מבנה נתונים וארגון קבצים

4.1 ארגון קבצים

כל רכיבי המערכת נבנו בארכיטקטורת אנגולר. לכל רכיב במערכת יש תיקיה נפרדת המכילה קובץ CSS- עיצוב, קובץ Type Script - לוגיקה, קובץ Type Script - בדיקות, וקובץ HTML – תצוגה, כל רכיבי המערכת מוכלים במודול ראשי המנהל אותם. הפרויקט מחולק לצד לקוח ולצד שרת ואובייקטים נשלחים לשרת בפורמט Json.

4.2 מבנה נתונים

הנתונים נשמרים כאובייקטים בבסיס הנתונים בענן MongoDB Atlas בתוך Collections כמסמכים.

מבנה המסמכים עומד מעל מימוש של Json

כל קובץ json מורכב מ:

- Id ייחודי שניתן אוטומטית בעת יצירת אובייקט.
- Keys
- Values

כאשר השדות הן keys , וערכי השדות הם values.

אחסנו במסד הנתונים את כל פרטי ההזמנות והשליחים.

ולכל אוסף במסד הנתונים קיימת מחלקה מקבילה בצד ה – client.

```
export class Order {
  constructor(firstName: string, lastName: string, mail: string, phone: string
    , orderId: number, lat: number, lng: number, deliverymanNum: number, id) {
    this.firstName = firstName;
    this.lastName = lastName;
    this.phone = phone;
    this.orderId = orderId;
    this.mail = mail;
    this.lat = lat;
    this.lng = lng;
    this.deliverymanNum = deliverymanNum;
    this.id = id;
  }
  firstName: string;
  lastName: string;
  mail: string;
  phone: string;
  orderId: number;
  id: number;
  lat: number;
  lng: number;
  deliverymanNum: number;
}
```

```

export class DeliveryMan {
  constructor(firstName: string, lastName: string, email: string, phone: string
    , password: string) {
    this.firstName = firstName;
    this.lastName = lastName;
    this.phone = phone;
    this.password = password;
    this.email = email;
  }
  firstName: string;
  lastName: string;
  email: string;
  phone: string;
  password: string;
}

```

5. תכנון

5.1 כללי

הפרויקט מתחלק לצד שרת וצד לקוח –

צד שרת נבנה ב WEB API בשפת #C

צד לקוח: נבנה בשפת JS ארכיטקטורת אנגולר.

הפרויקט נבנה לפי מבנה הארכיטקטורות הנ"ל כדי להשתמש במירב היכולות ולהפיק את התועלת המרבית.

5.2 עקרונות תכנות

בפרויקט זה עמדה לפנינו מטרה לכתוב קוד קצר ויעיל התואם את צרכי הלקוח, תוך שימוש מקסימאלי בטכנולוגיות העומדות לרשותנו בכלי התוכנה שבחרנו.

הפרויקט נבנה בגישת OOP – תכנות מונחה עצמים.

זוהי פרדיגמת תכנות המשתמשת באובייקטים לשם תכנות תכניות מחשב ויישומים.

שפת #C בנויה מאובייקטים ומחייבת שפיתוח הקוד יעשה על ידי מחלקות.

הפרדיגמה מספקת מספר כלי הפשטה וטכניקות ובהן הורשה, מודולריות, פולימורפיזם וקימוס.

החלוקה לשכבות הינה חלוקה ברורה בין ממשק המשתמש, התפקיד הלוגי של המערכת והתקשרות למערכות אחרות.

5.3 תיאור אלגוריתמים

האלגוריתם הראשי של הפרויקט הינו החישוב מסלול.

הפונקציה לוקחת מהמסד נתונים את כל ההזמנות שעדיין אין להן deliverymanId כלומר, עדיין לא הוקצו למסלול מסוים.

בעת הוספת הזמנה שמרנו במסד הנתונים את הקורדינאטות של מיקום כתובת הלקוח.

כעת, נבנית מטריצת מרחקים בין כל הכתובות של ההזמנות הנ"ל על ידי קריאה לשרת google maps לחישוב מרחק בכבישים ולא מרחק אווירי.

חלוקה אופטימלית למסלול הקצר ביותר עבור כל שליח פעיל.

```

1 reference
private async Task<DistanceMatrix> CreateDistanceMatrix(List<Order> orders)
{
    try
    {
        DistanceMatrix mat = new DistanceMatrix();
        int numofAddressesPerLoop = 10;
        int numofLoops = (int)Math.Ceiling(((double)orders.Count / (double)numofAddressesPerLoop));
        for (int i = 0; i < numofLoops; i++)
        {

            var addresses = string.Join('|', orders.Skip(i * numofAddressesPerLoop).Take(numofAddressesPerLoop).Select(o => $"{o.Lat},{o.Lng}"));

            var apiKey = "AIzaSyBFVQT8-gOzy3rhID9yuz8eJN_QL70qCqQ";
            var url = $"https://maps.googleapis.com/maps/api/distancematrix/json?origins={addresses}&destinations={addresses}&key={apiKey}";

            using (HttpClient client = new HttpClient())
            {
                HttpResponseMessage response = await client.GetAsync(url);
                response.EnsureSuccessStatusCode();
                string responseBody = await response.Content.ReadAsStringAsync();
                DistanceMatrix temp = Newtonsoft.Json.JsonConvert.DeserializeObject<DistanceMatrix>(responseBody);
                if (mat.origin_addresses == null)
                {
                    mat = temp;
                }
                else
                {
                    mat.origin_addresses = mat.origin_addresses.Concat(temp.origin_addresses).ToList();
                    mat.destination_addresses = mat.destination_addresses.Concat(temp.destination_addresses).ToList();
                    mat.rows = mat.rows.Concat(temp.rows).ToList();
                }
            }
        }
        return mat;
    }
    catch (Exception e)
    {
        return null;
    }
}

```

```

public async Task<List<string>> CalculateRoutes()
{
    var orders = _OrderRepository.DoneOrNot();
    List<string> readyRouteList = null;

    var warehouse = _OrderRepository.getWarehouse();

    orders.Insert(0, warehouse);
    int count = (orders.Count / 10) + 1;

    var distanceMatrix = await CreateDistanceMatrix(orders);

    var deliveyman = _deliveryManRepository.Get();

    RoutingIndexManager manager =
        new RoutingIndexManager(distanceMatrix.origin_addresses.Count, deliveyman.Count(), 0);

    RoutingModel routing = new RoutingModel(manager);

    int transitCallbackIndex = routing.RegisterTransitCallback((long fromIndex, long toIndex) =>
    {
        var fromNode = manager.IndexToNode(fromIndex);
        var toNode = manager.IndexToNode(toIndex);
        return distanceMatrix.rows[fromNode].elements[toNode].distance.value;
    });
}

```

```

routing.SetArcCostEvaluatorOfAllVehicles(transitCallbackIndex);

var maxValueDistance = distanceMatrix.rows.Max(r => r.elements.Sum(e => e.distance.value));

routing.AddDimension(transitCallbackIndex, 0, maxValueDistance * 2,
    true,
    "Distance");
RoutingDimension distanceDimension = routing.GetMutableDimension("Distance");

RoutingSearchParameters searchParameters =
    operations_research_constraint_solver.DefaultRoutingSearchParameters();
searchParameters.FirstSolutionStrategy = FirstSolutionStrategy.Types.Value.PathCheapestArc;

Assignment solution = routing.SolveWithParameters(searchParameters);
return PrintSolution(distanceMatrix, routing, manager, solution, orders, deliveyman);
}

```

5.4 בדיקת המערכת

באתר קיימות ולידציות על שדות קלט למניעת הכנסת אובייקטים לא תקינים. הבדיקות נעשות הן ברמת ה client לדוגמא: טופס שאינו חוקי אינו מאפשר לחיצה על כפתור ה submit . והן ברמת ה server . ערכתי בדיקות ידניות תוך מעבר על מקרי קצה.

6. מה הקנה הפרויקט

- ניסיון בתכנון ואפיון מהבסיס ועד לשלב הגמר.
- הבנה עמוקה על התנהלות השרת.
- הבנת התקשורת בין צד שרת לצד לקוח.
- ניסיון התחברות לממשקים חיצוניים בהצלחה.
- ניסיון מעשי בבניית שרת WEB API .
- ניסיון מעשי בעבודה עם ENTITY CORE .
- ניסיון מעשי בבניית DB ב MongoDB Atlas .
- ניסיון מעשי בבניית אתר ב Angular .
- ניסיון מעשי בבניית אפליקציה ב Ionic .
- פיתוח יכולת חזקה של לימוד עצמי.

7. ביבליוגרפיה

- <https://angular.io/>
- <https://ionicframework.com/docs/components>
- <https://stackoverflow.com/>
- <https://material.angular.io/>
- <https://getbootstrap.com/docs/5.1/getting-started/introduction/>

<https://w3schools.com/>

•

<https://developers.google.com/optimization/routing>

•

ועוד הרבה אתרים נוספים שעלו בגוגל בעת חיפוש ספציפי.

•