

שבץ וסע



מגישות: אפרת רייכנברג ומוריה מלכה

מנחה: הגב' שולמית זקש

תאריך הגשה: 13/3/2022

הצהרת סטודנטשם הסטודנט: מוריה מלכה ת.ז. 212221816שם הסטודנט: אפרת רייכנברג ת.ז. 322997024שם המכללה בה לומד הסטודנט: סמינר בית יעקב פתח תקוה

אני הח"מ, מצהיר בזאת כי פרויקט הגמר וספר הפרויקט המצ"ב נעשו על ידי בלבד.
 פרויקט הגמר נעשה על סמך הנושאים שלמדתי במכללה ובאופן עצמאי.
 פרויקט הגמר וספר הפרויקט נעשו על בסיס הנחייתו של המנחה האישי.
 מקורות המידע בהם השתמשתי לביצוע פרויקט הגמר מצוינים ברשימת המקורות
 המצוינים בספר הפרויקט.

אני מודע לאחריות שהנני מקבל על עצמי על ידי חתימתי על הצהרה זו שכל הנאמר בה
 אמת ורק אמת.

חתימת הסטודנט: .N.N תאריך: 13/3/2022

חתימת הסטודנט: ע.י. תאריך: 13/3/2022

אישור המנחה האישי

הריני מאשר שהפרויקט בוצע בהנחייתי, קראתי את ספר הפרויקט ומצאתי כי הוא מוכן
 לצורך הגשת הסטודנט להגנה על פרויקט גמר.

שם המנחה: שלומית זק"ש חתימה: של תאריך: 13/3/2022

סמינר מרכז בית יעקב למורות ולגננות פתח - תקוה (ע"ר)

סמל עמותה 0-335-58-040

רח' טעדיה גאון 15
פתח תקוה 49463
ת.ד. 99106 ת.ד. 103
סל. 9245315
פקס. 9214408
סמל מוסד 490128

לכבוד

יחידת הפרויקטים מה"ט

הצעה לפרויקט גמר

פרטי הסטודנטים:

שם	ת.ז.	טלפון	כתובת	תאריך סיום לימודים
אפרת רייכנברג	322997024	0556775678	אור יחזקאל 1 פ"ת	מרץ 2022
מוריה מלכה	212221816	0504107794	נועם אלימלך 13 ב' ב"ב	מרץ 2022

שם המכללה: סמינר בית יעקב פתח תקוה סמל המכללה: 76053

מסלול ההכשרה: הנדסאים

מגמת לימוד: הנדסת תוכנה מקום ביצוע הפרויקט: בבית

פרטי המנחה:

שם	מקום עבודה	טלפון	כתובת	תואר
שלומית זקש	סמינר ב"ע פ"ת	0533117151	סלומון 1 פ"ת	B.A

1. שם הפרויקט: שבץ וסע

2. רקע:

2.1. תיאור ורקע כללי:

האתר מספק שיבוץ יומי להסעת מטופלים לבתי חולים שונים בארץ, תוך סנכרון בין רשימת המתנדבים לרשימת המטופלים הזקוקים להסעה. האתר מאפשר למטופל הזקוק להסעה להגיש בקשה להסעה בהתאם לצרכיו ולמועד ההגעה הרצוי. האתר מבצע את ההתאמה היעילה ביותר בהתאם למקום המגורים, הזמן, ותנאים נוספים, ומעדכן הן את המתנדב והן את המטופל בהתאם, ואף מאפשר למטופל להשאיר משוב באזור האישי של המתנדב.

2.2. מטרת המערכת:

1. לתת מענה למטופלים הרבים הזקוקים לסיוע בהגעה למקום הטיפול בנוחות המירבית.
2. לקשר בין אנשים המעוניינים לנדב מזמנם, מרצם ורכבם לבין מטופלים הזקוקים לכך באופן היעיל.
3. לחסוך את עבודת הסנכרון והשיבוץ הידנית, תוך מתן תוצאות בעלות תאימות גבוהה בהתאם לנתוני המשתמש.
4. לפשט ולהקל על עבודתם של מנהלי מפעלי החסד העוסקים בתחום זה.

3. סקירת המצב הקיים בשוק- אילו בעיות קיימות:
על פי בדיקה שערכנו קיימים חולים רבים הזקוקים לטיפול מתמשכים בבתי חולים שונים, להרבה מהם אין את היכולת הפיזית ו/או הכספית לנסוע ברכב פרטי או במונית. נסיעה בתחבורה ציבורית מהווה עבורם אי נוחות וסבל גדול בהתחשב במצבם הבריאותי במקרים רבים. הפתרון לכך נמצא בסיועם של אנשים רבים וטובים שמקדישים מזמנם להסעת חולים ברכב הפרטי.
אך, כל שירותי ההסעות הקיימים כיום מסופקים ומשובצים בסיוע טלפוני או בווטסאפ. למעשה לא קיים אתר מסודר המבצע את עבודת השיבוץ בצורה יעילה.
מערך הסעות בהגדרתו כולל נתונים רבים הזקוקים להתייחסות על מנת לשבץ נכון, דבר שניתן להשגה בקלות רבה יותר על ידי שימוש בטכנולוגיות הקיימות כיום.
4. מה הפרויקט אמור לחדש או לשפר?
האתר חדש ויחיד מסוגו. הוא חוסך את עבודת השיבוץ והסנכרון המורכבת שלעיתים כמעט בלתי אפשרית ומפשט את עבודתם של מנהלי מפעלי החדש הללו, וכל זאת בהתייחסות לתנאי הנוחיות של החולה ושל המתנדב.
5. דרישות מערכת ופונקציונאליות:
 - 5.1. שרידות ביצועים והתמודדות עם עומסים
(סביבות הטמעה ושימוש)
המערכת תעבור קומפילציה והפצה בסביבת visual studio code .
תשתית לשרת היא סביבת Framework Express.js לשפת Node.js המאפשרת קבלת בקשות לתצוגת דפי אינטרנט.
המשתמש יוכל להריץ את האתר בכל מכשיר אשר מותקן עליו דפדפן אינטרנט.)
צד השרת מריץ Express.js המסוגל להתמודד עם מספר של קריאות בו זמנית.
עומס על שרת ה- MongoDB אינו צפוי בסדר גודל כזה של אתר מכיוון שהוא בנוי להתמודדות בהצלחה עם עומסים כבדים בהרבה.
 - 5.2. דרישות פונקציונאליות:
רשימת דרישות המשתמש מהמערכת:
 - 5.3.1- המערכת מאפשרת למתנדב להירשם למערכת, לעדכן את פרטיו מעת לעת ולצפות במרחב האישי.
 - 5.3.2- המערכת מאפשרת לנוסע להרשם למערכת, לעדכן את פרטיו האישיים, לשלוח בקשה לנסיעה בתנאים הנחוצים, ולאשר הגעה לנסיעה קבועה.
 - 5.3.3- המערכת מאפשרת למנהל המערכת לנהל את הרשאות ואישורי המערכת.
 - 5.3.4- המערכת מעדכנת במייל את המתנדב ברשימת הכתובות ליום המחר.
 - 5.3.5- המערכת מעדכנת במייל את הנוסע בפרטי המתנדב ופרטי הנסיעה המתוכננת.
 - 5.3.6- המערכת מאפשרת למתנדב ולנוסע לאשר את נסיעות יום המחר.
 - 5.3.7- המערכת מקיימת מערך תזכורות בהתאם להגדרות המשתמש.
 - 5.3.8- המערכת מאפשרת דיווח על תקלה ו/או עיכוב בנסיעה ומתריעה את הנוגעים בדבר בהתאם.
 - 5.3.9- המערכת מאפשרת לנוסע להשאיר פידבק למתנדב מסוים, ולמתנדב לצפות במשובים במרחב האישי שלו.

6. בעיות צפויות במהלך הפיתוח, ופתרונות תפעוליות, טכנולוגיות, עומס וכי' :

6.1. תיאור הבעיות- כפועל יוצא של דרישות המשתמש מהתוכנה:

● בעיה 1- הנוסע זקוק להסעה חזור אחרי ביצוע הטיפול, אך לא ניתן לדעת את שעת החזרה או לתכננה מראש.

● בעיה 2- האלגוריתם נדרש להתחשב במרחקי המתנדב והנוסע למען שיבוץ יעיל, איך אפשר לחשב את המרחקים?

● בעיה 3- איך האלגוריתם ידע להבחין ביחסי קרבה בשכונות מסוימות באותה העיר?

6.2. פתרונות אפשריים:

● לבעיה 1-

1. התעלמות כללית מנסיעות חזור. המערכת מספקת שיבוצים הלוך בלבד ומועילה גם כך

2. הוספת אפשרות למתנדב לדווח על המצאו בבית חולים ולעדכן בכך את הנוסעים השוהים במקום זמן מוסכם מראש קודם יציאתו.

● לבעיה 2-

1. שימוש בנתונים והתחברות לגוגל מאפס

2. תת חלוקה לאזורים, ערים ושכונות, כך שהאלגוריתם מחשב את המרחקים בהתאם לנתונים שמגדירים המתנדבים, באופן כזה שמאגר החלוקה מתרחב ביחס לכמות המתנדבים.

● לבעיה 3-

1. שימוש במאגר נתונים המאכסן את המידע המבוקש

2. נתינת אפשרות למתנדבים עצמם להגדיר באזור האישי את השכונות אליהם הם נכונים לנסוע וקרובים אליהם.

הפתרון הנבחר עבור כל אחת מהבעיות:

● לבעיה 1- פתרון 2 נבחר. כיוון שברצוננו לספק מערכת כמה שיותר מועילה, כך שאם נתן לגבור על הבעיה ועדיין לספק שירות חזור ברמה סבירה, הפתרון כמובן טוב יותר.

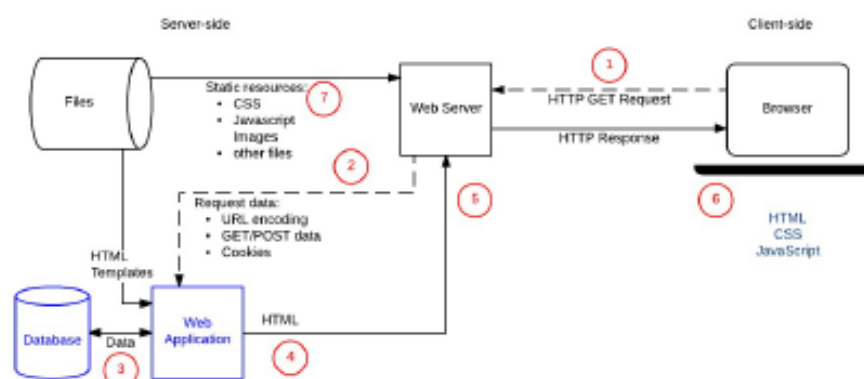
● לבעיה 2- הפתרון שנבחר הוא פתרון 2. זאת משום שעל מנת לשבץ בין מתנדב לנוסע באותה שכונה אין צורך להכנס לפרטים קטנים כל כך שמספק Google maps, וריבוי הנתונים שבו מיותר עבורנו. כך שחלוקה על פי אזורים וירידה עד לרמה של שכונות מהווה פתרון יעיל בהחלט בהתחשבות בממדי המרחקים הקיימים בארץ.

● לבעיה 3- פתרון 2 נבחר לבעיה זו, כיוון שראשית אין מאגר נתונים עבור המידע הנ"ל, ומלבד זאת נתינת אפשרות ההגדרה למתנדב משיגה עוד יתרון חשוב נוסף של התחשבות בהעדפותיו, תוך הסתמכות על היכרות המתנדב עם סביבתו הקרובה יותר מכל יכולת טכנולוגית אחרת. כך שהאלגוריתם יסרוק את המסלולים הקרובים על פי העדפות הנרשמות.

7. פתרון טכנולוגי נבחר:

7.1. טופולוגית הפתרון:

המערכת מורכבת משרת משרת Express ל- Node.js המריץ את האתר בסביבת ה- Server, מסד נתונים - של MongoDB. ממשק משתמש בצד הלקוח: דפדפן אינטרנט כלשהו כמו chrome על מחשב שולחני או על מכשיר iOS/ android.



7.2. טכנולוגיות בשימוש:

React.js : ספריית קוד פתוח מבית פייסבוק שמטרתה לאפשר פיתוח נוח ומהיר של אתר אינטרנט ואפליקציות.

MySQL : תבילה ל- Node המאפשרת את הגישה למסד הנתונים.

Web Api : טכנולוגיה המקשרת בין צד שרת - כלומר מסד נתונים לבין צד הלקוח - React.

7.3. שפות הפיתוח:

● **Node.js Express :** Node.js הוא מפרש לשפת JavaScript שנכתב ב- C++ ומבוסס על מנוע 8V. הוא תוכנן לכתיבת אפליקציות רשת סקלאביליות כגון שרת HTTP.

● **Express Framework :** זהו Framework שפותח ל- Node כדי לאפשר תקשורת על ידי בקשות HTTP.

● **React:** שפת פיתוח צד לקוח המוסיפה אבסטרקציות מעולם פיתוח Object Oriented, הכתיבה מתבצעת ע"י כתיבת JS ורנדור התצוגה ע"י בניית HTML דרך הקוד. יצירת Store ראשי כדי לשלוט על המידע של האפליקציה ועל שינויים שחלים בו. כתיבת Components כדי לשלוט על ה- Templates, כתיבה של Services כדי לשלוט על הלוגיקה, ושילוב של Modules ליצירת Components i- Services.

● **HTML5:** 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. HTML5 כוללת כלים חדשים שבאו להקל על עבודת המפתחים.

● **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.

7.4. תיאור הארכיטקטורה הנבחרת: הסבר בכמה מילים מדוע:

הארכיטקטורה הנבחרת היא חלוקה ל- 3 שכבות: Tier Architecture- 3 .
בפרויקטים שמבוססים על Database מומלץ להשתמש בגישה של מודל השכבות,
בארכיטקטורה זו קיימת הפרדה בין השכבות השונות בפרויקט.
היתרון העיקרי בחלוקה לשכבות הוא תחזוקה קלה, במיוחד במערכות גדולות, כך שניתן
יחסית בקלות להחליף כל שכבה בלי לגעת בשכבות האחרות כאשר נרצה לעשות שינויים
בתוכנית.

7.5 חלוקה לתכניות ומודולים:

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 - Logic Business): השכבה שאמונה על הלוגיקה של
המערכת, עוסקת בעיבוד המידע, בחישובים שונים ושליחתו לשכבת התצוגה.
בשכבה זו נממש את הפונקציונליות של המערכת.
3. שכבת הנתונים (DAL - Data Layer Access): שכבה זו מורכבת ממקור נתונים – מסד
הנתונים שלנו, ומספריית JS - MySQL אשר תפקידה לקרוא את המידע הנדרש למערכת,
לשמור את העדכונים, ולהוסיף מידע חדש או למחוק פרטי מידע קיימים.

7.6 סביבת השרת -

לצורך הפרויקט נשתמש בשרת מקומי - js.Express, המסופק על ידי ה- npm .
אם האתר יירכש על ידי לקוח, נעלה אותו לשרת אירוח כלשהו.

7.7 ממשק המשתמש - GUI-

שכבת ה GUI מורכבת מ-HTML, CSS, JS שמרונדרים במהדר של React ומוציגים למשתמש
דרך הדפדפן.

7.8 ממשקים למערכות אחרות/API-

WebApi, MailAPI, NodeJS, התממשקות למדפסות.

7.9 שימוש בחבילות תוכנה:

CSS, Express, MySQL, nodeJS, HTML, Visual studio Code, React

8 שימוש במבנה נתונים וארגון קבצים:

MySQL

8.1 מבנה הנתונים:

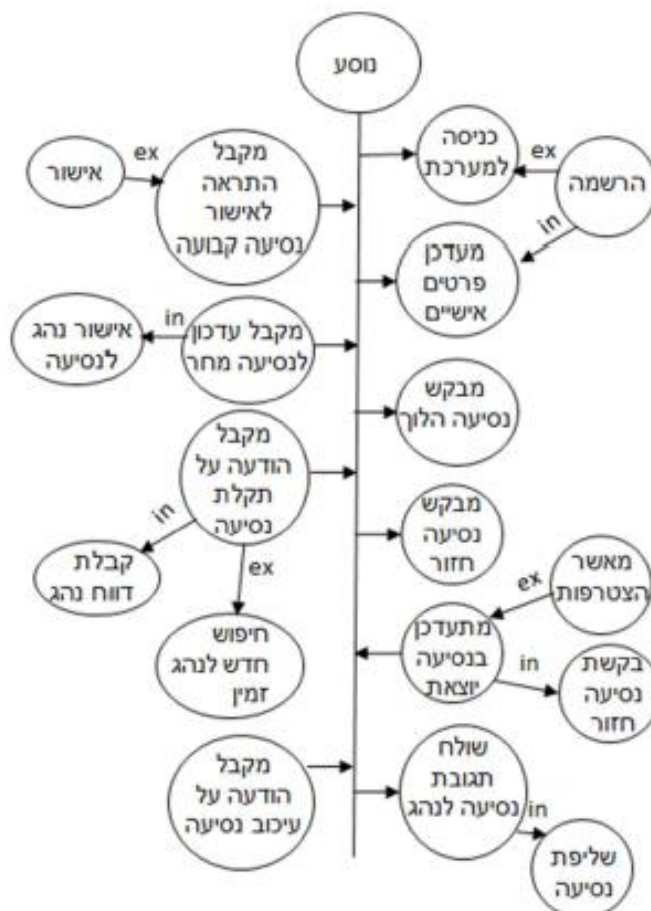
מתנדב: קוד, סיסמא, שם, טלפון, Mail, עיר, רחוב, מספר מקומות, מעלון, סטטוס
מזדמן, סטטוס מאושר.
נוסע: קוד, שם, טלפון, עיר, שכונה, רחוב, בית חולים, שעה.
סוג התנדבות: קוד, תיאור התנדבות.
שביעות רצון למתנדב: קוד, קוד מתנדב, טקסט, קוד נוסע, תאריך .
נסיעה למתנדב: קוד, קוד מתנדב, תאריך, שעה, סוג נסיעה, סטטוס תקלה, סטטוס אישור
מתנדב, סטטוס אישור נוסע.
נסיעה קבועה לנוסע: קוד, קוד נוסע, יום, שעה, סוג נסיעה.
נוסעים בנסיעה: קוד, קוד נסיעה, קוד נוסע, מספר נוסעים.
אפשרויות התנדבות: קוד, קוד מתנדב, יום, שעת התחלה, שעת סיום, סוג התנדבות.
מגבלות: קוד, קוד אפשרות מתנדב, תאריך, שעת התחלת מגבלה, שעת סיום מגבלה.

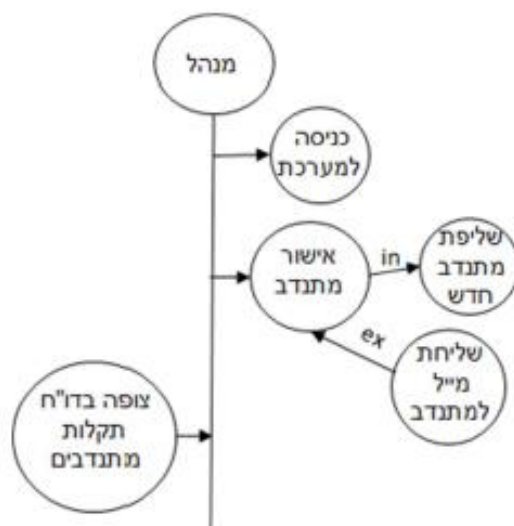
8.2 שיטת האחסון:

שיטת האחסון היא במסד נתונים של MySQL.
הגישה למסד הנתונים היא על ידי ספריית MySQL-JS.

נפילה\ קריסה\ תמיכה - MySQL בנוי לתמיכה במס' גדול של קריאות בו זמניות ואין חשש לקריסה ברמה של הפרויקט.

9. תרשימי מערכת מרכזיים:





10. תיאור המרכיב האלגוריתמי\ חישובי:
איזה בעיה בא לפתור, איך יפתור?
הבעיה: תכנון שיבוץ יעיל של מתנדב לנוסע על פי הדרישות
הפתרון: ניצור פונקציה שבעזרת אלגוריתם תחשב את המסלול ואת ההתאמה היעילים תוך שקלול
הנתונים הרשומים עם העדפות המשתמשים, לפי חשיבות העדפה.
איסוף מידע ונתונים סטטיסטיים- אנליטיקות:
באתר יוצגו נתונים סטטיסטיים וממוצעים ביחס לשימוש באתר ולכמות השימוש בו.
11. תיאור התייחסות לנושא אבטחת המידע:
יש לדאוג לאבטחת השרת מעומסים מופרדים הנגרמים על ידי גורמים זדוניים, ואת שרת
SQL Injection. מ-MySQL ה-
משתמש הגולש באתר מריץ אותה באופן אבסולוטי. קוד המשתמש ישלח לשרת ויצטרף לכל
רשומה הנשמרת עבורו. לא יתכן מצב בו משתמש מריץ אפליקציה בתור משתמש אחר.
מקרים להם ניתן מענה אבטחתי:
 - כניסת משתמש על ידי אמייל וסיסמא, כך שלכל משתמש אזור אישי ופרטי שאינו חשוף למשתמשים האחרים.
 - שליחת אישור הרשמה למייל כדי לוודא שהאמייל שהוזן אכן שייך לנרשם.
 - רמת הרשאת גישה גבוהה יותר למנהל המערכת כדי שלא כל משתמש יוכל לשנות את הנתונים באתר.
 - במקרה שבעת כניסת משתמש הסיסמא אינה תואמת לשם המשתמש שהקיש, המערכת תציג לו הודעת שגיאה ולא תאפשר כניסה.
 - אבטחת פרטיות מתבצעת בעת שיבוץ. הפרטים נשלחים וזקוקים לאישור שלא ייווצר מצב של חוסר נעימות.
12. משאבים הנדרשים לפרויקט:
 - 12.1 מספר שעות המוקדש לפרויקט: 700
 - 12.2 ציוד נדרש: CPU i5, RAM 8GB, HD SSD, חיבור אינטרנט.
 - 12.3 תוכנות נדרשות: MySQL, Visual Studio Code, browser
 - 12.4 ידע חדש שנדרש לצורך ביצוע הפרויקט:
הרחבת הידע בשפת js, Node, React ובשימוש בספריות חיצוניות שונות.
 - 12.5 ספרות ומקורות מידע:
<https://stackoverflow.com/>
<https://www.geeksforgeeks.org/>
<https://reactjs.org/docs/hooks-intro.html>
<https://github.com/>
<https://www.w3schools.com/>
https://he.wikipedia.org/wiki/%D7%A2%D7%9E%D7%95%D7%93_%D7%A8%D7%90%D7%A9%D7%99
13. תכנית עבודה ושילבים למימוש הפרויקט:
 - 13.1 ייזום רעיון, ניתוח מערכת, הכרת השטח ואפיון מסד נתונים- עד יוני 2021.
 - 13.2 תכנון UI-UX, עיצוב גרדי, זרימת מסכים עד יולי 2021.
 - 13.3 כתיבה לוגית עסקית עד ספטמבר 2021.
 - 13.4 כתיבת ממשק משתמש ועיצוב עד נובמבר 2021.

- 13.5. בדיקות QA עד דצמבר 2021.
13.6. התקנה והטמעה עד ינואר 2022.

14. תכנון הבדיקות שיבוצעו:

מספר בדיקה	מספר דרישה במסמך אפיון	מקרי הבדיקה	ידנית\אוטומטית	חשיבות	הערות
1	5.3.1,2	נשלח מייל בהצלחה למתנדב ולנוסע שנרשם למערכת	ידנית	בינוני	
2	5.3.3	אף מתנדב נרשם לא נכנס למאגר הנתונים ללא אישור המנהל	ידנית	גבוהה	
3	5.3.4,5	המתנדב והנוסע מקבלים בהצלחה את הפרטים הנכונים לשיבוצי הנסיעה העתידיים למייל	ידנית	גבוהה	
4	5.3.6	שיבוצ שלא מאושר על ידי המתנדב כעבור פרק זמן מסוים מועבר לשיבוצ מחדש	ידנית	גבוהה	
5	5.3.7	התזכורות נשלחות למייל המשתמש המתבקש בהתאם להגדרות שציין	ידנית	גבוהה	
6	5.3.8	דיווח התקלה אכן נשלח לכל המשתמשים הנוגעים בדבר בעת קבלתו	ידנית	גבוהה	
7	5.3.9	הפידבק התקבל באזור האישי של המתנדב הנכון	ידנית	בינוני	

15. בקרת גרסאות:

תשמרנה גרסאות במהלך הפיתוח, כאשר לכל גרסה מצורף תיאור של תוכנה וההבדלים שלה מקודמתה.

ג.ל .N.N
אפרת ר"כנברג מוריה מלכא

חתימת המנחה האישי:

הערות ראש המגמה במכללה:

אישור ראש המגמה: _____
שם: _____
חתימה: _____
תאריך: 23/11/2021

הערות הגורם המקצועי מטעם מה"ט:

א'שור הגורם המקצועי מטעם מה"ט:

שם: _____
חתימה: _____
תאריך: _____

-תודות-

לפני הכל, תודה לבורא עולם שנתן ונותן את היכולת, הדעת, הכישורים והעזרה שרק איתה הגענו עד כה.

תודה למנחה שלנו, שישבה אתנו שעות ארוכות לסיעור מוחות, בחינת אפשרויות והענות לשאלות בכל מיני שעות...

תודה לארגון "דרכי מרים" שסיפק לנו מידע מהימן ומפורט שאפשר בכלל לפרויקט הזה לרקום עור וגידים ולהפוך מרעיון מטושטש למערכת מרובת פרטים.

תודה לכל מי שנקלע לדרכנו ותרם ממרצו, זמנו ויכולותיו למען הפרויקט.

תודה גדולה למשפחותנו שקיבלו בניד ראש מבין את העלמותנו הכללית מהשטח, מלבד גיחות קצרות למטבח ולשחרור לחצים...

ותודה לכל מורה ומתרגלת שהקנו לנו את הידע והטיפים שאפשרו לנו להוציא לפועל את תכנוננו.

-תקציר הפרויקט-

האתר "שבץ וסע" מספק שיבוץ בין מתנדבים בעלי רכב למטופלים הזקוקים להסעה לצורך טיפול בבתי חולים שונים.

האתר נבנה בהשראת ארגון קיים- "דרכי מרים"- המנהל את השיבוץ ידנית ודרך הטלפון, את הגישה ושיטת הפעולה בין המשתמשים השונים יצרנו על פי מחקר שבעצנו אתם.

לאתר יכולים להירשם מתנדבים ונוסעים וניהולו נעשה על ידי הארגון העוסק במערכות חסד מהסוג הזה.

מטופל הזקוק לנסיעה נרשם לאתר וממלא את פרטיו האישיים והעדפותיו הנצרכות להתאמת שיבוץ טוב אותם יוכל לערוך בכל עת בפרופילו.

הוא יכול לשלוח בקשה להזמנת נסיעה ליום המחר שנכנסת למאגר הנסיעות שישלפו לצורך השיבוץ.

הוא יכול לצפות בפרטי הנסיעה שקיבל ובנסיעותיו הקודמות ולשלוח פידבק למתנדב אתו נסע בעבר.

משתמש החפץ להתנדב נרשם לאתר, ממלא את פרטיו האישיים ואת זמני התנדבותו, כולל פרטים לגבי רכבו.

הוא יכול לצפות באזורו האישי בנסיעותיו ליום המחר, בהסטורית נסיעותיו הקודמות, בפידבקים שנשלחו לו על ידי הנוסעים, וכמובן לערוך את פרטיו מעת לעת.

המנהל באזורו האישי יכול לבצע את בקשת השיבוץ למחר שמפעילה את אלגוריתם השיבוץ הכולל שימוש באלגוריתם ההונגרי (פירוט בהמשך) על כל הזמנות הנסיעה שנשלחו במהלך היום, על פי סדרי עדיפויות של זמן ומקום ואלו שהגדירו המשתמשים.

בסיום השיבוץ מוצגת לפניו טבלת הנסיעות למחר עם הפרטים הנחוצים, המנהל יכול לערוך אותה כראות עיניו ולאשר או לחזור למצב הקודם.

למנהל ישנה הגישה לאישור מתנדבים חדשים, וצפיה ברשימת המתנדבים והנוסעים המועדכנת.

הערך המוסף שניתן באתר הוא סיפוק שיבוץ חכם ומחושב על פי פרמטרים רבים ושונים- היעיל על פני עבודה ידנית, ובכך לסייע בעבודת הקודש של הארגונים, כמובן תוך סיפוק ממשק נח ונעים לחווית המשתמשים השונים.

אך כמובן כל זאת בלי להתיימר להחליף אי פעם את עובדי הארגון, שנסינום ידיעותיהם ואנושיותם לעולם לא יתחלפו ביכולת טכנולוגית זו או אחרת, אלא שילוב ביניהם לבין היכולת הטכנולוגית המסייעת- יביאו לתוצאות הטובות ביותר.

-תוכן העניינים-

17.....	1. הגדרת דרישות ותיאור כללי.....
17.....	1.1 תיאור כללי.....
18.....	1.2 תיאור חומרת המערכת.....
19.....	1.3 תיאור תוכנת המערכת.....
20.....	1.4 תיאור מבנה הפרויקט.....
20.....	1.5 זרימת המידע במערכת-תרשימים.....
22.....	2. ממשקים חיצוניים.....
22.....	3. ממשק אדם ומכונה.....
22.....	3.1 כללי.....
22.....	3.2 כללי.....
23.....	3.3 תיאור מסכים.....
30.....	4. מבנה נתונים וארגון קבצים.....
30.....	4.1 מבני נתונים.....
30.....	4.2 פירוט טבלאות מסד הנתונים.....
31.....	5. תכנון.....
31.....	5.1 כללי.....
31.....	5.2 עקרונות תכנות.....
31.....	5.3 תיאור אלגוריתמים.....
38.....	6. מה הקנה הפרויקט.....
38.....	7. ביבליוגרפיה.....
39.....	8. נספחים.....

1. הגדרת דרישות ותיאור כללי

1.1 תיאור כללי

האתר מספק שירות עבור שלושה משתמשים שונים, ביניהם המנהל, ומיועד להקל על ארגוני חסד לבצע את עבודת השיבוץ היומית בצורה חכמה, קלה, יעילה, ובמירב האופטימליות, ובכך לאפשר גם למתנדבים ולנוסעים ממשק נח לשליחת בקשותיהם, לצפייה בתכנים הרלוונטים להם ובעריכת פרטיהם בנוחות.

האתר שומר על פרטיות משתמשיו, ומספק למתנדב מידע מינימליסטי על הנוסע רק בעת אישור השיבוץ.

באתר יכולים המשתמשים לצפות בהסטורית נסיעותיהם.

הנוסעים יכולים להביע את תודתם בשליחת פידבק אישי למתנדב, והמתנדב יכול לצפות בכל הפידבקים שנשלחו לו להטמעת גודל משמעות מעשיו.

מתנדב חדש עובר את אישור המנהל על מנת להפוך למתנדב מן המניין על מנת שיהיה פיקוח בניהול המערך.

המנהל שולח בקשת שיבוץ בעת שנצרך לה, ויכול לערוך ולשפר על פי נסיונו והכרת האנשים אתם הוא בא במגע את תוצאות השיבוץ ובכך להביא אנושיות ומקצועיות טכנולוגית לכדי מצוינות.

1.1.1 מטרות המערכת

- להקל משמעותית על עבודת הארגון בסיפוק שיבוצים יעילים ומותאמים.
- לאפשר למתנדב להצטרף למערך ההתנדבותי בנוחות ובקלות ולספק לו ממשק נח למשך פעילותו.
- לתת מענה למטופלים הנדרשים לנסיעה בנוחות המירבית. לאפשר לו לקבל את השירות שזקוק לו תוך שימוש נעים ומהיר והצגת המידע הרלוונטי עבורו.

1.1.2 היקף הפרויקט

- 700 שעות.

1.1.3 מבנה וארגון

האתר מורכב מ:

Data Base - לשמירת נתוני המתנדבים והנוסעים, הנסיעות, השיבוצים, והמשובים
הנתונים מאוכסנים ב- MySQL.

Server - גישה לנתונים ושלפתם על ידי שרת Express ל-Node.js. גישה לקריאה ושמירה על הנתונים על ידי Web API.

Client - צד הכולל את מסכי המערכת, המימוש נעשה ב-React JS.

מודולים עיקריים

- **הוספת מתנדב** - מאפשר למשתמש החפץ להצטרף למערך ההתנדבותי להירשם ולהכניס את פרטיו וזמני פעילותו, כשעל ידי אישור המנהל חשבונו הופך לפעיל.
- **הוספת נוסע** - מאפשר למשתמש הנזקק לנסיעה להירשם לאתר ובין השאר להזמין את פרטי הנסיעה לה הוא זקוק.
- **ביצוע שיבוץ** - הפעלת אלגוריתם השיבוץ שבסופו יוצג למנהל לוח הזמנים ליום המחר שיחכה לאישור ו/או עריכתו, למתנדב - פרטי נסיעותיו ולנוסע - את פרט נסיעתו.
- **פידבקים** - מספק יכולת לנוסע לפדבק את המתנדב אתו נסע, ולמתנדב לצפות בפידבקים שנשלחו לו.
- **עריכת פרטים** - כיאה לאתר המתעסק בנתוני שיבוץ כל המשתמשים יכולים לעדכן את פרטיהם בהתאם לצרכיהם.
- **הסטוריה** - הן המתנדב והן הנוסע יכולים לצפות בפרטי נסיעותיהם הקודמות.

1.1.4 משימות המערכת

- מתנדב יוכל להירשם ולהכנס לאזורו האישי ולצפות בפרטי נסיעותיו ובפידבקים שקיבל.
- נוסע יוכל להירשם ולגשת לאזורו האישי בו יוכל לצפות בפרטי נסיעותיו.
- המתנדב יוכל לעדכן את פרטיו האישיים ואת זמני פעילותו, כולל חריגות.
- נוסע יוכל לעדכן את פרטיו האישיים והעדפותיו.
- נוסע יוכל לשלוח פידבק למתנדב אתו נסע בעבר.
- המנהל יוכל לשלוח בקשה להפעלת שיבוץ
- המנהל יוכל לערוך את תוצאות השיבוץ ולשמור אותן.
- המנהל יוכל לאשר מתנדבים חדשים
- המנהל יוכל לצפות ברשימת המתנדבים והנוסעים המעודכנת.

1.2 תיאור חומרת המערכת

1.2.1 כללי

להפעלת המערכת נדרש: מחשב הפועל על מערכת הפעלה Windows החלק מגרסא XP.

1.2.2 מרכיבי המערכת

מחשב PC עם גישה לדפדפן Chrome.

1.3 תיאור תוכנת המערכת

1.3.1 כללי

במימוש הפרויקט שמנו דגש על התאמה טכנולוגית המתאימה ביותר לצרכינו, ויחד עם זאת תהיה חדשנית ותספק את דרישותינו לאיכות הפרויקט.

שפות הפיתוח בצד השרת

Node.js - שפה זו היא שפת תכנות נפוצה ודינמית לשימוש שמיועדת לפיתוח כללי של מגוון אתרים. התחביר שלה קליט, והחופשיות והקלות בה ניתן לממש בה פונקציות ולוגיקות ולהמיר טייפים של משתנים מפשטת את העבודה אתה. כמו כן מודולים שונים המסייעים לצרכינו בפרויקט ממומשים בה דווקא וקיימים בהישג יד. השפה מתממשת עם אחוזי תקלות נמוכים מאד עם ה-DB בו בחרנו להשתמש ועם צד ה-CLIENT שלנו, כך נחסכת טרחה טכנית רבה סביב תקלות מערכת.

שפות הפיתוח בצד הלקוח

React JS - ספריית קוד פתוח בשפת- JavaScript המשמשת לפיתוח ממשק משתמש. ל-React יש יתרון בתכונה מרכזית שלה שהיא Virtual DOM הספרייה יוצרת מבנה נתונים בזכרון, וכך בחישוב השינויים הנדרשים מתעדכן ה-DOM שלה. ומרחיבה את יכולת התגיות הבסיסיות בשפת HTML כך ניתן ליצור יישומי אנטרנט חכמים יותר. HTML5 - שפת בניית אתרים מבוססת תגיות, שאחראית על אופן תצוגת דף האנטרנט, מבחינת טקסט, תמונות, טבלאות, עיצוב, מיקום ועוד. זוהי השפה הבסיסית לסביבת ה-WEB עבור בניית אתרים. HTML5 כוללת כלים חדשים שמקילים על המפתחים. Bootstrap - סביבת עבודה בקוד פתוח לצד הלקוח, המכילה אוסף כלים ליצירת אפליקציות WEB ואתרים ולעיצובם באופן חדשני.

1.3.2 כלי התוכנה לפיתוח המערכת

React JS

JavaScript
HTML5
CSS3
Mysql

1.4 תיאור מבנה הפרויקט

הארכיטקטורה הנבחרת היא חלוקה ל-3 שכבות: Tier 3 Architecture. בפרויקטים מבוססי DB מומלץ להשתמש בגישה של מודל השכבות, בארכיטקטורה זו קיימת הפרדה בין השכבות השונות בפרויקט. היתרון העיקרי בחלוקה לשכבות הוא תחזוקה קלה, במיוחד במערכות גדולות, כך שהניתן בקלות יחסית להחליף שכבות מסבלי לגעת בשכבות אחרות במקרה שנרצה לבצע שינויים בתכנית.

Presentation Layer:

שכבת ממשק המשתמש (UI).
שכבת התצוגה מורכבת מקבצי HTML, CSS, React.
השכבה מתקשרת בין המשתמש לבין שער שכבות המערכת.

Logic Business:

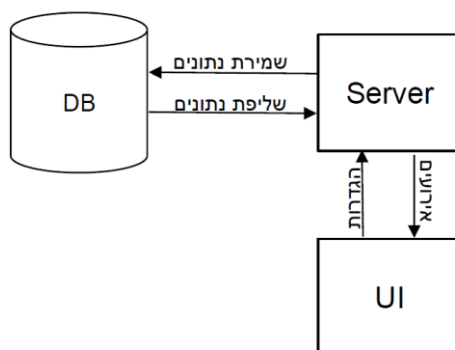
השכבה שאמונה על הלוגיקה של המערכת, עוסקת בעיבוד המידע, בחישובים שונים ושליחתם לשכבת התצוגה.
בשכבה זו ממומשות פונקציות ואלגוריתמיקת המערכת.

Logic Business:

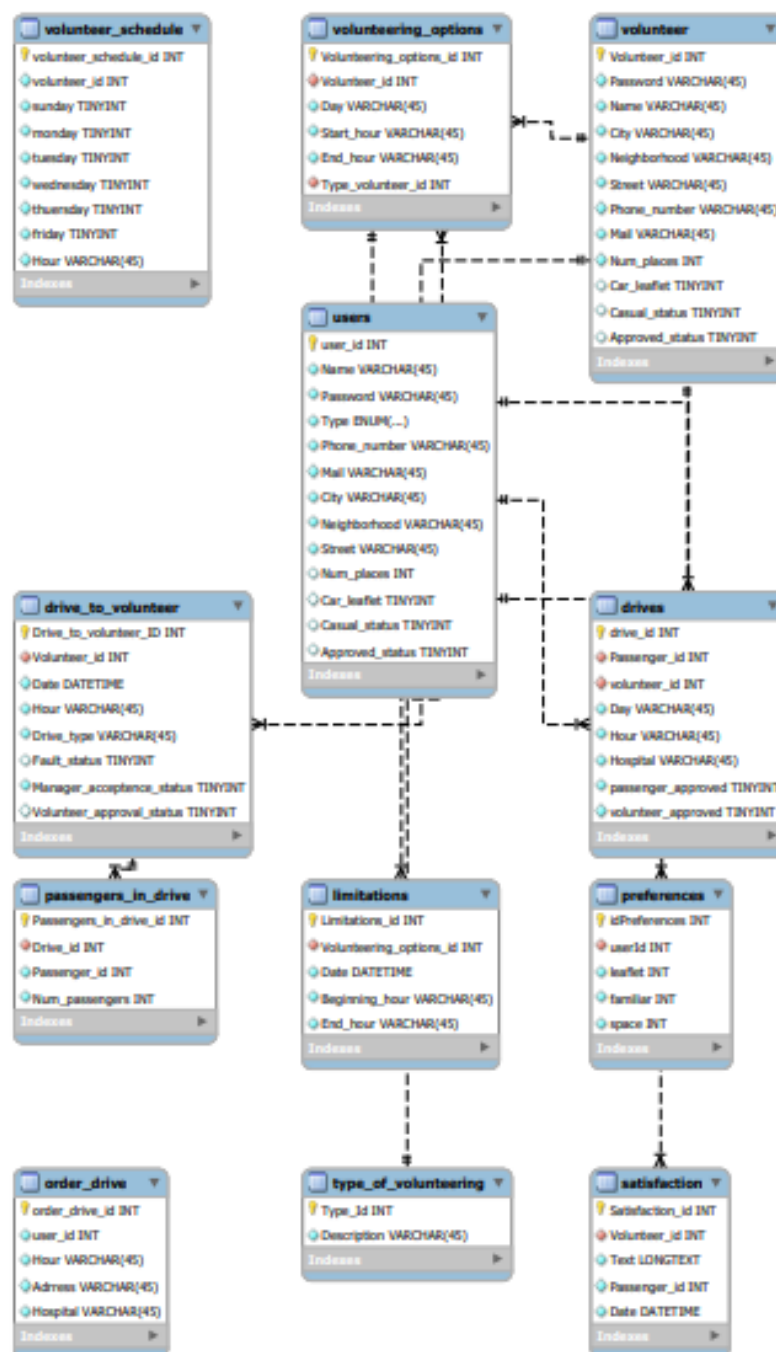
שכבת זו מורכבת ממקור הנתונים- מסד הנתונים שלנו, ומספרית MySQL אשר תפקידה לקרוא את המידע הנדרש למערכת, לשמור את העדכונים ולהוסיף מידע חדש או למחוק פרטי מידע קיימים.

1.5 זרימת המידע במערכת- תרשימים

1.5.1 תרשים זרימה בסיסי התנהגות המערכת



1.1.1 תרשים הטבלאות וקשרי הגומלין במודל



2. ממשקים חיצוניים

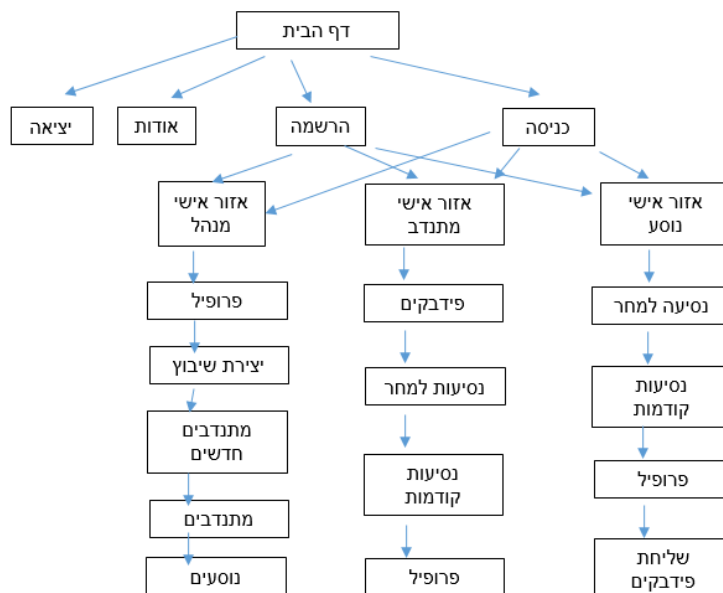
חישוב מרחקים וזמנים משוערים בין מוצא ליעד לצורך אלגוריתם השיבוץ ע"י שימוש בשירות של גוגל מאפס.

3. ממשק אדם ומכונה

3.1 כללי

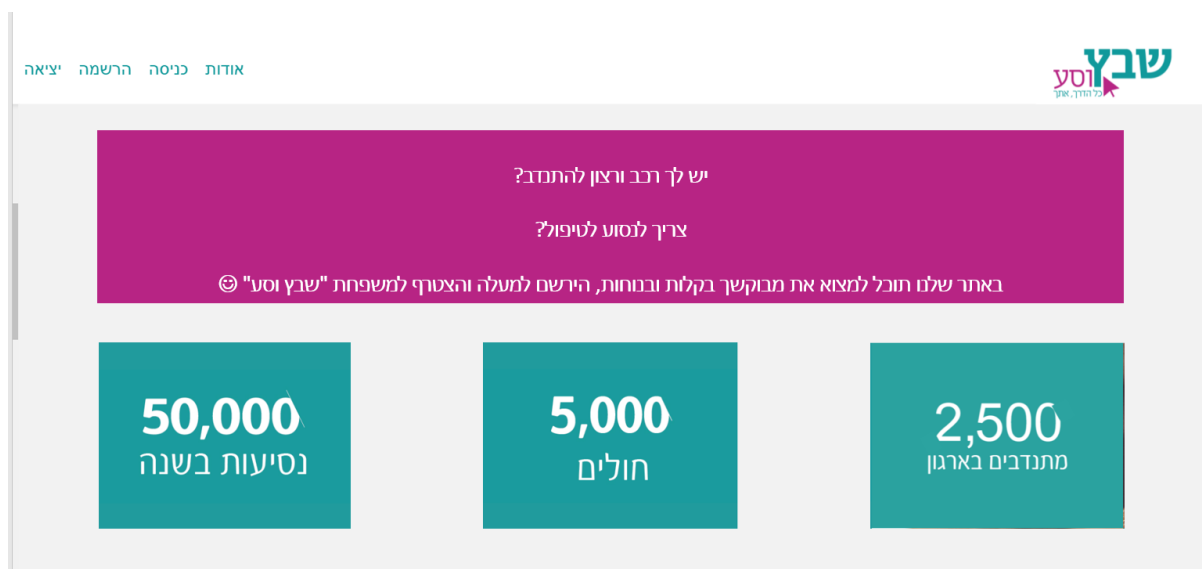
האתר ידידותי למשתמש. הקפדנו על עיצוב נעים לעין וברמת עיצוב גבוה ועדכנית, תוך שימת דגש על שימוש נח וקליט כדי לספק למשתמש חוויה נעימה ואת התחושה שהאתר מספק את צרכיו.

3.2 תרשים מסכים

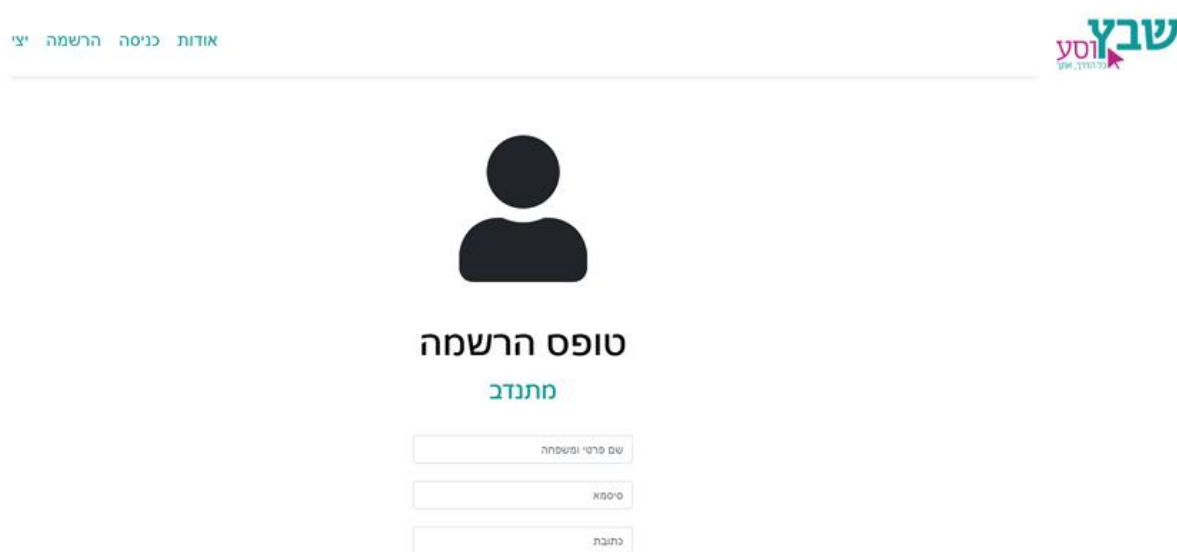


3.3 תיאור מסכים

דף הבית



הרשמה



המשך- הרשמה

אודות כניסה הרשמה יציאו

שבץ
וסע
מנהל, אשף

כתובת	<input type="text"/>
מיקוד	<input type="text"/>
פלאפון	<input type="text"/>
אימיל	<input type="text"/>
סוג רכב	<input type="text"/>
סטטוס	<input type="text"/>
שמירה	<input type="button" value="שמירה"/>

Powered by Efrat & Moriya

כניסה

אודות כניסה הרשמה יציאה

שבץ
וסע
מנהל, אשף



כניסה
מתנדב

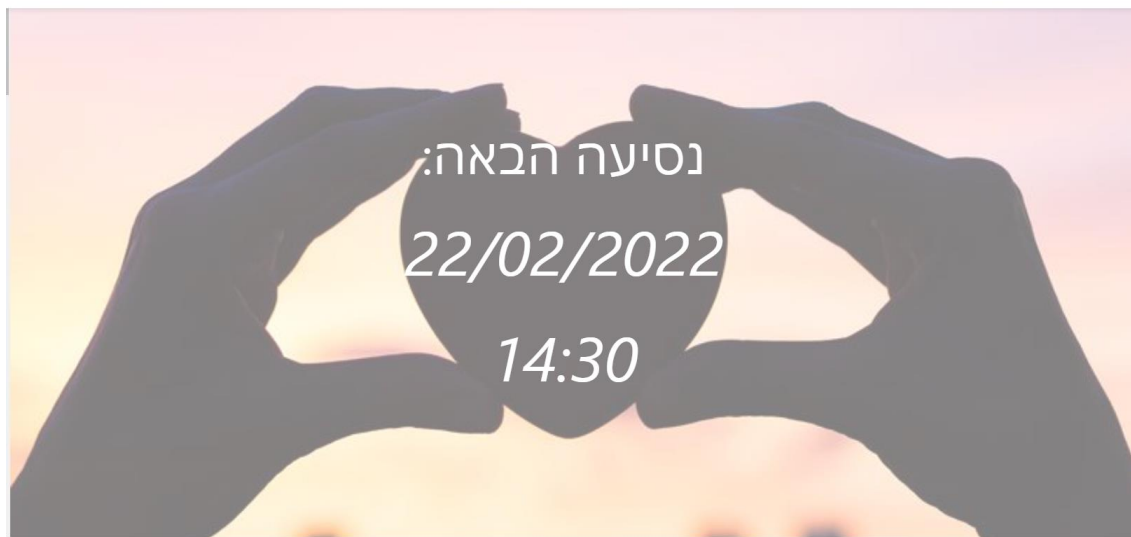
שם פרטי ומשפחה	<input type="text"/>
סיסמא	<input type="password"/>

פידבקים
נסיעות למחר
נסיעות קודמות
פרופיל

אזור אישי מתנדב ונוסע

אודות כניסה הרשמה יציאה

שבץ
נוסע
כל הודעה, נא



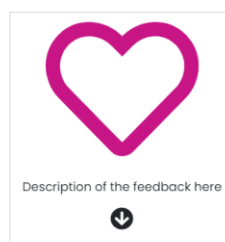
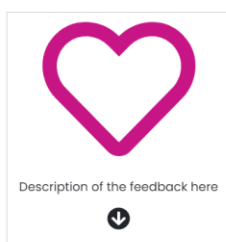
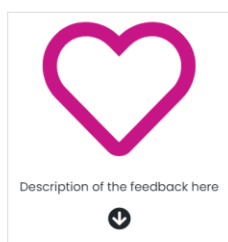
פידבקים
נסיעות למחר
נסיעות קודמות
פרופיל

פידבקים

אודות כניסה הרשמה יציאה

שבץ
נוסע
כל הודעה, נא

רצינו להגיד תודה



פידבקים
נסיעות למחר
נסיעות קודמות
פרופיל

נסיעות למחר

אודות כניסה הרשמה יציאה



נסיעות למחר

Handle	Last	First	#
mdo@	Otto	Mark	1
fat@	Thornton	Jacob	2
twitter@		Larry the Bird	3

פידבקים
נסיעות למחר
נסיעות קודמות
פרופיל

נסיעות קודמות

נסיעות קודמות

Handle	Last	First	#
mdo@	Otto	Mark	1
fat@	Thornton	Jacob	2
twitter@		Larry the Bird	3

פידבקים
נסיעות למחר
נסיעות קודמות
פרופיל

פרופיל

אודות כניסה הרשמה יציאה

שבץ
וסע
כל הודו, ארץ



הפרופיל שלי

מתנדב

שם פרטי ומשפחה

סימא

כתובת

פידבקים
נסיעות למחר
נסיעות קודמות
פרופיל

יצירת שיבוץ

אודות כניסה הרשמה יציאה

שבץ
וסע
כל הודו, ארץ

אישור

שיבוץ קודם

שיבוץ חדש

Handle	Last	First	#
mdo@	Otto	Mark	1
fat@	Thornton	Jacob	2
twitter@		Larry the Bird	3

פרופיל
יצירת שיבוץ
מתנדבים חדשים
מתנדבים
נוסעים

מתנדבים חדשים

אודות כניסה הרשמה יציאה



מתנדבים חדשים

Handle	Last	First	#
mdo@	Otto	Mark	1
fat@	Thornton	Jacob	2
twitter@		Larry the Bird	3

פרופיל

יצירת שיבוץ

מתנדבים חדשים

מתנדבים

נוסעים

מתנדבים

אודות כניסה הרשמה יציאה



מתנדבים

Handle	Last	First	#
mdo@	Otto	Mark	1
fat@	Thornton	Jacob	2
twitter@		Larry the Bird	3

פרופיל

יצירת שיבוץ

מתנדבים חדשים

מתנדבים

נוסעים

נוסעים



פרופיל

יצירת שיבוץ

מתנדבים חדשים

מתנדבים

נוסעים

אודות כניסה הרשמה יציאה

נוסעים

Handle	Last	First	#
mdo@	Otto	Mark	1
fat@	Thornton	Jacob	2
twitter@		Larry the Bird	3

4. מבנה נתונים וארגון קבצים

4.1 ארגון קבצים

הנתונים נשמרים בצורת טבלאית (structured) במסד הנתונים MySQL. כל המשתמשים שרשומים במערכת שמורים בטבלת המשתמשים, וכן כל זמני הפעילות, הזמנת הנסיעות, השיבוצי נסיעה, הסטורית נסיעות והעדפות שמורים ב-MySQL.

4.2 פירוט טבלאות מסד הנתונים

הנתונים נשמרים בתוך טבלאות, השדות נשמרים בצור הטבלאית (לכל שדה שם שנשמר במערכת של MySQL ולפי שמות השדות נשלפים הנתונים).

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
user_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Password	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Type	ENUM('Admin', 'Pass...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Phone_number	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Mail	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
City	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Neighborhood	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Street	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
City	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Neighborhood	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Street	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Num_places	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'5'
Car_leaflet	TINYINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
Casual_status	TINYINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
Approved_status	TINYINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
volunteer_schedule_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
volunteer_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
sunday	TINYINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
monday	TINYINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
tuesday	TINYINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
wednesday	TINYINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
thursday	TINYINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
friday	TINYINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
Hour	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
drive_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Passenger_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
volunteer_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Day	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Hour	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Hospital	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
passenger_approved	TINYINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
volunteer_approved	TINYINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
order_drive_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
user_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Hour	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Address	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Hospital	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idPreferences	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
userId	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
leaflet	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
familiar	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'
space	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Satisfaction_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Volunteer_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Text	LONGTEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Passenger_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Date	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

5. תכנון

5.1 כללי

האתר מתחלק לצד שרת וצד לקוח-

צד שרת נבנה ב-WEB API ובשפת NodeJS.

צד לקוח נבנה לפי מבנה הארכיטקטורות המובא לעיל, כדי להשתמש במירב היכולות ולהפיק את התועלת המירבית.

5.2 עקרונות תכנות

הפרויקט נבנה בגישת בשילוב OOP ו-Functional Programming

אלו שתי פרדיגמות כנות שונות, אך שתיהן מתאפשרות ב-JS.

ב-OOP השתמשנו על מנת לבנות את היררכית היישויות שלנו, והפונקציות הלאגוריתמיות נכתבו על פי תכנות פונקציונאלי.

5.3 תיאור אלגוריתם

האלגוריתם שיצרנו לשיבוץ המתנדבים עם הנוסעים כולל בתוכו כמה מרכיבים.

בתחילה- אנחנו שולפים ממסד הנתונים את כל המתנדבים שרלוונטים לנוסע שלנו- כלומר הם גרים באותה העיר, וטווח זמן הפעילות שלהם כולל את זמן הנסיעה של הנוסע.

לאחר מכן מגיע תורו של מרכיב עקרי באלגוריתם הכללי, והוא השימוש באלגוריתם ההונגרי המכונה גם-Munkres.

האלגוריתם הזה נועד לפתרון בעיות שיבוץ על פי הערכת עלות.

אנחנו השתמשנו בו עבור מיון מרשימת המתנדבים האפשריים לכל נוסע לרשימת מתנדבים אופציונליים, מדורגים מבעל הפוטנציאל הגדול ביותר לנמוך ביותר על פי האילוצים-העדפות המוגדרים לכל נוסע.

כעיקרון ישנן 2 דרכים לממש אותו, על ידי מטריצה ועל ידי גרף דו-צדדי.

אנחנו בחרנו לממש אותו בעזרת מטריצה, כיוון שזו דרך פשוטה וברורה יותר שיהיה קל לנו להכניס אליה את הנתונים.

עמודות המטריצה הן עבור המשימות-אילוצים במקרה שלנו, שעליהם יש לענות, והשורות הן העובדים- המתנדבים אותם מחפשים לשבץ.

האלגוריתם רץ בעזרת כמה שלבים עליהם הוא חוזר, שמתחילים במציאת התא בעל העלות המינמלית בכל שורה וממשיכים הלאה בשלבים הבאים על זו הדרך,

עד למציאת השיבוץ המתאים ביותר של כל מתנדב לכל אילוץ.

כיוון שהאלגוריתם המקורי עובד על מטריצות ריבועיות יש צורך להשתמש בפונקציה ייעודית שמוסיפה שורות דמה למען אופי עבודת האלגוריתם.

אדפטציה נוספת היא לעלות שההונגרי מחשב. עבורנו יש משמעות למציאת המתנדבים עם הרווח הכי גבוה, לצורך כך השתמשנו בפונקציה ייעודית נוספת שממירה את מטריצת הרווחים שלנו למטריצת עלויות כמו במקור, על ידי הפחתת הערך הגבוה ביותר בשורה בסך כל השורה, וכך נשמר היחס.

לאחר שקבלנו מההונגרי רשימת מתנדבים אפשריים עבור כל נוסע, אנחנו פונים לפונקציה שבדקת האם יש יותר מנוסע אחד שבמקום הראשון של רשימת המתנדבים שלו יש את אותו מתנדב-

אם לא- מצאנו שיבוץ אופטימלי והאלגוריתם נגמר.

אם כן, אנו שולחים לפונקציות למציאת הערכים של המתנדבים הכפולים הללו ומיקומם במערך המתנדבים.

את התוצאה שקבלנו אנחנו שולחות לפונקציה שבדקת על פי דירוג העדפות למי מהנוסעים בעלי המתנדב הכפול מגיע יותר, או דחוף יותר לזכות דווקא במתנדב הזה.

הנוסע בעל העדפות גבוהות יותר, יקבל עדיפות, כמובן ביחס לעדיפות שאנו קבענו לסך האילוצים.

מהנוסע השני אנו מסירות את המתנדב שלא זכה בו, ועוברות למתנדב הבא ברשימה שלו.

כך עבור כל נוסע ונוסע בדיקת הכפילות חוזרת ולאחריה בדיקת גובה העדיפות.

אם מגיעים למצב שנותר לנוסע מתנדב אחד אופציונלי, אנחנו נבדוק אם לו, ולנוסע השני אין בעיה מבחינת הגדרת עדיפות לנסוע באותו הרכב, ואם למתנדב המדובר יש מספיק מקום פנוי עבור 2.

במידה וכן- הן משובצים שתיהם לאותה נסיעה.

במידה ולא- הנוסע שלא מתאים לו נסיעה ב-2 עובר למתנדב הבא ברשימה, כיוון שלחברו הוא המתנדב היחיד.

במידה ושתי כפולים שאנו בודקים מולם בלעי מתנדב אחד בלבד- שוב נבדוק את אפשרות נסיעה ב-2 ואם לאו נסיר לעקשן מביניהם את המתנדב ובגמר השיבוץ כשנשלח את פלט השיבוץ נשלח גם את הרשימה של המתנדבים שלא נמצא להם שיבוץ כלל.

ובאן המקום למח האנושי- למנהל להכריע אם לשנות בעצמו חלק מהשיבוץ, או כפי שהאתר מציע לנסות ולהקפיץ מתנדבים שמוגדרים כ-"מזדמנים" מתוך הרשימה שבמסד הנתונים.

להלן תצוגות של חלק מהפונקציות באלגוריתם:

דגימות מהפונקציות של ההונגרי:

```
function make_cost_matrix(profit_matrix, inversion_function) {
  var i, j;
  if (!inversion_function) {
    var maximum = -1.0 / 0.0;
    for (i = 0; i < profit_matrix.length; ++i)
      for (j = 0; j < profit_matrix[i].length; ++j)
        if (profit_matrix[i][j] > maximum)
          maximum = profit_matrix[i][j];
    inversion_function = function (x) { return maximum - x; };
  }

  var cost_matrix = [];

  for (i = 0; i < profit_matrix.length; ++i) {
    var row = profit_matrix[i];
    cost_matrix[i] = [];

    for (j = 0; j < row.length; ++j)
      cost_matrix[i][j] = inversion_function(profit_matrix[i][j]);
  }

  return cost_matrix;
}
```

```

Munkres.prototype.__step5 = function () {
    var count = 0;

    this.path[count][0] = this.Z0_r;
    this.path[count][1] = this.Z0_c;
    var done = false;

    while (!done) {
        var row = this.__find_star_in_col(this.path[count][1]);
        if (row >= 0) {
            count++;
            this.path[count][0] = row;
            this.path[count][1] = this.path[count - 1][1];
        } else {
            done = true;
        }

        if (!done) {
            var col = this.__find_prime_in_row(this.path[count][0]);
            count++;
            this.path[count][0] = this.path[count - 1][0];
            this.path[count][1] = col;
        }
    }

    this.__convert_path(this.path, count);
    this.__clear_covers();
    this.__erase_primes();
    return 3;
};

```

```

Munkres.prototype.compute = function (cost_matrix, options) {

  options = options || {};
  options.padValue = options.padValue || DEFAULT_PAD_VALUE;

  this.C = this.pad_matrix(cost_matrix, options.padValue);
  this.n = this.C.length;
  this.original_length = cost_matrix.length;
  this.original_width = cost_matrix[0].length;

  var nfalseArray = []; /* array of n false values */
  while (nfalseArray.length < this.n)
    nfalseArray.push(false);
  this.row_covered = nfalseArray.slice();
  this.col_covered = nfalseArray.slice();
  this.Z0_r = 0;
  this.Z0_c = 0;
  this.path = this.__make_matrix(this.n * 2, 0);
  this.marked = this.__make_matrix(this.n, 0);

  var step = 1;

  var steps = {
    1: this.__step1,
    2: this.__step2,
    3: this.__step3,
    4: this.__step4,
    5: this.__step5,
    6: this.__step6
  };

  while (true) {
    var func = steps[step];
    if (!func) // done
      break;
    step = func.apply(this);
  }

  var results = [];
  for (var i = 0; i < this.original_length; ++i)
    for (var j = 0; j < this.original_width; ++j)
      if (this.marked[i][j] == 1)
        results.push([i, j]);

  return results;
};

```

פונקציה למציאת מתנדבים כפולים וסידורם:

```

var statusOfMoreThenOneInCar = false;

function FixDuplicateOptimalVolunteer(volunteersForPassengersList) {
  var str = JSON.stringify(volunteersForPassengersList);
  console.log("enter FixDuplicateOptimalVolunteer function with passengers: " + str);
  var problematicPassengers = [];
  var indexProblematicPassengers = [];
  var firstVolunteers = [];
  var indexOfTwoInCar = [];
  //get passengers first volunteers
  volunteersForPassengersList.map((o, i) => { firstVolunteers.push(o.volunteers[0]) });
  console.log(JSON.stringify(firstVolunteers));

  //return true if there are duplicates and false if not.
  var isThereDuplicate = firstVolunteers.some((val, i) => firstVolunteers.indexOf(val) !== i);
  if (!isThereDuplicate)
    console.log("no duplicates");
  else {
    while (isThereDuplicate) {
      problematicPassengers = getAllDuplicatesVolunteers(firstVolunteers);
      console.log("problematicPassengers: " + JSON.stringify(problematicPassengers));
      for (let i = 0; i < problematicPassengers.length; i++) {
        console.log("enter the for loopppp :");
        indexProblematicPassengers = getTheIndexesOfDuplicateVolunteer(problematicPassengers[i], firstVolunteers);
        console.log("indexProblematicPassengers: " + JSON.stringify(indexProblematicPassengers));
        checkHigherRankAndSwap(volunteersForPassengersList, indexProblematicPassengers);
        volunteersForPassengersList.map((o, i) => { firstVolunteers.push(o.volunteers[0]) });
        if (statusOfMoreThenOneInCar) {
          while (indexOfTwoInCar !== null) {
            firstVolunteers.splice(indexOfTwoInCar, indexOfTwoInCar + 2);
            indexOfTwoInCar.pop();
          }
        }
        isThereDuplicate = firstVolunteers.some((val, i) => firstVolunteers.indexOf(val) !== i);
      }
    }
  }
  console.log("schedule finished!");
}

```

פונקציה לבדיקת גובה האילוצים ושיבוץ המתנדבים בהתאם:

```

function checkHigherRankAndSwap(volunteersForPassengersList, indexProblematicPassengers) {
    var lowestRankIndex = -1;
    let indexOffTwoInCar = -1;
    while (indexProblematicPassengers.length > 1) {
        let i = indexProblematicPassengers.length - 1;
        if (volunteersForPassengersList[indexProblematicPassengers[i]].prefRank["leftlet"] < volunteersForPassengersList[indexProblematicPassengers[i - 1]].prefRank["leftlet"]) {
            lowestRankIndex = indexProblematicPassengers[i];
            console.log("check ranks: " + JSON.stringify(volunteersForPassengersList[indexProblematicPassengers[i]].prefRank["leftlet"]));
        }
        else if (volunteersForPassengersList[indexProblematicPassengers[i]].prefRank["leftlet"] > volunteersForPassengersList[indexProblematicPassengers[i - 1]].prefRank["leftlet"]) {
            lowestRankIndex = indexProblematicPassengers[i - 1];
        }
        else if (volunteersForPassengersList[indexProblematicPassengers[i]].prefRank["space"] < volunteersForPassengersList[indexProblematicPassengers[i - 1]].prefRank["space"]) {
            lowestRankIndex = indexProblematicPassengers[i];
            console.log("check ranks: " + JSON.stringify(volunteersForPassengersList[indexProblematicPassengers[i]].prefRank["space"]));
        }
        else if (volunteersForPassengersList[indexProblematicPassengers[i]].prefRank["space"] > volunteersForPassengersList[indexProblematicPassengers[i - 1]].prefRank["space"]) {
            lowestRankIndex = indexProblematicPassengers[i - 1];
        }
        else if (volunteersForPassengersList[indexProblematicPassengers[i]].prefRank["familiar"] < volunteersForPassengersList[indexProblematicPassengers[i - 1]].prefRank["familiar"]) {
            lowestRankIndex = indexProblematicPassengers[i];
            console.log("check ranks: " + JSON.stringify(volunteersForPassengersList[indexProblematicPassengers[i]].prefRank["familiar"]));
        }
        else if (volunteersForPassengersList[indexProblematicPassengers[i]].prefRank["familiar"] > volunteersForPassengersList[indexProblematicPassengers[i - 1]].prefRank["familiar"]) {
            lowestRankIndex = indexProblematicPassengers[i - 1];
        }
    }

    //if they have same preferences pop the first one- no matter who.
    if (lowestRankIndex === -1) {
        //if he only has one volunteer left
        if (volunteersForPassengersList[indexProblematicPassengers[i]].volunteers.length === 1) {
            indexOffTwoInCar.push(onlyOneVolunteerSituation(volunteersForPassengersList, indexProblematicPassengers, i));
        }
        else {
            volunteersForPassengersList[indexProblematicPassengers[i]].volunteers.shift();
        }
    }

    if (volunteersForPassengersList[lowestRankIndex].volunteers.length === 1) {
        indexOffTwoInCar.push(onlyOneVolunteerSituation(volunteersForPassengersList, indexProblematicPassengers, lowestRankIndex));
    }
    volunteersForPassengersList[lowestRankIndex].volunteers.shift();
    console.log("new volunteer list: " + JSON.stringify(volunteersForPassengersList[lowestRankIndex].volunteers));
    indexProblematicPassengers.splice(lowestRankIndex, lowestRankIndex + 1);
}
return indexOffTwoInCar;
}

```


6. מה הקנה הפרויקט

- 6.1 התנסות באפיון מערכת מתחילתה ותכנון כל חלקיה עד לגמר.
- 6.2 התנסות בתקשורת בין שרתים שונים, בין צד השרת לצד הלקוח ולמסד הנתונים
- 6.3 הכרת ספריות חדשות מטרותיהן ואופן השימוש בהן.
- 6.4 חידוד יכולת חיפוש מודולים ודומיהם חדשים שעשויים לשרת את מטרותינו ברגע.
- 6.5 ניסיון מעשי בפיתוח ב-React JS.
- 6.6 ניסיון מעשי בפיתוח צד שרת ב-JS.
- 6.7 ניסיון שלא יסולא בפז בתכנון אלגוריתם שיבוץ מאפס, וכתיבתו עד לתוצאה הרצויה ב-JS.
- 6.8 השתפשות בפתרון בעיות, והתמודדות אתן.
- 6.9 חידוד אופן תכנון מערכת ובניית האלגוריתם שלה.
- 7.0 פיתוח יכולת לימוד עצמי חזקה.
- 7.1 ניסיון בשימוש במסד הנתונים MySQL.
- 7.2 השחזת יכולת החיפוש באנטרנט וסינון התוצאות הרלוונטיות לתפרון בעיות באופו עצמאי.

7. ביבליוגרפיה

- 7.1 <https://stackoverflow.com/>
- 7.2 <https://github.com/>
- 7.3 <https://he.wikipedia.org/wiki>
- 7.4 <https://react-bootstrap.netlify.app/getting-started/introduction/>
- 7.5 <http://darcheimiriam.org.il/>
- 7.6 <https://www.w3schools.com/>

8. נספחים








דרכי מרים

מסודו של הרב ישראל וינגרטן זצ"ל לעיני רעותו מרת מרים ע"ה

לעזרת חולי סרטן ומשפחותיהם


בית / אודות / פעילות / חדשות / גלריה / המלצות / תרומות / התנדבות / צור קשר


English


18

שנות פעילות









720,000

מספר נשים שאנחנו עוזרים

5,000

מספר תרומות