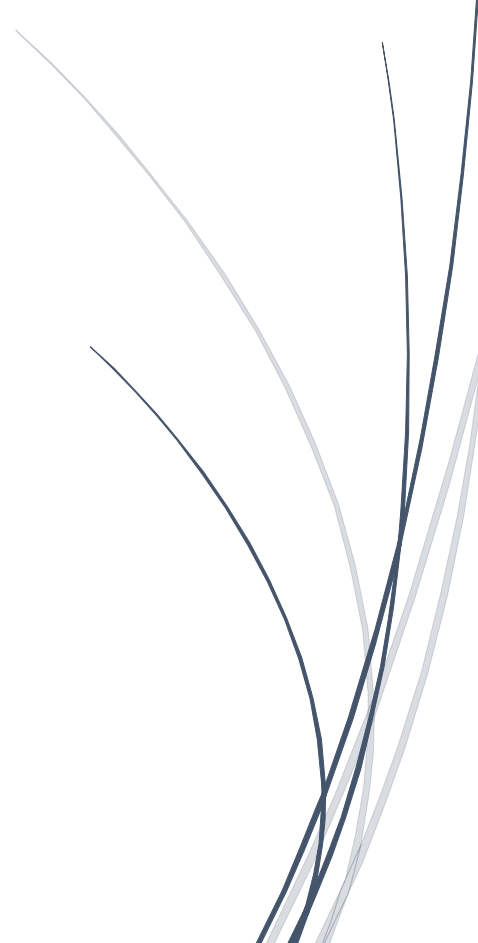


מגישה: שפרה כהן
ת.ז: 213238132

ספר פרויקט אתר תרומות



תוכן עניינים

1.....	תוכן עניינים
2.....	שלמי תודה
3.....	מבוא לפרויקט
4.....	תקציר
5.....	תיאור הפרויקט
7.....	הדרישות בפיתוח המערכת
7.....	מפרט תכני
7.....	מטרות ויעדים
7.....	תיחום המערכת
8.....	ניתוח המערכת
8.....	תרשים UML
9.....	תיאור ה DB וקישרי הגומלין
9.....	תיאור הטבלאות בפרויקט
13.....	מבנה הקוד בפרויקט
14.....	תיאור ה COMPONENTS
24.....	מסכים
30.....	מסקנות
31.....	סיכום
32.....	ביבליוגרפיה

שלמי תודה

- בפתח דבר אודה לבורא עולם שהביאני עד הלום וסייע לאורך כל הדרך.
- בנוסף אודה לכל אלו שנתנו רבות מכוחם, מרצם ומתבונתם במהלך העבודה.
- תודה- למנחה, גב' שולמית ברלין שתחי', שסייעה לי לאורך כל הדרך בהכנת הפרויקט, הנחתה בשלבים השונים, החל משלב התכנון ועד לשלב הביצוע.
 - ייעצה, בדקה ונתנה חוות דעת על מנת להגיע להישגים הטובים ביותר במסגרת הזמן והאמצעים אשר עמדו לרשותי.
 - תודה- לצוות המורות המסור על המקצועיות הרבה, העזרה הרבה, התמיכה וההשקעה לטווח הרחוק.
 - לסמינר בנות אלישבע, על המקצועיות הרבה, וההשקעה לטווח הרחוק.
 - תודה- להורי היקרים ולשאר בני המשפחה וידידים על תמיכה ועידוד לאורך כל הדרך החל מתקופת הלימודים ועד להגשת פרויקט גמר.

מבוא לפרויקט

מטרתו העיקרית של הפרויקט היא לתקשר בין צד ה server לצד ה client להעביר נתונים ל db, לקרוא נתונים מה db, למחוק נתונים מה db, ולעדכן נתונים ב db ע"י קריאות של ה client ל server וכתובת שאילתות ב MySQL, וכן מטרת ההתלמדות ורכישת הידע.

הפרויקט הוא אתר חינמי העוסק בתרומות לבתי כנסיות אפשר בקלות מאוד להפוך את האתר לאתר של תרומות לכל דבר אחר כגון תרומות לבתי חולים.

האתר כולל את ממשק התורם המאפשר: תרומה לפי פריט, תרומה לפי בית כנסת או תרומה חופשית, התורם יכול לראות את התרומות שלו התורם יכול לתרום רק עם שם משתמש וסיסמא.

וכן את ממשק הגבאי המאפשר: ניהול פריטים שנדרשים לבית הכנסת – הוספת פריט למערכת, מחיקת פריט מהמערכת, עדכון פריט במערכת, כל התרומות, התרומות שעדיין לא נתרמו, הקדשות שלא טופלו.

בנוסף האתר שלי עוזר למקבלי התרומות לדעת מה תרמו לבית כנסת שלהם איזה הקדשות עדיין לא טופלו ואיזה פריטים עדיין לא נתרמו בצורה מאוד מסודרת וקלה.

וכמו כן האתר מקל על התורם שרוצה לתרום פריט מסוים, התורמים לא צריכים לחפש איזה בית כנסת צריך את הפריט אלה האתר מציג להם את בתי הכנסיות הנזקקים לפריט שהם רוצים לתרום, וכמו כן אם התורם רוצה לדעת איזה פריטים בית כנסת מסוים צריך האתר יציג לו.

תקציר

האתר מנהל מערכת תרומות לבתי כנסיות, הוא מנהל את התורם ואת הגבאי, לתורם יש מספר אפשרויות איך לתרום – תרומה חופשית, תרומה לפי פריט, תרומה לפי בית כנסת.

האתר

האתר נועד לתווך בין שני אנשים אחד מציע נכס והשני מבקש.

ממשק

הפרויקט פותח בסביבת עבודה vsCode צד השרת נכתב בשפת node.js תוך חלוקה נכונה ומסודרת של הקוד. וטכנולוגית react לבניית צד הלקוח. האתר מספק מראה ויזואלי ברמה גבוהה המתאים את עצמו לגודל ההתקן שבו הוא נפת, באופן כללי הושקעה באתר מחשבה רבה הן לצורך יצירת אתר מתקדם ונאה לשימוש והן לצורך תפעול פרקטי, נוח ופשוט.

הספר

מעיין בספר זה ימצא תיאור נרחב של הפרויקט הן בפן הלוגי והן בפן הגרפי החיצוני שישתלב במערכת הכללית. במהלך הספר ניתן להבחין בעקרונות התכנון של המערכת - תהליכים שקיימים במערכת, תרשימים ועוד... ובנוסף מכיל הספר מדריך למשתמש בו נמצאים צילומי מסך והסברים כיצד ניתן להשתמש במגוון האפשרויות אותה האתר מספק.

תיאור הפרויקט

הפרויקט מחולק לשתי חלקים:

1. צד התורם – אם המשתמש נכנס כתורם המערכת תציג לו את האפשרויות הבאות: תרומה לפי פריט, תרומה לפי בית כנסת, תרומה חופשית, התרומות שנתרמו על ידו, ויציאה מהמערכת.
 2. צד הגבאי – אם משתמש נכנס כגבאי המערכת תציג לו את האפשרויות הבאות: הוספת פריט לתרומה, התרומות שעדיין לא נתרמו, כל התרומות, הקדשות שעדיין לא טופלו, ויציאה מהמערכת.
- האתר נועד לעזור לתווך בין התורם לבין הגבאי באופן יעל וקל.
- אם משתמש נרשם לאתר גם כגבאי וגם כתורם הוא יצטרך לבחור איך הוא נכנס – כתורם או כגבאי המערכת תפתח לו את הדף המתאים.
- נפרט על האופציות שלכל צד:

1. צד התורם:

1. תרומה לפי פריט – המשתמש בוחר את הפריט שהוא רוצה לתרום (יש אפשרות לחפש פריט מסוים) והמערכת מציגה לו את בתי הכנסיות שצריכים את הפריט שנבחר המשתמש בוחר את הבית כנסת הרצוי ולאחר מכן הוא יעבור לתשלום ששם יכניס את פרטיו ופרטי כרטיס האשראי והקדשה, כמובן רק אם הוא נרשם במערכת אחרת הוא יגיע לדף של הרישום למערכת ורק אחר כך יגיע לדף התשלום.
2. תרומה לפי בית כנסת - המשתמש בוחר את בית הכנסת שאליו הוא מעונין לתרום (יש אפשרות לחפש בית כנסת מסוים) והמערכת מציגה לו את הפריטים שבית הכנסת הבחור זקוק להם המשתמש בוחר את הפריט הרצוי ולאחר מכן הוא יעבור לתשלום ששם יכניס את פרטיו ופרטי כרטיס האשראי והקדשה, כמובן רק אם הוא נרשם במערכת אחרת הוא יגיע לדף של הרישום למערכת ורק אחר כך יגיע לדף התשלום.
3. תרומה חופשית – רק לאחר שהמשתמש נרשם למערכת ניתן לבצע את התרומה החופשית, המשתמש יכניס את הסכום הרצוי את פרטיו, פרטי כרטיס אשראי והקדשה.
4. התרומות שנתרמו על ידו – המערכת תציג לתורם את התרומות שתרם.
5. יציאה מהמערכת – המערכת תוציא אותו מהמערכת ע"י שתרוקן את `sessionStorage`.

2. צד הגבאי:

1. הוספת פריט לתרומה – הגבאי יגיע לדף של הוספת פריט ששם הוא יזין שם פריט, כמות, מחיר ליחיד, אם הפריט כבר קיים המערכת תודיע על כך.
2. התרומות שעדיין לא נתרמו – המערכת תציג את התרומות שעדיין לא נתרמו כך: מספר פריטים שעדיין לא נתרמו X מתוך Y, עם אופציות של מחיקה ועדכון של הפריט.
3. כל התרומות – המערכת תחזיר את כל התרומות שנתרמו ושלא נתרמו כך: נתרמו X מתוך Y.
4. הקדשות שעדיין לא טופלו – המערכת תציג את כל ההקדשות שלא טופלו, כאשר הגבאי מסמן הקדשה שהיא טופלה היא לא תוצג עוד
5. יציאה מהמערכת – המערכת תוציא אותו מהמערכת ע"י שתרוקן את ה sessionStorage.

הדרישות בפיתוח המערכת

בבניית המערכת יש לדאוג שתהיה נוחה, קלה ומובנת למשתמש ושהקוד יפעל בצורה המדויקת ביותר, וכמובן שיהיה יעיל ויעבוד כראות.

מפרט תכני

1. עמדת פיתוח: מחשב intel.
מערכת הפעלה: windows 10.
כלי התוכנה לפיתוח המערכת: visual studio code.
שפות תכנות: צד שרת – node js, צד לקוח – react.
מסד נתונים – MySql.
2. חומרה: מחשב.
חיבור לרשת: לא נדרש.

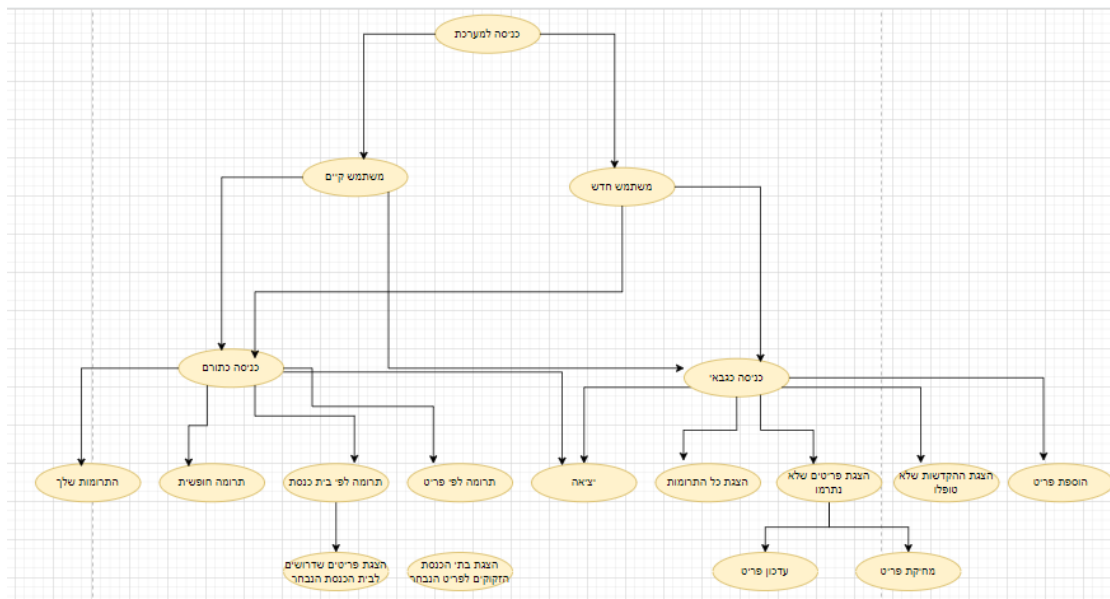
מטרות ויעדים

1. הבקשות להקדשות של תורמים נשמרים במערכת עד לטיפולם.
2. להקל הליכים בקבלת תרומות ובכך להגדיל את מחזור ההכנסות.
3. קבלת הקבלות על התרומות למייל וכך נשמרות הקבלות לצורך קבלת החזר מס בגין סעיף 46.

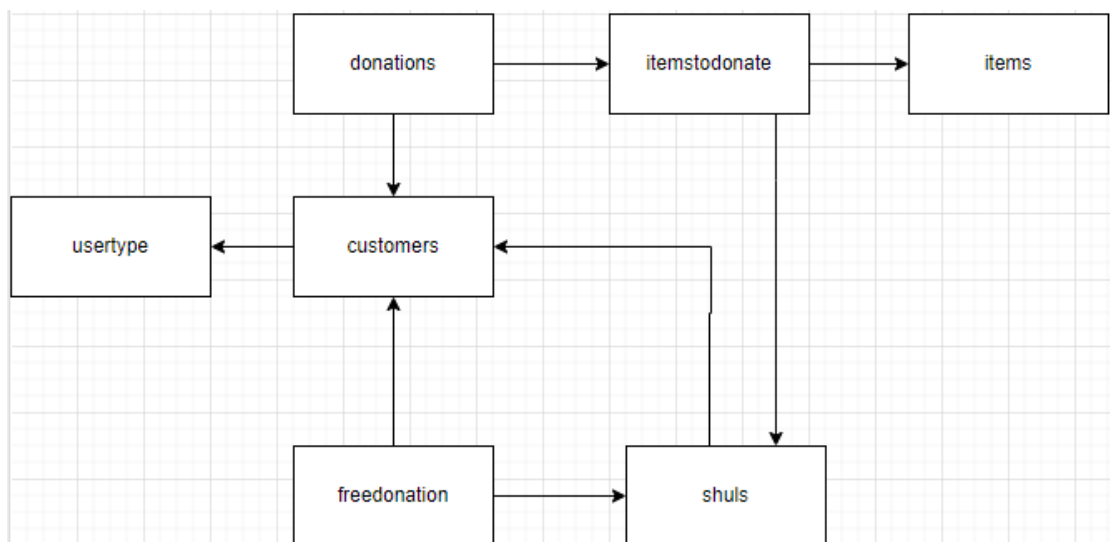
תיחום המערכת

1. המערכת מאפשרת להכניס נתונים תקינים בלבד.
2. המערכת מתחייבת ל100 אחוזי הצלחה בדיוק שליפת המידע.
3. המערכת מותאמת לתוכנית הלימודים הקיימת במגמה ותקיף פעולות אלו בלבד.

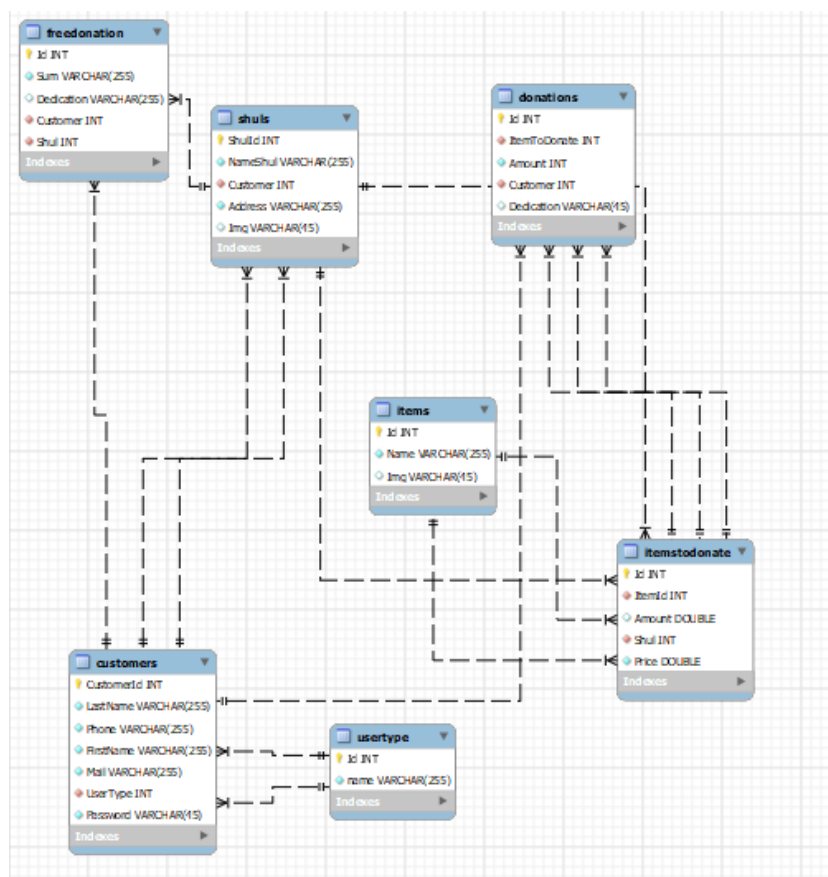
ניתוח המערכת



תרשים uml



תיאור ה db וקישרי הגומלין



תיאור הטבלאות בפרויקט

טבלת המשתמשים – customers

שם שדה	סוג שדה	תיאור	
customerId	Int	מזהה המשתמש (ר.ת.)	מפתח
FirstName	string	שם פרטי של המשתמש	
lastName	string	שם משפחת של המשתמש	
Phone	string	טלפון של המשתמש	

מייל של המשתמש	string	Mail	
סיסמא של המשתמש	string	Password	
קוד סוג המשתמש	int	UserType	

טבלת התרומות – donaions

תיאור	סוג שדה	שם שדה	
מזהה התרומה	Int	IdDonation	מפתח
קוד פריט	Int	ItemToDonate	
כמות מהפריט	Int	Amount	
קוד משתמש	Int	Customer	
הקדשה	string	Dedication	

טבלת התרומות – freedonation

תיאור	סוג שדה	שם שדה	
מזהה התרומה החופשית	Int	IdFreedonatio	מפתח
סכום לתרומה	Int	Sum	
הקדשה	string	Dedication	
קוד משתמש	Int	Customer	
קוד בית כנסת	int	Shul	

טבלת התרומות – items

מפתח	שם שדה	סוג שדה	תיאור
	IdItems	Int	מזהה פריט
	Name	Int	שם הפריט
	Img	string	תמונה של הפריט

טבלת התרומות – itemstodonate

מפתח	שם שדה	סוג שדה	תיאור
	IdItemstodonate	Int	מזהה פריט לתרומה
	ItemId	Int	קוד מזהה פריט
	Amount	int	כמות פריט
	Shul	Int	קוד בית כנסת
	Price	int	סכום לפריט בודד

טבלת התרומות – shuls

שם שדה	סוג שדה	תיאור	
idShul	Int	מזהה בית כנסת	מפתח
NameShul	string	שם בית כנסת	
Customer	Int	קוד משתמש	
Address	string	כתובת בית כנסת	
Img	string	תמונה של בית הכנסת	

טבלת התרומות – usertype

שם שדה	סוג שדה	תיאור	
idUserstype	Int	מזהה סוג משתמש	מפתח
name	string	שם סוג משתמש	

מבנה הקוד בפרויקט

צד הלקוח מתקשר עם צד השרת וצד השרת מתקשר עם ה database

- צד הלקוח – UI נכתב ב components ב react, היא מבצעת fetch לשרת כדי לקבל להוסיף לעדכן נתונים ב database.

- צד השרת נכתב ב node.js, השתמשתי ב mySql2 – ישנן 2 תיקיות:

1. routes – בה כל קובץ מטפל בנושא מסוים לדוגמא: קריות של הגבאי, קריאות של התורם וכו.

כל קריאה שמתקבלת מצד הלקוח מגיעה לקובץ server.js והוא מנווט לקובץ המתאים בתקיה הזאת, בתוך כל קובץ יש פונקציות לפי ה url של הקראות בתוך הפונקציות יש זימון של פונקציה בתוך התיקיה Services בתוך הקובץ שמתאים לנושא.

2. Services – בה יש קבצים שכל קובץ מטפל בנושא אחר כמו ה routes בתוך הקבצים יש פונקציות שכל פונקציה מבצעת שאילתת sql.

- bataBase – נכתב ב mySql.

תיאור ה components

:AddItem-component

מכילה פונקציה AddClicked() שהיא מוסיפה פריט נוסף לרשימת הפריטים שעדיין לא נתרמו או מוסיפה במספר הכמות שנמצאת במאגר, הפונקציה מבצעת את ה fetch הבא:

```
const response = await fetch('http://localhost:6200/api/item/add', {
  method: 'POST',
  body: formData,
})
```

ב server מקבל את הקריאה עם ה body שבמקרה הזה זה יהיה המידע על המוצר שהגבאי מעוניין להוסיף, לפי השורה הזאת בקובץ server.js:

```
app.use('/api/item', item);
```

הקריאה תגיעה לקובץ item שבתיקית routes ושם הוא יגיע לפונקציה הבאה לפי ה url של ה fetch:

```
router.post('/add', shul.upload.single('file'), async function (req, res) {
  try {
    res.json(await item.createItem(req.body))
  }
  catch (err) {
    console.error(`Error the shul not create`, err.message);
  }
})
```

בפונקציה הזאת מזמנים את הפונקציה createItem(req.body) ששולחים אליה את ה body שזה המידע על המוצר, בפונקציה הזאת בצעתי בדיקה אם המוצר שהגיע כבר קיים במאגר הפריטים במידה שהוא לא קיים הוספתי למאגר של הפריטים, לאחר מכן בדקתי אם הכמות שהגבאי רוצה להוסיף / לשנות הוא קטן מהכמות שכבר תרמו או שהכמות שהוא הכניס היא 0 לא נשנה ונחזיר הודאה מתאימה במידה ולא נכניס לפריטים שעדיין לא נתרמו עם הכמות שהגבאי הכניס.

:Confirmation-component

הלקוח יכניס את פרטי האשראי לביצוע התרומה בלחיצה על כפתור האישור
יתבצע ה fetch הבא:

```
const url = `http://localhost:6200/api/donation`;
let response = await fetch(url, {
  method: 'POST',
  headers: {
    'Accept': 'application/json',
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(date)
});
```

ב server מקבל את הקריאה עם ה body שבמקרה הזה זה פרטי הלקוח
התורם, לפי השורה הזאת בקובץ server.js:

```
app.use('/api/donation', donation);
```

הקריאה תגיע לקובץ donation שבתיקית routes ושם הוא יגיע לפונקציה הבאה
לפי ה url של ה fetch:

```
router.post('/', async function (req, res, next) {
  try {
    res.json(await donation.addDonation(req.body));
  } catch (err) {
    console.error(`Error`, err.message);
    next(err);
  }
});
```

בפונקציה הזאת מזמנים את הפונקציה addDonation(req.body) ששולחים
אליה את פרטי המשתמש ובפונקציה נכניס את המשתמש לתרומות שנתרמו
עם ההקדשה שלו.

:Dedications-component

הקומפוננטה מציגה לגבאי את ההקדשות של התורמים – בקשות של תרומות
של פריטים ותרומות חופשיות לכל הקדשה יש אפשרות לטיפול ובלחיצה עליו
אם זה תרומה חופשית יתבצע ה fetch הבא:

```
const url = `http://localhost:6200/api/dedication/freeHandling`;
let response = await fetch(url, {
  method: 'PUT',
```



```

    headers: {
      'Accept': 'application/json',
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(data)
  });

```

ואם זה יהיה תרומה של פריט יתבצע ה fetch הבא:

```

const url = `http://localhost:6200/api/dedication/itemHandling`;
let response = await fetch(url, {
  method: 'PUT',
  headers: {
    'Accept': 'application/json',
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(data)
});

```

ב server מקבל את הקריאה עם ה body שבמקרה הזה זה פרטים של ההקדשה שנבחרה, לפי השורה הזאת בקובץ server.js:

```
app.use('/api/dedication', dedication);
```

שתי הקריאות האלה יגיעו לקובץ dedication ולכל אחד מהקריאות תתבצע הפונקציה המתאימה לפי ה url:

```

router.put('/freeHandling', async function (req, res, next) {
  try {
    console.log(req.body)
    res.json(await dedication.freeHandling(req.body));
  } catch (err) {
    console.error(`Error`, err.message);
    next(err);
  }
});

router.put('/itemHandling', async function (req, res, next) {
  try {
    console.log(req.body)
    res.json(await dedication.itemHandling(req.body));
  } catch (err) {
    console.error(`Error`, err.message);
    next(err);
  }
});

```

בפונקציות freeHanging, itemHandling מזין null בעמודה של ההקדשה את השורה של ההקדשה בטבלת ההקדשות המתאימה.

:DonatedItems -component

הקומפוננטה מציגה לגבאי את התרומות שנתרמו – לפי fetch שכתבתי ב useEffect תרומות של פריטים ותרומות חופשיות לכל תרומה, אם לא נתרמו נציג הודאה מתאימה.

:Donates -component

מציג את בתי הכנסת הקיימים במאגר עם כפתור של הצג עוד בתי כנסת רק במידה ויש עוד בתי כנסת בלחיצה עליו נבצע fetch שיבדוק אם יש עוד במידה ואין הוא יסיר את הכפתור של הצג עוד ויצג את בתי הכנסת שהוא הביא מהשרת. עם אופציה של חיפוש בית כנסת, בלחיצה על בית כנסת מסוים ננווט את המשתמש לקומפוננטה ItemsOfShul ונשלח לו את ה props שיהיה הבית כנסת שנבחר.

:ItemsOfShul -component

בuseEffect נבצע fetch ונשלח את ID של הבית הכנסת שנבחר:

```
const url =
`http://localhost:6200/api/item/notDonations/${location.state.shul.ShulId}`;
```

ב server מקבל את הקריאה עם ה body שבמקרה הזה זה יהיה ID של בית הכנסת, לפי השורה הזאת בקובץ server.js:

```
app.use('/api/item', item);
```

הקריאה תגיעה לקובץ item שבתיקת routes ושם הוא יגיע לפונקציה הבאה לפי ה url של ה fetch:

```
router.get('/notDonations/:id', async function (req, res, next) {
  let shulId = req.params.id;
  console.log(shulId);
  try {
    res.json(await item.allItemsNotDonated(shulId));
  } catch (err) {
    console.error(`Error`, err.message);
    next(err);
  }
});
```

בפונקציה `allItemsNotDonation(shuld)` נשלח את כל הפריטים שעדיין לא נתרמו או שלא תרמו את הכמות שצריכים.

הקומפוננטה תציג את הפריטים שהגיעו מה `fetch` שלכל פריט יש אפשרות ללחוץ לתרומה, עם אפשרות לחיפוש מוצר בלחיצה על תרומה אם המשתמש לא התחבר נעביר אותו לדף `login` ולאחר מכן נעביר אותו לדף `confirmation` במידה והוא התחבר נעביר אותו ל `confirmation` בשתי המקרים נשלח ב `props` את ה פריט כמות לתרומה ומחיר.

:`DonationByItem-component`

הקומפוננטה תציג לתורם את כל הפריטים שנמצאים במאגר ע"י `fetch`:

```
let response = await
fetch(`http://localhost:6200/api/item/all/${fromIndex}`);
```

שנכתוב ב `useEffect`, עם כפתור של הצג עוד פריטים רק במידה ויש עוד פריטים בלחיצה עליו נבצע `fetch` שיבדוק אם יש עוד במידה ואין הוא יסיר את הכפתור ויציג את הפריטים שהוא הביא מהשרת. עם אופציה של חיפוש פריט. בלחיצה על פריט מסוים ננווט את המשתמש לקומפוננטה `ShulsOfItem` ונשלח לו את ה `props` שיהיה הפריט שנבחר.

:`ShulsOfItem -component`

ב `useEffect` נבצע `fetch` ונשלח את ה ID של הפריט שנבחר:

```
let response = await fetch(
`http://localhost:6200/api/item/${location.state.itemId}/getShuls`);
```

ב `server` מקבל את הקריאה עם ה `body` שבמקרה הזה זה יהיה ה ID של הפריט שנבחר, לפי השורה הזאת בקובץ `server.js`:

```
app.use('/api/item', item);
```

הקריאה תגיעה לקובץ `item` שבתיקת `routes` ושם הוא יגיע לפונקציה הבאה לפי ה url של ה `fetch`:

```
router.get('/:id/getShuls', async function (req, res, next) {
  let itemId = req.params.id;
  try {
    res.json(await item.allShulByItem(itemId));
  } catch (err) {
    console.error(`Error while getting allItems`, err.message);
    next(err);
  }
});
```

הפונקציה allShulByItem תתבצע והיא תביא את כל בתי הכנסת שצריכים את הפריט שהמשתמש בחר במקרה ולא יהיה בתי כנסת יחזר null.

הקומפוננטה תציג את בתי הכנסת שהגיעו מה fetch שלכל בית כנסת יש אפשרות ללחוץ לתרומה, עם אפשרות לחיפוש בית כנסת בלחיצה על תרומה אם המשתמש לא התחבר נעביר אותו לדף login ולאחר מכן נעביר אותו לדף confirmation במידה והוא התחבר נעביר אותו ל confirmation בשתי המקרים נשלח ב props את ה פריט כמות לתרומה ומחיר.

:DonationsDonated -component

מציגה לתורם את התרומות שהוא תרם, תרומות של פריטים ותרומות חופשיות ע"י שתי ה- fetch שנכתוב ב useeffect:

```
url =
`http://localhost:6200/api/user/${sessionStorage.getItem('email')}/item
Donation`;
let response2 = await fetch(url);
let url =
`http://localhost:6200/api/user/${sessionStorage.getItem('email')}/free
Donation`;
let response1 = await fetch(url);
```

:FreeDonation -component

קומפוננטה של תרומה חופשית, במידה ותורם רוצה לתרום סכום מסוים של כסף התורם יכניס את הפרטים – בית כנסת שאליו הוא רוצה לתרום כמובן שהוא יצטרך לבחור בית כנסת רק מהרשימה שנספק לו שזה הבתי כנסיות שנמצאים בdataBase, פירטי אשראי, סכום, בלחיצה על כפתור האישור נבצע את ה fetch הבא:

```
const url = `http://localhost:6200/api/donation/save`;
let response = await fetch(url, {
  method: 'POST',
  headers: {
    'Accept': 'application/json',
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(freeDonation)
});
```

שנשלח איתו את פרטי התרומה, ב server תתבצע השורה:

```
app.use('/api/donation', donation);
```

בקובץ donation תתבצע הפונקציה:

```
router.post('/save', async function (req, res, next) {
  try {
    res.json(await donation.saveDonation(req.body));
  } catch (err) {
    console.error(`Error`, err.message);
    next(err);
  }
});
```

בפונקציה saveDonation(req.body) נכניס את פירטי התרומה לטבלת
.freeDonation

:Gabay-component

מציגה את הניווט של הגבאי עם האופציות הוסף פריט, הצג בקשות לתרומה,
הצג את כל התרומות, הקדשות, יציאה.

:Login -component

קומפוננטה של התחברות של משתמש רשום בלחיצה על התחבר כמשתמש
חדש נעביר את המשתמש לדף signUp, בלחיצה על כפתור המשך נבצע את ה
fetch רק במידה והמשתמש מילא את הפרטים שלו והם תקינים:

```
const url = `http://localhost:6200/api/user/login`;
let response = await fetch(url, {
  method: 'POST',
  headers: {
    'Accept': 'application/json',
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(user)
});
```

ונשלח ב body את המייל והסיסמא שהמשתמש הזין.

ב server לפי השורה הזאת:

```
app.use('/api/user', user);
```

תתבצע הפונקציה:

```
router.post('/login', async function (req, res, next) {
  try {
    res.json(await user.whatType(req.body));
  } catch (err) {
    console.error(`Error while getting users`, err.message);
    next(err);
  }
}
```

```
});
```

שנמצאת בתוך הקובץ user, בפונקציה whatType(req.body) נבדוק אם המשתמש קיים אם הוא לא קיים נחזיר "לא קיים" ונעביר אותו לדף signUp, ואם הוא קיים נבדוק מה הסטטוס שלו – גבאי או תורם או גם וגם ולפי זה ננווט אותו לדף המתאים, אם הוא גם גבאי וגם תורם הוא יבחר המשתמש יבחר באיזה סטטוס הוא רוצה להכנס.

:SignUp-component

קומפוננטה של התחברות של משתמש חדש בלחיצה על התחבר כמשתמש קיים נעביר את המשתמש לדף login, בלחיצה על כפתור המשך נבצע את ה fetch רק במידה והמשתמש מילא את הפרטים שלו והם תקינים:

```
const url = "http://localhost:6200/api/user/add";
let response = await fetch(url, {
  method: 'POST',
  headers: {
    'Accept': 'application/json',
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(user)
});
```

ונשלח את פרטי המשתמש החדש, ב server לפי השורה:

```
app.use('/api/user', user);
```

תתבצע הפונקציה:

```
router.post('/add', async function(req, res, next) {
  try {
    res.json(await user.createUser(req.body));
  } catch (err) {
    console.error(`Error`, err.message);
    next(err);
  }
});
```

הפונקציה createUser(req.body) תבדוק שהכן המשתמש לא קיים במידה וכן היא לא תוסיף אותו ותוצג הודאה מתאימה, במידה ולא היא תוסיף אותו למאגר המשתמשים אם הוא נכנס בסטטוס גבאי המשתמש ימלא את פירטי בית הכנסת שלו ובלחיצה על אישור נבצע את ה fetch:

```
const response = await fetch('http://localhost:6200/api/shul/add', {
  method: 'POST',
  body: formData,
});
```

ונשלח ב body את פירטי בית הכנסת, ב server לפי השורה:

```
app.use('/api/shul', shul);
```

תתבצע הפונקציה:

```
router.post('/add', shul.upload.single('file'), async function (req, res) {
  try {
    res.json(await shul.createShul(req.body))
  }
  catch (err) {
    console.error(`Error the shul not create`, err.message);
  }
});
```

הפונקציה createShul(req.body) תוסיף את בית הכנסת למאגר בתי הכנסיות.

:NavDonate -component

מציגה את הניווט של התורם עם האופציות תרומה לפי פריט, תרומה לפי בית כנסת, תרומה חופשית, התרומות שלך, יציאה.

:NotDonations -component

מציגה לגבאי את התרומות שעדיין לא נתרמו בפורמט: נתרם X מתוך Y ע"י שב useeffect נבצע fetch :

```
const url = `http://localhost:6200/api/item/notDonations/${props.ShulId}`;
```

ב props נשלח את ה id של הבית כנסת ב server נשלח אף הפריטים שעדיין לא נתרמו לבית הכנסת לפי ה id ששלח.

מציגים את הפריטים ע"י הקומפוננטה Item שהיא מציגה פריט ויש לה שתי כפתורים של מחק ועדכן, בלחיצה על מחק נמחק את הפריט מהרשימה שנמצאת בצד ה client ונבצע fetch:

```
const url = `http://localhost:6200/api/item/delete`;
await fetch(url, {
  method: 'POST',
  headers: {
    'Accept': 'application/json',
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(data)
});
```

ב data נשלח את ה id והכמות שבמקרה הזה יהיה 0, ב server לפי השורה:

```
app.use('/api/item', item);
```

תתבצע הפונקציה:

```
router.post('/delete', async function (req, res, next) {
  try {
```

```

    console.log(req.body);
    res.json(await item.deleteItem(req.body));
  } catch (err) {
    console.error(`Error`, err.message);
    next(err);
  }
});

```

הפונקציה `deleteItem(req.body)` תציב 0 בכמות של הפריטים שעדיין לא נתרמו (לא מחקתי לגמרי כיוון שיכול להיות שבר תורם תרם חלק מהכמות שהגבאי צריך)

בלחיצה על כתור עידכון נעביר את המשתמש לקומפוננטה `UpdateItem`, ונמלא את הנתונים של הפריט והגבאי ישנה לפי רצונו, בלחיצה על אישור נבצע את ה `:fetch`

```

const url = `http://localhost:6200/api/item/update`;
let response = await fetch(url, {
  method: 'PUT',
  headers: {
    'Accept': 'application/json',
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(item)
});

```

ב `data` נשלח את ה נתונים החדשים של הפריט, ב `server` לפי השורה:

```
app.use('/api/item', item);
```

תתבצע הפונקציה:

```

router.put('/update', async function (req, res, next) {
  try {
    res.json(await item.updateItem(req.body));
  } catch (err) {
    console.error(`Error`, err.message);
    next(err);
  }
});

```

הפונקציה `updateItem(req.body)` תבדוק אם השם של הפריט קיים במאגר אם לא היא תכניס אותו, ותעדכן את השורה ב `database`.

מסכים

מסך ה – log in:

התחברות

תרומה לפי פריט

תרומה חופשית

התרומות שלך

התחבר כמשתמש קיים

שם משתמש

email

הכנס סיסמה

password

המשך

משתמש חדש

מסך ה – sign up:

התחברות

תרומה לפי פריט

תרומה חופשית

התרומות שלך

משתמש חדש

הכנס שם

FirstName

הכנס שם משפחה

LastName

טלפון

phone

מדינה

Name

סוג משתמש

▼

תורם

סיסמה

password

שמור

התחבר כמשתמש קיים

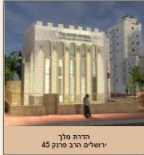
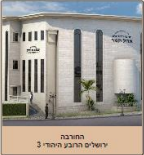
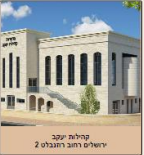
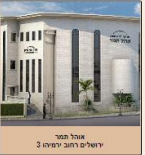
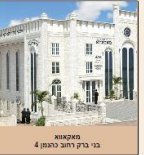
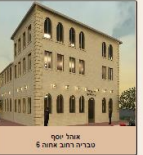

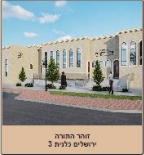
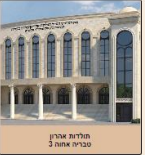

מסכים של התורם:

תרומה לפי בית כנסת:

shifrac2002@gmail.com - יציאה מ- | התרומות שלך | תרומה חופשית | תרומה לפי פריט | תרומה לפי בית כנסת

בחר את בית הכנסת אליה אתה רוצה לתרום

הכנס שם פריט לחיפוש:

 הרדף חור ירושלים הרב חנקל 45	 המזבנה ירושלים הרבג הכהן 3	 קולת עיני ירושלים הרב דובלעט 2	 אולד העיר ירושלים הרב יעקב 3	 המקדש בני ברק הרב נתן 4	 אולד העיר טבריה הרב אהרן 6	 שני דווי ירושלים הרב אלישבע 12
 זוהר העיר ירושלים לילי 3	 תורת ארץ טבריה אהרן 3	 בעלז ירושלים שמואל לביש 22				

הגד עד בית כנסת

בלחיצה על בית כנסת מסוים יוצגו הפריטים שהוא צריך:

shifrac2002@gmail.com - יציאה מ- | התרומות שלך | תרומה חופשית | תרומה לפי פריט | תרומה לפי בית כנסת

בחר פריט שהינך רוצה לתרום לבית הכנסת שפתי חיים

הכנס שם פריט לחיפוש:

<p>חומש במדבר</p> <p>סך לתרומה עבור פריט יחיד: 55 סך הכל חסרים בבית הכנסת: 33 כמות שהנך רוצה לתרום: <input type="text"/></p> <p>תרמו עכשיו!</p>	<p>מחזור לראש השנה</p> <p>סך לתרומה עבור פריט יחיד: 50 סך הכל חסרים בבית הכנסת: 29 כמות שהנך רוצה לתרום: <input type="text"/></p> <p>תרמו עכשיו!</p>	<p>ארון קודש</p> <p>סך לתרומה עבור פריט יחיד: 123 סך הכל חסרים בבית הכנסת: 123 כמות שהנך רוצה לתרום: <input type="text"/></p> <p>תרמו עכשיו!</p>
---	--	--

חזור

בלחיצה על פריט הרצוי:

avi@gmail.com - יציאה מ- | התרומות שלך | תרומה חופשית | תרומה לפי פריט | תרומה לפי בית כנסת

סך לתרומה - 120

הכנס הקדשה

הכנס פרטי אשראי

הכנס מספר אשראי

הכנס תוקף

הכנס שלוש ספרות בגב הכרטיס

תרומה לפי פריט:

החנות שלך תחנת השעות תחנה לוי פריס תחנה לוי בית שמש

תבחיך את המיפוש אותו אתה רוצה לתקוע

שולחן

ספרים

ארון קודש

סידורים

כרזות

לוח ד'תשפ"ו

בדורות

מנורה

חלון

מטבחון

הנג עד פרטים

בלחיצה על פריט מסוים:

[illegible]

בלחיצה על בית כנסת הרצוי:

תרום לפי כנסת

תרום לפי פריט

תרום חופשית

תרומות שלך

יציאה מ- shifrac2002@gmail.com

סך לתרומה - 111111

הכנס הקדשה

הכנס פרטי אשראי

הכנס מספר אשראי

הכנס תוקף

הכנס שלוש ספרות בגב הכרטיס

בלחיצה על תרומה חופשית:

תרומה לפי בית כנסת	תרומה לפי פריט	תרומה חופשית	התרומות שלך	יציאה מ- shifrac2002@gmail.com
--------------------	----------------	--------------	-------------	--------------------------------

תרומה חופשית

בחר בית כנסת

שפת חיים

הכנס הקדשה

הכנס סכום לתרומה

הכנס פרטי אשראי

הכנס מספר אשראי

הכנס תוקף

הכנס שלוש ספרות בגב הכרטיס

בלחיצה על התרומות שלך:

תרומה לפי בית כנסת	תרומה לפי פריט	תרומה חופשית	התרומות שלך	יציאה מ- shifrac2002@gmail.com
--------------------	----------------	--------------	-------------	--------------------------------

האזור האישי שלך

התרומות החופשיות שלך:

סכום תרומה: 10000
הקדשה: לעילוי נשמת פלוני בן פלוני
לבית כנסת: שפתי חיים

תרומות הפריטים שלך:

סידורים הקדשה: לבית כנסת: אוהל יוסף מחיר לבווד: 120 כמות: 3	נברשת הקדשה: לבית כנסת: שפתי חיים מחיר לבווד: 234 כמות: 2	סידורים הקדשה: לעילוי נשמת פלוני בן פלוני לבית כנסת: שפתי חיים מחיר לבווד: 123 כמות: 3	לוח דיגיטלי הקדשה: לעילוי נשמת פלוני בן פלוני לבית כנסת: מאקאווא מחיר לבווד: 1111111 כמות: 4	מחזור לראש השנה הקדשה: לעילוי נשמת פלוני בן פלוני לבית כנסת: שפתי חיים מחיר לבווד: 50 כמות: 1
--	--	---	---	--

מסכים של הגבאי:

בלחיצה על הוספת פריט:

הוסף פריט

הצג בקשות לתרומה

הצג את כל התרומות

הקדשות

יציאה מ- shifrac2002@gmail.com

הוספת פריט

שם פריט

כמות

סכום תרומה עבור פריט בודד

הוספת תמונה

☐ בחירת קובץ ☐ לא נבחר קובץ

הוסף

בלחיצה על הצג בקשות לתרומה:

הוסף פריט

הצג בקשות לתרומה

הצג את כל התרומות

הקדשות

יציאה מ- shifrac2002@gmail.com

הבקשות שעדיין לא נתרמו

נברשת

מחיר לתרומה 10000
מספר פריטים שעדיין לא נתרמו 4 מתוך 4

עדין **מחק**

מטבח

מחיר לתרומה 15000
מספר פריטים שעדיין לא נתרמו 1 מתוך 1

עדין **מחק**

ארון קודש

מחיר לתרומה 17850
מספר פריטים שעדיין לא נתרמו 1 מתוך 1

עדין **מחק**

ספר תורה

מחיר לתרומה 26258
מספר פריטים שעדיין לא נתרמו 3 מתוך 3

עדין **מחק**

פרוכת

מחיר לתרומה 10000
מספר פריטים שעדיין לא נתרמו 2 מתוך 2

עדין **מחק**

שולחנות

מחיר לתרומה 100
מספר פריטים שעדיין לא נתרמו 20 מתוך 20

עדין **מחק**

בלחיצה על הצג את כל התרומות:

הוסף פריט

הצג בקשות לתרומה

הצג את כל התרומות

הקדשות

יציאה מ- qq@gmail.com

התרומות שנתרמו בבית הכנסת שלך

ארון קודש

סכום לתרומה: 100
נתרמו 7 מתוך 7

סידורים

סכום לתרומה: 120
נתרמו 3 מתוך 12

לוח דיגיטלי

סכום לתרומה: 132
נתרמו 2 מתוך 3

ארון קודש

סכום לתרומה: 1512
נתרמו 0 מתוך 1

התרומות החופשיות שלך - 540 ש"ח

28

בלחיצה על הקדשות:

הוסף פריט	הצג בקשות לתרומה	הצג את כל התרומות	הקדשות	יציאה מ- haim@gmail.com
-----------	------------------	-------------------	--------	-------------------------

הקדשות פריטים

הקדשה: לעילוי נשמת פלוני בן פלוני
שם ומשפחה: מונסונגו אליהו
שם פריט: לוח דיגיטלי

הקדשה טופלה

הקדשות תרומות חופשיות

הקדשה: לעילוי נשמת פלוני בן פלוני
שם ומשפחה: כהן חיים
סכום תרומה: 3544

הקדשה טופלה

הקדשה: לעילוי נשמת פלוני בן פלוני
שם ומשפחה: רזונבלט אריה
סכום תרומה: 343

הקדשה טופלה

הקדשה: לעילוי נשמת פלוני בן פלוני
שם ומשפחה: כהן תמר
סכום תרומה: 10000

הקדשה טופלה

מסקנות

כשניגשתי לתכנן את פרויקט הגמר ראינו כי המשימה מורכבת מאד ודורשת השקעה רבה הן ובעיקר בתכנון והן בכתיבה עצמה.

ראשית השקעתי שעות רבות בבניית בסיס הנתונים ובחשיבה על מבנה הפרויקט. דבר זה לכשעצמו היה פרויקט די גדול מחשוב המערכת למסודרת ערך זמן רב וצרך מחשבה רבה. לאחר מכן ניגשתי לבניית הפרויקט, תחילה תכנתי אותו על דף: תכנתי מחלקות, מסכים ורכיבים. לאחר מכן פניי לבנותו על המחשב.

מחלק זה למדתי רבות כיצד לגלות טעויות שקוטעות את הרצת הפרויקט ותוקעות את המשך בנייתו, כיצד להשתמש בפונקציות רבות ומגוונות. לאחר מעשה אני יכולה לומר כי השעות שהשקעתי בבניה וביסוד לא היו מיותרות כלל והן לא היו לריק, הגענו למסקנה כי פרויקט שאינו בנוי בצורה נכונה וכי הבסיס שלו אינו יעיל וטוב מועד לכישלון. והשקעה בדברים אלו הינה השקעה לטווח ארוך.

הפרויקט תרם לי רבות ולימד אותי איך לחשוב על מבנה של פרויקט, לבצע ניתוח מערכת, לעמוד בהספקים ולהתנסות בלמידה עצמית.

לאחר שעות רבות מספור של עמל אני חשה סיפוק רב מבנית מערכת וכן אני חשה שהפקתי תועלת רבה מפרויקט הגמר ומקווה שיסיע לי בעתיד ויעיל בעז"ה.

סיכום

האתר נכתב במשך החודשים האחרונים בעבודה מאומצת ובהשקעה מרובה של תכנון עמל והשקעה. תוך שמירה על רמת תכנות גבוהה.

בהגיעי למועד זה של סיום הפרויקט ובמבט אחורה על הדרך אותה עברתי ניתן לומר כי קיבלתי הבנה מעמיקה ביסודות התכנות. הידע כיצד ללמוד להשתמש בכלים שלמדתי במהלך השנה וכן כלים רבים שנלמדו במהלך הפרויקט והרצון לפתח תוכנה ברמה גבוהה. ניתן לומר שהפרויקט קידם אותי באופן ניכר מבחינה מקצועית ואישית.

פרויקט זה אכן היווה אבן דרך משמעותית בהתמקצעות בתחום הנדסת התכנה.

בסיכומי של פרויקט אין לי אלא להודות על הכלים הרבים שרכשנו במהלך העבודה על הפרויקט.

ביבליוגרפיה

- [w3Schools.com](https://www.w3schools.com)
- [CAP.co.il](https://www.captainstack.com)
- [postman.com](https://www.postman.com)
- [javatpoint.com](https://www.javatpoint.com)