

מכללת תו"ם חרב לאת

מלאכי שלומי

כיתה יד'

# Ramat Gan Library

Frontend Application by Malachi Shlomi

Documentation and Summery



Source Code on GitHub:

<https://github.com/malachi-shlomi/library>

Live Demo online:

<https://malachishlomi.com/library>



### **The main purpose of the website:**

אתר הספרייה העירונית רמת גן משמש בעיקר בתור אתר שמספר מעט על הספרייה העירונית, על מה יש בתוכה ונועד להביא לקוחות חדשים לבקר בספרייה. ומספק מידע ודרכים שונות ליצירת קשר עם הספרייה. בנוסף, האתר משמש את לקוחות הספרייה לחיפוש ספרים בספרייה וקריאת מאמרים קצרים של תוכן עניינים על כל ספר שנמצא בספרייה.

### **Tools used in the Project:**

- HTML - to build the structure of the website.
- CSS - to apply design.
- JavaScript - to develop the main functionality of the website.
- Google Photos - to find photos on the internet for the website.
- Wikipedia - for books research.

### **Pages on the website**

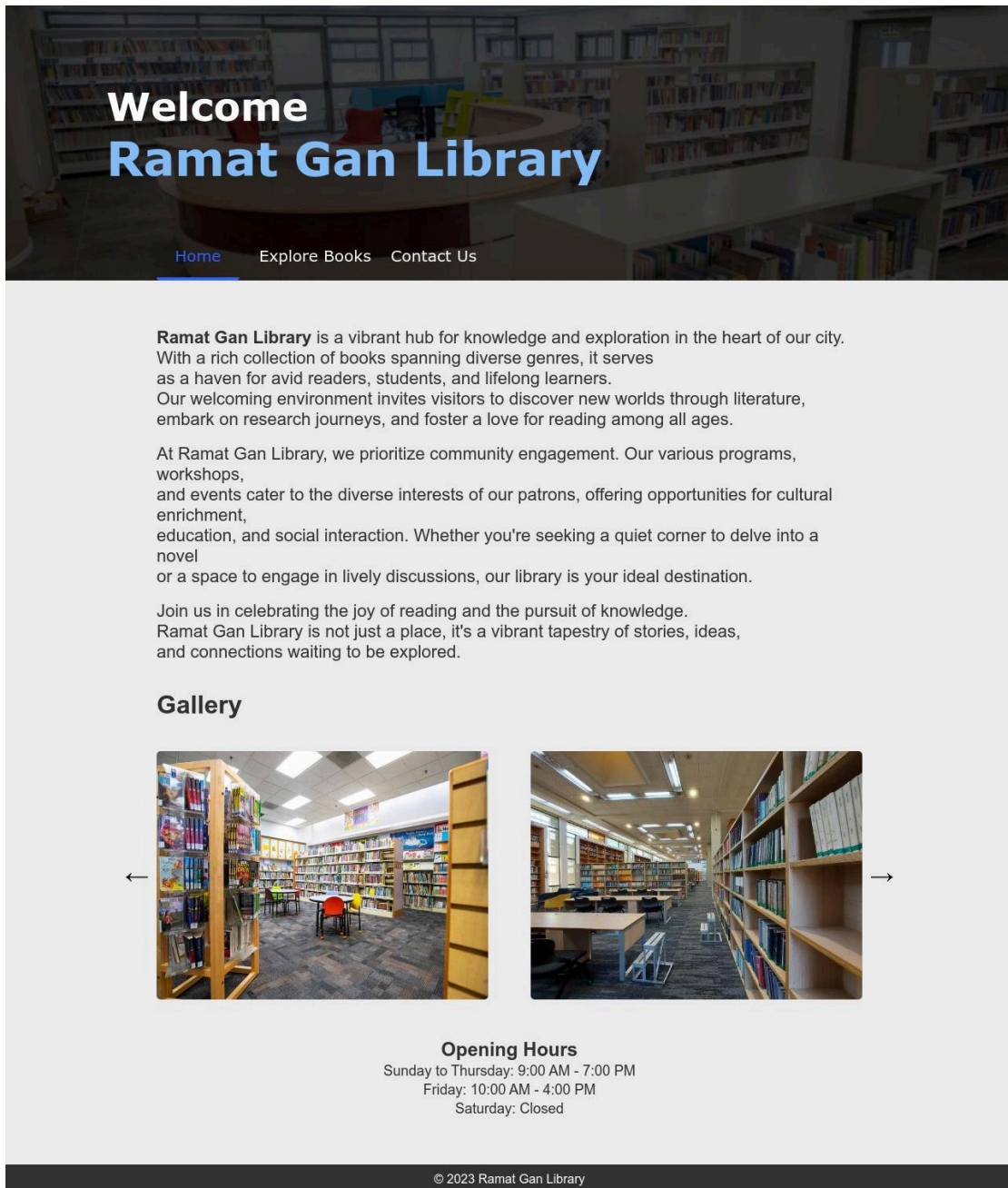
- עמוד הבית (Home Page - index.html)
- עמוד ספרים (Books Page - books.html)
- עמוד ספר (Book Page - book.html)
- עמוד צור קשר (Contact Page - contact.html)

# The Home Page

Source Code: <https://github.com/malachi-shlomi/Library/blob/main/index.html>

Demo: <https://malachishlomi.com/library/>

Screenshot:



בדף זה הוספתי תבנית קבועה של עיצוב ה-Header שמופיע בכל הדפים בתוספת כותבת גדולה ומזמינה.

בנוסף יופיעו navigation buttons אינטראקטיבים לניווט מהיר וקל באתר. לאחר מכן הוספתי ביוגרפיה (שלושה פסקאות) שמספרות על הספרייה, קרוסלת תמונות אינטראקטיבית (Gallery) להתרשמות מהספרייה, ושעות הפתיחה של הספרייה.

## הסבר ה-script שמשמש לנוויגציה בין התמונות בגלריה

```
1  const prevButton = document.getElementById('prev');
2  const nextButton = document.getElementById('next');
3  const pictureContainer = document.querySelector('.pictures-list');
4
5  let currentIndex = 0;
6
7  nextButton.addEventListener('click', () => {
8    if (currentIndex < pictureContainer.children.length - 2) {
9      currentIndex += 1;
10     pictureContainer.style.transform = `translateX(-${currentIndex * 53}%)`;
11   }
12 });
13
14 prevButton.addEventListener('click', () => {
15   if (currentIndex > 0) {
16     currentIndex -= 1;
17     pictureContainer.style.transform = `translateX(-${currentIndex * 53}%)`;
18   }
19 });
```

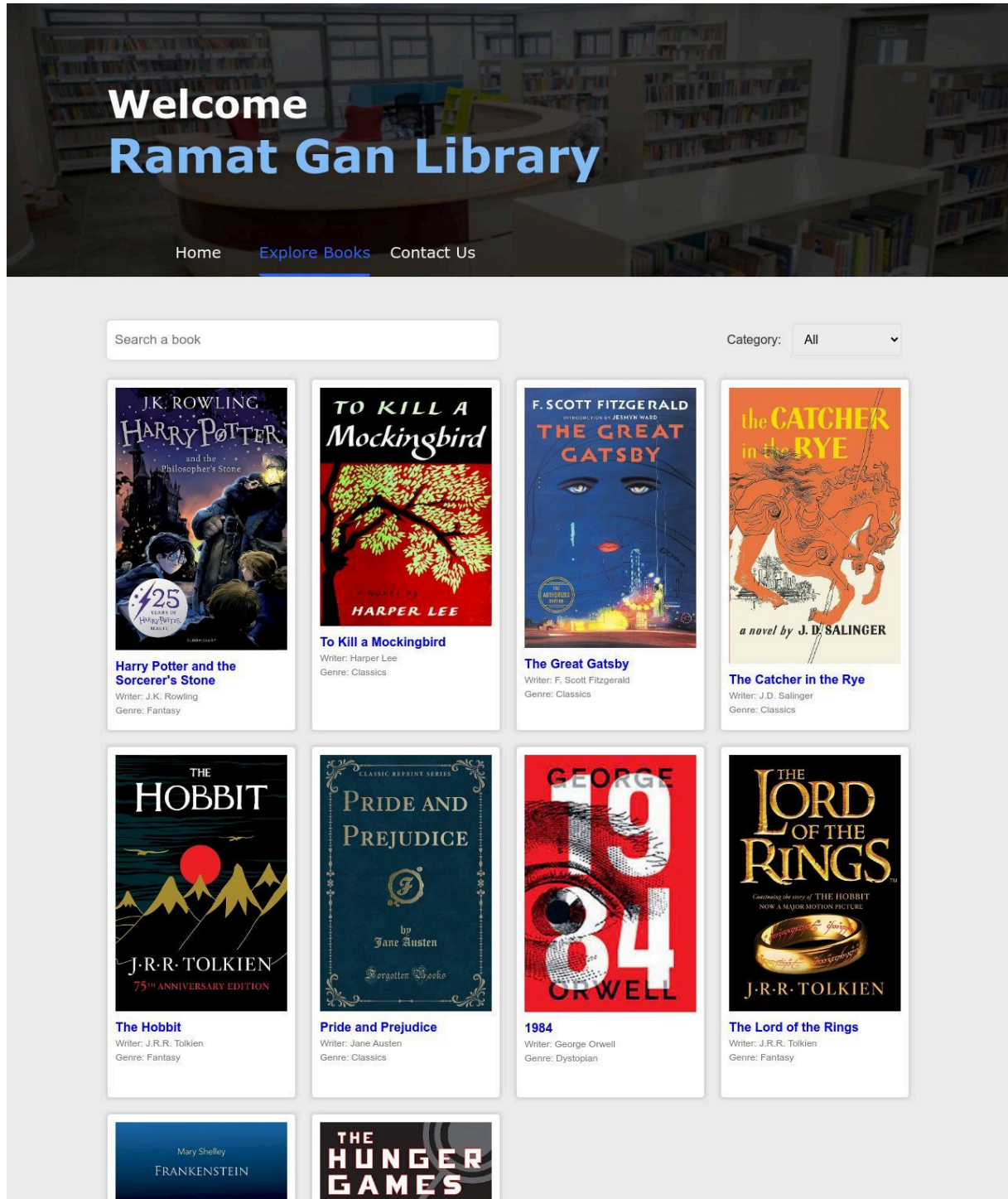
בחרתי את האלמנטים של הכפתורים וה-container של התמונות והכנסתי אותם למשתנים (prevButton, nextButton, pictureContainer), הצבתי משתנה נוסף בשם currentIndex שישמור את ה-index של התמונה הנוכחית. הוספתי Event listener שיעקוב אחר לחיצות על כל אחד הכפתורים - ובעט לחיצה ישנה את אינדקס התמונה במידה ויש עוד תמונות לצפות בהן.

# Books Page

Source Code: <https://github.com/malachi-shlomi/Library/blob/main/books.html>

Demo: <https://malachishlomi.com/library/books.html>

Screenshot:



בדף זה ניתן לחפש ולראות את כל הספרים הזמינים בספרייה.  
הוספתי שורת חיפוש לחיפוש מהיר של ספר על פי שם.

בנוסף הוספתי אפשרות מיון לפי קטגוריה של ספר למציאת ספרים מעניינים בז'אנר מסוים.  
ניתן ללחוץ על כל ספר כדי לראות מידע נוסף על הספר בדף.

### הסבר ה-script של הדף books

```
1  const books = [  
2    {  
3      bookid: 1001,  
4      name: "Harry Potter and the Sorcerer's Stone",  
5      writer: "J.K. Rowling",  
6      genre: "Fantasy",  
7      image: "./Books/harry-potter.jpeg",  
8    },  
9    {  
10     bookid: 1002,  
11     name: "To Kill a Mockingbird",  
12     writer: "Harper Lee",  
13     genre: "Classics",  
14     image: "./Books/mockingbird.jpg",  
15   },  
16   {  
17     bookid: 1003,  
18     name: "The Great Gatsby",  
19     writer: "F. Scott Fitzgerald",  
20     genre: "Classics",  
21     image: "./Books/gatsby.jpeg",  
22   },  
23   {  
24     bookid: 1004,  
25     name: "The Catcher in the Rye",  
26     writer: "J.D. Salinger",  
27     genre: "Classics",  
28     image: "./Books/catcher.jpeg",  
29   },  
]
```

בתחילה יצרתי משתנה קבוע בשם books שישמור רשימה של objects מסוג ספר.  
לכל ספר יש מספר תכונות בסיסיות כמו מספר מזהה, שם ספר, כותב הספר, ז'אנר ולינק לתמונת הכריכה,  
מה שיאפשר פונקציונליות רינדור פשוטה יותר ויכולת לערוך מידע על ספרים בקלות או למשוך אותם מ-API.

```

74   const bookContainer = document.getElementById("BooksSection");
75   const categorySelector = document.getElementById("category-selector");
76
77   function ShowBooks() {
78
79       const selectedCategory = categorySelector.value;
80
81       bookContainer.innerHTML = "";
82
83       const filteredBooks = (selectedCategory === "all")
84           ? books
85           : books.filter(book => book.genre === selectedCategory);
86
87       const searchInput = document.getElementById("searchInput").value.toLowerCase();
88
89       const Books = filteredBooks.filter(book => {
90           return book.name.toLowerCase().includes(searchInput);
91       });
92
93       Books.forEach(book => {
94           const bookElement = document.createElement("div");
95           bookElement.className = "book";
96
97           const bookPageURL = `book.html?bookid=${book.bookid}`;
98
99           bookElement.innerHTML = `
100               <a style="text-decoration: none" href="${bookPageURL}">
101                   
102                   <h2>${book.name}</h2>
103                   <p>Writer: ${book.writer}</p>
104                   <p>Genre: ${book.genre}</p>
105               </a>
106           `;
107           bookContainer.appendChild(bookElement);
108       });
109   }
110
111
112
113   categorySelector.addEventListener("change", ShowBooks);
114   ShowBooks();

```

השלב הבא היה לרנדור את רשימת הספרים אל הדף. ובנוסף, להוסיף את הפונקציונליות של ה-Search Bar וה-Category Selector.

על מנת לבצע זאת, יצרתי פונקציה בשם ShowBooks. בשלב הראשון, הפונקציה תיקח את הערך של הקטגוריה שנבחרה על ידי הערך value של הסלקטור, ואז תיצור משתנה חדש בשם filteredBooks שיבצע סינון באמצעות הפונקציה filter וישמור בתוכו אך ורק את הספרים שערך ה-genre שלהם שווה לז'אנר הנבחר. לאחר מכן, היא תעשה אותו הדבר ותיקח את הערך שהוזן בשורת החיפוש, תסנן את הספרים כדי לקבל רק את הספרים שבשם שלהם מופיע חלק מהערך בשורת החיפוש, ואז תיצור משתנה חדש בשם Books, שיכיל את כל הספרים לאחר המיון ולאחר הסינון.

בשלב האחרון, הפונקציה תעבור בלולאה על רשימת הספרים (Books) באמצעות הפונקציה forEach, ולכל אחד מהספרים - היא תיצור אובייקט div על ידי createElement, תתן לו class ואז תעצב את הכרטיס שלו באמצעות innerHTML, ותרנדור אותו לדף לתוך עמוד הספרים באמצעות appendChild. בנוסף, תוסיף לכל אובייקט ספר קישור לדף קריאת התיאור ופרטים נוספים על הספר.

לאחר שסיימנו עם הפונקציה, רק נשאר לקרוא לה פעם אחת כדי שתרנדור את כל הספרים, ולהוסיף Event Listener שיפעיל את הפונקציה מחדש בכל פעם שהמשתמש בוחר קטגוריה אחרת.

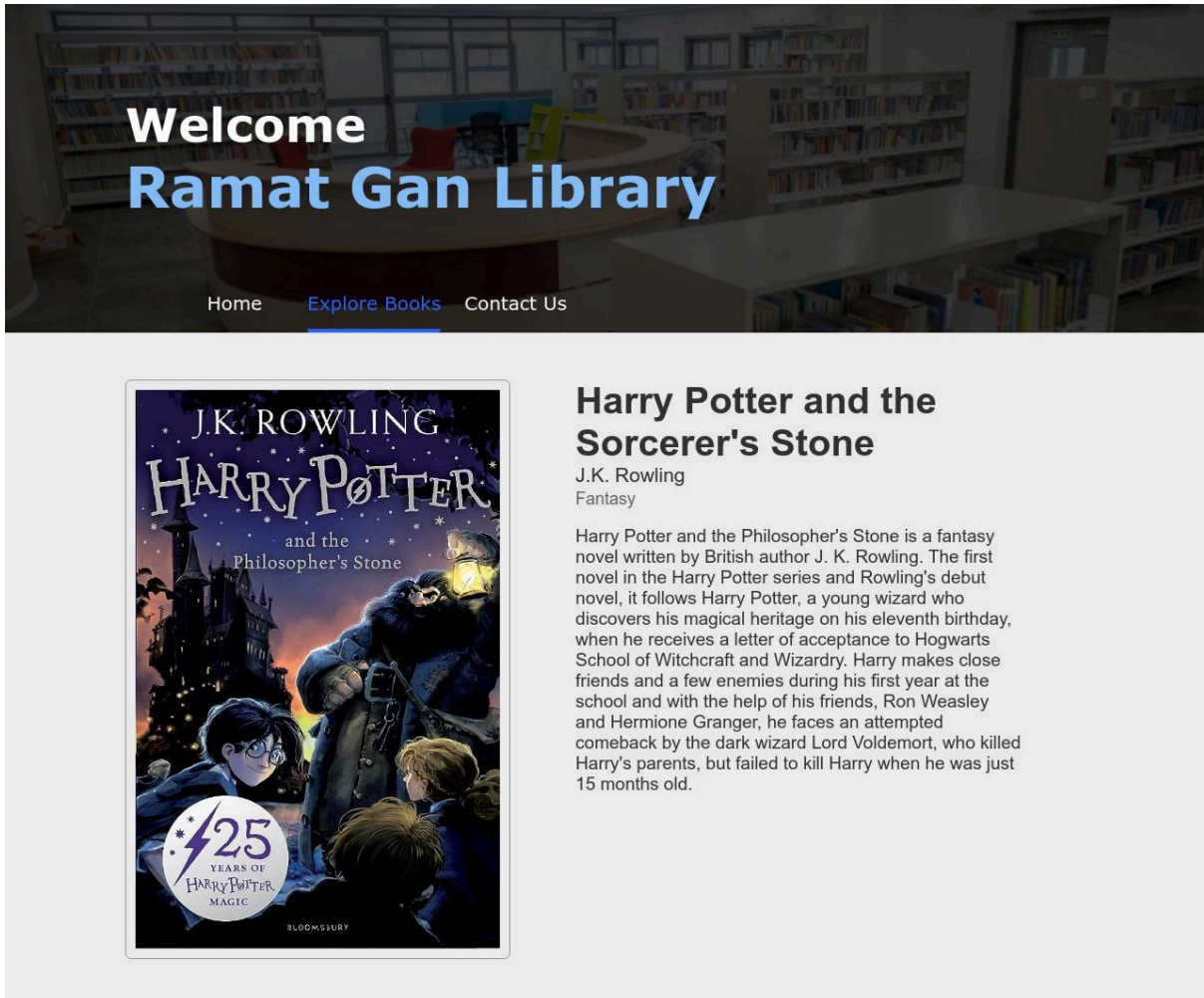


## Book Details Page

Source Code: <https://github.com/malachi-shlomi/Library/blob/main/js/book.js>

Demo: <https://malachishlomi.com/library/book.html?bookid=1001>

Screenshot:



אל דף זה ניתן להכנס בקלות על ידי לחיצה על אחד הספרים בדף Books.  
אל כל אובייקט פר בדף Books, משויך link עם href שמוביל ל-URL הזה:  
[https://malachishlomi.com/library/book.html?bookid=\[book-id\]](https://malachishlomi.com/library/book.html?bookid=[book-id])  
כל שהמשתנה book-id משתנה מספר לספר תלוי באיזה ספר בחרנו.  
הדף מראה תמונה גדולה של הספר, את שם הספר, כותב הספר, ותיאור ביוגרפיה קצר על הספר.



## הספר ה-script של הדף Book

```
1  const urlParams = new URLSearchParams(window.location.search);
2  const bookId = urlParams.get("bookid") || "";
```

בתחילה ניקח מהפרמטרים של החיפוש ב-URL את ה-bookId, בעזרת window.location.search, כדי לדעת איזה פרטים של איזה ספר להציג על המסך, דוגמא לערך של bookId: "1001"

```
90  const bookDetails = books.find(item => item.bookid == bookId);
```

לאחר מכן נמצא את פרטי הספר על ידי הרצת המתודה "find" על רשימת books (אותה הרשימה מהדף books), ונכניס את פרטי הספר לתוך משתנה בשם bookDetails.

```
92  if (bookDetails) {
93      const bookPageElement = document.querySelector('.BookPageWrapper');
94      bookPageElement.innerHTML = `
95          
96          <div>
97              <h2>${bookDetails.name}</h2>
98              <p> ${bookDetails.writer}<br><span>${bookDetails.genre}</span>
99              <br>
100             <p class="Description" >${bookDetails.description}<p>
101         </div>
102     `;
103 } else {
104     const bookPageElement = document.querySelector('.BookPageWrapper');
105     bookPageElement.innerHTML = `
106         <div>
107             <h2>Book Not Found!</h2>
108         </div>
109     `;
110 }
```

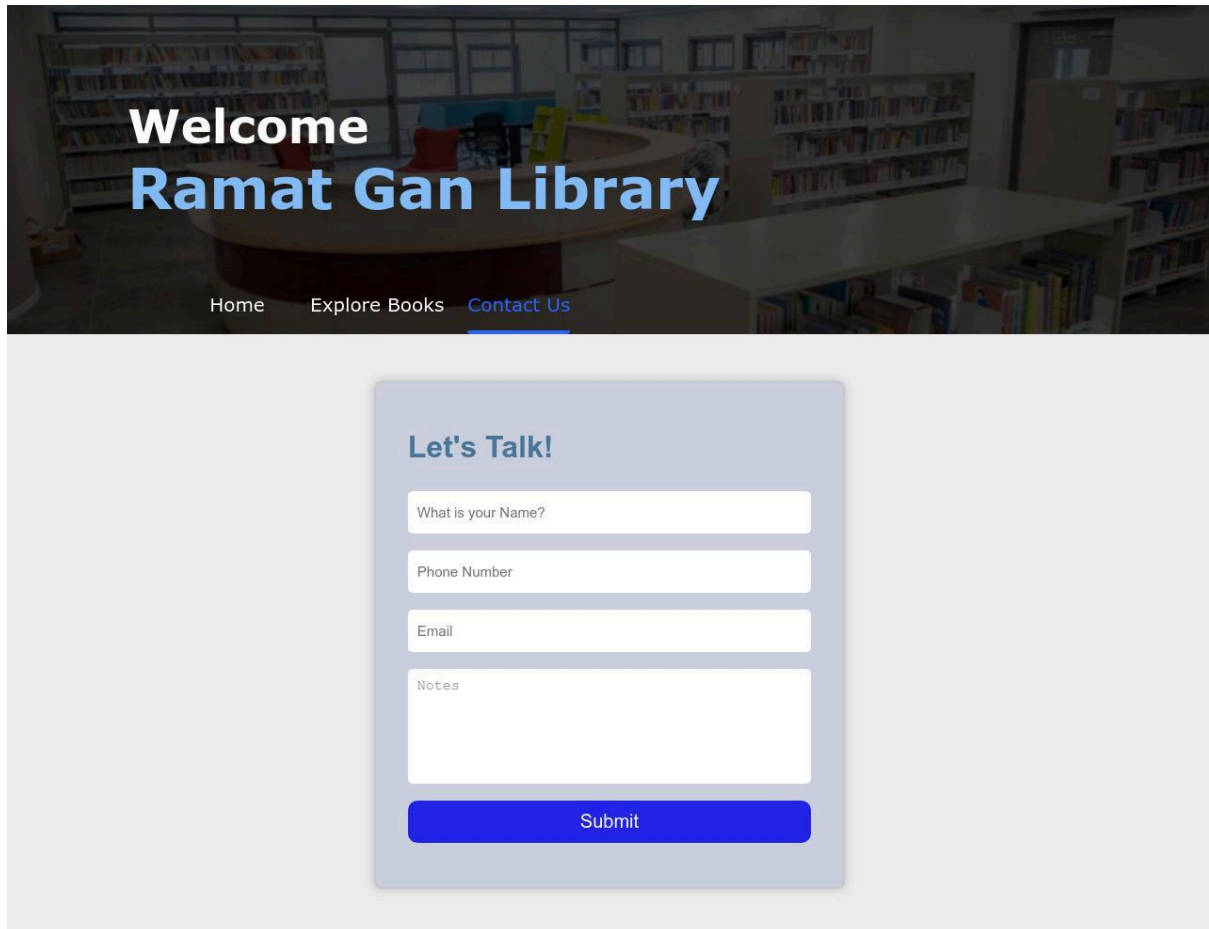
בשלב האחרון, נבדוק אם קיים ספר עם ה-bookId שמופיע ב-URL. במידה וכן, ניקח את ה-container שמופיע בו פרטי הספר, ונרנדר לתוכו את שם הספר, הביוגרפיה שלו, והתמונה על ידי innerHTML. במידה והספר לא קיים, נרנדר למסך הודעה שהספר לא קיים.

## Contact Page

Source Code: <https://github.com/malachi-shlomi/Library/blob/main/js/contact.js>

Demo: <https://malachishlomi.com/library/contact.html>

Screenshot:



**Welcome  
Ramat Gan Library**

[Home](#) [Explore Books](#) [Contact Us](#)

**Let's Talk!**

What is your Name?

Phone Number

Email

Notes

Submit

אל דף יצירת הקשר ניתן להכנס בקלות על ידי לחיצה על הכפתור "Contact Us" בכפתורי הנוויגציה. בדף ניתן לראות form ברור וקל לשימוש עם אופציה להוספת שם, מספר טלפון, אימייל והודעה לשליחה.

## Contact בדף Input Validation

```
2      const submitButton = document.querySelector('.submitButton');
3
4      submitButton.addEventListener('click', function(e) {
5          e.preventDefault();
6
7
8          const name = document.querySelector('input[placeholder="What is your Name?"]').value;
9          const phone = document.querySelector('input[placeholder="Phone Number"]').value;
10         const email = document.querySelector('input[placeholder="Email"]').value;
11
12
13         if (!name.trim()) {
14             alert('Please enter your name.');
```

```
15             return false;
16         }
17
18
19         const phonePattern = /^05\d{8}$/;
20         if (!phone.trim() || !phonePattern.test(phone)) {
21             alert('Please enter a valid phone number in the format 052XXXXXXX.');
```

```
22             return false;
23         }
24
25
26         const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
27         if (!email.trim() || !emailPattern.test(email)) {
28             alert('Please enter a valid email address.');
```

```
29             return false;
30         }
31
32
33         console.log('Form submitted successfully!');
34         alert('Thank you for your message!');
```

```
35
36
37     });
```

בחרתי את אלמנט כפתור ה-Submit והוספתי לו Event Listener.  
לאחר מכן בחרתי את שורות ה-Input באמצעות querySelector.

כדי לאמת שה-input של ה-name תקין - השתמשתי בתנאי if לבדוק אם הפונקציה trim על ה-input מחזירה undefined, במידה וכן - הסקריפט יציג alert עם הודעה מתאימה.

כדי לאמת שה-input של ה-phoneNumber תקין - השתמשתי ב-pattern של מספר טלפון, ובפונקציה test, כדי לבדוק אם מספר הטלפון מתחיל ב-"05" ואחריו 8 ספרות בדיוק. במידה והקלט לא תקין אציג alert בדפדפן עם הודעה מתאימה.

כדי לוודא שהמייל תקין, ביצעתי אותו הדבר רק עם pattern שונה, והשתמשתי בפונקציה test כדי לוודא, בנוסף השתמשתי במתודה trim כדי לבדוק אם הקלט לא קיים.

בסיום, במידה ואף אחד מהבדיקות לא נכשל והחזיר false - ניתן לבצע את ה-submission בהצלחה.

## רפלקציה על הפרויקט

הפרויקט אמנם היה קל אבל נתן לי רעיונות חדשים ליצירת אלמנטים אינטראקטיביים באתר כמו גלריית התמונות ועמוד הספרים. בנוסף היה נחמד לחשוב על עיצוב חדש שיראה טוב לאתר שימשוך את תשומת הלב של המבקרים ולעצב אותו ב-css. בפרויקט השתדלתי לפתח ולשפר מיומנויות בפיתוח כמו למשל עיצוב רספונסיבי שיתאים גם ל-mobile, ושימוש בgit למעקב אחר timeline של הגרסאות.