

## Software Documentation – GlassBook:

מגישים : אביתר אלטמן

שקד כהן

דניאל מאיר קרל

### **תוכן עניינים**

2	.....:data
5	.....:סכמה
8	.....:שאלות
16	.....:תיעוד הקוד

## :data

את המידע שלנו אפשר לחלק לשני חלקים.

חלק ראשון סופרים וספרים, חלק שני תגובות ומשתמשים.

החלק הראשון הוא החלק הארי של הפרויקט.

את הנתונים לחלק זה לקחנו מאתר Kaggle בקישור הבא:

<https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset?select=Books.csv>

כפי שניתן לראות לכל ספר יש את שמו, שם הכותב שלו, שם המפרסם, תמונה של הספר ותאריך הוצאה.

השתמשנו בכל המידע הנ"ל חוץ מתאריך ההוצאה (לא מצאנו לו צורך).

מידע נוסף שרצינו לכל ספר הוא קטגוריה, מידע זה היה חסר ולכן הוספנו אותו בצורה רנדומאלית בעזרת סקריפט שנדבר עליו בהמשך.

בנוסף בעזרת סקריפט פיצלנו את הקובץ Books.csv שהורדנו מהאתר לשני קבצי טקסט, new\_books.txt ו-new\_authors.txt בכדי שנוכל לייצר מכל אחד מהם טבלה, בתהליך הפיצול גם מחקנו ספרים ששמם אינו באותיות לטיניות/אנגליות בשביל שהמידע באתר יהיה יותר נגיש.

בסופו של התהליך נשארו עם כ-200,000 ספרים וכ-100,000 סופרים.

בחלק השני, את המשתמשים יצרנו בעזרת שני אתרים, אחד לג'נרט שמות משתמשים ופרטים והשני לג'נרט סיסמאות לאותם משתמשים.

<https://www.lastpass.com/features/password-generator#generatorTool>

<https://www.coolgenerator.com/username-generator>

בסופו של דבר ייצאנו את כולם לקובץ Users.csv עם כ-100 משתמשים שאותו העלנו לשרת.

את התגובות גם כן יצרנו בצורה רנדומאלית וקיישנו למשתמשים הקיימים בעזרת הסקריפט createComment.py.

כעת נפרט על תהליך עיבוד המידע שעשינו, כל קבצי המידע והסקריפטים השונים מצורפים למסמך זה.

## ספרים וסופרים:

כמו שהזכרנו לעיל, המידע נילקח מאתר Kaggle, את הקובץ, כמו שהוא, העברנו לסקריפט בשם `ord_data.py` בסקריפט פתחנו את הקובץ והפרדנו את המידע לשתי רשימות, רשימת ספרים ורשימת סופרים.

לכל סופר וספר נתנו ID ייחודי כך שכל ספר מחזיר במקום את שם הסופר שכתב אותו את ה-ID של אותו סופר.

על הדרך, הסרנו את כל הספרים ששמם אינו רשום באנגלית/לטינית כיוון שראינו שיש בעיה של קידוד ספרים אלו.

בסופו של דבר את שתי הרשימות שמרנו בקובצי טקסט, `new_books.txt` ו-`new_authors.txt`, כך שכל שורה ברשימה כתובה כשורה בקובץ עם הפרדה של פסיק בין נתון לנתון.

לאחר מכן השתמשנו בסקריפט `insert_data.py` בכדי להכניס את המידע מקבצי הטקסט שייצרנו לשרת ה-SQL.

בסקריפט אנו קודם יוצרים חיבור לשרת ה-SQL עם פרטי משתמש ה-root ולאחר מכן בעזרת ספריית `mysql.connector` אנו מעלים את המידע מקבצי הטקסט לטבלאות המתאימות בשרת.

## משתמשים:

קובץ המשתמשים הינו קובץ קטן ולכן נערך והותאם ידנית.

את הקובץ העלנו לשרת ה-SQL דרך ממשק הייבוא בשרת ישירות מקובץ `Users.csv`.

## תגובות:

את התגובות ייצרנו בצורה רנדומאלית בעזרת סקריפט.

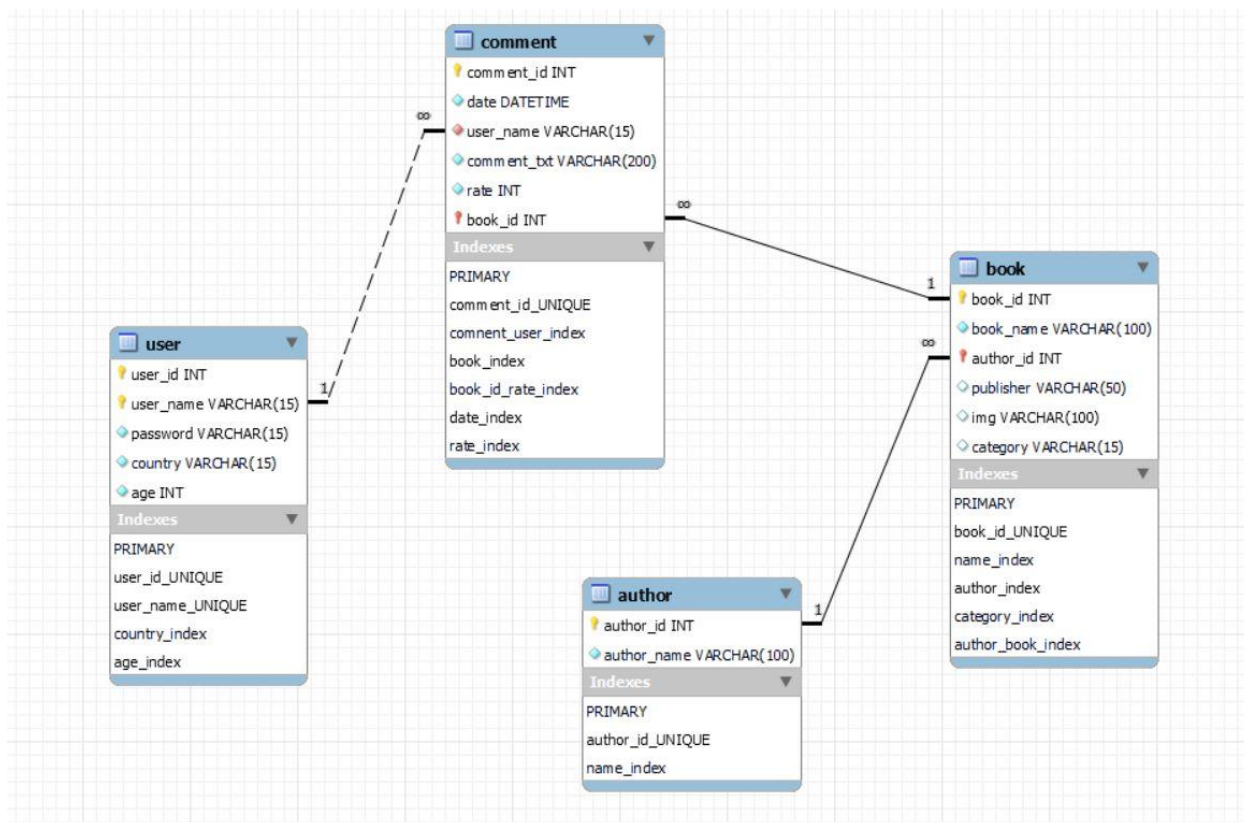
בסקריפט `createComment.py` כתבנו בצורה ידנית רשימה של כ-50 תגובות. לאחר מכן קראנו את הרשימות ה-ID של הספרים, התגובות ושמות משתמשים, מתוך קבצי הטקסט שלהם, לאחר מכן נגריל תאריך רנדומאלי (שנה, חודש, יום) נגריל מספר מזהה של ספר, נגריל תגובה, נגריל שם משתמש, נותן ID לתגובה וכותב בקובץ חדש של תגובות כך שכל ערך מופרד בפסיק. הקובץ של התגובות מגיע כבר כשלכל תגובה יש גם את הציון שלה מ-1 עד 5, מופרדים כבר בפסיק.

בסופו של דבר ייצאנו לקובץ טקסט בשם `comments.txt` כ-400,000 תגובות.

בעזרת הסקריפט `insert_comment.py` בצורה דומה לספרים והסופרים העלנו את המידע לטבלה המתאימה בשרת ה-SQL.



## סכמה:



הסבר על הסכמה:

לסכמה יש ארבע טבלאות ראשיות ושלוש foreign keys. נסביר על כל אחת בנפרד:

## Author

לטבלה זו יש שתי שדות:

– author\_id (המזהה הייחודי של הסופר – ע"י ערך מספרי)

– author\_name (מציין את שם הסופר)

- המפתח: author\_id

## אינדקסים:

כל השדות קיבלו אינדקס על מנת לשפר ביצוע בחיפוש אחר ספרים שחוברו ע"י הסופר

## User

לטבלה זו יש חמש שדות:

- user\_id (המזהה הייחודי של המשתמש – ע"י ערך מספרי)
- user\_name (המזהה הייחודי של המשתמש – ע"י שם משתמש)
- password
- country
- age

**המפתח:** user\_id, user\_name

### אינדקסים:

כל שאר השדות חוץ מ password קיבלו אינדקסים כדי ליעל את תהליך חיפוש הספרים לדוגמא ספרים במדינה מסוימת או לפי חיפוש חופשי.

## Book

לטבלה זו יש שש שדות:

- book\_id (המזהה הייחודי של הספר – ע"י ערך מספרי)
- book\_name
- author\_id (מזהה הסופר של הספר - ע"י ערך מספרי)
- publisher
- img (תמונה – במידה ויש, של הספר)
- category

**המפתח:** book\_id

### :Foreign keys

- בינו לבין טבלת author על השדה author\_id ביחס של אחד לרבים, כי סופר אחד יכול לכתוב הרבה ספרים.

### אינדקסים

- כל השדות קיבלו אינדקס כדי לשפר חיפוש של הספרים חוץ מהשדות img,publisher.
- השדה author\_id קיבל אינדקס ביחד עם book\_id כדי שספרים של אותו סופר ימוספרו לפי סדר המספר המזהה שלהם.

## Comment

לטבלה זו יש שש שדות:

- comment\_id (מזהה ייחודי של תגובה – ע"י ערך מספרי)
- date (התאריך שבו ניתנה התגובה)
- user\_name (שם המשתמש של מי שכתב את התגובה)
- comment\_text (טקסט חופשי שיש להוסיף לכל תגובה)
- rate (ערך מספרי בין 1 ל 5)
- book\_id (מזהה הספר שעליו כתבו את התגובה)

**המפתח:** comment\_id

### :Foreign keys

- בין comment ובין user על שדה user\_name ביחס של אחד לרבים כי משתמש אחד יכול לכתוב הרבה תגובות.
- בין comment ובין book על שדה book\_id ביחס של אחד לרבים כי ספ אחד יכול לקל הרבה תגובות

### אינדקסים:

- חוץ מהשדה comment\_text כל השדות קיבלו אינדקס על מנת לשפר מהירות בעיקר במקרה של חישוב ממוצע של דירוגים.
- שדה date קיבל אינדקס כדי ליעל את תהליך הצגת תגובות אחרונות שנוספו למערכת או של הצגת תגובות אחרונות של משתמש מסוים.
- השדות rate,book\_id קיבלו שני אינדקסים משותפים. פעם כאשר rate מספר 1 כדי למספר ספרים בעלי אותו דירוג לפי המספר המזהה שלהם ולהיפך, למספר דירוגים של אותו ספר לפי הדירוגים עצמם.

## שאלות:

\*הערה:

בשאלות "גנריות" שהשתמשנו רק בתבנית, מופיע הסימון {0} שמתחלף בזמן ריצה בערך ספציפי לפי הקלט מהמשתמש.  
בשאלות כאלו נוסיף דוגמא עם ערכים ספציפיים.

## שאלות פשוטות:

שאלתא שמחזירה עבור ספר את ממוצע הדירוג עבור התגובות שקיבל, לפי המספר המזהה של הספר

```
SELECT book_name,avg(rate) as avg_rate
FROM book as b, comment as c
WHERE c.book_id = b.book_id and b.book_id = {0};
```

דוגמא ספציפית עם מספר מזהה 33:

```
SELECT book_name,avg(rate) as avg_rate
FROM book as b, comment as c
WHERE c.book_id = b.book_id and b.book_id = 33;
```

שאלתא שמחזירה את 20 התגובות האחרונות לספר מסוים, ממיון לפי תאריך

```
SELECT b.book_id, b.book_name,b.img, a.author_name,c.comment_txt,
c.rate, b.publisher,c.date, c.user_name
FROM comment c JOIN book b
ON c.book_id = b.book_id
JOIN author a ON b.author_id = a.author_id
WHERE b.book_id = {0}
ORDER BY c.date DESC LIMIT 20;
```



לדוגמא עם מספר מזהה 33:

```
SELECT b.book_id, b.book_name, b.img, a.author_name, c.comment_txt,
c.rate, b.publisher, c.date, c.user_name
FROM comment c JOIN book b
ON c.book_id = b.book_id
JOIN author a ON b.author_id = a.author_id
WHERE b.book_id = 33;
ORDER BY c.date DESC LIMIT 20
```

שאלתא שמחזירה את 20 התגובות האחרונות שהוזנו לאתר, ממוינים לפי תאריך.

```
SELECT b.book_id, b.book_name, c.user_name, c.rate, c.comment_txt,
c.date
FROM comment AS c, book AS b
WHERE c.book_id = b.book_id
ORDER BY date DESC limit 20;
```

הוספת תגובה של משתמש מסוים עבור ספר מסוים

```
INSERT INTO comment (date, user_name, comment_txt, rate, book_id)
SELECT CAST(NOW() AS DATE), '{0}', '{1}', {2}, {3};
```

\*מה שמופיע בסוגריים אלו ערכים גנריים

לדוגמא, הוספת תגובה עם הערכים הבאים

: 'Festooned', 'Very good', 5, 33

```
INSERT INTO comment (date, user_name, comment_txt, rate, book_id)
SELECT CAST(NOW() AS DATE), 'Festooned', 'Very good', 5, 33;
```

שאלתא שמחזירה את 50 התגובות האחרונות של משתמש מסוים, ממוינים לפי תאריך

```
SELECT c.comment_txt AS comment, c.rate, b.book_name AS  
book_name, a.author_name AS author_name, c.date AS date, b.book_id  
FROM comment c INNER JOIN user u ON c.user_name = u.user_name  
INNER JOIN book b ON c.book_id = b.book_id  
INNER JOIN author a ON b.author_id = a.author_id  
WHERE u.user_name = '{0}'  
ORDER BY c.date desc limit 50;
```

לדוגמא, עם שם המשתמש

:Festooned

```
SELECT c.comment_txt AS comment, c.rate, b.book_name AS  
book_name, a.author_name AS author_name, c.date AS date, b.book_id  
FROM comment c INNER JOIN user u ON c.user_name = u.user_name  
INNER JOIN book b ON c.book_id = b.book_id  
INNER JOIN author a ON b.author_id = a.author_id  
WHERE u.user_name = 'Festooned'  
ORDER BY c.date desc limit 50;
```

שאלתא שמחזירה את פרטי משתמש עם שם משתמש וסיסמא מסוימים

```
SELECT * FROM glassbook.user as u  
WHERE u.user_name = '{0}' AND u.password = '{1}';
```

לדוגמא, עם פרטי המשתמש הבאים

User\_name = ChikDot

Password = VqpctFWH

```
SELECT * FROM glassbook.user as u
```

```
WHERE u.user_name = 'ChikDot' AND u.password = 'VqpctFWH';
```

שאלתא שמכניסה משתמש חדש לטבלת המשתמש (user)

```
INSERT INTO user (user_name, password, country, age) VALUES ('{0}',  
'{1}','{2}', {3});
```

לדוגמא, עם הערכים הבאים:

Picachu257,AshRull!,Japan,3

```
INSERT INTO user (user_name, password, country, age)  
VALUES ('Picachu257','AshRull!','Japan',3);
```

**שאלתות "מורכבות":**

שאלתא שמחזירה את פרטי הספרים שמכילים בשמם/שם הסופר שלהם מחרוזת מסוימת, ממוינים לפי דירוגם הממוצע. מציגה רק 50 תוצאות ראשונות.

```
SELECT b.book_id as id, b.book_name AS Title, a.author_name as  
Author,avg(c.rate) as Avg_rate, b.img as Image, b.publisher as Publisher
```

```
from book as b
```

```
INNER JOIN comment as c ON c.book_id = b.book_id
```

```
INNER JOIN author as a ON b.author_id = a.author_id
```

```
WHERE a.author_name LIKE '%{0}%' OR b.book_name LIKE '%{0}%'
```

```
group by b.book_name
```

```
ORDER BY Avg_rate desc
```

```
LIMIT 50;
```

לדוגמא, עם המחרוזת

: israel

```

SELECT b.book_id as id, b.book_name AS Title, a.author_name as
Author,avg(c.rate) as Avg_rate, b.img as Image, b.publisher as Publisher
from book as b
INNER JOIN comment as c ON c.book_id = b.book_id
INNER JOIN author as a ON b.author_id = a.author_id
WHERE a.author_name LIKE '%israel%' OR b.book_name LIKE '%israel%'
group by b.book_name
ORDER BY Avg_rate desc
LIMIT 50;

```

שאלתא שמחזירה את עשרת פרטי הספרים המדורגים גבוה ביותר **במדינה מסוימת**,  
ממוינים לפי דירוגם הממוצע.

```

select b.book_id as id, b.book_name as Title ,a.author_name as author,
avg(rate) as Avg_rating, b.img as Image, b.publisher as publisher
From book as b, comment as c, user as u, author as a
where c.book_id = b.book_id and u.user_name = c.user_name and
u.country = '{0}' and b.author_id = a.author_id
group by book_name
order by Avg_rating desc
LIMIT 10;

```

לדוגמא עם המדינה

: slovakia

```
select b.book_id as id, b.book_name as Title ,a.author_name as author,  
avg(rate) as Avg_rating, b.img as Image, b.publisher as publisher
```

```
From book as b, comment as c, user as u, author as a
```

```
where c.book_id = b.book_id and u.user_name = c.user_name and  
u.country = 'slovakia' and b.author_id = a.author_id
```

```
group by book_name
```

```
order by Avg_rating desc
```

```
LIMIT 10;
```

שאלתא שמחזירה את עשרת פרטי הספרים המדורגים גבוה ביותר **בקטגוריה מסוימת**,  
ממוינים לפי דירוגם הממוצע.

```
select b.book_id as id, b.book_name as Title ,a.author_name as author,  
avg(rate) as Avg_rating, b.img as Image, b.publisher as publisher
```

```
From book as b, comment as c, user as u, author as a
```

```
where c.book_id = b.book_id and b.category = '{0}' and u.user_name =  
c.user_name and b.author_id = a.author_id
```

```
group by book_name
```

```
order by Avg_rating desc
```

```
LIMIT 10;
```

לדוגמא, עם הקטגוריה

:Classics

```
select b.book_id as id, b.book_name as Title ,a.author_name as author,  
avg(rate) as Avg_rating, b.img as Image, b.publisher as publisher
```

```
From book as b, comment as c, user as u, author as a
```

```

where c.book_id = b.book_id and b.category = 'Classics' and u.user_name
= c.user_name and b.author_id = a.author_id
group by book_name
order by Avg_rating desc
LIMIT 10;

```

שאלתא שמחזירה את 10 הספרים המדורגים ביותר:

```

SELECT b.book_id, b.book_name, a.author_name, b.publisher, b.img,
AVG(c.rate) as avg_rate
FROM book b JOIN comment c
ON b.book_id = c.book_id
JOIN author a ON b.author_id = a.author_id
GROUP BY b.book_id
ORDER BY avg_rate DESC LIMIT 10;

```

שאלתא שמחזירה פרטים אודות משתמש מסוים (כולל כמה תגובות הוא כתב, ואת ממוצע הדירוג של כל תגובותיו)

```

select u.user_name,age,country,
COALESCE(avg(rate),0) as Avg_rate,COALESCE(count(comment_id),0)
as numComments
from user u left join comment as c
ON c.user_name = u.user_name
where u.user_name = '{0}'
GROUP BY u.user_name, u.age, u.country;

```

לדוגמא, עבור המשתמש

:LovelyFun

```
select u.user_name,age,country,  
COALESCE(avg(rate),0) as Avg_rate,COALESCE(count(comment_id),0)  
as numComments  
from user u left join comment as c  
ON c.user_name = u.user_name  
where u.user_name = 'LovelyFun'  
GROUP BY u.user_name, u.age, u.country;
```

## תיעוד הקוד:

### **צד שרת**

השתמשנו ב web API של ASP.NET.

ראשית, יצרנו מחלקות אב של ארבעת הטבלאות שיש לנו בdb.

כלומר, יצרנו ארבעה מחלקות ראשיות – author, book, comment, user

בנוסף, יצרנו מחלקת search\_book בשביל בניית התשובה שנרצה להחזיר על חיפושים באתר.

יצרנו ארבעה controllers כך שלכל מחלקה ראשית כנ"ל יש קונטרול מתאים שיחזיר תשובה בהתאם לשאילתת הsql

בתיקית Controllers שמנו את כל מחלקות הקונטרולים.

הפעולות המרכזיות שהקוד שלנו עושה – בכל פונקציה שבקונטרולר שמנו connectionString שבאמצעותו התחברנו ל mysql db שלנו. לאחר מכן פתחנו את החיבור, כתבנו מפורשות את השאילתא שאנו רוצים לבצע, ועברנו שורה שורה והכנסנו למילון את התשובות שקיבלנו משרת ה sql ולבסוף החזרנו את המילון – כלומר קיבלנו תשובת JSON ובכך יכלנו להשתמש בו בקוד צד הלקוח.



## דוג' לשימוש טיפוס ל: db

```
string connectionString = "server=localhost;database=glassBook;uid=root;pwd=MyPassword;";
using (MySqlConnection connection = new MySqlConnection(connectionString))
{
    connection.Open();
    string sql = "SELECT * FROM book limit 500";
    using (MySqlCommand command = new MySqlCommand(sql, connection))
    {
        using (MySqlDataReader reader = command.ExecuteReader())
        {
            while (reader.Read())
            {
                int book_id = reader.GetInt32(0);
                string book_name = reader.GetString(1);
                int author_id = reader.GetInt32(2);
                string publisher = reader.GetString(3);
                string img = reader.GetString(4);
                string cat = reader.GetString(5);

                Book book = new Book();
                book.Book_id = book_id;
                book.Book_name = book_name;
                book.Author_id = author_id;
                book.Publisher = publisher;
                book.Img = img;
                book.Category = cat;
                books.Add(book);
                // retrieve data for other columns as needed
            }
        }
    }
    connection.Close();
}
return books;
```

נעבור על ה APIs השונים **שהשתמשנו בהם בפועל באתר** ונסביר מה כ"א עושה (לכל API יש פונקציה בקונטרולר המתאים שיש בו שימוש כנ"ל של פניה לשרת מסד הנתונים, והחזרה של json)

## **BooksController**

api/Books/{id} – מחזיר את 20 התגובות האחרונות של הספר עם מזהה id  
api/Books/avg/{id} – מחזיר את שם הספר עם מזהה id ואת הדירוג הממוצע שלו  
api/Books/SearchText/{search\_text} – מחזיר את פרטי הספרים שיש בשם שלהם/שם המחבר שלהם את המחרוזת שבמשתנה search\_text  
api/Books/SearchCountry/{search\_text} – מחזיר את פרטי הספרים בעלי הדירוג הגבוה ביותר במדינה שבמשתנה search\_text  
api/Books/SearchCategory/{search\_text} – מחזיר את פרטי הספרים בעלי הדירוג הגבוה ביותר שהם בקטגוריה שבמשתנה search\_text

## **CommentsController**

api/Comments/ – מחזיר את 20 התגובות האחרונות שנכתבו באתר  
api/Comments/top10/ – מחזיר את פרטי עשרת הספרים המדורגים גבוה ביותר בממוצע  
api/Comments/mycomments/{user\_name}/ – מחזיר את 50 התגובות האחרונות של משתמש מסוים (שבמשתנה user\_name)  
Api של הוספת תגובה הוא ב post (כל הנל היו ב get) הוא ע"י הכנסה של הפרטים הבאים: User\_name, Comment\_txt, Rate, Book\_id  
ה api הוא api/Comments/addComment – מחזיר סטטוס 200 במידה והצליח להוסיף את התגובה לdb, אחרת – יחזיר סטטוס שגיאה 400.

## UserController

`/api/Users/{id}` - מחזיר את פרטי המשתמש ה `username` שבמשתנה `id`  
`/api/Users/Login?user_name={username}&password={pwd}` /  
להתחבר עם שם משתמש שבמשתנה `username` ועם הסיסמא שבמשתנה `pwd`  
במידה ואכן יש משתמש כנ"ל אז יוחזר סטטוס 200, אחרת – יוחזר סטטוס שגיאה 404.

`/api/Users/isExistUserName?user_name={username}` יחזיר `true` במידה וקיים  
שם משתמש במסד הנתונים עם `username` שבמשתנה `username` אחרת, הוא יחזיר  
שקר.

Api של הרשמת משתמש לאתר הוא ב `post` (כל הנל היו ב `get`) הוא ע"י הכנסה של  
פרטים הבאים: `Username, Password, Country, Age`  
ה `api` הוא `/api/Users/Register` - מחזיר סטטוס 200 במידה והצליח להוסיף את  
המשתמש ל `db` (במידה והגיל לא מתאים/סיסמא לא כוללת לפחות אות אחת וספרה  
אחת/שם משתמש כבר קיים ב `db` – יחזיר סטטוס שגיאה), אחרת – יחזיר סטטוס שגיאה  
400.

## צד לקוח

השתמשנו בריאקט כדי ליצור את האתר שלנו.

נשים לב שכדי לצרוך את APIs הנ"ל השתמשנו בפונקציית `fetch` שבה צרכנו את הדרוש, קיבלנו תשובה בתור `json` ועדכנו את האתר בהתאם.

הקומפוננטות המרכזיות:

`LoginPage.js` - קומפוננטה עבור דף ההתחברות לאתר

`RegisterPage.js` - קומפוננטה עבור דף ההתחברות לאתר

`MainPage.js` - קומפוננטה עבור הדף הראשי של האתר בו מוצגים עשרת הספרים המודרגים הכי גבוה, התגובות האחרונות ואופציות סינון וחיפוש שונות

`BookDetails.js` - קומפוננטה עבור פרטי ספר מסוים.

`NavBar.js` – קומפוננטה עבור החלק העליון שבאתר (שדרכו אפשר לחזור לדף הראשי/פרטי המשתמש/להתנתק מהמערכת)

`ProfileDetails.js` - עבור דף פרטי המשתמש המחובר כעת לאתר

`Search_bar.js` - קומפוננטה עבור אופציות החיפוש השונות שהמשתמש יכול לבחור

`SearchPage.js` - קומפוננטה עבור דף התוצאות לאחר החיפוש ב `Search_bar`

השתמשנו `useState` וב `useEffect` כדי שהאתר יתעדכן בצורה דינמית

בנוסף, השתמשנו ב `react-router-dom` כדי שחווית המעבר בין הדפים באתר תהיה חלקה ללא "רפרושים".