

## פרויקט גמר

### Book Store Website



**:מגישי**

шибלי עבד אלרחמאן ת.ז 305047540

דאהר נאסר קאיד ת.ז 311242127

הפרויקט מוגש כפרויקט גמר לקבלת דיפלומת הנדסאי

במגמת : תוכנה-התמחות סייבר

אוניברסיטת אריאל

## תוכן עניינים

3	רקע
4	תיאור חנוך הספרים
9	מה הפרויקט אמר לו לחדש או לשפר
10	סקירת מצב קיים בשוק, אילו בעיות קיימות
11	תיאור התוכנה
12	נתונים טכניים
13	ציוויל דריש לביצוע והרצה הפרויקט
14	בעיות ופתרונות במהלך הפיתוח
15	אלגוריתם המשמש להמלצה
16	גבולות המערכת
17	ארQUITטורה ופונקציונליות
37	טופולוגיה ארכיטקטורת המערכת
38	בסיס הנתונים
39	טכנולוגיות בשימוש
40	שפות הפיתוח
44	סביבת השירות
45	אבלחתה מידע
46	גומי סיון
47	אמצעים
48	פונקציות מתוך django.views.home ב django
51	Matrix Factorization Algorithm בקוד
53	model
56	taskid serializer
58	בדיקות
60	GUI – ל��וח משחק
64	טרשיימי מערכת מרכזים
66	ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט

# מבוא:

עם התפתחותה המהירה של טכנולוגיות האינטרנט, מספר אתרי אינטרנט המוכריםספרים גדל אשר הגבירו את התחרות ביניהם. האתר שלנו מסיע לקנות את הספרים באופן מקוון עם מערכת המלצה, שהוא אחד הכלים החזקים כדי להגדיל את הרוח ושמירה על הקונה. מערכת המלצה של הספר חיבת להמליץ על ספרים המתעניין בהן הקניין.

באתר שلنנו נעשה שימוש נרחב במערכות המלצה על מוצרים למשתמשים המתאימים ביותר. מערכת זו משתמשת בתכונות של סינון שיתופי להפקה המלצות עילות ואפקטיביות. המלצה שיתופית היא כנראה הטכנולוגיות המוכרות ביותר, המישמות ביותר והובורות ביותר. מערכות מליצות שיתופיות מצטרבות דירוגים של אובייקטים, מזהים המשותף בין המשתמשים על סמך הדירוגים שלהם, טכנולוגיה זו הוכיחה שהיא שימושית ביותר בכל הנוגע לייצור מכירות. המערכת עשויה שימוש בדירוגים בספרים מסוימים על ידי המשתמש הפועל להעיר את הדמיון בין פרופיל המשתמש כדי לחזות המלצות הטובות ביותר עבורו.

המערכת עשויה שימוש בטכנייקת "matrix factorization", אשר מאפשרת לנו לגלוות את התכונות הסמיוטיות העומדות בסיס האינטראקטיביות בין ספרים.

**מטרת הפרויקט:**

לחיפוש ורכישת (web application) מטרת הפרויקט היא פיתוח אפליקציה ויב ספרים המאוחסנים באופן מקוון.

- לספק פלטפורמה כדי להציג את הספרים באופן מקוון.
- לרכוש ספרים באופן מקוון.
- להמליץ על ספר למשתמש/לקונה.

# תיאור הchnoot:

The screenshot shows a website for a book store. At the top, there is a dark header bar with the text "SALE UP TO 70% OFF. SHOP NOW" on the left, "Welcome Abdo" and "Logout" in the center, and a shopping cart icon on the right. Below the header is a navigation bar with categories: Home, Classic, Horror, Biography, Fiction, Comic, Science, Recommendations, Orders, and Contact. The main content area features a large image of a window looking out onto a garden, with bookshelves visible on either side. Overlaid on the image is the text "BUY BOOKS ARE NOW ON LINE WITH US". Two red speech bubbles are overlaid on the image, one on the left and one on the right. The left bubble contains the Hebrew text "חיפוש לפי קטגוריה" (Search by category) and the right bubble contains "חיפוש לפי מילה" (Search by word). In the top right corner of the main content area, there is a search bar with the placeholder "Search for a book..." and a magnifying glass icon.

**SALE UP TO 70% OFF. SHOP NOW**

**127.0.0.1:8000 says**

Steve Jobs will be added to your cart

Order online or call us : +9725595

**OK**

for a book... 

[Home](#) [Classic](#) [Horror](#) [Biography](#) [Fiction](#) [Comic](#) [Science](#) [Recommendations](#) [Orders](#) [Contact](#)

[Home](#) / Books

**CATEGORIES**

- Classic
- Horror
- Fiction
- Comic
- Science
- Biography
- Self Development
- Novel

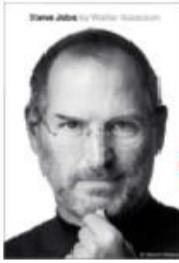
Item on page 9  Default sorting 



Michelle Obama - חסידת רשות

★ 0  
\$40 \$48

[ADD TO CART](#)



Steve Jobs

★ 4  
\$20 \$6

[ADD TO CART](#)

כאשר המשתמש בוחר  
לקנות את ספר סטיב גובו  
מתוך קטגורית ביוגרפיה  
האתר מופיע הودעה  
שהספר התווסף לעגלת  
הקניות

« 3 2 3 4 5 »

SALE UP TO 70% OFF. [SHOP NOW](#)

Welcome Abdo   Logout   Help

[!\[\]\(4fc2f7cc81639d6310529267a2ee602c\_img.jpg\)](#)

Order online or call us : +972559518504

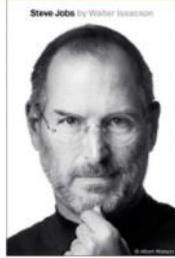
## BOOK STORE

Search for a book... 

Home   Classic   Horror   Biography   Fiction   Comic   Science   Recommendations   Orders   Contact

[!\[\]\(dc7d17b015a4a5f15a29473bc04652a8\_img.jpg\) Home](#) / Checkout Page

YOUR SHOPPING CART CONTAINS: 1

SL No.	Product	Quality	Product Name	Price	Remove
1		<a href="#"></a> <a href="#"></a>	Steve Jobs	20	<a href="#"></a>

Billing Address:

Full name:

Steve Jobs(2) - 20

\$00.00

Mobile number:

Total Service Charges -

\$40

Landmark:

Town/City:

[!\[\]\(21226b58c700e5231ab98d27101bac58\_img.jpg\) Continue Shopping](#)

Address:

Address type:  Office  Home  Other

[Place order](#)

כשהמשתמש נכנס לעגלת הקניות כפי שמצוון לעליה באדום הוא יוכל לראות את הספר שקבע וatz יזין את הפרטים שלו לאחר מכן לוחץ על place order כדי להזמין את הספר

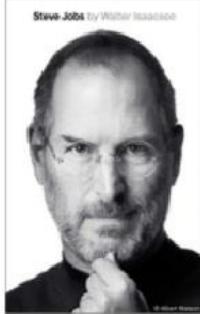
SALE UP TO 70% OFF. [SHOP NOW](#)[Welcome Abdo](#) [Logout](#) [Help](#)

Order online or call us : +972559518504

## BOOK STORE

Search for a book...

[Home](#) [Classic](#) [Horror](#) [Biography](#) [Fiction](#) [Comic](#) [Science](#) [Recommendations](#) [Orders](#) [Contact](#)[Home](#) / Orders Page**YOUR ORDERS:**

SL No.	Product	Product Name	Quantity	Price	Status
1		Steve Jobs	1	20	Ordered

 [Continue Shopping](#)

כאשר המשתמש לוחץ על התגית Orders הוא יכול לראות את ההזמנות שלו והוא  
 יראה את שם המוצר הכמות והמחיר גם סטטוס ההזמה  
**סטטוס ההזמנה משתנה: Ordered, shipped, out for delivery, delivered**

SALE UP TO 70% OFF. [SHOP NOW](#)[Welcome Abdo](#) [Logout](#) [Help](#)

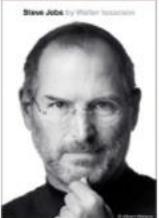
Order online or call us : +972559518504

## BOOK STORE

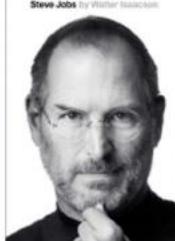
Search for a book...

[Home](#) [Classic](#) [Horror](#) [Biography](#) [Fiction](#) [Comic](#) [Science](#) [Recommendations](#) [Orders](#) [Contact](#)[Home](#) / Orders Page

Waiting for feedback:

SL No.	Product	Product Name	Rating	Comment	Submission
1		Steve Jobs	★★★★★	<input type="text"/>	<a href="#">Submit feedback</a>

YOUR ORDERS:

SL No.	Product	Product Name	Quantity	Price	Status
1		Steve Jobs	1	20	Delivered

כאר הסטוטו משתנה ל delivered המשתמש יכול לתת ריטיניג וגם לשלווח פידבק

## מה הפרויקט אמר לחדר או לשפר?

הפרויקט שלנו הוא אפליקציה ווב למכירת ספרים שנבנה בצורה קלה ונוחה למשתמשים בכל הгалאים שבא להציג את הספרים למעמדם הנוכחי במיוחד בתחום שבלנו, שבתקופה זו נסגרו כל מוסדי הלימוד וגם רוב תחומי העבודה שזה יצר הרבה זמן פניו לרוב אוכלוסייה שמעורר את האנשים לקנות ולקראן ספרים באתר שלנו אנו הולכים להשתמש באלגוריתם המשומש להמלצות ספרים שנמצאים בתוך המערכת באמצעות סינון שיתופי, שדרך הטכניקה זו יהיה קל מאוד علينا להמליץ ללקוח ספר שהאחוזה שלו יאהב ויקנה גבוהה במיוחד.

## סקירת מצב קיימם בשוק, אילו בעיות קיימות.

הweeneyון של חנות ספרים הוא רעיון נפוץ בזמן שלנו עם התפתחותה המהירה של טכנולוגיות האינטרנט, שזה גורם להרבה מוצריים דומים הקיימים היום בשוק אבל כדי להצליח באותו רעיון עם המספר הגדל של המתחרים פיתחנו אלגוריתם מתמטי ייחודי המשמש את הלקחות שלנו בהמלצה על ספרים בצורה יעודית שיתאים ללקוח באופן ספציפי ויגרום לו לקנות את הספר בצורה מהירה ונוחה ולהגיע לבחירה הנכונה בעבר בפחות זמן .

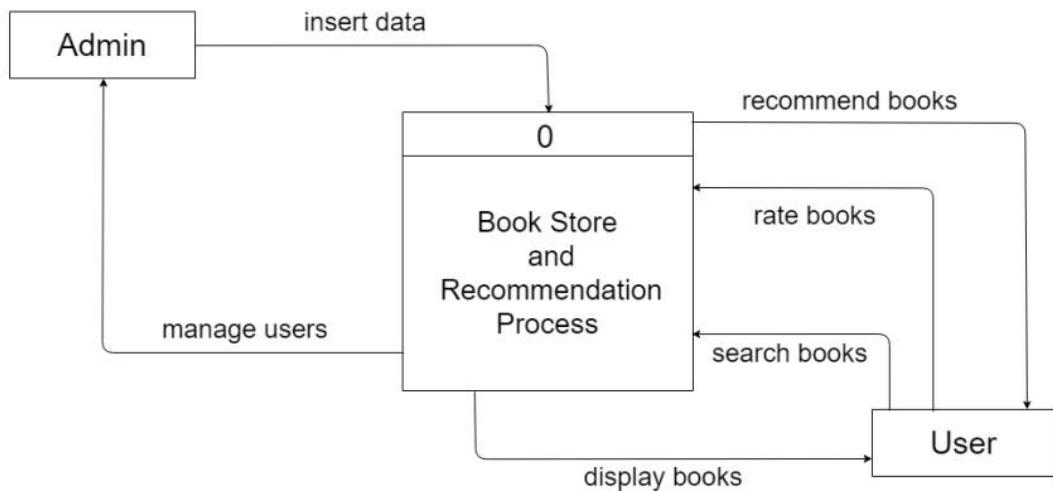
## תיאור התוכנה :

המערכת מספקת תוכנות שונות הן למנהל והן למשתמשים. המערכת מסוגלת לשרת מספר משתמשים לפחות אחד.

המערכת מספקת המלצות למשתמשים על בסיס matrix factorization algorithm וככל משתמש יכול להרשם באתר ולהשתמש בו, והוא מאד ידידותית למשתמש והרבה יותר קל לקיים אינטראקציה עם המשתמש, אין צורך להכשרה מיוחדת להפעלת המערכת.

המערכת פותחה בפייטון בשילוב bootstrap, JavaScript , Django Framework, html, CSS, jQuery שם פועלם יחד.

המערכת דורשת HTML CSS, Bootstrap, JavaScript ו-WSGI server, Django framework המשמש לצד לקוח וצד שרת.



האיור הוא הדמיה של עיבוד הנתונים ותבנית תוכנן תהליך חנות הספרים וזרימת העבודה

זרימת הנתונים של המערכת ותהליכיים של תהליך חנות הספרים. הוא מכיל את כל זרימת המשתמשים ואת הישיות שלהם

## נתוניים טכניים

### 1.3 ממשקים מוצעים

(כל חיבור לאתר ולשירותechibim חיבור לאינטרנט).

אתר אינטרנט.

### 2.3 סוגי שירותים

המערכת מכילה שני סוגי שירותים:

שירותות ללקוח מושעה - השירות יאפשר למשתמשים שנרשמו לאתר.

(השירות ידרש אimoto זהות משתמש ע"י login ואחרי שייאשר,

הוא יקבל גישה ל base Data – לצורך שליפת מידע אישי או שמירת מידע

חדש) שירות זה מאפשר ללקוח לנהל את החשבון שלו.

שירותות למנהל המערכת - השירות ידרש אimoto זהות משתמש ע"י, login ולאחר

שאושר שהמשתמש הינו רשום במערכת, הוא יקבל גישה ל base Data – לצורך

שליפת מידע אישי או שמירת מידע חדש

שירות זה מאפשר למנהל המערכת לנהל את מאגר המשתמשים והחנות, לצפות בתוצאות כל

האתר, לצפות בתוצאות כל הלקוחות.

## ציד חומרה ותוכנה הנדרשים לביצוע הפרויקט

visual studio code-Django Web Framework

MySQL Server

גישה לאינטרנט

## ציד חומרה ותוכנה הנדרשים להרצה הפרויקט:

דף

חיבור לאינטרנט

פתרונות אפשריים לאותם בעיות	בעיות צפויות במהלך הפיתוח ופתרונות
ליצור אפליקציה שנראית מעולה, מציעה שירות מועיל ומספקת חוות משתמש נعימה ונוחה כל שנייה.	פיתוח אפליקציה שתבלוט מול המתחרים.
הקפידו ליצור עיצוב רספנסיבי ( responsive design ) שמקנה יותר גמישות וונזרות ולכך מותאם בקלות יחסית לפורמטים ומוסכים שונים.	התאמת האפליקציה לפלטפורמות שונות .
אנו משתמש בבקשת <code>http</code> המחזירות במקרה של תקלה קוד שגיאה המספק מידע על גורמי התקלה שיאפשר לנו לתקן אותה בהירות ויעילות.	איובו תקשורת נתונים ובחיבור לאינטרנט.
יצירת סביבה "בטוחה" לצד השרת הכלולת פילטר זיהוי משתמשים, וכן טיפול מתאים בexceptions למניעת קriseה של המערכת בעת חריגה.	קriseה של צד השרת במהלך השימוש .
העברת נתונים בין השירותים באובייקט <code>NSObject</code> על מנת ליצור פורמט אחיד בין המרכזות.	עומס יתר של בקשות על השירות.
עבודה בצורה אסינכרונית מול מסד הנתונים שלנו.	עומס על מסד הנתונים.
כמו בחיים, גם בפיתוח תוכנה, היכולת להתמודד עם שינויים לא צפויים ועדרי עדיפויות מתחלפים מצריכה גמישות, פתיחות ויכולת להסתגל לתנאים משתנים. הדרך הטובה ביותר לשמר על כל היכולות האלה היא להבין שינויים והפטעות הם חלק בלתי נפרד מהפרויקט ולהיות מוכנים להתאים את צורת החשיבה וטורת העבודה לכל שינוי מבני שיפור עזרימה של התהילה.	קביעת סדרי עדיפויות , בכל פרויקט מעורבים מספר גורמים בעלי סדרי עדיפויות שונים. בនוסף בכך הפרויקט יכולם לצוץ גורמים לא צפויים שימושיים על התהילה ומיצרים היררכיות חדשות שיכולים להקשوت עליהם לדבוק בלוז'ז / או התקציב שהגדרתם מראש.

## אלגוריתם המשמש להמלצה

### אלגוריתם לפרק גורמים של מטריצות (Matrix Factorization Algorithm)

האלגוריתם המשמש במערכת זו הוא האלגוריתם לפרק גורמים של מטריצות דרך אוסף של אלגוריתמי סינון שיתופיים המשמשים במערכות ממלייצים.

אלגוריתם זה פועל על ידי פירוק מטריצת האינטראקציה של ה user-item במערכות. התוצר של שתי מטריצות מלכניות ממדיות נמוכות יותר, במערכות הוא משמש למליץ על ספרים ללקוחות. הוא ממליץ על המוצר הגבוה ביותר

שודרג על ידי הלקוחות. כאשר הלקוחות רוצים להשיג את הספרים המומלצים זה

אלגוריתם המליץ על הספרים המדורגים הכי גבוה מבין כל הספרים הזמינים. מערכת זה עוזר לחסוך זמן ללקוחות מכיוון שרק הספרים הטובים ביותר הם מומלצים.

### טכנית

פירוק ערך יחיד (SVD++) זהה גישה מבוססת-מודל factorizes מטריצת הדירוג של ה user-item כתוצר שלשתי מטריצות ממדיות תחתונות, הראשונה כוללת שורה לכל משתמש, ואילו לשניה יש עמודה לכל פריט. השורה או העמודה המשויכים למשתמש או פריט ספציפיים הם המכונה גורמים סינויים. הוא דוחס מטריצה של פריט משתמש לממד נמוך יציג במוניחים של גורמים סינויים. יתרון אחד בשימוש בגישה זו הוא בכך במקומות שישיה מטריצה ממדיות גבוהה המכילה מספר גדול של חסרי ערכים, שזה נותן לנו להתעסק במטריצה קטנה בהרבה במרחב הממד התחתון.

תהליך של מילוי מטריצת הניתוח על ידי ביצוע פקטורייזציה של מטריצה. מערכת שלנו שורות מטריצת הממלצות מייצגות את המשתמש והעמודות מייצגות את הספר והתאים מתמלאים עם הדירוגים הניתנים על ידי כל משתמש בכל ספר. ישנו תאים רבים ובעלים יתמלאו באפסים, לאחר מכן באמצעות טכניקת פקטורייזציה התא הריק מתמלא ויוצר רשימה ספרים עם הספר בעל הדירוג הגבוה על ידי המשתמש.

## גבולות המערכת

האתר לא יעבד ללא קישור לרשות (נדרשת גישה לשרת)

המשתמש יכול להירשם וליצור חשבון כדי לחפש ספרים מתוך אוסף רחב של ספרים אשר מצמצמים את זמן החיפוש של המשתמש כאשר הספרים מסודרים ומופצים על סמך קטגוריות. ניתן לרכוש ספרים באופן מקוון באמצעות שיטת cash (Cash) או delivery. מערכת המלצות מספקת את הספר המודרג ביותר למשתמש לרכישה. עבור מספר גדול של נתוני, יש צורך בשרת גדול יותר כדי להפעיל את התוכניות בצורה חלקה ובעילוות אשר איננו יכולים לישם בשרת המקומי שלנו. פרויקט זה מוגבל לאמצעי תשלום בזמןן בלבד. מומלץ רק ספרים בעלי דירוג גבוה ביותר מכיוון שמערכת זו אינה מאפשרת טכnicות המלצות היברידיות.

## ארQUITקטורה ופונקציונליות

### ארQUITקטורת המערכת

#### **שכבה תצוגה (Presentation layer):**

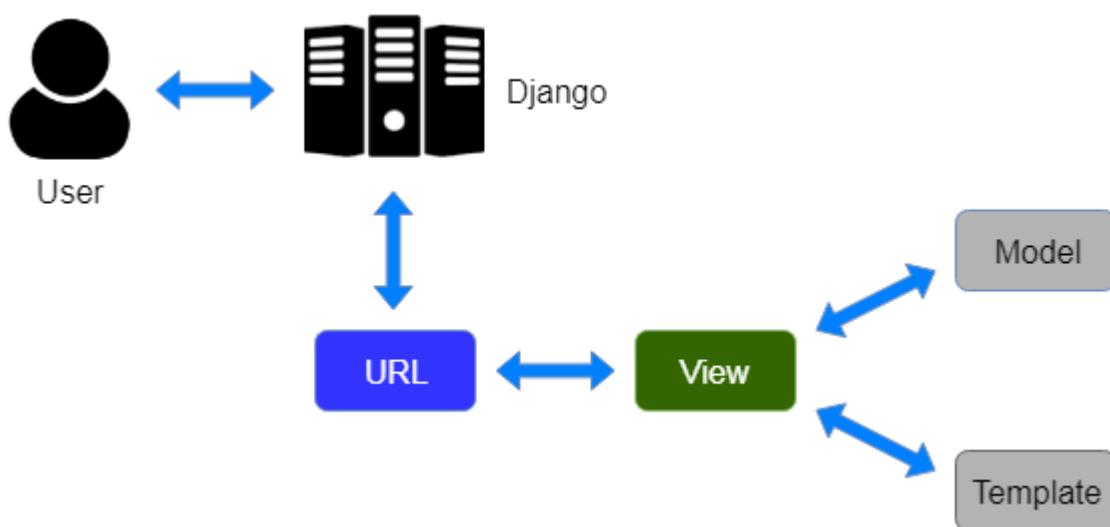
התבנית (template) היא שכבה תצוגה המטפלת בחלק ממשך המשמש. התצוגה (view) משמשת לביצוע היגיון העסקי ו互動אקטיבית עם מודל (model) להעברת הנתונים ועיבוד התבנית (template).

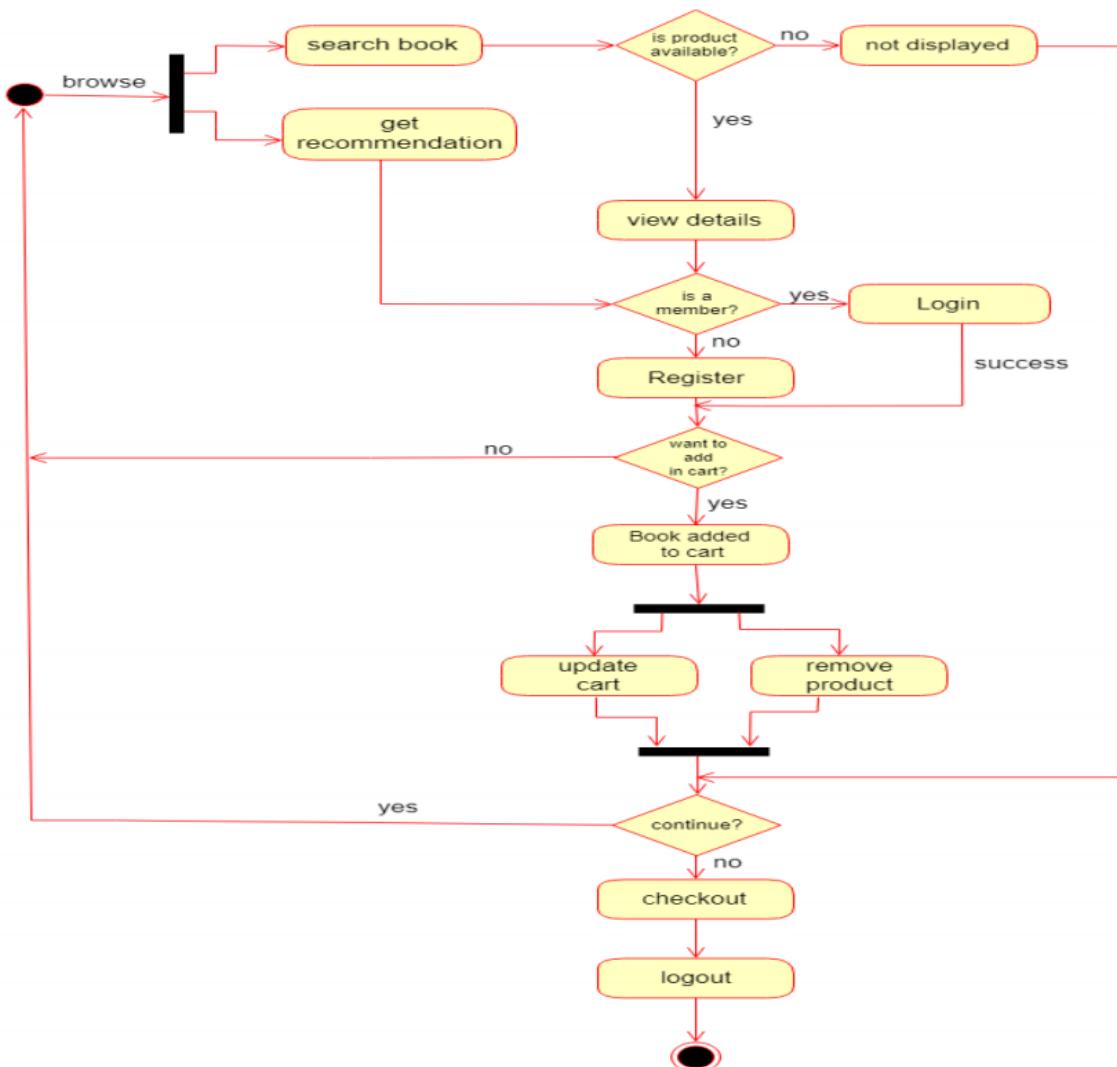
#### **שכבה היישום (Application layer):**

נקראת גם "שכבה הלוגיקה העסקית". שכבה זו שולחת פונקציונליות של היישום על ידי ביצוע עיבוד הנתונים הפרטני, בדיקות הולידציה הדרשות לננתונים, ניתוח הצורך של המידע הדרוש, והרכבת הבקשות שיועברו לבסיס הנתונים (לשכבה זו אין גישה ישירה לבסיס הנתונים).

#### **שכבה הנתונים (Data layer):**

שכבה זו עובדת שירותים מול בסיס הנתונים – מבצעת את השאלות הנדרשות, ומתקבלת את תוצאות השאלות שרצו בסיס הנתונים. שכבה זו שומרת על הנתונים עצמאיים ובלתי תלויים בשרטוי היישומים ובלוגיקה העסקית. ונכתבת בצורה גנרייה המאפשרת שימוש חוזר בשכבה זו עבור כל פרויקט אחר וכל אובייקט אחר. אחסון הנתונים בשכבה נפרדת משפר גם את הביצועים של היישום.





האיור זהה מראה לנו את הלוגיקה של ארכיטקטורת המערכת והפעולות שניתן לעשות באתר המשתמש יכול לגלוש באתר ויכול לבצע חיפוש לספר שהוא רוצה או לקבל המלצה על ספר. לאחר חיפוש אם המוצר זמין המשתמש יכול להציג את פרטי המוצר רק אם הוא / היא מחובר או עושה חיפוש לספר אחר. אם המשתמש אינו משתמש רשום עליו להירשם למערכת וליצור חשבון חדש. לאחר צפיה במוצר אם המשתמש מרוצה מה מוצר אז המשתמש יכול להוסיף את המוצר לעגלת או להמשיך לקנות. באופן דומה המשתמש יכול לעדכן את העגלת כמו הגדרת כמות המוצר או הסרת אותו מהעגלת, לאחר מכן המשתמש יכול להמשיך לגלוש שוב או לבצע צ'ק-אут ולהתנתק מהמערכת.

## פונקציה המציגת משתמשים באתר

<pre>router.register(r'users', views.UserViewSet)</pre>	חתימת הפונקציה:
<b>GET</b>  <pre>/users/ Content-Type: application/json</pre>	cotratet ha-beksha:
<pre>{     "username": "abed1",     "email": "agladsh65@gmail.com",     "first_name": "abed",     "last_name": "shibli",     "date_joined": "2020-10-24T11:16:18.352267Z",     "last_login": "2020-10-24T11:16:18.914761Z" }, {     "username": "abdo",     "email": "abedshibli90@gmail.com",     "first_name": "",     "last_name": "",     "date_joined": "2020-10-17T10:54:39.894811Z",     "last_login": "2020-10-24T15:14:30.574538Z" }</pre>	tocn ha-beksha :
<b>Status:</b>  <pre>200 OK</pre>	קוד תשובה:
<b>EMPTY RESPONSE</b>	tocn ha-tshuba:

## פונקציה המציגת ספרים באתר

חיתימת הפונקציה:	
<pre>return Response(serializer.data)</pre>	
<p><b>GET</b></p> <p>/book-list/</p> <p><b>Content-Type:</b> application/json</p>	<b>cotractה הבקשה:</b>
<pre>{     "category": 1,     "name": "The Stand-By Stephen King",     ,     "author": "Stephen King",     "price": 9,     "rating": 0,     "stock": 3 }, {     "category": 1,     "name": "Hell House-by Richard Matheson",     "author": "by Richard Matheson",     "price": 3,     "rating": 4,     "stock": 5 }, {     "category": 2,     "name": "החתא וטונשו",     "author": "פינדדור מיכאלוביץ דוסט'קוב",     "price": 15,     "rating": 0,     "stock": 7 }, {     "category": 5,</pre>	<b>תוכן הבקשה :</b>

```
        "name": "Amazing Spider-Man Epic Collection: Maximum Carnage",  
        "author": "Mark Bagley (Illustrator), Alex Saviuk (Illustrator), Tom Lyle (Illustrator), Sal Buscema (Illustrator), David Michelinie (Contributor), Terry Kavanagh (Contributor), J.M. DeMatteis (Contributor), Tom Defalco (Contributor)",  
        "price": 30,  
        "rating": 0,  
        "stock": 9  
    },  
    {  
        "category": 5,  
        "name": "Avengers Next #1 VF ; Marvel comic book",  
        "author": "Cyberspace Comics.",  
        "price": 14,  
        "rating": 0,  
        "stock": 3  
    },  
    {  
        "category": 3,  
        "name": "מיישל אובמה - הסיפור של לי"  
    },  
    {  
        "author": "מיישל אובמה",  
        "price": 40,  
        "rating": 0,  
        "stock": 6  
    },  
    {  
        "category": 3,  
        "name": "Steve Jobs",  
        "author": "Walter Isaacson",  
        "price": 20,  
        "rating": 4,  
        "stock": 2  
    },  
    {  
        "category": 4,  
        "name": "The Hunger Games (The Hunger Games #1)",  
        "author": "Suzanne Collins",  
        "price": 15,  
        "rating": 4,  
        "stock": 10  
    }]
```

```

"author": "by Suzanne Collins",
"price": 16,
"rating": 0,
"stock": 1
},
{
"category": 4,
"name": "Harry Potter and the Order
of the Phoenix (Harry Potter #5)",
"author": "by J.K. Rowling, Mary Gr
andPré",
"price": 32,
"rating": 0,
"stock": 4
},
{
"category": 6,
"name": "היקום האלגנט",
"author": "בראיין גריין",
"price": 36,
"rating": 0,
"stock": 7
},
{
"category": 6,
"name": "אבולוציון",
"author": "אורן הרמן",
"price": 12,
"rating": 0,
"stock": 3
}

```

**Status:**

200 OK

**קוד תשובה:**

EMPTY RESPONSE

**תוכן התשובה:**

## פונקציה שמצויה פתרים עבור ספר מסוים

<pre>return Response(serializer.data)</pre>	חתימת הפונקציה:
<pre>GET /book-detail/ The Stand-By Stephen King Content-Type: application/json</pre>	cotretaet haBeksha:
<pre>{     "category": 1,     "name": "The Stand-By Stephen King",     "author": "Stephen King",     "price": 9,     "rating": 0,     "stock": 3 }</pre>	tocan haBeksha :
Status:  200 OK	קוד תשובה:
EMPTY RESPONSE	tocan haTshuba:

## פונקציה שמוחקת ספר מסוים מהאטר:

<code>return Response(serializer.data)</code>	חתימת הפונקציה:
<code>DELETE</code> <code>/book-delete/The Stand-By Stephen King</code> <code>Content-Type: application/json</code>	cotrattת הבקשה:
<code>"Item successfully deleted!"</code>	תוכן הבקשה :
<code>Status:</code> <code>200 OK</code>	קוד תשובה:
<code>EMPTY RESPONSE</code>	תוכן התשובה:

## פונקציה שמוסיפה ספר לאתר:

חוובה למלא את כל שדות בהוספה ספר חדש

<pre>return Response(serializer.data)</pre>	חתימת הפונקציה:
<b>POST</b> /book-create/ <b>Content-Type:</b> application/json	cotrattet ha bokerha:
<pre>{   "name": "The Stand",   "description": "This is the way the world ends: with a nanosecond of computer error in a Defense Department laboratory and a million casual contacts that form the links in a chain letter of death. And here is the bleak new world of the day after: a world stripped of its institutions and emptied of 99 percent of its people. A world in which a handful of panicky survivors choose sides -- or are chosen. ",   "author": "by Stephen King",   "price": 3,   "image": "/media/media/thestand.jpg",   "rating": 4,   "on_offer": false,   "pre_discount_price": 6,   "stock": 5,   "category": 1 }</pre>	tocn ha bokerha :
<b>Status:</b> 200 OK	קוד תשובה:
EMPTY RESPONSE	tocn ha tshuba:

## פונקציה שמעדכנת פרטי ספר באתר

עלינו לעדכן את השדות שאנו רוצים לדוגמה price היה 6 שונה ל 8  
pre\_discount\_price היה 8 שונה ל 12

<pre>return Response(serializer.data)</pre>	חתימת הפונקציה:
<b>POST</b> /book-update/ <b>Content-Type:</b> application/json	cotnent-type הבקשה:
<pre>{     "name": "Hell House-by Richard Matheson",     "description": "Can any soul survive?\r\nRegarded as the Mount Everest of haunted houses, Belasco House has witnessed scenes of almost unimaginable horror and depravity. Two previous expeditions to investigate its secrets met with disaster, the participants destroyed by murder, suicide or insanity. Now a new investigation has been mounted - four strangers, each with his or her own reason for daring the unknown torments and temptations of the mansion.",     "author": "by Richard Matheson",     "price": 8,     "image": "/media/media/hell.jpg",     "rating": 4,     "on_offer": false,     "pre_discount_price": 12,     "stock": 5,     "category": 1 }</pre>	תוכן הבקשה :
<b>Status:</b> 200 OK	קוד תשובה:
EMPTY RESPONSE	תוכן התשובה:

## פונקציה שבודקת אם יש פריטים בעגלת הקניות

הfonkzia תבודוק אם קיימ פריטים בעגלת הקניות אם כן תחזיר There is items in your cart אם לא תחזיר there is no items in your cart

<pre>return Response('There is items in your cart') else:     return Response('there is no items in your cart')</pre>	חתימת הפונקציה:
GET  /checkcart/  Content-Type: application/json	cotretaet haBeksha:
"there is no items in your cart"	tocn haBeksha :
Status:  200 OK	koed haTshuba:
EMPTY RESPONSE	tocn haTshuba:

## יצירת משתמשים

```
CREATE TABLE `auth_user` (
    `id` int NOT NULL AUTO_INCREMENT,
    `password` varchar(128) NOT NULL,
    `last_login` datetime(6) DEFAULT NULL,
    `is_superuser` tinyint(1) NOT NULL,
    `username` varchar(150) NOT NULL,
    `first_name` varchar(150) NOT NULL,
    `last_name` varchar(150) NOT NULL,
    `email` varchar(254) NOT NULL,
    `is_staff` tinyint(1) NOT NULL,
    `is_active` tinyint(1) NOT NULL,
    `date_joined` datetime(6) NOT NULL,
    PRIMARY KEY (`id`),
    UNIQUE KEY `username` (`username`)
)
```

## מחיקת משתמשים

`DELETE FROM `bookstoredb`.`auth_user``

`; <WHERE <{where_expression}`

## בחירה משתמשים

לצורך log in | logout

```
SELECT `auth_user`.`id`,  
       `auth_user`.`password`,  
       `auth_user`.`last_login`,  
       `auth_user`.`is_superuser`,  
       `auth_user`.`username`,  
       `auth_user`.`first_name`,  
       `auth_user`.`last_name`,  
       `auth_user`.`email`,  
       `auth_user`.`is_staff`,  
       `auth_user`.`is_active`,  
       `auth_user`.`date_joined`
```

`FROM `bookstoredb`.`auth_user`;`

## עדכון פרטי משתמשים

`UPDATE `bookstoredb`.`auth_user``

## SET

```

`id` = <{id: }>,
`password` = <{password: }>,
`last_login` = <{last_login: }>,
`is_superuser` = <{is_superuser: }>,
`username` = <{username: }>,
`first_name` = <{first_name: }>,
`last_name` = <{last_name: }>,
`email` = <{email: }>,
`is_staff` = <{is_staff: }>,
`is_active` = <{is_active: }>,
`date_joined` = <{date_joined: }>
WHERE `id` = <{expr}>;

```

[הווסף ספר חדש](#)

```

CREATE TABLE `product_book` (
  `id` int NOT NULL AUTO_INCREMENT,
  `name` varchar(200) NOT NULL,
  `description` longtext NOT NULL,

```

```

`price` int NOT NULL,
`image` varchar(100) NOT NULL,
`category_id` int NOT NULL,
`rating` int NOT NULL,
`author` varchar(500) NOT NULL,
`pre_discount_price` int NOT NULL,
`on_offer` tinyint(1) NOT NULL,
`stock` int NOT NULL,
PRIMARY KEY (`id`),
KEY
`product_book_category_id_e3807c66_fk_product_cat
egory_id` (`category_id`),
CONSTRAINT
`product_book_category_id_e3807c66_fk_product_cat
egory_id` FOREIGN KEY (`category_id`) REFERENCES
`product_category` (`id`)
)

```

[מחיקת ספר מהאתר](#)

DELETE FROM `bookstoredb`.`product\_book`

WHERE <{where\_expression}>;

### עדכון פרטי ספר באתר

UPDATE `bookstoredb`.`product\_book`

SET

`id` = <{id: }>,

`name` = <{name: }>,

`description` = <{description: }>,

`price` = <{price: }>,

`image` = <{image: }>,

`category\_id` = <{category\_id: }>,

`rating` = <{rating: }>,

`author` = <{author: }>,

`pre\_discount\_price` = <{pre\_discount\_price: }>,

`on\_offer` = <{on\_offer: }>,

`stock` = <{stock: }>

WHERE `id` = <{expr}>;

```
CREATE TABLE `product_cart` (
    `id` int NOT NULL AUTO_INCREMENT,
    `timestamp` datetime(6) NOT NULL,
    `book_id` int NOT NULL,
    `user_id` int NOT NULL,
    `quantity` int NOT NULL,
    PRIMARY KEY (`id`),
    KEY `product_cart_book_id_4e727864_fk_product_book_id`(`book_id`),
    KEY `product_cart_user_id_4b8441ee_fk_auth_user_id`(`user_id`),
    CONSTRAINT `product_cart_book_id_4e727864_fk_product_book_id` FOREIGN KEY (`book_id`) REFERENCES `product_book`(`id`),
```

## CONSTRAINT

```
`product_cart_user_id_4b8441ee_fk_auth_user_id`  
FOREIGN KEY (`user_id`) REFERENCES `auth_user` (`id`)  
)
```

[מחיקת פרטים מעגלת הקניות](#)

```
SET id = <{row_id}>;  
  
DELETE FROM product_category  
    USING product_category  
    WHERE product_category.id =  
COMMIT;
```

[הוספה הזמן](#)

```
CREATE TABLE `product_order` (  
    `id` int NOT NULL AUTO_INCREMENT,  
    `timestamp` datetime(6) NOT NULL,  
    `book_id` int NOT NULL,  
    `user_id` int NOT NULL,  
    `quantity` int NOT NULL,  
    `address` longtext NOT NULL,
```

```
`city` varchar(100) NOT NULL,  
 `phone` int NOT NULL,  
 `status` varchar(100) NOT NULL,  
 `address_type` varchar(100) NOT NULL,  
 `full_name` varchar(100) NOT NULL,  
 `landmark` varchar(300) NOT NULL,  
 `rated` tinyint(1) NOT NULL,  
 PRIMARY KEY (`id`),  
 KEY  
 `product_order_book_id_67413561_fk_product_book  
 _id` (`book_id`),  
 KEY  
 `product_order_user_id_01a7f9c6_fk_auth_user_id`  
 (`user_id`),  
 CONSTRAINT  
 `product_order_book_id_67413561_fk_product_book  
 _id` FOREIGN KEY (`book_id`) REFERENCES  
 `product_book` (`id`),  
 CONSTRAINT  
 `product_order_user_id_01a7f9c6_fk_auth_user_id`  
 FOREIGN KEY (`user_id`) REFERENCES `auth_user` (`id`)
```

)

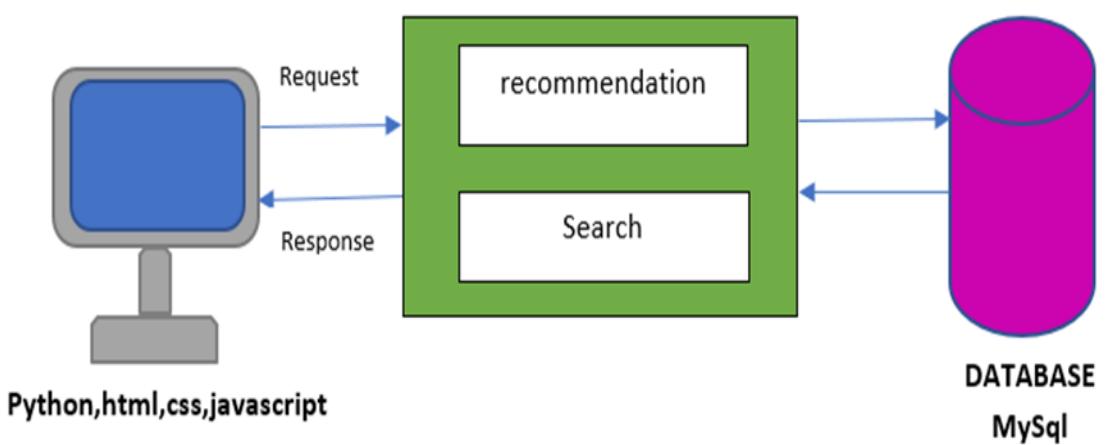
## מחיקת הזמן

```
DELETE FROM product_order  
    USING product_order  
 WHERE product_order.id =  
 COMMIT;
```

## יצירת קטגוריות

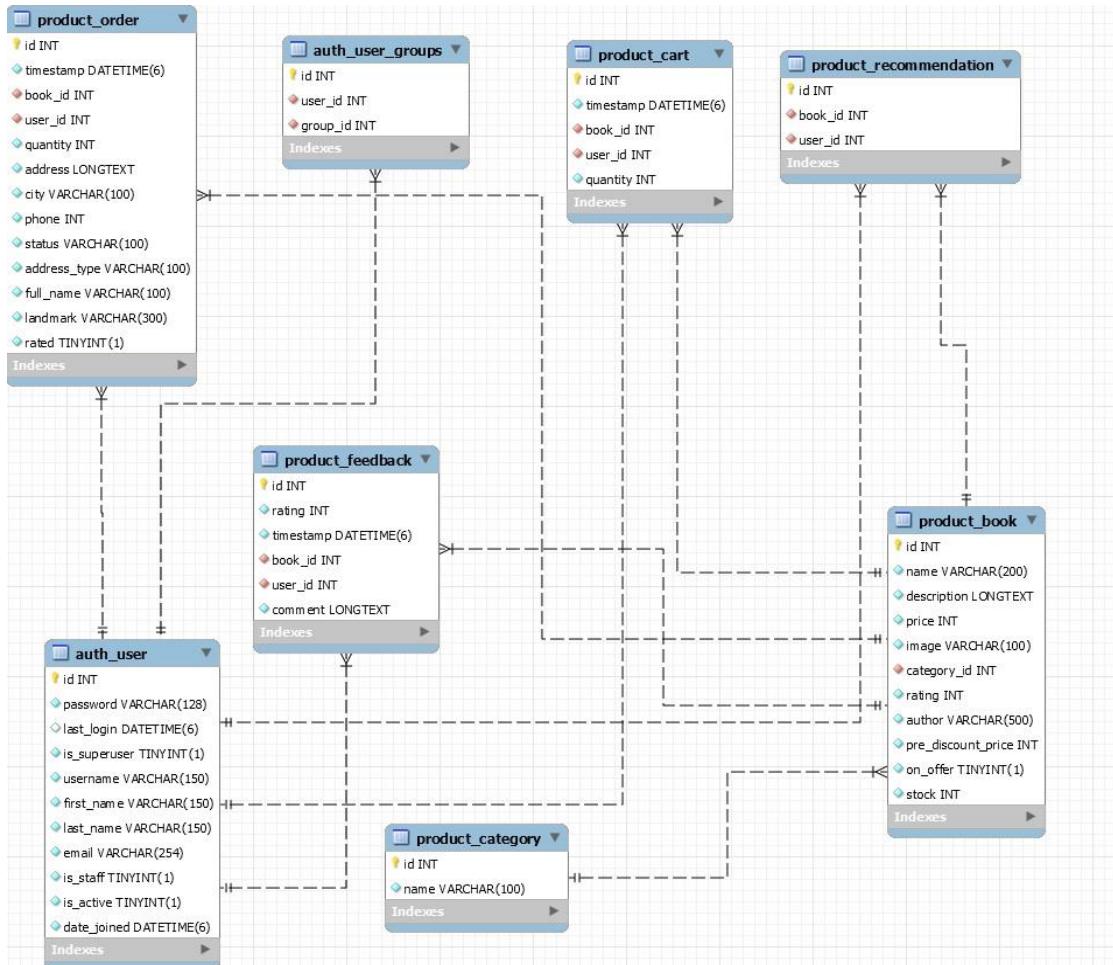
```
CREATE TABLE `product_category` (  
    `id` int NOT NULL AUTO_INCREMENT,  
    `name` varchar(100) NOT NULL,  
    PRIMARY KEY (`id`)
```

## טופולוגיה ארכיטקטורת המערכת



## בסיס הנתונים

בסיס הנתונים ימושב באמצעות טבלאות. בשרת נשתמש בטכנולוגיה המאפשרת עבודה אסינכרונית מול מסד הנתונים כדי ליעיל את הביצועים ולקצר את זמי הרמתנה.



## טכנולוגיות בשימוש

**Bootstrap** - תוכנה הבנויה על קוד פתוח ומאפשרת למשתמשים לבנות אתרים ואפליקציות ווב. ה-Bootstrap מורכבת משלושה חלקים עיקריים. החלק הראשון הוא ה-CSS הפורמט לעיצוב דפי אינטרנט מבחןת תבניות, צבעים ו��ונה. החלק השני הוא משק ידידותי עבור המשתמש, היכול כפותרים, הודעות או תמונות. החלק השלישי הוא הסקריפטים הרבים של Bootstrap אשר מאפשרים להרחב את שימושי האתר. נוסף על כך, אחת התכונות הבולטות של ה-Bootstrap, נכון להיום, היא יכולת להתאים את האתר למסך הקטנים של הסמארטפוןים, כמו למשל אתריםレスפונסיביים לניר.

**Django Frame work** - היא טכנולוגיה קוד פתוח ליישומי אינטרנט בצד השרת, שפותחה על ידי Django Software Foundation מבוססת על פיתון בחרנו בה כי לדעתנו קל לנו לבנות את הצד השרת עם שפת פיתון וגם יש בה ספריות שאנו צריכים כדי לבנות את הפROYKT שלנו.

**MySQL** - MySQL הוא מסד נתונים יחסית, רב נימי ורבי משתמשים מבוסס שפת SQL (Structured Query Language). נקרא ע"ש MySQL, בתו של מונטי וידניוס, אחד מהհackers המקוריים. התוכנה פותחה במקור על ידי החברה השוודית AB MySQL. כיום היא בבעלות חברת אורקל. התוכנה היא חלק מ-LAMP, אוסף תוכנות תשתית פופולריות שעומדות בסיסם של אתרים רבים, כגון ויקיפדיה.

**jQuery** - זו ספרייה של JavaScript שיכולה ליצור הרבה פונקציונליות במעט קוד. לדוגמה, ניתן ליצור אינימציה ואפקטים, AJAX שמאפשר להציג אלמנטים חדשים בדף בלי צורך לרענן את הדףFOX וולידציה של טפסים. רואים את jQuery בכל מקום באינטרנט, מתפריטים שנפתחים כשהעכבר עלייהם, תצוגת שkopיות, והשלמה אוטומטית של הטקסט בתיבת טקסט כמו שקרה כשאתם מקלידים משהו לתוך שורת הכתובות בגוגל. לא רק גוגל משתמש ב-jQuery, ולמעשה זו ספרייה פופולרית ביותר בז'אנר קלות הטעינה. הקלות היחסית של כתיבת הקוד והכמות הגדולה של פлаг-אינים שימושיפה הרבה יותר מאשר jQuery.

## שפות הפיתוח

.**HTML** - שפת תגיותلتצוגה ועיצוב דפי אינטרנט ותוכןلتצוגה בדף. זו שפת התגיות המרכזית בעולם האינטרנט, המהווה שלד למრבית עמודי התוכן באינטרנט.

( **CSS** ) - הם פורמט לעיצוב דפי אינטרנט. ה吉利ונות קובעים את עיצובה של תגים ב- HTML, XHTML וכל שפה דומה ל-XML לבניית אתר אינטרנט.

**JavaScript** - שפת תכנות דינמית מונחית עצמים המותאמת לשילוב באתר אינטרנט ורצה על ידי דפדפן האינטרנט לצד הלקוח

**Python** - פיתון היא שפה מרובת-פראגמות, המאפשרת תכנות מונחה-עצמים, תכנות פרוצדורלי, ובמידה מסוימת גם תכנות פונקציוני. לשפה ספרייה סטנדרטית גדולה וענפה, והיא תומכת באופן מובנה בהרחבה שלה אל שפות אחרות (בתחומיים: C, C++, Java, C# ו-).

## חבילות תוכנה

- **Numpy**

is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

- **Pandas**

pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series

- **Pillow**

Pillow is the friendly Python Imaging Library adds image processing capabilities to your Python interpreter. This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities.

- **scipy**

SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering. SciPy builds on the NumPy array object and is part of the NumPy stack which includes tools like Matplotlib, pandas and SymPy, and an expanding set of scientific computing libraries. This NumPy stack has similar users to other applications such as MATLAB, GNU Octave, and Scilab. The NumPy stack is also sometimes referred to as the SciPy stack

- **Django rest framework**

Django REST framework is a powerful and flexible toolkit for building Web APIs. Some reasons you might want to use REST framework: The Web browsable API is a huge usability win for your developers. Authentication policies including packages for OAuth1a and OAuth2. Serialization that supports both ORM and non-ORM data sources. Customizable all the way down - just use regular function-based views if you don't need the more powerful features. Extensive documentation, and great community support. Used and trusted by internationally recognised companies including Mozilla, Red Hat, Heroku, and Eventbrite.

פייתון היא שפת תכנות דינמית מהנפוצות ביותר. פייתון תוכנה תור שימת דגש על קריאות הקוד, וכוללת מבנים המיעדים לאפשר ביטוי של תוכניות מורכבות בדרך קצרה וברורה. אחד המאפיינים הבולטים בתחריר השפה הוא השימוש בהזחה להגדרת בלוקים של קוד (ללא שימוש בסוגרים או במלחים שמורות לצורך כך, כמו ברוב השפות הנפוצות).

מערכת הטיפוסים בפייתון היא בעלת המאפיינים הבאים:

- **חזקת**: לא ניתן להפעיל על משתנה פעולה שלא מוגדרת על הטיפוס אליו הוא שייך. לדוגמה, ניסיון לבצע פעולה חזקה בין שתי מחרוזות - גם אם הן מסמנות מספרים, למשל "5" בחזקת "6" - יגרור שגיאה.
- **דינמית**: האובייקטים, ולא המשתנים בשפה, הם השייכים לטיפוס. כל משתנה בשפה יכול להצביע על אובייקט מכל טיפוס. כאשר מגדירים פונקציה, אין הגבלה על סוג הטיפוס שיועבר בכל קריאה אליה. יתכן שבקריה אחת יועבר מספר שלם, ובקריה אחרת תועבר מחרוזת. הטיפוסיות היא "טיפוסיות ברווז", (Duck Typing) "דרינו" נקבעת על פי הפעולות אותן ניתן לבצע על המשתנה. אם יש צורך לוודא כי משתנה שהתקבל כפרמטר בפונקציה, לדוגמה, הוא מטיפוס "מספר שלם", יש לבצע בדיקה מפורשת (בדרך כלל בעזרת פקודות assert או if).
- **mobilitat**: אין הכרזה על קיום המשתנים. שם של משתנה מוגדר אם הוא מופיע בצדיה השמאלי של פעולות השמה, כפרמטר לפונקציה, כמשתנה לולה, כפרמטר לתפיסה של חריגה או בפקודת del מאפיין זה מאפשר גמישות רבה בכתיבת הקוד, וכתיבת קוד קצר יותר, אך מנגד מקשה על זיהוי ואיתור שגיאות בזמן ההידור, ומתקשה על איתור טעויות הקלדה.
- **סמנטייקת התיחסות (Reference)**: משתנים הם שמות של מצביעים (References) אל אובייקטים, ולא שמות של האובייקטים עצמם. כאשר מתבצעת פעולה השמה בין המשתנים בצדקה,  $u=x$  לא מועתק תוכן האובייקט אליו מצביע.  $u$  במקומות זה, המשתנה  $x$  מוגדר כך שיצביע אל אותו אובייקט אליו מצביע.  $u$  פרמטרים לפונקציות מועברים על ידי העתקה של המצביעים אליהם. טכנית זו "קריאה על ידי ערך" (Call by Value) "אר רבים" (ובهم מציא השפה) מכנים זאת "קריאה על ידי התיחסות לאובייקט" (Call by Object Reference). קריאה לפונקציה בצדקה,  $x(foo)$  שבתוכה מבצעים,  $x=5$  לא תגרום למשתנה  $x$  להתייחס לאובייקט אחר מזה שהוא התיחס אליו קודם.
- **טוווח הכרה סטטי (או לקסיקלי)**: (שם של משתנה מוגדר רק בקטעי קוד המוכלים בקטע שבו בוצעה אליו פעולה כריכה. למשל, משתנה בשם  $x$  המוגדר בתוך פונקציה אחת לא יהיה קשור בשום דרך למשתנה  $x$  המוגדר בפונקציה אחרת).

## סביבת השירות

Wsgi+mysqlserver

## מבוא וארגון הקבצים

המידע ישמר בבסיס נתונים המחבר לרשת.

### 5. אמצעי אחסון

שרת (מחשב המחבר) לאינטרנט ודף המציג דפי אינטרנט.

## סביבת הטמעה ושימוש

צד שרת

יכלול שימוש בbasis נתונים MySQL

יכלול שימוש בAPI – חיצוני.

צד השירות מאפשר שימוש בAPI – על מנת לשלוח בקשות On-line ובכך יצליח

להתmesh בכל סוג ה-UI

צד לקוחות

המערכת תתמוך בדף האינטרנט המוביילים ותהיה זמינה דרך דומיין פשוט. אתר

האינטרנט יהיה רספונסיבי ויתאים לכל הרוחולציות.

## אבטחת מידע

### עדים עיקריים

#### confidentiality: סודיות

\_amcui סודיות מגנים על מידע מפני גישה בלתי מורשתית ושימוש לרעה. ברוב מערכות המידע יש מידע שיש בו רגשות מסוימת. יתרון שמדובר במידע עסקי קנייני שהმתחרים יכולים להשתמש בו לטובתם, או במידע האישני הנוגע לעובדי ארגון, ללקוחותיו או לקוחותיו.

#### integrity: שלמות

amcui תקינות מגנים על מידע מפני שינוי בלתי מורשה. amcui אלה מספקים ביטחון לדיקוק ובשלמות הנתונים. הדרוש בהגנה על מידע כולל גם נתונים המאוחסנים במערכות וגם נתונים המועברים בין מערכות כמו דואר אלקטרוני. לשמירה על תקינות, אין צורך רק לשלוט על הגישה ברמת המערכת, אלא גם להבטיח שימושם המערכת יכולים לשנות רק מידע שהם מורשים לשנות באופן לגיטימי.

#### Availability: זמינות

על מנת שמערכת מידע תהיה שימושית עליה להיות זינה למשתמשים מורשים. amcui זמינות מגנים על גישה בזמן וללא הפרעה למערכת. חלק מהאומים הבסיסיים ביותר לזמן אינטנסיביים אופיים וכוללים כשלים בחומרה, השבתה בלתי מתוכננת של תוכנה ובעיות רוחב פס בראשת התקפות זדוניות כוללות צורות חבלה שונות שנעודו לגרום

נזק לארגון על ידי מניעת גישה למשתמשים למערכת המידע.

### גורם סיכון

הטכנולוגיה : הטכנולוגיה היא אחד מגורמי הסיכון矧ן מערכות החומרה תוכנה והתקשורת.

המשתמשים : האנשים שמשתמשים בטכנולוגיות הם מגורמי הסיכון המשבירים ביותר על אבטחת המידע שלנו, כי הם שייכים לחוליה החלשה במיוחד כי בעת השימוש באפליקציה הם עלולים לעשות שגיאות המיצרות פריצות במערך ההגנה הנובע משימוש לא נכון בלי או עם כוונה.

תהליכיים : טעויות ושגיאות הנובעות מטעויות אנוש או נקודת תורפה נוכל לצמצם את הנזק שנובע מהם דרך תכנון נכון ויעיל של תהליכיים בארגון שלנו שנוטן לנו אפשרות זיהוי מוקדם של פגיעות או ניסוי פגיעה במערכת שלנו וכן נוכל לצמצם את הנזקים ולפטור את הבעיה במהירות.

## אמצעים

הזהות – דרך השימוש באמצעות זיהוי והזהות שבו מי שմבקש את השירות הוא צריך לספק למערכת מידע המזהה אותו כך לא ניתן לאנשים לקבל שירות בלי הזהות,אמצעי הזיהוי גדלו בתקופה الأخيرة, האמצעי הנפוץ ביותר שהשתמשנו בו הוא שימוש בשם משתמש וסיסמה.

הרשאות – לביצוע פעולות הקשורות בשינוי ועדרון מהיקה או הוספת מידע על קובץ כלשהו או כל שינוי במערכת אנו צריכים להשתמש בהרשאות דרכם אנו נבחר למי יש את האפשרות לעשות שינויים כלשהו במערכת כגון אם זה משתמש רגיל (לפחות) או מנהל המערכת ADMIN(ADMIN) כך אנו נגביל את אפשרות שינוי המידע שתוויאו לאיבוד מידע או קרייסות.

## פונקציות מותר ב `home.views`

ב חלק זה מתרוצשנו פונקציות אלה נמצאות בקוד על מנת להתבצע באתר כל פונקציה מוחזירה דף

מרונדר Html

```

from django.shortcuts import render, redirect
from django.contrib.auth.models import User, auth
from product.models import *
from .models import *
from django.db.models import Q

from django.contrib.auth.decorators import login_required

# Create your views here.

def home(request):
    all_books = Book.objects.filter(on_offer=True)
    try:
        recommendations = Recommendation(user=request.user)
    except:
        recommendations = all_books.order_by('-rating')
    context = {
        'all_books':all_books,
        'recommendations':recommendations
    }

    return render(request, 'index.html',context)

def search(request):
    if request.method == 'POST':
        search = request.POST['search']
        searched_books = Book.objects.filter(Q(name__icontains = search))
        context = {
            'all_books':searched_books,
            'all_books_count':len(searched_books)
        }
        print(searched_books)
    return render(request,'products.html',context)

def register(request):
    if request.method == 'POST':

```

```
first_name = request.POST['first_name']
last_name = request.POST['last_name']
username = request.POST['username']
email = request.POST['email']
password1 = request.POST['password1']
password2 = request.POST['password2']

if password1 == password2:

    if User.objects.filter(username=username).exists():

        return redirect('/')

    elif User.objects.filter(email=email).exists():

        return redirect('/')

    else:
        user = User.objects.create_user(
            username=username, password=password1, email=email, first_name=first_name, last_name=last_name)
        user.save()

        print('user created')
        user_auth = auth.authenticate(
            username=username, password=password1)

        if user_auth is not None:
            auth.login(request, user_auth)

            return render(request, 'index.html')

    else:

        print('not working')
        return redirect('/')

else:
    return render(request, 'register.html')

def login(request):

    if request.method== 'POST':
        username = request.POST['username']
        password = request.POST['password']
```

```
user = auth.authenticate(username=username,password=password)

if user is not None:
    auth.login(request, user)
    return redirect('home')
else:
    print('invalid credentials')
    return redirect('login')

else:
    return render(request,'login.html')

def logout(request):
    auth.logout(request)
    return redirect('/')

@login_required(login_url='login')
def contact(request):

    if request.method == 'POST':

        contact_obj = Contact(
            user = request.user,
            name = request.POST['name'],
            email = request.POST['email'],
            message = request.POST['message']
        )
        contact_obj.save()

    return render(request,'contact.html')
```

## בחלק זהה קיים האלגוריתם algorithm

```

def recommendation(request):
    all_feedback = Feedback.objects.all()
    all_good_book = Book.objects.all().order_by('-rating')
    #print(all_feedback)
    current_user_feedback = Feedback.objects.filter(user= request.user).order_
by('-rating')

    try:
        feedback_dictionary = {}
        feedback_matrix = []
        for feedback in all_feedback:
            single_feedback = {}
            single_feedback['user'] = feedback.user
            single_feedback['book'] = feedback.book
            single_feedback['rating'] = feedback.rating

            feedback_dictionary[feedback.user] = single_feedback
            feedback_matrix.append(single_feedback)
        print(feedback_matrix)

        df = pd.DataFrame(data=feedback_matrix)
        print(df)
        book_rating_pivot = df.pivot(index = 'user', columns = 'book', values
= 'rating').fillna(0)
        print(book_rating_pivot)
        X = book_rating_pivot.values.T
        X.shape
        SVD = TruncatedSVD(n_components= len(book_rating_pivot)-
1, random_state=17)
        matrix = SVD.fit_transform(X)
        matrix.shape

        warnings.filterwarnings("ignore",category =RuntimeWarning)
        corr = np.corrcoef(matrix)
        corr.shape
        book_title = book_rating_pivot.columns
        book_title_list = list(book_title)
        #print(book_title_list)
        coffey_hands = book_title_list.index(current_user_feedback[0].book)
        print(coffey_hands)

        corr_coffey_hands = corr[coffey_hands]
    
```

```
print(corr_coffey_hands)
recommended_books = (list(book_title[(corr_coffey_hands >= 0.9)]))
print(recommended_books)

for recom in recommended_books:
    if Recommendation.objects.filter(user=request.user).filter(book=recom).exists():
        print('already recommended')
    else:
        recom_obj = Recommendation(
            user = request.user,
            book = recom
        )
        recom_obj.save()
        print('recom saved')

context = {
    'all_books':recommended_books,
    'all_books_count':len(recommended_books)
}

except:
    print('exception happent')
    context={
        'all_books': all_good_book,
        'all_books_count':len(all_good_book)
    }
```

## תפקיד model

תפקיד ה `model` בקוד הוא לשוף `data` מה `database` ולאחסן אותה בתוך `class` כ`R`ה `model` מאפשר לנו להשתמש בנוטוי ה `database` בקוד לדוגמה

```
from django.db import models
from django.contrib.auth.models import User
from django.utils.timezone import now

# Create your models here.

class Category(models.Model):
    name = models.CharField(max_length=100)

    def __str__(self):
        return self.name


class Book(models.Model):
    category = models.ForeignKey(Category, on_delete=models.CASCADE)
    name = models.CharField(max_length=200)
    description = models.TextField(blank=True)
    author = models.CharField(max_length=500, blank=True)
    price = models.IntegerField(blank=True)
    image = models.ImageField(upload_to = 'media')
    rating = models.IntegerField(default=0)
    on_offer = models.BooleanField(default=False)
    pre_discount_price = models.IntegerField(default=0, blank=True)
    stock = models.IntegerField(default=1, blank=True)

    def __str__(self):
        return self.name


class Cart(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    book = models.ForeignKey(Book, on_delete=models.CASCADE)
    quantity = models.IntegerField(default=0)
    timestamp = models.DateTimeField(default=now)

    def __str__(self):
        return self.user.username
```

```

class Order(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    book = models.ForeignKey(Book, on_delete=models.CASCADE)
    quantity = models.IntegerField(default=0)
    full_name = models.CharField(max_length=100, blank=True)
    city = models.CharField(max_length=100, blank=True)
    address = models.TextField(blank=True)
    landmark = models.CharField(max_length=300, blank=True)
    address_type = models.CharField(max_length=100, blank=True)
    phone = models.IntegerField(default=0, blank=True)

    ORDER_STATUS = (
        ('Ordered', 'Ordered'),
        ('Shipped', 'Shipped'),
        ('Out for delivery', 'Out for delivery'),
        ('Delivered', 'Delivered'),
    )

    status = models.CharField(
        max_length=100,
        choices=ORDER_STATUS,
        blank=True,
        default='Ordered'
    )

    rated = models.BooleanField(default=False)
    timestamp = models.DateTimeField(default=now)

    class Meta:
        ordering = ['-status']

    def __str__(self):
        return self.user.username + ' ordered ' + self.book.name + ' on ' +
str(self.timestamp)

class Feedback(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    book = models.ForeignKey(Book, on_delete=models.CASCADE)
    rating = models.IntegerField(default=0)
    comment = models.TextField(blank=True)
    timestamp = models.DateTimeField(default=now)

    def __str__(self):
        return self.user.username + ' rated '+self.book.name

```

```
class Recommendation(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    book = models.ForeignKey(Book, on_delete=models.CASCADE)

    def __str__(self):
        return self.user.username + ' ----> ' + self.book.name
```

## תפקיד ה serializer בקוד

ה serializer מחזיק במידע מתוך class ב model כלו או חלק ממנו כך למשתמש יוכל לקבל מידע מהשרת

בתוך fields אנחנו יכולים לציין איזה מידע אנו רוצים כך ש `__all__` הוא כל המידע או פשוט בתר [ ] איזה מידע אנו רוצים

```
from rest_framework import serializers
from product.models import Book,Cart
from django.contrib.auth.models import User

class UserSerializer(serializers.HyperlinkedModelSerializer):
    class Meta:
        model = User
        fields = [ 'username', 'email','first_name','last_name','date_joined',
        'last_login']

class BookSerializer(serializers.ModelSerializer):
    class Meta:
        model = Book
        fields = '__all__'

class CartSerializer(serializers.ModelSerializer):
    class Meta:
        model = Book
        fields = '__all__'
```

## פונקציות Django REST framework

השתמשנו django rest framework כדי להציג תשובות

ל(GET,POST,DELETE) להציג משתמשים, להציג ספרים לעדכן או למחוק אותם וגם  
לביקורת עגלת הקניות

```
from rest_framework import viewsets
from rest_framework import permissions
from product.serializers import UserSerializer
from rest_framework.decorators import api_view
from rest_framework.response import Response
from product.serializers import BookSerializer,Cartserializer

class UserViewSet(viewsets.ModelViewSet):
    queryset = User.objects.all().order_by('-date_joined')
    serializer_class = UserSerializer
    permission_classes = [permissions.IsAuthenticated]

@api_view(['GET'])
def apiOverview(request):
    api_urls = {
        'book':'/book-list/',
        'bookdetails':'/book-detail/<str:pk>/',
        'delete':'/book-delete/<str:pk>/',
        'Create':'/book-create/',
        'Update':'/book-update/<str:pk>/',
        'cart':'/checkcart/',
    }

    return Response(api_urls)

@api_view(['GET'])
def booklist(request):
    books = Book.objects.all()
    serializer = BookSerializer(books, many=True)
    return Response(serializer.data)

@api_view(['GET'])
def bookdetail(request, pk):
```

```
books = Book.objects.get(name=pk)
serializer = BookSerializer(books, many=False)
return Response(serializer.data)

@api_view(['POST'])
def bookupdate(request, pk):
    book = Book.objects.get(name=pk)
    serializer = BookSerializer(instance=book, data=request.data)

    if serializer.is_valid():
        serializer.save()

    return Response(serializer.data)

@api_view(['GET'])
def checkcart(request):

    cart_obj = Cart.objects.all()

    if len(cart_obj) > 0:
        return Response('There is items in your cart')
    else:
        return Response('there is no items in your cart')

@api_view(['DELETE'])
def bookdelete(request, pk):
    book = Book.objects.get(name=pk)
    book.delete()

    return Response('Item succesfully deleted!')

@api_view(['POST'])
def bookcreate(request):
    serializer = BookSerializer(data=request.data)

    if serializer.is_valid():
        serializer.save()

    return Response(serializer.data)
```

## בדיקות

### Test Case for Admin Login and logout

Test Case	Input	Expected Result	Test Result
Successful Admin Login	Username = abed Password = abed123	התחברות מוצלחת לAdmin	Admin התאחד בהצלחה
Admin Login Fail	Username = abed, Password = kayed1234	חיבת ליקוף הودעה "Please enter the correct username and password for a staff account."	תקפוּז הודעה Please enter the "correct username and password for a ".staff account
Successful Admin Logout	Click logout option	הדף ציריך להפנות את Admin לדף כניסה	הדף מפנה את Admin לדף כניסה והודעה logged out

### Test Case for User Login and Logout

Test Case	Input	Expected Result	Test Result
Successful User Login	Username = abed2 Password = 222	דף כניסה ציריך להפנות לדף הבית עם סטטוס מחובר ושם המשתמש מוצג בדף מלמעלה	דף כניסה מפנה את המשתמש לדף הבית עם סטטוס מחובר ושם המשתמש מוצג בדף מלמעלה
User Login Fail	Username=abed2, Password=rabe123	הודעה חיבת ליקוף למשתמש חדש עם check your user name or password	הודעה תקפוּז למשתמש בדף חדש check your user name or password
Successful User Logout	Click logout option	הדף ציריך לחזור לדף הבית והחשבון יצא	חזר לדף הבית והחשבון יצא

### Test Case for Search

Test Case	Input	Expected Result	Test Result

חיפוש מוצלח	Search keyword = hell	כל הספרים שמוופעים בשם שלהם המילה hell צריכים להופיע על המסך	כל הספרים שמוופעים בשם שלהם המילה hell הופיעו על המסך Hell House-By Richard Matheson
חיפוש לא מוצלח	Search keyword = "abcdefg"	הודעה חיובית לקפוץ "Sorry, No book found!!!"	הודעה תקפוץ למשתמש Sorry, No book found

#### Test Case for Categories

Test Case	Input	Expected Result	Test Result
גילישה בין הספרים מבוססת על קטגוריות	lezchim ul katgoriya specifit	רק הספרים ששיכים ללחיצה הקטgoriya cricim lahiot mazgim.	רק הספרים ששיכים ללחיצה הקטgoriya hozego ul masser

#### Test Case for Add to Cart

Test Case	Input	Expected Result	Test Result
Add to cart	lezchim ul cptor Add to cart lsfr mosiyim	hsper chib lhetasaf al tor uglat hnayot	hsper shahospono otmo ncnos uglat hnayot

## ממשק המשתמש/לקוח – GUI

### Register page

The screenshot shows the homepage of a book store. At the top, there's a dark header bar with "SALE UP TO 70% OFF. SHOP NOW" on the left, and "Create Account", "Login", and "Help" on the right. A shopping cart icon is also present. Below the header is a main navigation bar with categories: Home, Classic, Horror, Biography, Fiction, Comic, Science, Recommendations, Orders, and Contact. The "Home" link is highlighted. The main content area features a large orange "BOOK STORE" logo. To the left of the logo is a phone icon and the text "Order online or call us : +972559518504". To the right is a search bar with the placeholder "Search for a book..." and a magnifying glass icon.

### REGISTER HERE

The screenshot shows a registration form titled "PROFILE INFORMATION". It consists of several input fields: "First Name...", "Last Name...", "abdo" (pre-filled in the first name field), "Email Address", "...." (pre-filled in the last name field), and "Password Confirmation". Below the fields is a checkbox labeled "I accept the terms and conditions" and a blue "Register" button at the bottom.

PROFILE INFORMATION	
First Name...	
Last Name...	
abdo	
Email Address	
....	
Password Confirmation	
<input type="checkbox"/> I accept the terms and conditions	
<b>Register</b>	

## Login page

The screenshot shows the homepage of a website called "BOOK STORE". At the top, there's a dark header bar with the text "SALE UP TO 70% OFF. SHOP NOW" on the left, and "Create Account", "Login", and "Help" links on the right. A shopping cart icon is also present. Below the header, the main title "BOOK STORE" is centered in large orange letters. To the left of the title, there's a phone icon and the text "Order online or call us : +972559518504". To the right is a search bar with the placeholder "Search for a book..." and a magnifying glass icon. A navigation bar below the title contains links for Home, Classic, Horror, Biography, Fiction, Comic, Science, Recommendations, Orders, and Contact. The "Home" link is highlighted in blue. Below the navigation bar, a breadcrumb trail shows "Home / Login Page".

### LOGIN FORM

The login form is contained within a light gray box. It features two input fields: one for "Username" and one for "Password", both with black outlines. Below these fields is a small, dark blue button labeled "Login". Above the "Login" button, there is a link "Forgot Password?".

#### FOR NEW PEOPLE

[Register Here](#) (Or) go back to [Home >](#)

62

## Main page

SALE UP TO 70% OFF. [SHOP NOW](#)

Create Account   Login   Help   

Order online or call us : +972559518504

# BOOK STORE



Home   Classic   Horror   Biography   Fiction   Comic   Science   Recommendations   Orders   Contact



BUY BOOKS ARE NOW ONLINE WITH US

**TOP SELLING OFFERS**

## Books from biography category

SALE UP TO 70% OFF. [SHOP NOW](#)

Welcome Abdo   Logout   Help   

Order online or call us : +972559518504

# BOOK STORE

Search for a book... 

Home   Classic   Horror   Biography   Fiction   Comic   Science   Recommendations   Orders   Contact

 [Home](#) / Books

**CATEGORIES**

- Classic
- Horror
- Fiction
- Comic
- Science
- Biography
- Self Development
- Novel

Item on page 9 

Default sorting 

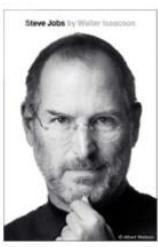


BECOMING  
הסיפור שלי  
מישל אובמה

מישל אובמה - הסיפור שלי

★ 0      \$40    \$48

[ADD TO CART](#)



Steve Jobs by Walter Isaacson

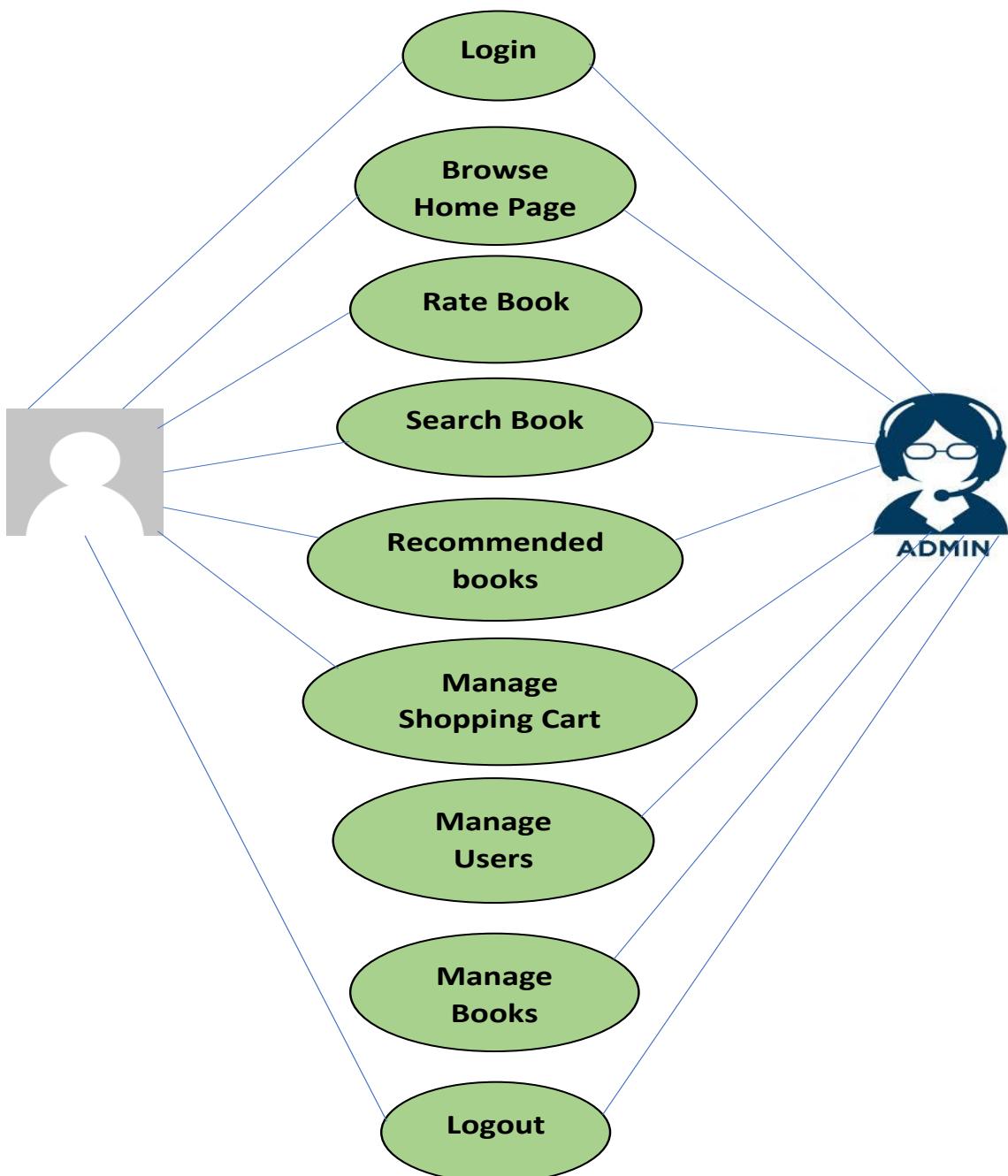
Steve Jobs

★ 4      \$20    \$0

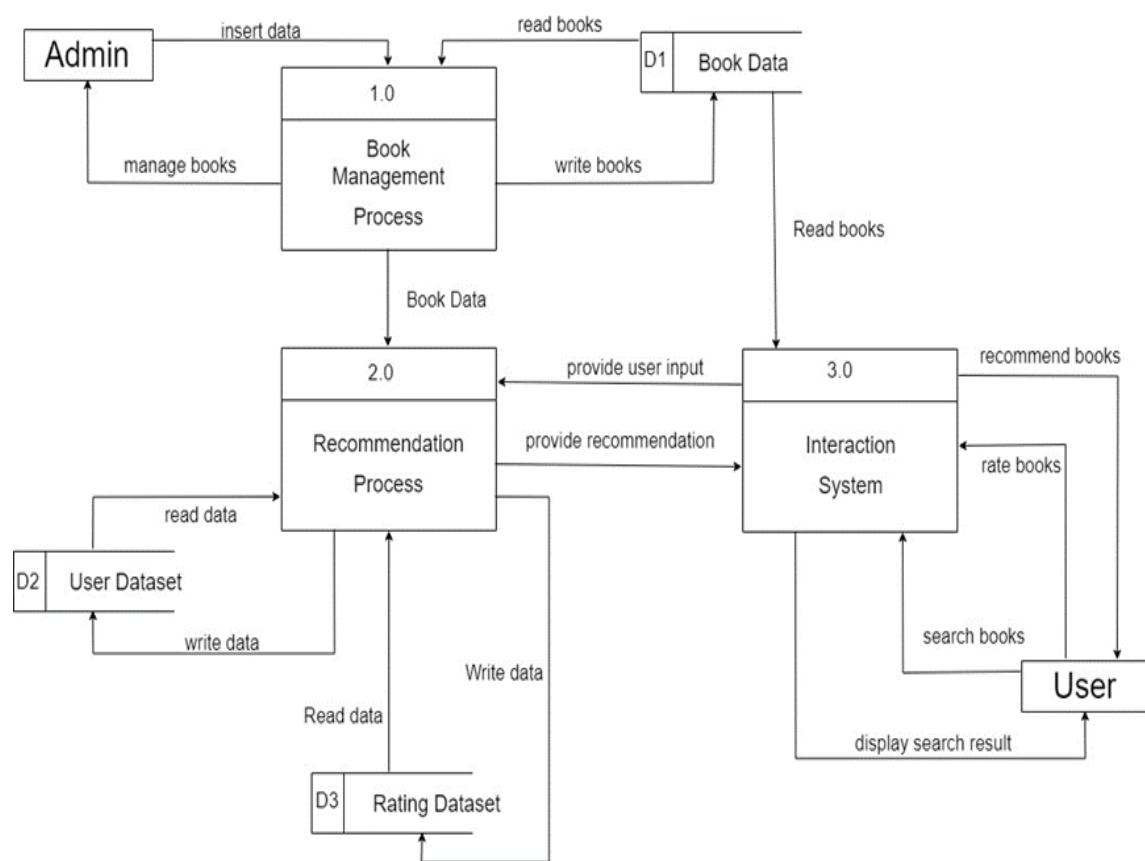
[ADD TO CART](#)

## תרשימי מערכת מרכזים

### Use Case



## Data flow



## ידע חדש שנדרש ללמידה לצורך ביצוע הפרויקט

jQuery

וחבילה Django

Django rest framework