**הכנה למימוש Good Reading**

**# תוכן ספר**

**# פורמט ספר**

**טבלאות:**

1. ספרים (PK-BID)
2. משתמשים (PK-ID)
3. לקוחות (PK&FK-ID)
4. ביקורות
5. תחום (PK-DID)
6. נושא (PK-SID. מכיל שדה תחום)
7. נושא-ספר (FK-BID, FK-SID)
8. לקוח-ספר (FK-BID, FK-ID)
9. עובדים (PK-WID, FK-ID)

**שאלה:**

**מחירים:**

30 חודשי 25%

300 שנתי 50%

**תהליכים:**

1. log in – בדיקת קלט, בדיקה סטטוס מחובר (מחובר, מנותק, חסום, ממתין לאישור), בדיקה אם הוא עובד, בדיקת תפקיד, פתיחת חלון בהתאם.
2. log out – בדיקת סטטוס חיבור, זריקה החוצה לחלון פתיחה
3. sign up – מילוי פרטים, בדיקה אם קיים, יצירת מופע משתמש חסום, עדכון ברשימת ממתינים.
4. search book – בחירה בין ספר לביקורת, קלט משתמש ע"פ קטגוריה שנבחרה (drop menu check list), הצגת תוצאות חיפוש (לביקורת יש להציג גם את פרטי הספר התואם, לספר יש להציג רק ספרים ששייכים לקטלוג ע"י בדיקת ה viewStatus שלהם, כל עוד הגיע פרמטר המעיד על חיפוש בקלוג ולא ב DB). עבור ספר, עדכון מס החיפושים לספר ע"י AddToSearchLog.
5. read review – שליפת כל הביקורות ממסד הנתונים, הצגה לפי סדר.
6. open account – קליטת סוג חשבון, יצירת אובייקט חדש של לקוח, עדכון הנתונים באובייקט כולל סטטוס חסום, שמירה במסד הנתונים.
7. buy book – קריאה ל ValidateAccount , בדיקת אם חסום הודעה ללקוח, אם לא חסום בדיקת סוג חשבון, חישוב מחיר לתשלום, אישור לקוח, קישור ספר ללקוח, עדכון מס הרכישות לספר ע"י AddToPurchaseLog, הצגת פורמטים להורדה, הורדה.
8. my books – הצגת כל הספרים שנרכשו, לחיצה על הספר, בחירת פורמט, הורדה.
9. download book – הורדה ????
10. submit review – קלט מהמשתמש, חתימת משתמש, אישור משתמש, יצירת אובייקט ביקורת, השמה של הביקורת לאובייקט, עדכון סטטוס approved=false , הכנסה למסד הנתונים, שליחת הודעה לכל העורכים המחוברים למערכת.
11. AddToSearchLog – בעת ביצוע חיפוש הפונקציה תיקרא, ותוסיף איבר (מסוג תאריך) נוסף למערך searchLog של הספר.
12. AddToPurchaseLog – בעת ביצוע רכישת ספר הפונקציה תיקרא, ותוסיף איבר (מסוג תאריך) נוסף למערך PurchaseLog של הספר, ותעדכן את ה purcaseCount.
13. InspectReview – קריאת כל הביקורת הממתינות לאישור ממסד הנתונים, קריאת פרטי הספרים (כותר ומחבר) התואמים לביקורות הנ"ל, בחירת ביקורת ע"י העורך, הצגת הביקורת ושלוש אפשרויות (פרסום- PublishReview, עריכה ומחיקה), וקריאה לפונקציה בהתאם.
14. PublishReview – משנה את שדה ה approved לחיובי.
15. PublishPartially – פותח חלון עריכה לטקסט הביקורת. הצגת 2 אפשרויות (שמירה וביטול), הצגת 2 אפשרויות (פרסום- PublishReview ומחיקה).
16. RejectReview – מחיקת הביקורת ממסד הנתונים.
17. AddNewUser – הצגת כל המשתמשים שממתינים לאישור, בחירת משתמש, שינוי סטטוס יוזר מ wait למנותק, שמירה במסד הנתונים.
18. FromUserToCostumer - הצגת כל הלקוחות שממתינים לאישור, בחירת משתמש, שינוי סטטוס יוזר מחסום לפתוח, שמירה במסד הנתונים.
19. UpdateAccountType – קריאה ל ValidateAccount, בדיקה אם חסום, אם פתוח, קליטת סוג חדש מלקוח, בדיקת סוג חשבון, אם יש מנוי - הודעת שגיאה, אם אין מנוי אז עדכון שדה \_waitingForChangeType לערך המתאים.
20. ValidateAccount – בדיקת חסום/לא חסום (accountStatus) , בדיקת סוג חשבון, בדיקת תאריך אם יש צורך, עדכון סוג אם תאריך פג והודעה ללקוח .  
    (הפונקציה מחזירה 0/1 אם חסום או לא)
21. EditCostumerAccount - הצגת כל הלקוחות שממתינים לעדכון סוג חשבון (שדה \_waitingForChangeType אינו "ריק"), בחירת לקוח, קריאה ל ValidateAccount, עדכון שדה ה \_accountType בהתאם לשדה waitingForChangeType, עדכון תאריך בהתאם (שנה/חודש), עדכון שדה waitingForChangeType ל "ריק".
22. EraseUser - הצגת כל המשתמשים, בחירת משתמש, מחיקה ממסד הנתונים.
23. ManageCatalog – הצגת כל הספרים שבקטלוג, בחירת ספר, שינוי שדה ה \_viewStatus.
24. RequestReport – חלון דוחות נפתח, יש 3 אפשרויות בחירה:  
    # רשימת ספרים שהוזמנו לכל לקוח – שאילתה שמחזירה עבור כל לקוח את רשימת הספרים שהוזמנו על ידו (יש שימוש בטבלה מקשרת בפר-לקוח)  
    # היסטוגרמה לספר שנבחר של הזמנות וחיפושים לפי תאריך – חיפוש ספר, בחירת ספר, קריאה קובץ ???????  
    # דירוג פופולריות לפי רכישות לספר שנבחר ביחס לכל הספרים או ביחס לספרים בתחום שלו – שאילתה ממוינת של ספרים לפי משתנה ה purchaseCount, בדיקת אורך הרשימה שהתקבלה ממסד הנתונים, חיפוש הספר בתוך הרשימה וקבלת האינדקס שלו בתוכה, הצגת התוצאות כך: 4/50.
25. PermissionManagement – הצגת שני לחצנים (משתמשים, חשבונות), בחירת קטגוריה, הצגת כל המשתמשים/חשבונות (לא כולל משתמשים שבמצב (wait.   
    ה view יכיל פרטים בסיסים, ואפשרות לשינוי הסטטוס של המשתמש/חשבון.   
    משתמש: מחסום למנותק ולהיפך (\_userStatus). חשבון: מחסום למאופשר ולהיפך (\_accountStatus).
26. ManageDatabase – בחירת תחום, נושא או ספר. בחירת קטגוריה לחיפוש (כותר, מחבר וכו'), לחיצה על חיפוש והפעלת הפונקציה לחיפוש ספר (מ user controller) עם משתנה המעיד על חיפוש ב DB, הצגת תוצאות החיפוש, לחיצה על הוספה *או* בחירת רשומה ולחיצה על עריכה או מחיקה.
27. AddToDB – פתיחת טופס למילוי פרטים בהתאם לפרמטר שהתקבל (תחום, נושא, ספר).

ספר: ייבוא רשימת נושאים ממסד הנתונים, הצגת טופס כולל חלון בחירה מרובה של הנושאים שיובאו, הזנת כל שדות הספר (אין צורך בתחום), בלחיצה על אישור תיעשה יצירת אובייקט ספר חדש, הוספת ספר לטבלת הספרים, הוספת רשומה לטבלה ספר-נושא.

נושא: ייבוא רשימת תחומים ממסד נתונים, הצגת טופס כולל חלון בחירה (יחידה) מהתחומים שיובאו, מילוי טופס, בלחיצה על אישור ייווצר אובייקט נושא חדש, הוספת נושא לטבלת נושאים.

תחום: הצגה ומילוי טופס פרטי תחום. הוספת תחום לטבלת תחומים.

\*בסוף, קריאה לפונקציה ManageDatabase.

1. EraseFromDB – מחיקה בהתאם לפרמטר שהתקבל(תחום, נושא, ספר) וה id:

ספר: שאילתת מחיקה של הספר ממסד הנתונים לפי ה id.

נושא: שאילתא שבודקת שהנושא לא קיים בטבלת נושא-ספר ומוחקת אותו מטבלת נושאים. אם הנושא קיים בטבלת נושא-ספר תשלח הודעה לספרן לשייך את כל הספרים לנושא אחר.

תחום: שאילתא שעוברת על טבלת נושא ובודקת שאין אף נושא עם התחום הנ"ל ומוחקת אותו מטבלת תחומים. אם התחום קיים בטבלת נושאים תשלח הודעה לספרן לשייך את כל הנושאים לתחום אחר.

\*בסוף, קריאה לפונקציה ManageDatabase.

1. EditInDB - פתיחת טופס לעדכון פרטים בהתאם לפרמטר (תחום, נושא, ספר) ול id שהתקבלו:

ספר: ייבוא רשימת נושאים ממסד הנתונים, ייבוא הספר לפי ה id, הצגת טופס שכולל את כל פרטי הספר הנוכחיים (הטופס כולל חלון בחירה מרובה של הנושאים שיובאו), עריכת שדות הספר, בלחיצה על אישור יעשה עדכון של האובייקט במסד הנתונים, הוספת/מחיקת רשומות מטבלה ספר-נושא.

נושא: ייבוא רשימת תחומים ממסד נתונים, ייבוא הנושא לפי ה id , הצגת טופס כולל חלון בחירה (יחידה) מהתחומים שיובאו, מילוי טופס עם הפרטים העדכניים, בלחיצה על אישור יעודכן הנושא בDB.

תחום: ייבוא התחום לפי ה ,id הצגה ומילוי טופס פרטי תחום עם הפרטים העדכניים. עדכון התחום ב DB.

\*בסוף, קריאה לפונקציה ManageDatabase.