



פרויקט גמר להנדסאי תוכנה

שם המכללה: המכללה למנהל

שם הפרויקט: בנק התורה

שם התלמידות: תמר אלאוף והדסה אחרר

בהנחיית: גב' מרים ויכלדר

שנת 2024

הצהרת סטודנט

שם הסטודנט: תמר אלאוף ת.ז. 209117043

שם הסטודנט: הדסה אלחרר ת.ז. 208801514

שם המכללה בה לומד הסטודנט: המכללה למנהל

אני הח"מ, מצהירה בזאת כי פרויקט הגמר וספר הפרויקט המצ"ב נעשו על ידי בלבד.

פרויקט הגמר נעשה על סמך הנושאים שלמדתי במכללה ובאופן עצמאי.

פרויקט הגמר וספר הפרויקט נעשו על בסיס הנחייתו של המנחה האישי.

מקורות המידע בהם השתמשתי לביצוע פרויקט הגמר מצוינים ברשימת המקורות המצוינים בספר הפרויקט .

אני מודעת לאחריות שהנני מקבלת על עצמי על ידי חתימתי על הצהרה זו שכל הנאמר בה אמת ורק אמת .

תאריך: 16/07/20224

חתימת הסטודנט:

אישור המנחה האישי

הריני מאשר שהפרויקט בוצע בהנחייתי, קראתי את ספר הפרויקט ומצאתי כי הוא מוכן לצורך הגשת הסטודנט להגנה על פרויקט גמר .

שם המנחה: _____ חתימה: _____ תאריך: _____

אישור ראש המגמה

הריני מאשר שספר הפרויקט מוכן לצורך הגשת הסטודנט להגנה על פרויקט הגמר .

שם ראש המגמה: _____ חתימה _____ תאריך: _____

תוכן העניינים

4	הצעת פרויקט
5	הגדרת דרישות ותאור כללי
6	תיאור חומרת מערכת
6	תיאור תוכנת מערכת
8	תרשים המערכת המרכזי
11	תיאור מסכים והסבר
18	מבנה נתונים
25	תכנון
25	פונקציות עיקריות
29	מחלקות
30	בקרת המערכת
30	אבטחת מידע
30	מה הקנה הפרויקט
30	ביבליוגרפיה

הצעה לפרויקט גמר**א. פרטי הסטודנט**

שם הסטודנט	ת.ז. 9 ספרות	כתובת	טלפון נייד	תאריך סיום הלימודים
תמר אלאוף	209117043	שדרות האמוראים 69 בית שמש	053- 4716342	2024
הדסה אחרר	208801514	משך חכמה 96 מודיעין עילית	052- 7270467	2024

שם המכללה: המכללה למנהל סמל המכללה: 72213

מסלול ההכשרה: הנדסאים

מגמת לימוד: תכנות מקום ביצוע הפרויקט: מכללה

ב. פרטי המנחה האישי

שם המנחה *	כתובת	טלפון נייד	תואר	מקום עבודה/תפקיד

* עבור מנחה אישי חדש יש לצרף קורות חיים, ניסיון מקצועי ותעודות השכלה לאישור מה"ט.

ג. שם פרויקט הגמר

בנק התורה

ד. מטרת פרויקט הגמר

יצירת קהילה והתאמת חברותות בין משתמשי האתר בהתאמה להגדרת העדפות וקביעה ביומן, לאפשר האזנה וצפייה בשיעורי תורה, פנייה לרבנים ואפשרות להעלאה של שיעורים עצמיים, אפשרות לתרומה לרב ספציפי לקידום הפעילות.

ה. ספרות עזר

מקורות מידע באינטרנט,
קורס עזר בריאקט

ו. תיאור הפרויקט

ו.1. תיאור:

המשתמשים יוכלו לקבוע חברותות ללימוד מתוך מאגר חברים שנרשמו למיזם (ע"פ העדפות של מיקום, גיל ונושאי לימוד) וכן לשבץ אירועי חברותא ביומן בנושאים שונים בטווחי תאריכים ושעות לפי בחירה. במערכת קיימת אפשרות לתרומה לרב, לתמיכה בפעילות וניתן לראות בכל זמן את היסטוריית התרומות ולהנפיק קבלות. משתמש יוכל לפנות לרב ע"פ בחירה בשאלות הלכתיות או תורניות ויקבל מענה אישי מהרב, כמו"כ תהיה למשתמש אפשרות לצפות בכל היסטוריית הפניות.

המערכת תאפשר לאנשים פרטיים ו/או לרבנים להפיץ את שיעורי התורה שלהם (לאנשים פרטיים בכפוף לאישור מנהל האתר), בנוחות מירבית וגישה מהירה. המשתמשים יוכלו לשמור שיעורים לצפייה או האזנה חוזרת או עתידית.

משתמש המוגדר כרב יוכל בנוסף לכל האפשרויות של משתמש "רגיל" גם לענות לפניות שהופנו אליו. למנהל האתר, בנוסף לאופציות שלעיל, תהיה אפשרות לאשר שיעורים שמשתמשים "רגילים" העלו, להוסיף רבנים למאגר, להנפיק דוחות מכל מאגרי הנתונים הקיימים ולנהל הרשאות משתמשים.

ו.2. פרטיות וגישה לנתונים:

למרות שמשתמש "רגיל" מורשה להעלות שיעורים לאתר, השיעורים אינם זמינים לצפייה/האזנה ללא אישור מנהל.

ניהול מאגר הרבנים יתאפשר למשתמש מנהל בלבד.

משתמשים בפעילות חברותא יכולים לשלוח בקשת חברותא אך אינם חשופים לכל פרטי המשתמש אלא לאחר אישור הבקשה ע"י המשתמש שאליו נשלחה הבקשה.

ו.3. סקירת מצב קיים בשוק:

מבדיקת המצב בשוק מצאנו תוכנות שונות לניהול מאגרי שיעורי תורה חינמיים. הן מיועדות בעיקר לצפייה והאזנה לשיעורי תורה, לא מצאנו תוכנה כזו אשר מהווה פלטפורמה בטוחה עבור המשתמש לקביעת חברותות. האתרים השונים המאפשרים תרומה לא מציגים למשתמש את היסטוריית התרומות במסך אחד וכן מלבד בעת ביצוע התרומה אין למשתמש אפשרות להנפיק בכל עת מחדש את הקבלות. וכן לא קיימת אפשרות לפניה דיסקרטית לרב ע"פ בחירה וללא פרסום השאלה והמענה לקהל הרחב.

ו.4. יתרונות המערכת המוצעת:

המערכת מציגה למשתמש את כל היסטוריית הפעילות ומאגדת תחת פלטפורמה אחת את הכלים ללימוד מיטבי ויעיל ומעשירה אותו בחיי קהילה פוריים.

למשתמש ישנה אפשרות לשאול באופן פרטי רב לבחירתו, לתרום לרב לבחירתו, לקבוע חברותות בהתאם להעדפותיו וכן לראות בכל זמן נתון את שיבוצי החברותות ביומן, ולשלוח את נתוני התרומות כולל אסמכתא בכל זמן.

ו.5. שימוש במבני נתונים וארגון קבצים:

הנתונים יאוכסנו במסד נתונים של SQL Server.
SQL Server בנוי לתמיכה במס' גדול של קריאות בו זמנית ואין חשש לקריסה ברמה של הפרוייקט.

6.1. דרישות המערכת:

מחשב PC מערכת הפעלה WIN10 ומעלה.

7.1. טכנולוגיות בשימוש:

Web Api . Net Core

React

8.1. שפות הפיתוח:

(C#) Web Api

JavaScript

9.1. משימות המערכת:

משימות המערכת מכילות אפשרויות שונות:

- צור קשר (כתיבת הודעות למשרד)
- צפייה בשיעורי רבנים/משתמשים
- צפייה במידע – תרומות/שאלות/חברות
- פתיחת חברותא חדשה
- צפייה בבקשות שנשלחו אלי
- הרשמה לאזור האישי (גם ללקוח וגם למנהל)
- כניסה לאזור האישי
- צפייה בפרטים אישיים
- אפשרות עדכון פרטים אישיים
- צפייה בשיעורים ששמרתי
- תרומה לרב
- אפשרות הדפסה והורדה של התרומות שלי
- התנתקות מהאזור האישי

10.1. תיאור חומרת מערכת

כללי:

המערכת תעבור קומפילציה והפצה בסביבת Visual Studio , Visual Studio Code , React .
המשתמש יוכל להריץ את האתר בכל מכשיר אשר מותקן עליו דפדפן אינטרנט.

מרכיבי המערכת:

מחשב, עכבר ומקלדת

מחשב שרת לפי דרישות ספק שירותי אינטרנט.

11. תיאור תוכנת המערכת

א. כללי

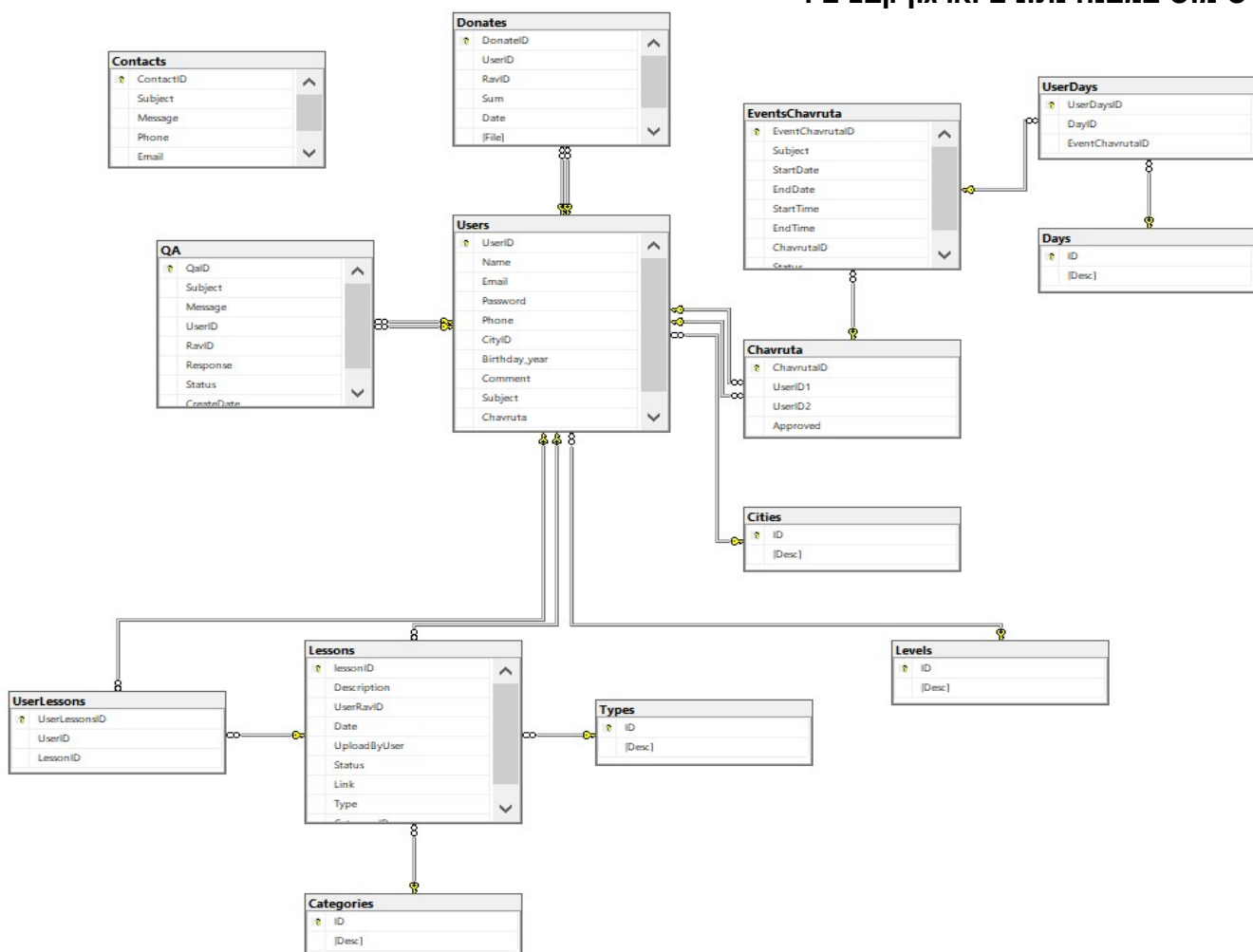
הפרויקט הינו אתר המאפשר צפייה וביצוע פעולות בהקשר של המיזם – צפייה בשיעורים, קביעת חברות ניהול יומן

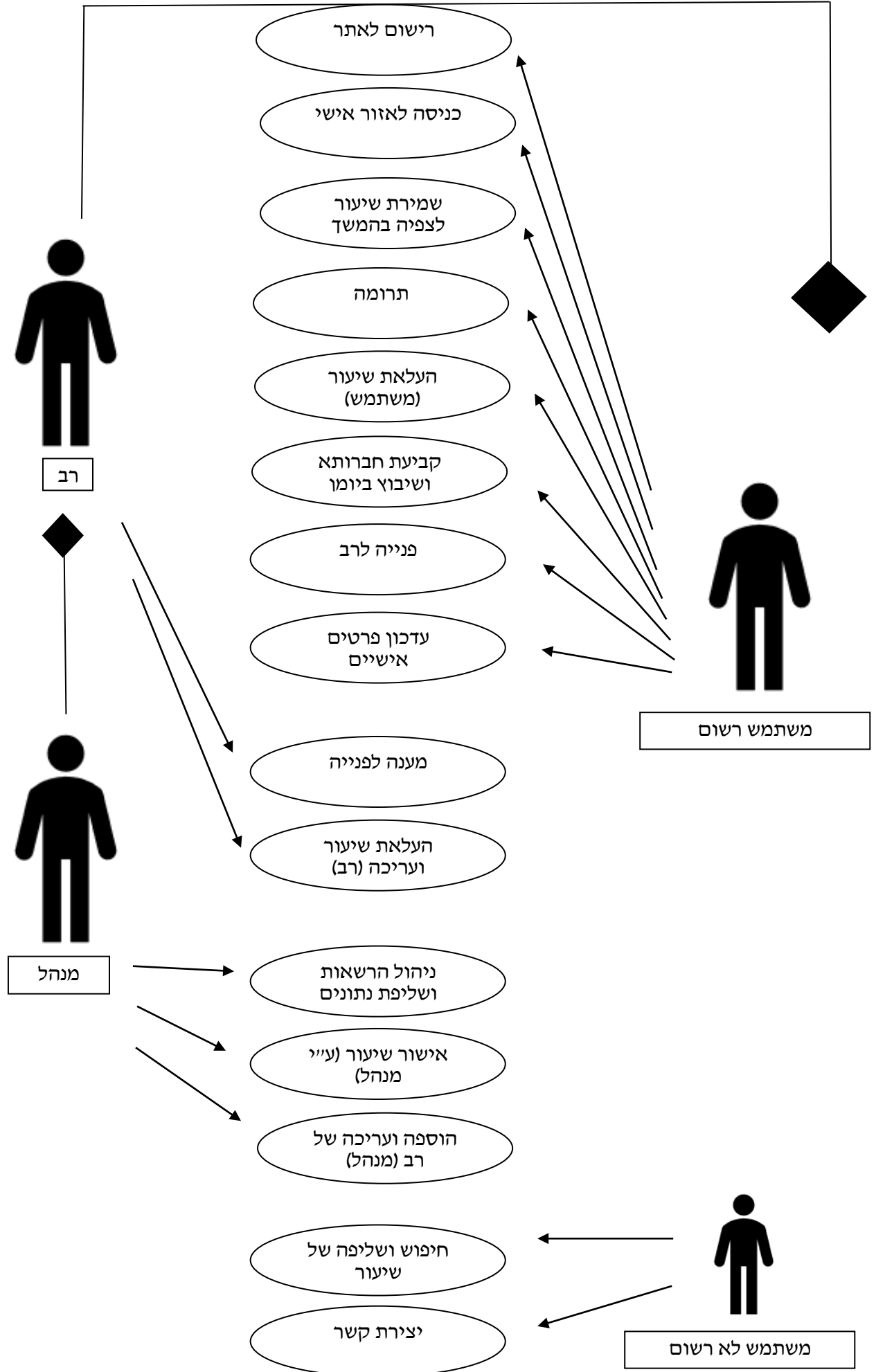
המנהל - יוכל לצפות בנתוני כלל המשתמשים הרשומים, לאשר שיעורים ומענה לפניות.
משתמש – יוכל לבצע מגוון פעולות בכל הקשור לנתוניו האישיים, מאגר החברות והחברות שנקבעו, כמו"כ לצפות בשיעורים ולתמוך ברבנים

ב. כלי התוכנה לפיתוח המערכת

- שפות תכנות: C#, HTML, SCSS, React
- Visual studio 2019
- Visual studio code
- SQL server

12. שימוש במבנה נתונים וארגון קבצים :





14. מה יקנה הפרוייקט לסטודנט :

- התמצאות בביתוח מערכות.
- הכרות מעמיקה עם API
- מיומנות בטכנולוגית React
- מיומנות באופן פניית שרת ללקוח.

15. שלבי פיתוח הפרוייקט :

- למידת הנושאים הרלוונטים.
- תכנון וביתוח המערכת.
- איפיון מפורט.
- כתיבת הקוד.
- בדיקה ותיקונים.

16. ארכיטקטורה נבחרת:

הארכיטקטורה הנבחרת היא חלוקה ל-3 שכבות Tier Architecture
בפרוייקטים שמבוססים על data base מומלץ להשתמש בגישה של מודל השכבות, בארכיטקטורה זו קיימת הפרדה בין השכבות השונות בפרוייקט.
היתרון העיקרי בחלוקה לשכבות הוא תחזוקה קלה במיוחד במערכות גדולות, כך שניתן יחסית בקלות להחליף כל שכבה בלי לגעת בשכבות האחרות כאשר נרצה לעשות שינויים בתכנית.

17. חלוקה לתוכניות ומודלים:

1. **שכבת תצוגה - (Layer Presentation)** שכבת ממשק המשתמש (User Interface)
שכבה זו מורכבת מפרוייקט ריאקט המציג את דפי Html.
השכבה מתקשרת בין המשתמש לבין שאר השכבות של המערכת React.
2. **שכבת הלוגיקה העסקית – (Logic Business-BL)**
השכבה שאמונה על הלוגיקה של המערכת, עוסקת בעיבוד המידע, בחישובים שונים ושליחתם לשכבת התצוגה.
בשכבה זו נממש את הפונקציונאליות של המערכת – WebApi – Controller.
3. **שכבת הנתונים – (DAL – Data Access Layer)**
שכבה זו מורכבת ממקור נתונים – מסד הנתונים שלנו, וממערכת תוכנה Entity framework אשר תפקידה לקרוא את המידע הנדרש למערכת, לשמור את העדכונים, ולהוסיף מידע חדש או למחוק פרטי מידע קיימים - Model Entities Framework.

18. סביבת שרת :

לצורך הפרוייקט נשתמש בשרת מקומי IISExpress המסופק עם סביבת העבודה של visual studio. אם האתר יירכש ע"י לקוח נעלה אותו לשרת אירוח כלשהו או Microsoft azure.

19. ממשק המשתמש/ לקוח GUI :

שכבת ה GUI מורכבת מדפי HTML שמוצגים למשתמש דרך הדפדפן.

20. שימוש בחבילות תוכנה:

21.1. אופציות להרחבה:

התממשקות עם יומן גוגל, הודעות push עבור אובייקטים שטרם נקראו

22.1. משאבים הנדרשים לפרוייקט:

↩ מס' שעות המוקדשות לפרוייקט: 700 שעות שיחולקו שווה בין שתי בנות הזוג.

↩ ציוד נדרש: מחשב ורשת.

↩ תוכנות נדרשות:

Visual Studio 2022

Visual Studio Code

SQL Server

Ils express

דפדפנים

23.1. תכנית עבודה ושלים למימוש הפרוייקט:

↩ אפיון ותכנון.

↩ כתיבת צד שרת, כתיבת צד לקוח, עיצוב צד לקוח.

↩ בדיקות של התוכנה.

↩ כתיבת ספר פרוייקט.

24.1. בקרת גרסאות (version control):

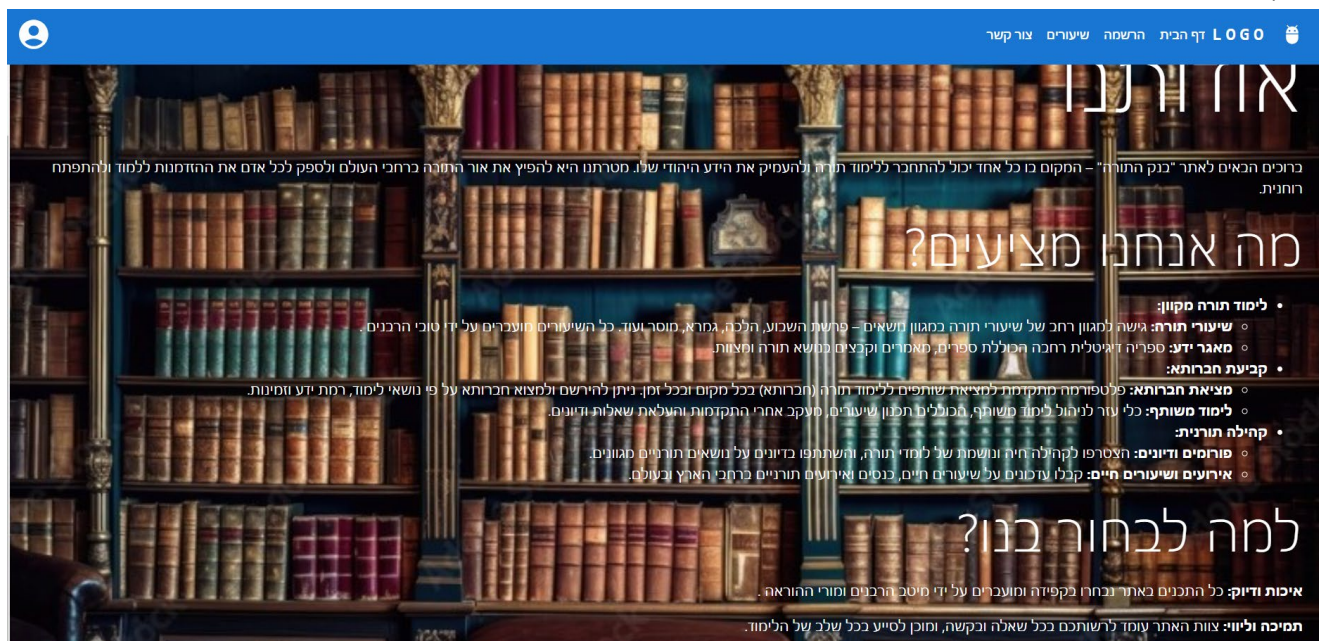
הפרוייקט נוהל במערכות של GITHUB לצורך תחזוקה, פיתוח ושיתוף קוד.

מתודולוגית הפיתוח כללה code reviews של הקוד שהובנס.

25. תיאור המסכים:

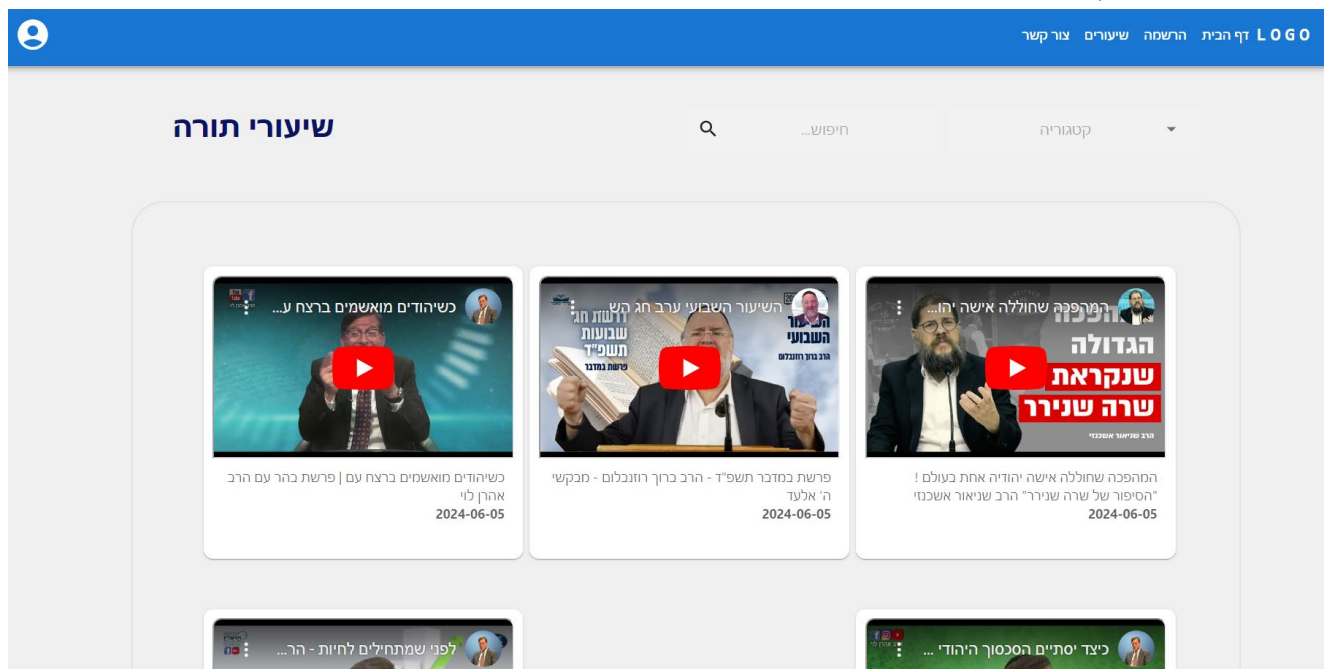
מסך ראשי: מסך הבית

ממסך זה המשתמש יוכל ללמוד על מטרות המיזם ולנווט באמצעות התפריט בראש העמוד



מסך שיעורים:

ממסך זה ניתן לצפות בשיעורים באמצעות סינון וחיפוש (כאשר המשתמש מחובר יש כפתור להוספה למועדפים - שיעורים שמורים)



יצירת קשר:

פניה בכל נושא למנהל האתר

טופס יצירת קשר

נושא *

הודעה *

טלפון *

דואר אלקטרוני *

שליחה

מסך הרשמה:

רישום משתמש חדש

הרשמה

דואר אלקטרוני *

שדה חובה

סיסמא *

שדה חובה

שם מלא *

שדה חובה

טלפון *

שדה חובה

שנת לידה *

שנת לידה חייבת להיות בדיוק 4 ספרות

☐ חברותא

הרשמה

מסך התחברות:
התחברות משתמש קיים

כאן התחברות

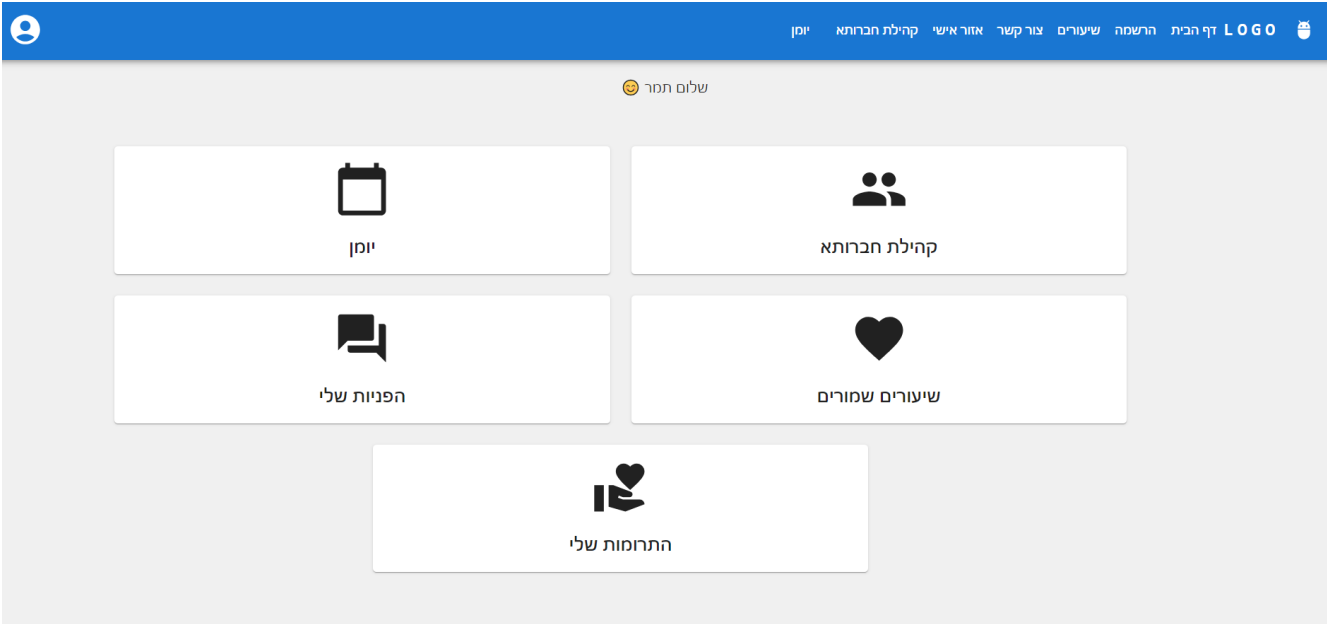
Email

Password

התחברות

[להירשם כאן](#) אין לך חשבון?

אזור אישי משתמש:



כניהול אזור אישי חברותא:
ממסך זה משתמש רשום יוכל לאשר חברותות ביומן וכן לצפות בחברותות שקבע

LOGO

דף הבית

הרשמה

שיעורים

צור קשר

אזור אישי

קהילת חברותא

יומן

שלום תמר

לחיפוש חברותא

החברותות שלי

אשר

נושא: תנ"ך
טלפון: 0534716342
אימייל: n@g
הערות:

משה

נושא: דף היומי
טלפון: 532659874
אימייל: m@g
הערות:

הזמנות לחברותא הממתינות לאישורי

יעקב

נושא: הדף היומי
טלפון: 0562121212
אימייל: gh@h
הערות:

סטטוס: לא מאושר

לאישור החברותא

לדחיית החברותא

צפייה במאגר משתמשי חברותא ושליחת בקשה

LOGO

דף הבית

הרשמה

שיעורים

צור קשר

אזור אישי

קהילת חברותא

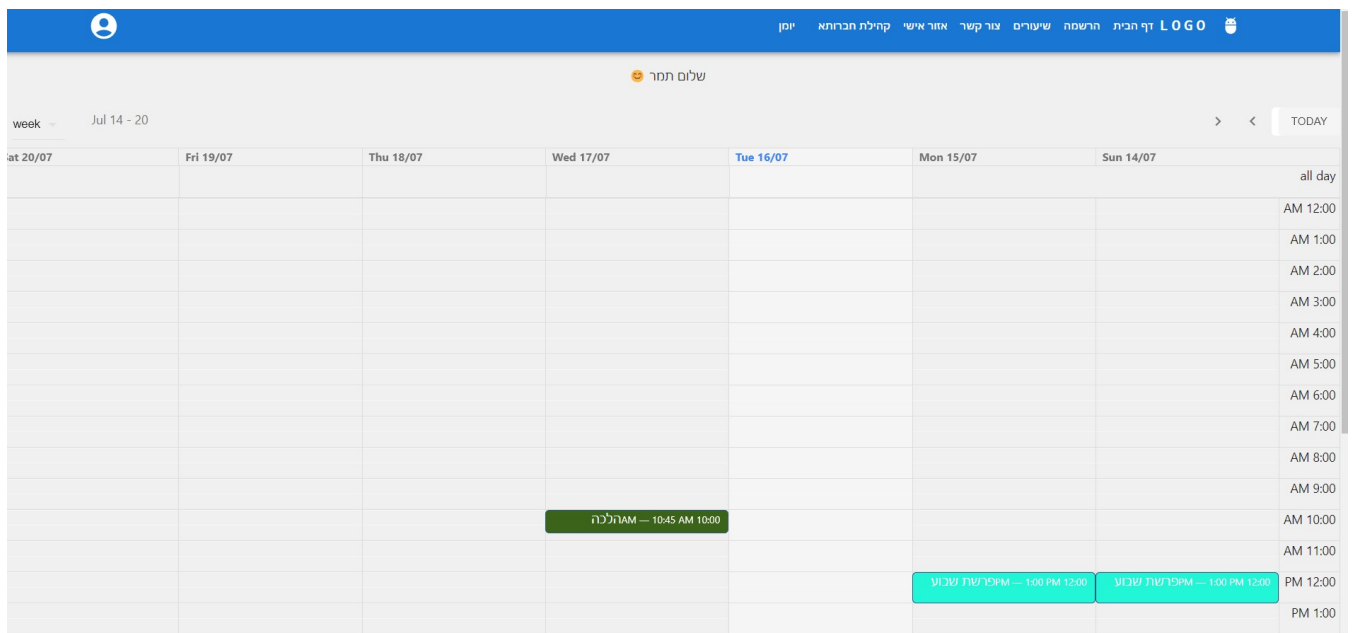
יומן

שלום תמר

אופציות לחברותא

שם	גיל	נושא	הערות	עיר	
לוי	24	דף היומי	zxcv	jerusalem	הזמנה לחברותא
שמעון	24	גמרא בעיון	111	jerusalem	הזמנה לחברותא
זבולון	24	גמרא בבקאות	222	jerusalem	ממתין לאישור
יששכר	24	משניות	3333	jerusalem	הזמנה לחברותא
אפרים	24	שבת, זכאים	444	jerusalem	הזמנה לחברותא
מתתיהו	27	נדיקין	555	jerusalem	הזמנה לחברותא
...	הזמנה לחברותא

ניהול אזור אישי יומן: צפייה באירועי חברות ששובצו ביומן



אזור אישי שיעורים שמורים: אפשרות לצפייה בשיעורים שהמשתמש שמר

אזור אישי פניות לרב: אפשרות לשלוח שאלה לרב וקבלת מענה וכן צפייה בהיסטורית השאלות והתשובות

תרומות:

אופציה לתרומה לרב וכן צפייה בהיסטוריית התרומות

יומןקהילת חברותאזור אישיצור קשרשיעוריםהרשמהדף הביתLOGO

שלום תמר 😊

החזקת תורה

התרומות שלי

שם רב: משה סכום: 6000 תאריך: 2024-06-24 קבלה: string	שם רב: משה סכום: 5000 תאריך: 2024-06-24 קבלה: string	שם רב: משה סכום: 1500 תאריך: 2023-01-12 קבלה: images
שם רב: משה סכום: 9000 תאריך: 2024-06-24 קבלה: string	שם רב: משה סכום: 9000 תאריך: 2024-06-24 קבלה: string	שם רב: משה סכום: 7000 תאריך: 2024-06-24 קבלה: string
שם רב: משה סכום: 10000 תאריך: 2024-06-24 קבלה: string	שם רב: משה סכום: 1000 תאריך: 2024-06-24 קבלה: string	שם רב: משה סכום: 10000 תאריך: 2024-06-24 קבלה: string

יומןקהילת חברותאזור אישיצור קשרשיעוריםהרשמהדף הביתLOGO

שלום תמר 😊

הוספת תרומה חדשה

* רב

סכום *

תאריך

2024-07-16

שלח תרומה

צפייה בקבלה עבור התרומה

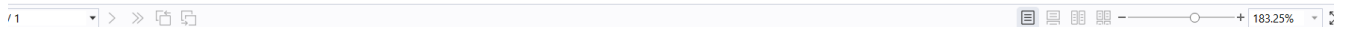
קבלה בגין תרומה מס' 49

לכבוד התורם/ת: תמר

שם הרב: מתתיהו

סכום התרומה: 545

תאריך התרומה: 15/07/2024



מבנה נתונים:

ארגון קבצים - כללי

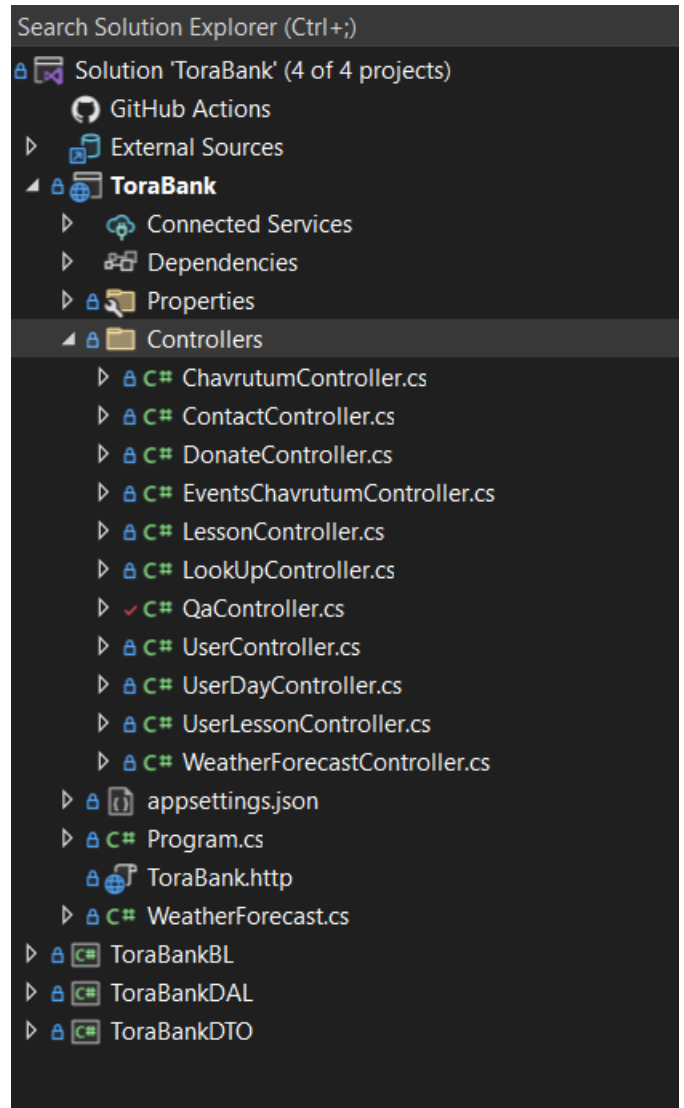
המערכת מתבססת על שרת השואב את הנתונים מבסיס נתונים של SQLserver

ניתן לגלוש לאתר מכמה מוקדים בו זמנית.

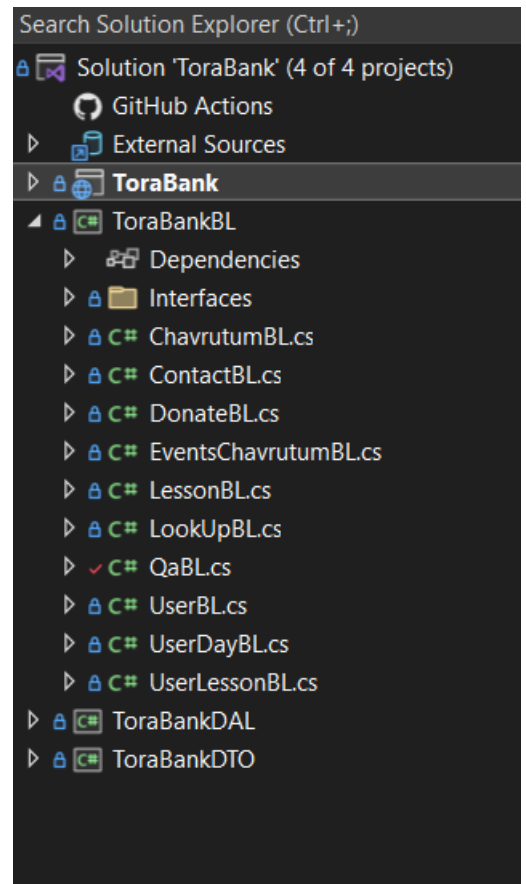
בצד השרת מתבצעות העיבוד, השליפות של הנתונים והלוגיקה, ובצד לקוח מתבצעת התצוגה וחלק מן הלוגיקה.

להלן תמונת ארגון הקבצים של הפרויקט:
צד השרת:

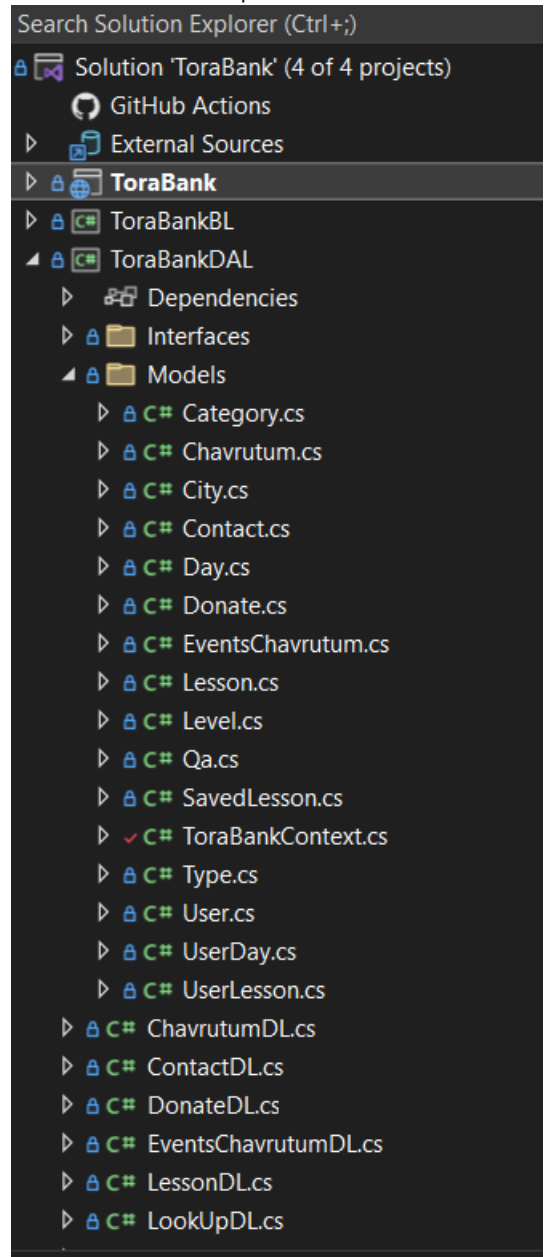
התיקיה Controllers מכילה את ה- Controllers שמהווים את החיבור בין שכבת ה- UI לשכבת ה- BL.



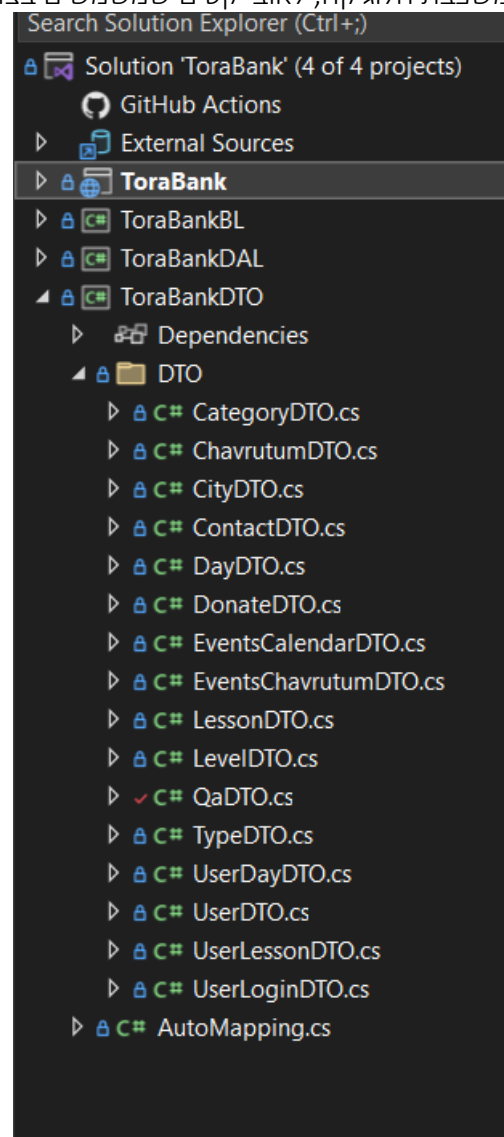
שכבת ה-BL - השכבה מכילה פונקציות ולוגיקה.



שכבת ה DAL מכילה את קבצי ה- EF המחברים את מסד הנתונים למערכת.

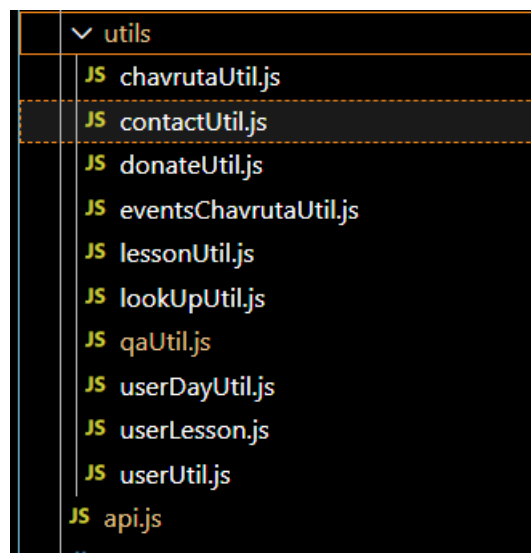


שכבת ה DTO- מכילה את קבצי ה MAPPER, בהם מוגדר המיפוי מאובייקטים משכבת הלוגיקה, לאובייקטים שמשמשים בצד ה UI.

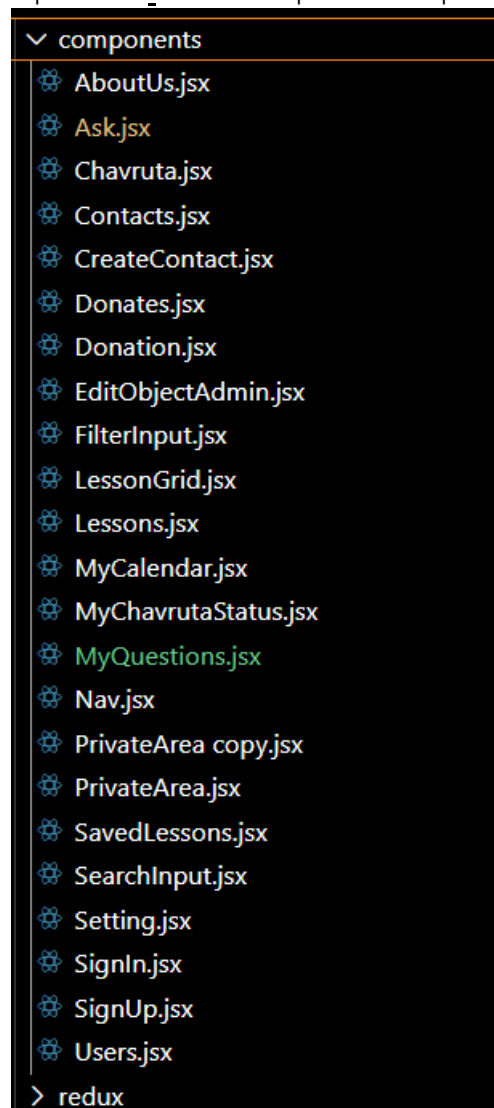


צד ה־UI:

התיקיה `utils` מקבילה למחלקות בצד שרת ומופיעים בו כל הפונקציות בהתאם לפונקציות הנמצאות בו בצד השרת.



התיקיה `Components` מכילה את כל הקומפוננטות



המערכת מתבססת על מסד נתונים המורכב מטבלאות המכילות מידע אודות משתמשים רשומים, הרשאות, שיעורים, חברות, תרומות ושאלות ותשובות.

אתר מקושר למסד הנתונים באמצעות EF, כל ישות הינה טבלה המתורגמת למחלקה בצד השרת.

The diagram illustrates a database schema with the following tables and their attributes:

- Contacts**: ContactID (PK), Subject, Message, Phone, Email
- Donates**: DonateID (PK), UserID, RaviID, Sum, Date, [File]
- Users**: UserID (PK), Name, Email, Password, Phone, CityID, Birthday_year, Comment, Subject, Chavruta
- QA**: QaID (PK), Subject, Message, UserID, RaviID, Response, Status, CreateDate
- EventsChavruta**: EventChavrutaID (PK), Subject, StartDate, EndDate, StartTime, EndTime, ChavrutaID, Status
- Chavruta**: ChavrutaID (PK), UserID1, UserID2, Approved
- Cities**: ID (PK), [Desc]
- UserDays**: UserDaysID (PK), DayID, EventChavrutaID
- Days**: ID (PK), [Desc]
- UserLessons**: UserLessonID (PK), UserID, LessonID
- Lessons**: LessonID (PK), Description, UserRaviID, Date, Link, Type
- Types**: ID (PK), [Desc]
- Categories**: ID (PK), [Desc]
- Levels**: ID (PK), [Desc]

The relationships between the tables are as follows:

- Users** is the primary entity for **Contacts**, **Donates**, **QA**, **EventsChavruta**, **Chavruta**, **Cities**, **UserLessons**, **Lessons**, and **Categories**.
- EventsChavruta** is the primary entity for **UserDays**.
- Chavruta** is the primary entity for **EventsChavruta**.
- Lessons** is the primary entity for **UserLessons**.
- Lessons** is the primary entity for **Types**.
- Lessons** is the primary entity for **Categories**.
- Lessons** is the primary entity for **Levels**.

ברשת קיימים אתרים שונים שמטרתם לתמוך בהפצת תורה ויהדות, אך יצרנו מעטפת ששמה דגש על מבנה קהילתי, פיתוח ועידוד קשרי למידה תוך שמירה על פרטיות המשתמש והנתונים האתר מוגש לכל משתמש וכל גולש יוכל לקחת חלק במיזם ולתרום בהתאם להעדפותיו המערכת חדשנית בכך שמציעה לכל משתמש פלטפורמה להפצת שיעורים או חומר תורני, כל התכנים באתר נבחרו בקפידה ומועברים על ידי מיטב הרבנים ומורי ההוראה או באישור מנהל האתר. פלטפורמה מתקדמת למציאת שותפים ללימוד תורה (חברותא) בכל מקום ובכל זמן. ניתן להירשם ולמצוא חברותא על פי נושאי לימוד, רמת ידע וזמינות. המערכת מציגה בצורה נוחה את פריסת שיבוצי החברותא ביומן (יומי/שבועי/חודשי – ע"פ העדפה) כמו"כ המערכת מציעה דיסקרטיות מלאה ואינה חושפת פרטים אישיים כגון טלפון ומייל הפניות מועברות בצורה פרטית לרבנים שמשיבים על שאלות מקצועיות, ללא חשיפת המידע באתר לציבור המשתמשים. המשתמש יוכל לשלוף את כל הקבלות עבור תרומותיו בכל עת משום שהן נשמרות בענן ואין לו צורך לתייק אותן באחסון פרטי.

עקרונות התכנות:

צד שרת: HTML5, SCSS, C#, Web API

צד לקוח: React

בפרויקט הושם דגש על יעילות הקוד ושלילת נתונים רלוונטיים לכל פונקציה.

פונקציות עיקריות

פונקציית הרשמה לאתר

```
2 references
public async Task<User> AddUser(User user)
{
    try
    {
        if (user.CityId == -1)
        {
            user.CityId = null;
        }
        _toraBankContext.Users.AddAsync(user);
        _toraBankContext.SaveChanges();
        User newUser = await _toraBankContext.Users.OrderByDescending(item => item.UserId).FirstOrDefaultAsync();
        return newUser;
    }
    catch (Exception ex)
    {
        throw ex;
        return null;
    }
}
2 references
```

פונקצייה לשליפת היוזרים הרשומים לפרוייקט חברותא ללא המשתשים שלמשתמש המחובר כבר נוצרה חברותא ואושרה

```
2 references
public async Task<List<User>> GetUsersByChavruta(int id)
{
    List<User> users = await _toraBankContext.Users.Where(user => user.Chavruta == true && user.UserId != id)
        .Include(u => u.ChavrutumUserId2Navigations)
        .Include(u => u.ChavrutumUserId1Navigations)
        .ToListAsync();
    // מכל רשימה של חברותות
    foreach (var user in users)
    {
        user.ChavrutumUserId1Navigations = user.ChavrutumUserId1Navigations
            .Where(c => c.UserId2 == id)
            .ToList();

        user.ChavrutumUserId2Navigations = user.ChavrutumUserId2Navigations
            .Where(c => c.UserId1 == id)
            .ToList();
    }

    // פילטרנו מי שיש לנו איתו חברותא מאושרת
    var filteredUsers = users
        .Where(user => !user.ChavrutumUserId1Navigations.Any(c => c.Approved) && !user.ChavrutumUserId2Navigations.Any(c => c.Approved))
        .ToList();

    return filteredUsers;
}
```

פונקציה לשליפת כמות מוגבלת של שיעורים בכל טעינה

```
2 references
public async Task<(List<LessonDTO>, bool)> GetLessonsByPage(int page)
{
    try
    {
        int pageSize = 18;
        int skipCount = (page - 1) * pageSize;
        // Retrieve books from the repository based on skipCount and pageSize
        (List<Lesson> lessons, bool hasNext) = await _lessonDL.GetLessonsByPage(skipCount, pageSize);

        List<LessonDTO> lessonDTOs = _mapper.Map<List<LessonDTO>>(lessons);

        return (lessonDTOs, hasNext);
    }
    catch (Exception ex)
    {
        // Handle exceptions or log them as needed
        Console.WriteLine(ex.ToString(), "GetLessonsByPage in LessonBL");
        return (null, false); // Propagate the exception to the controller for centralized error handling
    }
}
```

2 references

פונקציה לשליפת שיבוצי חברותות של המשתמש המחובר לצורך הצגה ביומן

```

public async Task<List<EventsCalendarDTO>> GetEventsChavrutumById(int userId)
{
    List<EventsChavrutum> EventsChavrutum = await _eventsChavrutumDL.GetEventsChavrutumById(userId);
    List<EventsCalendarDTO> Events = new List<EventsCalendarDTO>();

    string randomColor = GetRandomColor();

    int count = 0;
    foreach (var item in EventsChavrutum)
    {
        DateOnly date = item.StartDate;
        while (date <= item.EndDate)
        {
            if(item.UserDays.Select(x => x.Day.Id).Contains((int)date.DayOfWeek + 1)){
                EventsCalendarDTO itemEvent = new EventsCalendarDTO()
                {
                    Id = count++,
                    Title = item.Subject,
                    Start = date.ToDateTime(item.StartTime),
                    End = date.ToDateTime(item.EndTime),
                    BgColor = randomColor
                };
                Events.Add(itemEvent);
            }
            date = date.AddDays(1);
        }
    }
    return Events;
}

```

פונקציה לשמירת פרטי משתמש מחובר באמצעות session storage (ריאקט)

```

import { createSlice } from "@reduxjs/toolkit";

const initialValue = {
    // loggedUser : sessionStorage.getItem('user') ? JSON.parse(sessionStorage.getItem('user')) : null,
    // loggedUser : JSON.parse(sessionStorage.getItem('loggedUser'))
};

const userSlice = createSlice({
    name: "user",
    initialState: initialValue,
    reducers: {
        setLoggedInUser: (state, action) => {
            state.loggedUser = action.payload;
            sessionStorage.setItem('user', JSON.stringify(action.payload));
        },
        setUnLoggedInUser: (state, action) => {
            state.loggedUser = action.payload;
            sessionStorage.removeItem('user');
        },
        setAddSavedLesson: (state, action) => {
            state.savedLessons.push(action.payload); // הוספת שיעור למערך השיעורים השמורים
        },
        setRemoveSavedLesson: (state, action) => {
            state.savedLessons = state.savedLessons.filter(lessonId => lessonId !== action.payload); // השיעורים השמורים
        }
    }
});

export const { setLoggedInUser, setUnLoggedInUser, setAddSavedLesson, setRemoveSavedLesson } = userSlice.actions;
export default userSlice.reducer;

```

פונקציות הניווט הראשית המציגה למשתמש מחובר או למבקר את התפריט ע"פ הרשאותיו

```

const pages = [{ name: "דף הבית", url: 'homePage' }, { name: 'הרשמה', url: 'signUp' }, { name: "שיעורים", url: 'lessons' }, { name: "אזור אישי", url: 'privateArea' }, { name: "קהילת חברותא", url: 'privateArea/myChavrutaStatus' }, { name: 'יומן', url: 'myCalendar' }];
const settingsConnected = [{ name: "עדכון פרטים", url: 'setting' }, { name: 'התנתקות', url: 'logout' }];
const settingsUnConnected = [{ name: 'התחברות', url: 'signIn' }];

const Nav = () => {
  const loggedUser = useSelector(state => state.user.loggedUser);
  const [anchorElNav, setAnchorElNav] = useState(null);
  const [anchorElUser, setAnchorElUser] = useState(null);
  const [settings, setSettings] = useState(loggedUser ? settingsConnected : settingsUnConnected);
  const dispatch = useDispatch();

  useEffect(() => {
    if (loggedUser == null) {
      setSettings(settingsUnConnected);
    }
  }, [loggedUser]);

  const navigate = useNavigate();

  const handleNavigate = (setting) => {
    let permission = ["lessons", "createContact", "homePage", "signUp", "signIn"];
    if (loggedUser == null && !permission.includes(setting.url)) {
      navigate('notFound');
    } else {
      if (setting.url === "logout") {
        const confirmed = window.confirm("האם אתה בטוח/ה שברצונך להתנתק?");
        if (confirmed) {
          dispatch(setUnLoggedUser(null));
          navigate('/homePage');
        }
      } else {
        navigate(setting.url);
      }
    }
  };

  const handleOpenNavMenu = (event) => {
    setAnchorElNav(event.currentTarget);
  };
  const handleOpenUserMenu = (event) => {
    setAnchorElUser(event.currentTarget);
  };

  const handleCloseNavMenu = () => {
    setAnchorElNav(null);
  };

  const handleCloseUserMenu = () => {
    setAnchorElUser(null);
  };

  const userPages = loggedUser ? [...pages,
    { name: 'אזור אישי', url: 'privateArea' },
    { name: 'קהילת חברותא', url: 'privateArea/myChavrutaStatus' },
    { name: 'יומן', url: 'myCalendar' }
  ] : pages;

```

מחלקות – פירוט המחלקות ומאפייניהן

LookUp (Category, Day, City, Level, Type)

Id, Desc

Chavrutum

Chavrutaid, UserId1, UserId2, Approved

Contact

ContactId, Subject, Message, Phone, Email

Donate

DonatId, UserId, RavId, Sum, Date, File

EventsChavrutum

EventChavrutald, Subject, StartDate, EndDate, StartTime, EndTime, Chavrutald, Status

Lesson

LessonId, Description, UserRavId, Date, UploadByUser, Status, Link, Type, CategoryId

Qa

Qald, Subject, Message, UserId, RavId, Response, Status, CreateDate

User

UserId, Name, Email, Password, Phone, BirthdayYear, Comment, Subject, Chavruta, LevelId

UserDay

UserDaysId, DayId, EventChavrutald

UserLesson

Id, LessonId, UserId

בקרת המערכת

עבור שדות קלט יש בדיקה האם הערך המוזן עונה על הדרישות, וכן שכל שדות החובה מולאו, במידה ולא, לא ניתן לבצע כניסה/ עידכון פרטים.
במידה והמשתמש החסיר הכנסת נתונים או שהנתונים שהוכנסו לא הוכנסו כראוי, פה תפעל פונקציית הולידציה ותוצג הודעה על כך על המסך.
בעת כניסה למערכת מבוצעת בדיקה ששם המשתמש והסיסמה אכן קיימים במערכת, כמו כן כאשר משתמש פותח תביעה במערכת מתבצעת בדיקה מול המסד נתונים כי אכן קיים ללקוח ביטוח מן הסוג אותו הוא רוצה לפתוח, במידה ולא המערכת לא תאפשר לו פתיחת התביעה.
כמו כן במידה ומבוצעת הרשמת מנהל, מכיון שמנהל הוא חשוף למידע רגיש על כל לקוחותיו מתבצעת אימות נוסף בעת הרשמתו עליו להזין קוד שקיבל ממשרדו, במידה וקוד זה תואם לקוד הנמצא במסד נתונים יכול להמשיך בהרשמתו אשר שלביו ככל לקוח

אבטחת מידע

כדי לדאוג לריבוי משתמשים וריבוי קריאות ואבטחה בחרתי להשתמש ב SQL Server המטפל בדרישות.
דוגמאות למקרים ותגובות להם ניתן מענה אבטחתי:
• במידה ובעת כניסה שם המשתמש והסיסמה אינם תואמים את הקיים במערכת, לא ניתן להתחבר למערכת.
• הנתונים ישמרו ב SQL Server שזהו מסד נתונים המתמודד עם עומסים.
• לא ניתן לצפות באזור האישי ללא התחברות למערכת.
• בעת פתיחת תביעה במידה וללקוח לא קיים פוליסה במסד נתונים לא תתאפשר לו פתיחת תביעה.

מה הקנה הפרוייקט?

לימוד ושליטה בסביבת API Net Core ובשפת #C ו JavaScript, התמקצעות ב SQL Server

- רכישת ניסיון ופיתוח הידע בסביבת React
- ידע והתמקצעות בעיצוב אתרים ואפליקציות על אופניו השונים
- פתרון בעיות
- ניסיון בביתוח מערכת
- ניסיון בתכנון נתונים ועיבוד יעיל
- ניסיון בבדיקות מקיפות
- פיתוח יכולת הלמידה העצמית

ביבליוגרפיה:

stackoverflow.com

chatGPT

אתרים שונים.

ז. הערות ראש המגמה במכללה

ח. אישור ראש המגמה

שם: מרים ויכלדר חתימה: _____ תאריך: _____

ט. הערות הגורם המקצועי מטעם מה"ט

י. אישור הגורם המקצועי מטעם מה"ט

שם: _____ חתימה: _____ תאריך: _____