



המכללה הטכנולוגית באר-שבע



מוגש ע"י: אופירה סולטן ומירב שימלמן

מגמת תוכנה

שם המנחה: גב' מ. שמעוןוביץ

המכללה הטכנולוגית באר-שבע



שמות המגישות: אופירה סולטן ומירב שימלמן

_____ חתימה:

_____ חתימה:

שם המנחה: גב' מ. שמעוןוביץ

_____ חתימת המנחה:

שם רכו המגמה בב"ש: מר א. ברגמן

_____ חתימה:

פברואר, 2018

תודות

ראשית לכל הננו רוצות להודות לרבנו של עולם שהביאנו עד הלום בידו הרחומה והאוהבת.
אנו אסירות תודה לכמות כה רבה של אנשים שליוונו בסבלנות, מאור פנים ואוזן קשבת:

לרכזת המגמה גב' בת שבע לוי שמנהלת את המגמה ביד רמה, במסירות ובאחריות מרובה שנים כה
רבות. לכל צוות המורות במגמה על הכשרתנו המקצועית ועל סבלנותן הרבה,
וכמובן למנחה היקרה גב' מרים שמעונוביץ שליוותה אותנו בסבלנות מרובה במשך התקופה ונתנה לנו
הרגשה נעימה, עודדה ותמכה והייתה תמיד לצידנו.

וכן תודה ענקית שלוחה להורינו ולמשפחותינו שליוו אותנו במסירות ונתנו לנו עידוד רב. וכן לכל
החברות במגמה ששאלו, התעניינו ועודדו...

שה' ישלם שכרכם!

תוכן העניינים

III.....	תודות
IV	תוכן העניינים
1	הצהרה
2	הצעה לפרויקט גמר
7	אישור הפרויקט שחזר ממה"ט
8	גוף העבודה:
8	מבוא
10	1. מדריך למתכנת:
10	1.1 אסטרטגיות טכנולוגיות:
12	1.2 תיאור מבנה הפרויקט:
13	1.3 עקרונות התכנון/ הבניה/ הניתוח:
17	1.4 תרשימים:
20	1.5 מבנה נתונים מאוכסנים:
24	1.6 תוכן הפרויקט:
43	3. מדריך למשתמש:
47	2.1 הוראות כלליות לשימוש באתר:
44	2.2 מסכים:
57	4. סיכום ומסקנות:
58	4. נספחים:
59	5. שונות:
60	6. ביבליוגרפיה:
61	תקציר

הצהרה

הצעה לפרויקט גמר

א. פרטי הסטודנט

שם הסטודנט	ת.ז. 9 ספרות	כתובת	טלפון נייד	תאריך סיום הלימודים
אופירה סולטן	204848956	רש"י 15 רחובות	052- 7149139	2017
מירב שימלמן	318406220	הקבלן 55 ירושלים	053- 3124584	2017

שם המכללה: טכנולוגית ב"ש סמל המכללה: 72204

מסלול ההכשרה: הנדסאים

מגמת לימוד: תוכנה

מקום ביצוע הפרויקט: סמינר אופקים

ב. פרטי המנחה האישי

שם המנחה	כתובת	טלפון נייד	תואר	מקום עבודה/תפקיד
מרים שימעונוביץ	חזון איש	052- 7171295	ר"צ פיתוח והוראת מחשבים	המכללה למנהל

ג. שם פרויקט הגמר

מור(ה) שת

ד. מטרת פרויקט הגמר

יצירת אתר המכיל מידע למורות פרטיות בצורה מאורגנת, מקוטלגת, ממוינת ונוחה לחיפוש. האתר יעוצב בצורה נוחה וברורה לשימוש המאפשרת למשתמש איתור מורות לפי מספר קריטריונים רב והוספת מורות למערכת, האתר מנהל לוח שנה ותשלומים למורה. מטרה נוספת היא לימוד מעמיק של טכנולוגיות שימושיות.

ה. ספרות עזר

אתרים:

1. בלוגים ואתרים נוספים
2. code Master
3. MSDN
4. Stack overflow

ו. תיאור קצר ונתונים טכניים:

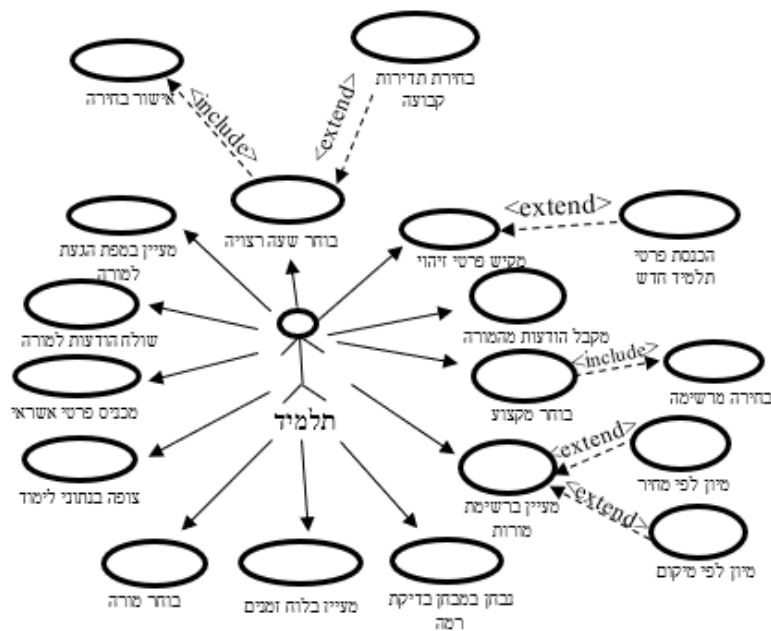
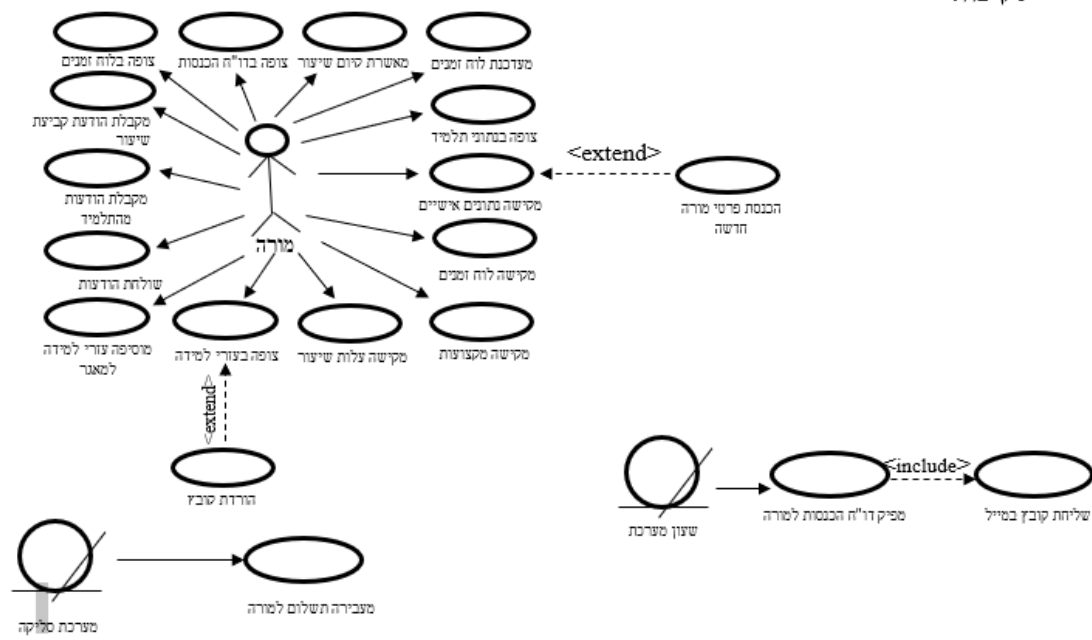
אתר לאיתור מורות פרטיות,

לאתר ישנה גישה לתלמיד ומנגד למורה, האתר מציע את שירותיו לציבור הרחב- מידע לשיעורים פרטיים ואמצעי עזר ללימוד.

באתר תהינה אופציות רבות:

- חיפוש מורה ע"פ קטגוריות מסויימות כגון: מחיר, מיקום, מקצוע וכו'...
- ניהול הרשאות גישה לתלמיד, למורה או לאורח.
- ניהול עזרי למידה- שכפולים, עזרים ותרגילים.
- ניהול תשלומים למורה.
- גביית תשלומים מתלמיד.
- ניהול לוח זמנים לשיעורים.
- תקשורת- צא'ט עם המורה.
- בדיקה של מבחני IQ לשיקוף רמת התלמיד.

סקיצה:



ז. פירוט הדרישות

האתר יפותח ב MVC- עם חיבור ל Entity Framework לגישה רחוקה ע"י Web Api.

ח. הערות ראש המגמה במכללה

ט. אישור ראש המגמה

שם: _____ חתימה: _____
תאריך: _____

י. הערות הגורם המקצועי מטעם מה"ט

יא. אישור הגורם המקצועי מטעם מה"ט

שם: _____ חתימה:

תאריך: _____

אישור הפרויקט שחזר ממה"ט

גוף העבודה:

מבוא

כאשר תלמיד נצרך לסיוע לימודי הוא חפץ למצוא עזרה שתתאים בצורה המקסימלית לדרישותיו; למקצוע שבו מתקשה או צריך תגבור, לסכום הכסף שמוכן להשקיע, למקום מגוריו או לימודיו וכיוצא בזה.

נוסף על כך התלמיד אף יהיה חפץ בתגובות על אופן לימוד המורה ושביעות רצון "הלקוחות" – מה שימקד אותו עוד יותר בבחירתו.

ועל סמך הידיעות הללו יוכל התלמיד לבטח לבחור את המורה הטובה ביותר עבורו.

אך על מנת שאכן כל דרישותיו תתמלאנה עליו להשקיע זמן **רב** בבירורים – מה שלא כל אדם יוכל להרשות לעצמו, ומפאת חוסר אותו זמן יזדקק להתפשר ולוותר על חלק מדרישותיו והמה כמובן יבואו לידי ביטוי בהתאם, נוסף על כך, חרף הבירורים הרבים לא יוכל להגיע למירב התוצאות משום שחיפושיו הינם רק בקרב מכריו ותו לא.

אי לכך ובהתאם לזאת – ראינו צורך לתכנת אתר – שבזמן מינימלי מפנה למורה המתאימה מקסימלית.

התוכנה תכיל ממשקים שונים עבור הרשאות שונות:

- ממשקים לאורח מזדמן – האורח יכול לבצע חיפוש והמידע שיופק מחיפושו יהיה מצומצם מאוד עם פרטים מינימליים. לא ניתנת בידו אופציה לקביעת שיעור, ולצפייה בנתוני מורה אישיים (– אלו יינתנו רק בתנאי שהמשתמש הינו תלמיד במערכת).
- ממשקים לתלמיד במערכת – התלמיד רשאי לבצע חיפוש, לצפות בנתוני מורה אישיים, לקבוע שיעור ולשבצו במערכת אישית שלו, לבצע תשלום למורה על שיעור שבוצע, בנוסף – התלמיד יכול לשנות ולעדכן פרטים אישיים שלו במערכת.

- ממשקים למורה במערכת – המורה יכולה לבטל שיעור שנקבע עם תלמיד, לערוך את השעות השבועיות הקבועות שלה, להכניס נושאים בהם הינה מאפשרת שיעור פרטי, לתעד שיעור שנעשה, לעלות קבצי העשרה וכו' עבור התלמיד, לערוך את הפרטים האישיים שלה בפרופיל.

האתר מעניק למשתמש חיפוש מורה על פי קריטריונים לבחירתו – מחיר, מיקום, והמלצות.

התלמיד יכול ליצור עם המורה קשר טלפוני\ דרך האתר בצ'אט\ דרך מייל המורה.

יצירת אתר המכיל מידע למורות פרטיות בצורה מאורגנת, מקוטלגת, ממוינת ונוחה לחיפוש. האתר יעוצב בצורה נוחה וברורה לשימוש המאפשרת למשתמש איתור מורות לפי מספר קריטריונים רב והוספת מורות למערכת, האתר מנהל לוח שנה ותשלומים למורה. מטרה נוספת היא לימוד מעמיק של טכנולוגיות שימושיות.

1. מדריך למתכנת:

1.1 אסטרטגיות טכנולוגיות:

השתמשנו בשפות הבאות בכתיבת הפרויקט:

- **C#**: שפת C# היא שפת תכנות שפותחה ע"י מיקרוסופט ונחשבת לאחת משפות התכנות הפופולריות בעולם. היא מיועדת לפיתוח כללי של מגוון אפליקציות בכל התחומים מאתרי Web, דרך משחקים, מאפליקציות למכשירי מובייל וטאבלטים ועד לשירותי ענן. התחביר והעקרונות שלה הם פשוטים מצד אחד אך עשירים ביכולות מצד שני.
- **Html**: זוהי השפה הטבעית ליצירת דפי אינטרנט ברשת. זו שפה פשוטה, אוניברסלית המאפשרת לעורכי אתרים ליצור דפים מורכבים שמכילים טקסט ותמונות, שיכולים להראות בידי כל המשתמשים ברשת האינטרנט ללא תלות בסוג המחשב או בסוג הדפדפן.
- **JavaScript**: JavaScript היא שפת תכנות של HTML והאינטרנט. מאפשרת למפתחים להטביע קטעי סקריפט ("תוכניות קטנות") בתוך מסמכי HTML. ובכך להפוך את הדפים לדינמיים יותר.
- **CSS**: CSS היא שפת עיצוב המגדירה תבנית העיצוב של מסמכי HTML. CSS למשל מטפלת בגופנים, צבעים, גבולות, שורות, גובה, רוחב, תמונות רקע, מיקום מתקדם ודברים רבים נוספים.
- **jQuery**: jQuery היא תשתית פיתוח עבור JavaScript. ההופכת פעולות נפוצות ב JavaScript לקלות הרבה יותר, עוזרת לשפר את חווית המשתמשים באתרים ומטפלת בהבדלים באופן המימוש של JavaScript בין דפדפנים שונים.

כתבנו את הפרויקט בטכנולוגיית Asp.net MVC.

Asp.net MVC היא הטכנולוגיה המתקדמת ביותר מבית מיקרוסופט לפיתוח אפליקציות WEB. הטכנולוגיה שמה דגש על פיתוח מהיר, שימוש בארכיטקטורה נכונה והפרדת שכבות, שילוב ספריות קוד פתוח ותמיכה מובנית בבדיקות אוטומטיות.

Asp.net MVC הינה תשתית אשר משמשת לבניית יישומי אינטרנט, ע"י החלת העקרונות של תבנית מודל-תצוגה-בקר (Design Pattern Model-View-Controller) על תשתית ASP.NET.

תבנית MVC מהווה כבר שנים רבות תבנית ארכיטקטית חשובה בתחום מדעי המחשב. התבנית מייצגת דרך יעילה להפרדה בתוך היישום (למשל הפרדת רכיב הגישה לנתונים מרכיב הטיפול בתצוגה) והיא יעילה לעיצוב יישומי אינטרנט.

ההפרדה המוחלטת של המרכיבים מוסיפה מידת מה של מורכבות ליישום, אולם התועלת הרבה שמתקבלת בתמורה מצדיקה את המאמץ.

תבנית MVC מפרידה את היישום לשלושה היבטים עיקריים:

- 1. Model** - קבוצה של מחלקות המתארות את הנתונים עליהם פועלים. המודלים יהיו ברוב המקרים מעין שכבת גישה לנתונים, תוך שימוש בכלים כגון Entity Framework.
- 2. View** - הגדרת אופן ההצגה של ממשק המשתמש.
- 3. Controller** השכבה האחראית על היחסים בין ה Model לבין ה View. מגיב לקלט מהמשתמש, -מדבר עם המודל, וקובע איזו תצוגה צריך לממש.

השתמשנו בסביבות העבודה הבאות:

✓ Visual Studio 2015:

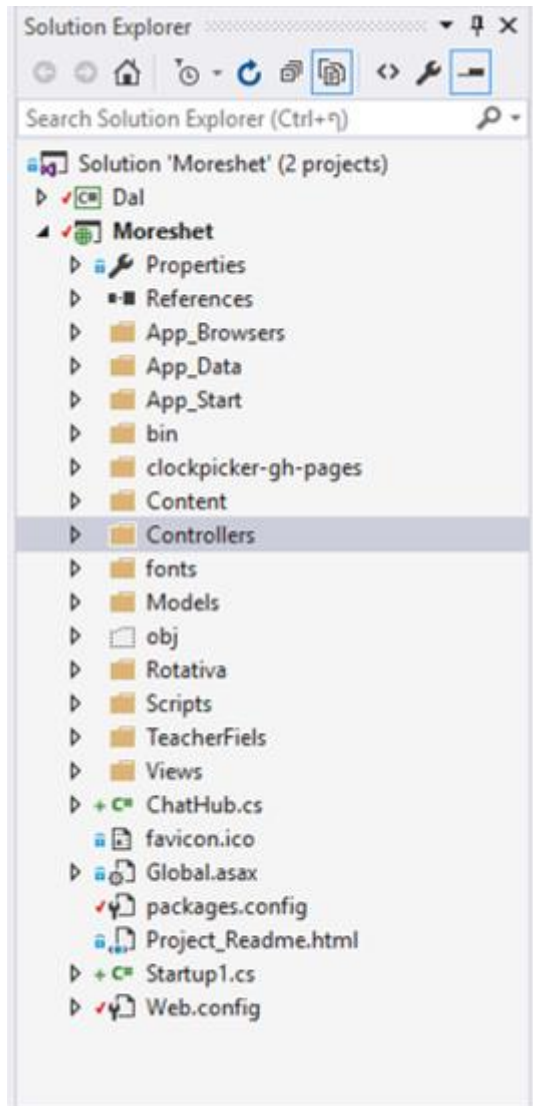
סביבת הפיתוח המרכזית בעולם המייקרוסופטי נקראת Visual Studio. סביבה זאת מכילה מאות אפשרויות. ותומכת במספר שפות תכנות. בנוסף, יש לציין כי רוב שפות התכנות והטכנולוגיות הנתמכות על ידי סביבת פיתוח זו הן חלק מתשתית הפיתוח של חברת מייקרוסופט .NET Framework..תשתית זו מכילה מספר עצום של ספריות קוד וטכנולוגיות המקלות עלינו בתהליך יצירת התוכנה.

✓ SQL Server 2017:

Microsoft SQL Server היא פלטפורמת שרת מידע ומבנה נתונים המיועדת למפתחים, ארגונים קטנים וגדולים כאחד אשר מציעה חבילה טכנולוגית שלמה של כלים ארגוניים המאפשרים להשיג את הערך המרבי מתוך המידע שלנו במחיר ובעלות הכוללת הנמוכים ביותר. תוכנת ניהול השרת מיועדת למשתמשים הזקוקים לפתרונות ניהול מידע באמצעות בסיסי נתונים בצורה יעילה ומאובטחת. בנוסף ניתן לפתח ולנהל תוכנות למחשבים שולחניים ותוכנות מבוססות רשת העושות שימוש בבסיס נתונים המנוהל תחת שרת SQL Server במהלך העבודה עימן.

1.2 תיאור מבנה הפרויקט:

ה- Solution שלנו כולל 2 פרויקטים:



1. **"Dal"** – הפרויקט מובנה בשיטת Entity Framework CodeFirst ולכן המחלקות והמשתנים נכתבו הכתובות בשפת #C ועל סמך זאת- נבנה הDB.

2. **"Moreshet"** – הפרויקט העיקרי ממנו מובנה האתר. מחולק ל-3 שכבות עיקריות:
:Models

קבוצה של מחלקות הכתובות בשפת #C המתארות את הנתונים עליהם פועלים. כל מחלקה כוללת מאפיינים לפי צורך השימוש בפרויקט על סמך סנכרון הנתונים מתוך ה- DataBase.

לדוגמא: מחלקת EmailDetails המתארת נתוני מייל לשליחה וכוללת את המאפיינים הבאים: כתובת מייל הנמען, תוכן המייל וכיוצא.

:Views

הגדרת אופן ההצגה של ממשק המשתמש. כל View מציד דף Html אחר בדפדפן, וב- View מגדירים את תוכן וצורת התצוגה.

:Controllers

השכבה האחראית על היחסים בין ה- Model לבין ה- View. מגיב לקלט מן המשתמש, "מדבר" עם ה- Model, וקובע איזו תצוגה צריך לממש. ה- Controller מקבל נתונים נצרכים מה- View, עושה חישובים שונים עם שימוש ב- DataBase וב- Model, ומחזיר נתונים לתצוגת ה- View הרצוי.

1.3 עקרונות התכנון/ הבניה/ הניתוח:

1.3.1 עקרונות תיאוריים:

- הורשה:

בהורשה משתמשים בתכנות מונחה-עצמים, ירושה היא דרך לבסס יחס "סוג-של" (is-a) בין עצמים. המימוש הנפוץ של הגדרה זו היא בעזרת מחלקות. מחלקה יכולה לקבל בירושה תכונות והתנהגות של מחלקת-האם שלה (נקראת גם "מחלקת-על", "מחלקת-אב" או "מחלקת בסיס"). היחסים בין מחלקות מגדירים היררכיה של מחלקות. מגגנון הירושה מסייע במידול של תחומים בדרך דומה לזאת המוגדרת באופן "טבעי" על ידי העוסקים בהם.

לדוגמה: אם ריבוע הוא סוג-של (או מקרה פרטי של) מצולע, ניתן להגדיר יחס ירושה בין המחלקה מרובע למחלקה מצולע; על פי רוב, אובייקט של המחלקה היורשת (מרובע) יתמוך בכל הפעולות המוגדרות על מצולע כלשהו, ובנוסף יתמוך בפעולות נוספות, המיוחדות למרובע. בדומה, ניתן להגדיר יחס ירושה בין המחלקה ריבוע למחלקה מרובע.

שימושים:

ירושה מאפשרת להשיג מספר מטרות:

1. שימוש חוזר בקוד. בעזרת ירושה ניתן לכתוב מחלקות בעלות טווח התנהגות נרחב, תוך כתיבת קטעי קוד קטנים יחסית שמרחיבים מחלקה קיימת.
2. מידול של העולם האמיתי באופן דומה לזה המוגדר על ידי האדם. למשל, ניתן לכתוב היררכיה של מחלקות המקבילה לטקסונומיה המקובלת בביולוגיה (יען היא סוג של ציפור, האדם הוא סוג של יונק, וכן הלאה).
3. פולימורפיזם (רב צורתיות) של זמן-ריצה. משתנה מטיפוס (סטטי) מסוים יכול להתנהג בצורות שונות, לפי הטיפוס של האובייקט שהוא מתייחס אליו בזמן ריצה. למשל משתנה מהמחלקה **בעל חיים** שנדרש לבצע פעולה מסוג "השמע קול" יבצע פעולות שונות לחלוטין במקרה שהאובייקט הוא מהמחלקה **כלב** או **חתול**. ניתן לבצע זאת גם ללא ירושה (בעזרת פקודות תנאי), אך ירושה מאפשרת מימוש מודולרי, גמיש וקל להרחבה.

בפרויקט השתמשנו בהורשה במקרים רבים:

- מחלקת ה-chat:

יורשת ממחלקת **Hub** המקושרת ל- `Microsoft.AspNet.SignalR`. ממחלקה זו נוצר הצאט-תקשורת בין מורה לתלמיד.

```
namespace Moreshet
{
    public class ChatHub : Hub
    {
        public void Send(string name, string message)
        {
            Clients.All.addNewMessageToPage(name, message);
        }
    }
}
```

- מחלקת ה-Db:

יורשת ממחלקת **DbContext** המקושרת ל- `System.Data.Entity`. ממחלקה זו נוצר מאגר הנתונים - database ע"י codefirst עם חיבור ל- `entity framework`.

```
{
    public class Db:DbContext
    {
        public Db():
            base(@"Data Source = (localdb)\MSSQLLocalDB;Initial Catalog = MoreshetNew; Integrated Security = True;
                Connect Timeout = 30; Encrypt=False;TrustServerCertificate=True;ApplicationIntent=ReadWrite;
                MultiSubnetFailover=False")
        {
        }
    }
}
```

▪ מחלקות ה-views:

יורשות ממחלקות ה-models.

לדוגמה: ה-view של create new user יורש ממחלקת ה-user.

```
@model Dal.User
<title>הוספת תלמיד חדש - ברוך הבא</title>
<link href="/Content/css.css" rel="stylesheet" />
<script>
    $('input[type=file]').change(function () {
        alert(this.files[0].mozFullPath);
    });
</script>
<style>
    label.form-horizontal {
        direction: rtl;
        font-family: David;
        font-size: large;
    }
</style>
<h3 class="text-right" style="font-family:Arial">הוספת תלמיד/ה חדש</h3>
<hr />
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div style="align-content:center;align-items:center" dir="rtl" class="form-horizontal">
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div dir="rtl" class="form-group">
            @Html.LabelFor(model => model.UserCode)
            <div dir="rtl" class="col-md-11">
                @Html.EditorFor(model => model.UserCode, new { htmlAttributes = new { @class = "form-control" } })
                <span class="glyphicon glyphicon-user"></span>
                @Html.ValidationMessageFor(model => model.UserCode, "", new { @class = "text-danger" })
            </div>
        </div>
    </div>
}
```

• הפרדת שכבות:

כל תוכנה/ אתר שנפגוש בעולם מבוססת על ארכיטקטורת שכבות הנקראת בשם Three-Tier-Application
ז"א שלש שכבות במבנה של DAL-BL-UI.
זוהי תבנית עיצוב בסיסית שמגדירה הפרדת האפליקציה ל: שכבת נתונים, שכבת לוגיקה, ושכבת ממשק משתמש.

לתבנית עיצוב זו יתרונות רבים:

1. תחזוקה:

ניתן להחליף או לתקן מימוש פנימי של שכבה אחת בארכיטקטורה בלי לשנות שכבה אחרת.

2. נוחות פיתוח:

אדם אחד עובד על רכיב בתכנה, אדם אחר עובד על רכיב אחר, כל עוד שהחתימות זהות ניתן לשלב כוחות ולייעל זמני פיתוח.

3. בדיקות:

תקלה כלשהיא מבודדת בכל שכבה בנפרד, לדוגמא אם לא קיבלנו רשימת נתונים לתצוגה נבדוק קודם את שלב הביניים (שכבת ה-BL) אם הנתונים שם תקינים נדע בוודאות שגם ברמת שכבת ה-DAL הנתונים תקינים וכל שנותר הוא לפתור את התקלה ברמת ה-UI

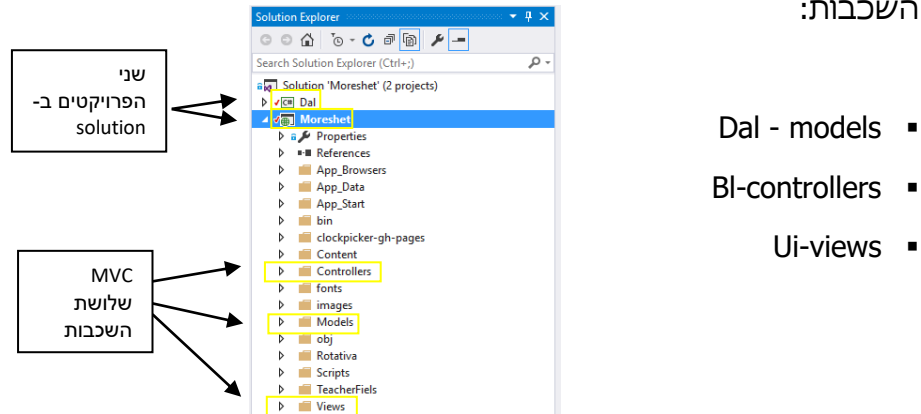
4. שימוש חוזר:

נניח שהרכיבים DAL + BL אהובים עלינו ועובדים היטב, ניתן להחליף את פלטפורמת ה-UI לטכנולוגיה אחרת במינימום מאמץ.

5. אבטחה:

נוח יותר להגדיר Interface-API לשכבה מסוימת בלי לחשוף מבני נתונים, Logic או DATA שלא רלוונטי למשתמש, כמו גם חסימה בפני האקרים (רלוונטי יותר בטכנולוגיות WEB) ברמות שונות.

הפרויקט נכתב בשפת c# ומיושם בטכנולוגיית mvc. טכנולוגיה זו בנויה על שלושת השכבות:



הפרויקט בנוי משני פרויקטים:

- Dal - פרויקט בו ישנן מחלקות הכתובות בשפת c#. ממנו נבנה ה-database.
- Moreshet - הפרויקט העיקרי המחולק לשלושת השכבות שתוארו לעיל.

1.3.2 עקרונות תכנותיים:

- Session:

שמירת מידע בצד שרת-

הוא משאב אישי ללקוח, המידע אינו משותף ואינו נגיש ללקוחות האחרים. כאשר הלקוח מבקר באתר, השרת מקצה לו אובייקט Session ושומר אותו בזיכרון. Session האובייקט נשמר בזיכרון כל עוד המשתמש נשאר באתר. אם במשך זמן (מוגדר מראש) האתר ננטש, האובייקט Session נמחק. הוא נמחק גם כאשר סוגרים את הדפדפן או כאשר נותנים הוראה למחיקתו.

יישום:

בעת כניסה לאתר ע"י שם משתמש וסיסמה נשמר המשתמש ב-session. וכך במעבר בין דפים נשמרים הנתונים למשתמש הנוכחי.

```
c.Value = user.Telephone;  
System.Web.HttpContext.Curren  
Session["user"] = user;  
switch (user.Role)
```

בעת יציאה המשתמש ב-session נמחק.

```
/// <summary> To get out from account  
public ActionResult GetOut()  
{  
    Session["user"] = null;  
    return RedirectToAction("Index", "Home");  
}
```

במהלך ריצת האתר ניתנת למשתמש השמור ב-session הרשאת גישה לדפים המיוחדים לו. בדוגמה זו רק למורה ניתנת האפשרות להוסיף מקצוע...

```
@if (Session["user"] != null && (Session["user"] as Dal.User).Role == "מורה")  
{  
    <li class="c2" onmouseout="moveLiOut(this)" onmouseover="moveLi(this)"  
        style="font-style:inherit;font-weight:bold;font-family:Tahoma;font-size:larger";  
        @Html.ActionLink("הוספת מקצוע", "AddSubjectForTeacher", "First") </li>  
}
```

- Cookie:

שמירת מידע בצד לקוח-

עוגיות הן אחד מכמה דרכים לאחסון נתונים על משתמשים באתר אינטרנט, בתקופה שבה הוא כדי לזכור Cookie שרת ודפדפן אינטרנט אינם מחוברים. שימוש נפוץ של קבצי הוא קובץ טקסט קטן שנשלח על ידי Cookie משתמשים בין ביקורים. למעשה, קובץ שרת אינטרנט ונשמר ע"י דפדפן אינטרנט במחשב הלקוח.

עוגיות עשויות לשמש אימות, זיהוי של פעילות משתמש, העדפות משתמש, תוכן עגלת קניות, או כל דבר אחר שיכולים להשיג באמצעות אחסון נתוני טקסט. עוגיות יכולות לשמש גם עבור העברת נתונים מדף אחד למשנהו.

יישום:

במשך הפרויקט שמרנו את המשתמש ב- session (כנכתב למעלה). אך נתקלנו בבעיה בשמירת ה-session בשליחת לוח השנה להדפסה. המחלקה action as pdf מאבדת את ה-session כשמחזירים view. ומשום כך השתמשנו ב-cookie לשמירת המשתמש, ואח"כ החזרנו את המשתמש ל- session.

```
HttpCookie c = new HttpCookie("user");
c.Value = user.Telephone;
System.Web.HttpContext.Current.Response.Cookies.Add(c);
Session["user"] = user;
```

בכניסה המשתמש נשמר
ב-Cookie וכן ב-session

```
public ActionResult PrintCalendar(int? month, int? year, int? id, int? idSubject)
{
    ActionAsPdf report;
    Dictionary<string, string> cookieCollection = new Dictionary<string, string>();
    HttpCookie cookie = System.Web.HttpContext.Current.Request.Cookies["user"];
    cookieCollection.Add(cookie.Value, cookie.Name);
    if (id == null && idSubject == null)
    {
        report = new ActionAsPdf("Overview2", new { month = month, year = year, frompdf = true })
        {
            FileName = "MyCalender.pdf",
            FormsAuthenticationCookieName = System.Web.HttpContext.Current.Request.Cookies["user"].Value,
            Cookies = cookieCollection
        };
    }
    else
    {
        report = new ActionAsPdf("Overview2", new { month = month, year = year, id = id, idSubject = idSubject, frompdf = true })
        {
            FileName = "MyCalender.pdf",
            FormsAuthenticationCookieName = System.Web.HttpContext.Current.Request.Cookies["user"].Value,
            Cookies = cookieCollection
        };
    }
    return report;
}
```

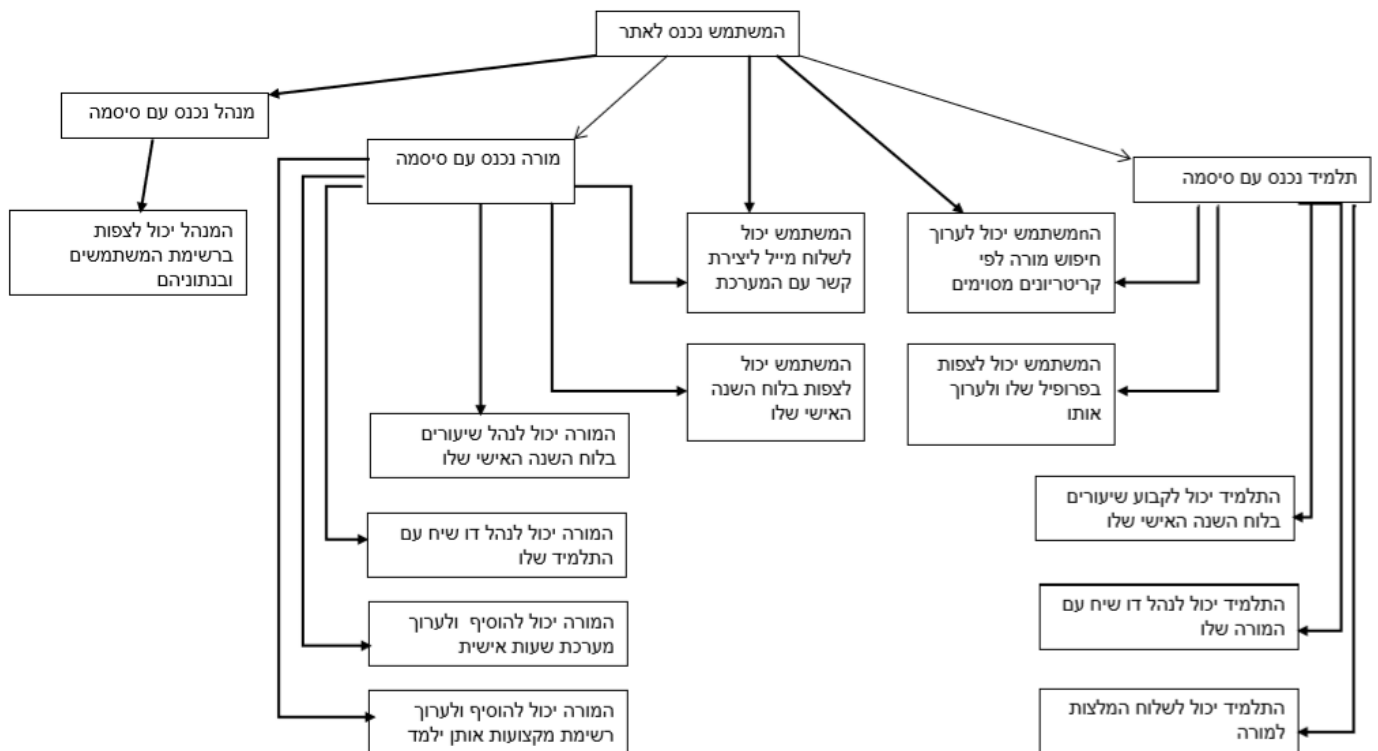
בפונקציית ההדפסה שמירת
ה- cookies בחזרה מהפונקציה

שליפת ה-cookie והכנסתו חזרה
ל- session

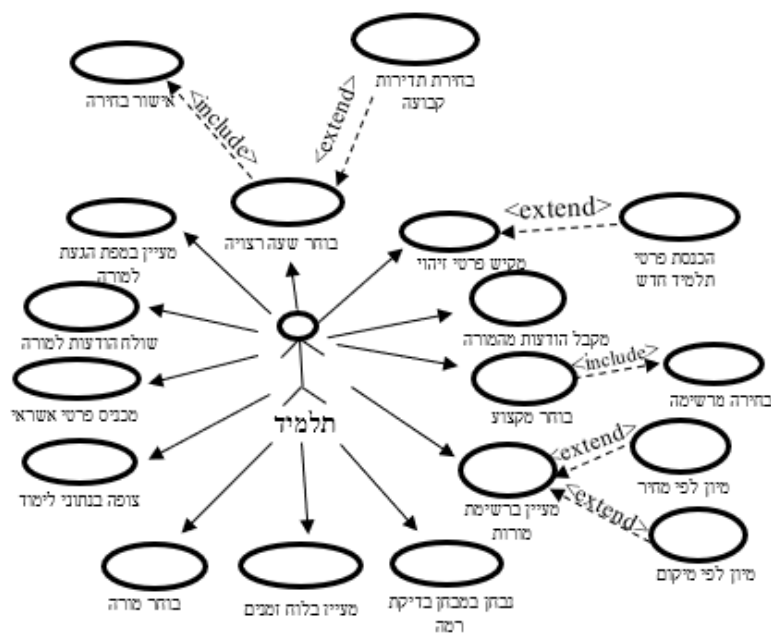
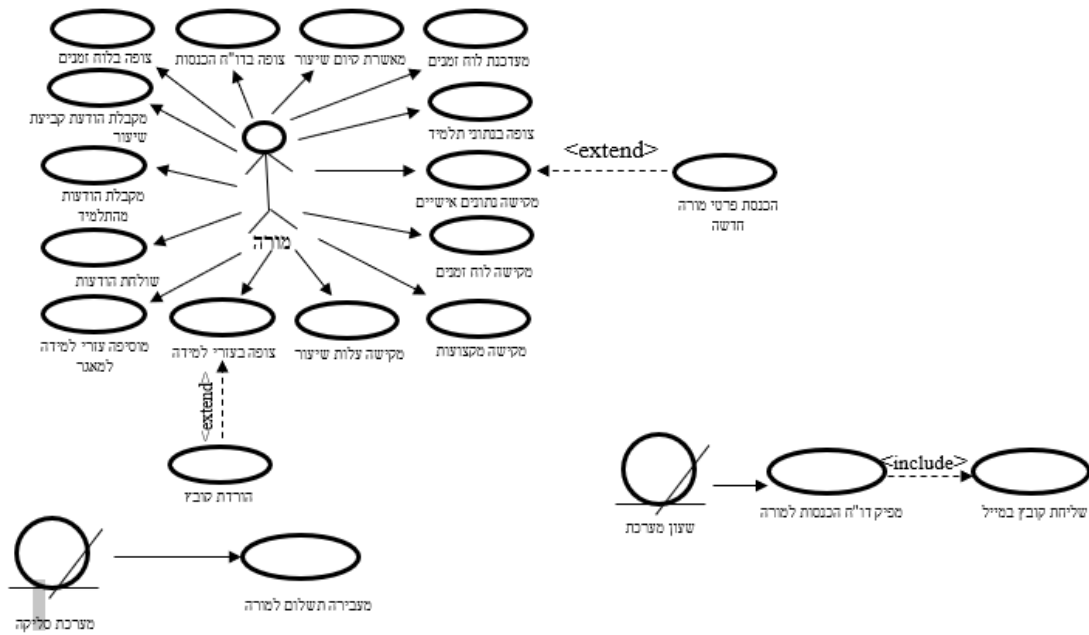
```
public class CalendarController : Controller
{
    /// <summary> First Instance for Calendar
    public ActionResult Overview2(int? month, int? year, int? id, int? idSubject, bool frompdf=false)
    {
        bool degel = false;
        if (frompdf == true)
        {
            HttpCookie cookieValueFromReq = System.Web.HttpContext.Current.Request.Cookies.Get(0);
            using (Dal.Db db = new Db())
            {
                User user = db.UsersT.FirstOrDefault(i => i.Telephone == cookieValueFromReq.Name);
                Session["user"] = user;
            }
        }
    }
}
```

1.4 תרשימים:

1.4.1 עץ תהליכים:

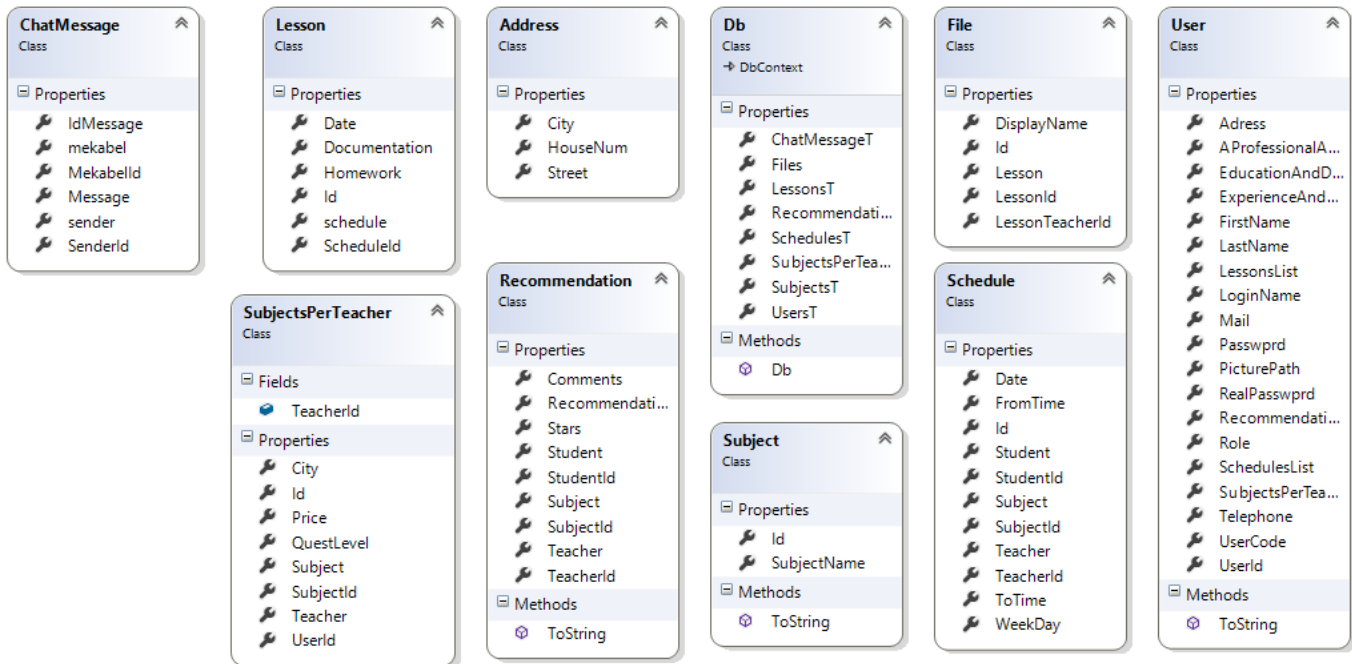


1.4.2 תרשים Uml:

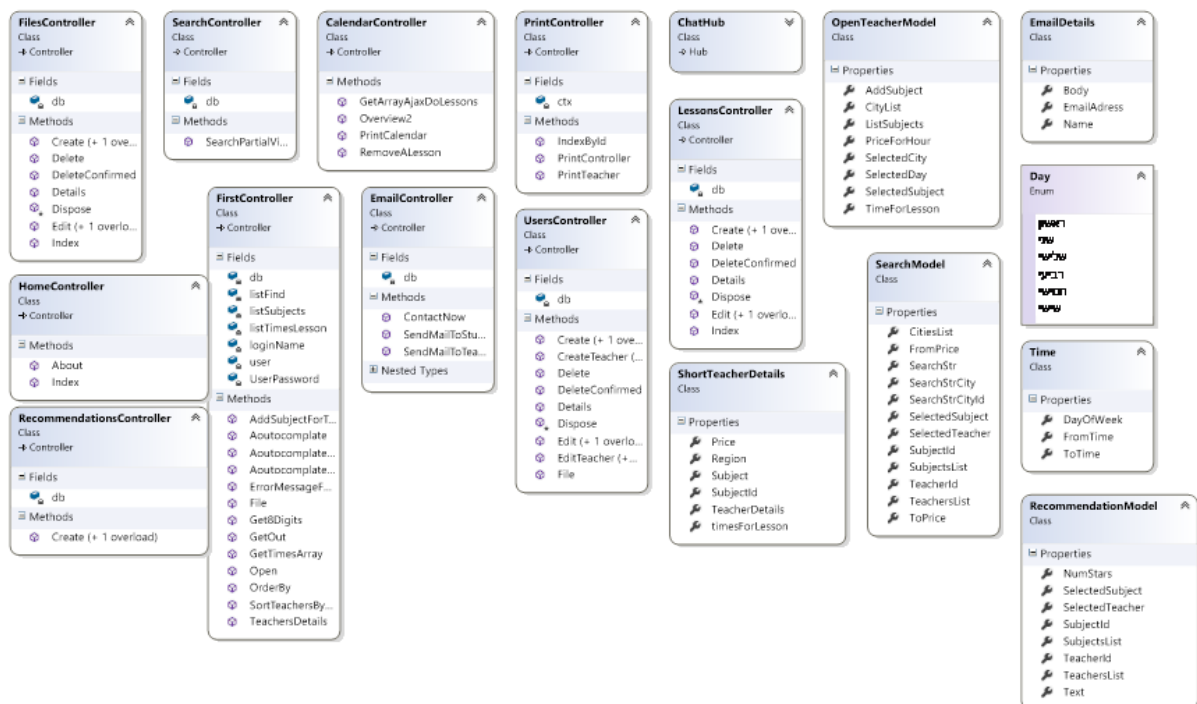


1.4.3 תרשים מראה המחלקות:

• פרויקט ה-Dal:

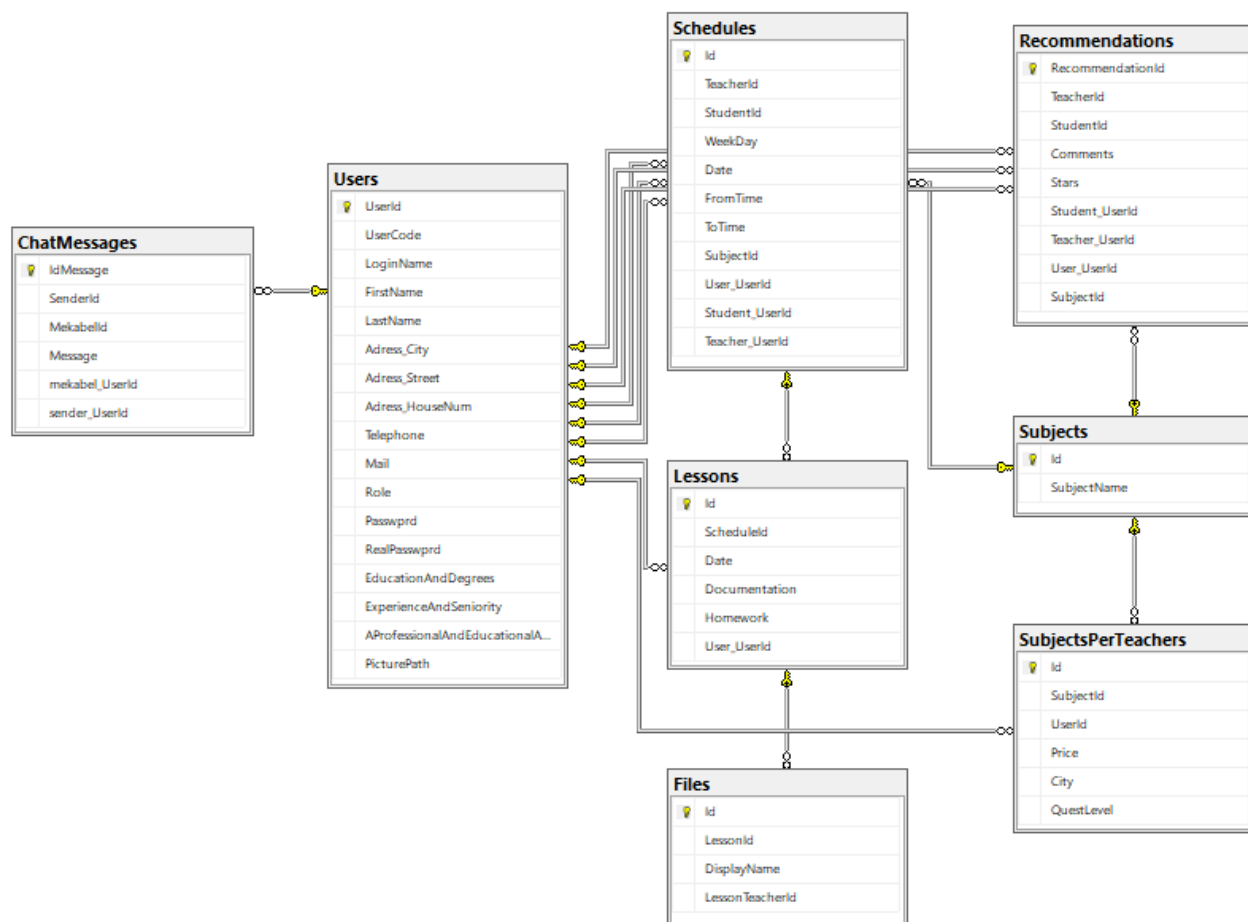


• פרויקט ה-Moreshet:



1.5 מבנה נתונים מאוכסנים:

1.5.1 תרשים SQL:



1.5.2 תיאור מבנה קבצי XML:

XML הינם ראשי תיבות של (Extensible Markup Language) הוא תקן לייצוג נתונים במחשבים. שימוש ב־XML מקל על החלפת נתונים בין מערכות שונות שפועלות על גבי תשתיות שונות. תקן ה־XML לא מגדיר איזה מידע יוצג אלא מגדיר כיצד לייצג מידע באופן כללי. תקן XML שייך למשפחת שפות הסימון (markup language) ומבוסס על תקן משנת 1986 בשם SGML.

מבחינה טכנית XML, איננה שפת תכנות, שכן אף על פי שהתחביר (syntax) שלה מוגדר היטב, אין לה אוצר מילים ולא בקרת זרימה. למעשה, ניתן לראותה כתבנית ליצירת שפות אחרות, ביניהן XHTML, XAML, XUL, SVG, XBRL וכדומה.

גם תקן ה־XML בדומה לשפת HTML, נקבע על ידי ה־World Wide Web Consortium.

תכונות ה־XML:

- ייצוג המידע באופן טקסטואלי.
- שמירת המטא-מידע ביחד עם המידע עצמו. כלומר, שמירת תיאור הנתונים עם הנתונים עצמם.
- סידור המידע במגוון צורות ולא רק בטבלה כמו ברוב הפורמטים האחרים. ב־XML טבעי מאוד לסדר מידע באופן היררכי (עץ).

קבצי ה־XML בפרויקט:

1. SubjectsForTeacherSch - השעות השבועיות למורה

הקובץ מוחלק לפי מורות ולכל מורה השעות שלה. בקובץ זה השתמשנו על מנת לרכז שעות למורה ועל סמך כך לאפשר לתלמיד לקבוע שיעור עם אותה מורה בזמנים הנוחים לה, אותם היא עדכנה. לדוגמא:

```
<Teacher id="3002" FirstName="חנה" LastName="שימלמן">
  <timeOfLesson dayOfWeek="1" fromTime="13:50" toTime="14:35" />
  <timeOfLesson dayOfWeek="2" fromTime="13:50" toTime="14:35" />
</Teacher>
```

הסבר -

המורה: חנה שימלמן,
השעות השבועיות: יום ראשון- משעה 15:50 עד שעה: 14:35
יום שני- משעה 13:50 עד שעה: 14:35

2. HebrewHoliday

קובץ זה מכיל את החגים היהודיים, השתמשנו בקובץ זה על מנת להציג את החגים והמועדים בלוח השנה בו נקבעים השיעורים עבור תלמיד או מורה.

```
<Holidays>
  <Holiday day="ט" month="אלול">ראש השנה</Holiday>
  <Holiday day="א" month="תשרי">ראש השנה</Holiday>
  <Holiday day="ב" month="תשרי">ראש השנה</Holiday>
</Holidays>
```

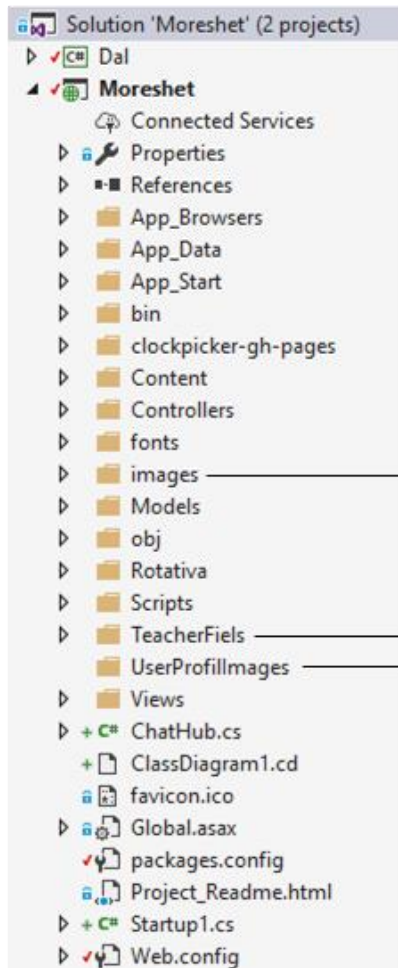
3. CitiesXml – ריכוז ערים וישובים בארץ ישראל

בקובץ זה מופיעה רשימה המכילה את שמות הערים והישובים בארץ ישראל. השתמשו במידע זה עבור השלמה אוטומטית בהכנסת כתובת משתמש וכו'.

```
<ROW>
  <טבלה/>טבלת ישובים<טבלה>
  <סמל_ישוב/>31<סמל_ישוב>
  <שם_ישוב/>אופקים<שם_ישוב>
  <שם_ישוב_לועזי/>OFAQIM<שם_ישוב_לועזי>
  <סמל_נפה/>62<סמל_נפה>
  <שם_נפה/>באר שבע<שם_נפה>
  <סמל_לשכת_מנא/>62<סמל_לשכת_מנא>
  <לשכה/>באר שבע<לשכה>
  <סמל_מועצה_איזורית/>0<סמל_מועצה_איזורית>
  <שם_מועצה/>
</ROW>
```

1.5.3 מבנה קבצים ותיקיות:

1. תיקיית תמונות עיצוב האתר-
תיקייה זו מכילה את התמונות בהן השתמשו לעיצוב האתר.
2. קבצים המועלים לתאר ע"י המורה עבור התלמיד-
הינה תיקייה המכילה, כשמה כן היא, קבצים שהמורה מעלה, קבצים לימודיים עבור התלמיד, קבצי העשרה וכיוצא.
3. תיקיית תמונות פרופיל המשתמש-
כאשר משתמש נרשם לאתר יש באפשרותו להוסיף תמונה לפרופיל האישי שלו, תמונות אלו כאמור, נכנסים לתיקייה זו.



1. תיקיית תמונות עיצוב האתר

2. קבצים המועלים לתאר ע"י המורה עבור תלמיד

3. תיקיית תמונות פרופיל המשתמש

1.6 תוכן הפרויקט:

1.6.1 תיאור המחלקות:

מחלקות ב- Dal:

- **Address (כתובת)**

מחלקה זו מייצגת כתובת של המשתמש, מחלקה זו אינה נבנית בתור טבלה ב- DB אלא מסוג Complex-Type.

```
namespace Dal
{
    [ComplexType]
    public class Address
    {
        [Display(Name = "עיר")]
        public string City { get; set; }
        [Display(Name = "רחוב")]
        public string Street { get; set; }
        [Display(Name = "מספר בית")]
        public int HouseNum { get; set; }
    }
}
```

- **Chat Message (הודעת צא'ט)**

מחלקה זו מייצגת את הודעות הצא'ט המתנהלות בין המשתמשים.

```
namespace Dal
{
    public class ChatMessage
    {
        [Key]
        public int IdMessage { get; set; }
        [Required, Display(Name = "קוד השולח")]
        public int SenderId { get; set; }
        public virtual User sender { get; set; }
        [Required, Display(Name = "קוד המקבל")]
        public int MekabelId { get; set; }
        public virtual User mekabel { get; set; }
        [Required, Display(Name = "גוף ההודעה")]
        public string Message { get; set; }
        //public string ConnectionId { get; set; }
    }
}
```

• User (משתמש)

מחלקה זו מייצגת נתוני משתמש. כולל רשימה של שיעורים שנקבעו, עבור מורה מחיר לשיעור וכו'.

```
namespace Dal
{
    public class User
    {
        [Key, DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int UserID { get; set; }
        [Display(Name = "תעודת זהות"), MinLength(9, ErrorMessage = "נדרש 9 ספרות"), MaxLength(9, ErrorMessage = "נדרש 9 ספרות"), Required]
        public string UserCode { get; set; }
        [Required, Display(Name = "שם משתמש")]
        public string LoginName { get; set; } //שם משתמש
        [Display(Name = "שם פרטי")]
        public string FirstName { get; set; }
        [Display(Name = "שם משפחה")]
        public string LastName { get; set; }
        [Display(Name = "כתובת")]
        public Address Address { get; set; }
        [Required, Phone, Display(Name = "טלפון")]
        public string Telephone { get; set; }
        [EmailAddress, Display(Name = "ימייל")]
        public string Mail { get; set; }
        [Display(Name = "הרשאה")]
        public string Role { get; set; }
        [Required, MinLength(8, ErrorMessage = "אורך הסיסמה המינימלי הינו 8 ספרות"), Display(Name = "סיסמה")]
        public string Passwprd { get; set; }
        [Required, MinLength(8, ErrorMessage = "אורך הסיסמה המינימלי הינו 8 ספרות"), Display(Name = "אימות סיסמה")]
        public string RealPasswprd { get; set; }
        [Display(Name = "השכלה ותעריים")]
        public string EducationAndDegrees { get; set; }
        [Display(Name = "נסיון וותק")]
        public string ExperienceAndSeniority { get; set; }
        [Display(Name = "אג'נדה מקצועית ומינוכית")]
        public string AProfessionalAndEducationalAgenda { get; set; }
        public string PicturePath { get; set; }
        public virtual List<SubjectsPerTeacher> SubjectsPerTeacherslist { get; set; } //למורה שיעורים
        public virtual List<Schedule> Scheduleslist { get; set; }
        public virtual List<Lesson> Lessonslist { get; set; }
        public virtual List<Recommendation> Recommendationslist { get; set; }

        public override string ToString()
        {
            return FirstName+" "+LastName;
        }
    }
}
```

• Subject (מקצוע)

מחלקה המאפשרת למורה להוסיף מקצוע שאינו קיים במערכת.

```
namespace Dal
{
    public class Subject
    {
        [Key, Display(Name = "קיד")]
        public int Id { get; set; }
        [Required]
        [Display(Name = "שם המקצוע")]
        public string SubjectName { get; set; }

        public override string ToString()
        {
            return SubjectName;
        }
    }
}
```

- **SubjectPerTeacher (מקצוע עבור מורה)**

נתונים אלו מהווים מקצוע שמורה מלמדת, הוי אומר, מחיר עבור אותו מקצוע, עיר בה ניתן לקבל שיעור באותו מקצוע ועוד נתוני זיהוי.

```
namespace Dal
{
    public class SubjectsPerTeacher
    {
        public int TeacherId;

        [Required]
        [Key, Display(Name = "קוד")]
        public int Id { get; set; }
        public virtual Subject Subject { get; set; }
        [Display(Name = "קוד נושא")]
        [Required, DatabaseGenerated(DatabaseGeneratedOption.None)]
        public int SubjectId { get; set; }
        public virtual User Teacher { get; set; }
        [Required]
        [Display(Name = "קוד מורה")]
        public int UserId { get; set; }
        [Required]
        [Display(Name = "מחיר")]
        public double Price { get; set; }
        [Required]
        [Display(Name = "עיר")]
        public string City { get; set; }
        [Required]
        [Display(Name = "רמת ביקום")]
        public int QuestLevel { get; set; }
    }
}
```

- **Schedule (שיעור)**

מחלקה לקביעת שיעור (טרם ביצוע השיעור).

```
namespace Dal
{
    public class Schedule
    {
        [Key, Display(Name = "קוד")]
        public int Id { get; set; }
        [Required]
        [Display(Name = "קוד מורה")]
        public int TeacherId { get; set; }
        public virtual User Teacher { get; set; }
        [Required]
        [Display(Name = "קוד תלמידה")]
        public int StudentId { get; set; }
        public virtual User Student { get; set; }
        [Display(Name = "יום בשבוע")]
        public string WeekDay { get; set; } //or
        [Display(Name = "תאריך")]
        public DateTime? Date { get; set; } //or
        [Required, Display(Name = "חשעה")]
        public double FromTime { get; set; }
        [Required, Display(Name = "עד שעה")]
        public double ToTime { get; set; }
        //[Required]
        //public int Hour { get; set; }
        [Required]
        [Display(Name = "קוד מקצוע")]
        public int SubjectId { get; set; }
        public Subject Subject { get; set; }
    }
}
```


- **Recommendation (המלצה למורה)**
מחלקה זו מייצגת את המלצת התלמיד על המורה.

```
namespace Dal
{
    public class Recommendation
    {
        [Key]
        [Display(Name = "קוד המלצה")]
        public int RecommendationId { get; set; }
        public virtual User Teacher { get; set; }
        [Required]
        [Display(Name = "קוד מורה")]
        public int TeacherId { get; set; }
        public virtual User Student { get; set; }
        [Required]
        [Display(Name = "קוד תלמיד")]
        public int StudentId { get; set; }
        [Required]
        [Display(Name = "קוד מקצוע")]
        public int SubjectId { get; set; }
        public virtual Subject Subject { get; set; }
        [Required]
        [Display(Name = "הערות")]
        public string Comments { get; set; } //הערות
        [Display(Name = "כוכבים")]
        public int Stars { get; set; }
        public override string ToString()
        {
            return Comments+"/"+ Stars;
        }
    }
}
```

- **File (קובץ)**
מחלקה הכוללת פרטי קובץ (חומר לימודי) אותו המורה מעלה לאתר.

```
namespace Dal
{
    public class File
    {
        [Key]
        [Display(Name = "קוד קובץ")]
        public int Id { get; set; }
        [Display(Name = "קוד שיעור")]
        public int LessonId { get; set; }
        public virtual Lesson Lesson { get; set; }
        [Display(Name = "שם קובץ לתצוגה")]
        public string DisplayName { get; set; }
        [Display(Name = "שם קובץ")]
        public string LessonTeacherId { get; set; } //שם הקובץ יהיה בנוי מקוד מורה+קוד שיעור
    }
}
```

- **Lesson (שיעור)**
מחלקה לתיעוד שיעור שעבר.

```
namespace Dal
{
    public class Lesson
    {
        [Key]
        [Display(Name = "קוד שיעור שנעשה")]
        public int Id { get; set; }
        [Required]
        [Display(Name = "קוד שיעור שנקבע")]
        public int ScheduleId { get; set; }
        public virtual Schedule schedule { get; set; }
        [Required]
        [Display(Name = "תאריך")]
        public DateTime Date { get; set; }
        [Required]
        [Display(Name = "תיעוד")]
        public string Documentation { get; set; } //תיעוד
        [Display(Name = "שעור בית")]
        public string Homework { get; set; }
    }
}
```

• DB (בסיס נתונים)

מחלקה הכוללת את כל המחלקות ועל ידה נוצר בסיס הנתונים ובעזרת מופע שלה ניתן לגשת אל בסיס הנתונים.

```
namespace Dal
{
    public class Db:DbContext
    {
        public Db():
            base(@"Data Source = (localdb)\MSSQLLocalDB;Initial Catalog = MoreshetNew; Integrated Security = True;
                Connect Timeout = 30; Encrypt=False;TrustServerCertificate=True;ApplicationIntent=ReadWrite;
                MultiSubnetFailover=False")
        {
        }

        public virtual DbSet<CancelALesson> CancelALessonT { get; set; }
        public virtual DbSet<ChatMessage> ChatMessageT { get; set; }
        public virtual DbSet<Definetion> DefinetionsT { get; set; }
        public virtual DbSet<Lesson> LessonsT { get; set; }
        public virtual DbSet<PossibleTime> PossibleTimesT { get; set; }
        public virtual DbSet<Recommendation> RecommendationsT { get; set; }
        public virtual DbSet<Schedule> SchedulesT { get; set; }
        public virtual DbSet<Subject> SubjectsT { get; set; }
        public virtual DbSet<User> UsersT { get; set; }
        public virtual DbSet<VacationDatesForTeacher> VacationDatesForTeacherT { get; set; }

        public System.Data.Entity.DbSet<Dal.File> Files { get; set; }
        public System.Data.Entity.DbSet<Dal.SubjectsPerTeacher> SubjectsPerTeachers { get; set; }
    }
}
```

:Controllers

- SearchController (חיפוש המורה):
ב-controller זה מוחזר partial view הכולל חיפוש מורה לפי קריטריונים מסוימים.

```
namespace Moreshet.Controllers
{
    public class SearchController : Controller
    {
        private Db db = new Db();
        // GET: Search
        /// <summary> partial view for the search
        public ActionResult SearchPartialView()...

    }
}
```

- RecommendationController
(המלצת תלמיד למורה):

```
namespace Moreshet.Controllers
{
    public class RecommendationsController : Controller
    {
        private Db db = new Db();
        // GET: Recommendations/Create
        /// <summary> view to Add Recommendation to teacher-empty
        public ActionResult Create()...
        /// <summary> add recommendation to teacher-post- get values from view
        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<ActionResult> Create( Moreshet.Models.RecommendationModel recommendation)...
```

- PrintController (הדפסת פרטי מורה):

```
namespace Moreshet.Controllers
{
    public class PrintController : Controller
    {
        Dal.Db ctx;
        /// <summary> ctor
        public PrintController()...
        /// <summary> view with teacher details table
        public ActionResult IndexById(int id)...)
        /// <summary> print Teacher Details- convert to pdf file
        public ActionResult PrintTeacher(int id)...
```

- HomeController (בית):
ב-controller זה מוזכרים דף הבית ודף האודות.

```
namespace Moreshet.Controllers
{
    public class HomeController : Controller
    {
        /// <summary> The first view
        public ActionResult Index()...
        /// <summary> About view
        public ActionResult About()...
    }
}
```

- FirstController (ראשי):
ה-controller הראשי בו מתבצעות פונקציות רבות.

```
public class FirstController : Controller
{
    Del.Db db = new Db();
    string loginName;
    string UserPassword;
    List<Subject> listSubjects = new List<Subject>();
    List<User> listFind = new List<User>();
    User user;
    // GET: First ...
    /// <summary> Open the first view to guest, teacher or student
    public ActionResult Open()...
    /// <summary> return view with teacher list after a search
    public async Task<ActionResult> TeachersDetails(Moreshet.Models.SearchModel Search)...
    /// <summary> Order teacher list from the first view
    public ActionResult OrderBy()...
    /// <summary> function that order teacher list from the first view
    public List<User> SortTeachersByPrice( List<User> listUsers) ...

    // public ActionResult DoALesson(int? id,string sub) ...
    /// <summary> error view to guest
    public ActionResult ErrorMessageForGuest()...
    /// <summary> get list to add schedules and another list to Delete schedules from xml file
    List<string> listTimesLesson = new List<string>();
    public ActionResult GetTimesArray(List<string> TimesArray, List<string> DeleteArrayXml) ...
    /// הספת מקצוע למורה מסוימת ...
    /// <summary> Add Subject For Teacher- schedules times
    public ActionResult AddSubjectForTeacher(int? id) ...

    /// <summary> To get out from account
    public ActionResult GetOut()...
    /// <summary> function return Password- 8 digits
    public string Get8Digits()...
    /// <summary> Autocomplete Subject textbox
    public JsonResult Autocomplete( string term = "")...
    /// <summary> Autocomplete TeacherName textbox
    public JsonResult AutocompleteTeacher(string term = "")...
    /// <summary> Autocomplete to City textbox
    public JsonResult AutocompleteCity(string term = "")...
    /// <summary> Save file to the project
    [HttpPost]
    public ActionResult File(HttpPostedFileBase file) ...
}
```

- **EmailController (מייל):**
שליחת דואר אלקטרוני.

```
namespace Moreshet.Controllers
{
    public class EmailController : Controller
    {
        Dal.Db db = new Db();
        /// <summary> Contact to Moreshet
        public ActionResult ContactNow()...
        /// <summary> Send mail to teacher with list cancel lessons after change time scedule
        public ActionResult SendMailToTeacher(int idTeacher, List<string> ListMailStudent)...
        /// <summary> Send Email to Student After teacher cancel a lesson
        public ActionResult SendMailToStudent(int idTeacher, int StudentId, string Date, string Time, string Subject)...
        /// <summary> public function to send email
        public static class SendEmailpublic...
    }
}
```

- **CalendarController (לוח שנה):**
לוח שנה בו מופיעים השיעורים שנקבעו.

```
namespace Moreshet.Controllers
{
    public class CalendarController : Controller
    {
        /// <summary> First Instanse for Calendar
        public ActionResult Overview2(int? month,int? year,int? id,int? idSubject)...
        /// <summary> Remove A lesson
        public ActionResult RemoveALesson(string DeleteLesson)...
        /// <summary> Add to database lessons
        public ActionResult GetArrayAjaxDoLessons(List<string> DoALessonsArray)...
        /// <summary> Print The Calendar (convert to pdf file)
        public ActionResult PrintCalendar(int? month, int? year, int? id, int? idSubject)...
    }
}
```

• UserController (משתמש):

יצירת, עריכת ומחיקת סוגי המשתמשים ועוד...

```

class UserController : Controller
{
private DB db = new DB();

// GET: users ...

// summary: view to return user details
public async Task

```

הפונקציה ChangeRadioLengthHour:

פונקציה זו נכתבה ב-javascript, ומקושרת למסך השעות שלי למורה. פונקציה זו מקבלת את בחירת המשתמש (המורה) לקביעת שיעור בין 45 דקות או שמא 60 דקות. לפי בחירה זו ולפי הבחירות הקודמות אליה- בחירת שעת התחלה ובחירת שעת סיום, הפונקציה מחשבת מס' שיעורים (הנכנסים בין שעת ההתחלה לסיום) ומכניסה לטבלה הנוצרת דרך קוד הפונקציה את שורות השעות המחושבות.

```
//function on change radio button to choose between lesson 45 minutes or 60 minutes
//after choose this function calculate
//A clock system built from the same hours The function calculates from the start
//time until the end time according to the length of the lesson that the teacher chose
function ChangeRadioLengthHour(valu)
{
    $('#r1')[0].disabled = true;
    $('#r2')[0].disabled = true;
    var count=finalHour-Firsthour;
    radio=parseInt(valu)
    if(radio==45)
        //The number of lessons increases if the lesson is 45 minutes
        count+=parseInt(count/3)
        //If there is another hour of study, add to count
        var minCount=finalHour2+Firsthour2
        if(minCount>=radio)
            count+=parseInt(minCount/radio);
        //Time2=the time in minutes
        var index= $('#TeHours').children.length;
        if(count==0)
            alert("אין שעות לטווח שעות זה")
        else if(count<0)
        {
            alert("טעות בהכנסת הנתונים")
        }
        else
        {
            $('#hideRow')[0].style.visibility="visible"
            for (var i = 0; i <count; i++) {
                for (var j = 0; j <=6; j++) {
                    var newel = document.createElement('td');
                    if(j==0)
                    {
                        //Current end time in minutes
                        var toTime=Firsthour*60+Firsthour2+radio
                        var fin=finalHour*60+finalHour2
                        if(toTime>fin)
                            break;
                        var newTr= document.createElement('tr');
                        var div=document.createElement('div');
                        div.className='fc-event';
                        var span=document.createElement('span');
                        span.className='fc-title';
                        var span2=document.createElement('span');
                        span2.lang='he';
                        span2.dir='rtl';
                        Firsthour=toTime/60
                        //Getting decimal minutes
                        Firsthour2=Firsthour-parseInt(Firsthour)
                        Firsthour=parseInt(Firsthour)
                        Firsthour2*=60
```

```

Firsthour2+=0.01
Firsthour2=parseInt(Firsthour2)
//Calculation performed:
span2.innerText=fromTime+"-"+Firsthour+":"+Firsthour2
fromTime=Firsthour+":"+Firsthour2
span.appendChild(span2)
div.appendChild(span)
newel.appendChild(div)
newTr.appendChild(newel)
$('#TeHours').append(newTr)
}
else
{
if(j!=6||(j==6&&parseInt(fromTime.split(':')[0])<14))
{
var input=document.createElement('input');
input.type="checkbox"
input.addEventListener("change", function(){
    ChangeValCheckBox(this)
});
newel.appendChild(input)
}
newel.id=j.toString()+i.toString()
//add td to tr in the table
newTr.appendChild(newel)
}
}
}
$('#search')[0].style.visibility="visible"
var heightTable=$('#TeHours').height();
$('#search')[0].style.top=(parseInt($('#TeHours')[0].style.top)+heightTable+10).toString()+"px"; }

```


הפונקציה ChangeValCheckBox:

פונקציה זו נכתבה ב-javascript . ומקושרת למסך השעות שלי למורה. פונקציה זו מקבלת ב-onchange של ה-checkbox את האלמנט הנוכחי. (כל תיבת checkbox מסמלת בחירת שיעור ביום מסוים ובשעה מסוימת).
פונקציה זו בודקת האם שעת השיעור מסתדרת עם מערכת השעות של המורה הנוכחית (בדיקת חפיפה)
שעות והאם השעה קיימת במערכת, וכן הפונקציה מקבלת ביטולי שעות ובאם ישנם שיעורים שתלמידים קבעו על שעות אלו הפונקציה שולחת מייל למורה ולתלמיד בהתאם.

```
//This function checks whether the time of the lesson is consistent with the current teacher's schedule (overlap test)
//Hours and whether the time exists in the system), as well as the function receives cancellations of hours and if there
//are lessons that students have determined on these hours the function sends an email to the teacher and to the student accordingly.
```

```
function ChangeValCheckBox(ThisCheckBox)
{
    //Receiving the day of the week
    var indexCell= ThisCheckBox.parentNode.cellIndex;
    //marked
    var ezer=true
    var checked1=ThisCheckBox.checked
    //Accept the index row in the table
    var indexRow=ThisCheckBox.parentNode.parentElement.rowIndex
    var t
    var TimeForLesson
    if(degel3==true)
    {
        t= $('#tableSch > tbody > tr').eq(indexRow)
        if(parseFloat(t[0].cells[1].children[0].children[0].innerText.split('-')[1].replace(":", "."))-
        parseFloat(t[0].cells[1].children[0].children[0].innerText.split('-')[0].replace(":", "."))<1)
            radio=45
        else
            radio=60
        //Receiving lesson time
        TimeForLesson =t[0].cells[1].children[0].children[0].innerText + " " +(parseInt(indexCell)-1).toString();
    }
    else
    {
        t= $('#TeHours > tbody > tr').eq(indexRow)
        TimeForLesson =t[0].cells[0].children[0].children[0].innerText + " " +indexCell;
    }

    if (checked1.toString()=="true")
    {
        if(ListXmlFile!=null)
        {
            for (var h = 0; h < ListXmlFile.length; h++)
            {
                if(parseInt(ListXmlFile[h].Teacher["@id"])==session)
                {
                    for (var n = 0; n < ListXmlFile[h].Teacher.timeOfLesson.length; n++) {
                        if(ListXmlFile[h].Teacher.timeOfLesson[n]["@dayOfWeek"]==TimeForLesson.split(" ")[1]
                        //hours
                        &&ListXmlFile[h].Teacher.timeOfLesson[n]["@fromTime"].split(':')[0]==TimeForLesson.split(" ")[0].split(':')[0])
                        {
                            //Minutes
                            //The time I want to insert before an xml hour

```

```

if(parseInt(ListXmlFile[h].Teacher.timeOfLesson[n]["@@fromTime"].split(':')[1])>
parseInt(TimeForLesson.split(" ")[0].split(':')[1].split('-')[0])
)
{
if((parseInt(ListXmlFile[h].Teacher.timeOfLesson[n]["@@fromTime"].split(':')[1])-
parseInt(TimeForLesson.split(" ")[0].split(':')[1].split('-')[0])>=radio))
{
continue;
}
else
{
ezer=false;
ThisCheckBox.checked=false
alert("שעות אלו קיימות במערכת")
break
}
}
else
{
//The time I want to insert after an xml hour
if((parseInt(TimeForLesson.split(" ")[0].split(':')[1].split('-')[0])-
parseInt(ListXmlFile[h].Teacher.timeOfLesson[n]["@@fromTime"].split(':')[1])>=radio))
{
continue
}
else
{
ezer=false
ThisCheckBox.checked=false
alert("שעות אלו קיימות במערכת")
break
}
}
}
}
}
}

if(ezer==true)
TimesArray.push(TimeForLesson);
else
ezer=true

}
else
{
var i = TimesArray.indexOf(TimeForLesson);
if (i != -1)
{
TimesArray.splice(i, 1);
}
else
{
//Option to delete the prescribed lessons in the xml file
for (var h = 0; h < ListXmlFile.length; h++)
{
for (var n = 0; n < ListXmlFile[h].Teacher.timeOfLesson.length; n++) {
if(ListXmlFile[h].Teacher.timeOfLesson[n]["@@dayOfWeek"]==TimeForLesson.split("
")[1]&&ListXmlFile[h].Teacher.timeOfLesson[n]["@@fromTime"]==
TimeForLesson.split(" ")[0].split('-
')[0]&&ListXmlFile[h].Teacher.timeOfLesson[n]["@@toTime"]==TimeForLesson.split(" ")[0].split('-')[1])
{
for (var k = 0; k < listSchedule.length; k++) {
var fromTimeHour=parseFloat(listSchedule[k].FromTime)

```

```

var toTime=listSchedule[k].ToTime
var teacherId=listSchedule[k].TeacherId
Firstt= $('#hideRow2')
var day= Firstt[0].cells[indexCell].innerText
if(@((Session["user"] as Dal.User).UserId)==teacherId)
{
if(listSchedule[k].WeekDay==day)
{
if((parseInt(fromTimeHour)).toString()==TimeForLesson.split(":")[0]&&(parseInt(toTime)).toString()==
TimeForLesson.split(" ")[0].split("-")[1].split(":")[0])
{
var r = confirm("שיעור במועד זה. האם ברצונך לבטל?");
if (r == true) {
var d=listSchedule[k].Date.split('T')[0]
var ti=TimeForLesson.split(' ')[0]
$.ajax({
type: "POST",
url: "@Url.Action('SendMailToStudent', 'Email')",
datatype: "json",
traditional: true,
data: { 'idTeacher':@((Session["user"] as Dal.User).UserId),
'StudentId':listSchedule[k].StudentId,
'Date':d,
'Time':ti,
'Subject':listSchedule[k].Subject.SubjectName,
},
success: function (data) {
if(data == "success")
{
if(x==false)
alert("נשלח כרגע מייל לתלמיד כי שיעור זה בוטל, צור קשר עם התלמיד כדי לוודא שאכן מודע לשינוי. למחיקה");
x=true
ListStudentForMail.push(listSchedule[k]);
}
else if(data=="Error")
{
alert("המחיקה נכשלה");
}
},
complete: function (data) {
DeleteArrayXml.push(TimeForLesson.split(" "));
}
});
}
else
{
ThisCheckBox.checked=true
}
}
}
else
{
DeleteArrayXml.push(TimeForLesson.split(" "));
}
}
}
}
}
}
}
}
if(ListStudentForMail!=null)
{

```

```

$.ajax({
type: "POST",
url: "@Url.Action("SendMailToTeacher", "Email")",
datatype: "json",
traditional: true,
data: { 'idTeacher':@((Session["user"] as Dal.User).UserId),
'ListMailStudent':ListStudentForMail,
},
success: function (data) {
if(data == "success")
{
alert("נשלח לך מייל מפורט עם רשימת התלמידים להם בוטל שיעור");
}
}
});
}
}
}
}
{

```

הפונקציה Drop:

פונקציה זו נכתבה ב-javascript. מקושרת לדף לוח השנה שלי. כשתלמיד נכנס ללוח השיעורים שלו. ניתנת לו אופציה לגרירת שיעור אותו הוא מעוניין. בגרירת השיעור הארוע מגיע לפונקציה זו. הפונקציה מחשבת ובודקת האם השיעור יכול להתקיים בזמן שאליו גרר המשתמש את השיעור. השיעור נקבע אך ורק אם עבר תנאים מסויימים: לדוגמה: לא בשבת ולא בחגים, השיעור קיים במערכת השעות של המורה המבוקשת, השיעור פנוי ולא נתפס ע"י משתמש אחר וכו'...

//The function calculates and checks whether the lesson can exist at the time the user dragged the lesson. The lesson is determined only if certain conditions have been met:

//For example, not on Shabbat or holidays, the lesson is in the teacher's schedule, the lesson is free and not perceived by another user

```
function Drop(ev)
{
    eventTd=ev.path[0];
    var userId=@((Session["user"] as Dal.User).UserId);
    var day=parseInt(ev.path[0].children[3].innerHTML)+1;
    var dateOfDay=parseInt(ev.path[0].children[0].innerText);
    if(ev.path[0].children[2].attributes.style.nodeValue=="visibility:collapse"||(ev.path[0].children[2].attributes.style.no
deValue=="visibility:visible;display:block"&&(ev.path[0].children[2].innerText.substring(3,0)=="יום"||
ev.path[0].children[2].innerText.substring(4,0)=="אסרו"||ev.path[0].children[2].innerText.substring(3,0)=="ט"ו"||e
v.path[0].children[2].innerText.substring(3,0)=="ל"ג"||ev.path[0].children[2].innerText.substring(5,0)=="חנוכה"))
    {
        if(ListXmlFile.Teacher["@@id"]==TeacherId.split(" ")[0])
        {
            for (var m = 0; m < ListXmlFile.Teacher.timeOfLesson.length; m++)
            {
                if(ListXmlFile.Teacher.timeOfLesson[m]["@@dayOfWeek"]==day)
                {
                    //Testing with these hours is taken by another student in the system
                    for (var n = 0; n < listSchedule.length; n++) {
                        var yearFromList=listSchedule[n].Date.split('-')[0];
                        var monthFromList=listSchedule[n].Date.split('-')[1];
                        var dayFormList=listSchedule[n].Date.split('-')[2].split('T')[0];
                        var monthOfDay= parseInt(@ViewBag.Month);
                        var yearOfDay=parseInt(@ViewBag.ListDays[0].Year);
                        monthFromList=parseInt(monthFromList)
                        dayFormList=parseInt(dayFormList)
                        var fintH= parseInt(listSchedule[n].FromTime)
                        var fDouM=listSchedule[n].FromTime-fintH
                        fDouM*=60;
                        if(fDouM==0)
                        {
                            var fromTime=fintH+":"+fDouM+0;
                        }
                        else
                        {
                            var fromTime=fintH+":"+fDouM;
                        }
                    }
                    var tintH= parseInt(listSchedule[n].ToTime)
                    var tDouM=listSchedule[n].ToTime-tintH
                    tDouM*=60;
                    var toTime=tintH+":"+tDouM;
                    if(listSchedule[n].TeacherId==TeacherId.split(" ")[0]&&
listSchedule[n].StudentId==userId&&dayFormList==dateOfDay&&monthFromList==monthOfDay&&yearFromList=
=yearOfDay&&fromTime==ListXmlFile.Teacher.timeOfLesson[m]["@@fromTime"]&&toTime==ListXmlFile.Teacher.
timeOfLesson[m]["@@toTime"])
                    {
                        finalArray.push(ListXmlFile.Teacher.timeOfLesson[m]["@@fromTime"]+"-
"+ListXmlFile.Teacher.timeOfLesson[m]["@@toTime"]+"-תפוס")
                        continue;
                    }
                }
            }
        }
    }
}
```

```

}
else
{
var i = finalArray.indexOf(ListXmlFile.Teacher.timeOfLesson[m][["@@fromTime"]+"-"+
ListXmlFile.Teacher.timeOfLesson[m][["@@toTime"]]);
if (i == -1)
{
finalArray.push(ListXmlFile.Teacher.timeOfLesson[m][["@@fromTime"]+"-"+
ListXmlFile.Teacher.timeOfLesson[m][["@@toTime"]])
}
}
};
}
}
}

if(finalArray.length!=0)
{
for (var i = 0; i < finalArray.length; i++)
{

var newTr= document.createElement('tr');
var newel = document.createElement('td');
var div=document.createElement('div');
div.className='fc-event';
if(finalArray[i].split('-')[2]=="תפוס")
{
div.style.backgroundColor = "gray";
div.style.border = "1px solid gray";
}

div.addEventListener("click", function(){
if(this.children[0].innerText.split('-')[2]=="תפוס")
{
alert("שיעור זה תפוס")
}
else{
if(this.style.backgroundColor != "gray"&&this.style.border != "1px solid gray")
{
DoALessonsArray.push(dateOfDay+" "+this.innerText+" "+TeacherId.split(" ")[0]+" "+Subject+" "+day+"
"+TeacherId.split(" ")[1])
this.style.backgroundColor = "gray";
this.style.border = "1px solid gray";
}
else
{
var i = DoALessonsArray.indexOf(dateOfDay+" "+this.innerText+" "+TeacherId.split(" ")[0]+" "+Subject+"
"+day+" "+TeacherId.split(" ")[1]);
if (i != -1)
{
DoALessonsArray.splice(i, 1);
this.style.backgroundColor = "#3a87ad";
this.style.border="1px solid #3a87ad";
}
}
});
var span=document.createElement('span');
span.className='fc-title';

var span2=document.createElement('span');
span2.lang='he';

```

```

span2.dir='rtl';
span2.innerText=finalArray[i];
span.appendChild(span2)
div.appendChild(span)
newel.appendChild(div)
newTr.appendChild(newel)
$('#table').append(newTr)
}
$("#ModelShowHours").modal('show');
finalArray=[]
}
else
{ if(ev.path[0].children[3].innerHTML=="6")
alert("אנו שומרים שבת!");
else
alert("למורה זו אין שעות ביום זה")
}

}
else
{
alert("לא ניתן לקבוע שיעור ביום מיוחד זה");
}

```

}

3. מדריך למשתמש:

2.1 הוראות כלליות לשימוש באתר:

מדריך למשתמש:

- מדריך לאורח

בעת כניסת אורח לאתר יש ביכולתו לבצע מספר פעולות אפשריות אך מוגבלות: האורח יכול לערוך חיפוש מורה באתר על סמך הקריטריונים הנתונים לדוגמא: מקצוע, מחיר, מיקום וכו' ויקבל את רשימת המורות בהתאם לקריטריונים שמילא, אך לא תהיה עבורו האפשרות לקבוע שיעור עם המורה או, לשוחח עמה בצא'ט ואו לראות הפרטים האישיים שלה כמו כתובת, כתובת מייל וכו' אלא רק את שמה, המלצות ומספר כוכבים שקיבלה מתלמידים אחרים ומחיר לשעה. באפשרותו של האורח לגשת לאודות האתר. האורח יכול אף ליצור קשר עם מנהל האתר על מנת לדווח על עניין שמרגיש לנכון לעדכן את המנהל. האורח יכול להירשם לאתר כמורה או כתלמיד.

- מדריך למורה:

ממשקים למורה במערכת – המורה יכולה לבטל שיעור שנקבע עם תלמיד, לערוך את השעות השבועיות הקבועות שלה, להכניס נושאים בהם הינה מאפשרת שיעור פרטי, לתעד שיעור שנעשה, לעלות קבצי העשרה וכו' עבור התלמיד, לערוך את הפרטים האישיים שלה בפרופיל.

- מדריך לתלמיד:

ממשקים לתלמיד במערכת – התלמיד רשאי לבצע חיפוש, לצפות בנתוני מורה אישיים, לקבוע שיעור ולשבצו במערכת אישית שלו, לבצע תשלום למורה על שיעור שבוצע, בנוסף – התלמיד יכול לשנות ולעדכן פרטים אישיים שלו במערכת.

2.2 מסכים:

באתרנו 3 משתמשים אופציונאליים:

1. אורח מזדמן –

היכנס לאתר, חפש מורה לפי הקריטריונים המופיעים,

לחץ על כפתור "הרשם/הכנס" על מנת להירשם לאתר כמשתמש חדש – מורה או תלמיד,
או צפה בתוצאות.

שם פרטי	שם משפחה	כתובת	המלצות	כוכבים	מחיר לשעה	מייל
אופירה	סולטן	עיר רחובות רחוב רש"י מספר בית 13			ofirarachel@gmail.com	לפרטים לקביעת שיעור
מיכל	קובר	עיר קרית יערים רחוב גורדון מספר בית 5			micalbkuber@gmail.com	לפרטים לקביעת שיעור
חני	שימלמן	עיר ירושלים רחוב הקבלן מספר בית 55			efshim@gmail.com	לפרטים לקביעת שיעור

הכנס אם ברשותך משתמש קיים. וכן, ישנה אפשרות להירשם:

הדף שמתקבל:
















באם הינך מעוניין להרשם כתלמיד לחץ על הקישור הרשם כתלמיד:

ובאם הינך מעוניין להירשם כמורה לחץ על הקישור הירשם כמורה:

בית אודות צור קשר הרשם/הכנס



הוספת מורה חדש

 כתובת מייל	 תעודת זהות
 סיסמה	 שם משתמש
 אימות סיסמה	 שם פרטי
 השכלה ותארים	 שם משפחה
 נסיון וותק	 עיר
 אג'נדה מקצועית וחיסוכית	 רחוב
 לחץ להבאת תמונה	 מספר בית
	 טלפון

קו"ח המלצות צור

2. תלמיד במערכת –

בעת כניסתך לאתר בתור תלמיד תוכל לערוך חיפוש של מורה על פי קריטריונים נבחרים,

לחיפוש

חפשו

עד מחיר

ממחיר

אזור

שם מורה

מקצוע

בעת לחיצה על "חיפוש" תוכל לראות רשימה של כל המורות העונות על הקריטריונים שהכנסת.

על מנת לקבוע עם המורה שיעור יש להקיש על מקש "לקביעת שיעור" המופיע בשורת המורה הרצויה.

פרטי מורות:

שם פרטי	שם משפחה	כתובת	המלצות	כוכבים	מחיר לשעה	מייל
אופירה	סולטן	עיר רחובות רחוב רשיי מספר בית 13			ofirarachel@gmail.com	לפרטים לקביעת שיעור
מיכל	קובר	עיר קרית יערים רחוב גורדון מספר בית 5			micalbkuber@gmail.com	לפרטים לקביעת שיעור
חני	שימלמן	עיר ירושלים רחוב הקבלן מספר בית 55			efshim@gmail.com	לפרטים לקביעת שיעור

מיון לפי מחיר

לקביעת שיעור

על מנת להוסיף המלצה למורה יש להקיש על הכפתור "להוספת המלצה למורה" המופיע בדף הבית.

בית אודות צור קשר לוח שיעורים צא פרופיל

מורה מנחם בן-דוד

אם ראית תלמיד שלימודו קשה עליו כברזל בשביל רבו שאינו מסביר לו פנים

חפש

עד מחיר ממחיר אזור שם מורה מקצוע

המועדים שלי

להוספת המלצה למורה

מיון לפי מחיר

נפתח חלון הוספת המלצה למורה:

בית אודות צור קשר לוח שיעורים צא פרופיל

מורה מנחם בן-דוד

הוספת המלצה למורה

שם מורה

מקצוע

טקסט

סוכיות

צור

לחץ על כפתור "לוח שיעורים" על מנת לראות את לוח השיעורים האישי שלך בתוך לוח השנה.



בית אודות צור קשר לוח שיעורים צא פרופיל

למערכת לפי מקצוע למערכת לפי מורה הדפס הודד

הבא > ספטמבר אוקטובר נובמבר דצמבר ינואר פברואר מרץ אפריל מאי יוני יולי אוגוסט

ינואר 2018

ראשון	שני	שלישי	רביעי	חמישי	שישי	שבת
	1 י"ד טבת, תשע"ח	2 ט"ו טבת, תשע"ח	3 ט"ז טבת, תשע"ח	4 י"ז טבת, תשע"ח	5 י"ח טבת, תשע"ח	6 י"ט טבת, תשע"ח
7 כ" טבת, תשע"ח	8 כ"א טבת, תשע"ח	9 כ"ב טבת, תשע"ח	10 כ"ג טבת, תשע"ח	11 כ"ד טבת, תשע"ח	12 כ"ה טבת, תשע"ח	13 כ"ו טבת, תשע"ח
14 כ"ז טבת, תשע"ח	15 כ"ח טבת, תשע"ח	16 כ"ט טבת, תשע"ח	17 א' שבט, תשע"ח	18 ב' שבט, תשע"ח	19 ג' שבט, תשע"ח	20 ד' שבט, תשע"ח

3. מורה במערכת –

בעת כניסתך לאתר בתור מורה תראה מיד את לוח השיעורים שלך בלוח השנה.



בית אודות צור קשר לוח שיעורים צא פרופיל

למערכת לפי מקצוע למערכת לפי מורה הדפס הוד

הבא ספטמבר אוקטובר נובמבר דצמבר ינואר פברואר מרץ אפריל מאי יוני יולי אוגוסט

ינואר 2018

ראשון	שני	שלישי	רביעי	חמישי	שישי	שבת
	1 י"ד טבת, תשע"ח	2 ט"ו טבת, תשע"ח	3 ט"ז טבת, תשע"ח	4 י"ז טבת, תשע"ח	5 י"ח טבת, תשע"ח	6 י"ט טבת, תשע"ח
7 כ"ט טבת, תשע"ח	8 כ"א טבת, תשע"ח	9 כ"ב טבת, תשע"ח	10 כ"ג טבת, תשע"ח	11 כ"ד טבת, תשע"ח	12 כ"ה טבת, תשע"ח	13 כ"ז טבת, תשע"ח
14 כ"ז טבת, תשע"ח	15 כ"ח טבת, תשע"ח	16 כ"ט טבת, תשע"ח	17 א' שבט, תשע"ח	18 ב' שבט, תשע"ח	19 ג' שבט, תשע"ח	20 ד' שבט, תשע"ח

לחץ על כפתור "השעות שלי" על מנת לערוך את השעות בהן תוכל להעניק שיעורים פרטיים, בחר את שעת ההתחלה הרצויה, שעת הסיום ואורך שעת לימוד, ומיד תיפתח טבלה המחולקת על סמך הנתונים שהכנסת, סמן שעות רצויות, לאחר מכן לחץ על לחצן "הוסף" (המערכת תתריע על מצב חפיפה של זמנים הקיימים כבר במערכת). במידה ותרצה לערוך את השעות הקיימות במערכת לחץ על לחצן "להצגת או לעריכת השעות שלי" וכן במידה ותרצה לערוך את המקצועות הקיימים במערכת וערי הלימוד לחץ על לחצן "להוספת/ עריכת המקצועות שלי".



בית אודות צור קשר השעות שלי צא פרופיל

הוספת מערכת שיעור

הכנסו/ שעת תחילת הלימוד

הכנסו/ שעת סיום הלימוד

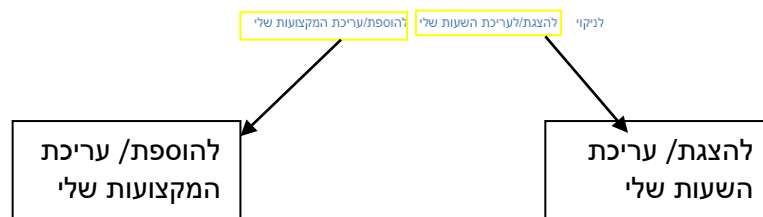
הכנסו/ אורך שעת לימוד

☒ 45 דקות

☐ 60 דקות

ראשון	שני	שלישי	רביעי	חמישי	שישי
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

הוסף



מסך הצגת/עריכת השעות שלי-
כאן מורה מוסיפה מקצועות וערי לימוד:

מסך הוספת/ עריכת המקצועות שלי:
כאן מורה יכולה להוסיף/ לבטל שעות לימוד

מערכת השעות שלי

ראשון	שני	שלישי	רביעי	חמישי	שישי
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13:10-13:55					
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
13:55-14:55					
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14:40-15:25					
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
14:55-15:55					
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
15:55-16:55					
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
16:15-17:00					
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
17:45-18:30					
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
17:55-18:55					
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
18:25-19:10					
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
18:30-19:15					
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
19:55-20:40					

כאן ניתן לבטל
שיעור ע"י
הורדת הסימן

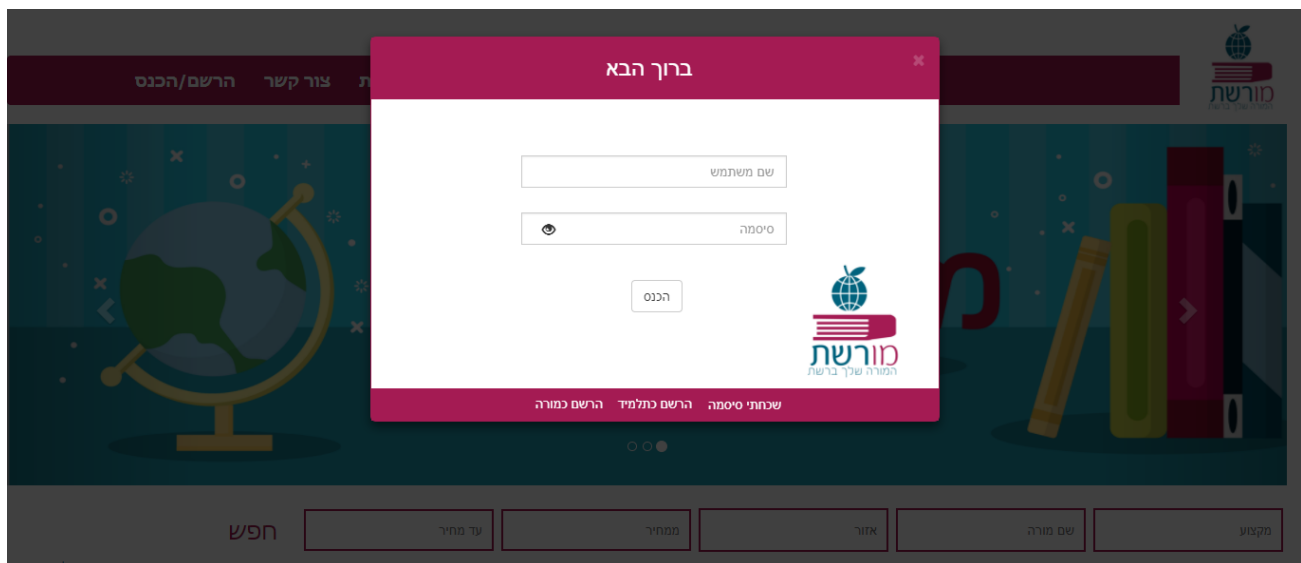
המורה יכולה לבטל שיעור שנקבע עם תלמיד, לערוך את השעות השבועיות הקבועות שלה, להכניס נושאים בהם הינה מאפשרת שיעור פרטי, לתעד שיעור שנעשה, לעלות קבצי העשרה וכו' עבור התלמיד, לערוך את הפרטים האישיים שלה בפרופיל.

בית אודות צור קשר לוח שיעורים צא פרופיל

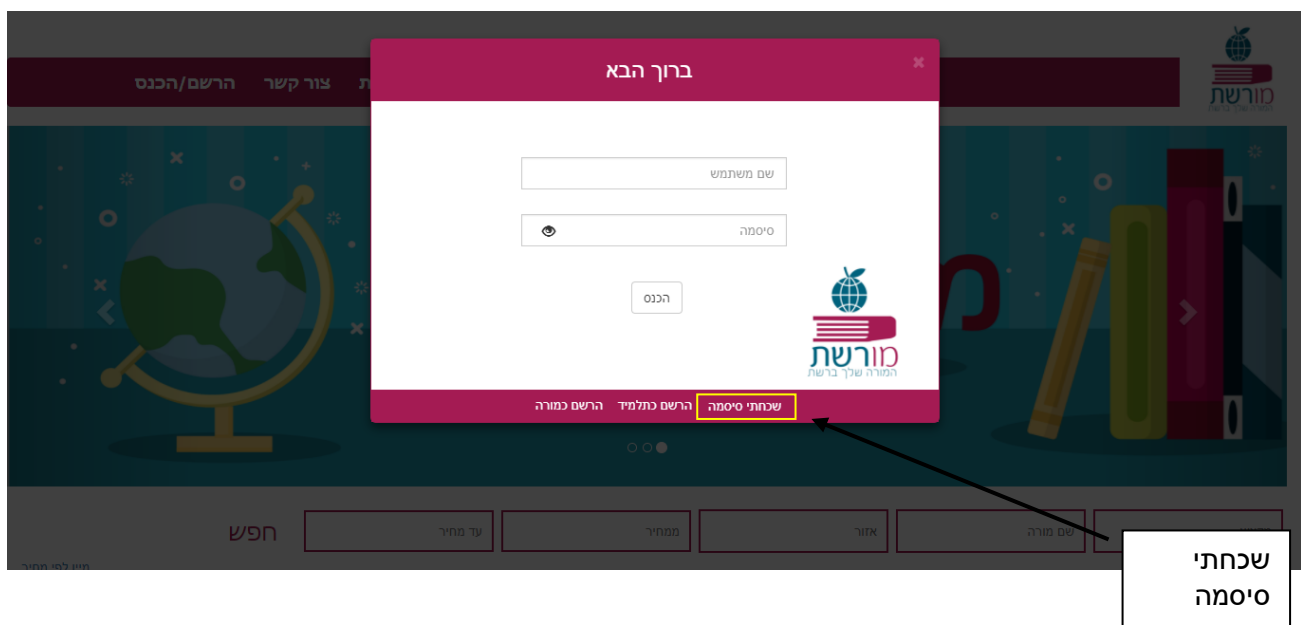
שם פרטי
שם משפחה
כתובת
מירב
שימלמן
עיר
ירושלים
רחוב
הקבלן
מספר בית
55
טלפון
0533124584
מייל
me3124584@gmail.com
תמונה

עריכה מחיקה

- משתמש במערכת (מורה או תלמיד) - היכנס לאתר לחץ על כפתור "הרשם/הכנס" על מנת להיכנס לאתר בתור משתמש קיים, מלא את שם המשתמש והסיסמא בהתאם ולחץ על הכנס.



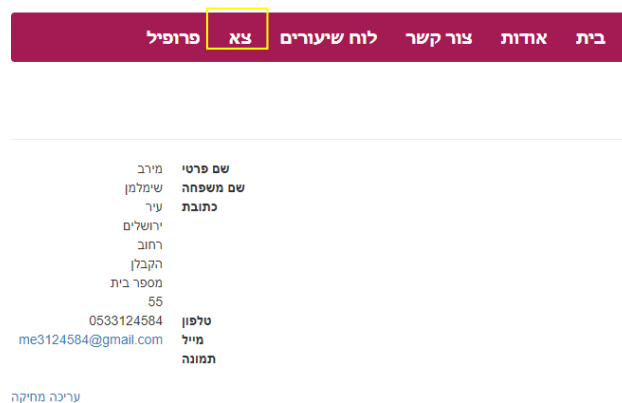
במקרה בו שכחת סיסמתך – לחץ על המקש "שכחתי סיסמא" ותישלח סיסמא חדשה למייל שלך, אתה תוכל להיכנס לאתר עבור שם המשתמש שלך במקום הסיסמא הישנה אותה שכחת,



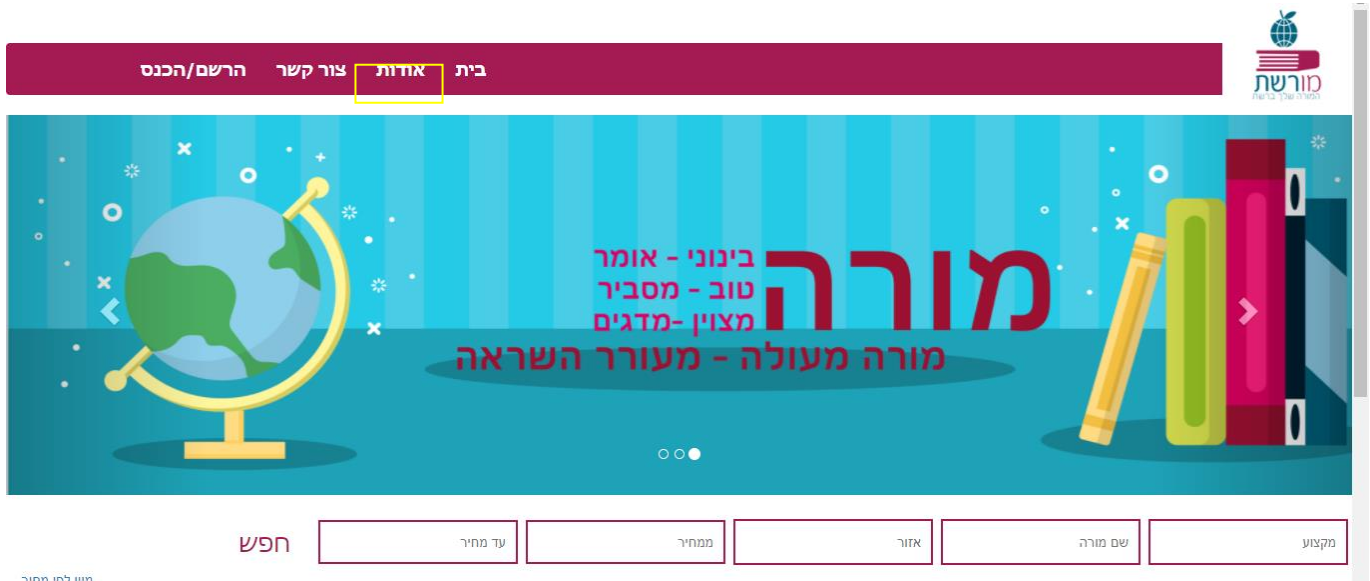
על מנת לשנות את הסיסמא שקיבלת לסיסמא לבחירתך, גש אל הכפתור "פרופיל" בו יופיעו פרטייך האישיים לחץ על כפתור "עריכה" בו תוכל לערוך את פרטייך האישיים כולל את הסיסמא. את הסיסמא תוכל לשנות רק בתנאי שתקיש את הסיסמא שאותה אתה מתכוון לשנות והיא תהיה נכונה.



לחץ על כפתור "פרופיל" בו תראה את פרטייך האישיים המופיעים באתר ותוכל לשנותם על ידי "עריכה" או למחוק את המשתמש על ידי "מחיקה". לחץ על מקש "צא" על מנת לצאת מן האתר (מעבר למצב אורח).



- כל משתמש (מורה או תלמיד או אורח) – לחץ על כפתור "אודות" על מנת לצפות באודות האתר.

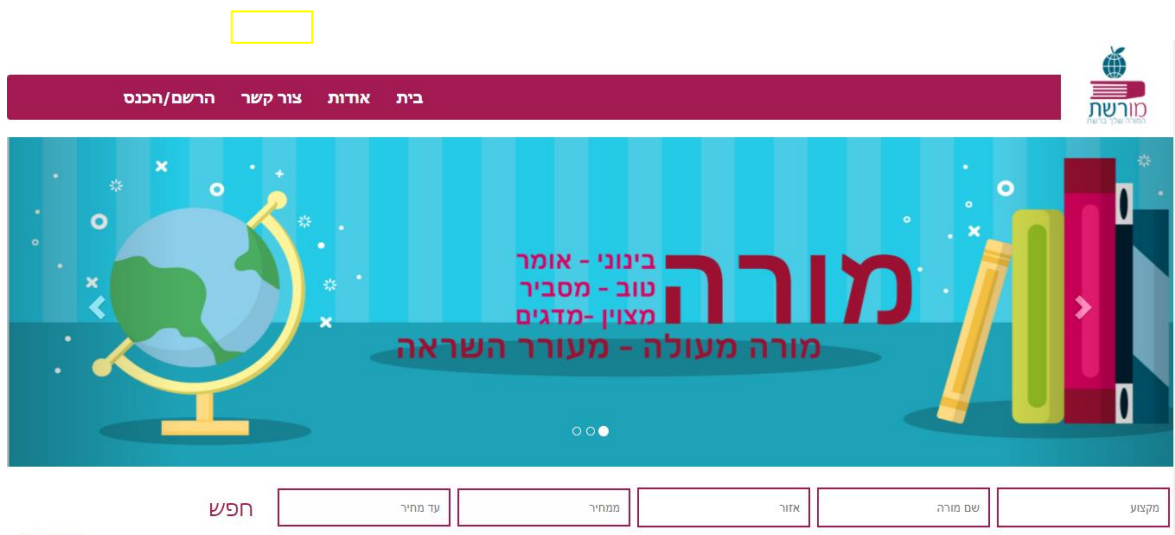


הדרך שמתקבל:



מורשת קמה מתוך הבנה כי חסר בשוק האינטרנטי אתר לשיעורים פרטיים נוח לשימוש ומהיר. מורשת הינו אתר המקיים תקשורת בין מורה לתלמידו בצורה הנוחה והאידיאלית ביותר. במורשת מושקעים משאבים אדירים בפיתוח הטכנולוגיה, ההתפתחות נמשכת כל הזמן ללא לאות, ואנו מקווים כי תהיו מרוצים...

לחץ על כפתור "צור קשר" על מנת לשלוח מייל למנהל האתר בכל נושא שהוא.



הדף שמתקבל:

צור קשר

שם משתמש

תוכן

מחק שלח

משמח להיות בקשר

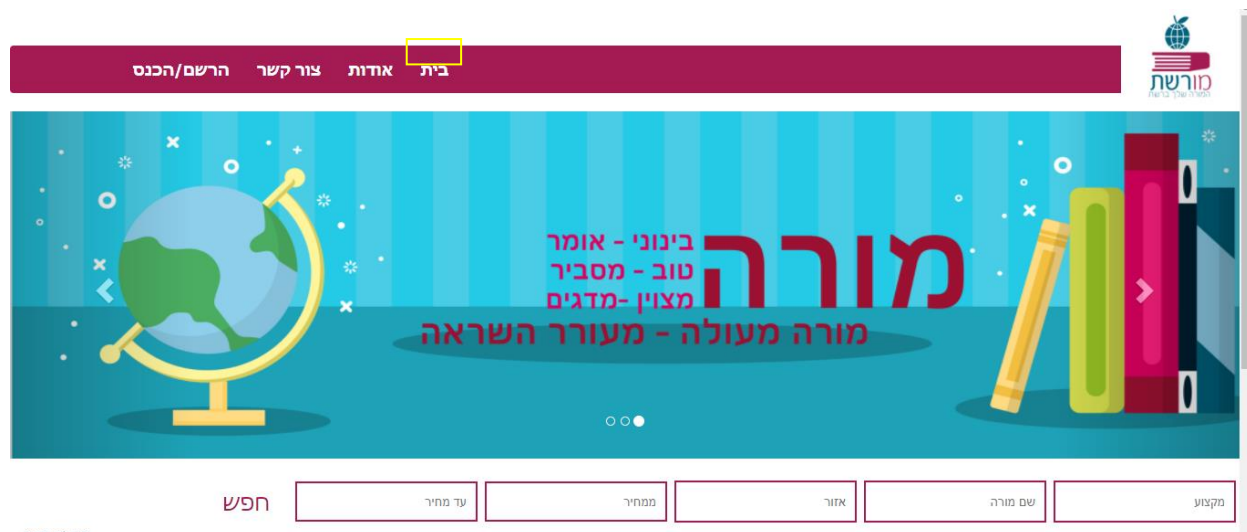
מורשת
המורה שלך ברשת

הרשם/הכנס צור קשר

שם מורה אזור ממחיר עד מחיר חפש

מיקום שלי

לחץ על כפתור "בית" על מנת לחזור לעמוד הראשי – אשר נפתח לך בעת כניסתך הראשונית.



4. סיכום ומסקנות:

אחרי עבודה רבה על האתר אנו רואות שהוא יכול לשמש לצרכים לשמם החלטנו לפתח אותו. האתר משרת מגוון רחב של משתמשים מהאוכלוסייה: מורים ותלמידים. כך יכולה להיווצר תקשורת קלה כדי לקבוע שיעורים בקלות ונוחות. למשתמש ניתנת גישה נוחה וברורה באתר למגוון אופציות: לחיפוש מורה, לכניסה ללוח שיעורים פרטי המסכם בלוח חודשי את השיעורים בצורה ברורה ומסודרת. ליצירת קשר עם מנהל האתר וכן אופציות רבות ונוחות.

דגש גדול שמנו על ברירות האתר. **האתר ברור ונח למשתמש**. ניתן להרחיבו שיהיה כלל עולמי- בשפות שונות ובארצות רבות וכן ניתן להוסיף שיעורים באונליין לתלמיד דרך האתר וכו'...

בפרויקט, הכנסנו נתונים בסיסיים לדוגמא כדי להראות את אופן השימוש באתר. כמובן שלשימוש יעיל, תצטרך מזכירה להכניס נתונים רבים.

האתר שלנו יכול להעלות לרשת האינטרנט ובכך לשרת ולהקל מעל התלמידים- חיפוש מהיר של מורים. וכן מעל המורים- סדר במערכת השעות ותפוצה רבה...

6. ביבליוגרפיה:

אתרים בנושא תכנות:

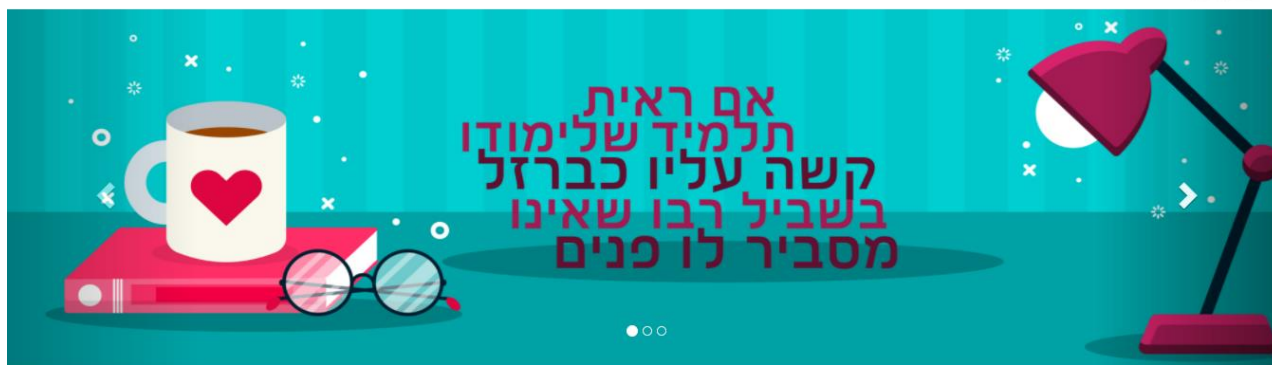
- stackoverflow.com
- webmaster.org.il
- w3schools.com

כאשר תלמיד נצרך לסיוע לימודי הוא חפץ למצוא עזרה שתתאים בצורה המקסימלית לדרישותיו. אך על מנת שאכן כל דרישותיו תתמלאנה עליו להשקיע זמן רב בבירורים – מה שלא כל אדם יוכל להרשות לעצמו, ואף לא ודאי שאכן יעלו מאמציו פירות. אי לכך ובהתאם לזאת – ראינו צורך לתכנת אתר – שבזמן מינימלי מפנה למורה המתאימה מקסימלית.

באתרנו 3 משתמשים אופציונאליים:

- אורח מזדמן – האורח יכול לבצע חיפוש והמידע שיופק מחיפושו יהיה מצומצם מאוד עם פרטים מינימליים. לא ניתנת בידו אופציה לקביעת שיעור, ולצפייה בנתוני מורה אישיים – אלו יינתנו רק בתנאי שהמשתמש הינו תלמיד במערכת).
- תלמיד במערכת – התלמיד רשאי לבצע חיפוש, לצפות בנתוני מורה אישיים, לקבוע שיעור ולשבצו במערכת אישית שלו, לבצע תשלום למורה על שיעור שבוצע, בנוסף – התלמיד יכול לשנות ולעדכן פרטים אישיים שלו במערכת.
- מורה במערכת – המורה יכולה לבטל שיעור שנקבע עם תלמיד, לערוך את השעות השבועיות הקבועות שלה, להכניס נושאים בהם הינה מאפשרת שיעור פרטי, לתעד שיעור שנעשה, לעלות קבצי העשרה וכו' עבור התלמיד, לערוך את הפרטים האישיים שלה בפרופיל.

האתר מעניק למשתמש חיפוש מורה על פי קריטריונים לבחירתו – מחיר, מיקום, והמלצות. התלמיד יכול ליצור עם המורה קשר טלפוני\ דרך האתר בצ'אט\ דרך מייל המורה. האתר נוח לשימוש, מוצג בצורה ברורה ומאירה לעיניים. האתר מקנה אפשרות לדיווח או לתקשור כללי עם מנהל האתר בלחצן "צור קשר" (נשלח מייל למנהל האתר). מטרת הפרויקט העיקרית הינה "לשדך" בין תלמיד למורה בצורה נוחה וקלה לשני הצדדים, תוך שמירה על פרטיות המשתמש (המשתמש רשאי לערוך את פרופיל הנתונים שלו בכל עת וכך בידו הבחירה אילו נתונים להציג ואילו לא).



חפש

עד מחיר

ממחיר

אזור

שם מורה

מקצוע

מיון לפי מחיר

מועדפים שלי