

ב. מבוא

הנושא של האתר: דינוזאורים.

הנושא המרכזי של כל האתרים הוא קיימות וסביבה. בתחילת השנה, בחרתי בנושא: עצים. אך עד מהרה, הבנתי שאני צריך לעשות פרויקט על נושא שמעניין אותי, ולכן בחרתי בדינוזאורים.

קהל היעד: אנשים שרוצים להרחיב את הידע שלהם לגבי דינוזאורים, יכולים להרחיב את הידע שלהם בעקבות הכניסה לאתר.

הטכנולוגיה בה השתמשתי:

DB , HTML ,Visual Studio, ASPX ,ASP.NET, Java Script, C-sharp, SQL

שם משתמש וסיסמא של מנהל באתר:

UserName : ANDREW

Password: AAAbb1

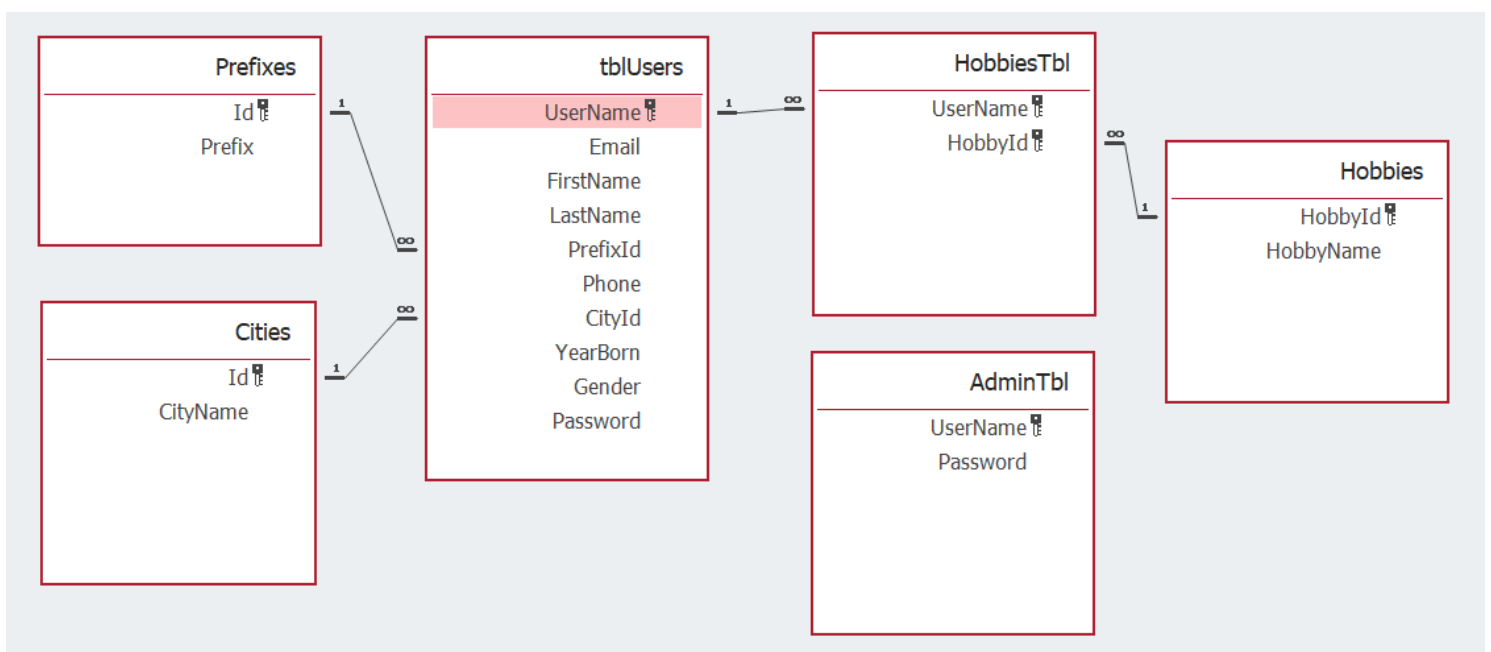
שם משתמש וסיסמא של משתמש באתר:

UserName : NormalUser

Password: AAAbb1

ג. טבלאות ונתונים באתר:

קשרי גומלין:

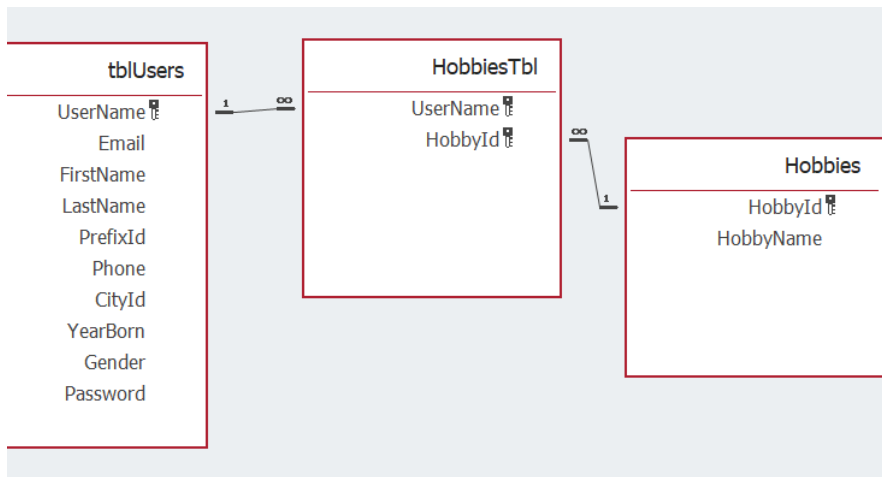


באתר, כחלק מהנתונים שעלי לשמור על המשתמש, בחרתי "לנרמל" את התחביבים של כל משתמש. טבלאות מנורמלות :

1. יותר נוחות לקריאה
2. קלות יותר לשדרוג ולהרחבה
3. מוגנות מפני טעויות בזמן הוספה/עדכון/מחיקה.

רציתי לקשר בין הטבלאות tblUsers ו-Hobbies. לשם זאת השתמשתי בטבלת HobbiesTbl.

בזכות הטבלה HobbiesTbl יש קשר של רבים לרבים בין tblUsers לבין Hobbies.



הסבר:

בטבלת HobbiesTbl:

1. לכל משתמש יכולים להיות כמה תחביבים.
2. למשתמשים שונים יכול להיות את אותו תחביב.

דוגמא מהעמוד SHOW

HOBBIES:

- (בכתום: דוגמא לסעיף 1.
בירוק: דוגמא לסעיף 2.)

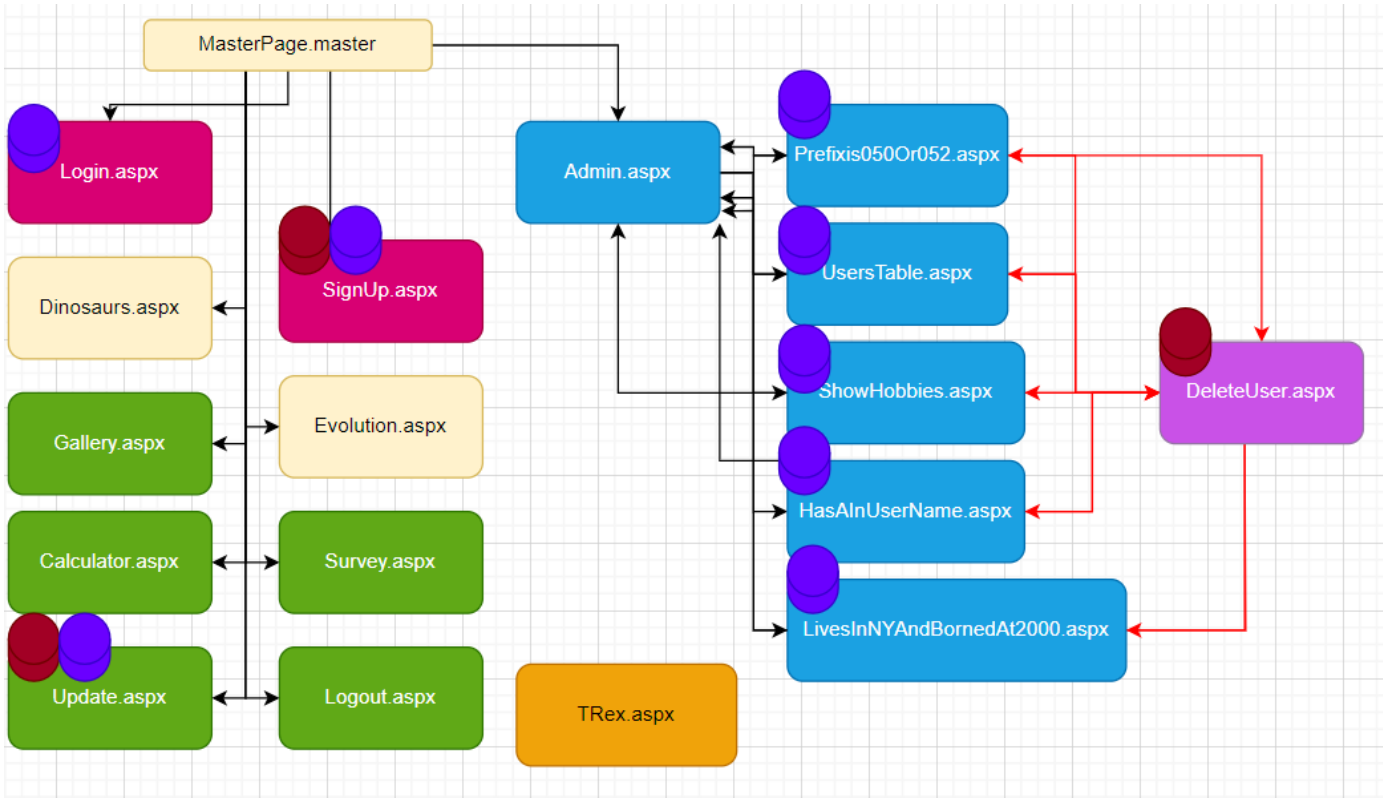
User Name	Hobby Name
ANDREW	Swimming
ANDREW	Singing
ANDREW	Tennis
ANDREW	Football
ANDREWW	Tennis
ANDREWW	Football
example	Singing

ד. פירוט טבלאות האתר: (מפתח ראשי, מפתח זר)

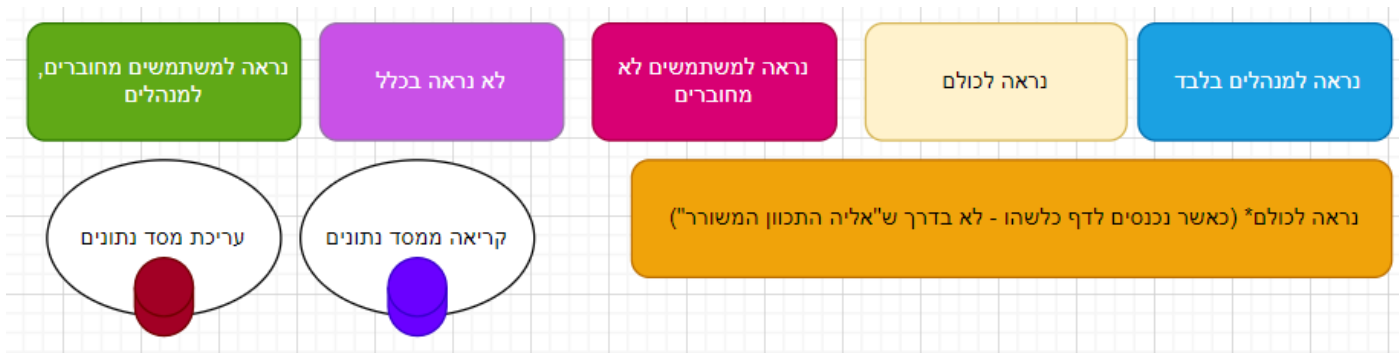
טבלה	שדות
Cities לשם שם העיר ומספר סידורי של העיר.	<ul style="list-style-type: none"> Id - מספר סידורי CityName - שם העיר
Hobbies טבלה לשם התחביב ומספר סידורי של התחביב.	<ul style="list-style-type: none"> HobbyId - מספר סידורי HobbyName - שם התחביב
HobbiesTbl טבלה לשם שמירת המשתמשים עם התחביבים	<ul style="list-style-type: none"> UserName - שם משתמש (גם מפתח זר) HobbyId - מספר סידורי של התחביב (גם מפתח זר)
Prefixes לשם מספר סידורי של הקידומת והקידומת	<ul style="list-style-type: none"> Id - מספר סידורי Prefix - קידומת
tblUsers טבלה שבה קיימים כל הפרטים על המשתמשים (חוץ מתחביבים)	<ul style="list-style-type: none"> UserName - שם משתמש Email - אימייל FirstName - שם פרטי LastName - שם משפחה PrefixId - מזהה של הקידומת Phone - טלפון CityId - מזהה של העיר YearBorn - שנת לידה Gender - מגדר Password - סיסמא Votes - כמות הצבעות לסקר באתר
AdminTbl טבלה שמכילה את הפרטים (שם משתמש וסיסמא) של המנהלים	<ul style="list-style-type: none"> UserName - שם משתמש (של מנהל) Password - סיסמא

ה. הקוד המטפל במסד הנתונים: קובץ ה-DBHELPER. אין צורך לצרף.

ו. מפת אתר:



מקרא:



קו שחור ← מעבר שנוצר על ידי המשתמש מדף לדף.

קו אדום ← מעבר אוטומטי שנוצר על ידי האתר ולא על ידי המשתמש.

*** (אין מעבר לעמוד TRex.aspx באופן ישיר, כיוון ששימוש נכון באתר לא יוביל אליו).**

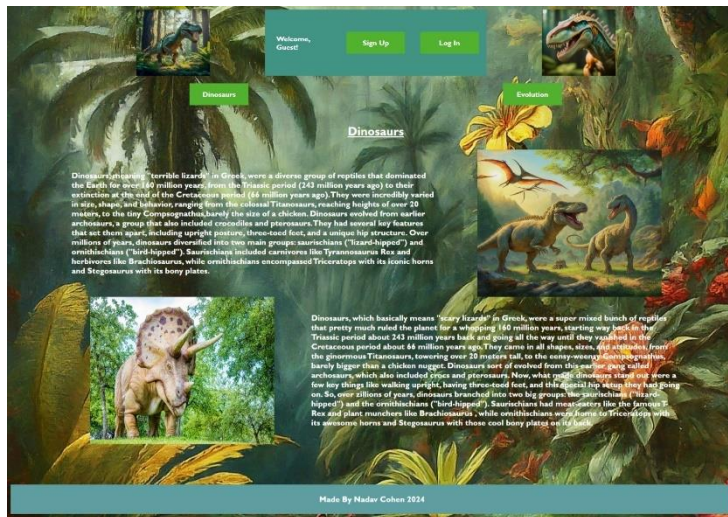
ז. הכלים התכנותיים בהם השתמשתי :

הכלים בהם השתמשתי הם המצגות ודפי העבודה שניתנו בשיעור. גם נעזרתי באתרים ובכלי בינה מלאכותית שמופרטים בהמשך

ח. באיזה אתרים הסתייעתי:

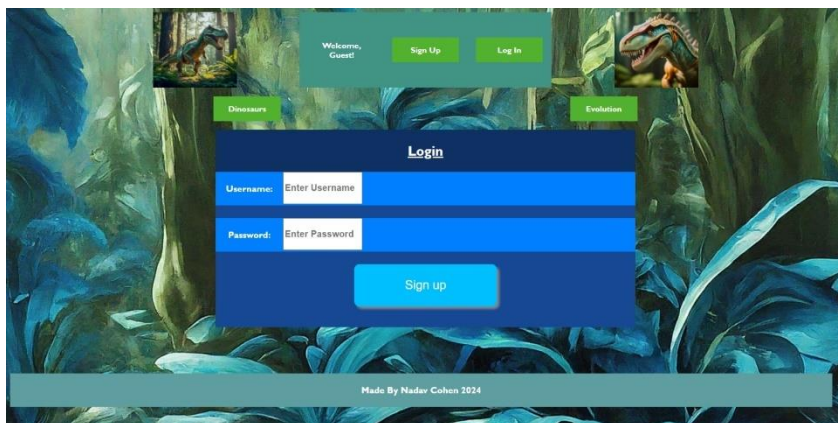
1. נעזרתי הרבה באתר w3 schools (לדוגמא: להבין בצורה טובה יותר JS בהכנת העמוד Calculator.aspx).
קישור לאתר: <https://www.w3schools.com>
2. נעזרתי גם באתר הזה, כדי ליצור את השעון האנאלוגי המוצג למשתמשים מחוברים ולמנהלים באתר.
קישור לאתר: <https://www.geeksforgeeks.org/how-to-create-analog-clock-using-html-css-and-javascript/>
3. נעזרתי באתר הבא, כדי ליצור את התמונה של השעון האנאלוגי ששמתי באתר.
קישור לאתר: <https://www.canva.com/>
4. נעזרתי ב Gemini | ChatGPT כלי בינה מלאכותית - בכתיבת קוד לאתר, כמו שאילתות/קוד C#. נעזרתי בכלים אלה גם כדי ליצור תמונות של דינוזאורים באתר.
קישורים למטה:
<https://openai.com/chatgpt>
<https://gemini.google.com>

ט. שלבים עיקריים של הפרויקט וכיצד להפעיל אותו

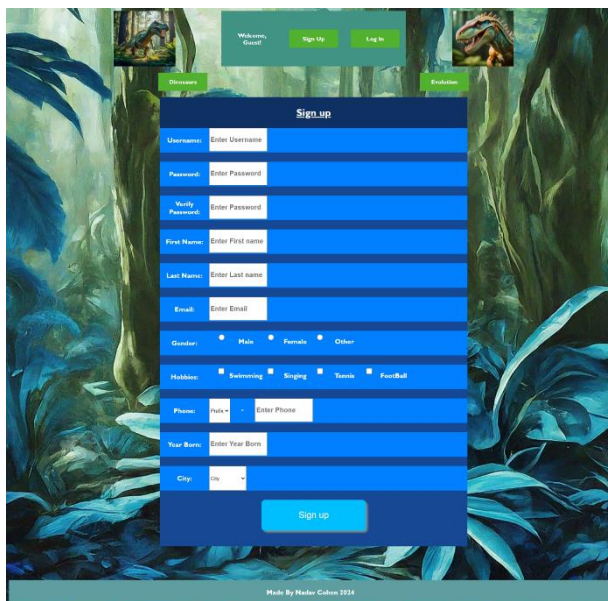


1. כשנכנסים לאתר, נכנסים לעמוד הראשי, Dinosaurs שמסביר על דינוזאורים

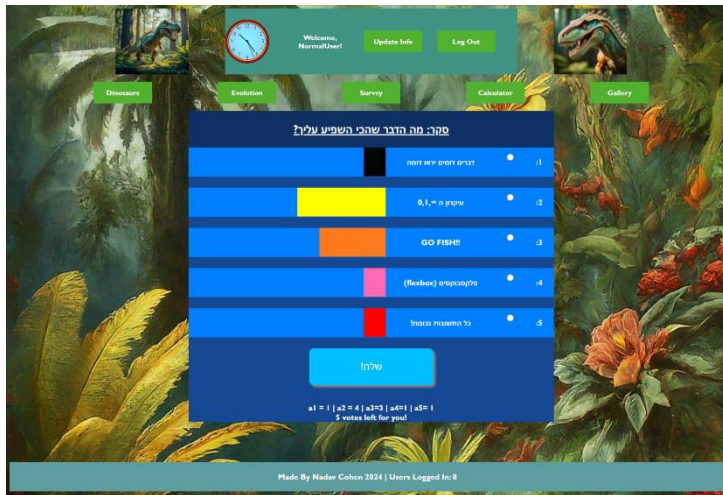
2. לפי עיקרון "דברים דומים יראו דומה" – כך גם העמוד "Evolution" מסביר על האבולוציה של הדינוזאורים.



3. כשנרצה להתחבר לאתר, נזין את פרטי המשתמש – שם משתמש וסיסמא בדף Login.

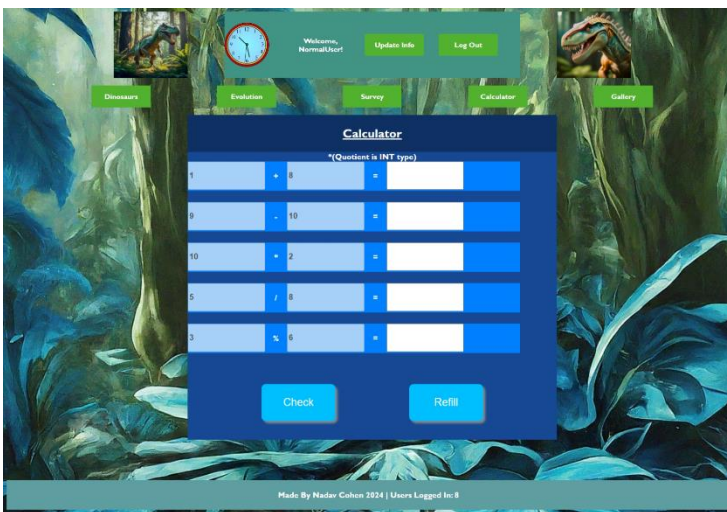


4. אם אין לנו משתמש, נרצה להירשם אל האתר. נעשה זאת בדף Sign Up. לאחר הנתונים עברו את הבדיקות הנדרשות – נתונים אלה יתווספו לטבלת המשתמשים, משמע המשתמש "ייצטרף" לאתר.

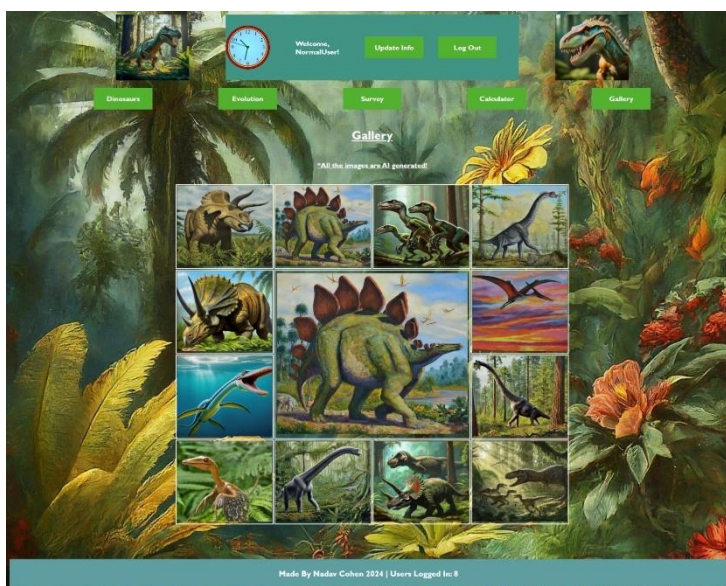


כששתמש רגיל ייתחבר לאתר, יופיעו לו מספר דפים נוספים. שמאפשרים למשתמש למקסם את החוויה שלו באתר. למשתמש מחובר מוצגת הודעת ברוך הבא בהתאם וגם שרון אנאלוגי. בנוסף משתמש מחובר:

5. יוכל להצביע (ולהשפיע!) בסקר, בעמוד Survey. הסקר בכוונה בעברית, כדי להיצמד למקור עד כמה שאפשר. בכל פעם משתמש מתחבר יש לו 5 הצבעות.



6. יוכל גם להתנסות במחשבון (בעמוד Calculator), שבו ניתנים למשתמש תרגילים פשוטים בחשבון, ועליו לפתור אותם.



7. למשתמש גם תפתח גלריה של תמונות של דינוזאורים. (דינוזאורים נכחדו הרבה לפני שהומצאו המצלמות, לכן כל התמונות נוצרו בידי טכנולוגיות בינה מלאכותית – AI.)

8. כשהמשתמש ירצה לצאת מהאתר הוא יילחץ על כפתור הLogout. (לאחר מכן המשתמש יוחזר לעמוד Dinosaurs)

9. כשהמשתמש יירצה לעדכן את פרטיו, הוא ייכנס לעמוד Update, ושם הוא יכול לשנות אותם. הפרטים בדף זה ממולאים אוטומטית מהנתונים שיש במסד הנתונים.

למנהל:

10. למנהל תהיה אפשרות להיכנס לעמוד המנהלים, שבו מוצגות טבלאות על פי שאילתות.

11. לפי עיקרון "דברים דומים ייראו דומה", התמונה הבאה מציגה טבלה של כל המשתמשים. בכל טבלה יש אפשרות מחיקה לשורה. לכן בשאילתות שמציגות משתמשים ניתן למחוק משתמשים, אך למשל בהצגת התחביבים, המחיקה היא של התחביב אצל המשתמש.

User Name	Hobby Name	Delete Hobby
ANDREW	Swimming	[DELETE]
ANDREW	Singing	[DELETE]
ANDREW	Tennis	[DELETE]
ANDREW	Football	[DELETE]
ANDREW	Tennis	[DELETE]
ANDREW	Football	[DELETE]
example	Singing	[DELETE]
newnew	Singing	[DELETE]
NormalUser	Swimming	[DELETE]
NormalUser	Singing	[DELETE]
NormalUser	Tennis	[DELETE]
NormalUser	Football	[DELETE]

The total of hobbies is 12

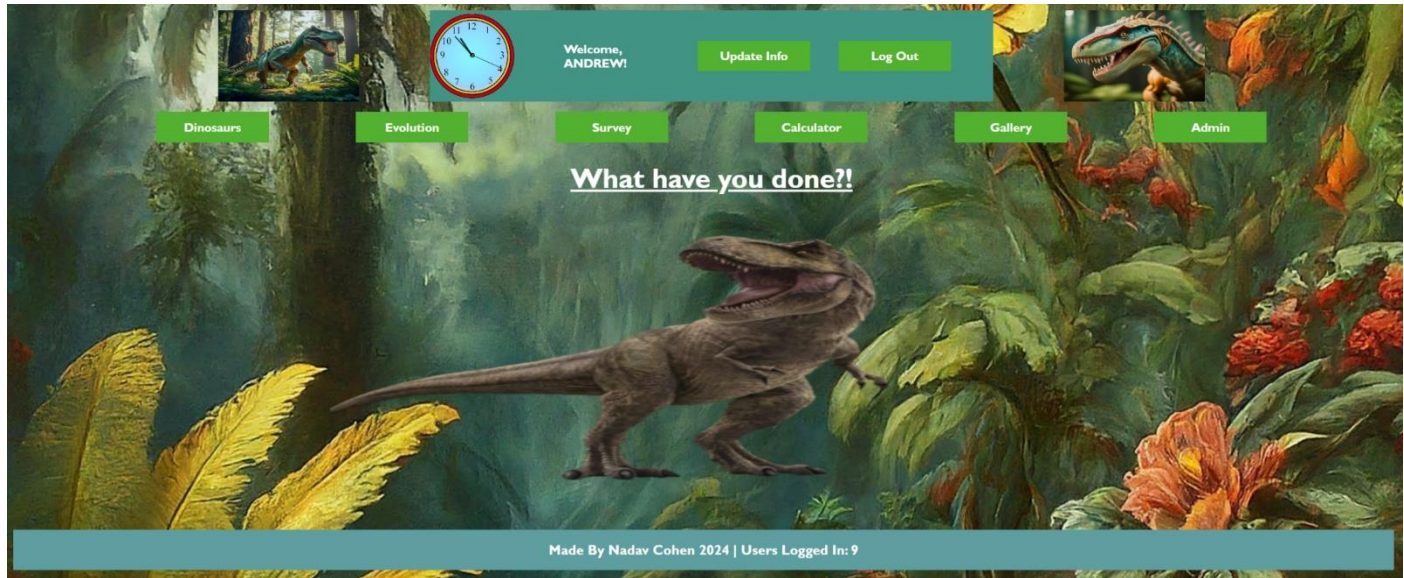
[Go to Admin Page](#)

User Name	First Name	Last Name	Phone	City	Year Born	Gender	Email	Password	Delete User
ANDREW	tugb	rtuwr	054 - 5503499	Petah-Tikva	2000	male	friu@vryullgfrag	AAAbb1	[DELETE]
ANDREW	nyjok	btui	054 - 1234589	New York	1984	other	btui@btui.dfs	AAAbb1	[DELETE]
example	ytyjklb	tvuktrv	059 - 1237846	Paris	1998	other	vtykl@vtylu.gtd	AAAbb1	[DELETE]
newnew	ngui	btgui	054 - 1237846	London	1998	female	btui@vtykl.gf	AAAbb1	[DELETE]
NormalUser	normal	user	052 - 1598432	New York	2000	other	norm@gmail.com	AAAbb1	[DELETE]

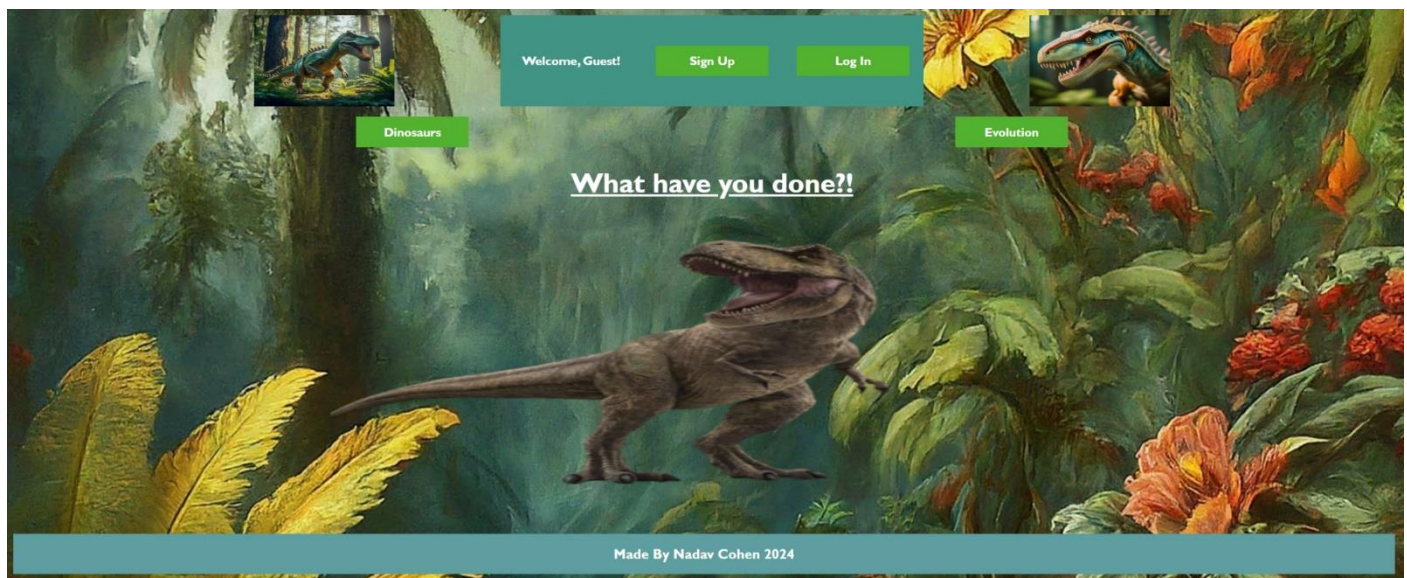
The amount of users is 5

[Go to Admin Page](#)

שימוש באתר לא נכון, כמו למשל לנסות להגיע לדף Admin כשלמשתמש אין הרשאה זו, או לנסות לעדכן פרטים (Update) או להתנתק (Logout) המשתמש יגיע לעמוד זה: TRex:



כשמשתמש לא מחובר ינסה להיכנס ללא הרשאה, יופיעו לו את האותם הדפים שהופיעו לו לפני כן.



י. קוד נוסף שלא נמצא במצגות:

בדף הglobal:

אם נתעלם מבדיקות התקינות של JS, משתמש יכול לקרוא לעצמו "Guest" בזכות הSession: "Is Connected" שכשמו כן הוא.

```
void Session_Start(object sender, EventArgs e)
{
    Session["UserName"] = "Guest";
    Session["Admin"] = "NO";
    Session["IsConnected"] = "NO";
    Session["Votes"] = 5;
}

void Session_End(object sender, EventArgs e)
{
    Session["UserName"] = "Guest";
    Session["IsConnected"] = "NO";
    Session["Admin"] = "NO";
    Session["Votes"] = 5;
}
```

דף MasterPage

עיצוב ה"מיני תפריט" באתר למעלה משתנה בהתאם לסטטוס המשתמש.
בצד הלקוח:

```
<div class="ItemInBannerContainer"></div>
<div class="<% =welcomeClass%>">

<div class="<% =clockClass %>">
<div id="clockContainer">
    <div id="hour"></div>
    <div id="minute"></div>
    <div id="second"></div>
</div>
</div>
```

וגם – יראה למשתמשים מחוברים את כמות המשתמשים שהתחברו לאתר:
(באותה דרך, כך מוצגים/לא מוצגים דפים ספציפים למשתמשים לפי הסטטוס שלהם מחוברים/מנהלים)

```
<%if (Session["IsConnected"].ToString() == "YES") {
    ...
    <div class="FooterContainer">
    <div class="Footer">Made By Nadav Cohen 2024 | Users Logged In: <%=Application["UsersLoggedIn"] %></div>
    </div>
    <%} %>
<%else {%>
    <div class="FooterContainer">
    <div class="Footer">Made By Nadav Cohen 2024</div>
    </div>
    <%} %>
```

בצד השרת:

```
public string welcomeClass="class"; //doesn't matter
public string clockClass="class"; //doesn't matter

0 references
protected void Page_Load(object sender, EventArgs e)
{
    if (Session["isConnected"].ToString() == "YES")
    {
        welcomeClass = "welcomeWhenUserIsConnected";
        clockClass = "";
    }
    else
    {
        welcomeClass = "welcome";
        clockClass = "Invisible";
    }
}
```

שעון אנאלוגי – שנמצא בקובץ clock.js וממוקם בmasterpage.

```
setInterval(() => {
    d = new Date(); //object of date()
    hr = d.getHours();
    min = d.getMinutes();
    sec = d.getSeconds();
    hr_rotation = 30 * hr + min / 2; //converting current time
    min_rotation = 6 * min;
    sec_rotation = 6 * sec;

    hour.style.transform = `rotate(${hr_rotation}deg)`;
    minute.style.transform = `rotate(${min_rotation}deg)`;
    second.style.transform = `rotate(${sec_rotation}deg)`;
}, 1000);
```

את השעון האנאלוגי יוכלו לראות רק המשתמשים המחוברים/מנהלים.

לכן בדף MasterPage

```
<div class="<% =clockClass %>">
  <div id="clockContainer">
    <div id="hour"></div>
    <div id="minute"></div>
    <div id="second"></div>
  </div>
</div>
```

הוספת favicon לאתר:

```
<head runat="server">
  <title>Dinosaurs</title>
  <link rel="icon" type="image/x-icon" href="/Images/Favicon.png"/>
```

דף DeleteUser בנוי אחרת, ולכן אציג את כל הקוד שבצד השרת:

הדף משמש למחיקת משתמש מכל הטבלאות באתר/ מחיקת תחביב של משתמש בטבלת התחביבים.

קביעת/קבלת משתנים – שם המשתמש, והדף ממנו נשלחה הבקשה למחיקה

```
protected void Page_Load(object sender, EventArgs e)
{
    string fileName = "Database.mdf";
    string tableName = "tblUsers";
    string hobbyTableName = "HobbiesTbl";

    string username = Request.QueryString["Username"];
    string page = Request.QueryString["page"];
```


אם המשתמש לא נכנס לדף בדרך שבה צריך (דרך הכפתור המיועד).....
אם הדפים הם הדפים שמציגים את המשתמשים, תמחק את המשתמש מכל
הטבלאות

```
if (username == null || page == null)
{
    Response.Redirect("TRex.aspx");
}

if (page == "UsersTable" || page == "HasAINUserName" ||
    page == "LivesInNYAndBornedAt2000" || page == "PrefixIs0500r052")
{

    string sqlDeleteUser = $"DELETE FROM {tableName} WHERE UserName = '{username}'";
    string sqlDeleteHobby = $"DELETE FROM {hobbyTableName} WHERE UserName = '{username}'";
    string sqlDeleteAdmin = $"DELETE FROM AdminTbl WHERE UserName = '{username}'";

    DBHelper.DoQuery(fileName, sqlDeleteHobby);
    DBHelper.DoQuery(fileName, sqlDeleteUser);
    DBHelper.DoQuery(fileName, sqlDeleteAdmin);
}
```

אם המשתמש מחק את עצמו, תנתק אותו מהאתר

```
//if the Admin deletes himself, than he should be disconnected
//the same code from the logout.cs
if (username == Session["UserName"].ToString())
{
    Response.Cache.SetExpires(DateTime.Now.AddMinutes(-1));
    Response.Cache.SetCacheability(HttpCacheability.NoCache);
    Response.Cache.SetNoStore();
    Response.Clear();

    Session.Clear();
    Session.RemoveAll();
    Session.Abandon();

    Response.Redirect("Dinosaurs.aspx");
}
```

אם הדף שנבחר הוא הצגת טבלת התחביבים, קלוט את מזהה התחביב, ותמחק אותו מהתחביבים של המשתמש

```
if (page == "ShowHobbies")
{
    int hobbyId = Convert.ToInt32(Request.QueryString["hobby"].ToString());
    string sqlDeleteHobby = $"DELETE FROM {hobbyTableName} WHERE UserName" +
        $" = '{username}' AND HobbyId = {hobbyId}";

    DBHelper.DoQuery(fileName, sqlDeleteHobby);
}
```

ולבסוף, תחזיר את המשתמש לדף ממנו בא:

```
string goToPage = page + ".aspx";
Response.Redirect(goToPage);
```

כך בדפים האחרים (בצד השרת שלהם) ניתנת הפנייה לדף זה עם המשתנים המתאימים, למשל בדף PrefixIs0500r052 (שחוץ מהשם של הדף, שווה ערך לשאר הקודים בדפים שלא מציגים תחביבים):

```
string username = table.Rows[i]["UserName"].ToString();
string page = "PrefixIs0500r052";

st += $"<td><a href = 'DeleteUser.aspx?Username={username}&page={page}' class='deleteUser'>[DELETE]</a></td>";
```

ובדף הצגת התחביבים:

```
string username = table.Rows[i]["UserName"].ToString();
string page = "ShowHobbies";
int hobby = Convert.ToInt32(table.Rows[i]["HobbyId"].ToString());

st += $"<td><a href = 'DeleteUser.aspx?Username={username}&page={page}&hobby={hobby}' class='deleteUser'>[DELETE]</a></td>";
```

דף Updaten

מכיוון שאצלי התחביבים מנורמלים, כשהמשתמש מעדכן את התחביבים שלו, אני מוחק את התחביבים שכבר יש לו – ומוסיף את אלה שבחר בטופס.

*להתעלם מsqlMsg

```
string sqlDeleteHobbies = $"DELETE FROM {hobbyTableName} WHERE UserName = '{Session["UserName"]?.ToString()}';

sqlMsg += sqlDeleteHobbies + '\n';

DBHelper.DoQuery(fileName, sqlDeleteHobbies);

string[] hobbyIds = hobbies.Split(',');

foreach (string hobbyId in hobbyIds)
{
    string sqlInsertHobby = $"INSERT INTO HobbiesTbl (UserName, HobbyId) " +
        $"VALUES ('{userName}', {hobbyId.Trim()})";

    DBHelper.DoQuery(fileName, sqlInsertHobby);

    sqlMsg += sqlInsertHobby + '\n';
}
```

בצד ה ASPX של כל הטפסים (Update, SignUp, Login) יש בדיקות JS, שמופעלות גם בהקלדת תו של המשתמש (onkeyup). למשל בדף הLogin:

```
<label for="userName"><b>Username:</b></label>
</div>
<input type="text" placeholder="Enter Username" id="userName" name="userName" class="input" onkeyup="userNameCheck();" >

<input type="text" id="userNameMsg" class="msg" style="display: none;" disabled="disabled" />
</div>
```

דף ה-Survey

החלטתי שבכל פעם שמשתמש מתחבר לאתר, יש לו 5 הצבעות זמינות בסקר. את ההצבעות אני מאתחל בדף global.aspx.cs

```
0 references
void Session_Start(object sender, EventArgs e)
{
    Session["UserName"] = "Guest";
    Session["Admin"] = "NO";
    Session["IsConnected"] = "NO";
    Session["Votes"] = 5;
}

void Session_End(object sender, EventArgs e)
{
    Session["UserName"] = "Guest";
    Session["IsConnected"] = "NO";
    Session["Admin"] = "NO";
    Session["Votes"] = 5;
}
```

בצד השרת של הסקר: גם פה, מפרט את כל צד השרת.

תאתחל את כמות ההצבעות לפי הסשן

```
public string msg = "";
public string usersLeft;
public double a1;
public double a2;
public double a3;
public double a4;
public double a5;
0 references
protected void Page_Load(object sender, EventArgs e)
{
    if (Session["IsConnected"].ToString() == "NO")
    {
        Response.Redirect("TRex.aspx");
    }

    int votes = (int)(Session["Votes"]);
```


אם הטופס נשלח עם תשובה, ולמשתמש יש הצבעות, תוריד הצבעה.

```
if (Request.Form["submit"] != null && Request.Form["answer"] != null && votes > 0)
{
    int ans = int.Parse(Request.Form["answer"].ToString());
    votes = votes - 1;

    Session["Votes"] = (int)(Session["votes"]) - 1;
}
```

בחרתי לעשות את התוצאות כתוצאה של פילוג (מסכום כל ההצבעות).
לכן אני סוכם את כל ההצבעות, ובהתאם קובע את $w[i]$ (בודק גם חלוקה ב 0)

```
int sum = s[0] + s[1] + s[2] + s[3] + s[4];
double[] w = new double[5];
for (int i = 0; i < w.Length; i++)
{
    if (sum == 0)
    {
        w[i] = 0;
    }
    else
    {
        w[i] = (s[i] * 10 / (double)(sum)) * 50;
    }
}
```

והודעה בסוף למשתמש לגבי כמות ההצבעות שנותרו:

```
if (votes > 0)
{
    usersLeft = $"{votes} votes left for you!";
}
else
{
    usersLeft = "NO MORE VOTES LEFT";
}
```

דף ההLogin

וויתרתי על דף ההLogin ובמקומו שמתי את השורות האלה בצד השרת של דף ההתחברות:

תבחר את כל המנהלים המטבלה, אם נמצא מנהל עם שם משתמש וסיסמא זהה – המשתמש ייחשב כמנהל באמצעות Session["Admin"]. שאילתת הבחירה היא שאילתה בטוחה עם פרמטרים.

```
Session["UserName"] = table.Rows[0]["UserName"];
Session["IsConnected"] = "YES";

string safeAdminSQL = $"SELECT * FROM AdminTbl WHERE UserName = @UserName AND Password = @Password";
DataTable tableForAdmin = DBHelper.Login(fileName, safeAdminSQL, parameters);
int lengthAdmin = tableForAdmin.Rows.Count;

if (lengthAdmin == 1)
{
    Session["Admin"] = "YES";
}
else
{
    Session["Admin"] = "NO";
}
```

בדף Calculator

הודעה שמראה את כמות התשובות הנכונות שהמשתמש ענה. בצד הלקוח:

```
<h2 id="CountAll"></h2>
```

JS - calculator.js – קביעת משתנה count לפי כמות התשובות הנכונות.

```
function CheckSum() {
    var count = 0;

    if ((num1Sum + num2Sum) === answerSum) {
        document.getElementById("msgSum").innerHTML = "Yes!";
        count++;
    }

    if (num1Sub - num2Sub === answerSub) {
        document.getElementById("msgSub").innerHTML = "Yes!";
        count++;
    }
}
```

וכו'..

לבסוף, הצגת ההודעה בדף:

```
document.getElementById("CountAll").innerHTML = "You answered " + count + " Correct Answer(s).";
```

בדף ShowHobbies

קוד ה-ASPX זהה בכל אחד מדפי ההצגת הנתונים, השונה הוא צד השרת במקרה זה.

אם משתמש לא רשאי להיכנס לדף.....

```
if (Session["Admin"].ToString() != "YES")
{
    Response.Redirect("TRex.aspx");
}
string fileName = "Database.mdf";
```

השאלת SELECT ו"ציקת" טבלה:

```
sqlSelect = $"SELECT HobbiesTbl.UserName, HobbiesTbl.HobbyId, Hobbies.HobbyName" +
    $" FROM HobbiesTbl INNER JOIN Hobbies ON Hobbies.HobbyId = HobbiesTbl.HobbyId;";
DataTable table = DBHelper.ExecuteDataTable(fileName, sqlSelect);

string sqlSelectHobby = $"SELECT * FROM HobbiesTbl;";
DataTable tableHobby = DBHelper.ExecuteDataTable(fileName, sqlSelectHobby);
```

הצגת הטבלה: הודעה בהתאם / עמודות

```
int length = table.Rows.Count;
if (length == 0)
{
    msg = "NO USERS";
}
else
{
    st += "<tr>";
    st += "<th>User Name</th>";
    st += "<th>Hobby Name</th>";
    st += "<th>Delete User</th>";

    st += "</tr>";
}
```

הצגת הטבלה: שורות עם מידע:

```

for (int i = 0; i < length; i++)
{
    st += "<tr>";

    st += $"<td>{table.Rows[i]["UserName"]}</td>";
    st += $"<td>{table.Rows[i]["HobbyName"]}</td>";
    string username = table.Rows[i]["UserName"].ToString();
    string page = "ShowHobbies";
    int hobby = Convert.ToInt32(tableHobby.Rows[i]["HobbyId"].ToString());

    st += $"<td><a href = 'DeleteUser.aspx?Username={username}&page={page}&hobby={hobby}' " +
        $" class='deleteUser'>[DELETE]</a></td>";

    st += "</tr>";
}

```

הודעה על כמות התחביבים באתר:

```
msg = $"The total of hobbies is {length}";
```