פרויקט תכנות בסביבת אינטרנט

אנדריי לשצ'נקו

תוכן עניינים

[מבוא 3](#_Toc388798796)

[מטרות האתר 3](#_Toc388798797)

[קהל היעד 3](#_Toc388798798)

[תאור האתר 3](#_Toc388798799)

[האתר ואני 4](#_Toc388798800)

[כלי פיתוח 4](#_Toc388798801)

[הפעלת האתר 4](#_Toc388798802)

[משתמש רגיל 4](#_Toc388798803)

[מנהל 4](#_Toc388798804)

[מפת האתר 5](#_Toc388798805)

[מסד הנתונים 6](#_Toc388798806)

[קוד האתר 7](#_Toc388798807)

[דפי html 7](#_Toc388798808)

[Master Page 8](#_Toc388798809)

[Homepage.aspx 10](#_Toc388798810)

[טופס רישום 12](#_Toc388798811)

[Registration.aspx 13](#_Toc388798812)

[FormValidation.js 14](#_Toc388798813)

[דפי ASP.NET 16](#_Toc388798814)

[Datalink.cs 16](#_Toc388798815)

[Login.aspx.cs 19](#_Toc388798816)

[Registration.aspx.cs 19](#_Toc388798817)

[UpdateDetails.aspx.cs 20](#_Toc388798818)

[ניהול האתר 23](#_Toc388798819)

[Admin.aspx 24](#_Toc388798820)

[Admin.aspx.cs 25](#_Toc388798821)

# מבוא

## מטרות האתר

האתר נבנה כפרויקט לתרגול החומר הנלמד במסגרת לימודי תכנות בסביבת אינטרנט בבית הספר.

המטרה העיקרית היתה לבנות אתר פועל, שיכלול רכיבים שונים כמו דפי מידע, גלריה, רישום משתמשים ודף ניהול בו מנהלים יוכלו למצוא מידע על משתמשים ולמחוק אותם.האתר נבנה בדמות אתר שמספק מידע על המשמר האזרחי, על ההיסטוריה שלו, על התנדבות במשמר האזרחי, ואף כולל גלרית תמונות הקשורות בנושא.

בחרתי בנושא זה מכיוון שאני מתנדב במשמר האזרחי במסרגת המחויבות האישית.

חשוב לציין כי העיקר באתר זה הוא הטכנולוגיה, ולא התוכן. האתר נבנה כפרויקט טכנולוגי, ולכן התוכן הוא כדי למלא את דפי האתר, ולא כדי לספק מידע אמיתי.

## קהל היעד

האתר לא נבנה כאתר עם קהל יעד מוגדר.

נושא האתר מתאים לאנשים שמחפשים מידע על המשמר האזרחי, אך במציאות קהל היעד הוא לרוב מורים וחברים להם הראיתי את האתר.

## תאור האתר

לא כל התמשתמשים יכולים לראות את כל הדפים.

משתמש רגיל יכול לראות את הדפים הבאים:

* דף בית המכיל מידע כללי על המשמר האזרחי
* דף המתאר את מטרות המשמר האזרחי
* דף המוסר מידע על התנדבות במשמר האזרחי
* דף על ההיסטוריה של המשמר האזרחי

משתמש רשום יכול בנוסף לגשת לדפים הבאים

* גלרית תמונות
* דף עדכון פרטים

מנהל באתר יכול לגשת לדף ניהול שם הוא יכול לראות מידע על המשתמשים (אך לא את סיסמאתם),

להפוך משתמשים אחרים למנהלים, ולמחוק משתמשים.

הרישום מתבצע באמצעות טופס רישום המחייב מענה על השדות הבאים:

* שם פרטי
* שם משפחה
* סיסמא
* מספר תעודת זהות
* דואר אלקטרוני

## האתר ואני

במהלך בניית האתר למדתי דברים רבים שלא יצא לי להתעסק איתם לפני כמו ASP.NET ו- SQL.

פעמים רבות למדים דברים חדשים כאשר חיפשתי פתרונות לבעיות בהן נתקלתי, ואז מצאתי דרכים חדשות לפתור בעיות שכבר פתרתי בדרכים פחות יפות.

## כלי פיתוח

האתר משתמש בטכנולוגיות הבאות:

* HTML
* JavaScript
* ASP.NET
* CSS
* Microsoft SQL Server

בהתחלה הקוד נכתב ב- Visual Studio Professional 2013, אך בגלל שהאתר צריך לעבוד על משחבי בית ספר עם גרסאות יותר ישנות של SQL Server הייתי צריך לעבור בצער רב ל- Visual Web Developer 2008.

האתר לא כולל תמונות בשביל העיצוב (פרט לגלריה ולתמונת הרקע). לכן לא היה כמעט צורך בתוכנות כמו Adobe Photoshop

השתמשתי בתוכנת ניהול הגרסאות GIT. קוד האתר, וכל ההיסטוריה שלו נשמרים באתר : https://github.com/andrey-leshenko/school-website

קוד האתר פתוח וניתן להורדה מהאתר.

## הפעלת האתר

### משתמש רגיל

שם משתמש: [regular@guy.com](mailto:regular@guy.com) . סיסמא: ilovecats

### מנהל

שם משתמש: [admin@adminland.com](mailto:admin@adminland.com) סיסמא: ADMIN (יש חשיבות לאותיות גדולות)

כמובן שניתן לשנות את כל הפרטים.

# מפת האתר

Admin.aspx

Admin.aspx.cs

Gallery.aspx

משתמש רשום

אורח

מנהל

UpdateDetails.aspx.cs

UpdateDetails.aspx

Registration.aspx.cs

Login.aspx

Login.aspx.cs

Registration.aspx

Volunteering.aspx

Logout.aspx

About.aspx

Homepage.aspx

# מסד הנתונים

מסד הנתונים מאורגן באופן הבא:

|  |  |
| --- | --- |
| Data Type | Column Name |
| Nchar(50) | Email |
| Nchar(9) | ID |
| Nchar(30) | FirstName |
| Nchar(30) | LastName |
| Nchar(40) | PassHash |
| bit | Admin |

תפקיד רוב השדות ברור מאד, חוץ משדה PassHash.

בשדה זה לא שמורה הסיסמא עצמה של המשתמש, אלה הפלט של פונקצית גיבוב (HASH) שפעלה על הסיסמא.פונקציית גיבוב(או פונקציית ערבול) היא פונקציה שממירה קלט חופשי באורך משתנה לפלט מעורבל באורך קבוע.

לפי הפלט לא ניתן לדעת מה היה הקלט, אבל אם ניקח קלט אחר, נפעיל עליו את הפונקציה, ואז נשווה את התוצאות, נוכל לדעת האם הקלט החדש שווה לקלט הישן.במקרה שלנו, פונקציית הגיבוב היא SHA-1, פונקציה פופולרית למטרה זו.

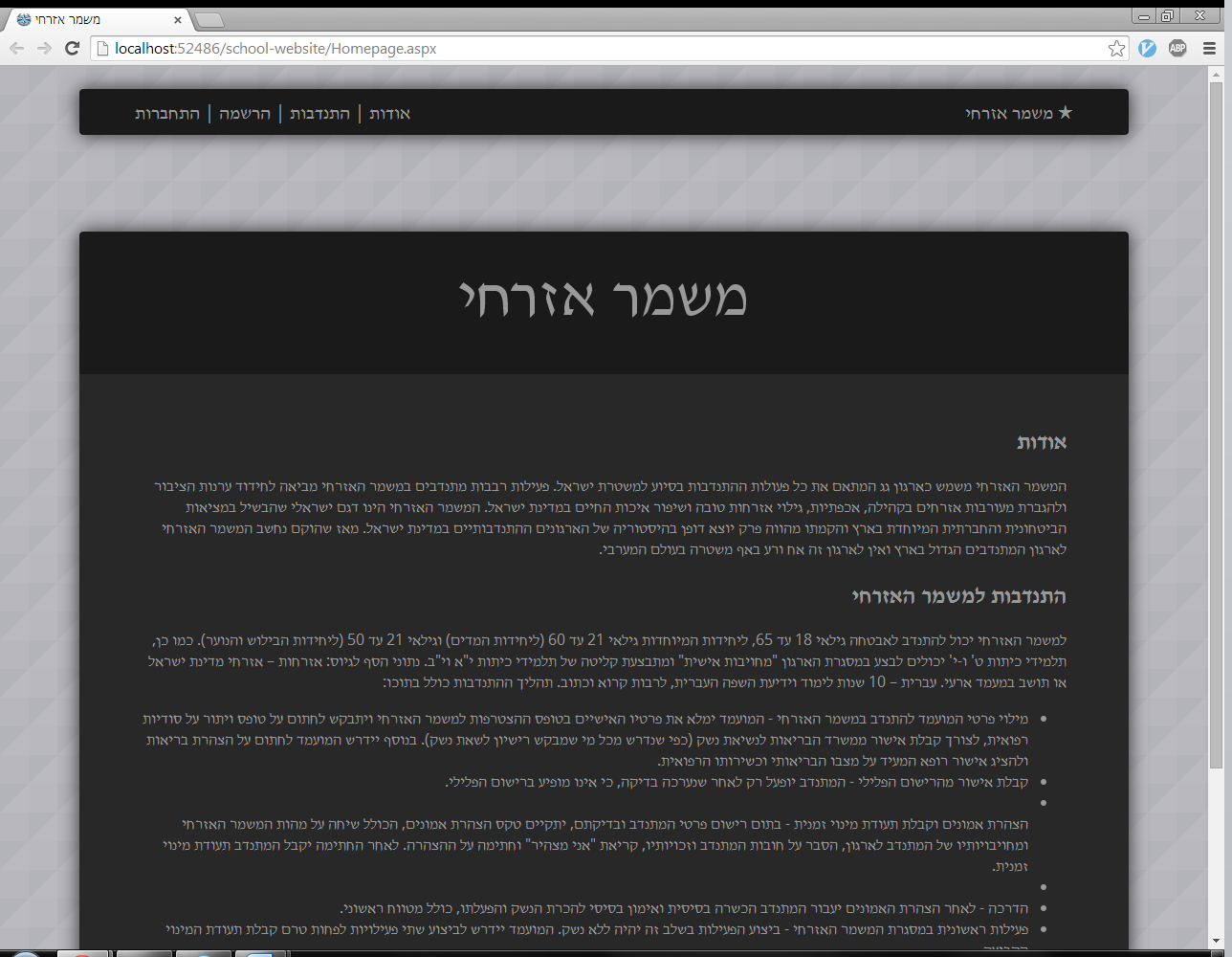
הפלט שלה הוא באורך של 160 ביטים, והוא נשמר כמחרוזת הקסדצימלית.בגלל שהסיסמא עצמה לא שמורה, גם מנהלי האתר לא יודעים מה הסיסמא של המשתמשים.

הגיבוב יקשה מאד על מציאת הסיסמא לפי מה ששמור במסד.



# קוד האתר

## דפי html



### Master Page

<%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head id="Head1" runat="server">

<link rel="shortcut icon" href="Images/icon.ico" />

<meta name="author" content="Andrey Leshenko" />

<title>

<asp:ContentPlaceHolder ID="PageTitle" runat="server">

</asp:ContentPlaceHolder>

</title>

<link rel="Stylesheet" href="Styles/Main.css" />

<asp:ContentPlaceHolder ID="Head" runat="server">

</asp:ContentPlaceHolder>

</head>

<body dir="rtl">

<div id="page-container">

<div id="toolbar">

<nav>

<div id="toolbar-logo">

<a href="Homepage.aspx">

★ משמר אזרחי

</a>

</div>

<div id="toolbar-buttons">

<a href="About.aspx">אודות</a> |

<a href="Volunteering.aspx">התנדבות</a> |

<span runat="server" id="galleryButton" visible="false">

<a href="Gallery.aspx">תמונות</a> |

</span>

<span runat="server" id="registrationButton" visible="false">

<a href="Registration.aspx">הרשמה</a> |

</span>

<span runat="server" id="updateButton" visible="false">

<a href="UpdateDetails.aspx">שינוי פרטים</a> |

</span>

<span runat="server" id="adminButton" visible="false">

<a href="Admin.aspx">ניהול</a> |

</span>

<span runat="server" id="loginButton" visible="false">

<a href="Login.aspx">התחברות</a>

</span>

<span runat="server" id="logoutButton" visible="false">

<a href="Logout.aspx">התנתקות</a>

</span>

</div>

</nav>

</div>

<span class="logged-in-as">

<%=user%>

</span>

<div id="page-body">

<h1>

<asp:ContentPlaceHolder ID="MainTitleText" runat="server">

</asp:ContentPlaceHolder>

</h1>

<div class="text-main-container">

<asp:ContentPlaceHolder ID="PageBodyText" runat="server">

</asp:ContentPlaceHolder>

</div>

</div>

</div>

<div class="page-footer" dir="ltr">

&copy Andrey Leshenko 2014

</div>

</body>

</html>

### Homepage.aspx

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true" CodeFile="Homepage.aspx.cs" Inherits="\_Default" %>

<asp:Content ID="Content1" ContentPlaceHolderID="Head" Runat="Server">

</asp:Content>

<asp:Content ID="Title" ContentPlaceHolderID="PageTitle" Runat="server">

משמר אזרחי

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainTitleText" Runat="Server">

משמר אזרחי

</asp:Content>

<asp:Content ID="Content3" ContentPlaceHolderID="PageBodyText" Runat="server">

<section>

<h2>

אודות

</h2>

<p>

</p>

<p>

המשמר האזרחי משמש כארגון גג המתאם את כל פעולות ההתנדבות בסיוע למשטרת ישראל. פעילות רבבות מתנדבים במשמר האזרחי מביאה לחידוד ערנות הציבור ולהגברת מעורבות אזרחים בקהילה, אכפתיות, גילוי אזרחות טובה ושיפור איכות החיים במדינת ישראל.‏‏ המשמר האזרחי הינו דגם ישראלי שהבשיל במציאות הביטחונית והחברתית המיוחדת בארץ והקמתו מהווה פרק יוצא דופן בהיסטוריה של הארגונים ההתנדבותיים במדינת ישראל. מאז שהוקם נחשב המשמר האזרחי לארגון המתנדבים הגדול בארץ ואין לארגון זה אח ורע באף משטרה בעולם המערבי.

</p>

</section>

<section>

<h2>

התנדבות למשמר האזרחי

</h2>

<p>

למשמר האזרחי יכול להתנדב לאבטחה גילאי 18 עד 65, ליחידות המיוחדות גילאי 21 עד 60 (ליחידות המדים) וגילאי 21 עד 50 (ליחידות הבילוש והנוער). כמו כן, תלמידי כיתות ט' ו-י' יכולים לבצע במסגרת הארגון "מחויבות אישית" ומתבצעת קליטה של תלמידי כיתות י"א וי"ב. נתוני הסף לגיוס: אזרחות – אזרחי מדינת ישראל או תושב במעמד ארעי. עברית – 10 שנות לימוד וידיעת השפה העברית, לרבות קרוא וכתוב. תהליך ההתנדבות כולל בתוכו:

</p>

<ul>

<li>

מילוי פרטי המועמד להתנדב במשמר האזרחי - המועמד ימלא את פרטיו האישיים בטופס ההצטרפות למשמר האזרחי ויתבקש לחתום על טופס ויתור על סודיות רפואית, לצורך קבלת אישור ממשרד הבריאות לנשיאת נשק (כפי שנדרש מכל מי שמבקש רישיון לשאת נשק). בנוסף יידרש המועמד לחתום על הצהרת בריאות ולהציג אישור רופא המעיד על מצבו הבריאותי וכשירותו הרפואית.

</li>

<li>

קבלת אישור מהרישום הפלילי - המתנדב יופעל רק לאחר שנערכה בדיקה, כי אינו מופיע ברישום הפלילי.

</li>

<li>

</li>

הצהרת אמונים וקבלת תעודת מינוי זמנית - בתום רישום פרטי המתנדב ובדיקתם, יתקיים טקס הצהרת אמונים, הכולל שיחה על מהות המשמר האזרחי ומחויבויותיו של המתנדב לארגון, הסבר על חובות המתנדב וזכויותיו, קריאת "אני מצהיר" וחתימה על ההצהרה. לאחר החתימה יקבל המתנדב תעודת מינוי זמנית.

<li>

</li>

<li>

הדרכה - לאחר הצהרת האמונים יעבור המתנדב הכשרה בסיסית ואימון בסיסי להכרת הנשק והפעלתו, כולל מטווח ראשוני.

</li>

<li>

פעילות ראשונית במסגרת המשמר האזרחי - ביצוע הפעילות בשלב זה יהיה ללא נשק. המועמד יידרש לביצוע שתי פעילויות לפחות טרם קבלת תעודת המינוי הקבועה.

</li>

<li>

קבלת תעודת מינוי קבועה ומעמד של חבר מן המנין במשמר האזרחי.

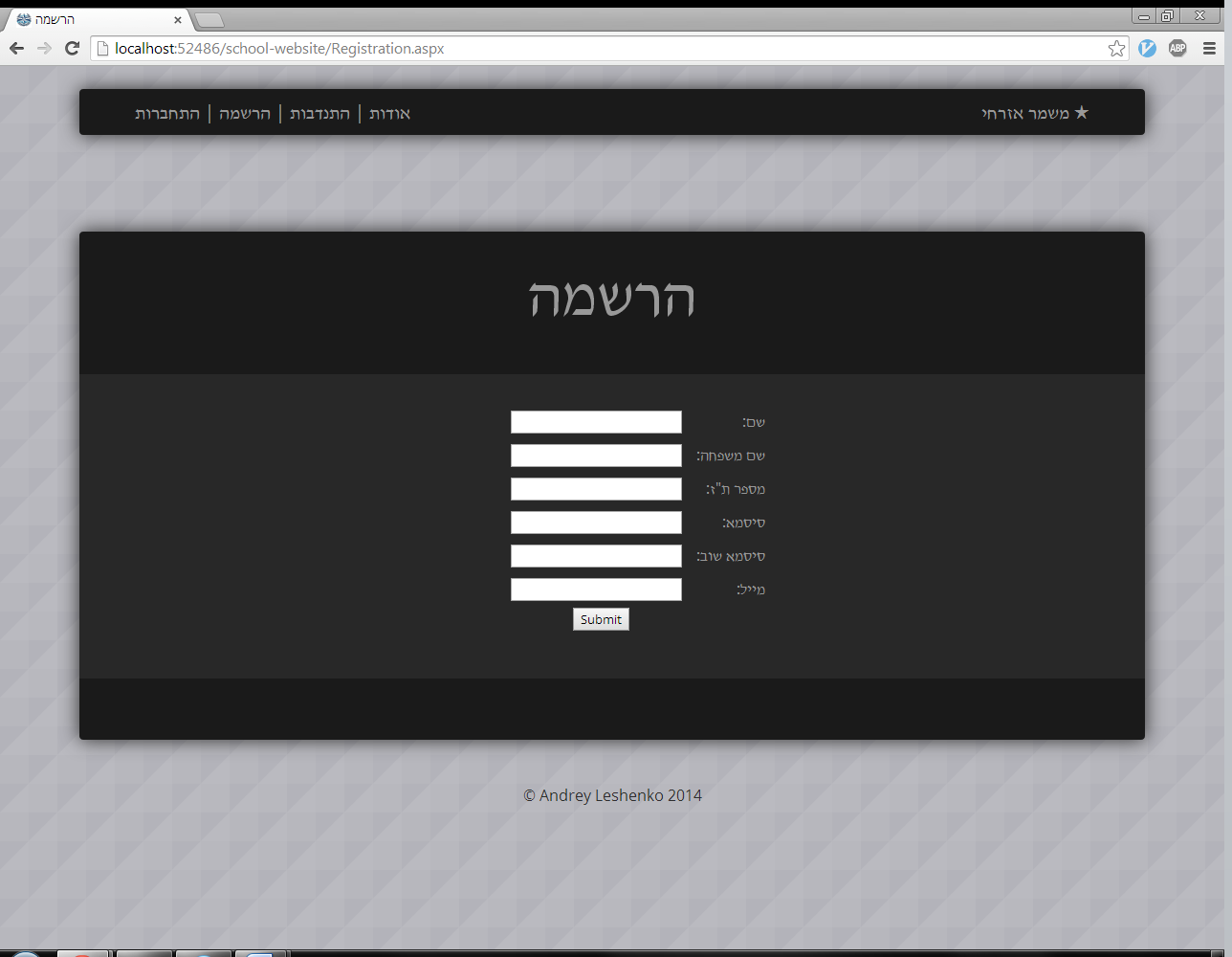
</li>

</ul>

</section>

</asp:Content>

## טופס רישום



### Registration.aspx

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true" CodeFile="Registration.aspx.cs" Inherits="\_Default" %>

<asp:Content ID="Content1" ContentPlaceHolderID="PageTitle" runat="Server">

הרשמה

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="Head" runat="Server">

<script type="text/javascript" src="Scripts/FormValidation.js">

// Exports: checkDetailsForm(form)

</script>

</asp:Content>

<asp:Content ID="Content3" ContentPlaceHolderID="MainTitleText" runat="Server">

הרשמה

</asp:Content>

<asp:Content ID="Content4" ContentPlaceHolderID="PageBodyText" runat="Server">

<div class="form-container">

<form runat="server" onsubmit="return checkDetailsForm(document.forms[0])">

<div class="form-descriptions">

שם: <br />

שם משפחה: <br />

מספר ת"ז: <br />

סיסמא: <br />

סיסמא שוב: <br />

מייל:

</div>

<div class="form-fileds">

<input type="text" name="firstName" /> <br />

<input type="text" name="lastName" /> <br />

<input type="text" name="id" /> <br />

<input type="password" name="userPassword" /> <br />

<input type="password" name="repeatPassword" /> <br />

<input type="text" name="email" />

</div>

<div class="submit-container">

<input type="submit" name="submit" />

</div>

</form>

<p>

<%=serverResponse%>

</p>

</div>

</asp:Content>

### FormValidation.js

function checkDetailsForm(form) {

var firstName = form.firstName.value;

var lastName = form.lastName.value;

var id = form.id.value;

var firstPassword = form.userPassword.value;

var secondPassword = form.repeatPassword.value;

var email = form.email.value;

if (firstName == "" || lastName == "" || firstPassword == "" || secondPassword == "" || email == "" || id == "")

return failWithMessage("The entire form must be filled");

if (firstName.length > 30)

return failWithMessage("The first name can't be longer than 30 characters");

if (lastName.length > 30)

return failWithMessage("The last name can't be longer than 30 characters");

if (email.length > 50)

return failWithMessage("The email can't be longer than 50 characters");

if (!isValidName(firstName))

return failWithMessage("Invalid first name.\nMust contain only hebrew letters, spaces, and hypens");

if (!isValidName(lastName))

return failWithMessage("Invalid last name.\nMust contain only hebrew letters, spaces, and hypens");

if (!isValidId(id))

return failWithMessage("Invalid ID number");

if (firstPassword != secondPassword)

return failWithMessage("Passwords don't match");

if (!isValidEmail(email))

return failWithMessage("Invalid email");

return true;

}

function isValidName(name) {

name = name.toLowerCase();

for (var i = 0; i < name.length; i++) {

var c = name.charAt(i);

if ((c < 'א' || c > 'ת') && c != ' ' && c != '-')

return false;

}

return true;

}

function isValidEmail(email) {

name = name.toLowerCase();

var atIndex = email.indexOf('@');

if (email.indexOf('.') == -1 || email.indexOf('.') == 0 || email.lastIndexOf('.') < atIndex)

return false;

if (atIndex == -1 || atIndex == 0 || atIndex == email.length - 1 || atIndex != email.lastIndexOf('@'))

return false;

for (var i = 0; i < email.length; i++) {

var c = email.charAt(i);

if ((c < 'a' || c > 'z') && (c < '0' || c > '9') && c != '-' && c != "\_" && c != '.' && c != '@')

return false;

}

return true;

}

function isValidId(id) {

if (id.trim().length != 9 || isNaN(id))

return false;

else

return true;

}

function failWithMessage(message) {

alert(message);

return false;

}

## דפי ASP.NET

בשביל להבין את מבנה דפי השרת, צריך לראות את Datalink.cs.

### Datalink.cs

כל דף שרוצה לעשות משהו עם מסד הנתונים, קורא לפונקציה מתאימה. הפונקציה ב-DataLink מרכיבה את ה-SQL המתאים, ומבצעת אותו על המסד באמצעות MyAdoHelper.cs.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Data;

public static class DataLink

{

public static bool Exists(string email, string password)

{

string sql = string.Format("SELECT \* FROM Users WHERE Email='{0}' AND PassHash='{1}'", email, GetHashedPassword(password));

return MyAdoHelper.IsExist(database, sql);

}

public static void AddUser(string email, string fname, string lname, string password, string id)

{

string sql = string.Format("INSERT INTO Users (Email, ID, FirstName, LastName, PassHash, Admin) VALUES ('{0}','{1}','{2}','{3}', '{4}', 'False')",

email, id, fname, lname, GetHashedPassword(password));

MyAdoHelper.DoQuery(database, sql);

}

public static bool IsEmailRegistered(string email)

{

string sql = string.Format("SELECT \* FROM Users WHERE Email='{0}'", email);

return MyAdoHelper.IsExist(database, sql);

}

public static bool IsIDRegistered(string id)

{

string sql = string.Format("SELECT \* FROM Users WHERE ID='{0}'", id);

return MyAdoHelper.IsExist(database, sql);

}

public static void UpdateUser(string oldEmail, string email, string fname, string lname, string password, string id)

{

string sql = string.Format("UPDATE Users SET Email='{0}', ID='{1}', FirstName='{2}', LastName='{3}', PassHash='{4}'",

email, id, fname, lname, GetHashedPassword(password));

sql += " " + string.Format("WHERE Email='{0}'", oldEmail);

MyAdoHelper.DoQuery(database, sql);

}

public static DataTable GetUsers(string searhString)

{

if (searhString == null || searhString == "")

return MyAdoHelper.ExecuteDataTable(database, "SELECT \* FROM Users");

string first = string.Format("FirstName LIKE '%{0}%'", searhString);

string last = string.Format("LastName LIKE '%{0}%'", searhString);

string email = string.Format("Email LIKE '%{0}%'", searhString);

string id = string.Format("ID LIKE '%{0}%'", searhString);

string sql = string.Format("SELECT \* FROM Users WHERE {0} OR {1} OR {2} OR {3}", first, last, email, id);

return MyAdoHelper.ExecuteDataTable(database, sql); ;

}

public static DataTable GetUser(string email)

{

string sql = String.Format("SELECT \* FROM Users WHERE Email='{0}'", email);

return MyAdoHelper.ExecuteDataTable(database, sql); ;

}

public static void Delete(string email)

{

string sql = string.Format("DELETE FROM Users WHERE Email='{0}'", email);

MyAdoHelper.DoQuery(database, sql);

}

public static void SetAdmin(string email, bool admin)

{

string sql = string.Format("UPDATE Users SET Admin='{0}' WHERE Email='{1}'", admin, email);

MyAdoHelper.DoQuery(database, sql);

}

public static bool IsAdmin(string email)

{

string sql = string.Format("SELECT \* FROM USERS WHERE Email='{0}' AND Admin='True'", email);

return MyAdoHelper.IsExist(database, sql);

}

private static string database = "Database.mdf";

private static string GetHashedPassword(string unhashed)

{

string salt = "c5g3uatKK6HECzAqK9uuUr5Xs6hF9n3X";

return GetHashString(unhashed + salt);

}

private static string GetHashString(string unhashed)

{

byte[] inputBytes = System.Text.Encoding.ASCII.GetBytes(unhashed);

System.Security.Cryptography.SHA1 sha = System.Security.Cryptography.SHA1.Create();

byte[] hash = sha.ComputeHash(inputBytes);

System.Text.StringBuilder sb = new System.Text.StringBuilder();

for (int i = 0; i < hash.Length; i++)

sb.Append(hash[i].ToString("x2"));

return sb.ToString();

}

}

### Login.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class Default3 : System.Web.UI.Page

{

public string loginResponse;

protected void Page\_Load(object sender, EventArgs e)

{

string email = Request["loginEmail"];

string password = Request["loginPassword"];

if (email != null && password != null && Request["submit"] != null)

{

email = email.Replace("'", "");

if (DataLink.Exists(email, password))

{

AccessControl.LogIn(this, email, DataLink.IsAdmin(email));

loginResponse = "You have logged in successfully.";

Response.Redirect("Homepage.aspx");

}

else

{

loginResponse = "Email and password don't match";

}

}

}

}

### Registration.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Security.Cryptography;

public partial class \_Default : System.Web.UI.Page

{

public string serverResponse = "";

protected void Page\_Load(object sender, EventArgs e)

{

if (Request["submit"] != null)

{

string firstName = Request["firstName"];

string lastName = Request["lastName"];

string password = Request["userPassword"];

string email = Request["email"];

string id = Request["id"];

if (DataLink.IsEmailRegistered(email))

serverResponse = string.Format("'{0}' is already registered", email);

else if (DataLink.IsIDRegistered(id))

serverResponse = string.Format("'{0}' is already registered", id);

else

{

DataLink.AddUser(email, firstName, lastName, password, id);

serverResponse = "User created";

AccessControl.LogIn(this, email, false);

Response.Redirect("Homepage.aspx");

}

}

}

}

### UpdateDetails.aspx.cs

using System;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Data;

public partial class \_Default : System.Web.UI.Page

{

public string serverResponse = "";

public string formInputs = "";

protected void Page\_Load(object sender, EventArgs e)

{

if (!AccessControl.IsLoggedIn(this))

{

Response.Redirect("Homepage.aspx");

return;

}

DataTable user = DataLink.GetUser(AccessControl.GetLoggedUser(this));

formInputs = ContructInputs(

GetFirstFromTable(user, "FirstName"),

GetFirstFromTable(user, "LastName"),

GetFirstFromTable(user, "ID"),

GetFirstFromTable(user, "Email"));

if (Request["submit"] != null)

{

string firstName = Request["firstName"];

string lastName = Request["lastName"];

string password = Request["userPassword"];

string email = Request["email"];

string id = Request["id"];

if (email != GetFirstFromTable(user, "Email") && DataLink.IsEmailRegistered(email))

serverResponse = string.Format("'{0}' is already registered", email);

else if (id != GetFirstFromTable(user, "ID") && DataLink.IsIDRegistered(id))

serverResponse = string.Format("'{0}' is already registered", id);

else

{

DataLink.UpdateUser(GetFirstFromTable(user, "Email"), email, firstName, lastName, password, id);

AccessControl.LogIn(this, email, DataLink.IsAdmin(email));

WriteScript("alert('Data Updated')");

// If we would have used Response.Redirect, it wouldn't have loaded the alert

WriteScript("window.location = 'Homepage.aspx'");

}

}

}

private string ContructInputs(string first, string last, string id, string email)

{

HtmlInputText inputFirst = new HtmlInputText();

HtmlInputText inputLast = new HtmlInputText();

HtmlInputText inputId = new HtmlInputText();

HtmlInputPassword inputPass = new HtmlInputPassword();

HtmlInputPassword inputRepeatPass = new HtmlInputPassword();

HtmlInputText inputEmail = new HtmlInputText();

inputFirst.ID = "firstName";

inputLast.ID = "lastName";

inputPass.ID = "userPassword";

inputRepeatPass.ID = "repeatPassword";

inputId.ID = "id";

inputEmail.ID = "email";

inputFirst.Value = first;

inputLast.Value = last;

inputId.Value = id;

inputEmail.Value = email;

string br = "<br />";

return

RenderControlToHtml(inputFirst) + br +

RenderControlToHtml(inputLast) + br +

RenderControlToHtml(inputId) + br +

RenderControlToHtml(inputPass) + br +

RenderControlToHtml(inputRepeatPass) + br +

RenderControlToHtml(inputEmail);

}

private string RenderControlToHtml(Control ControlToRender)

{

System.Text.StringBuilder sb = new System.Text.StringBuilder();

System.IO.StringWriter stWriter = new System.IO.StringWriter(sb);

System.Web.UI.HtmlTextWriter htmlWriter = new System.Web.UI.HtmlTextWriter(stWriter);

ControlToRender.RenderControl(htmlWriter);

return sb.ToString();

}

private string GetFirstFromTable(DataTable dt, string columnName)

{

return dt.Rows[0][columnName].ToString().Trim();

}

private void WriteScript(string script)

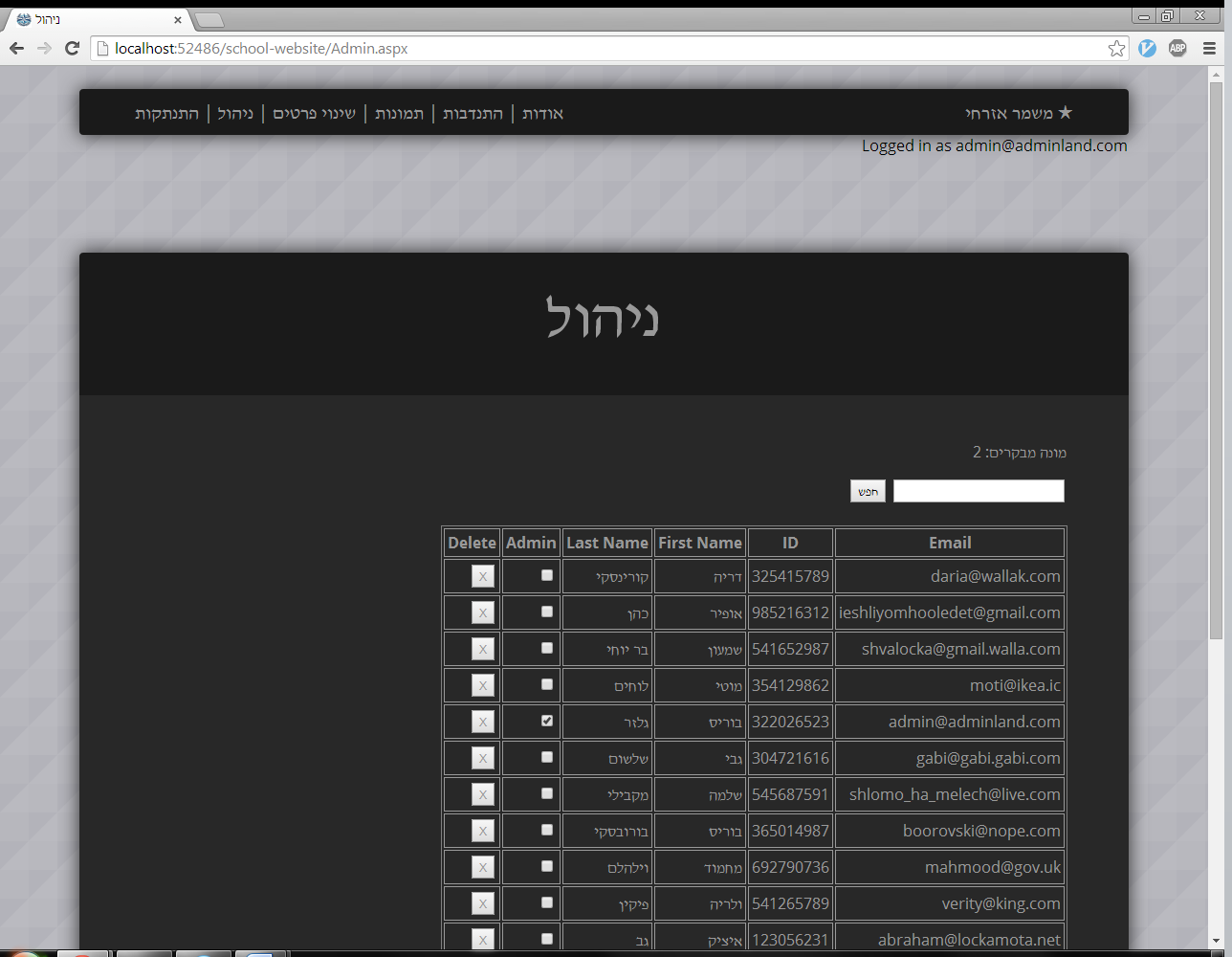
{

Response.Write(string.Format("<script>{0}</script>", script));

}

}

## ניהול האתר



### Admin.aspx

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true" CodeFile="Admin.aspx.cs" Inherits="\_Default" %>

<asp:Content ID="Content1" ContentPlaceHolderID="PageTitle" Runat="Server">

ניהול

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="Head" Runat="Server">

<style>

table, th, td

{

border-width: thin;

border-style: solid;

}

th, td

{

padding: 0.2em;

margin: 0;

}

</style>

<script src="Scripts/FormValidation.js"></script>

<script>

function deleteUser(email) {

var form = document.forms[0];

form.deleteUser.value = email;

form.submit();

}

function setAdmin(email) {

var form = document.forms[0];

form.setAdmin.value = email;

form.submit();

}

function unsetAdmin(email) {

var form = document.forms[0];

form.unsetAdmin.value = email;

form.submit();

}

</script>

</asp:Content>

<asp:Content ID="Content3" ContentPlaceHolderID="MainTitleText" Runat="Server">

ניהול

</asp:Content>

<asp:Content ID="Content4" ContentPlaceHolderID="PageBodyText" Runat="Server">

<p>

מונה מבקרים: <%=visitors %>

</p>

<form runat="server" method="post">

<input type="hidden" value="" name="deleteUser"/>

<input type="hidden" value="" name="setAdmin" />

<input type="hidden" value="" name="unsetAdmin" />

<input type="text" value="" name="search" runat="server" id="searchString"/>

<input type="submit" value="חפש"/>

</form>

<br />

<%=usersTable %>

</asp:Content>

### Admin.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.HtmlControls;

using System.Data;

public partial class \_Default : System.Web.UI.Page

{

public string usersTable = "";

public int visitors;

protected void Page\_Load(object sender, EventArgs e)

{

if (!AccessControl.IsAdmin(this))

Response.Redirect("Homepage.aspx");

if (Request["deleteUser"] != null)

DataLink.Delete(Request["deleteUser"].Trim().Replace("'", ""));

if (Request["setAdmin"] != null)

DataLink.SetAdmin(Request["setAdmin"].Trim().Replace("'", ""), true);

if (Request["unsetAdmin"] != null)

DataLink.SetAdmin(Request["unsetAdmin"].Trim().Replace("'", ""), false);

usersTable = ConstructTable(DataLink.GetUsers(searchString.Value.Replace("'", "")));

}

// This will load after MasterPage.On\_Load()

protected override void OnLoadComplete(EventArgs e)

{

visitors = VisitorCounter.GetCount(this);

base.OnLoadComplete(e);

}

private string ConstructTable(DataTable dt)

{

HtmlTableCell emailHeader = new HtmlTableCell("th");

HtmlTableCell idHeader = new HtmlTableCell("th");

HtmlTableCell firstHeader = new HtmlTableCell("th");

HtmlTableCell lastHeader = new HtmlTableCell("th");

HtmlTableCell adminHeader = new HtmlTableCell("th");

HtmlTableCell deleteHeader = new HtmlTableCell("th");

emailHeader .InnerText = "Email";

idHeader .InnerText = "ID";

firstHeader .InnerText = "First Name";

lastHeader .InnerText = "Last Name";

adminHeader .InnerText = "Admin";

deleteHeader .InnerText = "Delete";

HtmlTableRow headerRow = new HtmlTableRow();

headerRow.Cells.Add(emailHeader);

headerRow.Cells.Add(idHeader);

headerRow.Cells.Add(firstHeader);

headerRow.Cells.Add(lastHeader);

headerRow.Cells.Add(adminHeader);

headerRow.Cells.Add(deleteHeader);

HtmlTable table = new HtmlTable();

table.Rows.Add(headerRow);

for (int i = 0; i < dt.Rows.Count; i++)

{

HtmlTableCell emailCell = new HtmlTableCell();

HtmlTableCell idCell = new HtmlTableCell();

HtmlTableCell firstNameCell = new HtmlTableCell();

HtmlTableCell lastNameCell = new HtmlTableCell();

HtmlTableCell adminCell = new HtmlTableCell();

HtmlTableCell deleteCell = new HtmlTableCell();

HtmlInputCheckBox adminCheckbox = new HtmlInputCheckBox();

string onClickFunction = ((bool)dt.Rows[i]["Admin"]) ? "unsetAdmin" : "setAdmin";

adminCheckbox.Attributes.Add("onclick", string.Format("{0}('{1}')", onClickFunction, dt.Rows[i]["Email"]));

adminCheckbox.Checked = (bool)dt.Rows[i]["Admin"];

HtmlButton deleteButton = new HtmlButton();

deleteButton.Attributes.Add("onclick", string.Format("deleteUser('{0}')", dt.Rows[i]["Email"]));

deleteButton.InnerText = "X";

emailCell .InnerText = (string)dt.Rows[i]["Email"];

idCell .InnerText = (string)dt.Rows[i]["ID"];

firstNameCell .InnerText = (string)dt.Rows[i]["FirstName"];

lastNameCell .InnerText = (string)dt.Rows[i]["LastName"];

adminCell.InnerHtml = RenderControlToHtml(adminCheckbox);

deleteCell.InnerHtml = RenderControlToHtml(deleteButton);

HtmlTableRow row = new HtmlTableRow();

row.Cells.Add(emailCell);

row.Cells.Add(idCell);

row.Cells.Add(firstNameCell);

row.Cells.Add(lastNameCell);

row.Cells.Add(adminCell);

row.Cells.Add(deleteCell);

table.Rows.Add(row);

}

return RenderControlToHtml(table);

}

private string RenderControlToHtml(Control ControlToRender)

{

System.Text.StringBuilder sb = new System.Text.StringBuilder();

System.IO.StringWriter stWriter = new System.IO.StringWriter(sb);

System.Web.UI.HtmlTextWriter htmlWriter = new System.Web.UI.HtmlTextWriter(stWriter);

ControlToRender.RenderControl(htmlWriter);

return sb.ToString();

}

}