

|  |
| --- |
| מגמת מדעי המחשב |
| בית ספר: מקיף צפית |
| סמל בית ספר: 660076 |

מעבדה במדעי המחשב

מבוא לתיכנות בסביבת האינטרנט 1 יח"ל

|  |
| --- |
| שם האתר - ToDi |
| תאור קצר – ToDo lists app manager |

|  |  |
| --- | --- |
| שם התלמיד: | עידו אנגל |
| ת.ז. : | 217389857 |
| כיתה : | י"א 4 |
| שנה : | 2025 תשפ"ה |
| מורה : | אורי שרעבי |

# תוכן העניינים

[תוכן העניינים 2](#_Toc39591286)

[1. מבוא 3](#_Toc39591287)

[א. נושא האתר 3](#_Toc39591288)

[ב. מטרה 3](#_Toc39591289)

[ג. טכנולוגיות 3](#_Toc39591290)

[2. מבנה האתר 3](#_Toc39591291)

[3. מדריך למשתמש 4](#_Toc39591292)

[א. תרשים זרימה של האתר 4](#_Toc39591293)

[ב. תאור דפי האתר 5](#_Toc39591294)

[4. בסיס הנתונים 6](#_Toc39591295)

[א. טבלאות הנתונים 6](#_Toc39591296)

[ב. בדיקות תקינות 6](#_Toc39591297)

[ג. בדיקות אמינות 6](#_Toc39591298)

[5. סיכום אישי 7](#_Toc39591299)

[6. בבליוגרפיה 7](#_Toc39591300)

[7. נספחים 7](#_Toc39591301)

# מבוא

## נושא האתר

* + האתר מאפשר למשתמשים לנהל משימות ולשמור אותן על מנת לנהל זמן

## מטרה

* + לעזור לאנשים לנהל את הזמן שלהם
  + אנשים עסוקים שצריכים כלי זמין לניהול משימות פשוטות
  + הדף העיקרי הוא טבלת המשימות שיש לכל אחד

## טכנולוגיות

* + האתר נבנה בשפות ASP.Net, C#, JavaScript, HTML, CSS
  + באתר יש שימוש במסד נתונים מסוג SQL Server 2019 Express
  + האתר משתמש בcookie על מנת לנתק אוטומטית משתמשים לא מחוברים

# מבנה האתר

הישויות אשר קיימות באתר: ישנם רק משתמשים רשומים, והם רשאים להוסיף ולהסיר משימות

אורח: כלום, לא ניתן להיכנס לאתר ללא משתמש (הגנה עם זריקת שגיאה 403 מותאמת אישית)

חבר/ משתמש רשום: להוסיף משימות ולראות משימות קיימות

מנהל: אין

פונקציות האתר: (לפרט לכל פעילות מה תפקידה ומי משתמש בה )

רישום: למשתמשים לא רשומים שרוצים להשתמש באתר, חובה

התחברות: למשתמשים שכבר נמצאים במסד הנתונים

\*הוספת תוכן: כל המשתמשים המחוברים

\*עדכון פרטי המשתמש:

\*חיפוש משתמשים באתר: אין אפשרות

\*ניהול: אין ניהול כללי

התנתקות: התנתקות חשבון מהאתר

\*זריקת משתמשים לא מחוברים: משתמשים לא מחוברים שמנסים לגשת לדפים נעולים ייזרקו מהם

# מדריך למשתמש

## תרשים זרימה של האתר

דף הבית

חסימת משתמשים לא רשומים

Error403

תפריט אחיד

Consts.Master

כניסת חברים

UserNav.aspx

משתמש חדש

UserNav.aspx

הוספה של משימה חדשה

AddTask.aspx

צפייה במשימות

MostImportant.asxp

Home.aspx

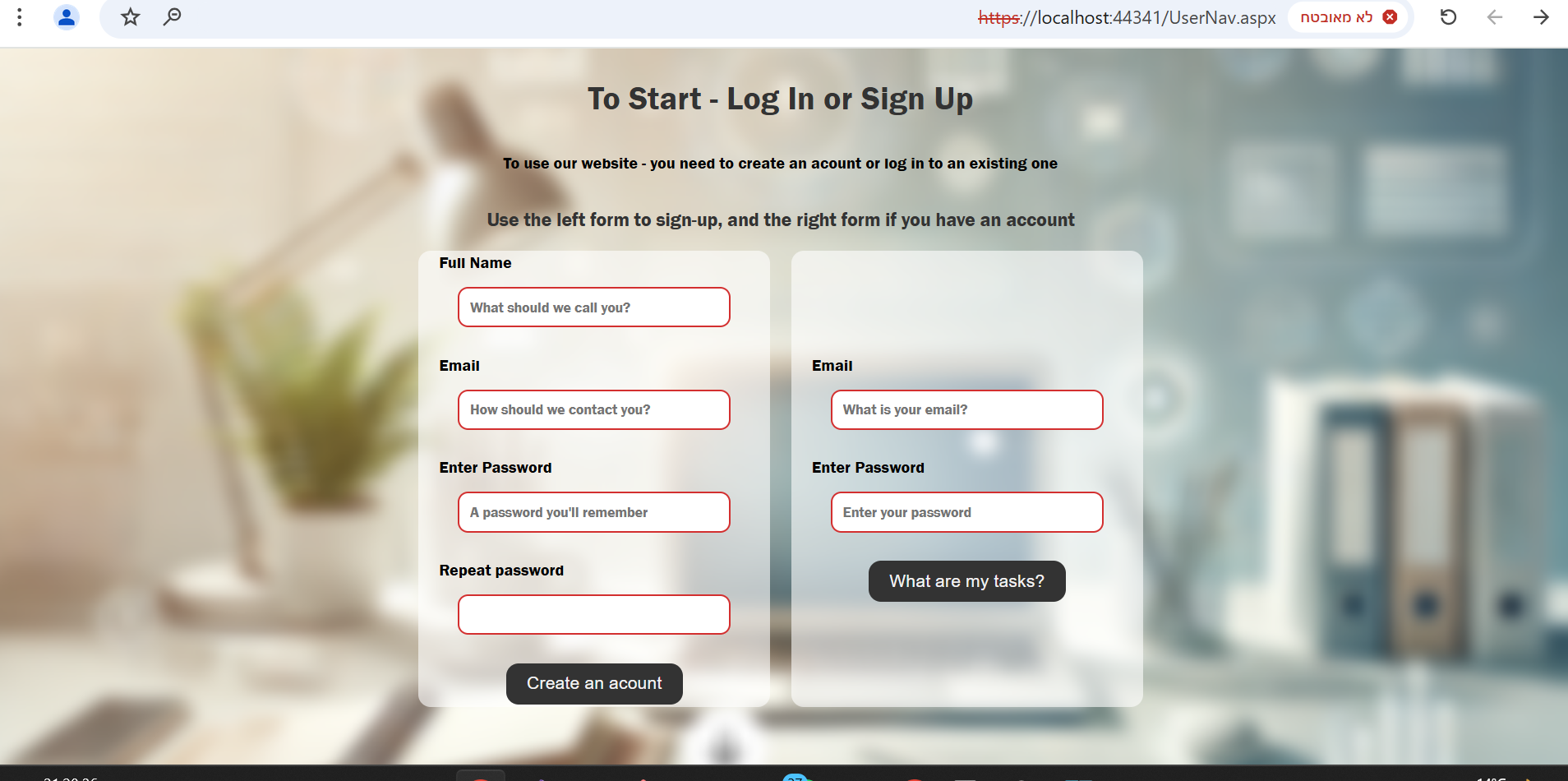
DataBase

לרשומים בלבד קריאה/כתיבה למסד הנתונים

פתוח לכולם קריאה ממסד הנתונים

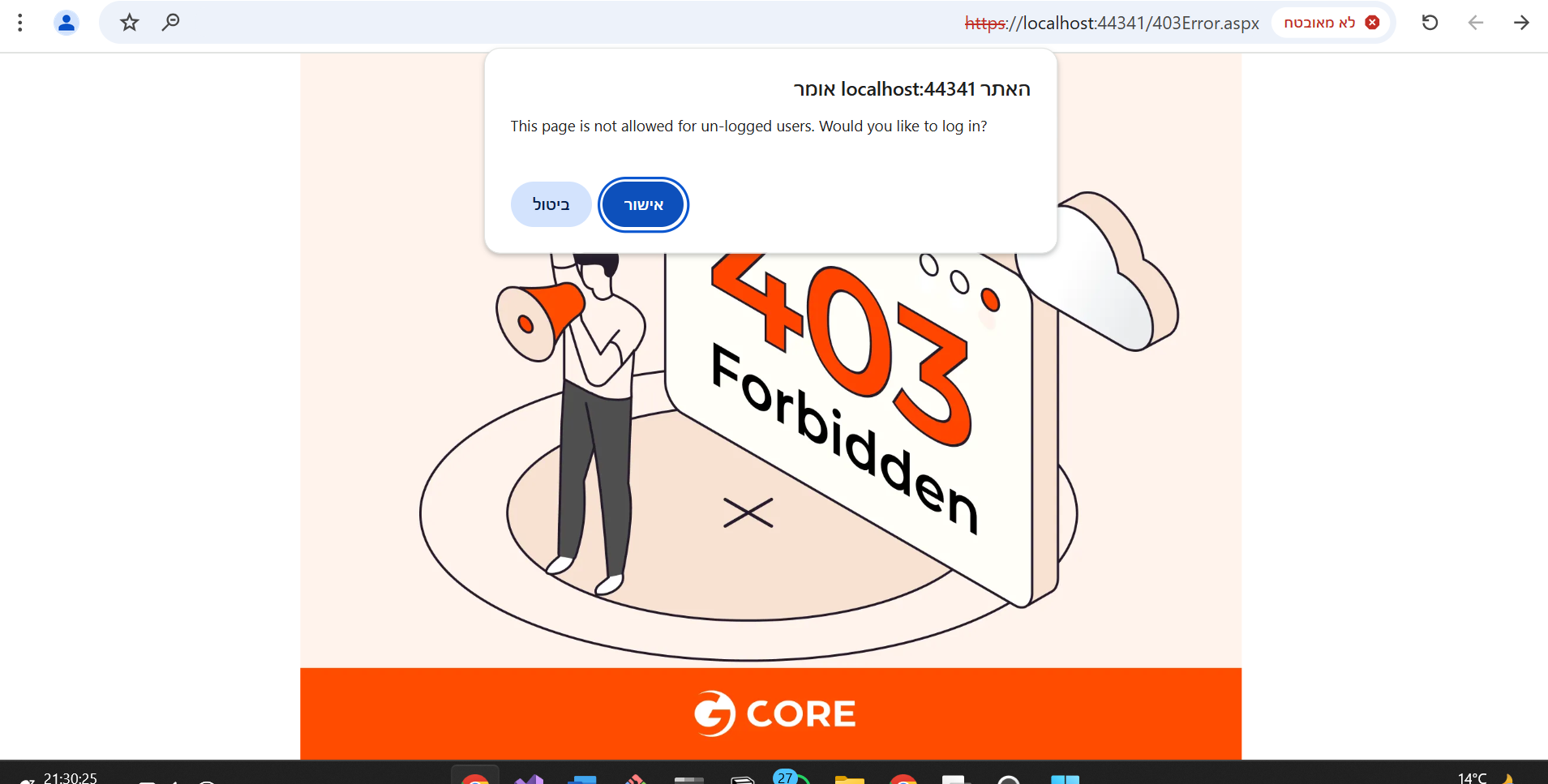
## תאור דפי האתר

* userNav.aspx



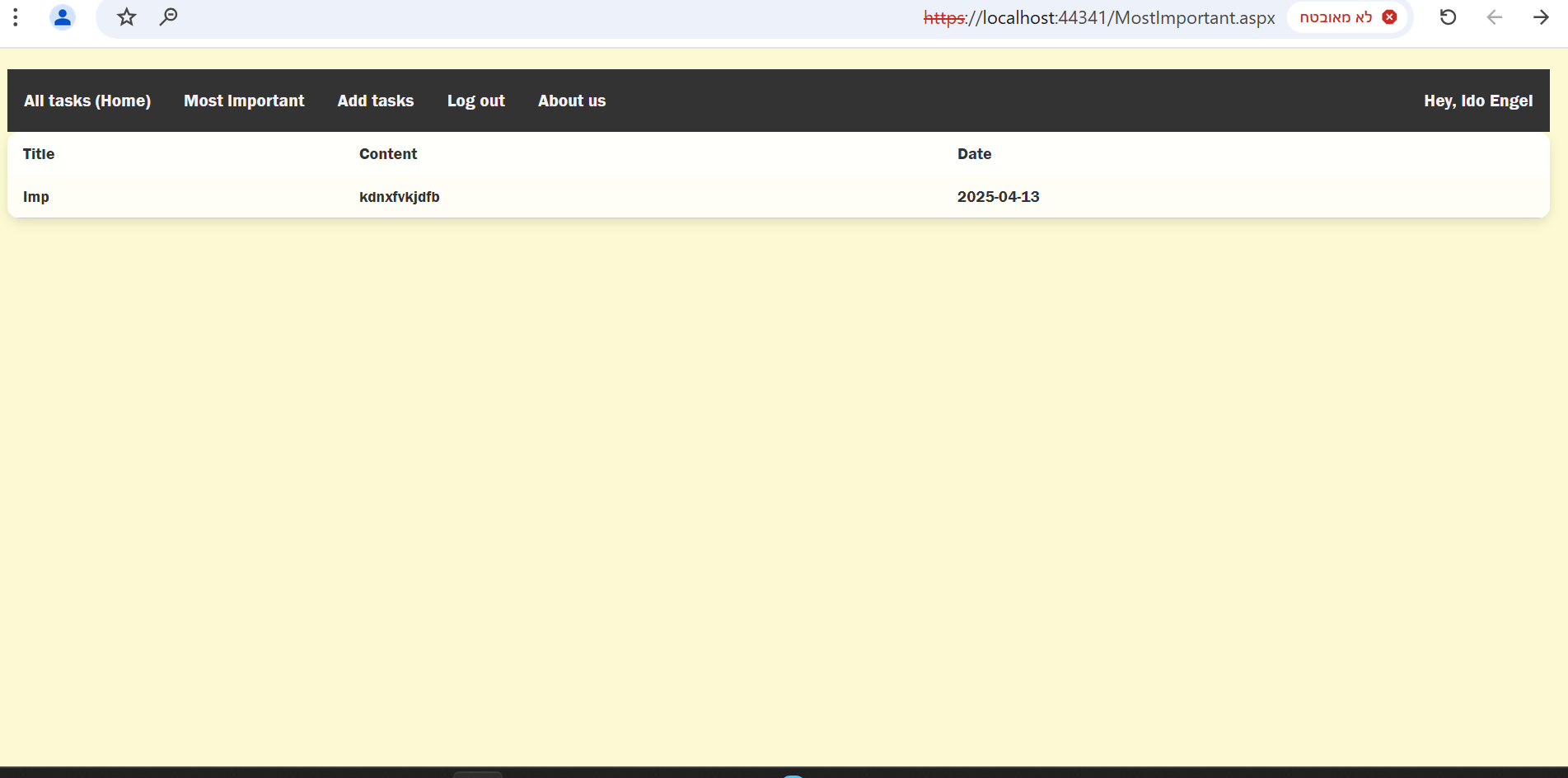
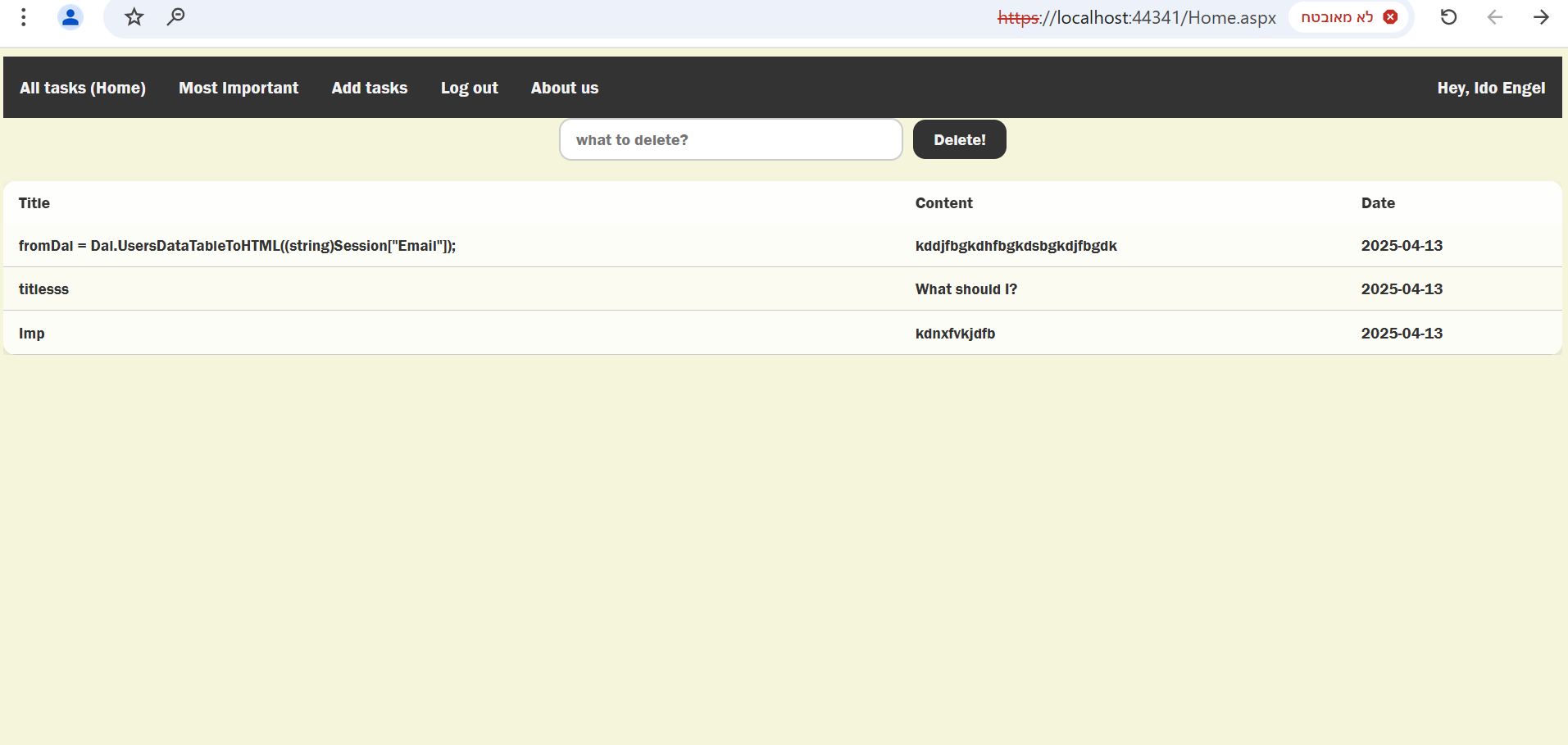
שני טפסים של רישום, אחד למשתמשים קיימים והשני לרישום של משתמשים חדשים. יש למלא את כל השדות כדי להשלים טופס. כפתורי הרישום השחורים תואמים ומקושרים אל הטופס שמעליהם

* Error403.aspx



דף זריקה של משתמשים לא רשומים. במידה ומשתמש לא רשום\לא מחובר מנסה להיכנס אל הדפים הפנימיים של האתר נזרקת לו שגיאה ונאסר עליו להיכנס. עם הצעה לגשת אל אתר הרישום

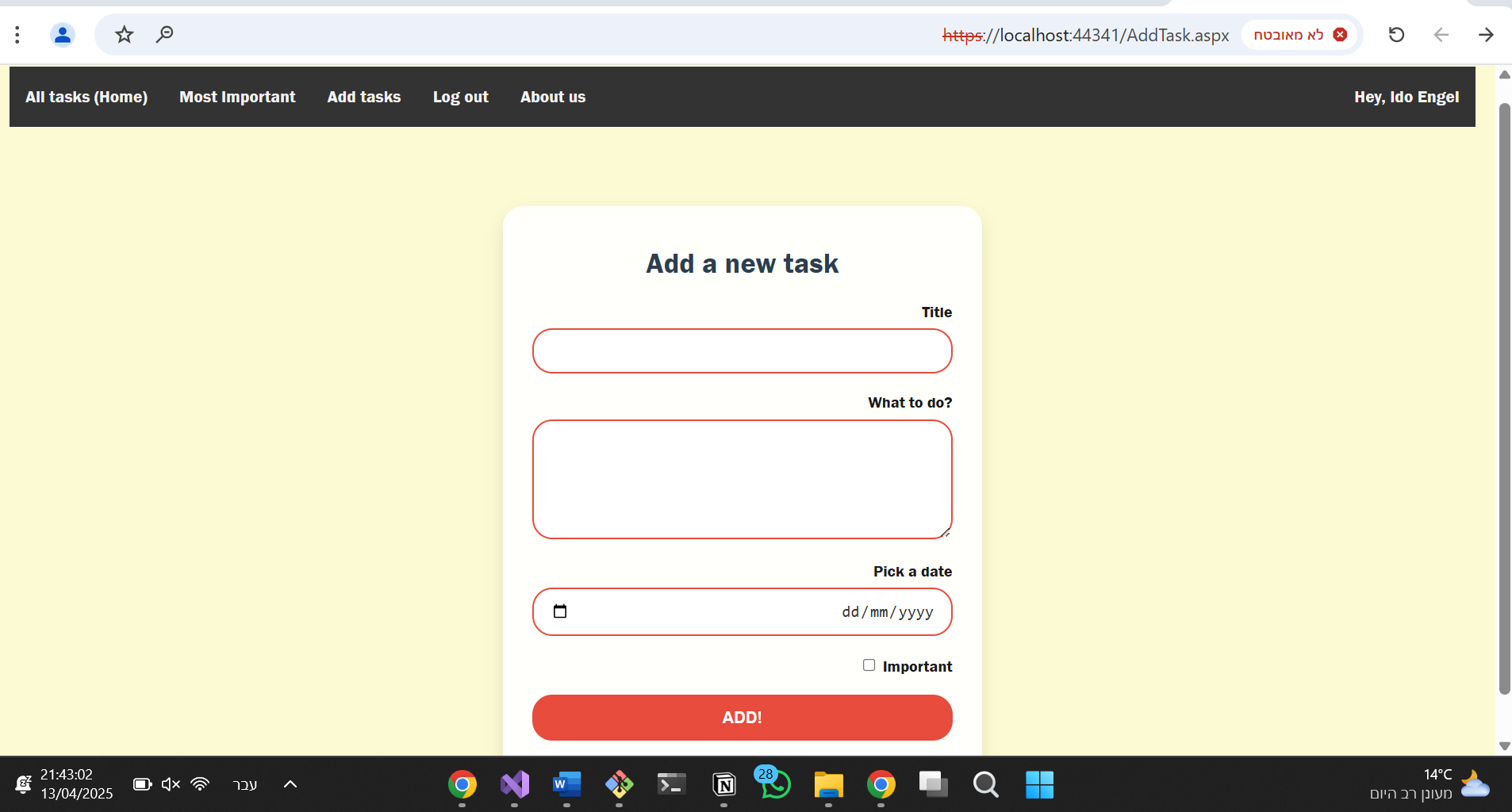
* דפי הצגת משימות – Hone.aspx, MostImportant.aspx



דפים שבמהותם דומים. עליון – Home.aspx, תחתון – MostImportant

למשתמש יש את היכולת לסמן משימות חשובות ושיופיעו בדף נפרד שכולל רק משימות שסומנו כחשובות. בתמונה העליונה קיים כפתור מחיקה (יש להכניס את הכותרת של המשימה). כפתור המחיקה לא קיים בדף החשובים כדי למנוע מחיקה בטעות.

* AddTask.aspx



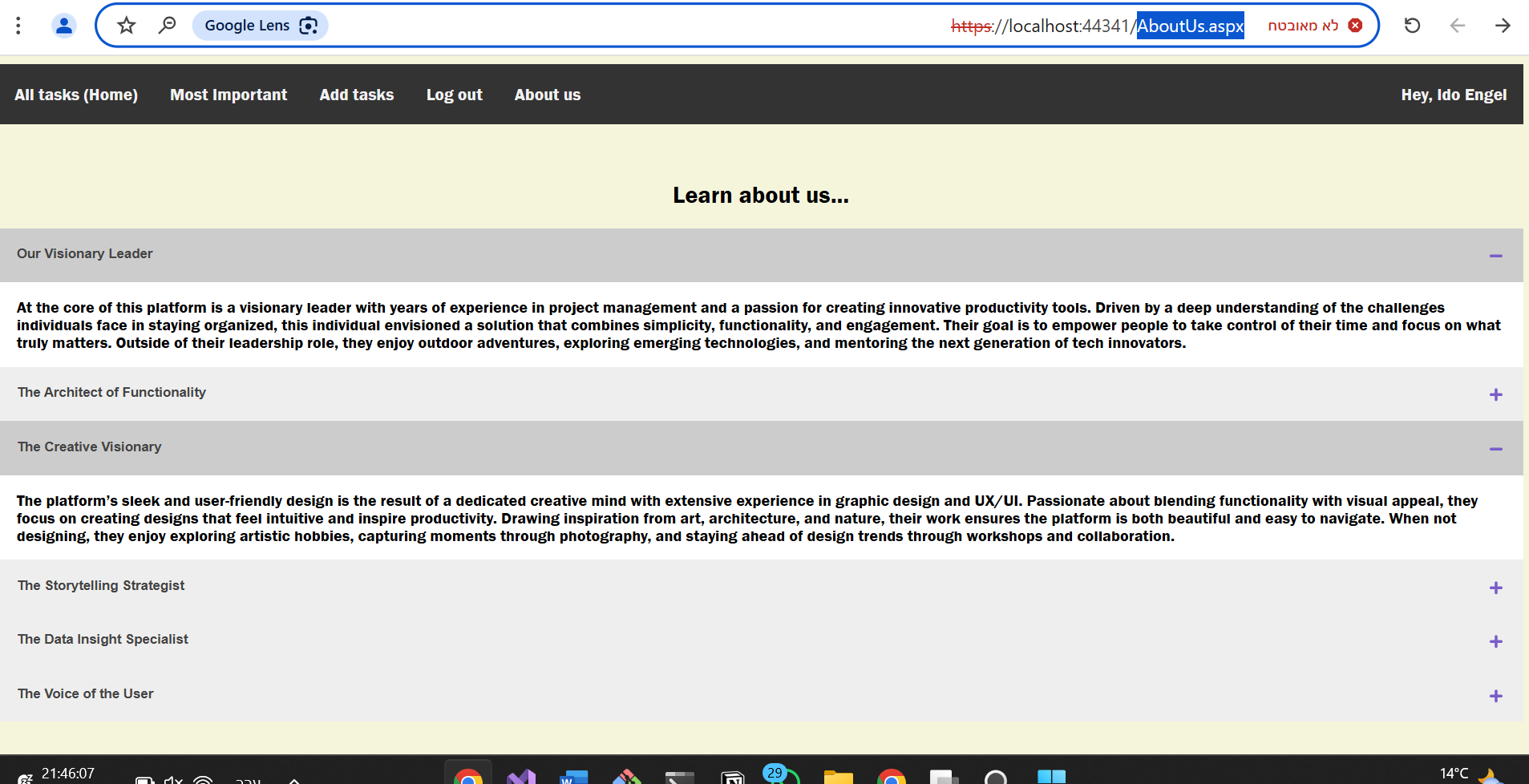
דף הוספה של משימה חדשה למשתמש. על המשתמש להזין כותרת, תיאור, תאריך, והאם המשימה חשובה.

בחירת התאריך היא באמצעות בורר תארכים מובנה של הדפדפן.

סימון המשימה כחשובה יוסיף אותה הדף של Most Important.

הכפתור יצור רשומה בטבלת הDB ויחזיר את המשתמש אל דף הבית

* AboutUs.aspx



דף סטטי שמציג מידע מchatGPT על יוצרי האתר (היוצרים ידועים באינטרנט וgpt מכיר אותם)

# בסיס הנתונים

## טבלאות הנתונים

שם המאגר : DB.mdf

טבלה : משתמשים Users

|  |  |  |  |
| --- | --- | --- | --- |
| שם שדה | שם באנגלית | תפקיד | בדיקות |
| דואר אלקטרוני | Email | שם המשתמש. מפתח ראשי | בדיקה שיש מבנה של אימיל (@, נקודה) |
| סיסמא | Password | סיסמא, לכניסה לאתר | 8 תווים לפחות, מספר אחד, אות אחת גדולה ואחת קטנה, תו מיוחד |
| שם | Name | השם שהמשתמש בחר לקרוא לעצמו | אין, יכול לקרוא לעצמו בכל דרך |
|  |  |  |  |

טבלה: משימות Tasks

|  |  |  |  |
| --- | --- | --- | --- |
| שם שדה | שם באנגלית | תפקיד | הערות |
| מזהה משתמש | User\_Id | קישור בעזרת מפתח חיצוני לטבלת המשתמשים | מקושרת עם אילוצי SQL אל טבלת המשתמשים כדי לבצע מחיקה של כל המשימות במידה ומשתמש מוחק את החשבון שלו. להמנע מזבל |
| כותרת | Title | הכותרת של המשימה | כאן המשתמש יכול להכניס מידע קצר על המשימה |
| תיאור | Content | תיאור ותכולת המשימה. הרחבה | כאן המשתמש יכול להכניס מידע מורחב על המשימה |
| האם חשוב? | IsImportant | המשתמש יכול לבחור האם המשימה חשובה |  |
| תאריך | Date | בחירת תאריך בורר תאריכים מובנה של הדפדפן | בדיקות תקינות ופורמט אוטומטיות בעזרת הבורר המובנה |

## בדיקות תקינות

בדיקות הנעשות בטופס לפני המשלוח. בשפת JS

|  |  |
| --- | --- |
| הבדיקה | תאור בקוד |
| 1.בדיקת חוזק סיסמה | function isPasswordValid() {  const passwordID = "signUp\_password";  const repPasswordID = "repPassword";  let pass = document.getElementById(passwordID);  // בדיקה אם הסיסמה מכילה לפחות אות אחת גדולה  let hasUpperCase = false;  let hasLowerCase = false;  let hasNumber = false;  // לולאת בדיקה לכל התו בסיסמה  for (let i = 0; i < pass.value.length; i++) {  if (/[A-Z]/.test(pass.value[i])) {  hasUpperCase = true; // אות גדולה נמצאה  }  if (/[a-z]/.test(pass.value[i])) {  hasLowerCase = true; // אות קטנה נמצאה  }  if (/\d/.test(pass.value[i])) {  hasNumber = true; // מספר נמצא  }  }  // בדיקה אם הסיסמה מכילה לפחות שמונה תווים  if (pass.value.length < 8) {  return false; // אם יש פחות משמונה תווים  }  // אם כל הדרישות התקיימו  return hasUpperCase && hasLowerCase && hasNumber; // כל הדרישות צריכות להתקיים  } |
| בדיקת חוזק אימייל | function isEmailValid() {  const emailID = "signUp\_email";  let email = document.getElementById(emailID);  const emailPattern = /^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;  return !emailPattern.test(email);  } |
| בדיקה שהסיסמאות זהות | function SamePassword() {  const passwordID = "signUp\_password";  const repPasswordID = "repPassword";  let pass = document.getElementById(passwordID);  let rePass = document.getElementById(repPasswordID);  return (pass.value == rePass.value)  } |

יש למלא את הקוד המתאים לכל בדיקה, טווח אורך, מספרים, נוכחות תו וכל מה שלמדנו

## בדיקות אמינות

בדיקות מול מאגר הנתונים. אלה בדיקות שרתיות שמתבצעות בקבצי ה aspx.cs

|  |  |
| --- | --- |
| הבדיקה | תאור בקוד |
| 1. בדיקה אם שם משתמש כבר קיים ( לפני הוספת משתמש חדש ) 2. בדיקה כניסה בהתחברות למערכת (login ) | public partial class UserNav1 : System.Web.UI.Page  {  public string msg = "";  protected void Page\_Load(object sender, EventArgs e)  {  bool canRedirect = false;  if (Request["signUp\_name"] != null) //the signup form called  {  canRedirect = Dal.InsertUser(Request["signUp\_email"], Request["signUp\_password"], Request["signUp\_name"]);  }  else //the login form called  {  canRedirect = Dal.IsValidUserPassword(Request["logIn\_email"], Request["logIn\_password"]);  }  if (canRedirect)  {  if (Request["signUp\_name"] != null) //the signup form called  {  Session["name"] = Request["signUp\_name"];  Session["email"] = Request["signUp\_email"];  }  else  {  Session["name"] = Dal.GetUserName(Request["logIn\_email"]);  Session["email"] = Request["logIn\_email"];  }    Response.Redirect("Home.aspx");  }  else //if query is not correct  {  if (!(Request["signUp\_name"] == null && Request["logIn\_email"] == null)) //check if not two forms are empty  {  if (Request["signUp\_name"] != null) //the signup form called  {  msg = "User with the same email is already signed up!";  }  else //the login form called  {  msg = "Password not matching the user name<br/> (Did you create an account?)";  }  }    }  }  } |
|  |  |
| 3\*. דפים מוגנים – זריקת משתמשים. מתוך Master | private void checkAccess()  {  // בודק אם יש קוקית שהמשתמש עבר את הבדיקה  HttpCookie accessCookie = Request.Cookies["UserLoggedIn"];  if (accessCookie != null && accessCookie.Value == "true" && Session["name"] != null)  {  // המשתמש כבר עבר את הבדיקה  return;  }  string referer = Request.UrlReferrer?.ToString();  if (referer == null || !referer.Contains("UserNav.aspx"))  {  Response.Redirect("403Error.aspx");  Response.End();  }  else //the user logged in |
| דפים מוגנים – התנתקות אוטומטית | string referer = Request.UrlReferrer?.ToString();  if (referer == null || !referer.Contains("UserNav.aspx"))  {  Response.Redirect("403Error.aspx");  Response.End();  }  else //the user logged in  {  // אם המשתמש עבר את הבדיקה, הגדר קוקית  HttpCookie cookie = new HttpCookie("UserLoggedIn");  cookie.Value = "true"; // ערך הקוקית שמורה על כך שהמשתמש עבר את הבדיקה  cookie.Expires = DateTime.Now.AddHours(0.5); // תקופת זמן שהקוקית תהיה בתוקף  Response.Cookies.Add(cookie);  } |

דפים מוגנים – לוגיקה להתנתקות אוטומטית, זריקת משתמשים

# סיכום אישי

כאן עליכם לספר:

* איך היתה עבורכם העבודה על הפרוייקט
* מה קיבלתם
* מה לקחתי לעתיד: עבודה של מספר סוגי קבצים יחד (תקשורת בין JS לcs )
* מה היה לי קשה: עבודה עם עיצוב וcss
* מה המסקנות שלי: המשתמש יכול להסתפק בconsole לדעתי
* מה הייתי עושה אחרת: מפריד את הטפסים של ההרשמה וההתחברות לשני דפים נפרדים
* ועוד שאלות חקר לשיקולכם
* מצופה לכתוב בין חצי עמוד לעמוד בפרק זה

במהלך העבודה על הפרויקט והגשת אבני הדרך הבנתי בצורה הדרגתית איך עובדים אתרים ואת כמות שפות התכנות שמעורבות ביצירת אפילו אתר פשוט מאוד. ראיתי את כמות הכלים המובנים שיש לכל פיצר, ומאוד הפתיע אותי הפשטות לגשת ולקיים אינטראקציה בין קבצי קוד שונים (סוגי פורמטים שונים). היה מאוד מעצבן שאי אפשר להגדיר נקודת הזנקה לתוכנית והיא מוזנקת על פי העמוד הנוכחי.

ראיתי שמבחינת שפות תכנות, html היא פשוט קובץ טקסט רגיל, ובכל זאת הן ניגשות אל אלמנטים בתוכו.

העבודה עם קבצי עיצוב (css) הייתה מאוד מאתגרת כי היא שונה מאוד, ואפילו יותר מhtml, משפות התכנות שאני רגיל לעבוד איתם.  
אחרי שהגענו לצד השרת הרגשתי בנוח יותר מכיוון שהחלק הלוגי יותר הגיע ויכולתי להפגין את היכולות שלי באופן ישיר יותר.

העבודה עם עיצוב גרמה לי להבין שזה לא התחום שאני רוצה להתעסק בו בעתיד, ושאני אשאיר את התחום הזה לfrontend, אני אתעסק בבנייה של השרתים והמאחורי הקלעים. יותר נהניתי מזה.

הייתי מקדיש יותר זמן בתכנון דפי הhtml, לדוגמא, הפרדה של הטפסים של ההרשמה וההתחברות, רק בשביל לקבל לוגיקה פחות מסורבלת לעין בצד הcs.

בזמן הלמידה הצלחתי ללמוד דברים עם הזמן בנושאים שאני לא מבין בהם בכלל ובלי בסיס אמיתי ואני מאמין שזה ישרת אותי בעתיד. בגלל שהפרויקט היה ארוך, למדתי על קונספטים כמו regex וDB בלי קשר לפרויקט אבל בזמן העבודה עליו, וזה העמיק את הידע שלי על שורות הקוד שרשמתי.

בנוסף, בעקבות זה שעזרתי הרבה לאחרים, למדתי יותר לקרוא קוד של אנשים אחרים (וגם בשפות שאני לא מתמצא בהן מאוד) וזה עזר לי לרשום קוד ברור יותר וגם בהבנה של קוד באופן כללי, וזה באמת יכולת שאני אקח איתי לעתיד

# נספחים

נעשה שימוש בעוגיות כדי לנתק משתמשים אוטומטית מהאתר אם לא היו פעילים זמן רב, זה הצריך מחקר באינטרנט כדי להבין איך להכניס את העוגיות לאתר שלי