תכנות באינטרנט

ASP.Net Core

Razor pages

גלעד מרקמן

קריית החינוך פארק המדע,
נס ציונה

Microsoft
ASP.NET Core

GM
Internet Programing
Courses

עבודה בשיטה
הלא מקושרת

סינון ומיון טבלה
asp-page-handler

Games of throne Home Razor Pages Login Register Users welcome Gilad Logout

UsersTable

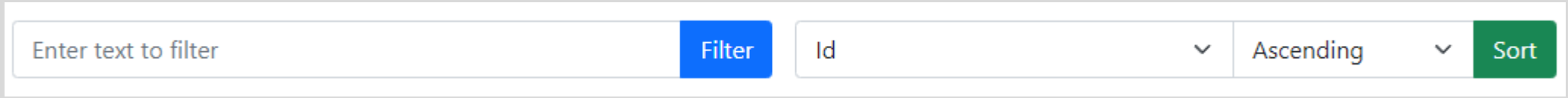
Enter text to filter Filter Id Ascending Sort 0 Delete

Id	Username	Password	Firstname	Lastname	Email	Phone	Admin	Birthday	
1	Gilad	1968	Gilad	Markman	gilad@markman.co.il	052-2653722	True	10/01/1968 00:00:00	update
2	Orly	1234	Orly	Markman	orly@gmail.com	052-6458789	False	06/08/1969 00:00:00	update
3	Yoav	123	Yoav	Markman	Yoav@gmail.com	052-9876112	False	21/02/2011 00:00:00	update
6	Yoram	Yoram1111	Yoram	Hagay	Yoram@gmail.co.il	052-9876540	False	06/06/2024 00:00:00	update

שלבי העבודה - חזרה

- התחברות למסד הנתונים
- SqlConnection con = new SqlConnection(connectionString);
- בניית פקודת SQL
- SqlCommand cmd = new SqlCommand(“SQL String”, con);
- בניית DataAdapter
- SqlDataAdapter adapter = new SqlDataAdapter(cmd);
- בניית DataSet לאכסון נתונים מקומי
- DataSet ds = new DataSet();
- טעינת הנתונים ממסד הנתונים ל DataSet
- adapter.Fill(ds, "users");
- עדכון הנתונים ב DataSet
- ....
- עדכון בסיס הנתונים על פי ה DataSet המעודכן
- SqlCommandBuilder builder = new SqlCommandBuilder(adapter);
- adapter.Update(ds, “users");

עדכון דף ה View



- נוסיף בראש הדף שני טפסים שונים.
- טופס לסינון הכולל:
  - שדה בשם filter
  - כפתור Filter
- טופס מיון
  - תיבת בחירה עם שמות העמודות
  - תיבת בחירה עם סדר המיון (מיון עולה או יורד)
  - כפתור Sort.

View דף ה

```
<div class="input-group mb-3">
    <form method="post" asp-page-handler="Filter" style="width:40%">
        <div class="input-group mb-3">
            <input asp-for="filter" type="text" class="form-control" placeholder="Enter text to filter">
            <button type="submit" class="btn btn-primary me-3" type="button" id="Filter_btn">Filter</button>
        </div>
    </form>
    <form method="post" asp-page-handler="Sort" style="width:40%">
        <div class="input-group mb-3">
            <select asp-for="column" class="form-select" style="width:50%">
                <option value="Id" selected>Id</option>
                <option value="Username">Username</option>
                <option value="Email">Email</option>
                <option value="Firstname">Firstname</option>
                <option value="Lastname">Lastname</option>
            </select>
            <select asp-for="Order" class="form-select" style="width:25%">
                <option value="ASC">Ascending</option>
                <option value="DESC">Desending</option>
            </select>
            <button class="btn btn-success me-3" type="Submit" id="Sort_btn">Sort</button>
        </div>
    </form>
```

asp-page-handler

- אנו מעוניינים שלחיצה על כל כפתור בדף תפעיל פונקציה אחרת בצד השרת.
- לצורך כך יצרנו לכל כפתור טופס post נפרד עם asp-page-handler.

```
<div class="input-group mb-3">
    <form method="post" asp-page-handler="Filter" style="width:40%">...
    <form method="post" asp-page-handler="Sort" style="width:40%">...
    <form method="post" asp-page-handler="Delete" style="width:20%">...
</div>
```

- ה- handler יאפשר לנו ליצור בצד השרת פונקציה נפרדת לכל כפתור.

```
<div class="input-group mb-3"> flex
  <form method="post" style="width:40%" action="/Users?handler=Filter">...</form>
  <form method="post" style="width:40%" action="/Users?handler=Sort">...</form>
  <form method="post" style="width:20%" action="/Users?handler=Delete">...</form>
</div>
```

פונקציות עם Handler

- בצד השרת אנחנו יכולים ליצור פונקציות Post נפרדת לכל Handler.
- שם ה- Handler מתווסף לשם הפונקציה.
- כך, טופס עם "Handler=Filter" יקרא לפונקציה()OnPostFilter .

```
<div class="input-group mb-3">
    <form method="post" asp-page-handler="Filter" style="width:40%">...
    <form method="post" asp-page-handler="Sort" style="width:40%">...
    <form method="post" asp-page-handler="Delete" style="width:20%">...
</div>
```

```
public IActionResult OnGet()...

public IActionResult OnPostFilter()...

public IActionResult OnPostSort()...

public IActionResult OnPostDelete()...
```

סינון Filter

- נוסיף מאפיין הקשור לשדה ה Filter.
- נכתוב את הפעולה השולפת טבלה מסוננת מבסיס הנתונים בהתאם למחרוזת הסינון שכתב המשתמש. הסינון מבוצע מול שדות שם פרטי ומשפחה.

```
[BindProperty]
public string filter { get; set; } = string.Empty;
```

```
public IActionResult OnPostFilter()
{
    Helper helper = new Helper();
    string SQL;
    if (filter == string.Empty)
    {
        SQL = "SELECT * FROM Users";
    }
    else
    {
        SQL = $"SELECT * FROM Users WHERE Firstname LIKE '%{filter}%' OR Lastname Like '%{filter}%'";
    }
    dt = helper.RetrieveTable(SQL, "Users");
    return Page();
}
```

Sort - מיון

- הפעם נשתמש בשני שדות: שם העמודה לפי נמיין וסדר המיון.
- נכתוב את הפעולה השולפת את הטבלה כשהיא ממויינת מבסיס הנתונים ומדפיסה אותה.

```
public string column { get; set; }
[BindProperty]
public string Order { get; set; }
[BindProperty]
```

```
public IActionResult OnPostSort()
{
    Helper helper = new Helper();
    string SQL = $"SELECT * FROM Users ORDER BY {column} {Order}";
    dt = helper.RetrieveTable(SQL, "Users");
    return Page();
}
```

Login - כניסה

בשיטה הלא מקושרת

Login

- נבצע שאילתא לבדיקה אם קיימת רשומה עם שם משתמש וסיסמה שהוזנו בטופס
- הפעולה OnGet מחזירה טופס ריק.
- הפעולה OnPost מבצעת את הבדיקה מול בסיס הנתונים.

```
public class Login_SQLModel : PageModel
{
    public string msg { get; set; } = string.Empty;
    [BindProperty]
    public string Username { get; set; }
    [BindProperty]
    public string password { get; set; }

    public void OnGet()
    {
    }

    public IActionResult OnPost()
    {
        string SQLStr = $"SELECT * FROM Users WHERE Username LIKE '{Username}' AND Password LIKE '{password}'";
        Helper helper = new Helper();
        DataTable dt = helper.RetrieveTable(SQLStr, "Users");

        if (dt.Rows.Count > 0)
        {
            HttpContext.Session.SetString("Login", Username);
            HttpContext.Session.SetString("Admin", dt.Rows[0]["Admin"].ToString());
            return RedirectToPage("/Index");
        }
        msg = "Wrong username or password.";
        return Page();
    }
}
```

- נבדוק אם מספר הרשומות בטבלה גדול מ- 0. אם כן, סימן שיש משתמש מתאים.

תקינות קלט

- תזכורת – חובה לבצע תקינות קלט בצד המשתמש.
- למשל – אם המשתמש לא יכניס נתונים נקבל שגיאת הרצה.

עדכון בסיס הנתונים
Insert

עבודה בשיטה הלא מקושרת

שלבי העבודה - חזרה

- התחברות למסד הנתונים
- SqlConnection con = new SqlConnection(connectionString);
- בניית פקודת SQL
- SqlCommand cmd = new SqlCommand(“SQL String”, con);
- בניית DataAdapter
- SqlDataAdapter adapter = new SqlDataAdapter(cmd);
- בניית DataSet לאכסון נתונים מקומי
- DataSet ds = new DataSet();
- טעינת הנתונים ממסד הנתונים ל DataSet
- adapter.Fill(ds, "users");
- עדכון הנתונים ב DataSet
- ....
- עדכון בסיס הנתונים על פי ה DataSet המעודכן
- SqlCommandBuilder builder = new SqlCommandBuilder(adapter);
- adapter.Update(ds, “users");

טופס ההרשמה - Register

- נבנה טופס שכולל את כל השדות ברשומה של המשתמש.
- בתחתית הטופס נוסיף מקום להודעות שגיאה מהשרת (כגון: שם המשתמש כבר קיים).
- הפעולה OnGet – תחזיר למשתמש טופס ריק.
- הפעולה OnPost – תבצע את העדכון בבסיס הנתונים.
- בצד השרת אנחנו נשתמש במאפיין מסוג User הכולל את כל השדות הנדרשים. הזנה בטופס תעדכן את המאפיין הרלוונטי באובייקט.
- את האובייקט נשלח לפעולה Insert ב Helper. הפעולה תחזיר 1 אם ההוספה הצליחה ו 1- אם המשתמש כבר קיים.

דף view

```
<h1>Register</h1>

<h4>User</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form method="post">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="NewUser.FirstName" class="control-label"></label>
                <input asp-for="NewUser.FirstName" class="form-control" />
            </div>
            <div class="form-group">
                <label asp-for="NewUser.LastName" class="control-label"></label>
                <input asp-for="NewUser.LastName" class="form-control" />
            </div>
            <div class="form-group">
                <label asp-for="NewUser.Username" class="control-label"></label>
                <input asp-for="NewUser.Username" class="form-control" />
            </div>
            <div class="form-group">
                <label asp-for="NewUser.Password" class="control-label"></label>
                <input asp-for="NewUser.Password" class="form-control" />
            </div>
            <div class="form-group">
                <label asp-for="NewUser.Email" class="control-label"></label>
                <input asp-for="NewUser.Email" class="form-control" />
            </div>
```

```
            <div class="form-group">
                <label asp-for="NewUser.Phone" class="control-label"></label>
                <input asp-for="NewUser.Phone" class="form-control" />
            </div>
            <div class="form-group">
                <label asp-for="NewUser.Birthday" class="control-label"></label>
                <input asp-for="NewUser.Birthday" class="form-control" />
            </div>
            @if(Model.msg != "")
            {
                <div class="form-group">
                    <div class="alert alert-danger" role="alert">
                        @Model.msg
                    </div>
                </div>
            }
            <div class="form-group">
                <input type="submit" value="Create" class="btn btn-primary" />
            </div>
        </form>
    </div>
</div>
```

```cs
public class RegisterModel : PageModel
{
    [BindProperty]
    public User NewUser { get; set; }
    public string msg { get; set; } = string.Empty;

    public void OnGet()
    {
    }
    public IActionResult OnPost()
    {
        Helper helper = new Helper();
        int n = helper.Insert(NewUser, "Users");
        if (n == -1)
        {
            msg = "Username already taken.";
            return Page();
        }
        return RedirectToPage("Index");
    }
}
```

- המאפיין NewUser – קשור לטופס.
- המאפיין msg – נועד לכתיבת הודעות שגיאה.
- הפעולה OnPost קוראת לפעולה Insert ב Helper אשר מבצעת את עדכון בסי הנתונים.
- אם מוחזר 1- סימן ששם המשתמש כבר קיים.
- לא לשכוח בדיקת טופס ב JS.

הפעולה Insert

```
public int Insert(User user, string table)
// The Method recieve a user objects and insert it to the Database as new row.
// if the user is already taken the method will return -1.
{
    // התחברות למסד הנתונים
    SqlConnection con = new SqlConnection(conString);

    // בניית פקודת SQL
    string SQLStr = $"SELECT * FROM {table} WHERE Username Like '{user.Username}'";
    SqlCommand cmd = new SqlCommand(SQLStr, con);

    // בניית DataSet
    DataSet ds = new DataSet();

    // טעינת סכימת הנתונים
    SqlDataAdapter adapter = new SqlDataAdapter(cmd);
    adapter.Fill(ds, table);

    if (ds.Tables[table].Rows.Count > 0)
    {
        return -1;
    }
```

- אנחנו מבצעים שאילתא עם שם המשתמש החדש שהוזן.
- אם קיים כבר שם משתמש כזה אנחנו מחזירים 1- ולא מעדכנים.
- אם לא קיים שם משתמש כזה אנחנו ממשיכים.

הפעולה Insert

```
    // בניית השורה להוספה
    DataRow dr = ds.Tables[table].NewRow();
    dr["Firstname"] = user.FirstName;
    dr["Lastname"] = user.LastName;
    dr["Username"] = user.Username;
    dr["Password"] = user.Password;
    dr["Email"] = user.Email;
    dr["Phone"] = user.Phone;
    dr["Birthday"] = user.Birthday.ToString();

    ds.Tables[table].Rows.Add(dr);

    // עדכון הדאטה סט בבסיס הנתונים
    SqlCommandBuilder builder = new SqlCommandBuilder(adapter);
    int n = adapter.Update(ds, table);
    return n;
}
```

- יוצרים שורה חדשה DataRow באמצעות הטבלה, כך שהיא תואמת למבנה הטבלה.
- מזינים לשדות בשורה החדשה את הנתונים מהאובייקט User.
- מוסיפים את השורה לטבלה.
- מעדכנים את הטבלה בבסיס הנתונים.
- מקבלים את מספר השורות ששונו (תמיד יהיה 1).

Update, Delete

- פעולות עדכון ומחיקה של משתמש יודגמו באמצעות השיטה המקושרת.

- למימוש פעולות אילו בשיטה הלא מקושרת הנכם מופנים אל הפעולות ב Helper:
  - Update_disconnected.
  - Delete_disconnected.

קישור למחלקה Helper ב- GitHub

סיכום

- הדגמנו פעולות שונות באמצעות השיטה הלא מקושרת:
  - סינון ומיון
  - כניסה Login
- הדגמנו פעולות עדכון של בסיס הנתונים באמצעות השיטה הלא מקושרת. פעולת ההרשמה Register המוסיפה רשומה לטבלה.
- למדנו על asp-page-handler. כיצד ניתן ליצור פעולות OnPost שונות בצד השרת, כגון: OnPostFilter, OnPostSort