Summary Module

Final Project and Cloud Deployment

דגשים לפרוייקט מסכם
חזרה על נושאים נבחרים
העלאה לענן

TomerBu

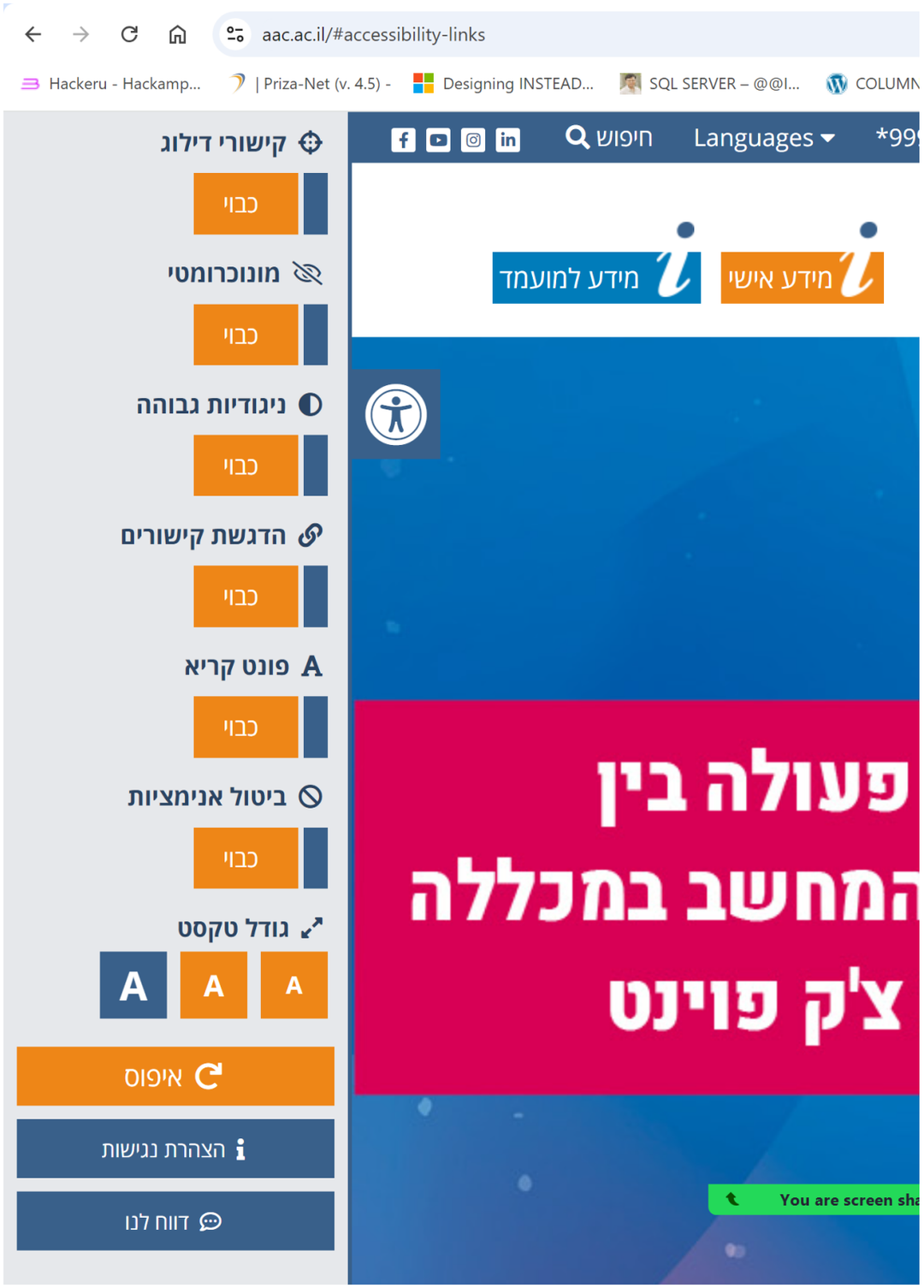
מבנה הפרוייקט - פרוייקט מסכם
צד לקוח react
צד שרת - asp net core web api

תבניות עיצוב:

תבניות/פתרונות מפורסמים לבעיות מוכרות:
דוגמאות לתבניות שעבדנו איתן:

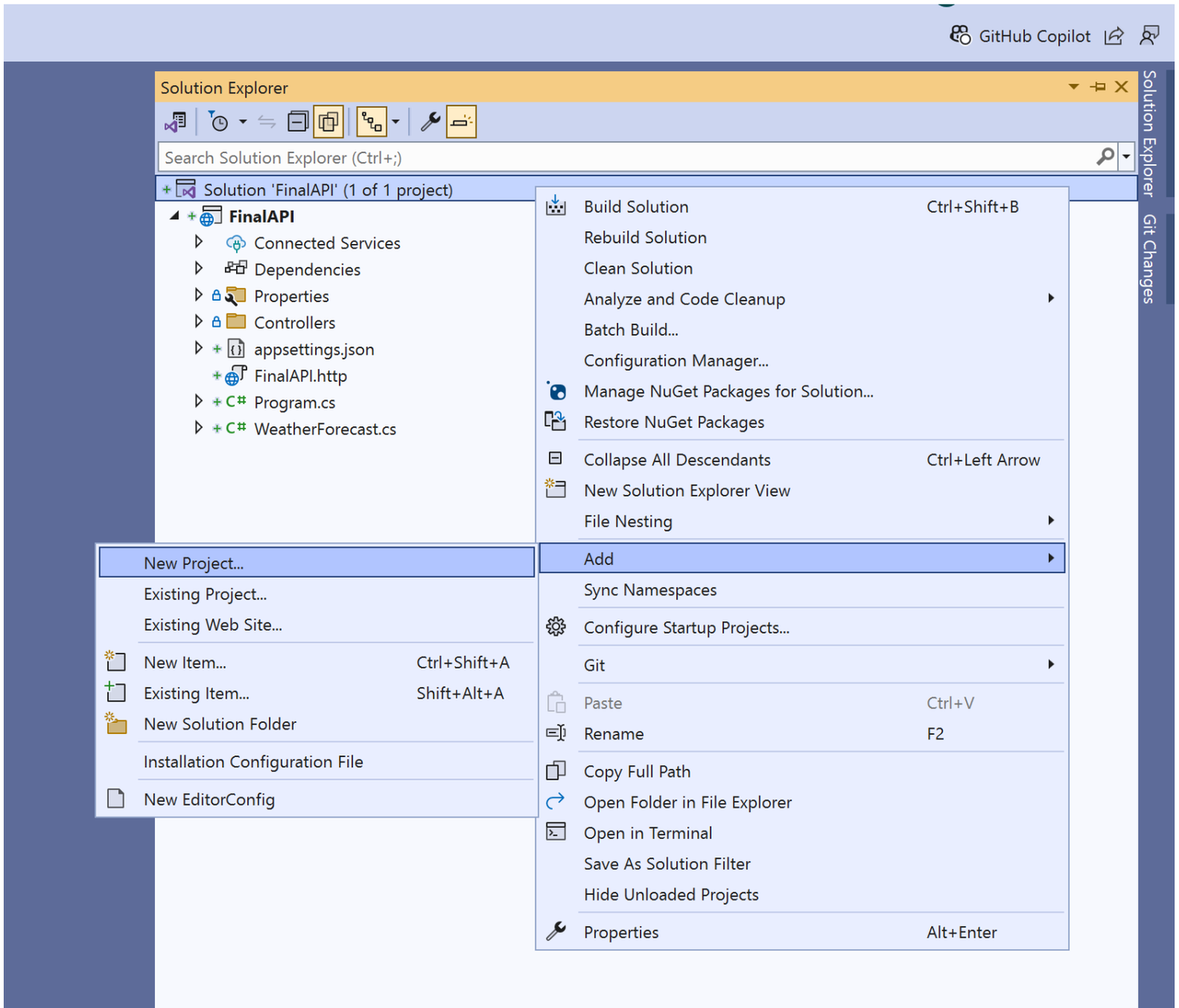
DTO
DI
Repository
Singleton
Factory method

סרגל נגישות -
לפי החוק במדינת ישאל - חובה ליצור הצהרת נגישות, ביטול אנימציות, שינוי צבעים, הגדלת פונטים, הסבר לכל תמונה, ואפשרות לניווט עם מקלדת



פרוייקט חדש - asp net web api

עבודה על פרוייקט מסכם



פרוייקט חדש - asp net web api

עבודה על פרוייקט מסכם

לטובת מודולציה - ניצור פרוייקט נוסף בתוך הsolution:

אפשר ליצור את זה כclass library - כי כך אנחנו נשתמש בזה
אבל יהיה לנו יותר נוח לבצע ***scafolld*** עם פרוייקט מסוג API

שם הפרוייקט **DAL**

בפרוייקט DAL ניצור תיקית מודלים models

פרוייקט חדש - asp net web api

מודלים: מוצר וקטגוריה:

```
using System.ComponentModel.DataAnnotations.Schema;

namespace DataAccess.Models;

public class Product
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Description { get; set; }


    [Column(TypeName = "money")]
    public decimal Price { get; set; }
    public string ImageUrl { get; set; }

    //Likes/Rating (when we add users/identity)


    //Navigation props:
    public int CategoryId { get; set; }

    public Category Category { get; set; }
}
```

```
namespace DataAccess.Models;

public class Category
{
    public int Id { get; set; }
    public string Name { get; set; }

    //Navigation Props:
    public ICollection<Product> Products { get; set; }
}
```

תזכורת:

בעבודה עם Entity Framework
אנחנו יוצרים מחלקות והספריה מסייעת
לנו ליצור את הדטה-בייס ולתקשר עם
הדטהבייס ללא צורך בכתיבת SQL

פרוייקט חדש - asp net web api

תיקון של הQuery String

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "ContextDAL": "Server=DESKTOP-GHIEC0F\\SQLEXPRESS;Database=FinalProjectDB;TrustServerCertificate=True;Trusted_Connection=True;MultipleActiveResultSets=true"
  }
}
```

שם השרת

שם הדטה-בייס הרצוי

TrustServerCertificate=True

(השרת עובד עם חתימת SSL
לטובת אבטחת התקשורת
אבל החתימה לא נוצרה ע"י
Certified Certificate Authority)

כשנעבוד עם ענן יהיה לנו SSL תקני לחלוטין

פרוייקט חדש - asp net web api

ביצוע מיגרציות / עדכון הדטה-בייס

```
Add-Migration InitDB
```

```
Update-Database
```

יש לנו דטה-בייס.

תזכורת Repository:

מחלקה שבה נכתוב מתודות לעבודה עם הדטה-בייס שלנו
המתודות יפשטו את העבודה עם Entity Framework
במקום שנצטרך לזכור לבצע Save Changes בכל פעם נוכל לכתוב
מתודה לנוחיות וסדר.

תזכורת Repository:

בשיעור הבא ניצור Repository גנרי
וכך נוכל למעשה ליצור Repository לכל אחד מהטיפוסים בפרוייקט שלנו מבלי להעתיק קוד

עקרון DRY = Don't repeat yourself

עלינו לשאוף לכתיבה של קוד פעם אחת
ולהשתמש באותו קוד בכל מקום
כך כשנעדכן - העדכון יהיה במקום אחד בלבד (ולא בכל המקומות שאליהם העתקנו).

```
using DataAccess.Models;

namespace DAL.Data;

public class Repository(ContextDAL context)
{
    public void AddProduct(Product product)
    {
        context.Products.Add(product);
        context.SaveChanges();
    }

    public void DeleteProduct(Product product)
    {
        context.Products.Remove(product);
        context.SaveChanges();
    }

    public void DeleteProduct(int id)
    {
        var product = context.Products.FirstOrDefault(p => p.Id == id) ?? throw new Exception("Item not found");
        context.Products.Remove(product);
        context.SaveChanges();
    }
}
```

שיעורי בית:

זה הזמן לחשוב על רעיון לפרוייקט.

מי שכבר מצא רעיון - מוזמנים לממש את הרעיון שלכם
ולהתחיל לעבוד על הפרוייקט המסכם לפי ההנחיות שחולקו.