Steps to run the project:

1. Ensure visual studio is downloaded on the machine

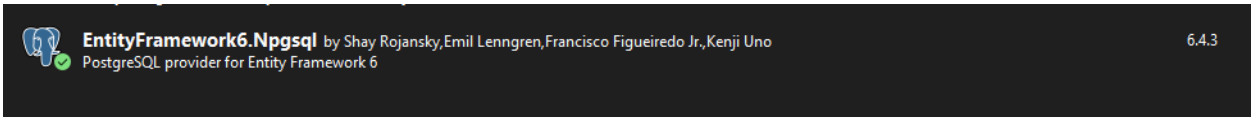
2. Open up the .**csproj** file or **.sln** file stored in the FinalProject.Zip

3. Add an environmental variable onto your machine that contains the connection string to YOUR specific database. This allows the **GetConnectionString()** method to properly return the connection string to your local database:

```
33 references
public class ApplicationDbContext : IdentityDbContext<ApplicationUser>
{
    18 references
    public DbSet<Topic> Topics { get; set; }
    10 references
    public DbSet<Article> Articles { get; set; }
    14 references
    public DbSet<Source> Sources { get; set; }

    16 references
    public ApplicationDbContext()
        : base(GetConnectionString(), throwIfV1Schema: false)
    {
    }

    1 reference
    private static string GetConnectionString()
    {
        var env = Environment.GetEnvironmentVariable("TG_DB_CONN");
        if (!string.IsNullOrEmpty(env))
            return env;

        return "Host=localhost;Port=5432;Database=unKnown;Username=unKnown;Password=unKnown;";
    }
```

Note: Usually the connection string is hardcoded into the webconfig however an environmental variable was used here to improve security and hide secrets

4. For whatever database you are connecting to, ensure that the NuGet package for connectivity to that database is installed in your visual studio (my project uses EntityFramework6.Npgsql 6.4.3)

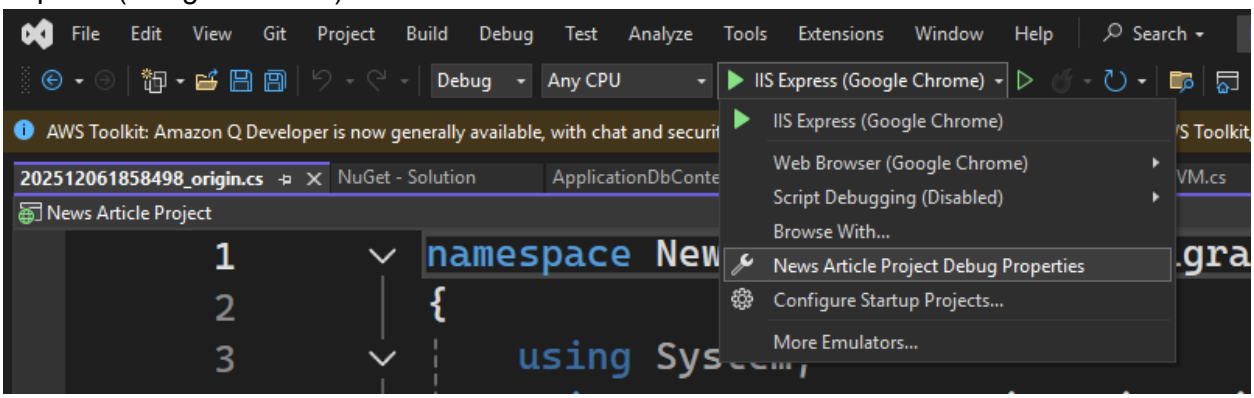


5. Run a migration to transfer the code-first approach of the database into the database located at the connection string
    a. To do this, open the Package Manager Console
    b. Type add-migration origin (or whatever name you want)

c. Type update-database

```
PM> add-migration origin
Scaffolding migration 'origin'.
The Designer Code for this migration file includes a snapshot of your current Code First model. This snapshot is
used to calculate the changes to your model when you scaffold the next migration. If you make additional changes to
your model that you want to include in this migration, then you can re-scaffold it by running 'Add-Migration origin'
again.
PM> update-database
Specify the '-Verbose' flag to view the SQL statements being applied to the target database.
Applying explicit migrations: [202512061858498_origin].
Applying explicit migration: 202512061858498_origin.
Running Seed method.
PM>
```

6. Ensure that News Article Project is the startup project and we are running it with IIS Express (Google Chrome)

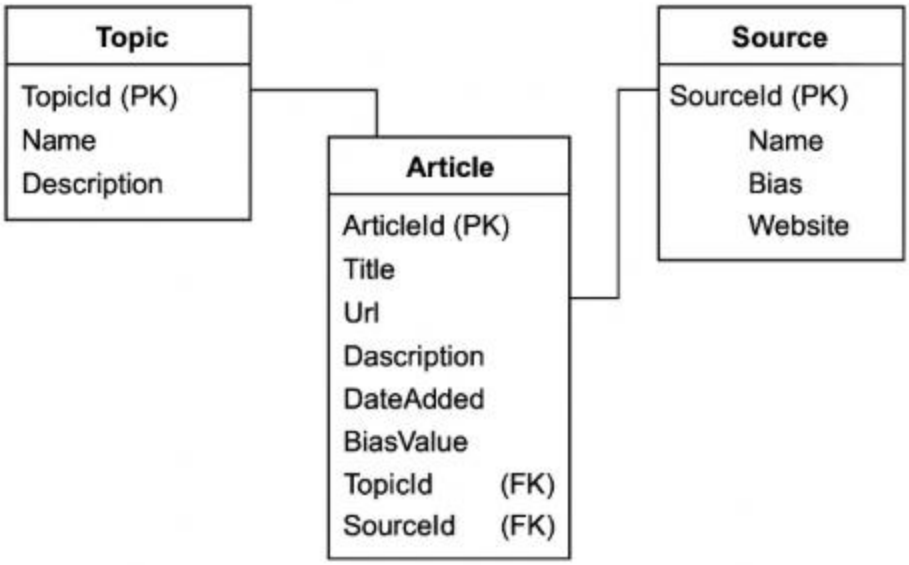


7. Change username to whatever user you would like to have Admin privileges to add and delete articles, topics, sources, and see table info (Path: Shared\LoginPartial):

```
<ul class="dropdown-menu dropdown-menu-end shadow-sm rounded-3 border-0" aria-labelledby="userDropdown">
    @if (User.Identity.GetUserName() == "afolabitest.o.soetan@gmail.com")
    {
        <li>@Html.ActionLink("Add Article", "Article", "Home", null, new { @class = "dropdown-item" })</li>
        <li>@Html.ActionLink("Add Source", "Source", "Home", null, new { @class = "dropdown-item" })</li>
        <li>@Html.ActionLink("Add Topic", "Topic", "Home", null, new { @class = "dropdown-item" })</li>
        <li>@Html.ActionLink("Delete Article", "DeleteArticle", "Home", null, new { @class = "dropdown-item" })</li>
        <li>@Html.ActionLink("Delete Source", "DeleteSource", "Home", null, new { @class = "dropdown-item" })</li>
        <li>@Html.ActionLink("Delete Topic", "DeleteTopic", "Home", null, new { @class = "dropdown-item" })</li>
        <li>@Html.ActionLink("See Table Info", "DatabaseSummary", "Home", null, new { @class = "dropdown-item" })</li>
        <li><hr class="dropdown-divider" /></li>
    }
```

8. To run the python program that allows you to insert Article, Topic, or Source CSV data into the database:
    a. Open the python scripts in the Insert_CSV_Data_To_News_Articles_Database folder in pycharm
    b. Create an environmental variable called TG_DB_CONN2 which stores the password to your database
    c. Run the program for either Article insertion, Topic insertion, or Source insertion

ER-diagram



Sample CRUD statements

Delete:

```
Topic deleted = dbContext.Topics.FirstOrDefault(x => x.TopicId == id);
DeleteTopicVM dVM = new DeleteTopicVM();
if(deleted == null)
{
    DeleteTopicVM d = new DeleteTopicVM();
    d.topics = dbContext.Topics.ToList();
    d.message = "No Topic Found";
    return View(d);

}

for(int i =0; i<deleted.Articles.Count(); i++)
{
    dbContext.Articles.Remove(deleted.Articles[i]);
}

dbContext.Topics.Remove(deleted);
```

Create:

```
public ActionResult Topic(TopicVM topic)
{
    if(topic.Description == null)
    {
        topic.message = "Topic description cannot be empty";
        return View(topic);
    }

    if(topic.Name == null)
    {
        topic.message = "Topic name cannot be empty";
        return View(topic);
    }

    ApplicationDbContext dbContext = new ApplicationDbContext();
    Topic newTopic = new Topic();
    newTopic.Description = topic.Description;
    newTopic.Name = topic.Name;
    dbContext.Topics.Add(newTopic);
```