
ITE Program

Course: Web Programming 1 – BWP401

# A Movie Website

Student: mai_133545

Class: C1

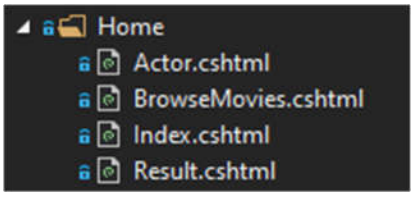
GitHub link: https://github.com/pudding2001/MayMovies

**Note**: I also uploaded the Database file with the project on github.

## ❖ Home Page:

The home page includes the following links:

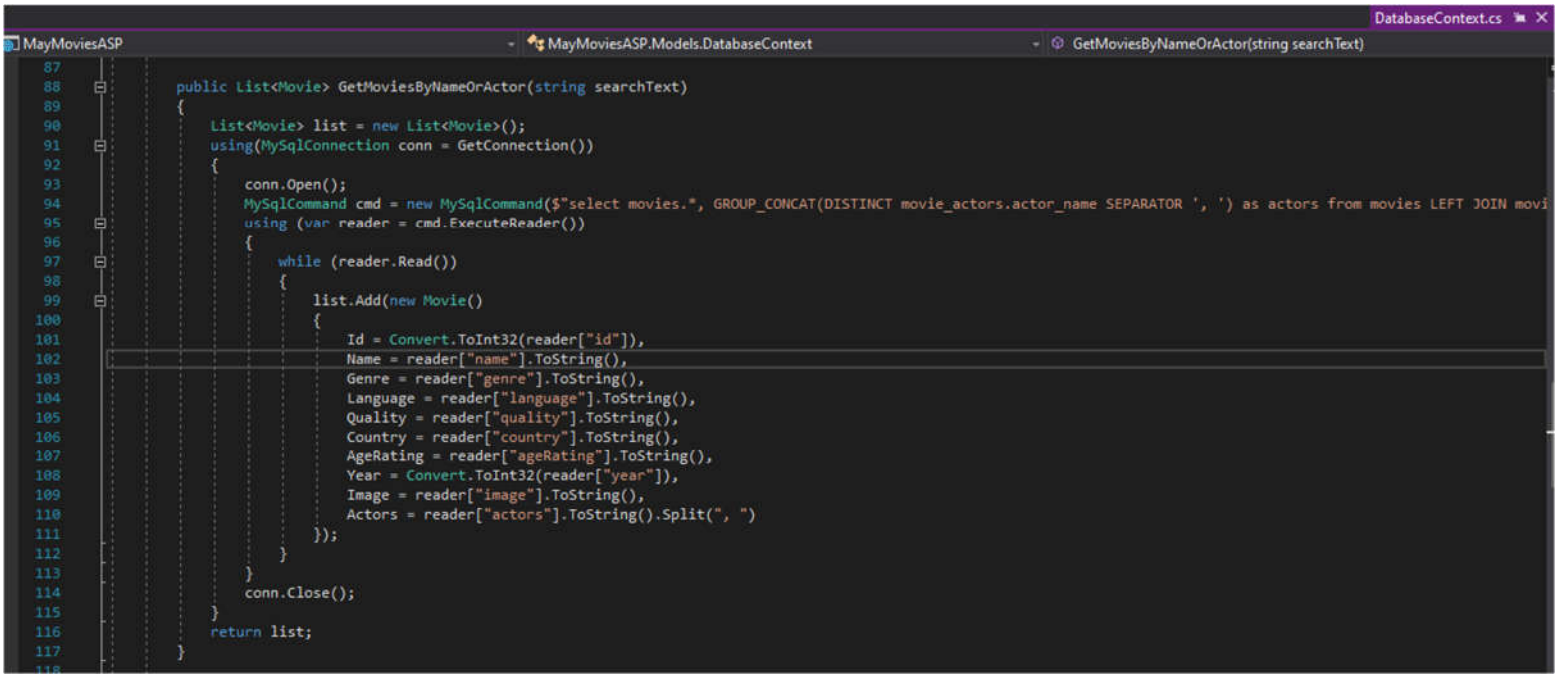
- The Browse Movies link.
- The Movie details link.
- The Actor/Actress info and movies.




It also includes the search bar at the top of the home pages. The user can search for a movie by typing a text pattern in the search box (the movie's name or the actor's /actress's name). The search result will appear in the BrowseMovies page.

```
// POST: HomeControler/SearchMovies
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult SearchMovies(string searchText)
{
    DatabaseContext context = HttpContext.RequestServices.GetService(typeof(DatabaseContext)) as DatabaseContext;
    Console.WriteLine(searchText);
    List<Movie> movies = context.GetMoviesByNameOrActor(searchText);
    return View("BrowseMovies", movies);
}
```

```
<form asp-action="SearchMovies" method="post" asp-controller="home">
    <div class=" input-group-container p-1 bg-light rounded rounded-pill shadow-sm">
        <div class="input-group">
            <input name="searchText" type="text" id="searchText" placeholder="Search for a movie or an actor/actress..." class="form-control border-0">
            <div class="input-group-append">
                <button id="button-addon1" type="submit" class="btn btn-link text-primary"><i class="fa fa-search"></i></button>
            </div>
        </div>
    </div>
</form>
```


DatabaseContext.cs
MayMoviesASP
MayMoviesASP.Models.DatabaseContext
GetMoviesByNameOrActor(string searchText)
public List<Movie> GetMoviesByNameOrActor(string searchText)
{
List<Movie> list = new List<Movie>();
using(MySqlConnection conn = GetConnection())
{
conn.Open();
MySqlCommand cmd = new MySqlCommand($"select movies.*, GROUP_CONCAT(DISTINCT movie_actors.actor_name SEPARATOR ', ') as actors from movies LEFT JOIN movi
using (var reader = cmd.ExecuteReader())
{
while (reader.Read())
{
list.Add(new Movie()
{
Id = Convert.ToInt32(reader["id"]),
Name = reader["name"].ToString(),
Genre = reader["genre"].ToString(),
Language = reader["language"].ToString(),
Quality = reader["quality"].ToString(),
Country = reader["country"].ToString(),
AgeRating = reader["ageRating"].ToString(),
Year = Convert.ToInt32(reader["year"]),
Image = reader["image"].ToString(),
Actors = reader["actors"].ToString().Split(", ")
});
}
}
conn.Close();
}
return list;
}

## Browse Movies:

Here the user can search for a movie by typing a text pattern in the search box, as well as clicking on the dropdown menu to narrow the search of the required info of the movie.

Unfortunately, I could not complete this kind of search, I did make the design but did not apply the required functionality to it…

## Movie Details:

Here we see the movie details displayed - after searching or clicking on one of the movies displayed in the website.

As supposed, all the details and information are grabbed from the data base at the back-end. Data is structured in a data base using “mySQL” data base server.

I could not complete the work here and so I could not manage to do the previous functionality.

However, the design could be previewed.

## Actor/Actress info and movies:

Here the actor information (Name and Image) is displayed in addition to the number of movies he/she acted in.

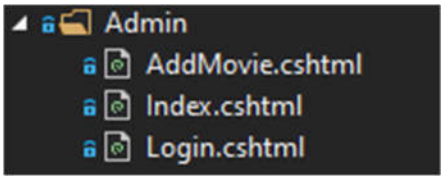
A number of movies related to the actor is displayed in the page down below.

Again, I could not complete the work here and so I could not manage to show the required information of the actor/actress.

However, the design could be previewed.

## ❖ Admin Panel:

1. Admin log-in page.
2. Admin panel page.
3. Admin “Add movie” page.



## Admin log-in page:

Here the admin enters his username and password to be verified at the back-end.

The design is done but the functionality is not available.

## Admin panel page:

After a successful validation of the log-in credentials, the admin panel page is sent to the browser.

Three choices are available: Add movie, Edit movie and Enable/Disable movie. However, only the Add movie choice is working.

## Movie details managing page:

After clicking on the “Add movie” button, the Add Movie page is sent to the browser. The movie details are displayed in a list where the admin can add and choose the details.

After finishing the adding process, the admin can click on the “Done” button.

```
public void AddMovie(Movie movie)
{
    using (MySqlConnection conn = GetConnection())
    {
        conn.Open();
        MySqlCommand cmd = new MySqlCommand($"INSERT INTO `movies`(`name`, `genre`, `language`, `quality`, `year`, `country`, `ageRating`, `image`) VALUES ('{movie.
        cmd.ExecuteNonQuery();
        int id = Convert.ToInt32(cmd.LastInsertedId);
        movie.Actors = movie.Actors[0].Split(",");
        string actors = "";
        for(int i=0; i< movie.Actors.Length; i++)
        {
            if(i < movie.Actors.Length -1 )
                actors += $"({id},'{movie.Actors[i]}'),";
            else
                actors += $"({id},'{movie.Actors[i]}')";
        }
        cmd = new MySqlCommand($"INSERT INTO `movie_actors`(`movie_id`, `actor_name`) VALUES {actors}", conn);
        cmd.ExecuteNonQuery();
        conn.Close();
    }
}
```

```
// POST: AdminController/Create
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult CreateMovie(Movie movie)
{
        DatabaseContext context = HttpContext.RequestServices.GetService(typeof(DatabaseContext)) as DatabaseContext;
    if (movie.ImageFile != null)
    {
        var uniqueFileName = GetUniqueFileName(movie.ImageFile.FileName);
        var uploads = Path.Combine(hostingEnvironment.WebRootPath, "images");
        var filePath = Path.Combine(uploads, uniqueFileName);
        movie.ImageFile.CopyTo(new FileStream(filePath, FileMode.Create));
        movie.Image = uniqueFileName;
        //to do : Save uniqueFileName  to your db table
    }
    // to do  : Return something
        context.AddMovie(movie);
        return RedirectToAction(nameof(Index));
}

private string GetUniqueFileName(string fileName)
{
    fileName = Path.GetFileName(fileName);
    return Path.GetFileNameWithoutExtension(fileName)
              + "_"
              + Guid.NewGuid().ToString().Substring(0, 4)
              + Path.GetExtension(fileName);
}
```