MINI PROJECT

MOVIE SEARCH USING OMDB API

Developed By

Name: R.Yazhinni

Reg.no: R201174601398

Name of the Coordinator: Indhumathi

Date of Submission: 24.08.2020

ABSTRACT

The advancements in Web Technology have made it possible to create visually appealing websites and the development of API has made the interaction with the website faster than ever before. In this project, a website is created using HTML, CSS and JavaScript for the frontend which is connected to the movie database using the OMBD API that helps in request and retrieval of queries.

CONFIGURATION

Software required:

Software requirements of the project are mentioned below:

- Platform: Windows10
- Eclipse IDE

Frontend: HTML, CSS, JavaScript.

Hardware Required:

Hardware Requirements of the project are mentioned below:

- Processor: Ryzen, 1.8GHz
- RAM: 8gb

CONTENT

1. SYNOPSIS
2. INTRODUCTION
3. DEVELOPMENT STEPS
4. CREATE AN ECLIPSE DYNAMIC WEB PROJECT
5. PROJECT STRUCTURE
6. CREATE a HTML-index.html
7. CREATE a CSS-style.css
8. CREATE a JavaScript-index.js
9. DEMO

1. SYNOPSIS

Online Movie Search using the OMDB API project.

2. INTRODUCTION

Every year over tens of thousands movies are released around the world in various languages in a variety of genres. There are plenty of websites which give reviews related to the movies released that also have a short storyline describing about the plot and the cast crew who have played major roles in the movie. These websites are used by people to determine which movie to watch respective of their interest and preference. The advancements in Web Technology have made it possible to create visually appealing websites and the development of API has made the interaction with the website faster than ever before. In this project, a website is created using HTML, CSS and JavaScript for the frontend which is connected to the movie database using the OMBD API that helps in request and retrieval of queries.

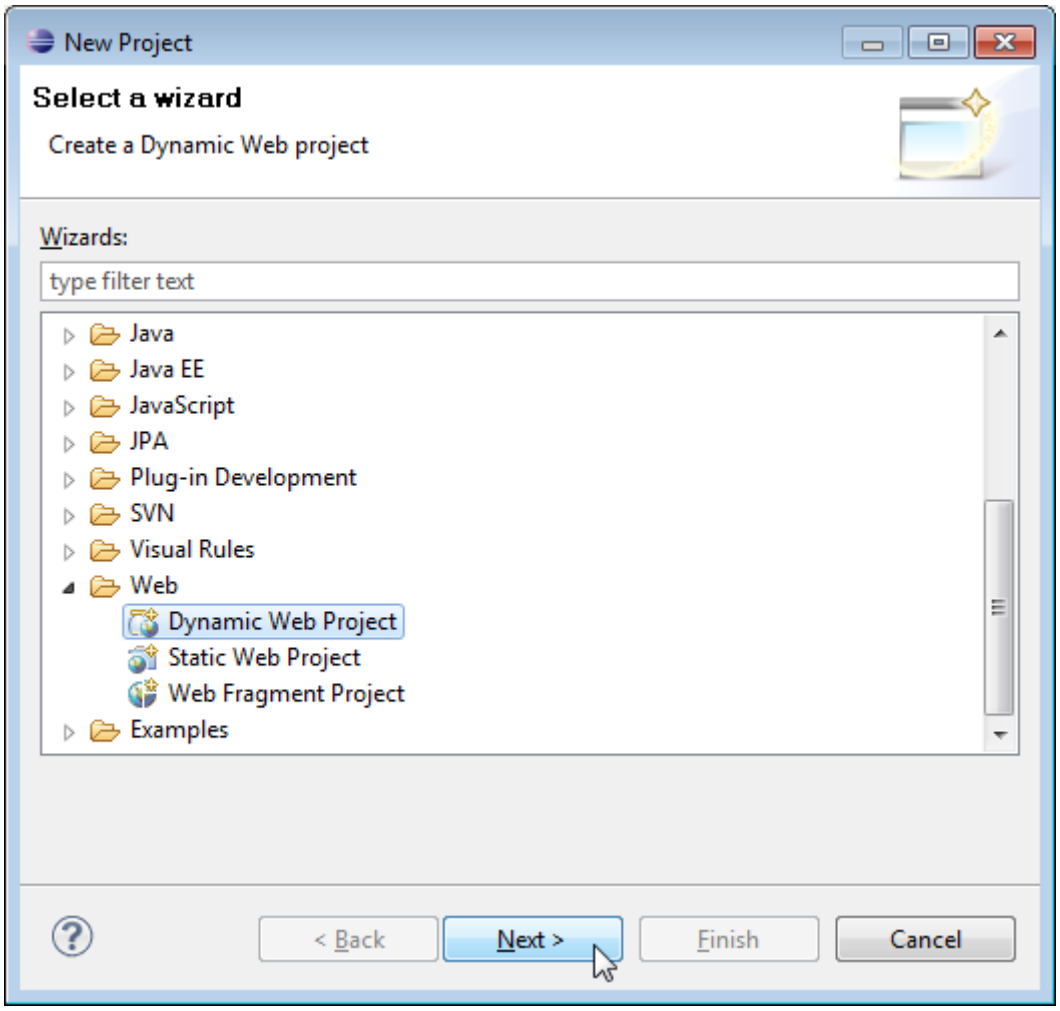
3. DEVELOPMENT STEPS

1. Create an Eclipse Dynamic Web Project
2. Project Structure
3. CREATE a HTML-index.html
4. CREATE a CSS-style.css
5. CREATE a JavaScript-index.js
6. Demo

4. CREATE AN ECLIPSE DYNAMIC WEB PROJECT

To create a new dynamic Web project in Eclipse:

1. On the main menu select File > New > Project.
2. In the upcoming wizard choose Web.



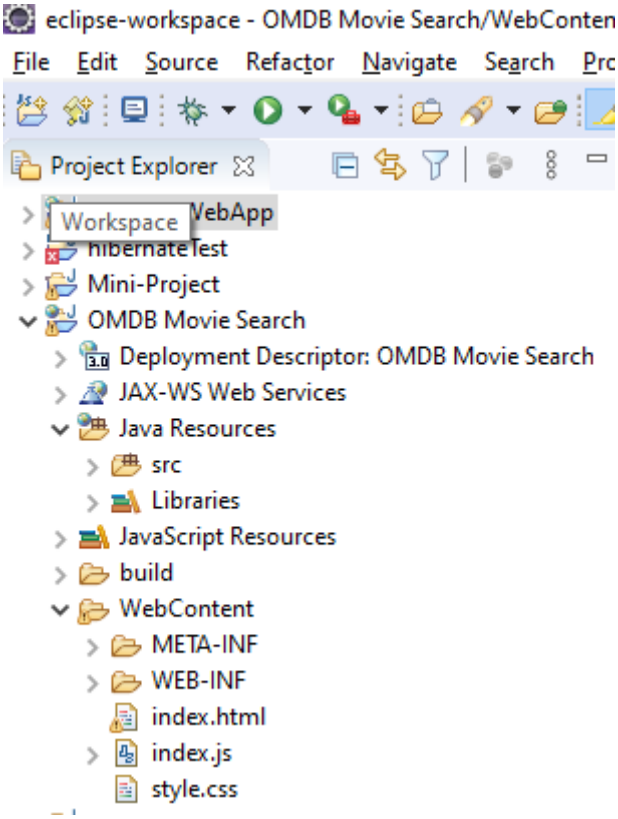
3. Select **Dynamic Web Project** and Click on **Next**.

4. Enter project name as "OMDB Movie Search";

5. Click Finish.

5. PROJECT STRUCTURE

Standard project structure for your reference:



6. CREATE a HTML file-index.html

Let's create an index.html in the java resource=>src =>package, Let's create a HTML file which contains HTML code. Add the following code to an index.html file:

<!DOCTYPE html>

<html lang="en">

<head>

```
<meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>OMDB Movie Search</title>
 <link rel="stylesheet" href="style.css">
 <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.4/angular.min.js"></scr
ipt>
 <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></sc
ript>
</head>
<body ng-controller='MovieAppCtrl'>
<div class="site"></div>
<div class="container">
<h1>OMDB movie records</h1>
<div class="row">
<div class="form-group">
<div class="input-group mb-3">
<input type="text" class="form-control" placeholder="Search a movie" aria-
label="Recipient's username" aria-describedby="button-addon2" ng-
model="Search" ng-keyup="GetMoviesData()">
<div class="input-group-append">
<button class="btn btn-primary" type="button" id="button-
addon2">Search</button>
```

```
</div>

</div>

</div>

</div>

</div>

<div class="container" style="margin-top:10px;">

<div class="lightboxgallery-gallery clearfix">

<a ng-repeat="movie in MovieData.Search | limitTo:5" class="lightboxgallery-gallery-item" target="_blank" ng-href="{{movie.Poster}}"
data-title="{{movie.Title}}"
data-link="http://www.omdb.com/title/{{movie.imdbID}}"
data-alt="{{movie.Title}}">

<div class="col-md-4">

<img ng-src="{{movie.Poster}}" title="{{movie.Title}}"
alt="{{movie.Title}}">

<div class="lightboxgallery-gallery-item-content">

<span class="lightboxgallery-gallery-item-title">{{movie.Title}}</span>

</div>

</div>

</a>

</div>

</div>

<script src="index.js"></script>

<script src="http://cdn.asimplify.com/lightboxgallery/js/lightboxjs-custom.js"></script>

</body>
```

7. CREATE CSS file using style.css.

CSS stands for Cascading Style Sheets

It is a language that describes the style of an HTML document.

SYNTAX:

h1 {color: red; font-size: 14px;}

| |

Selector Declaration

style.css

```
body{
    margin:0;
    top:0;
}

.site {
   height: 100%;
    width: 100%;
    position: absolute;
    background-image: url('C:\Users\Yazhinni\Downloads\bac.jpg');
    background-repeat: repeat;
    opacity: 0.3;
    background-attachment: inherit;
}
```

```
.container {
    margin: 0 auto;
    background-color: rgb(0, 0, 0);
    text-align: center;
    color: orange;
    font-size: x-large;
    z-index: 10;
}

.row{
    border-top: 1px solid tomato;
    padding: 20px;
    display: flex;
    justify-content: center;
    align-self: center;
}
.form-group .input-group .form-control {
    position: relative;
    z-index: 2;
    margin-bottom: 0;
    width: 250%;
    border-radius: 4px;
}

.form-group .input-group .input-group-append #button-addon2 {
    position: absolute;
}
```

8. CREATE index.js file using javascript

This file can be found at this path WebContent/WEB-INF/index.js

index.js

```
jQuery(function($) { $(document).on('click', '.lightboxgallery-gallery-item',
function(event) {
    event.preventDefault();
    $(this).lightboxgallery({
      showCounter: true,
      showTitle: true,
      showDescription: true
    });
  });
});

angular.module('MovieApp', [])
.controller("MovieAppCtrl", ['$http', '$scope', function($http, $scope) {
                $scope.MovieData = [];
                $scope.Search = null;
                $scope.GetMoviesData = function() {

try {
 $http({ url: 'http://www.omdbapi.com/?apikey=3e8d9048&s='+$scope.Search,
        method: "GET",
        }).then(function(payload) {  $scope.MovieData = payload.data; },
        function(){  alert("Something is wrong. Please try again."); });
}
```

```
catch (error) {  alert("Exception occured while fetching movie data.");  }
        }
    }
  ]);
angular.element(function() {
  angular.bootstrap(document, ['MovieApp']);
});
```

9. DEMO

It's time to see a demo of the above development.

Movie Search using OMDB API

Once you deploy this application successfully then hit this link into a browser.

C:\Users\Yazhinni\eclipse-workspace\OMDB Movie Search\WebContent\index.html

After entering the URL into the Web Browser you will get to see a Webpage of this format: