**SW2 – Project Evaluation Form**

* **Each team must submit the following Documentation that contains:**
  + **Project Description in detail.**
  + **Class Diagram. And Database Schema.**
* **Each team must submit the project via GitHub:**
* **Source Code.**
* **Video Demo for running ( 2 – 5 Minutes ).**
* **Documentation and Evaluation Form.**
* **The Evaluation will start with giving all teams 30 marks then check the following criteria:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Violation Level** | **Full** | **Medium** | **Small** | **Grade** |
| **Documentation** | **-5** | **-2** | **-1** |  |
| **Not Apply MVC (it does not Separate Business logic from GUI ).**  **Example of violation: write the implantation for a method such as an inset item into the**  **database inside the Button Action method)** | **-6** | **- 3** | **-1** |  |
| **Violate clean code – Variables** | **-2** | **-1** | **-.05** |  |
| **Violate clean code – Functions** | **-2** | **-1** | **-.05** |  |
| **Violate Single-responsibility Principle** | **-2** | **-1** | **-.05** |  |
| **Violate Open-closed Principle** | **-2** | **-1** | **-.05** |  |
| **Violate the Liskov Substitution Principle** | **-2** | **-1** | **-.05** |  |
| **Violate Interface Segregation Principle** | **-2** | **-1** | **-.05** |  |
| **Violate Dependency Inversion Principle** | **-2** | **-1** | **-.05** |  |
| **Not Upload code to GitHub** | **-1** | | |  |
| **Only One Branch Without Merge (GitHub)** | **-2** | | |  |
| **Only One Contribution (GitHub)** | **-2** | | |  |
| **Total Minus from Grade** |  | | |  |

|  |  |  |
| --- | --- | --- |
| **Design Pattern Bounce** | **+4** |  |
| **Bounce on Overall Work** | **+2** |  |
| **Total Team Grade / 30** |  | |

**Full violated –**They violate the principle in all cases in the Code.

**Medium Violated –** They apply the principle in some cases and violate it in some cases (in total the student applies the principle on average).

**Small violated –** They Apply the principle in most cases except for a very few cases, no more than one or two.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name (Arabic)** | **ID** | **Individual Bounce +2** | **Grade** |
| عمر اiس عiد الحكiم اسماعiل | 201900502 |  |  |
| عمر عادل عiد الحلiم محمد | 201900512 |  |  |
| احمد السiد محمود السiد | 201900019 |  |  |
| سما جمال حمدي قطب | 201900352 |  |  |
| أصالة سعiد ابوالخ yᢕ عطiﻪ | 201900157 |  |  |
| شمس سiد محمود محمد | 201900372 |  |  |
|  |  |  |  |

**Project Description:**

User view:

1. -User can create a new account.
2. -User login to his/her account and edit his/her profile (profile image, first name, last name, favorite movies, favorite actors, favorite directors), and can also add his favorite movies, actors, and directors from a dropdown list that filled from the data added by the admin.
3. -User can like or dislike movies.
4. -User can open movies’ profile and write comments.

Admin view:

1. -Admin can create , update and delete movies’ profiles, which include (movie name, movie images, actors exist in the movie, the director of the movie), and if we clicked on the actors and the directors that exist in the movie then it will navigate to their profiles
2. -Admin can create , update and delete actors’ profiles, which include (first name, last name, age, movies that he participate in), and if we clicked on the movies that the actor participated in, then it will navigate to the movies profiles
3. Admin can create , update and delete directors’ profiles, which include (first name, last name, age, movies that he produced), and if we clicked on the movies that the director produced, then it will navigate to the movies profiles
4. Author can search for movies, actors, and directors by name, and open the profiles of the selected movies, actors, and directors to check their latest work



