**JS Applications Exam – Movies SPA**

You are assigned to implement a **Single** **Web Application** (SPA) using HTML5, JavaScript, AJAX, REST and JSON with cloud-based backend (Firebase, Backendless or other). Using libraries like **Handlebars** and **Sammy** is allowed but is not obligatory. The app keeps **users** and **movies**. **Guests** should be able to **register** and **login**. Logged-in users should be able to view **all movies**, **add** **movie**, **like movie**, see **details** about a **movie** and **logout**. Logged-in users should also be able to **edit** or **delete** the movies **they have added**.

## Create a REST Service

Use any cloud-based databaseand create application to keep your data in the cloud.

Create a collection called **movies.** Each **movie** has a **name**, **description**, **imageUrl**, **creator** and **people liked it**.

## Create Movies Application

**HTML and CSS**

You have been given the web design of the application as **HTML** + **CSS** files.

* Initially all views and forms are shown by the HTML. Your application may **hide**/**show elements** by CSS (display: none) or **delete**/**reattach** from and to the DOM all unneeded elements, or just display the views it needs to display.
* You may render the views/forms/components with **JavaScript** or **Handlebars**.

**Important**: Don’t change the elements’ **class names** and **ids**. Don’t rename form fields/link/ids. You are **allowed** to add **data attributes** to any elements. You may modify href **attributes** of links and add action**/**method **attributes** to **forms**, to allow the use of a routing library.

**Client-Side Web Application**

**Design** and **implement** a client-side front-end app (SPA) for managing **movies**. Implement the functionality described below.

**Navigation Bar (5 Pts)**

Navigation links should correctly change the current page (view).

* Clicking on the links in the **NavBar** should display the view behind the navigation link.
* Your application may **hide**/**show elements** by CSS (display: none) or **delete**/**reattach** from and to the DOM all unneeded elements, or just display the views it needs to display.
* **The Logged-in** user navbar should contain the following elements:[**Movies**] a **link** to the Home page, the user caption ("Welcome, {email}"), [Logout].Each link navigates to the named page.

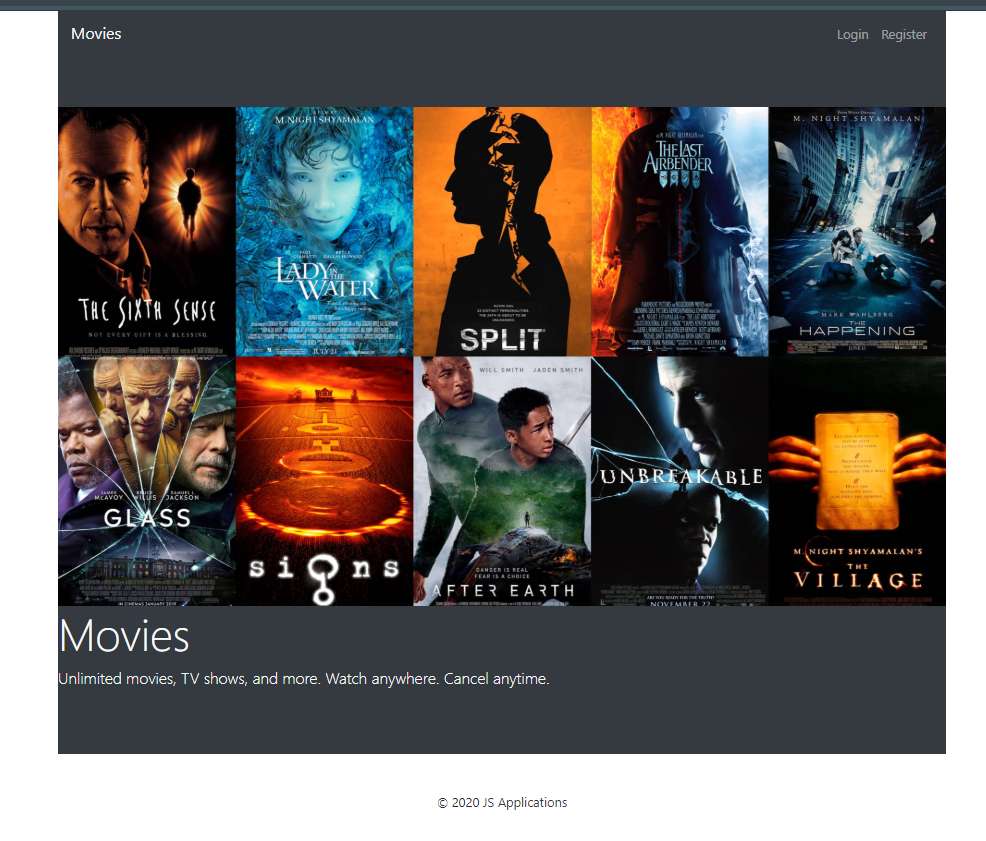


* The guest users navbar should contain the following elements: : [**Movies**] which is a **link** to the Home page and [Login], [**Register**].



**Home Page (Guest) (5 Pts)**

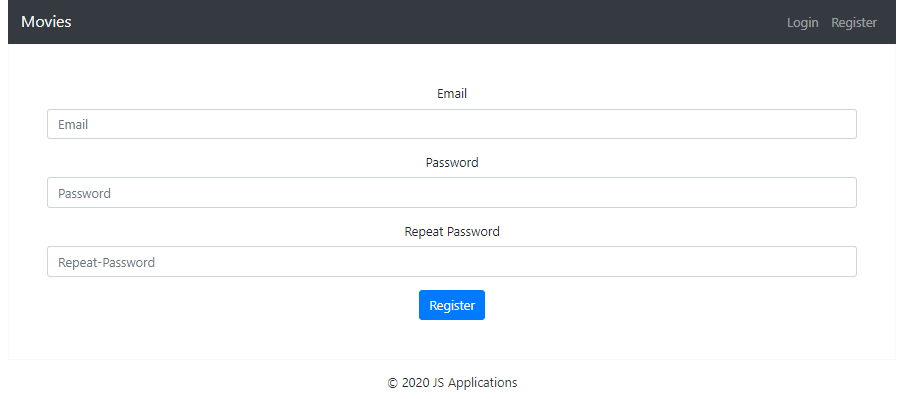
The initial page (view) should display the **guest** **navigation bar** + **Guest Home Page** + Footer.

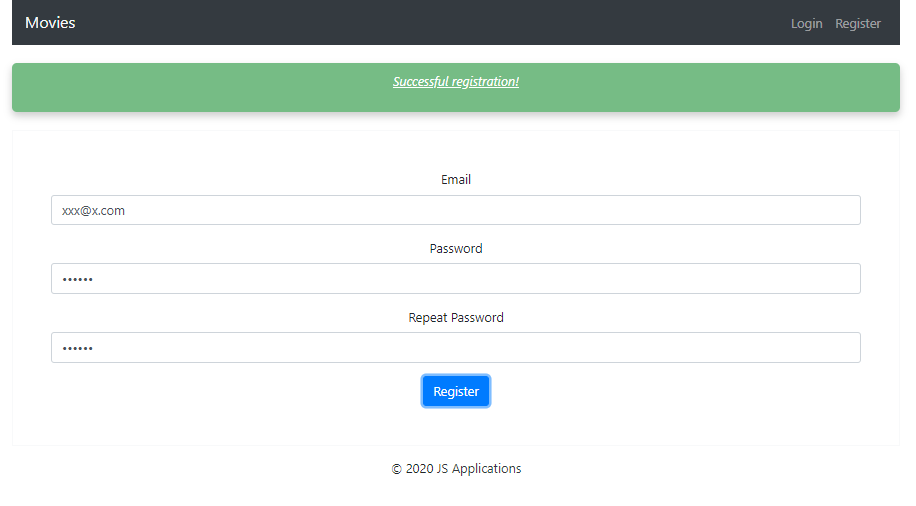


**Register User (5 Pts)**

By given **email** and **password,** the app should register a new user in the system.

* The following validations should be made:
  + The **email** input must be **filled**
  + The **password** should be **at least 6 characters** long
  + The **repeat password** should be **equal to the password**
* After a **successful registration**, a notification message **"Successful registration!"** should be displayed and the app should **redirect** to the **home page with the right navbar**.
* In case of **error** (eg. invalid username/password), an appropriate error **message** should be displayed, and the user should be able to **try** to register again.
* Keep the user data in the browser's **session or local storage**.
* After a **successful registration** redirect to **Home page.**

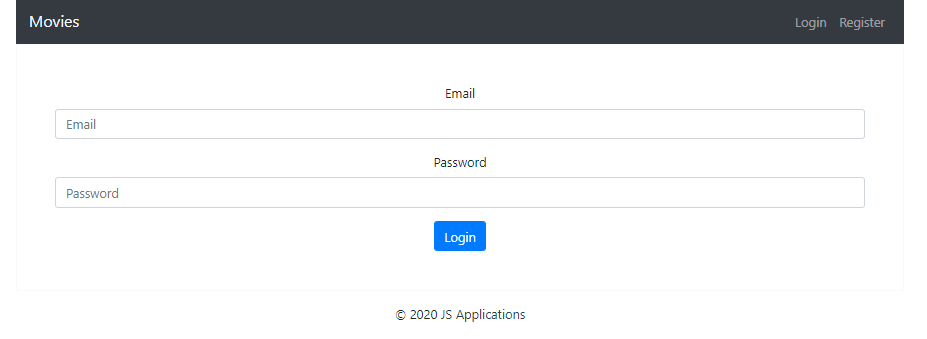


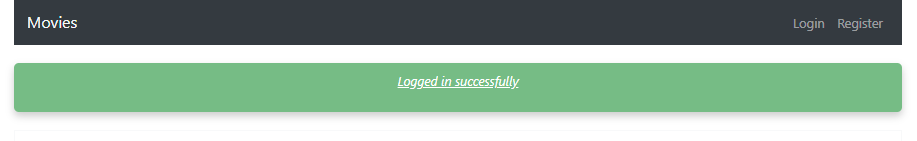


**Login User (5 Pts)**

By given **username** and **password,** the app should login an existing user.

* After a **successful login**, a notification message **"Login successful."** should be shown and the user **home page should be displayed**.
* In case of **error**, an appropriate error **message** should be displayed and the user should be able to fill in the login form again.
* Keep the user data in the browser's **session or locale storage**.
* After a **successful login** redirect to **Home page.**

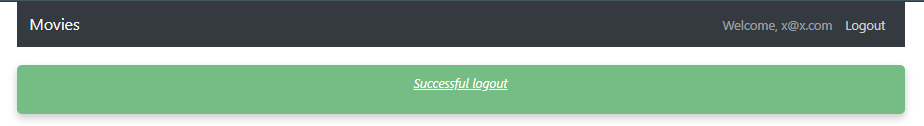




**Logout (5 Pts)**

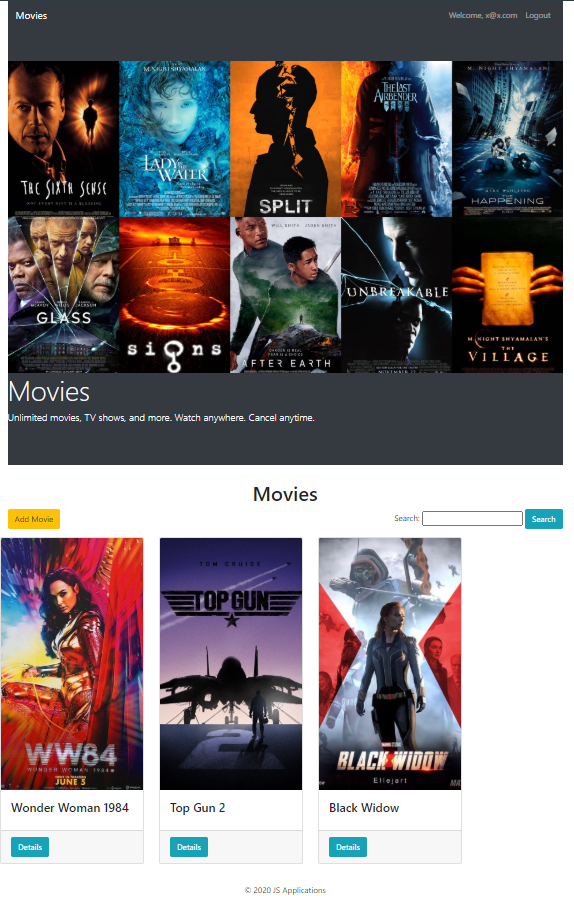
Successfully logged in users should be able to **logout** from the app.

* After a **successful** logout, a **notification** message **"Successful logout"** should be displayed and the **anonymous screen** should be shown
* The **"logout" REST service** at the back-end **must** be called at logout
* All local information in the browser (**user session data**) about the current user should be deleted
* After a **successful logout** redirect to **Login page.**

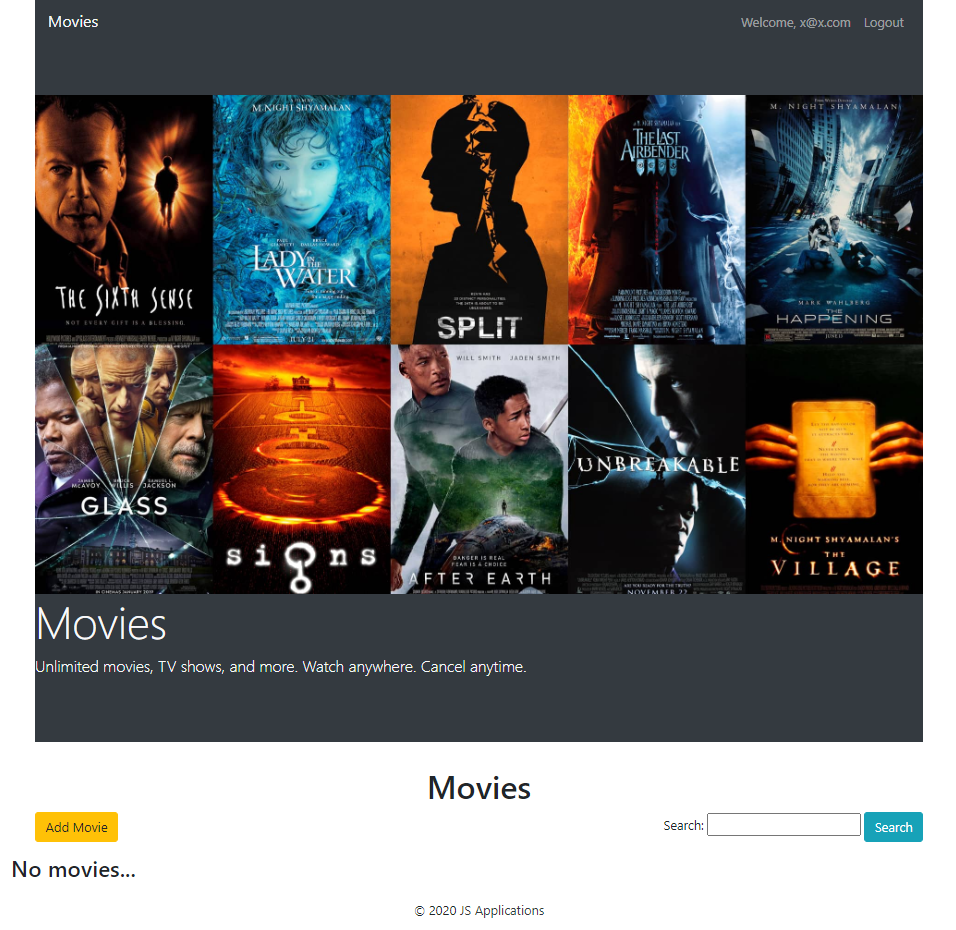


**Home Page(Logged in User) (30 Pts)**

Successfully logged-in users should be welcomed by the Homepage. They should be able to see all added movies:



If there are **NO** such, the following view should be displayed:



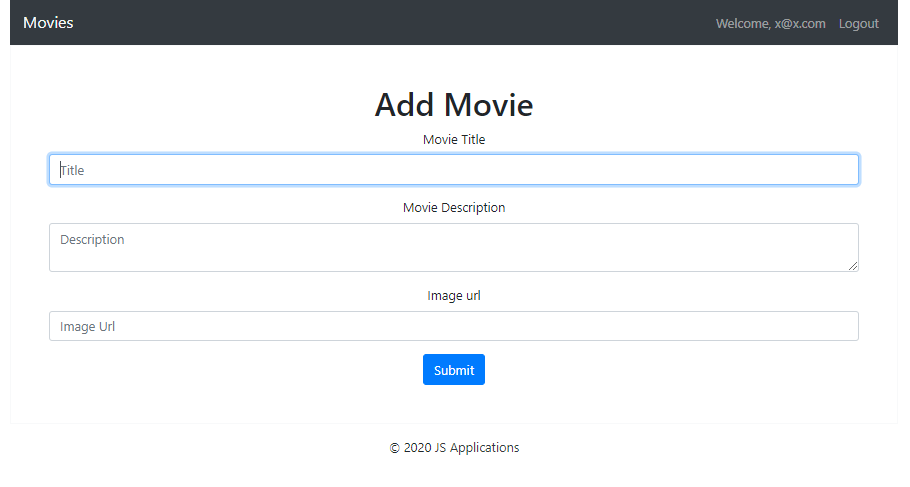
[**Add Movie**] **button** should refer to the **add** **movie** **form**

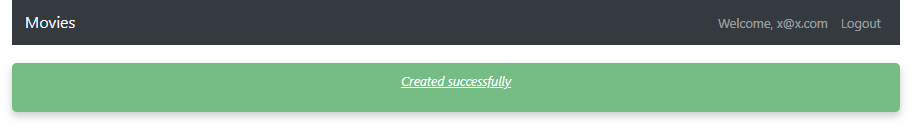
**Add Movie(10 Pts)**

Logged-in users should be able to **add movie**.

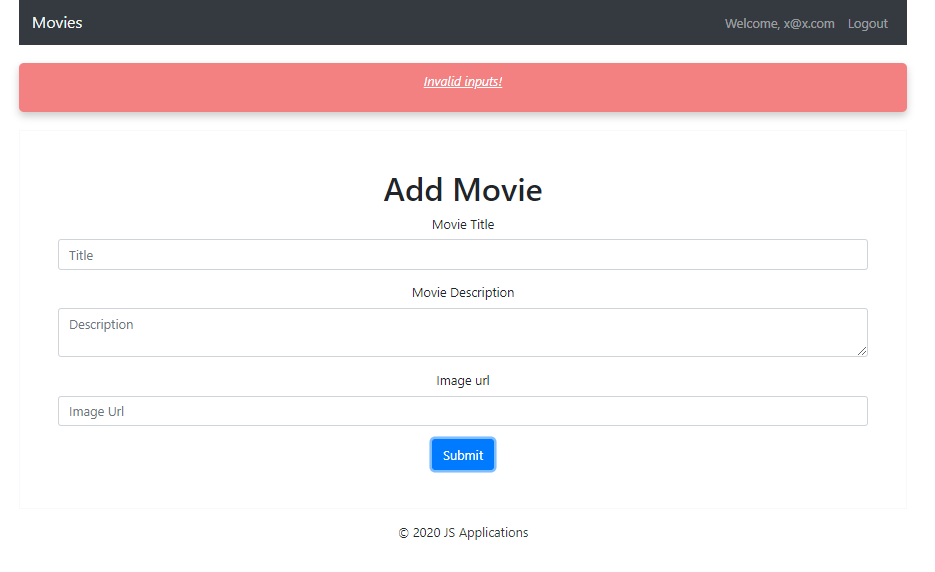
Clicking the [Add Movie] **button** in the **Home page** should **display** the Add Movie page.

* The form should contain the following validations:
  + The **title, description and image** shouldn’t be **empty**.
  + **By** **default**, every newly created movie must have additional information:
    - **Creator:** string representing the current movie creator;
    - **People liked:** keeping data of users liked the movie;
  + After a **successful** movie adding, a notification message **"Created successfully!"** should be displayed and the Home pageshould be shown.
* The newly added movie should be stored in the database collection "movies".





Trying to submit empty feelds shood displаy error:

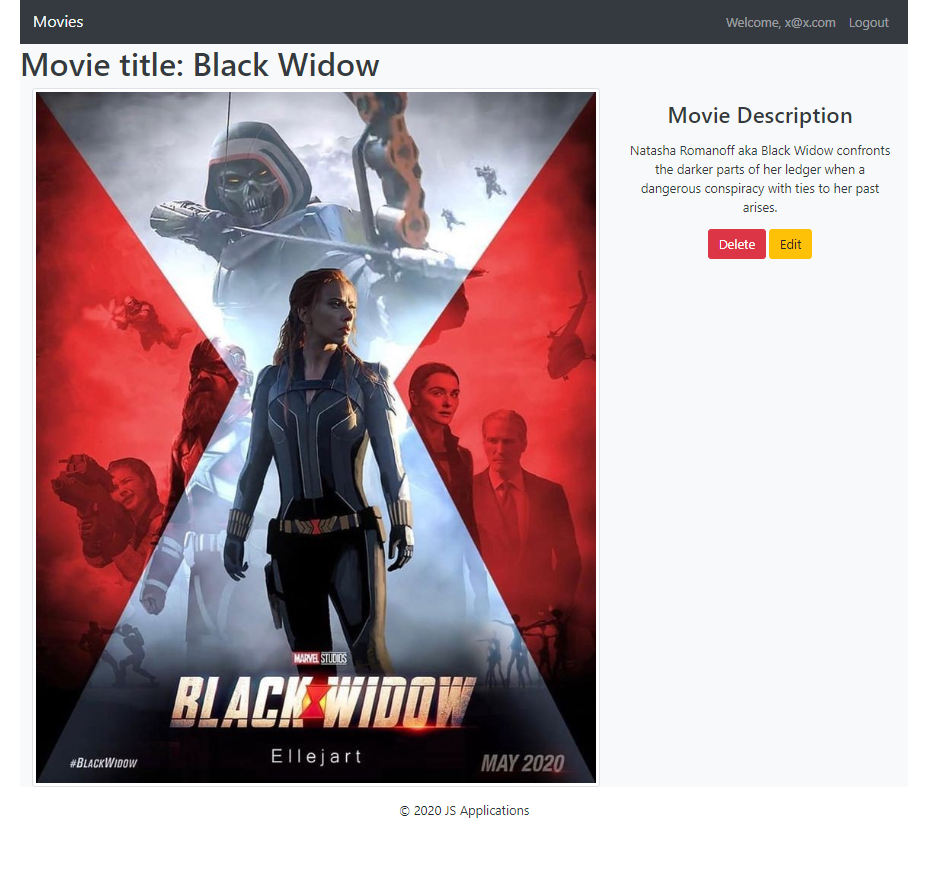


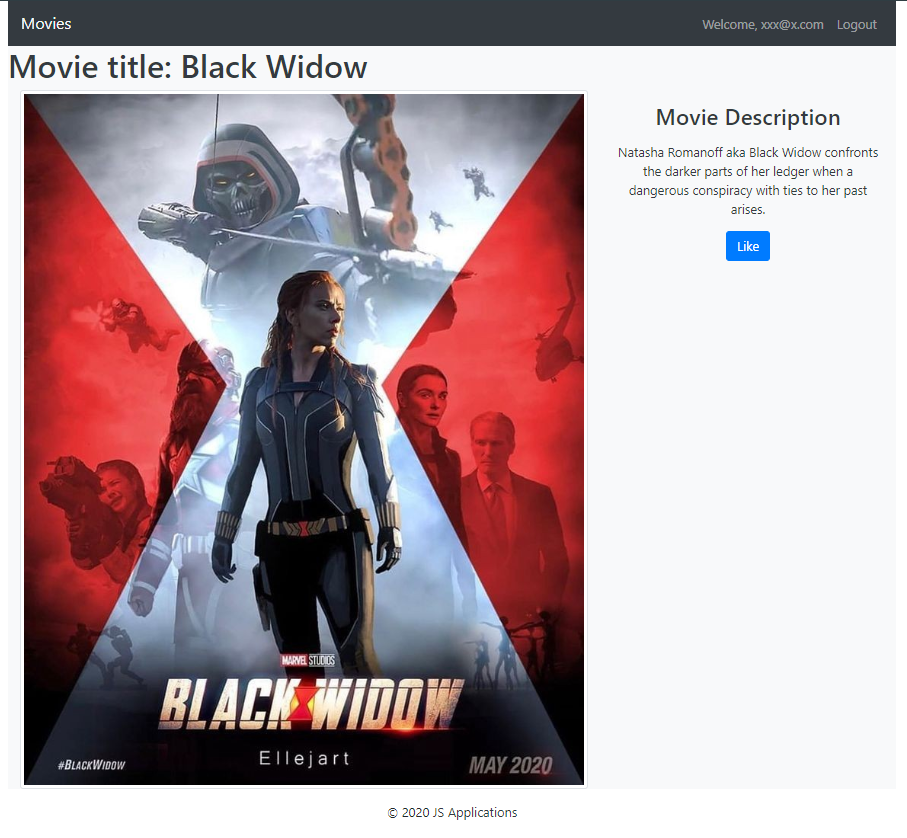
**Details (15 Pts)**

Logged-in users should be able to **view details** aboutmovies.

Clicking the [Details] **button** of a **particular movie** should **display** the Movie Details page.

* If the currently logged-in user is the creator of the movie, the [Delete] and [Edit] **buttons** should be set to **visible**, otherwise there should be only 1 button [Like].





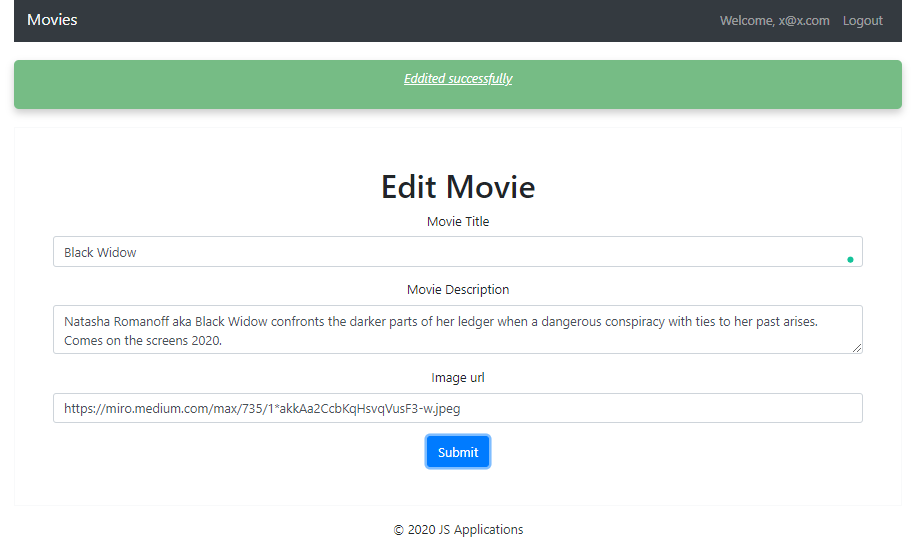
**Edit Movie (10 Pts)**

Logged-in users should be able to **edit** movies, added by them.

Clicking the [Edit] **button** of a **particular movie** on the Movie Details page should **display** the Edit Movie page inserting the additional information of the movie in the input feelds:



* After a **successful edit**, a notification message "Eddited successfully" should be **displayed**, and the user should be redirected to the current movie Details page.



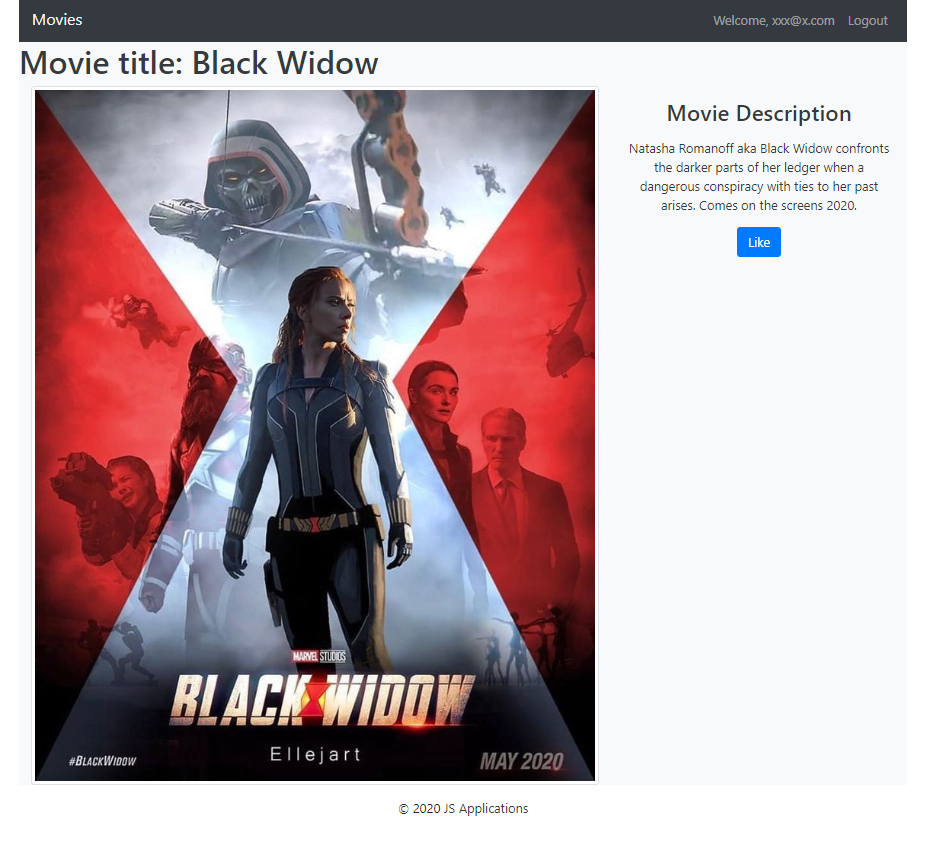
**Like Movie (10 Pts)**

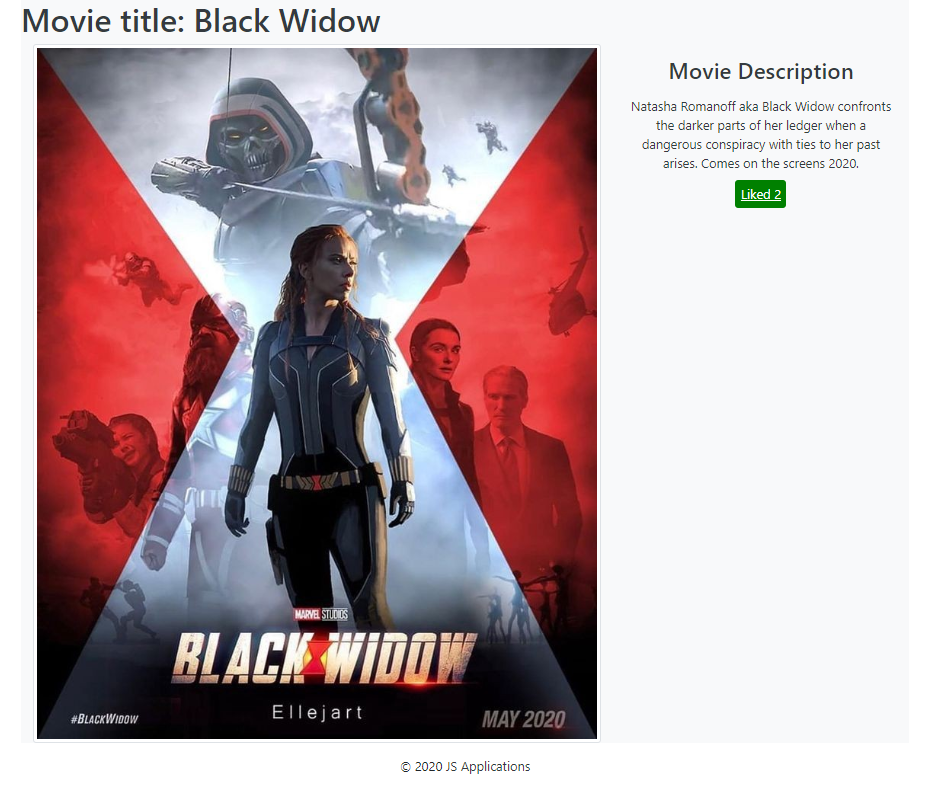
Logged-in users should be able to **like** movie, added by **other** **user**.

**NOTE**: A user should **NOT** be able to like an **movie**, created by **himself**.

Clicking the [Like] **button** of an **movie** (on the Movie Details page) should **add the current user email to the property People liked.** After **successfully like**:

* [Like] **button** changes to [Liked {number of likes}] **span so** users can‘t **like a movie** multiple times**.**
* Notification message **"Liked successfully"** should be displayed.



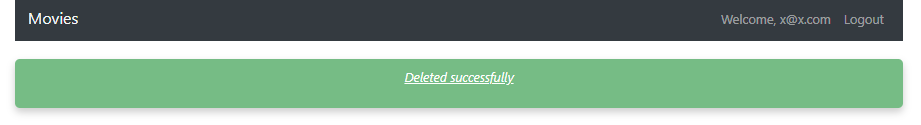


**Delete Movie (5 Pts)**

Logged-in users should be able to **delete their** movies.

Clicking the [Delete] **button** of an **movie** (on the Movie Details page) should **delete** the **movie**.

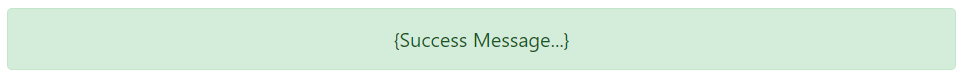
* After **successful** **movie delete** a notification message **"Deleted successfully"** should be displayed and the Home pageshould be **shown**



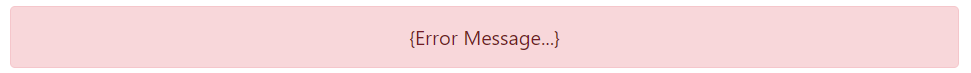
**(BONUS) Notifications (5 Pts)**

The application should notify the users about the result of their actions.

* In case of a **successful** action, a **notification message** **(green)** should be shown, which disappears automatically after 1 second



* In case of **error**, an **error notification message (red)** should be shown, which disappears automatically after 1 second.



* **NOTE**: You get all the points if **all** the notifications have the **exact content** as described in each section above.

**(BONUS) Search: (5 Pts)**

**Registered** users, should be able to search in listed movies by entered string into the given search input**.**