**JS Applications Workchop – Movies SPA**

You are assigned to implement a **Web application** (SPA) using HTML5, JavaScript, AJAX, REST and JSON with cloud-based backend. 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**, **create** **movies**, **buy tickets** for the movies, see **details** about a **movie** and **logout**. Logged-in users should also be able to **edit** or **delete** the movies **they have created**. There should also be a **section** where users can **see only the movies they have created**.

**Movies – 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 **Handlebars**.

**Movies – Client-Side Web Application**

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

### Notifications

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

* In case of successful action an **informational (green) notification message** should be shown, which disappears automatically after 3 seconds or manually when the user clicks it.



* In case of **error**, an **error notification message** (red) should be shown which disappears on user click.



* During the AJAX calls a **loading notification message (blue)** should be shown. It should disappear automatically as soon as the AJAX call is completed.



### Navigation Bar

Navigation links should correctly change the current page (view). **The navbar should be displayed in every view!**

* Clicking on the links in the **NavBar** should display the view behind the link (views are represented as sections in the HTML code).
* 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 links [Cinema], [Add Movie], [My Movies], [Logout], and the user caption   
  "Welcome, {username}!" should be visible **only** for logged-in users.



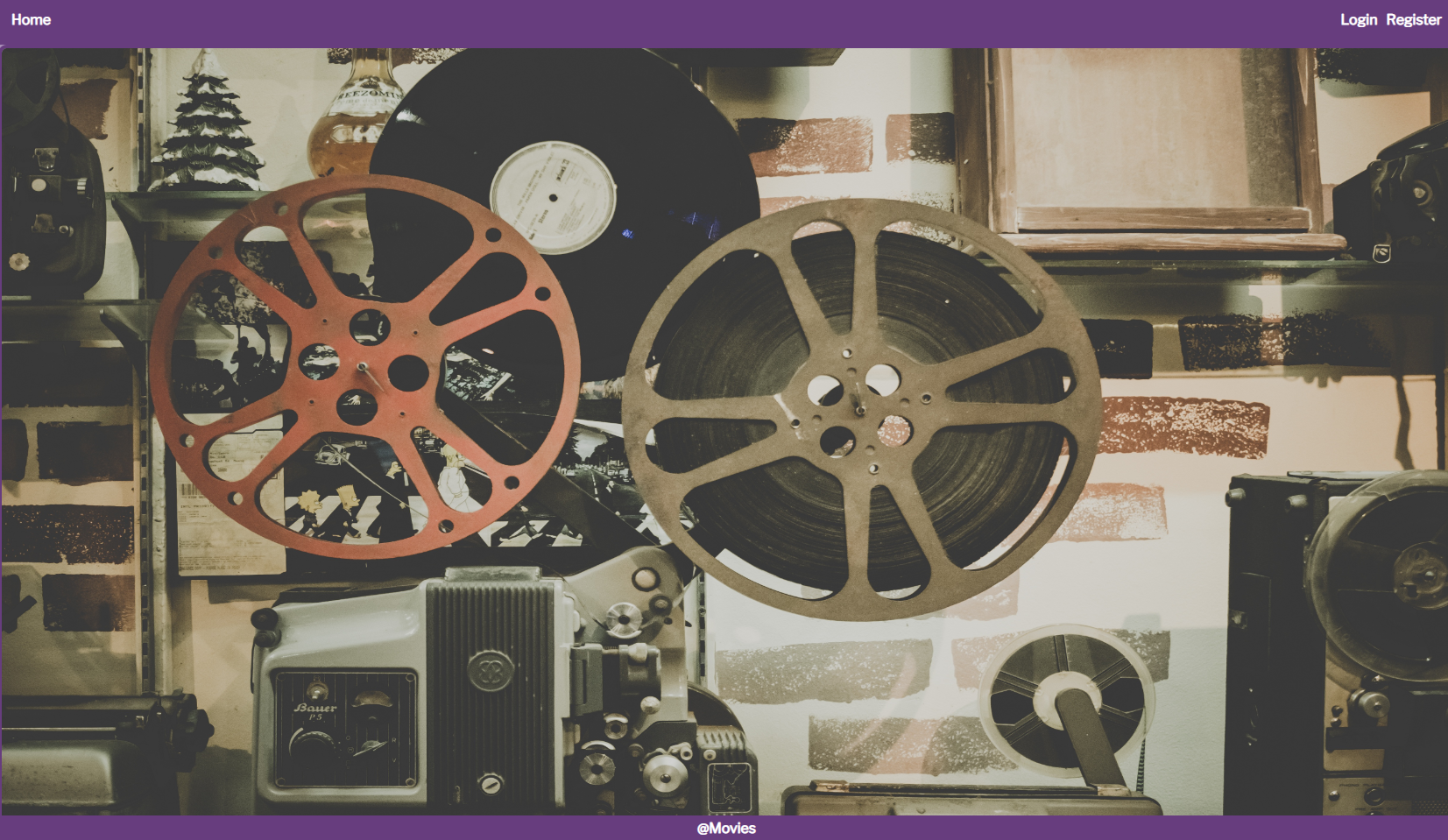
* Anonymous users have access only to the following links: [Home], [Login]and[Register].



### Home Screen

The initial page (view) should display:

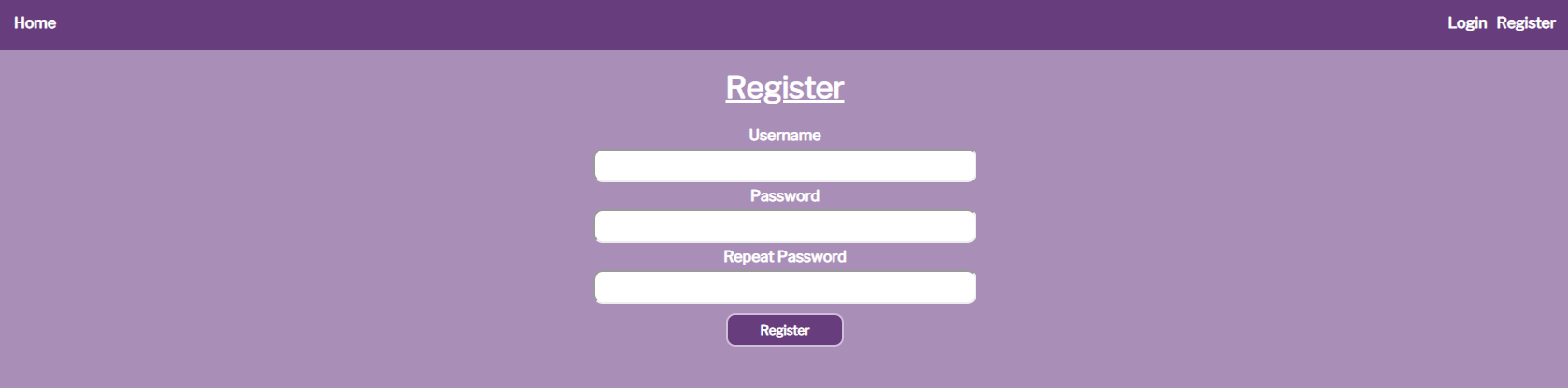
* **NavBar** with the following links [Home], [Login], [Register]
* The **image** from the resources (background.jpg)
* The **Footer**



### Register User

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

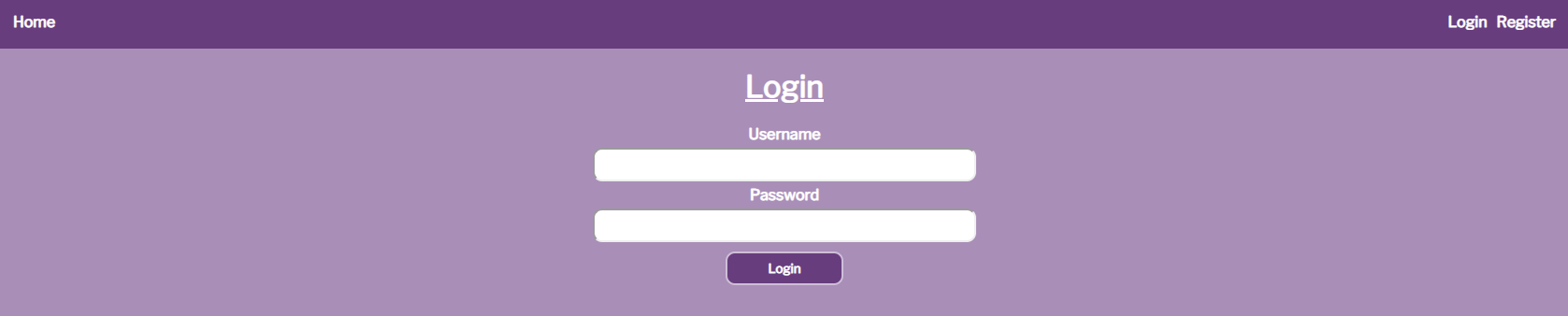
* You should make the following validations:
  + The **username** should be **at least 3 characters** long
  + The **password** should be **at least 6 characters** long
  + The **repeat password** should be **equal to the password**
* After a **successful registration**, a notification message **"User** **registration** **successful."** should be **displayed** and the user should be **logged-in**.
  + Home pageshould be **displayed** with the **User navbar**.
* In case of **error** (eg. **duplicate** username, **non-matching** passwords), an appropriate **error** **message** should be **displayed** and the client should be able to **try** to register again.
* Keep the **user session data** in the browser’s **session storage**.



### Login User

By given **username** and **password** the app should be able to login an existing user.

* After a **successful login**, a notification message **"Login** **successful."** should be displayed and the user should be **logged-in**.
  + Home pageshould be **displayed** with the **User navbar**.
* In case of **error**, an appropriate **error message** should be **displayed** and the user should be able to fill the **login form** again.
* Keep the **user session data** in the browser’s **session storage**.



### Logout

Logged-in users should be able to **logout** from the app.

* After a **successful** logout, a **notification** message **"Logout** **successful."** should be displayed and the user should be **logged-out**.
  + Login pageshould be **displayed** with the **user navbar**.
* The **Logout REST service** at the back-end should be obligatory **called** at logout.
* All local information in the browser (**user session data**) about the current user should be **deleted**.

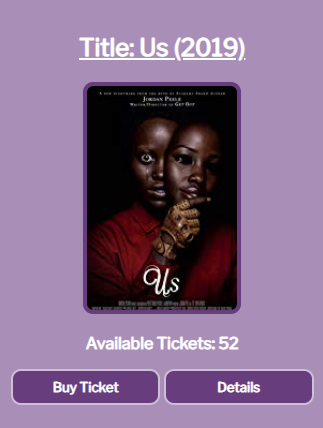
### Cinema

Logged-in users should be able to **view all movies**.

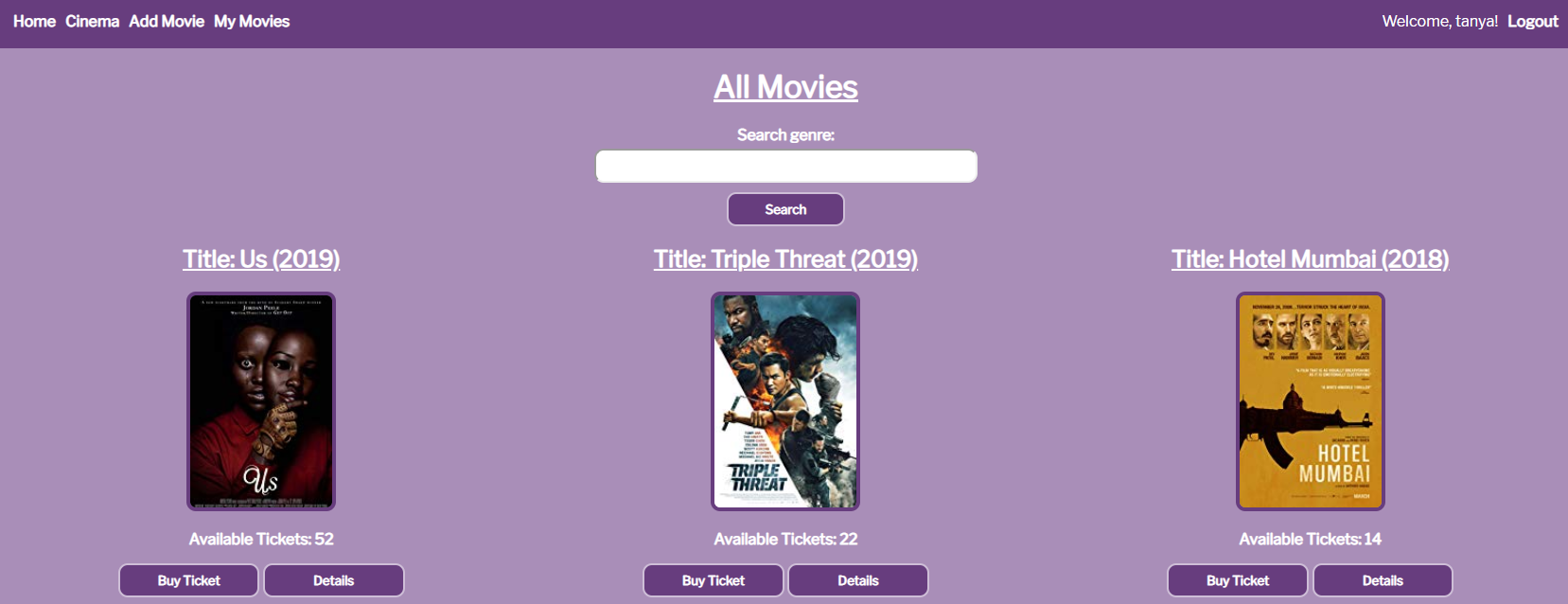
Clicking the [Cinema] **link** in the **NavBar** should **display** the Cinema page.

The Cinema page should **list all** of the **movies**, in a **formatted** and **ordered** manner:

* The **movies** should be listed in the **format** as shown in the web design (see the screenshot below).
* The **movies** must be **sorted** by the **available tickets** in **descending**.
* Each **movie** has **title**, **image, available tickets, buttons:** [Buy Ticket] **and** [Details].



* The **Cinema** page should look like this:

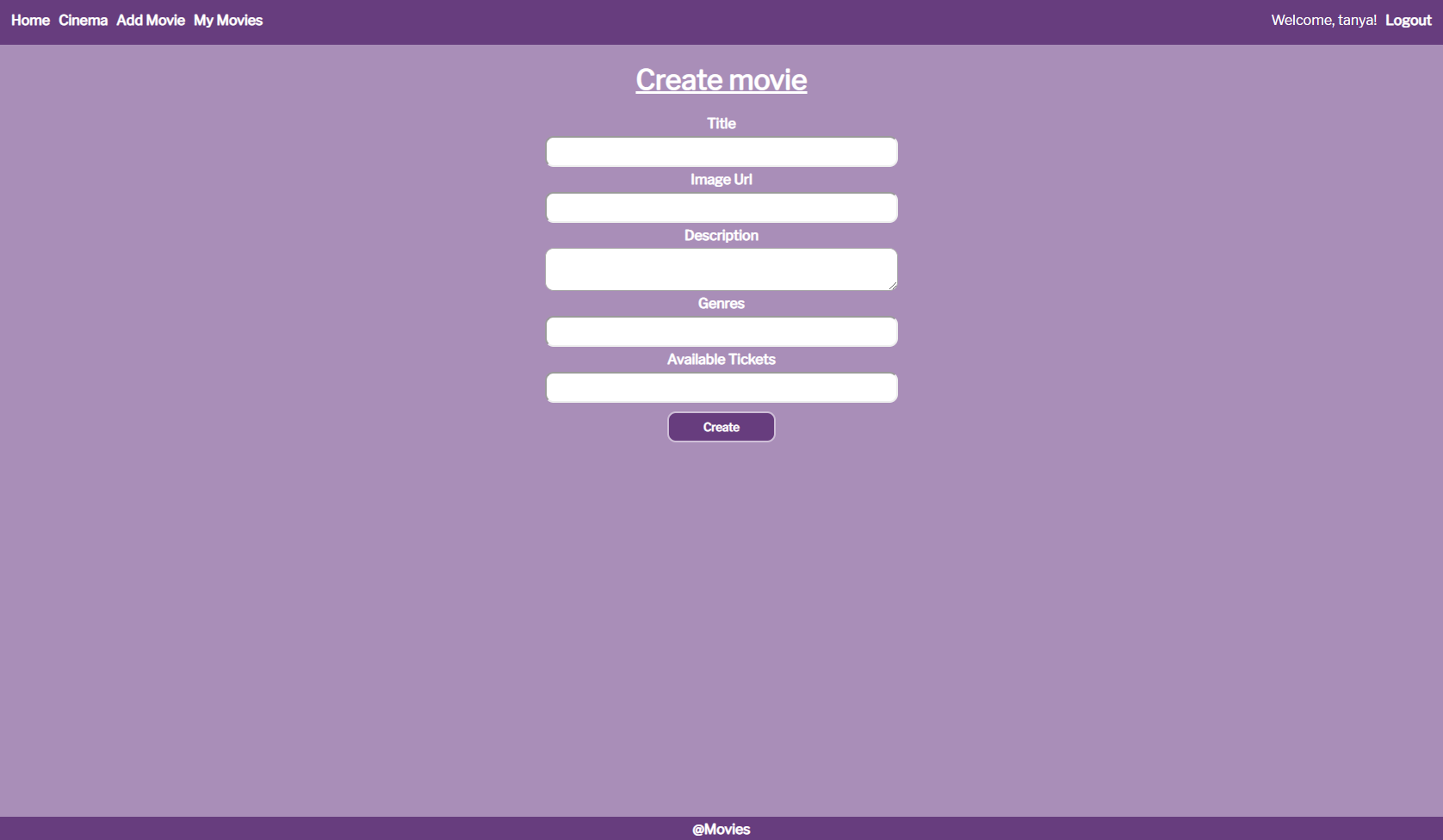


### Add Movie

Logged-in users should be able to **Add** movies.

Clicking the [Add Movie] **link** in the **NavBar** should **display** the Add Movie page.

* The form should contain the following validations:
  + The **title** should be **at least 6 characters** long.
  + The **description** should be **at least 10 characters** long.
  + The **image** should start with **"http://"** or **"https://"**.
  + The **available** **tickets** should be a **number**.
  + The **genres** must be separated by a **single space**.
* After a **successful** **movie** **creation**, a notification message **"Movie** **created** **successfully."** should be **displayed** and the Home pageshould be shown.
* The inputs fields in the form **should be cleared.**
* The newly created movie should be stored in the database collection "movies".

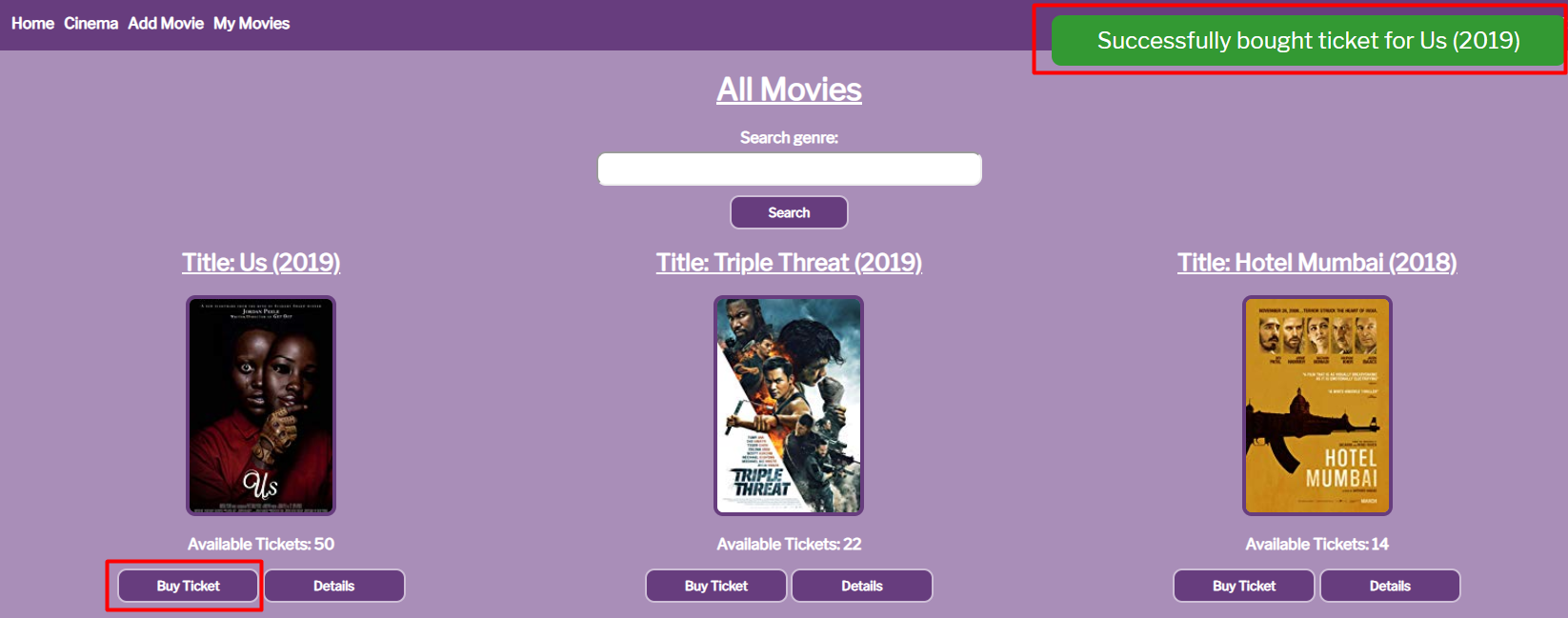


### Buy Ticket

Logged-in users should be able to **Buy tickets** formovies.

Clicking the [Buy Ticket] **link** of a **particular movie** should **decrease** its **available tickets**. If there are **no available tickets**, anappropriate **notification** should be **displayed** and the **available tickets** should **not be changed**.   
Users can **buy multiple tickets** for **one movie.**

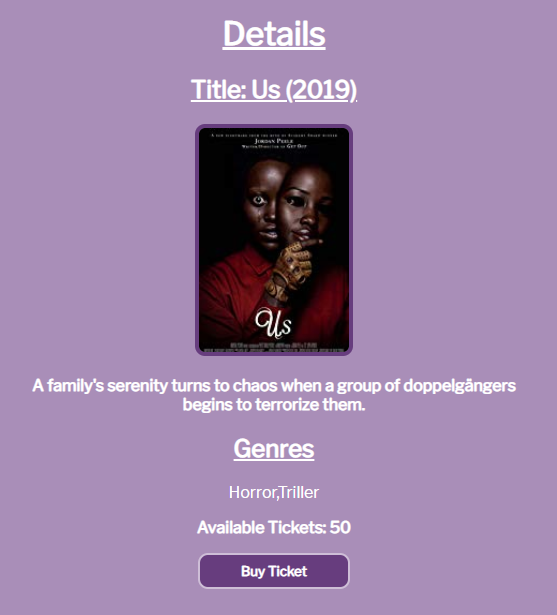
* After **successfully buying** a **ticket**, a notification message **"Successfully** **bought** **ticket** **for** **{movie** **name}!"** should be displayed.



### Movie Details

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

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



* From this **view**, the user should also be able to **buy tickets** - [Buy Tickets] link.

### My Movies

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

Clicking the [My Movies] **link** in the **NavBar** should **display** the My Movies page.

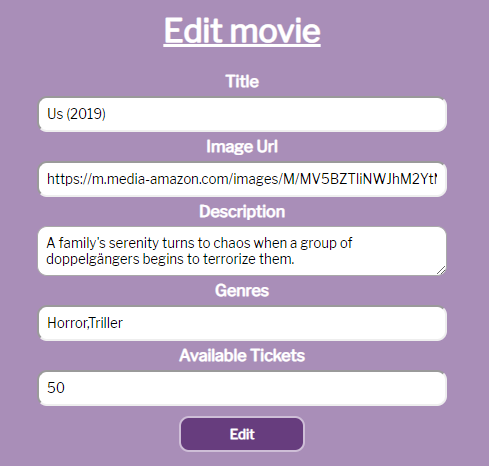
* The **movies** should be listed like in the **cinema** section, except theuser should see only the **movies** that he created and each movie should have [Edit]and [Delete] buttons, which will **show** the **corresponding pages**.



### Edit Movie

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

Clicking the [Edit] **link** of a **particular movie** on the My Movies page should **display** the Edit Movie page:

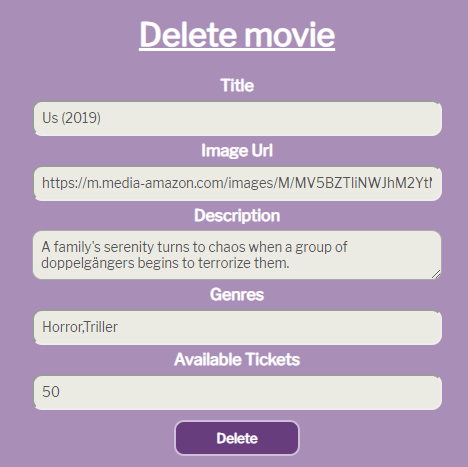


* After a **successful edit**, an appropriate notification should be **displayed**!

### Delete Movie

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

Clicking the [Delete] **link** of a **particular movie** on the My Movies page should **display** the Delete Movie page:



* The **input fields** should be **disabled**.
* After **successful** **movie delete** a notification message **"Movie** **removed** **successfully!"** should be displayed and the Home pageshould be **shown**.

### Search:

Logged-in users should be able to **search** formovies, by typing some **genre** (**case sensitive**) on the **search field** of the Cinema page.

Clicking the [Search] link should **display** the **same view** (Cinema page), but **only** with the **movies** which **contain** the **given genre**:

