

Project Report

1. Project Title: Online movie store

2. Name for the Website: Movie Store

3. Project Description:

(1) 3 kind of served users: guest user, regular user and admin user.

(2) User's available function:

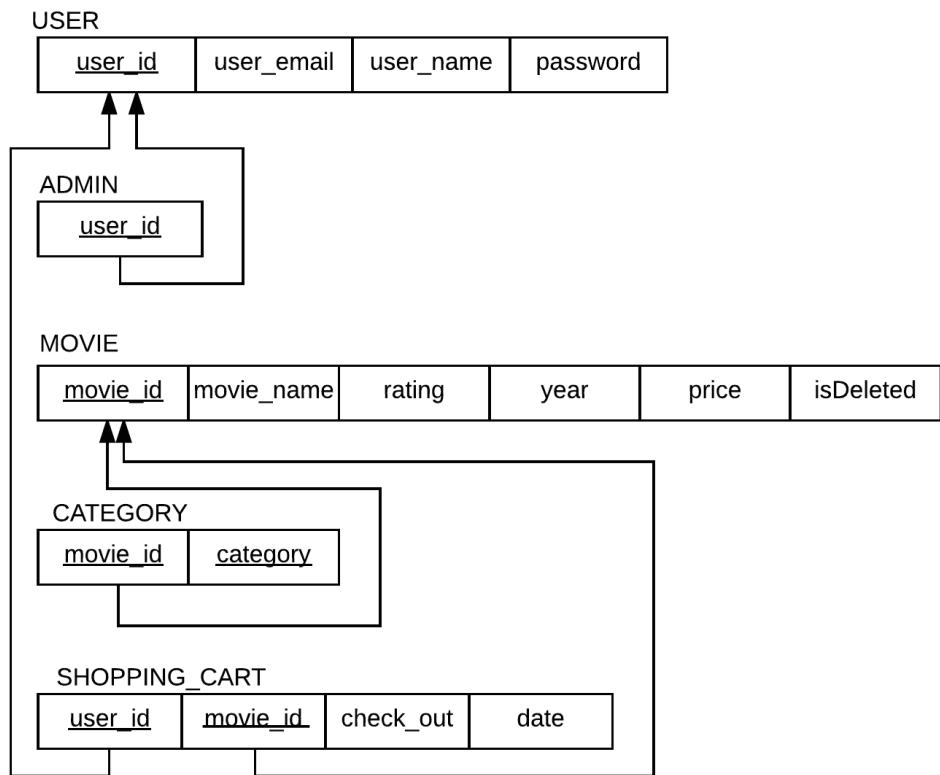
Guest user: seeing all the movie's list, searching specific movie by name, filter the result by category, see the detail of a specific movie, register with validated user name, email and password.

Regular user: log in with registered username/email and password, add to and remove movie from shopping cart and check out, see the purchase history, log off.

Admin user: log in with registered username/email and password which added into admin table of database by database administrator, add new movie (including upload movie picture and summery), update exist movie with new content or delete exist movie (soft delete: mark the isDeleted attribute in movie table as 1 which default 0.), log off.

(3) default search will list all items.

4. Database Design:



SQL statement:

```

DROP DATABASE IF EXISTS MOVIE_STORE;
CREATE DATABASE MOVIE_STORE;

USE MOVIE_STORE;

DROP TABLE IF EXISTS USER;
CREATE TABLE USER (
    user_id      int      not null AUTO_INCREMENT,
    user_email   varchar(40)  not null,
    user_name   varchar(40)  not null,
    password     varchar(80)  not null,
    CONSTRAINT pk_user PRIMARY KEY (user_id),
    CONSTRAINT uk_user_email UNIQUE (user_email),
    CONSTRAINT uk_user_name UNIQUE (user_name)
);

```

```
DROP TABLE IF EXISTS ADMIN;
CREATE TABLE ADMIN (
    user_id      int      not null,
    CONSTRAINT pk_admin PRIMARY KEY (user_id),
    CONSTRAINT fk_admin FOREIGN KEY (user_id) REFERENCES
USER(user_id)
);
```

```
DROP TABLE IF EXISTS MOVIE;
CREATE TABLE MOVIE(
    movie_id      int      not null AUTO_INCREMENT,
    movie_name    varchar(40) not null,
    movie_rating  varchar(10) not null,
    year         varchar(5)  not null,
    price        float(10,2) not null,
    isDeleted    int(2)     DEFAULT 0,
    CONSTRAINT pk_movie PRIMARY KEY (movie_id),
    CONSTRAINT uk_movie UNIQUE (movie_name)
);
```

```
DROP TABLE IF EXISTS CATEGORY;
CREATE TABLE CATEGORY (
    movie_id      int      not null,
    category     varchar(40) not null,
    CONSTRAINT pk_category PRIMARY KEY (movie_id, category),
    CONSTRAINT fk_category FOREIGN KEY (movie_id) REFERENCES
MOVIE(movie_id)
);
```

```
DROP TABLE IF EXISTS SHOPPING_CART;
CREATE TABLE SHOPPING_CART (
    user_id      int      not null,
    movie_id     int      not null,
    check_out   boolean not null,
    date        date    not null,
    CONSTRAINT pk_shopping_cart PRIMARY KEY (user_id, movie_id),
    CONSTRAINT fk_shopping_cart_user FOREIGN KEY (user_id)
REFERENCES USER(user_id),
    CONSTRAINT fk_shopping_cart_movie FOREIGN KEY (movie_id)
REFERENCES MOVIE(movie_id)
```

);

5. Languages/frameworks used for implementation

- (1) Language: HTML, CSS, JavaScript, PHP, JSON, AJAX.
- (2) Framework: Bootstrap

6. Screenshots for main functionalities

Movie Store

[home](#)[login](#)[register](#)

A Dog's Purpose



Comedy \$12.99

Beauty and Beast



Adventure \$20.99

Guardians Of The



Action \$14.99

Rogue One



Action \$14.99

Ghost in the Shell



Crime \$20.99

Fantastic Beasts



Fantasy \$15.99

Kong: Skull Island



Action \$20.99

Going in Style



Comedy \$20.99

1 2 3 4 5

picture 1: homepage

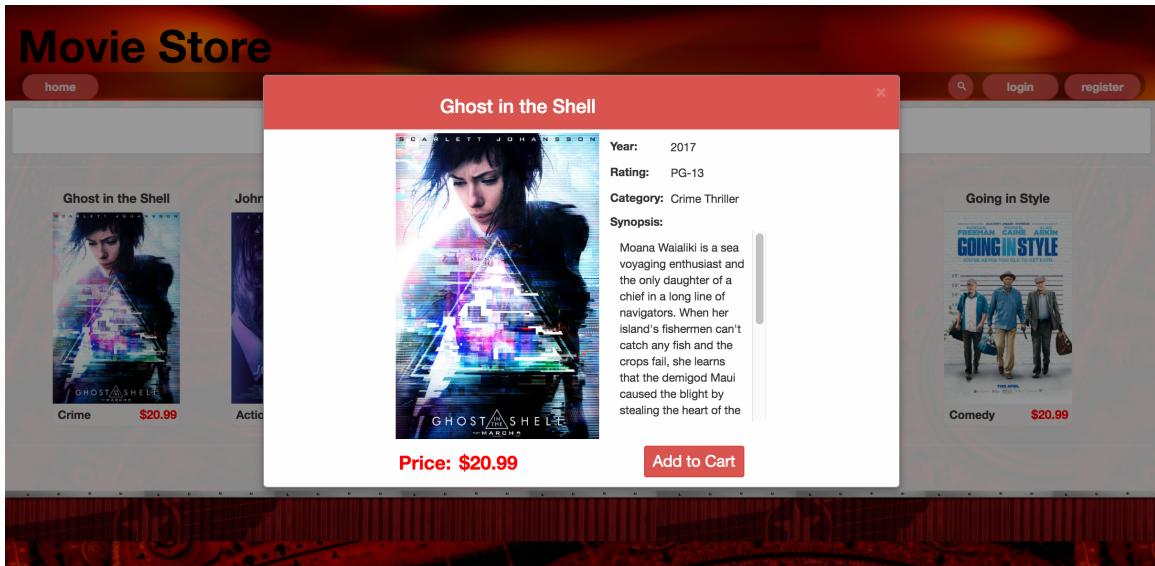
Movie Store

home login register

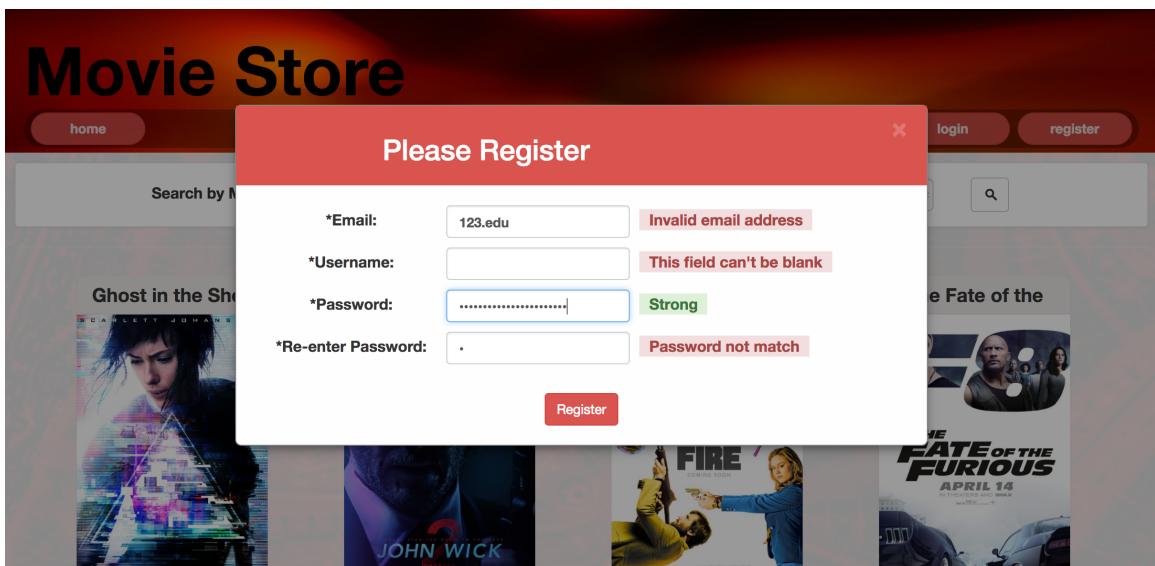
Search by Movie Name: Search by Category: Action

Ghost in the Shell  Crime \$20.99	John Wick: Chapter  Action \$15.99	Free Fire  Action \$16.99	The Fate of the  Action \$22.99
Get Out  Crime \$21.99	Going in Style  Comedy \$20.99	1	

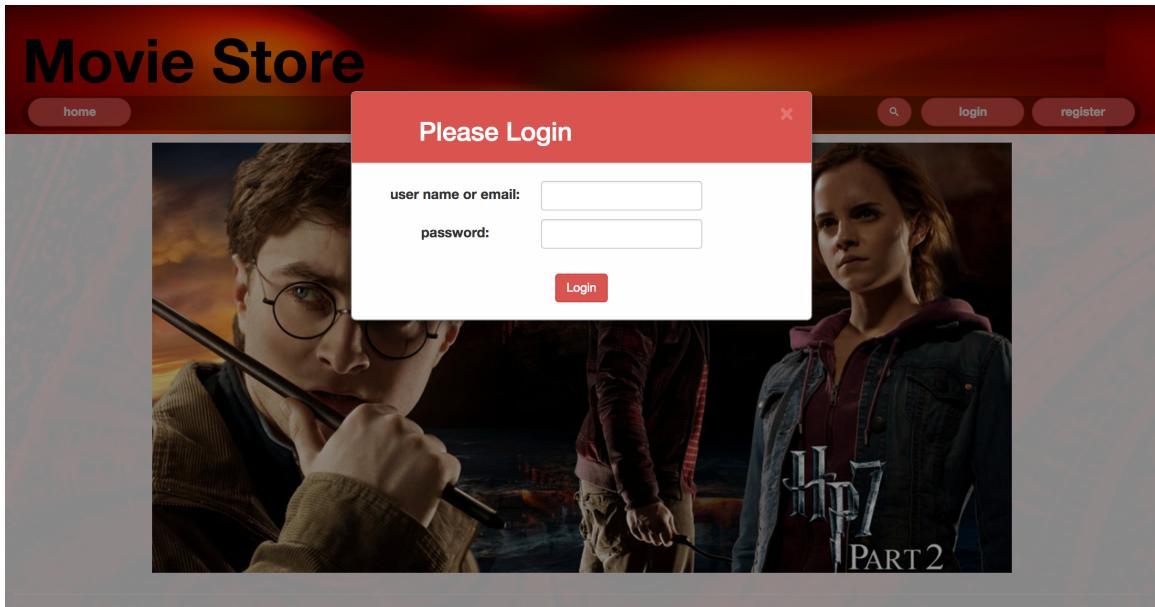
picture 2: search and filter



picture 3: detail



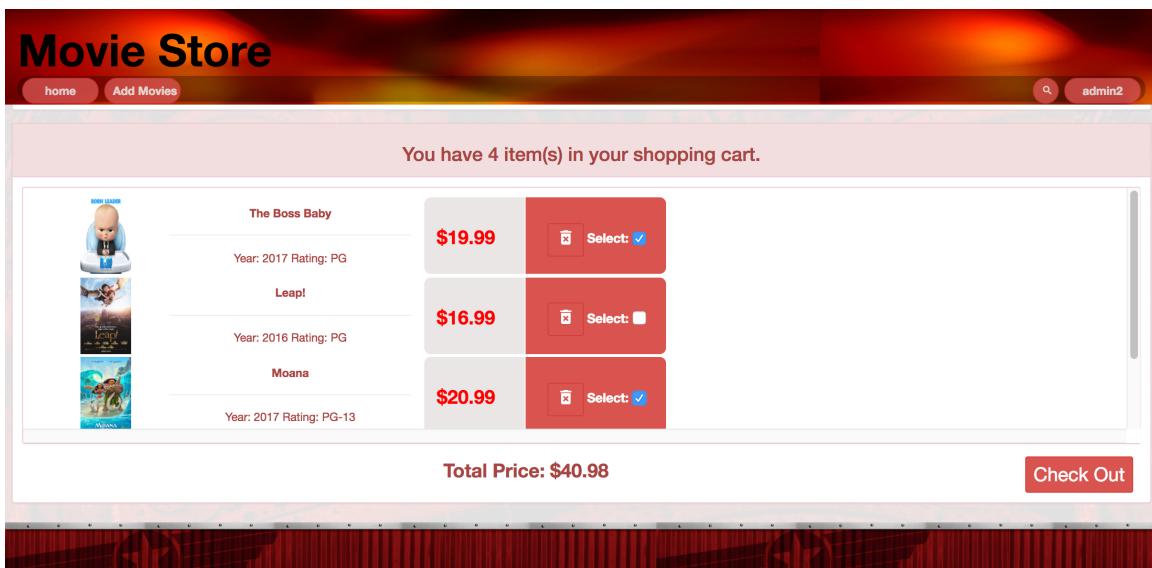
picture 4: register and validation



picture 5: login



picture 6: menu



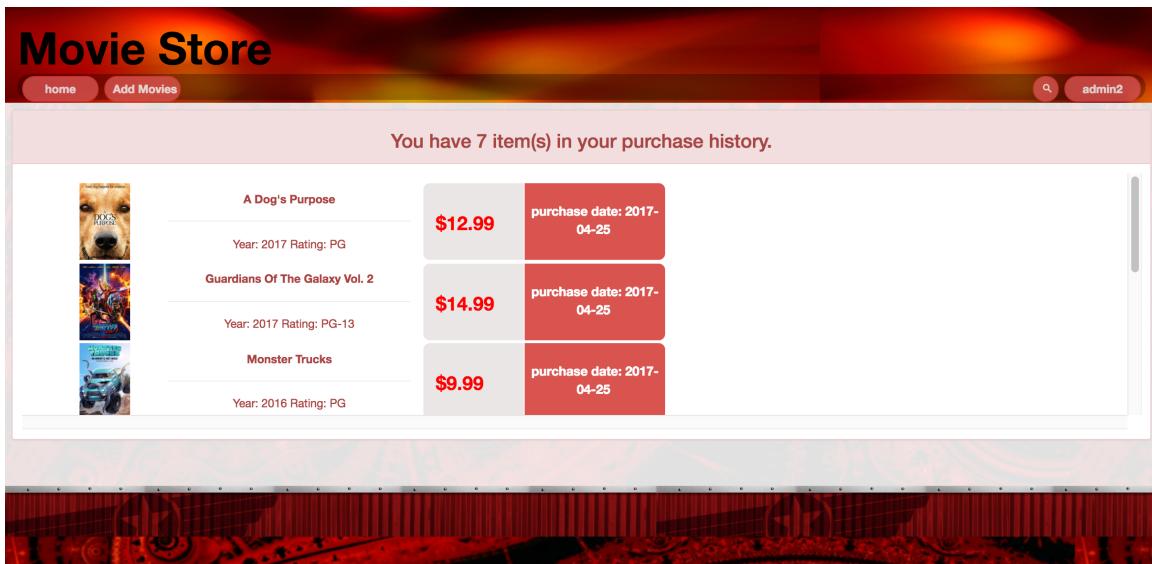
The screenshot shows a movie store interface with a red and orange gradient background. At the top, there's a navigation bar with 'Movie Store' and links for 'home' and 'Add Movies'. On the right, there's a search icon and a user account labeled 'admin2'. Below the navigation, a message says 'You have 4 item(s) in your shopping cart.' A table lists four movies with their prices and selection checkboxes:

	The Boss Baby	\$19.99	<input checked="" type="checkbox"/> Select: <input checked="" type="checkbox"/>
	Leap!	\$16.99	<input type="checkbox"/> Select: <input type="checkbox"/>
	Moana	\$20.99	<input checked="" type="checkbox"/> Select: <input checked="" type="checkbox"/>
Year: 2017 Rating: PG-13			

Total Price: \$40.98

Check Out

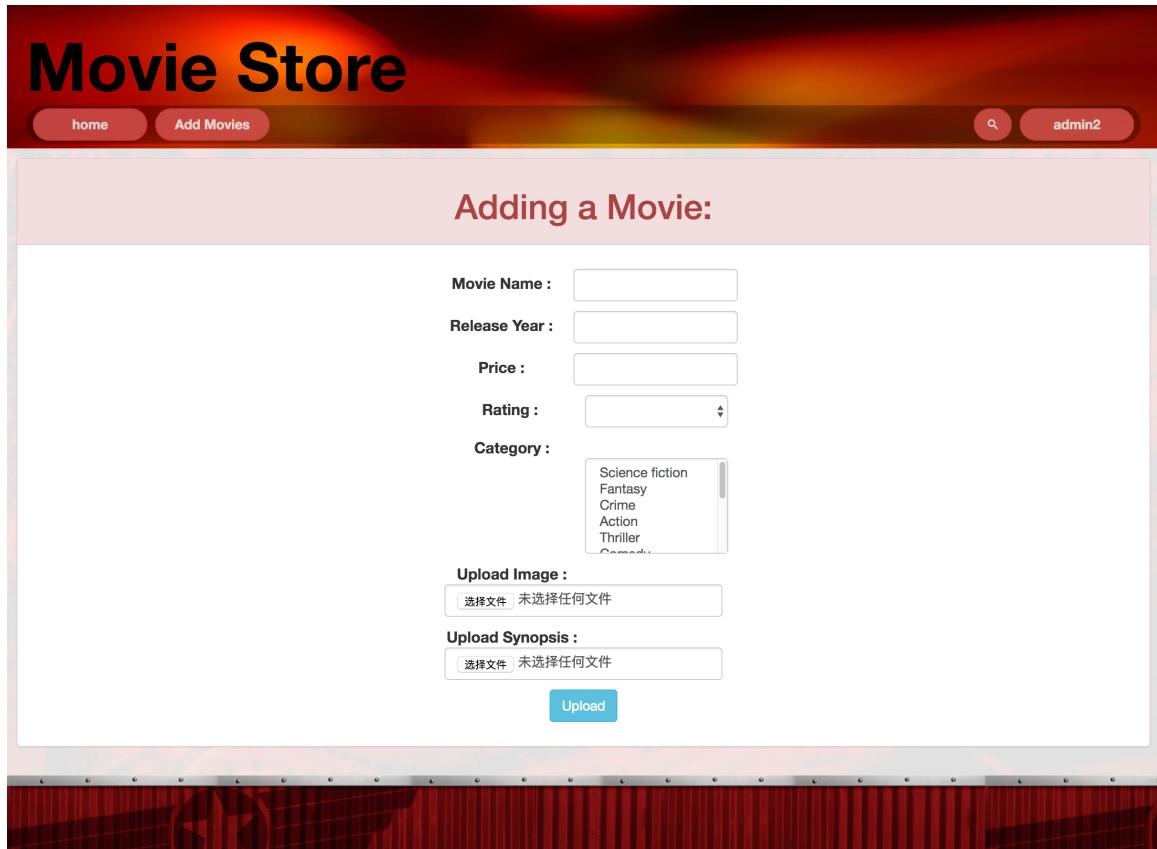
picture 7: shopping cart and check out



The screenshot shows a movie store interface with a red and orange gradient background. At the top, there's a navigation bar with 'Movie Store' and links for 'home' and 'Add Movies'. On the right, there's a search icon and a user account labeled 'admin2'. Below the navigation, a message says 'You have 7 item(s) in your purchase history.' A table lists three movies with their purchase details:

	A Dog's Purpose	\$12.99	purchase date: 2017-04-25
	Guardians Of The Galaxy Vol. 2	\$14.99	purchase date: 2017-04-25
	Monster Trucks	\$9.99	purchase date: 2017-04-25
Year: 2017 Rating: PG-13			

picture 8: purchase history



picture 9: add movie



picture 10: update and delete

7. Team Members and work division:

Name	NetID	Work devision
Jiafeng Liao	jxl155830	Add movie module, Filter movie module, Update movie module, Front and back end ajax communication regulation, Web frame design and edit, Web style sheet edit, front end script.
Biyun Shi	bxs161130	Paging creation, Movie detail module, Shopping cart module, homepage movie slider,
Yu Gong	yxg162930	Login module, Register module, Delete movie module, History module, Database design, ajax file communication,