CS 353 Project
"DMGTV"
Project Final Report

Gökberk Beydemir - 21902638

Melis Atun - 21901865

Mert Barkın Er - 21901645

Doruk Kantarcıoğlu - 21902319

TA: Zülal Bingöl

TABLE OF CONTENTS

Brief Description

DMGTV is an online movie rental system that allows users to rent and buy movies according to their interests. The system can be used by two different types of users: customers and employees. Users can log in to the system by entering their username and password. When a user logs in to the system, they see a menu at the top of the page which consists of the following choices: Movies, Profile, Friends, Go to my Movies, and Log out. By clicking on the Movies button, the customer is directed to the Movies page. On this page, they can select a movie to rent or buy based on their interest. They can see the production year, rating, price per month, price to buy, whether the movie is age restricted or not, IMDB rating, like count of the movies, and they can rent or buy the movie as well as see the reviews of the movie. Also, they can filter these features if they like from the top right corner and they can search for a movie using the name of it on this page as well. Furthermore, by clicking on the Profile button, the user is directed to the Profile page which is the page that the users first see when they log in to the system. They can see their wishlist, profile details, and credit card information on this page. They can edit their profile information and credit card information from here as well. Moving on, by clicking on the Friends button, the user is directed to the Friends page where they can see the list of friends that they have added and they can remove their existing friends as well as add a new friend from this page by entering the username of their friend. If the username is not valid, there will be an error message displayed at the bottom left corner of the page. Moreover, by clicking on the Go to my Movies button, the user is directed to the Go to my Movies page which consists of the movies that they have rented or bought from the system. They can also review the movie from this page as well as see the reviews that they have made before. Also, users can return the movie that they have rented back to the system. Finally, users can Log out by clicking on the Log out button at the top right of the menu and they will be directed to the login page where they can log in to the system again.

Contribution of Each Member

- ❖ Backend Team

  -Gökberk Beydemir: Implemented backend codes using Java and Spring Framework.

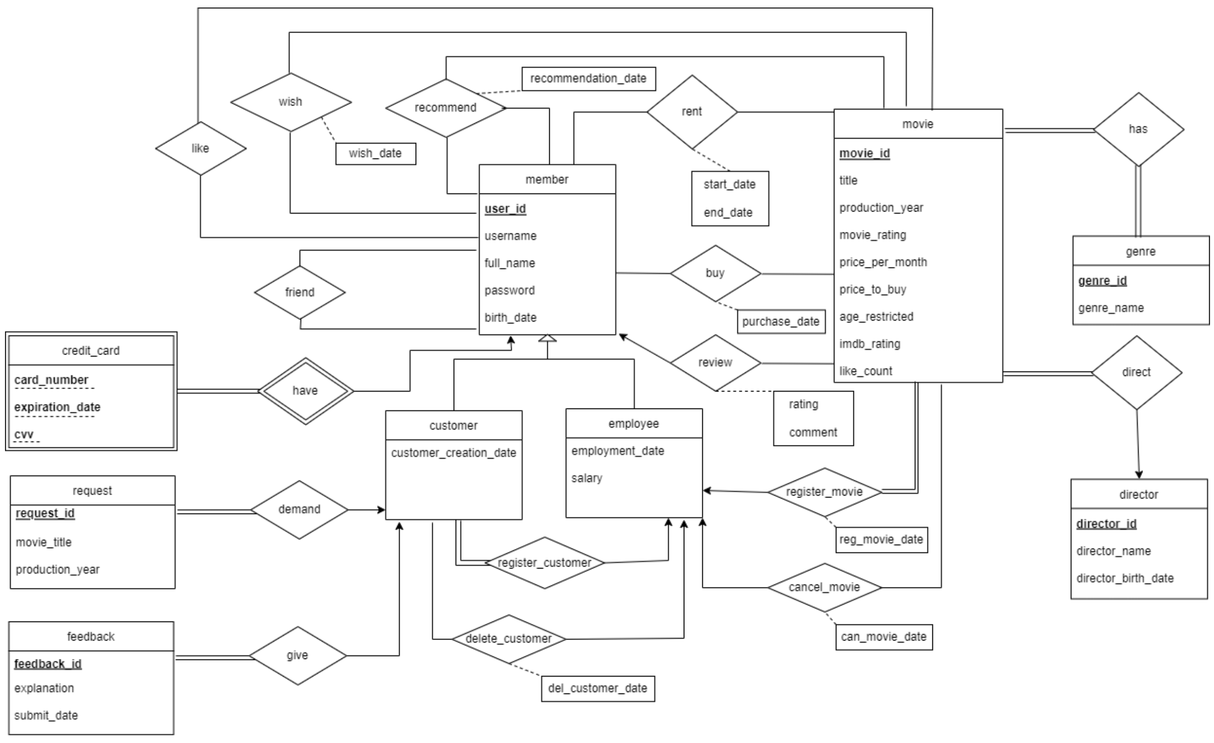
  -Mert Barkın Er: Implemented backend codes using Java and Spring Framework.

- ❖ Frontend Team

  -Doruk Kantarcıoğlu: Implemented frontend of the pages using ReactJS.

  -Melis Atun: Implemented CSS components of the pages.

Final E/R Diagram



Final List of Tables

- **Movie**

  Relational Model

movie (movie_id, title, production_year, rating, price_per_month, price_to_buy, age_restricted, imdb_rating, like_count)

Primary Key: movie_id

```
CREATE TABLE movie (
	id UUID PRIMARY KEY,
	title VARCHAR(64) NOT NULL,
	production_Year INTEGER NOT NULL,
	rating NUMERIC(2, 1) NOT NULL,
	price_per_month INTEGER NOT NULL,
	price_to_buy INTEGER NOT NULL,
	age_restricted BOOLEAN NOT NULL,
	imdb_rating NUMERIC(2, 1) NOT NULL,
	like_count INTEGER
);
```

- **Users**

  Relational Model

  member (user_id, username, full_name, password, birth_date)

  Primary Key: user_id

```
CREATE TABLE users (
	id UUID PRIMARY KEY,
	username VARCHAR(20) NOT NULL UNIQUE,
	password VARCHAR(16) NOT NULL,
	full_name VARCHAR(50) NOT NULL,
	birth_date DATE NOT NULL
);
```

- **Review**

  Relational Model

  review (user_id, movie_id, rating, comment)

  Primary Keys: user_id, movie_id

  Foreign Keys: user_id, movie_id

```
CREATE TABLE review (
	id UUID PRIMARY KEY,
	user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE ON UPDATE CASCADE,
	movie_id UUID NOT NULL REFERENCES movie(id) ON DELETE CASCADE ON UPDATE CASCADE,
	rating NUMERIC(2, 1) NOT NULL,
	comment TEXT,
	UNIQUE (user_id, movie_id)
);
```

- **Request**

  Relational Model

  request (request_id, movie_title, production_year)

  Primary Key: request_id

```
CREATE TABLE request (
	id UUID PRIMARY KEY,
	movie_name VARCHAR(64) NOT NULL,
	production_year NUMERIC(4, 0) NOT NULL
);
```

- **Rent**

  Relational Model

  rent(user_id, movie_id, start_date, end_date)

  Primary Keys: user_id, movie_id, start_date

  Foreign Keys: user_id, movie_id

```
CREATE TABLE rent (
        id UUID PRIMARY KEY,
        user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE ON
UPDATE CASCADE,
        movie_id UUID NOT NULL REFERENCES movie(id) ON DELETE CASCADE
ON UPDATE CASCADE,
        start_date DATE NOT NULL,
        end_date DATE,
        UNIQUE (user_id, movie_id, start_date)
);
```

- **Buy**

  Relational Model

  buy (user_id, movie_id, purchase_date)

  Primary Keys: user_id, movie_id

  Foreign Keys: user_id, movie_id

```
CREATE TABLE buy (
        id UUID PRIMARY KEY,
        user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE ON
UPDATE CASCADE,
        movie_id UUID NOT NULL REFERENCES movie(id) ON DELETE CASCADE
ON UPDATE CASCADE,
        purchase_date DATE NOT NULL,
        UNIQUE (user_id, movie_id)
);
```

- **Friend**

    Relational Model

    friend (first_user_id,second_user_id)

    Primary Keys: first_user_id, second_user_id
    Foreign Keys: first_user_id, second_user_id

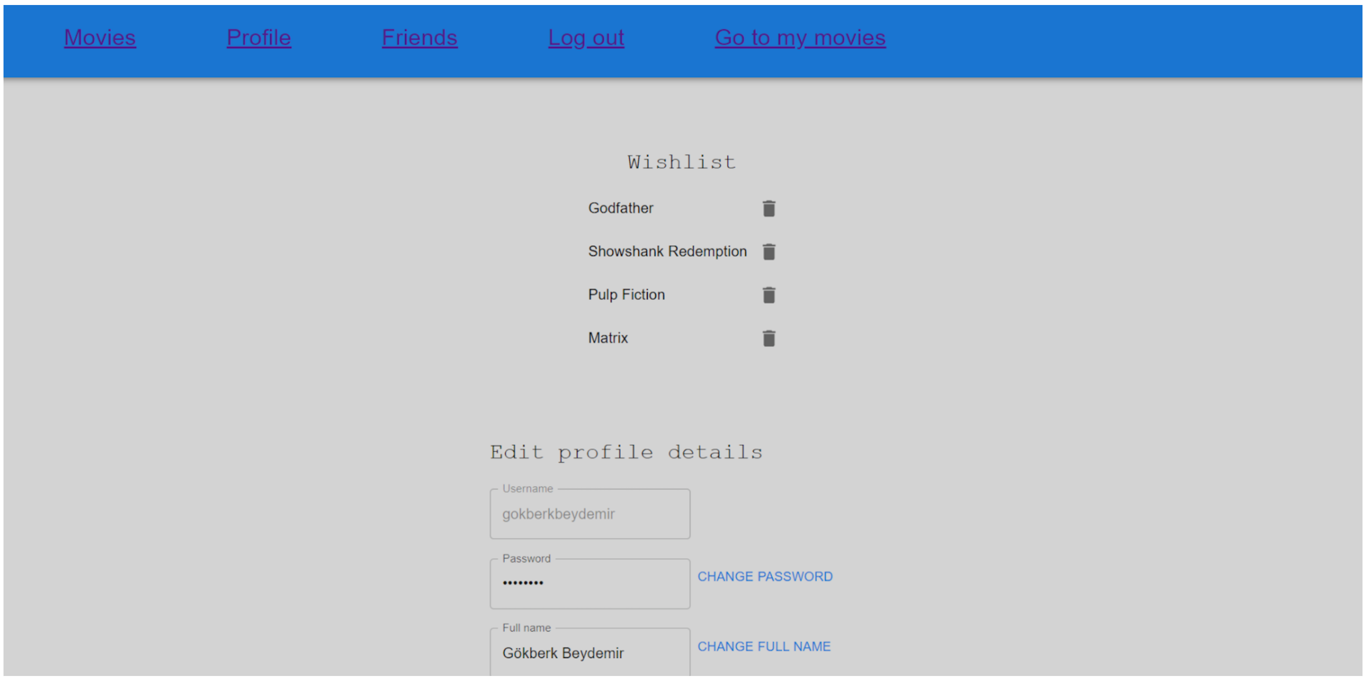
```
CREATE TABLE friend (
        id UUID PRIMARY KEY,
        first_username VARCHAR(20) NOT NULL REFERENCES users(username) ON
DELETE CASCADE ON UPDATE CASCADE,
        second_username VARCHAR(20) NOT NULL REFERENCES users(username) ON
DELETE CASCADE ON UPDATE CASCADE,
        UNIQUE (first_username, second_username)
);
```

Implementation Details

For the backend of our project, we used PostgreSQL for the database and Java and Spring Framework. We decided to use those tools since we were familiar with them from our internships. For frontend, we used React framework since our frontend team was familiar with React. We used the PgAdmin interface to check if our queries worked or not. We also used Postman to manually check if our queries worked or not. Normally, Spring Framework is not used for SQL, it automatically creates SQL queries in the background for the developer. However, by writing custom queries using @Query annotation, we have managed to use it. While initializing the database, we entered the tables and default data in the file called “data.sql”. This file runs before the Spring project, and creates our database.

For the frontend of our project, we used ReactJS along with CSS for the components. In our React page components, functional programming was utilized. We used axios library to execute HTTP requests to the backend server. The data that was obtained/altered was rendered on the pages. This was how the client-server integration was maintained.

Sample Outputs



Wishlist

Godfather

Showshank Redemption

Pulp Fiction

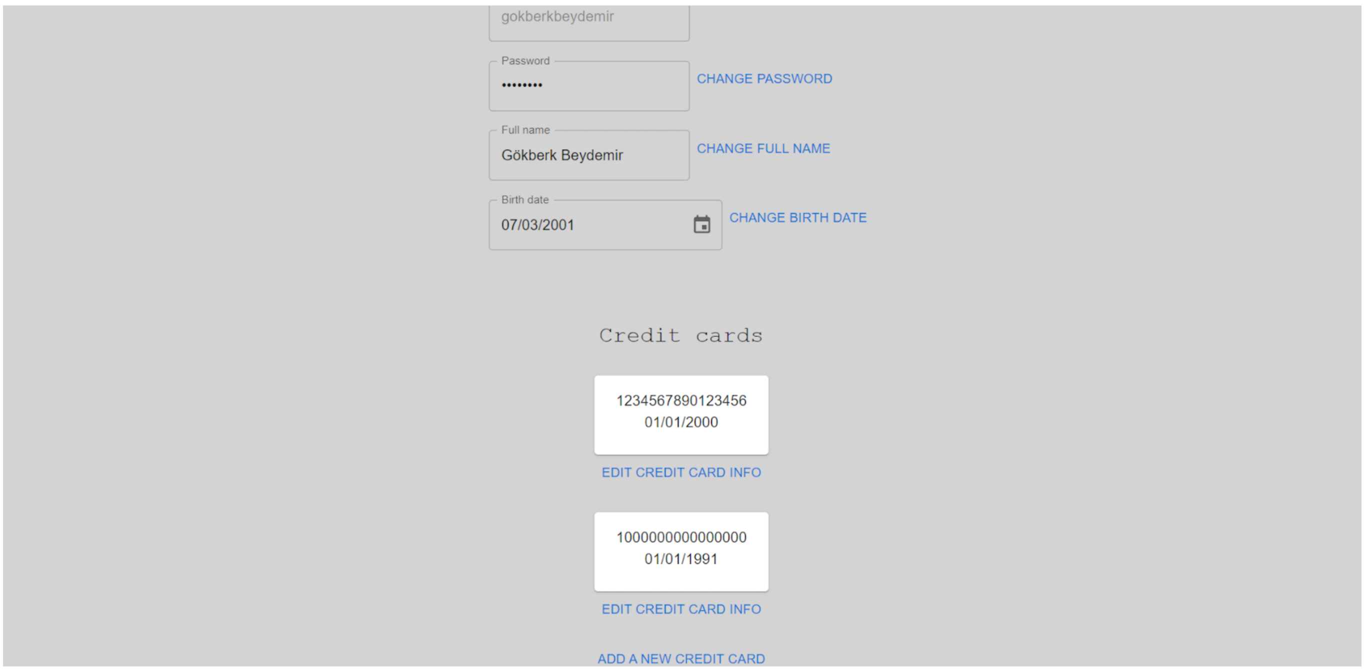
Matrix

Edit profile details

Username: gokberkbeydemir

Password: ••••••••

Full name: Gökberk Beydemir



gokberkbeydemir

Password: •••••••• CHANGE PASSWORD

Full name: Gökberk Beydemir CHANGE FULL NAME

Birth date: 07/03/2001 CHANGE BIRTH DATE

Credit cards

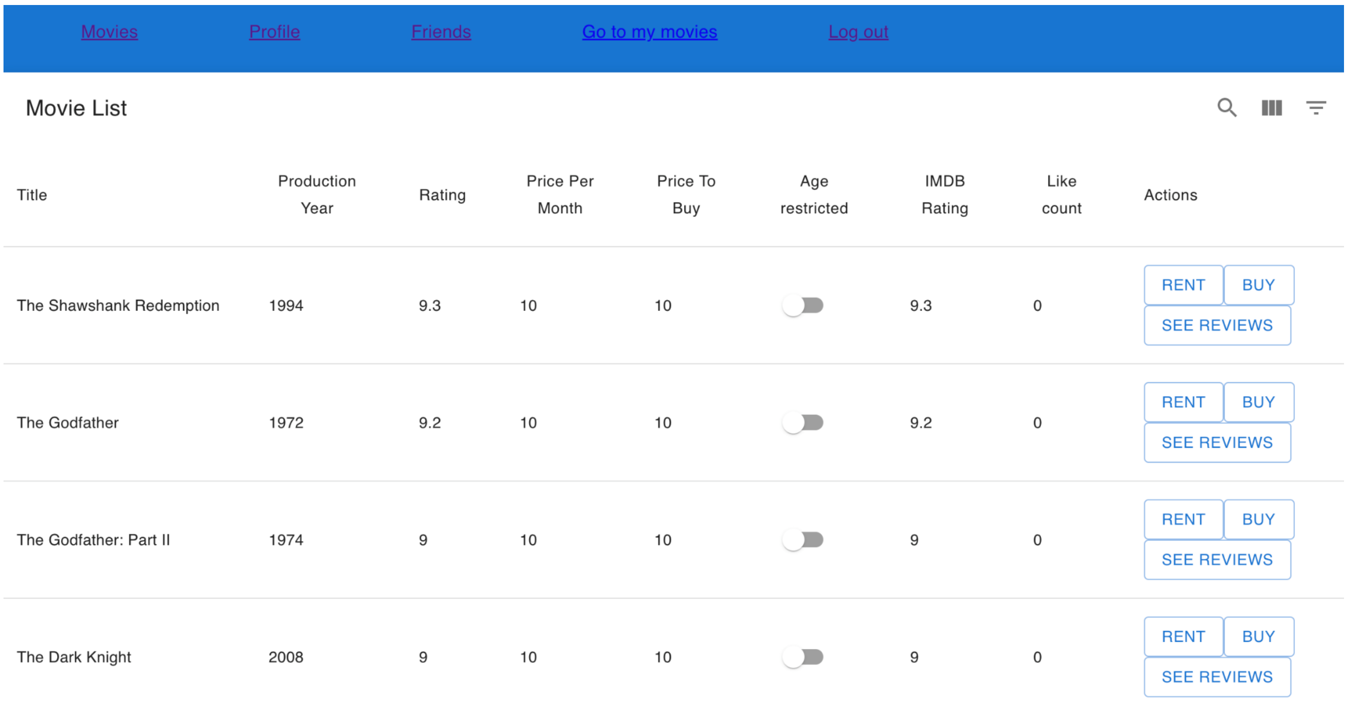
1234567890123456
01/01/2000

EDIT CREDIT CARD INFO

1000000000000000
01/01/1991

EDIT CREDIT CARD INFO

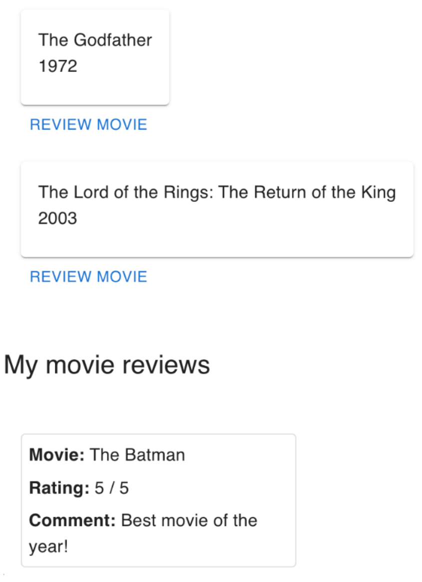
ADD A NEW CREDIT CARD



Movies Profile Friends Go to my movies Log out

Movie List

Title	Production Year	Rating	Price Per Month	Price To Buy	Age restricted	IMDB Rating	Like count	Actions
The Shawshank Redemption	1994	9.3	10	10		9.3	0	RENT BUY SEE REVIEWS
The Godfather	1972	9.2	10	10		9.2	0	RENT BUY SEE REVIEWS
The Godfather: Part II	1974	9	10	10		9	0	RENT BUY SEE REVIEWS
The Dark Knight	2008	9	10	10		9	0	RENT BUY SEE REVIEWS


The Godfather
1972

REVIEW MOVIE

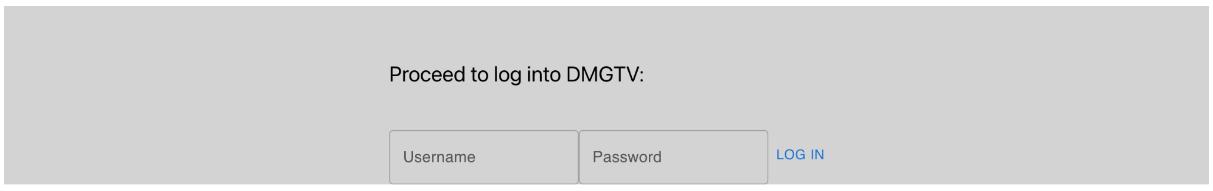
The Lord of the Rings: The Return of the King
2003

REVIEW MOVIE

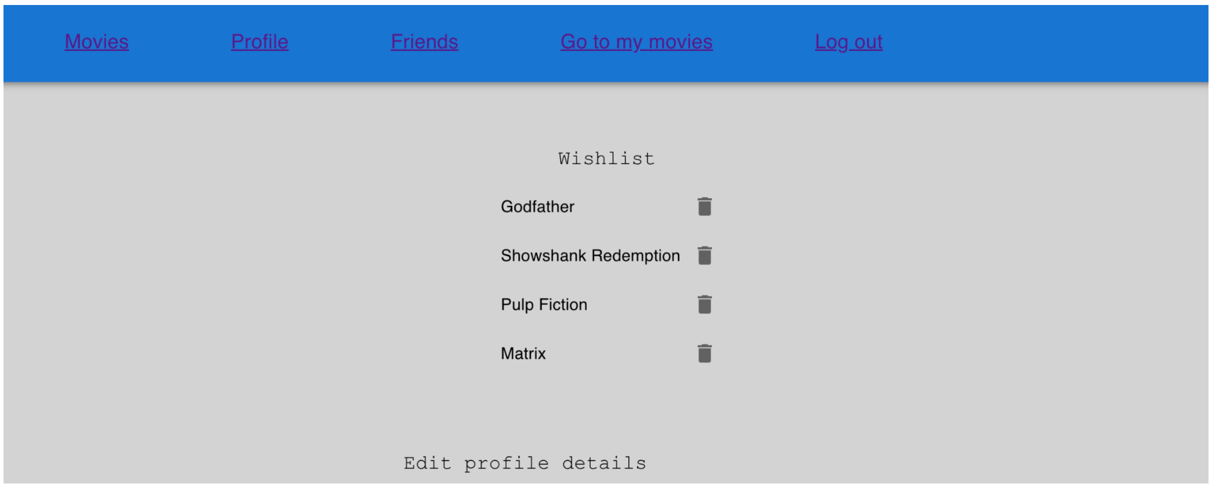
My movie reviews

Movie: The Batman
Rating: 5 / 5
Comment: Best movie of the year!

User's Manual


Proceed to log into DMGTV:

Users of our system can log in to the system using the login page by entering their username and password.


Movies Profile Friends Go to my movies Log out

Wishlist

Godfather

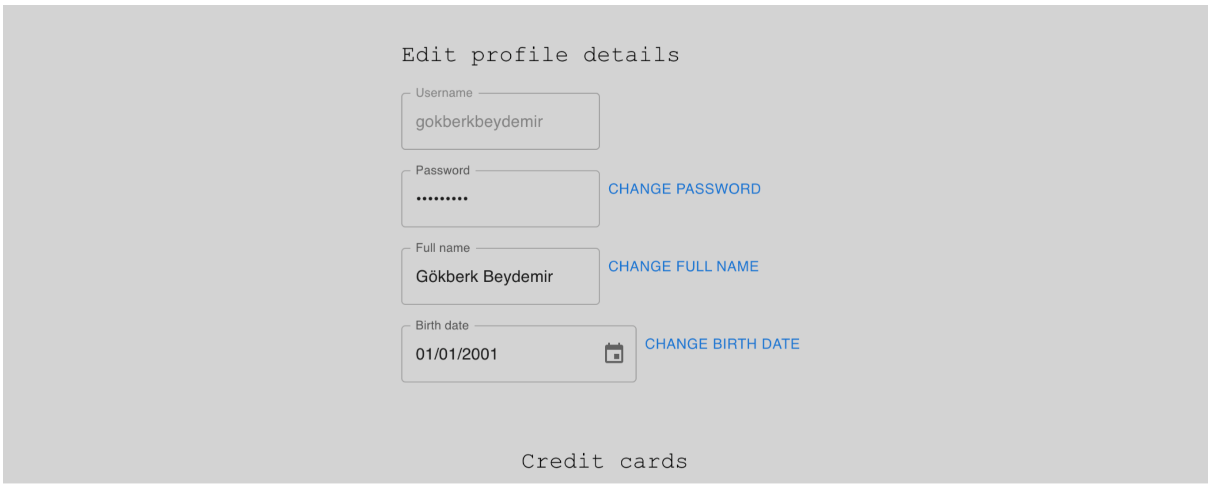
Showshank Redemption

Pulp Fiction

Matrix

Edit profile details

The first page for a user after logging in displays the movies that they have added to their wishlist. The customer can delete the movies from their wishlist from here.


Edit profile details

Username: gokberkbeydemir

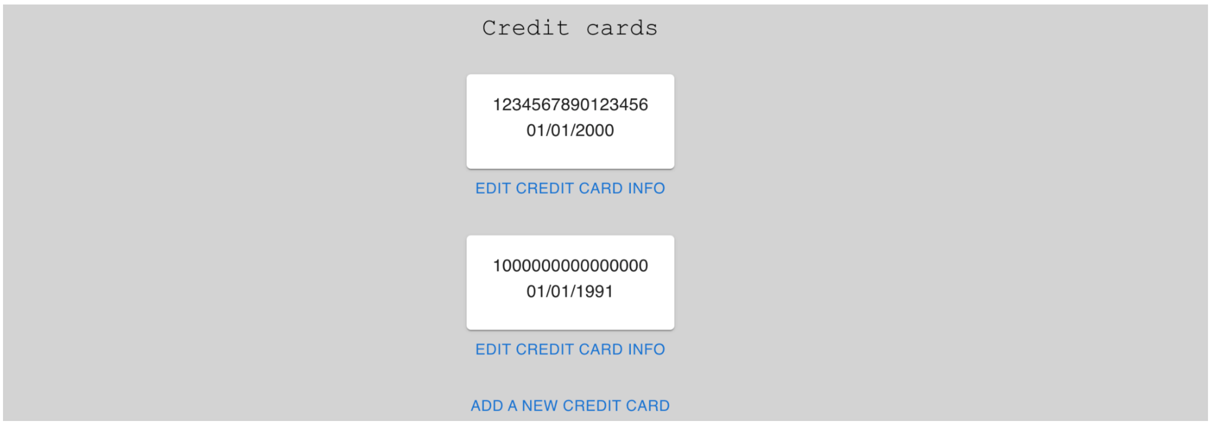
Password: ••••••••• CHANGE PASSWORD

Full name: Gökberk Beydemir CHANGE FULL NAME

Birth date: 01/01/2001 CHANGE BIRTH DATE

Credit cards

The user can also edit their profile details from here: change their password, change their full name, change their birth date.


Credit cards

1234567890123456
01/01/2000

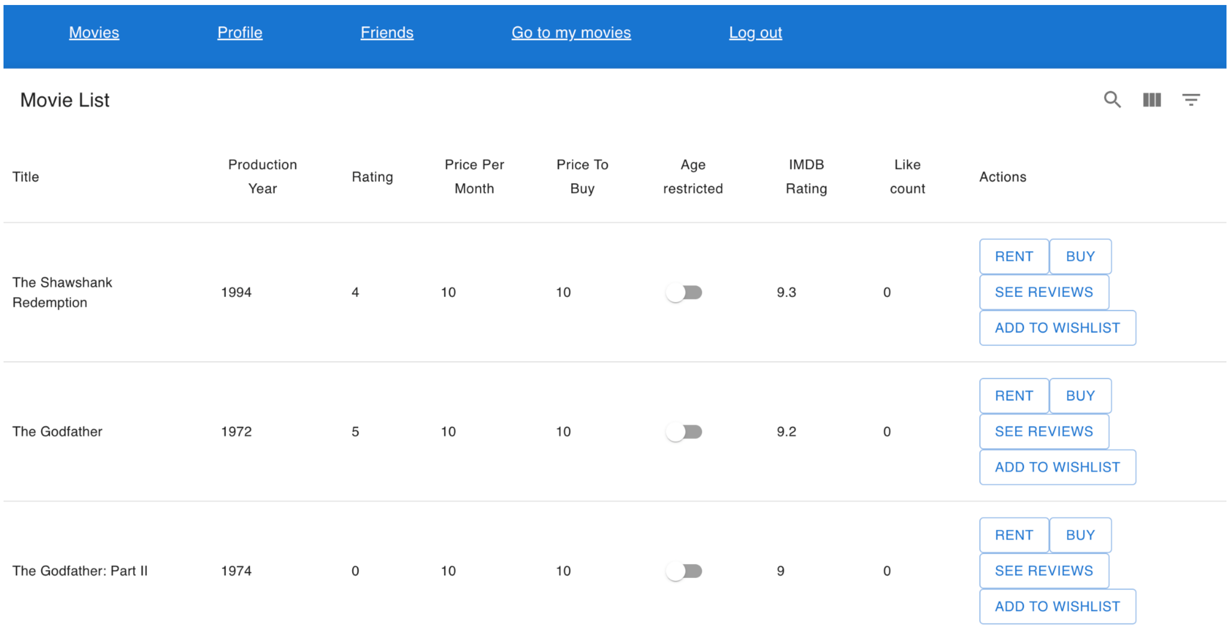
EDIT CREDIT CARD INFO

1000000000000000
01/01/1991

EDIT CREDIT CARD INFO

ADD A NEW CREDIT CARD

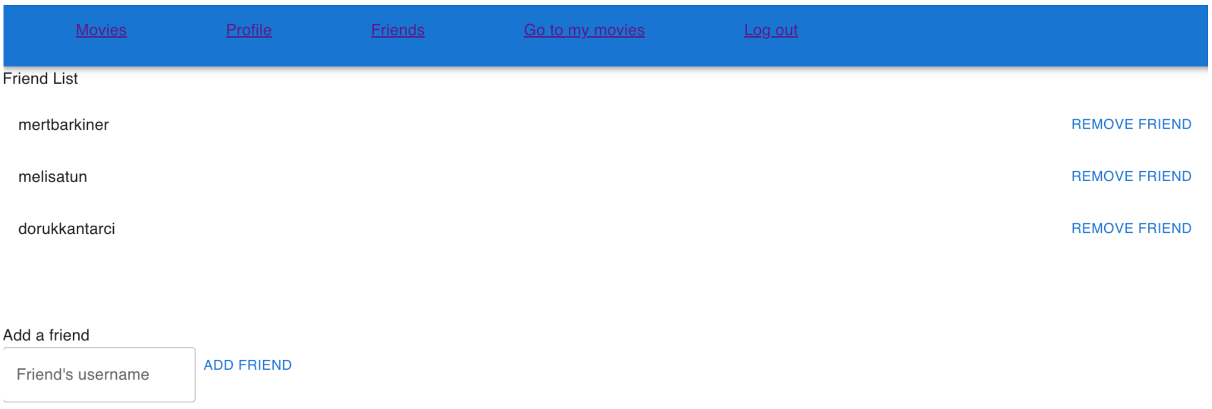
The credit cards of a customer are also shown at the bottom of this page and they can edit their credit card information or they can add a new credit card from here.



Movie List

Title	Production Year	Rating	Price Per Month	Price To Buy	Age restricted	IMDB Rating	Like count	Actions
The Shawshank Redemption	1994	4	10	10		9.3	0	RENT BUY SEE REVIEWS ADD TO WISHLIST
The Godfather	1972	5	10	10		9.2	0	RENT BUY SEE REVIEWS ADD TO WISHLIST
The Godfather: Part II	1974	0	10	10		9	0	RENT BUY SEE REVIEWS ADD TO WISHLIST

By clicking on "Movies" from the menu at the top left corner, users can see the movie list along with the production year, rating, price per month, price to buy, whether the movie is age restricted or not, IMDB rating, like count and they can rent or buy the movie as well as see the reviews of the movie from this page. Also, they can filter these features and search for a movie using its name at the top right corner of this page.



Friend List

mertbarkiner REMOVE FRIEND

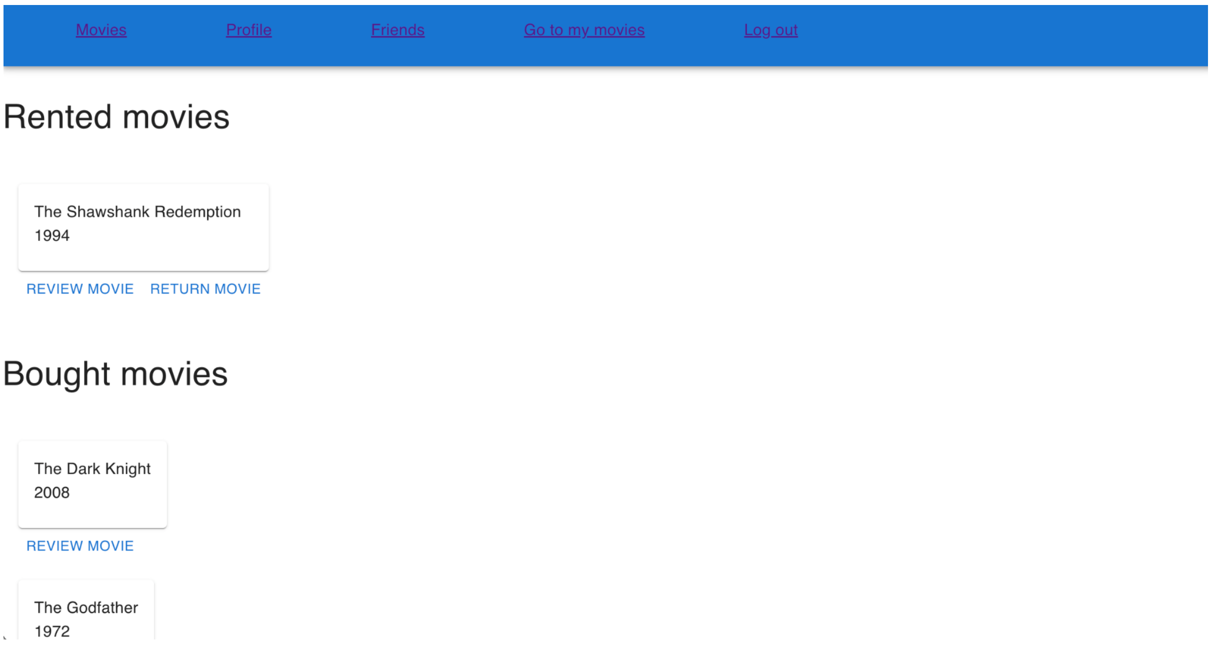
melisatun REMOVE FRIEND

dorukkantarci REMOVE FRIEND

Add a friend

Friend's username ADD FRIEND

By clicking on “Friends” from the menu at the top, users can see their existing friend list and they can remove their friends from here. They can also add a new friend from this page by entering the username of their friend.



Rented movies

The Shawshank Redemption
1994

REVIEW MOVIE RETURN MOVIE

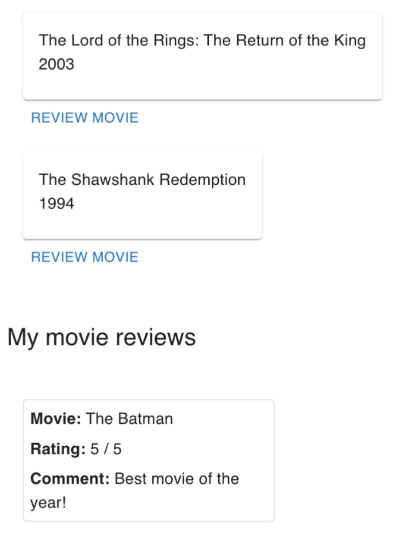
Bought movies

The Dark Knight
2008

REVIEW MOVIE

The Godfather
1972

On the “Go to my movies” page, users can see the movies that they have rented from the system and they can review this movie or return it back to the system. They can also see the movies that they have bought.


The Lord of the Rings: The Return of the King
2003

REVIEW MOVIE

The Shawshank Redemption
1994

REVIEW MOVIE

My movie reviews

Movie: The Batman
Rating: 5 / 5
Comment: Best movie of the year!

Users can also see the reviews of movies that they have made before on this page.