# APPENDIX OF REPORT: INSTALLATION AND USER GUIDE

## CONTENT

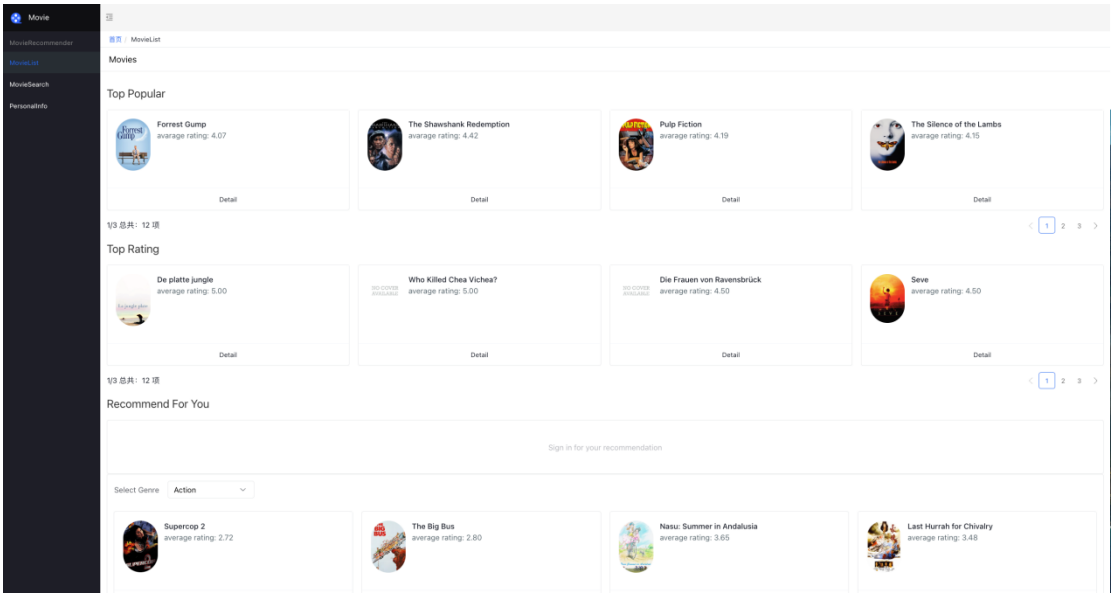
# 1. Use The Online Movie Recommendation System

## Prepare

A device which has a modern web browser.

## Enter The Website

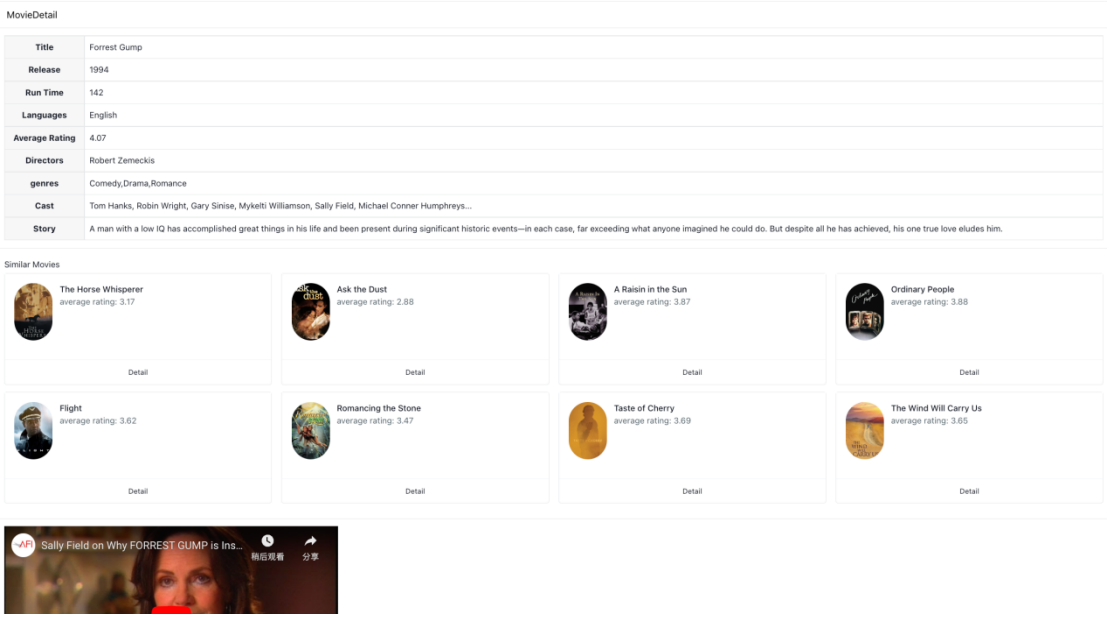
Launch a web browser, go to https://nus2.com




Now you can see different movie lists. But “Recommend For You” is empty, because you haven’t logged in.

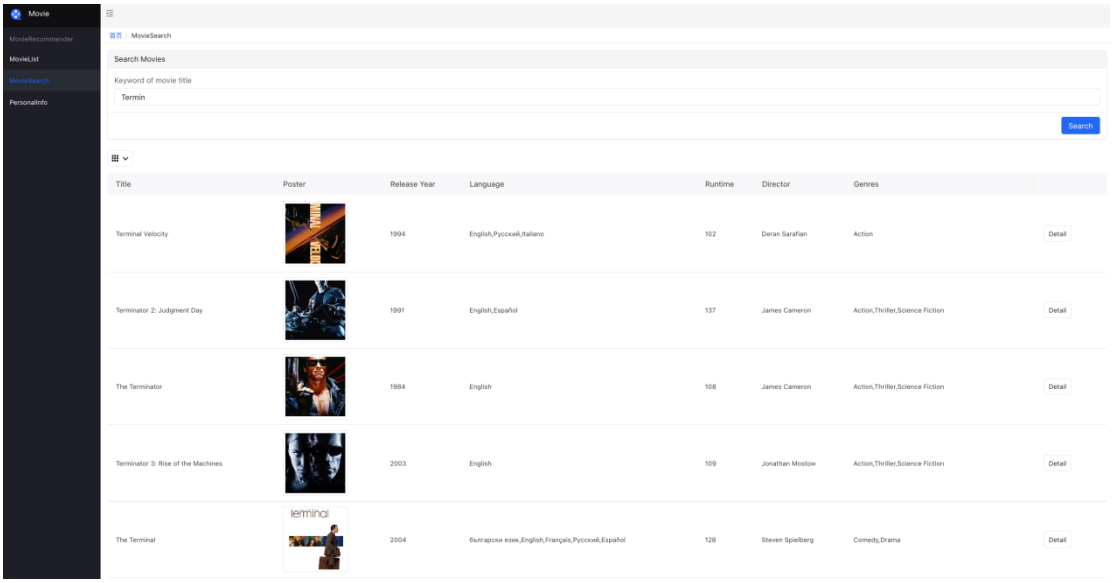
## MovieDetail

Each movie card has a “Detail” button. Push the button, then you enter the movie detail page.



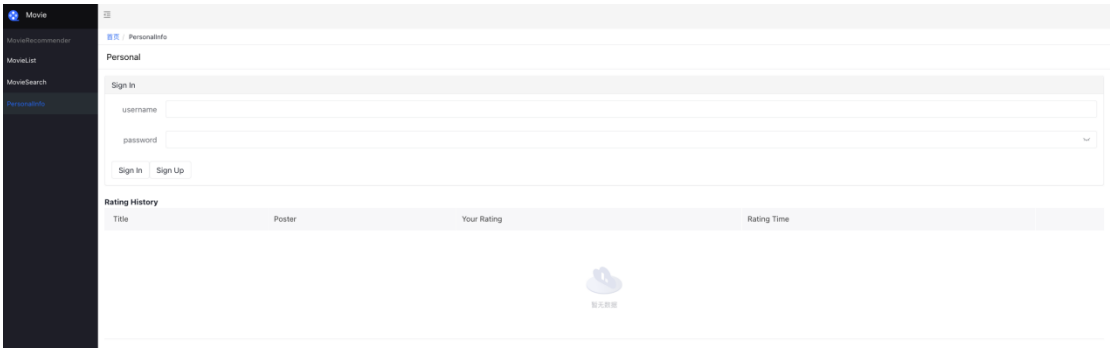
On this page, you can not only get the basic information, posters and YouTube video about this movie but also get a similar movie list which is calculated by our algorithm.

## MovieSearch

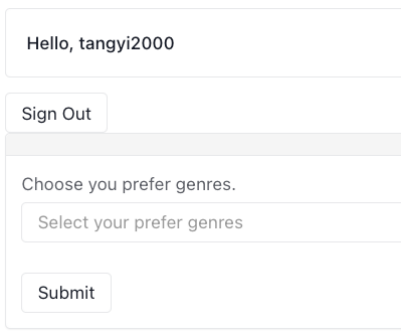



On MovieSearch page, you enter the keyword, and then get a movie list as the searching result.

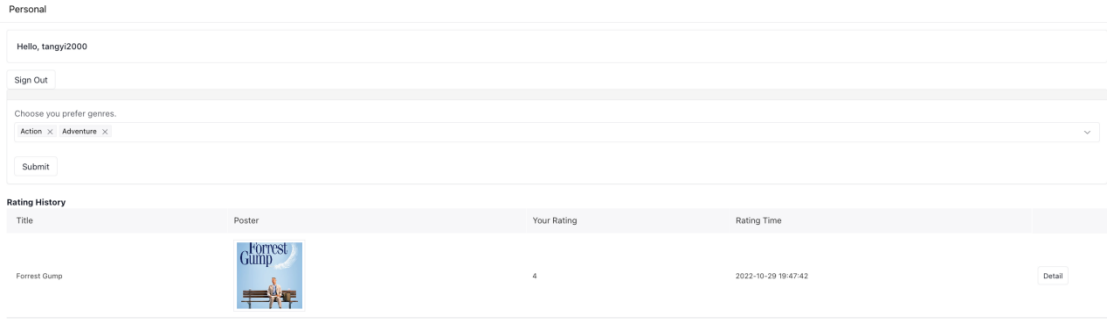
## PersonalInfo



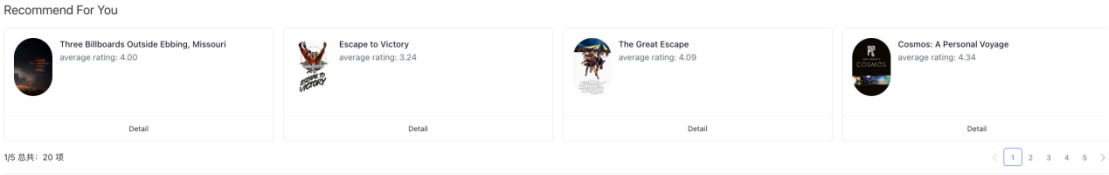

In PersonalInfo page, if you are a new user, you are supposed to sign up to release more functions.



After you sign up and sign in, you can select your preferred genres, which help us to recommend movies for you.




You can also rate a movie on MovieDetail page, and your rating history is showed on your PersonalInfo page.




And now, you can have your own recommend list in the MovieList page.

# 2. Deploy The System Yourself

All of these above are guidance for Ubuntu 20.04 System.
Please pay attention to the software and system versions, we do not guarantee that the same results can be obtained under different versions.
You can download the files from:
https://github.com/MichaelGu718/IRS-PM-Group-8-Movie-Recommendation-System

change every "{password}" in the code with "nusiss"
First, download package information from all configured sources.
***sudo apt-get update***

## 2.1 Back End

Install Java 8.
***sudo apt install openjdk-8-jdk***

confirm Java 8 is installed successfully.
***java -version***

Download the "MovieServer.jar", and enter the download directory.
***nohup java -jar MovieServer.jar </dev/null &>/dev/null &***

Now the spring application is running on localhost:8088.

## 2.2 Front End

install Node.js and npm
***sudo apt install nodejs npm***

confirm that Node.js is installed successfully.
***nodejs --version***

Download the MovieRecommendSystemFrontEnd.zip and unzip it.
***unzip MovieRecommendSystemFrontEnd.zip***

Go into the directory, install the dependencies and start the application.

***cd MovieRecommendSystemFrontEnd.zip***
***npm i***
***npm start***

### 2.3 Reverse Proxy

Then, you need to configure the reverse proxy by using Nginx.

Install Nginx.
***sudo apt install nginx nano -y***

Edit Nginx config file.
***sudo vim /etc/nginx/sites-enabled/default***

In the server{} block, add content:

```
location /api/ {
  rewrite ^/b/(.*)$ /$1 break;
  proxy_set_header Host $host;
  proxy_set_header X-Real-IP $remote_addr;
  proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
  proxy_pass http://localhost:8088/;
}

location / {
  rewrite ^/b/(.*)$ /$1 break;
  proxy_set_header Host $host;
  proxy_set_header X-Real-IP $remote_addr;
  proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
  proxy_pass http://localhost:3000/;
}
```

Then save it.
Now visit “localhost”, you can see the page.

# 3. Customize The System

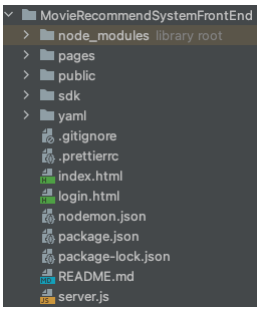
Our system may not meet all your needs. So you can make changes and improvements based on the existing code.

## 3.1 Front End

**Baidu amis**

We use Baidu amis framework to develop the frontend.

Amis is a low-code front-end framework launched by Baidu. It provides rich components and powerful renderers, using JSON configuration to generate pages.




If you want to learn how to use amis, you can refer to the online documentary.

https://baidu.github.io/amis

**Other Possibilities**

Since our system is on the separation of back-end and front-end, you can re-develop the front end, using whatever framework you want, like React, Vue, Angular and so on. You can also develop Android or iOS applications. You only need to do http requests to the APIs with needed parameters or request body, get the response data, and present it to the users.

## 3.2 Back End

We use Java as our back-end development language and SpringBoot Framework. Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run".

We use Maven as our project management and comprehension tool. Dependencies needed are written in the pom.xml files.

You can download the project code, make modifications, and use maven the package it again for deployment.

### 3.3 Data

We have already provided the MongoDB database in a cloud server. If you want to use your own database, you need to change the URL in the "application.properties", and Scala files.