

# USER GUIDE

## MY MOVIES — — INTELLIGENT MOVIES RECOMMANDATION



LU QINWEN

CAI ZIMO

YU HANCHUN

HU ZHIQING

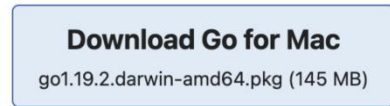
## **User Guide**

USER GUIDE .....	1
1. Requirement .....	3
1.1 Golang server configuration .....	3
1.2 Python server configuration .....	4
1.3 Java server configuration .....	5
1.4 Redis configuration .....	6
1.5 Mysql Configuration .....	7
1.6 Docker configuration .....	9
1.7 Browser requirement .....	10
2. Best Practice .....	11
2.1 User information .....	11
2.2 Functions of system .....	12
3. Interface document .....	17

# 1. Requirement

## 1.1 Golang server configuration

Click the button below to download the Go installer.



Don't see your operating system here? Try one of the [other downloads](#).

**Note:** By default, the go command downloads and authenticates modules using the Go module mirror and Go checksum database run by Google. [Learn more](#).

(1) Open the package file you downloaded and follow the prompts to install Go.

(2) The package installs the Go distribution to `/usr/local/go`. The package should put the `/usr/local/go/bin` directory in your PATH environment variable. You may need to restart any open Terminal sessions for the change to take effect.

(3) Verify that you've installed Go by opening a command prompt and typing the following command:

```
$ go version
```

```

qinwenlu@QinwenluMacBook-Pro: local:~ $ go env
GO111MODULE="on"
GOARCH="arm64"
GOBIN=""
GOCACHE="/Users/qinwenlu/Library/Caches/go-build"
GOENV="/Users/qinwenlu/Library/Application Support/go/env"
GOEXE=""
GOEXPERIMENT=""
GOFLAGS=""
GOHOSTARCH="arm64"
GOHOSTOS="darwin"
GOINSECURE=""
GOMODCACHE="/Users/qinwenlu/go/pkg/mod"
GONOPROXY=""
GONOSUMDB=""
GOOS="darwin"
GOPATH="/Users/qinwenlu/go"
GOPRIVATE=""
GOPROXY="https://goproxy.cn,direct"
GOROOT="/opt/homebrew/Cellar/go/1.17.8/libexec"
GOSUMDB="sum.golang.org"
GOTMPDIR=""
GOTOOLDIR="/opt/homebrew/Cellar/go/1.17.8/libexec/pkg/tool/darwin_arm64"
GOVCS=""
GOVERSION="go1.17.8"
GCCGO="gccgo"
AR="ar"
CC="clang"
CXX="clang++"
CGO_ENABLED="1"
GOMOD="/dev/null"
CGO_CFLAGS="-g -O2"
CGO_CPPFLAGS=""
CGO_CXXFLAGS="-g -O2"
CGO_FFLAGS="-g -O2"
CGO_LDFLAGS="-g -O2"
PKG_CONFIG="pkg-config"
GOGCCFLAGS="-fPIC -arch arm64 -pthread -fno-caret-diagnostics -Qunused-arguments -fmessage-length=0 -fdebug-prefix-map=

```

(4) Confirm that the command prints the installed version of Go.

## 1.2 Python server configuration

For simplifying the configuration process of our system, we used Flask framework and deployed the python-based server for recommendation-related service in Google Colaboratory. The configuration steps are as follows:

(1) Register and sign in google account and upload the project files to the google drive.

我的云端硬盘 > ... > recommendation_server > code			
名称 ↓	所有者	上次修改日期	文件大小
ml-latest-small	我	2022年10月22日 我	—
MF_recommendation	我	2022年10月22日 我	—
measure	我	2022年10月22日 我	—
data_preprocessing	我	2022年10月22日 我	—
data	我	2022年10月22日 我	—
CF_recommendation	我	2022年10月22日 我	—
CB_recommendation	我	2022年10月22日 我	—
main.py	我	2022年10月22日 我	4 KB
main.ipynb	我	15:39 我	21 KB
.DS_Store	我	2022年10月20日 我	6 KB

(2) Open main.ipynb file in google colab.

- ```
main.pynb ☆
文件 修改 视图 插入 代码执行程序 工具 帮助 已保存所有更改

+ 代码 + 文本

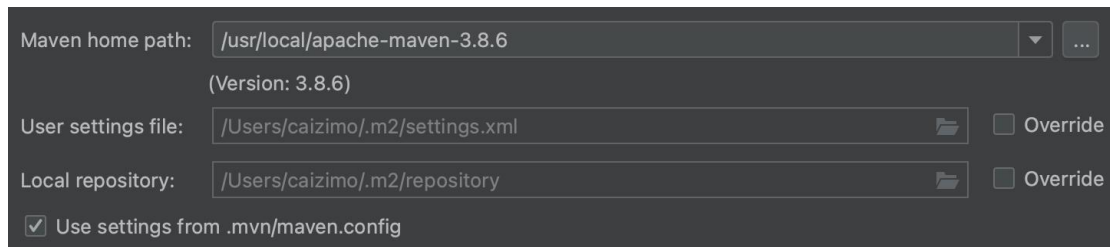
!pip install flask-ngrok flask pyngrok
Ingrid authtoken "2DfVh33cQ2h24OmUvB3tHsKaw_7tiQw7qhWLbyF6YihC5"
!pip install pymysqlSQL

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting flask-ngrok
  Downloading flask_ngrok-0.0.25-py3-none-any.whl (3.1 kB)
Requirement already satisfied: flask in /usr/local/lib/python3.7/dist-packages (1.1.4)
Collecting pyngrok
  Downloading pyngrok-5.1.0.tar.gz (745 kB)
    [REDACTED] | 745 kB 5.1 MB/s
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from flask-ngrok) (2.23.0)
Requirement already satisfied: Jinja2<3.0,>=2.10.1 in /usr/local/lib/python3.7/dist-packages (from flask) (2.11.3)
Requirement already satisfied: Werkzeug<2.0,>=0.15 in /usr/local/lib/python3.7/dist-packages (from flask) (1.0.1)
Requirement already satisfied: click<8.0,>=5.1 in /usr/local/lib/python3.7/dist-packages (from flask) (7.1.2)
Requirement already satisfied: itsdangerous<2.0,>=0.24 in /usr/local/lib/python3.7/dist-packages (from flask) (1.1.0)
Requirement already satisfied: MarkupSafe>0.23 in /usr/local/lib/python3.7/dist-packages (from Jinja2<3.0,>=2.10.1->flask) (2.0.1)
Requirement already satisfied: PyYAML in /usr/local/lib/python3.7/dist-packages (from pyngrok) (6.0)
Requirement already satisfied: charset-normalizer<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>=>flask-ngrok) (3.0.4)
Requirement already satisfied: certifi==2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=>flask-ngrok) (2022.9.24)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=>flask-ngrok) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests>=>flask-ngrok) (1.2)
Building wheels for collected packages: pyngrok
  Building wheel for pyngrok (setup.py) ... done
Created wheel for pyngrok: filename=pyngrok-5.1.0-py3-none-any.whl size=19007 sha256=d9fd6b6065b51ccb30a1fe4435fe4762f64dd94594f8dbce93125delb2
Stored in directory: /root/.cache/pip/wheels/bf/ef/af/ccf6598ecefcd44104069371795cb9b3afbcd16987f6ccfb3
Successfully built pyngrok
Installing collected packages: pyngrok, flask-ngrok
```

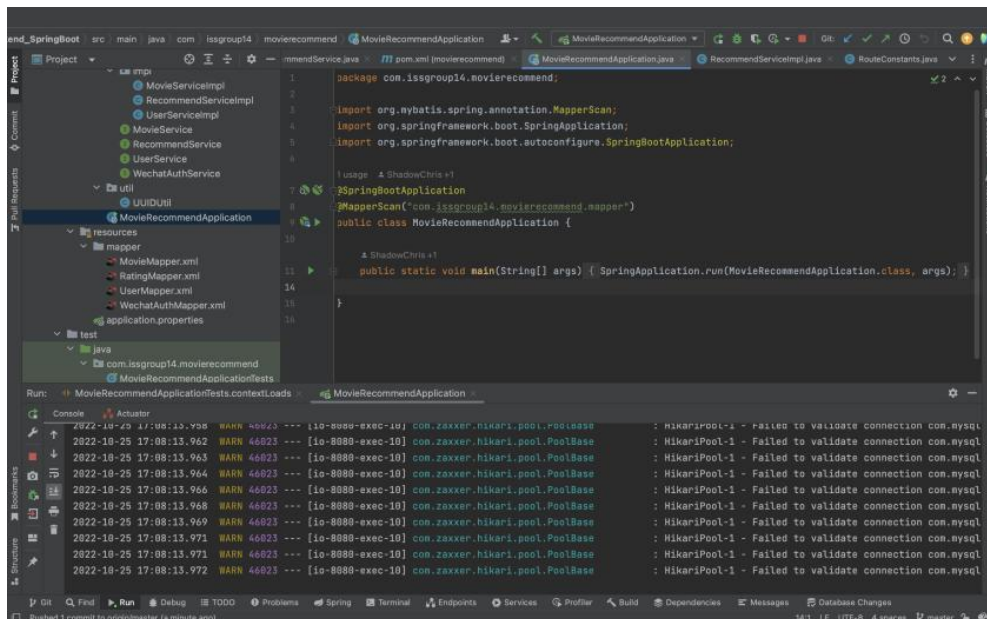
- ```
... * Serving Flask app "__main__" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
INFO:werkzeug: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Running on http://4630-35-196-17-11.ngrok.io
* Traffic stats available on http://127.0.0.1:4040
```

<https://www.oracle.com/sg/java/technologies/javase/jdk11-archive-downloads.html>

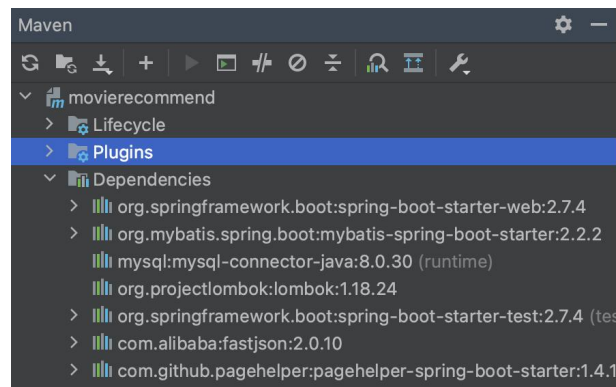
- External Libraries
  - < JavaSE-11 > /Library/Java/JavaVirtualMachines/jdk-11.0.16.1.jdk/Contents/Home
  - Maven: ch.qos.logback:logback-classic:1.2.11



### (3) Import and run the project



### (4) The dependencies are as follows:



## 1.4 Redis configuration

Download address: <https://github.com/tporadowski/redis/releases>

Redis supports 32-bit and 64 bit. This needs to be selected according to the actual situation of your system platform. Here we download

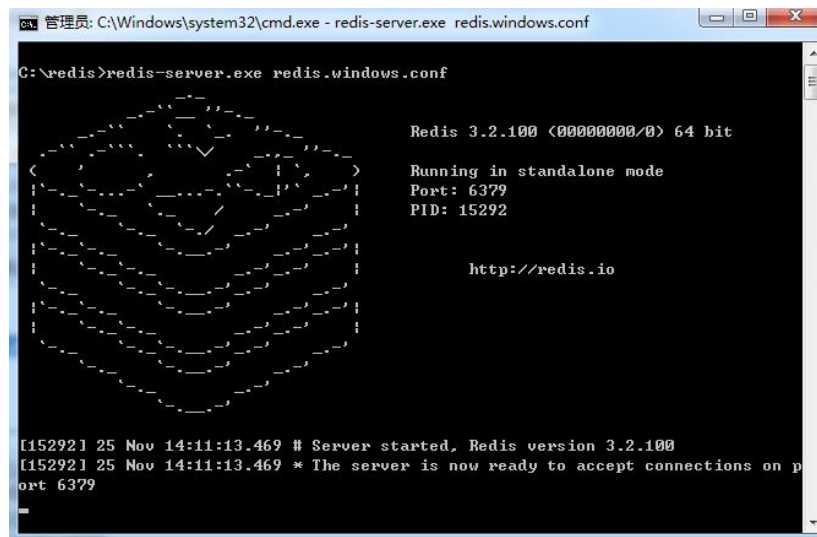
Redis-x64-xxx Zip the package to disk C. After decompression, rename the folder Redis.

Redis supports 32-bit and 64 bit. This needs to be selected according to the actual situation of your system platform. Here we download Redis-x64-xxx Zip the package to disk C. After decompression, rename the folder Redis.

### Downloads

 <a href="#">Redis-x64-3.2.100.msi</a>	5.8 MB
 <a href="#">Redis-x64-3.2.100.zip</a>	4.98 MB
 <a href="#">Source code (zip)</a>	
 <a href="#">Source code (tar.gz)</a>	

Open a cmd window and use the cd command to switch the responding directory to run it:



```
管理员: C:\Windows\system32\cmd.exe - redis-server.exe redis.windows.conf

C:\redis>redis-server.exe redis.windows.conf

Redis 3.2.100 (00000000/0) 64 bit

Running in standalone mode
Port: 6379
PID: 15292

http://redis.io

[15292] 25 Nov 14:11:13.469 # Server started, Redis version 3.2.100
[15292] 25 Nov 14:11:13.469 * The server is now ready to accept connections on port 6379
```

At this time, another cmd window is opened, and the original one should not be closed, otherwise the server will not be accessible.

Switch to the Redis directory to run:

```
redis-cli.exe -h 127.0.0.1 -p 6379
```

## 1.5 Mysql Configuration

First, we use the yum command to install MySQL on Centos7. Note that the



Red Hat Enterprise Linux 7 / Oracle  
Linux 7 (Architecture Independent), RPM  
Package

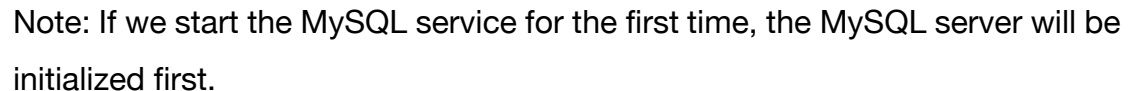
(mysql57-community-release-el7-9.noarch.rpm)

9.0K

Download

MD5: 1a29601dc380ef2c7bc25e2a0e25d31e

You can also use Cloud MySql Server:





## 1.6 Docker configuration



Install cmd :

```
curl -fsSL https://get.docker.com | bash -s docker --mirror Aliyun
```

Update apt package index:

```
$ sudo apt-get update
```

Install the apt dependency package to obtain the warehouse through HTTPS:

```
sudo apt-get install \  
    apt-transport-https \  
    ca-certificates \  
    curl \  
    gnupg-agent \  
    software-properties-common
```

Test whether the Docker is successfully installed. Enter the following instructions and print out the following information. The installation is successful:

```
$ sudo docker run hello-world
```

Unable to find image 'hello-world:latest' locally

latest: Pulling from library/hello-world

1b930d010525:

Pull complete

Digest:

sha256:c3b4ada4687bbaa170745b3e4dd8ac3f194ca95b2d0518b417fb47e5  
879d9b5f

Status: Downloaded newer image for hello-world:latest

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:

<https://hub.docker.com/>

For more examples and ideas, visit:

<https://docs.docker.com/get-started/>

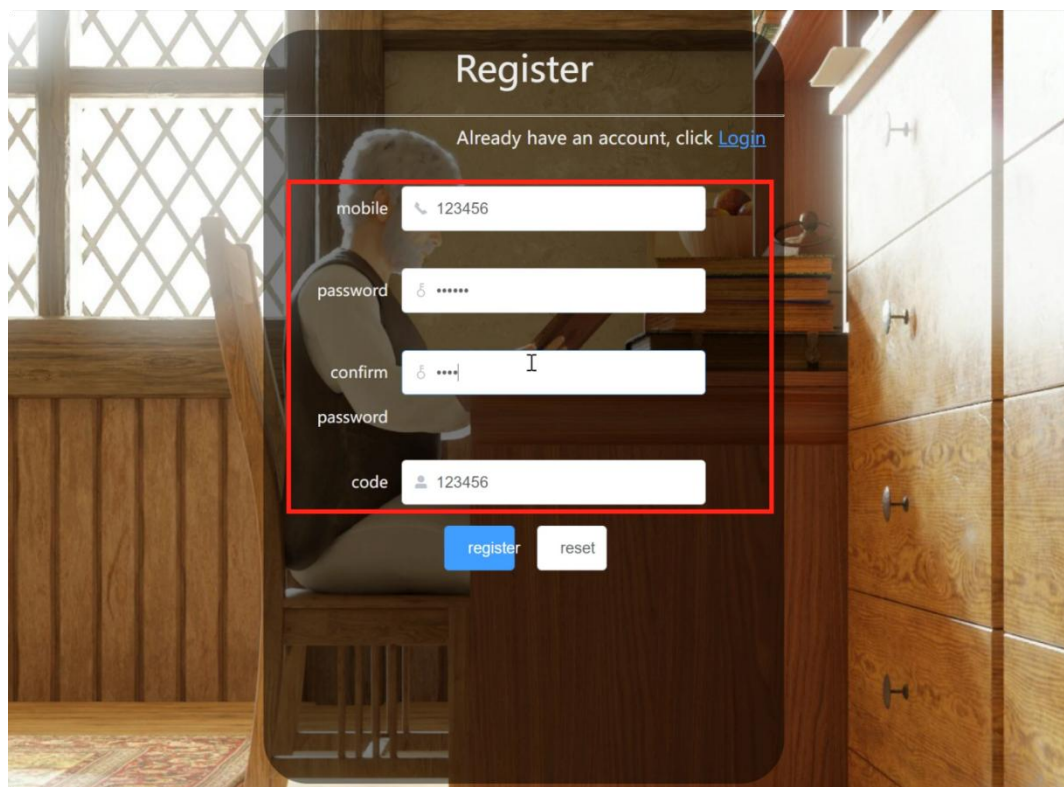
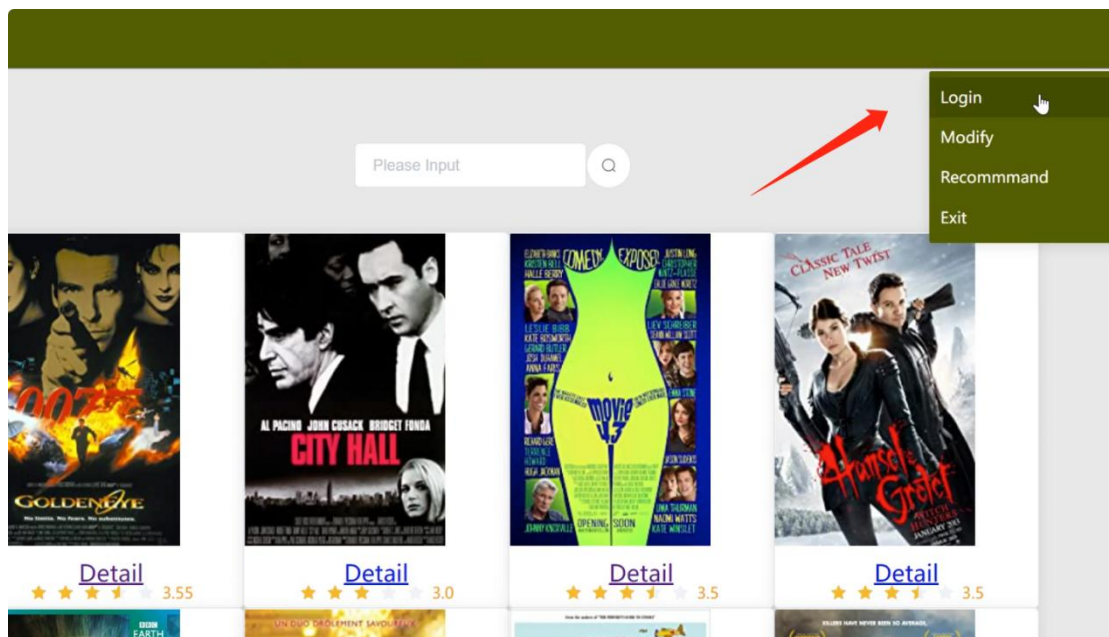
## 1.7 Browser requirement

IE10、IE11、Edge、Firefox、Chrome、safari、opera

## 2. Best Practice

### 2.1 User information

#### 2.1.1 Login/Logout



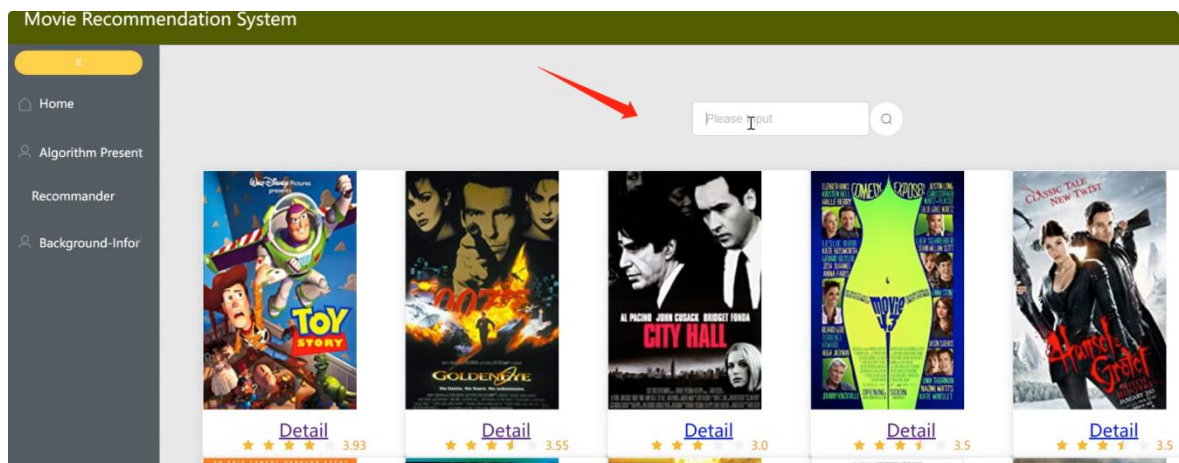
If users need to log in to obtain personalized recommendation results, new users need to complete the registration function before logging in for the first time.

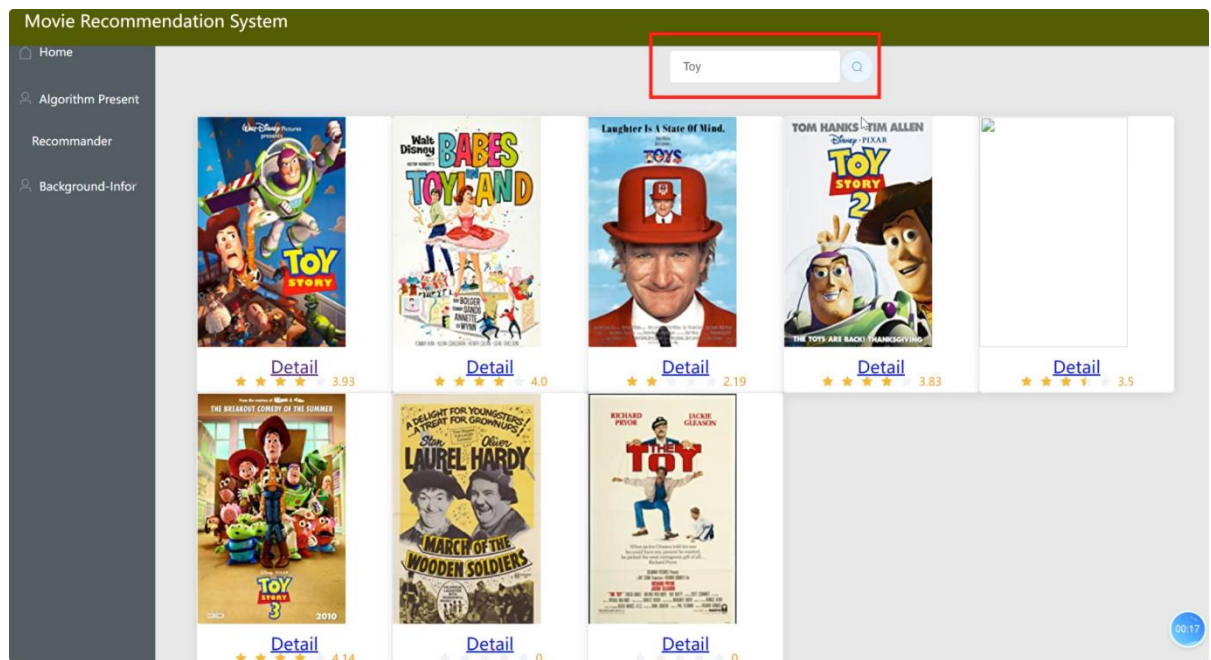
### 2.1.2 Trace user IP

mysql	id	uid	type	ip	ext	ctime
nus	2	2	0	2130706433		16646151
performance_sc...	3	3	0	2130706433		16659182
seckill	4	4	0	3232235885		16659184
ssodb	5	5	0	3232235887		16659186
表	6	6	0	3232235887		16659188
device	7	7	0	3232235887		16659188
movies	8	8	0	3232235887		16659189
ratings	9	9	0	2886994438		16666961
trace	10	10	0	2886994434		16666961
users	11	11	0	2886994434		16666991
视图						

## 2.2 Functions of system

### 2.2.1 Search movies

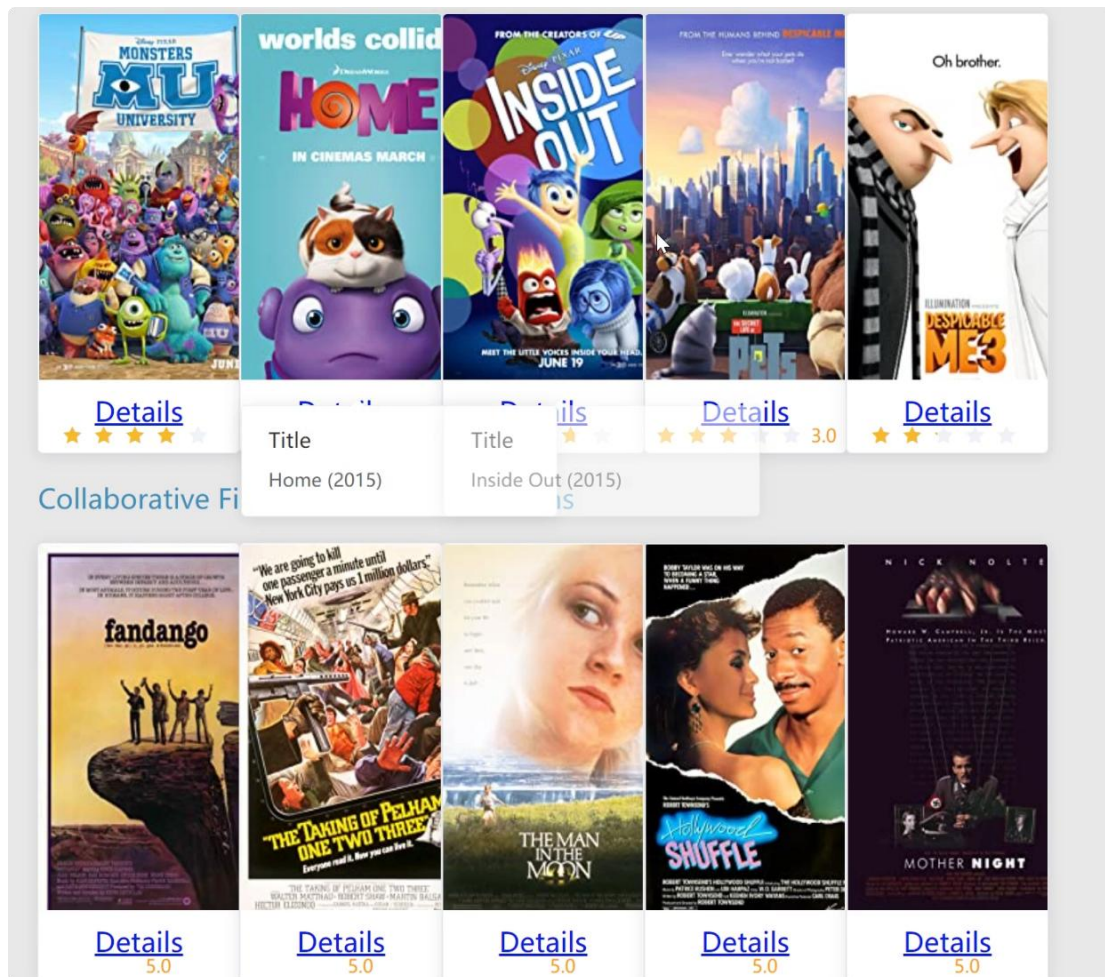




Enter the name of movie that you want to search in the input box to get relevant matching results.



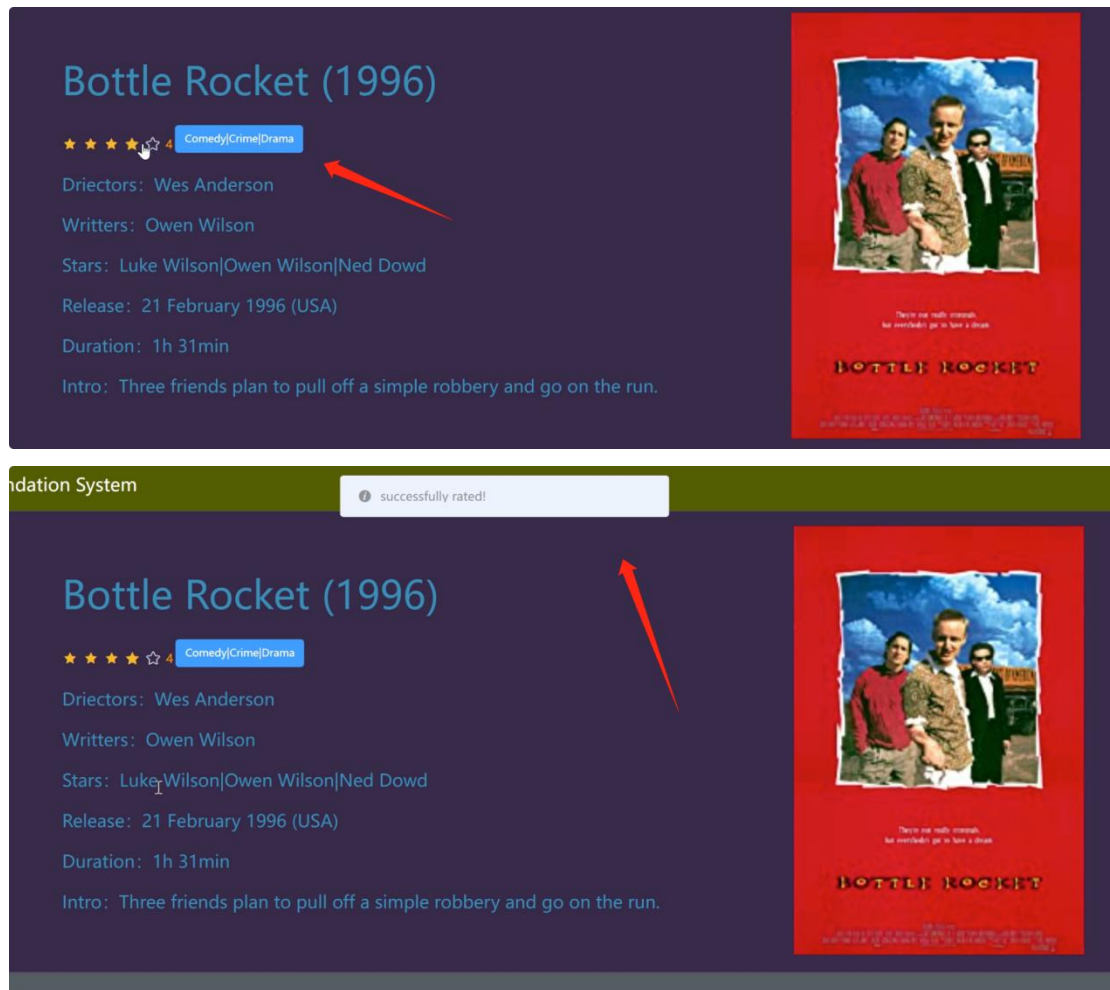
## 2.2.2 Recommend movies



The recommendation system will select and recommend movies that may be of interest to users according to their personalized data.

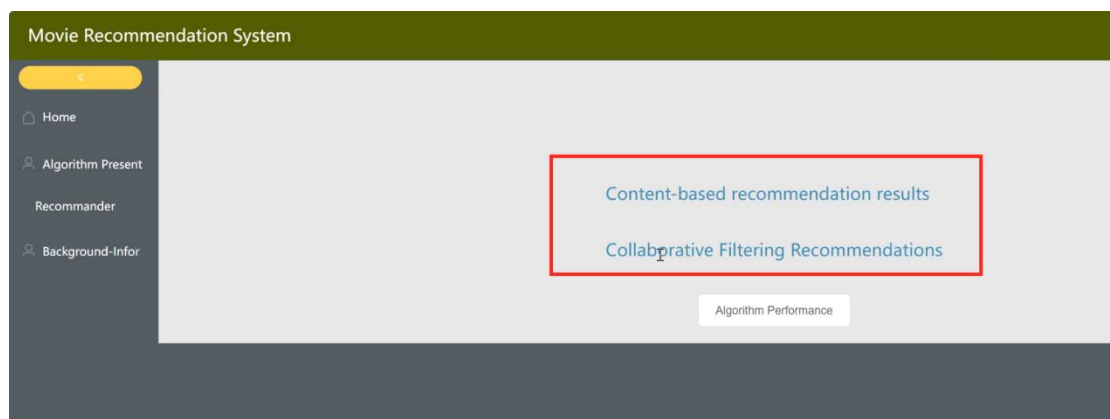
It is based on the recommendation algorithm. We also implement the high-performance TopK algorithm. The result will vary from person to person.

### 2.2.3 Rating a movie



The system will record the users' data based on user's feedback. On the movie details page, users can rate a movie by hovering and dragging the mouse ; After rating successfully, there will be corresponding prompt information

### 2.2.4 Performance of system





Algorithm name	MSE	RMSE
Content-Based Recommendation Algorithms	0.96124462	0.98043083
Collaborative Filtering Recommendation Algorithm	1.255092442	1.12030908
Matrix Factorization Recommendation Algorithm	0.95787807	0.97871245

We also provide the administrator with a visual page to view the performance of the current algorithm.

MSE (Mean Squared Error)

$$\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2$$

RMSE (Root Mean Squared Error)

$$\sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2}$$

### 3. Interface document

Method	URL	parameters	remarks
POST	/signup/mobile	{ "mobile": "1314520", "passwd": "123456", "code": "123456" }	Sign up
POST	/login	{ "mobile": "1314520", "passwd": "123456" }	Log in
POST	/logout	clean token and cookie	Log out
POST	/login/mobile	{ "mobile": "1314520", "passwd": "123456", "code": "123456" }	Log in by mobile phone
POST	/rating/rateAMovie	{ "movieId": "1314520", "rating": "5.0", "userId": "123456" }	Rate a movie
POST	/rating/updateAMovieRating	{ "movieId": "1314520", "rating": "3.0", "userId": "123456" }	Change a rate for a movie

POST	/home/queryMovieList	{ "pageNum": "1", "pageSize": "10" }	Query movie list by page (sorted by movie ID by default) Parameters: Page Several data items on one page
POST	/home/queryMovieList	{ "name": "Toy Story 2 (1999)", "genre": "Animation Adventure Comedy", "releaseTime": "24 November 1999 (USA)", "directors": "John Lasseter Ash Brannon 1 more credit", "writers": "John Lasseter Pete Docter", "stars": "Tom Hanks Tim Allen Joan Cusack",  "pageNum": "1", "pageSize": "10" }	Fuzzy search for a movie according to one of the keywords (movie name/type/release year/director/screenwriter/star), and return the matching results in pages
GET	/movie/{movie_id}	None	Query movie details according to movie ID
POST	/recommend/get-content-based-rec	{ "k": 10, "userId": 10, }	Content-based personalized recommendation, returns k movies

		<pre>         "pageNum": "1",         "pageSize": "10"       } </pre>	recommended by the system under the current user through paging
POST	/recommend/get-collaborative-filtering-rec	<pre> {   "k": 10,   "userId": 10,    "pageNum": "1",   "pageSize": "10" } </pre>	Personalized recommendation based on collaborative filtering returns k movies recommended by the system under the current user through paging
GET	/recommend/preprocess	None	Update the preprocessing data of the recommendation system
GET	/recommend/getPerformance	None	Get the performance score (MSE, RMSE) of the two algorithms

### Flask API Document

Method	URL	Parameters	Remark
GET	/preprocessing	None	Update the preprocessed data of the model
POST	/cb-recommend	<pre> {   "userId": 1,   "k": 10 } </pre>	Get the top10 movie ID list returned by the user with ID 1 through the content based

			recommendation algorithm Return value: list of k movieIds
POST	/cf-recommend	{ "userId":1, "k": 10 }	Get the top10 movie ID list returned by the user with ID 1 through the recommended algorithm of collaborative filtering Return value: list of k movieId
GET	/get-recommend-perfor mance	None	[ { "MSE": 1.3207583528140168, "Name": "CB", "RMSE": 1.1492425126203853 }, { "MSE": 1.567236810564991, "Name": "CF", "RMSE": 1.251893290406571 }]

## 4. Necessary SQL

```
CREATE TABLE `movies` (
  `movieId` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  `title` varchar(255) DEFAULT NULL,
  `genres` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`movieId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE `users` (
  `id` bigint unsigned NOT NULL AUTO_INCREMENT COMMENT '主键',
  `name` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL DEFAULT '' COMMENT '用',
  `email` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL DEFAULT '' COMMENT '电',
  `mobile` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL DEFAULT '' COMMENT '手',
  `passwd` varchar(40) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL COMMENT '密码',
  `salt` char(4) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL COMMENT '盐值',
  `ext` text CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL COMMENT '扩展字段',
  `status` tinyint NOT NULL DEFAULT '0' COMMENT '状态 (0: 未审核, 1: 通过 10删除)',
  `ctime` int unsigned NOT NULL DEFAULT '0' COMMENT '创建时间',
  `mtime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '修改时间',
  PRIMARY KEY (`id`),
  KEY `ctime` (`ctime`)
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE `trace` (
  `id` bigint unsigned NOT NULL AUTO_INCREMENT COMMENT '主键',
  `uid` bigint unsigned NOT NULL DEFAULT '0' COMMENT '用户主键',
  `type` tinyint NOT NULL DEFAULT '0' COMMENT '类型 (0: 注册 1:: 登录 2: 退出 3: 修改 4: 删除)',
  `ip` int unsigned NOT NULL COMMENT 'ip',
  `ext` varchar(1000) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL COMMENT '扩展字段',
  `ctime` int unsigned NOT NULL DEFAULT '0' COMMENT '注册时间',
  PRIMARY KEY (`id`),
  KEY `UT` (`uid`, `type`) USING BTREE
) ENGINE=MyISAM AUTO_INCREMENT=12 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE `ratings` (
  `user_id` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  `movie_id` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  `rating` varchar(255) DEFAULT NULL,
  `timestamp` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`user_id`, `movie_id`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```