Polytechnic University of Cartagena

Higher Technical School of Telecommunications Engineering

PRACTICAL APPLICATIONS ON THE INTERNET

Final project proposal 2020/2021

Web application for video rating and recommending

revision 1.3

Teachers:
Esteban Egea Lopez
Juan José Alcaraz Espín

Index.

1 General considerations.

1.1 goals

In this work the students will develop in pairs an application for movie rating and recommending. The goal is that students become familiar with the web technologies most used today (HTML, CSS, PHP, JavaScript) as well as data processing and analysis algorithms.

1.2 Introduction

You will develop an application which mimics in a simplified manner the operation of applications such as http://www.filmaffinity.com, for example. It is an application that has a catalog of films with various information about it, such as the poster, release date and gender, that any user can view. The application allows users to register. Registered users can rate movies, add comments, and receive a recommendation system.
In order to make the application, we have compiled a catalog of films and scores of users from different sources. The dataset Movielens 10k (http://grouplens.org/datasets/movielens/) was used. It is completed with information extracted IMDB (http://www.imdb.com/) And a list of fictitious names automatically generated. This database information is available to students to implement application on it, as described in later sections.

1.2.1 What students should do

Students must implement a web application paired with functionality specified herein. It should include:

1. PHP programming scripts for the application logic.
2. Any additional file with HTML, CSS or JavaScript required for the application.
3. Matlab programming scripts with data analysis algorithm.

The application must be fully functional and must be testable. To do this, the application will be hosted on a server lab account (labit601.upct.es) and accessed it via the HTTP server lab.

You should deliver a report of the work performed. The contents of memory and the evaluation criteria are also described in this proposal.

1.2.2 How it is organized this proposal.

In section 2, the minimum functionality to implement and optional parts will be described.
In section 3, we provide evaluation criteria for report and material to be delivered.
Finally, in section 4 there are some suggestions on the implementation and use of additional libraries.

2 Application Development

The minimum functionality to be provided by the application is as follows:

1. **Catalogue**. Show and allow browsing the entire catalog of films. To do this show:
    a. Image with movie poster.
    b. Title of the film hyperlinked to a new page that will display the content described in 4 (see below).
    c. Description of the film.
    d. Release date.
    e. Average rating for the movie, number and weighted scores received by the movie.
    f. Should allow reordering of elements based, at least the fields name and average score.
2. **Login**. Show a link or a form on the home page that allows logging to registered users or register a new user.
3. **Registry**. Add a new user. The data that can provide the user are:
    a. Name.
    b. Age.
    c. Sex.
    d. Occupation.
    e. Password.
    f. Profile picture.
4. **Movie**. Show the fields associated with the film (see 1) and:
    a. Genres to which it belongs.
    b. Comments received and the name of the user who wrote it.

    c. Rating received by the user (if he has already rated) if the user has been identified and logged in previously.
    d. Ability to score or change the rating if the user has been identified and logged in previously.
    e. Ability to add a comment if the user has been identified and logged in previously..
5. **Username**. Show the user's personal data and allow to:
    a. **Generate personalized recommendations**. It will run the recommendation algorithm and genereate a recommended rating for each of the films in the catalog score for that user.
    b. **Show the recommendations generated**. Display either all or a subset, ordered by higher score.
    c. Modify personal data.
6. **Format**. You must provide at least a stylesheet to format your application. The format is free but proper use should be considered.
7. **Dynamic content**. You must use JavaScript to generate the content, format or respond to user actions.

To implement this functionality you will be provided a database already filled with the data you need. The database is described in 2.2 and Annex 1. To generate recommendations you must use the collaborative filtering algorithm that you develop in the labs and provide an interface with the Matlab server that will run it, as described in 2.3. The main part of the application is the application logic in PHP, as described in 2.1.
The grade of this work is divided between a mandatory part with minimal functionality (75%), and a free extension (25%).

You can find an example implementation of this application with the functionality required in http://labit601.upct.es/video/to guide you in his work. You can log in with any user from 1 to 900 and the ai04 password.
ATTENTION: This is just an example.

- **No need to copy exactly the structure of the sample application.**
- **No need to mimic the format of the sample application.**
- **No need to use the same techniques in the sample application.**

2.1 Application logic

The application logic.
It will be implemented through PHP scripts that will provide at least the minimum required functionality described in the previous section. Organize functionality into separate scripts and use functions to reuse parts of code that are repeated. You can use require and require_once function to load libraries or other PHP scripts. In particular, consider grouping the common HTML header or footer.

Images
The images used have been compressed into a file available the AV.

SQL query interface.
The logic of the application queries the database using SQL with PHP.
To make the minimum functionality should develop a set of SQL queries at least for:

- Select information from a given user and check your password.
- Insert or modify information from a user.
- Select information from a given film.
- Select, insert or modify a movie score for a given user.
- Select the genre to which a movie belongs.
- Select the comments associated with a movie.
- Add a comment to a movie by a user.
- Select, insert or update the recommendations for a given user.
- Select the scores of all users and all movies.
- Select the information of all the movies with the average score of each film and the number of scores for each film.

Style and javascript.
Develop a style sheet and, if it thinks fit, javascript code to implement the functionality or style.

Weighted score: Bayesian Ranking
You can verify that these applications have the problem of how to add scores to compare a movie with another. For example, what is most valued a film with 356 votes and a 3.8 average or one with 10 votes and average 4.8?

The answer to this question is relatively complex and you can find a very interesting discussion about [1, Chapter 5]. If you read it you will receive an explanation of the weighting method to be used in this project and described below. For a weighted score, the following formula applies each film i:

$$p_i = \frac{NR + n_i r_i}{N + n_i}$$

where N is the total number of films, R is the average score of all movies, nor is the number of scores of the film i and ri is the average score of the film i.

MATERIAL TO BE DELIVERED:

- **PHP scripts, stylesheets, JavaScript and HTML files.** They will be located in your public_html directory under the video directory, so that the application can be tested.
- **Report** of the application logic with a rationale for design decisions. Do not include the full code. Stick to describing how you have organized the logic, where the functionality has been implemented and discuss the design decisions that you think more relevant.

2.2 Design of the database

You have a database with the data required for the application. Every pair will be assigned a user and password to work with it. You can use phpmyadmin on the lab server (http://labit601.upct.es/phpmyadmin/) Or directly use the mysql command to work with your database. If you are going to work locally, you have to export the database from the server and import it to your computer.

The structure of the tables in the database is detailed in Annex 1. It is not established any relationship between tables, which means that although there are obviously fields which refer to fields in other tables, the database will not check referential integrity automatically.

The database is designed from a set of files with data available at the AV, which are provided in case you decide to modify the database provided or design your own. Each line of the file contains the record of a film and each field is separated by a tab ("\ t"). Review them before making the design. The fields of each file are:

- u.movie5: Film identifier, title, release date, IMDB URL, image name and description associated.
- u.user2: user ID, name, age, occupation, password.
- u.genre: name of a film genre and identifier associated with the genre.
- u.moviegenre: identifier of the film and 19 fields worth 1 if the film is labeled within the genus (see u.genre) and 0 otherwise.
- u.data: user identifier, identifier film score (1 to 5) and timestamp in UTC seconds since 1/1/1970.
- imagenes.rar: posters available for movies (not all). The name of the image corresponds to the name of the image in u.movie5 for each film.

2.3 Interface with analysis algorithms application

Analysis algorithms (recommendation) are implemented using Matlab, so that an interface between PHP and Matlab is necessary. In a real application of this style it is very unlikely to work this way, but the algorithms would be reimplemented in a more appropriate language for use in a production environment. In our case, to simplify development and avoid reimplementing, we directly use Matlab.

To do this we will do as follows:

1. Matlab is invoked through a server running in the laboratory. The TCP server will receive requests, to port 1111, from a PHP script that will send the absolute path where the scripts are and the Matlab function to be executed including the required parameters. You do not have to implement this TCP server. It is already running in the laboratory. The server will invoke the script for the Matlab algorithm.
2. The algorithm will obtain the necessary data directly from the database using SQL queries and update the database with the data generated.

More precisely, you have to do the following:

- **A PHP script** to connect via a TCP socket to the server and pass it a:
    - the absolute path of scripts that need to run on Matlab

- and the name of the function you have to run. That is precisely the function that implements collaborative filtering algorithm.

- The Matlab function that implements the algorithm and will contain the modified code of the lab: Recommendation Systems. In that lab, the R and Y and MovieList matrices are read from a local file. For the application, these matrices have to be constructed from data from the DB. Therefore you have to implement a getData () function that will create the matrices R, Y and MovieList from tables from the database. Moreover, in the lab the results are just shown on screen. In the application, the data must be stored in database, by an updateRecommendation () function that must be implemented. These functions therefore run as part of the function code that implements the algorithm of collaborative filtering.

Recommendation server implementation with Java and Matlab

An immediate way to implement this functionality is to use Java code directly on Matlab. Matlab runs a Java virtual machine and allows you to run Java code. That is, you can use Java classes within Matlab scripts. Just be careful and use the proper syntax, which is a mixture of Java and Matlab (seehttp://es.mathworks.com/help/matlab/matlab_external/bringing-java-classes-and-methods-into-matlab-workspace.html). Annex 2 provides an example of implementing a TCP server with Matlab, using java.net classes. You can see how the syntax is a mixture of Java and Matlab.

The server expects to receive:

1. A string with the absolute path to the directory where your Matlab scripts are, finished with a carriage return. For example, in PHP you should establish a socket with the server and send a variable such as $ path = "/ home / students / ai / matlab \ r \ n";
2. A string with the invocation of the function that will execute its algorithm. For example, if the function is declared as filtering (id), that is, it is passed as a parameter a user ID, you must use a variable in PHP such as $ fun = "filtering (". $ Userid. ") \ r\ n ".

Note the "\ r \ n" at the end of the strings to the server, to separate the two fields, since the readline () Java function is used to receive data from the server. Finally, before writing these string on the socket, concatenate with chr (0) to ensure you force the sending and receiving correctly. With the above example, you would have to do

```
$info = $path.$fun.$chr(0);
$sent=socket_write($socket, $info, strlen($info));
```

The TCP server running Matlab listens to port 1111. The server saves a matlab.log file with a summary execution. in the path that is provided for the Matlab scripts (see 1 above). Please note that for the server to generate this log file, the Matlab script folder must have write permissions for others. You can (and should) check that file for debugging.

SUMMARY. MUST BE IMPLEMENTED:

- Matlab scripts with functions:
  - *getData*. Function generating the matrices R, Y and MovieList from tables from the database.
  - *updateRecommendation*. Function that receives a matrix of recommendations for a user and user ID and updates the recommendations table on the database (*recs*).
    VERY IMPORTANT: Matlab distribution we use in the lab, has not licensed the Database toolbox of Matlab. Therefore, YOU CAN NOT USE THE TOOLBOX MATLAB DATABASE to implement the above functions. You must use Java JDBC on Matlab to obtain and update data in the databases Matlab scripts. IF YOU USE THE TOOLBOX DATABASE IT WILL NOT WORK ON THE LABORATORY SERVER.
- A PHP function that connects to the Matlab server at the laboratory and passes the data to run the collaborative filtering algorithm as described in this section.

2.4 Algorithm of collaborative filtering

You should incorporate into the application the code realized in the lab: Recommendation Systems, for collaborative filtering. For the matrices R and Y may use the getData () function above. After training the algorithm and generate scores (using the code done in the lab), you must update the database using the function updateRecommendation above.

2.5 Free development and extension

A percentage of the grade of this project (25%) is allocated to innovation or extension of the application described herein. Therefore, students are encouraged to incorporate additional functionality they deem necessary or attractive for the application.

In grading it will be taken into account:

- Extensions functionality,
- Any variations of the recommendation algorithm or comparative experiments.

Some ideas to extend the application are as follows:

- For example, note that users in the BBDD are identified by ID, no name or email. An obvious extension is to allow identification by another one of those options.
- You can allow users to assign tags or labels to movies (both from a predefined list, as their own tags). These tags can be used to search or to establish similarity between them. The tag does not have to match the genre.
- You can use the API (https://developers.themoviedb.org/3/getting-started) The Movie Database (https://www.themoviedb.org/) To add additional functionality. For example, you can display a background picture film related to a user query or complete movie information.
- You can allow the user to create custom lists: movie you liked, those that have outstanding view or interest you, etc.
- You can do all this interactively, using javascript, for example allowing the user to drag the photo from a movie to your list to add to it.
- You can sow similar movies when the user checks a movie page or similar users when she checks her profile.
- You can set the regularization parameter. You can ask the user to rate their recommendation and use these scores to select the regularization parameter and check whether this method is effective. You can check whether recommendations vary by changing scores.

3 Deliverables and evaluation criteria

Students must submit a brief report on the work, in addition to hosting the application code in the lab, so that it is accessible by the HTTP server. Delivery and evaluation will take place in January. You will be informed well in advance of the deadline. Reports will be uploaded as a deliverable to the virtual classroom, per duplicate, ie, each group member must upload his own copy of the report. No report after the deadline date will be accepted. It is possible that in some cases teachers need to meet with students in a group to evaluate their work. For such cases ,a list of groups that should meet with teachers to explain their work will be published.

3.1 Report and Code

The report must contain:

- full name of the group members, email and at least one telephone contact.
- Index
- URL access to the application.
- Memory described in section 2.1, describing the implementation of the application.
- A list of all deployed PHP scripts, describing the functionality they implement.
- A list of all the additional files you've used (CSS, javascript), describing the functionality they implement.
- If you used a development framework for PHP, CSS or Javascript, a description of the use it makes.
- A description of any additional functionality implemented.

The code. It will be in the public_html directory under a *video* directory to be accessible to the laboratory HTTP server, and should be fully functional, ie, teacher muste be able to test the application with a browser.

3.2 Evaluation criteria

Although it is mentioned above here we summarize again the evaluation criteria. The total grade of the work has been divided between the various parts:

Work	Maximum grade
Minimal functionality	7.5
Free extensions	2.5

For the evaluation, it will be taken into account:

- Students have correctly implemented the necessary functionality.
- The interpretation of the results of the recommendation algorithm will be valued and appreciated especially is students demonstrate having consulted literature, Internet, etc., to improve its conclusions.
- Clarity, rigor and conciseness in the explanations given in the report.
- Clarity and order in the code.
- It is appreciated that the functionality has been separated in files and functions properly.
- Using of CSS styles. Although the aesthetic quality of the application itself is subjective and is not evaluated as such if the proper use of stylesheets and associated techniques will be evaluated.

Extensions of functionality and / or proposals for other algorithms and study the influence of various parameters on performance will be positively graded. The contributions are valued for their originality, quality, and arguments, not by their quantity or their length. These contributions may be useful to improve the final grade of the score.

3.3 Single person groups

At the beginning of this proposal it indicates that the work must be done in pairs, but sometimes circumstances prevent building or maintaining a group and only one person stays. Since it is not fair to apply the same evaluation criteria in these cases the following distribution of points shall apply:

Single person groups:

Work	Maximum grade
Minimal functionality	8.5
Free extensions	1.5

4 Additional notes for implementation

4.1 Implementation of logic in PHP

- It is recommended to implement logic incrementally by functional units. For example, you can start doing the / scripts that allow viewing and browsed the catalog of films. Then the / scripts that allow you to view the details of a movie and discuss it. And so on.
- You can leave the format and style of your application to the end, but it is recommended that before implementing decide at least approximate visual layout or organization. That way you can group the elements into blocks and make a common headers and feet to be displayed on all pages. For example, you may decide that your application will only contain a block header and other content.

4.2 Alternative design database

- If you believe the design of the database that is provided is inadequate, it is free to use an alternative design or add tables as necessary. You should briefly describe and justify their alternative design in memory.

4.3 Using additional libraries and frameworks

- If the student prefers, the development of application logic using development frameworks are allowed, such as CodeIgniter, Zend, CakePHP or as long as they are based on PHP. THE USE IS SUBJECT TO APPROVAL BY TEACHERS. You should send an email to Professor Esteban Egea indicating the framework or libraries intended to be used. Note that in some cases their use can be counterproductive. Note also that some

frameworks provide specific mechanisms for data persistence, so shall describe in memory also the structure of the database in that case.

- Installation, configuration and operation and use form of the framework and a description of how they implemented the application logic in the context of framework: If a framework is used the report should include a description of use.
- The use of libraries for style and format also allows, as can be jQuery or other, but their use should be documented in the report.

5 Bibliography

[1] Mung Chiang. "Networked Life. 20 Questions and Answers ", Cambridge University Press, Cambridge, UK, 2012

6 Annex 1. Structure database

Then the tables used in the database are described. Fields in bold act as the primary key of the table. Recs USER_SCORE two tables and fields that act as primary key, ie, that in these columns may appear repeated values but the combination of both values always have to be unique in the table.

6.1 Genre table

Stores the movie genre.

Column	Type	Null	Default
name	text	No	
id	tinyint(3)	No	

6.2 table movie

Stores information about each film.

Column	Type	Null	Default
id	int(10)	No	
title	text	No	
date	date	Yes	NULL
url_imdb	text	No	
url_pic	text	Yes	NULL
desc	text	Yes	NULL

6.3 moviecomments table

Stores the comments made movies.

Column	Type	Null	Default
movie_id	int(11)	No	
user_id	int(11)	No	
comment	text	No	

6.4 moviegenre table

Stores each film genres

Column	Type	Null	Default
movie_id	int(10)	No	
genre	tinyint(3)	No	0

6.5 table recs

Stores recommendations for each user, along with the date on which the recommendation was created

Column	Type	Null	Default
user_id	int(11)	No	
movie_id	int(11)	No	
rec_score	double	No	
time	timestamp	No	CURRENT_TIMESTAMP

6.6 table users

Stores information for each user. WARNING: passwd field stores SHA1 hash of the user's password.

Column	Type	Null	Default
id	int(10)	No	
name	text	No	
edad	tinyint(3)	Yes	NULL
sex	enum('M', 'F')	Yes	NULL
ocupacion	enum('administrator', 'artist', 'doctor', 'educator', 'engineer', 'entertainment', 'executive', 'healthcare', 'homemaker', 'lawyer', 'librarian', 'marketing', 'none', 'other', 'programmer', 'retired', 'salesman', 'scientist', 'student', 'technician', 'writer')	Yes	NULL
pic	text	Yes	NULL
passwd	varchar(40)	No	

6.7 USER_SCORE table

Stores user ratings to movies.

Column	Type	Null	Default
id_user	int(11)	No	
id_movie	int(11)	No	
score	tinyint(3)	Yes	NULL
time	timestamp	Yes	NULL

7 Annex 2. Using Java in Matlab

There is provided a server code as an example of using Java in Matlab. You can also use this script from your own Matlab for local testing, ie, without using the lab server.

```
import java.net. *;
import java.io. *;

ServerSocket = ServerSocket (4450);

display ( 'Starting server');

try
 while (1)
 socket = serverSocket.accept ();
```

```
 display ( 'new incoming connection');
       in = BufferedReader (InputStreamReader (socket.getInputStream ()));
% Read the user path
       pathstr = in.readLine ();
% Read the user function run

 funcstr = in.readLine ();
       PrintWriter out = (BufferedWriter (OutputStreamWriter
(socket.getOutputStream ())), true);
 % Code that satisfy the request
 % Establish the path of the user with the value tells the client
 userpath (char (pathstr));
            % We evaluate the function that tells us the customer
              status = eval (char (funcstr));
 % Return a status value and close
             out.println (strcat (int2str (status), char (0)));
 out.flush ();
 socket.shutdownOutput ();
 % We restore the path
 userpath ( 'reset');

 display ( 'closing connection with client');
 Socket.close ()
 end
catch and
 e.Message
 if (isa (e, 'matlab.exception.JavaException'))
 ex = e.ExceptionObject;
 ex.printStackTrace;
 end
 display ( 'exception')
 Socket.close ();
 serverSocket.close ();
end
serverSocket.close ();
```

Note that in the above example uses the syntax Matlab to create the ServerSocket (the new operator does not appear in the constructor call) and the while loop and has used the Java syntax to invoke methods of the Java object, the ServerSocket . Notice how Java String objects are converted to char string using Matlab. Finally note that due to limitations in the use of Java on Matlab, the server does not allow concurrent requests address.