ΠΛΗ-518 Services in Cloud and Fog Computing

Winter Semester 2020-2021

1st PROGRAMMING EXERCISE:

WEB APPLICATION DEVELOPMENT FOR DATABASE MANAGEMENT SYSTEM

Assistants: Tsakos Konstantinos (email: ktsakos@isc.tuc.gr), Tzavaras Emilios (email: atzavaras@isc.tuc.gr)

Professor: Petrakis Evripidis

EXERCISE DESCRIPTION

The aim of the exercise is to become familiar with Web technologies for the development of a Web implementation. For this exercise will be used some of the most common and widely used technologies for web application development such as HTML, CSS, JS, PHP, MYSQL.

1. IMPLEMENTATION SCENARIO

The application is a system for registering and informing users about movies on cinema.

Users can register themselves in the application by filling out a form in which they also fill the role they want to play in it (account type).

Users of the application can have one of the following 3 roles:

Admins are the only ones who can manage the user's table. More specifically, they can accept and delete users and also update the role and the information of existing ones.

CinemaOwners owns movie theaters. They can add, update and delete movies related to their own cinemas. Make sure they see and have the privileges to manage only the information that concern them.

Finally, the ordinary **Users** have the ability to search for movies based on specifics criteria (Date, Category, Title) and add their favorites movies to Favorites. They will of course be able to see their own Favorites list at any time.

The implementation of the application requires the creation of 3 tables in the MySql database:

-**Table: Users**

-ID: string

-NAME: string

-SURNAME: string

-USERNAME: string

-PASSWORD: string

-EMAIL: string

-ROLE: ENUM [“ADMIN”, ”CINEMAOWNER”, ”USER”]

-CONFIRMED: BOOLEAN

-**Table: Movies**

-ID: string

-TITLE: string

-STARTDATE: date

-ENDDATE: date

-CINEMANAME: string

-CATEGORY: string

-**Table: Favorites**

-ID: string

-USERID: string

-MOVIEID: string

-Table: Cinemas

-ID: string

-OWNER: string

-NAME:string

The basic functions that the system must have and you have to build are:

2.1 Login

The first page of the index.php application should be authenticated user. The user must enter the username & password in a form. Pressing the "login" button should check the correctness of the data from the database and if it is correct allow him to enter the next page otherwise a message appears for an incorrect username or password. Be careful not allow access to any page without authentication even if one user is trying to access the url.

In such cases, you have to redirect them to the index.php page. Also after successful login, each page of the application should be accessible only to users relevant to. For example, only movie owners will be able to access the page that you can add a movie. Also Admins will only be able to access the page that you can update the role of users or deleting them and so on.

2.2 Sign up

The first page of the index.php application should also be enabled via a special button for registration of a user in the system. At the touch of a button another special form should be displayed by filling in details (ID, name, surname, username, password, email, role) and press a special button to enter the user and the original login form reappears. The login form should reappears when a registration with a special button is canceled. Let it be done first phase with redirection to another page each time (signup.php and index.php). In order for o user to enter the application after registration, must be done first accepted by the Administrator.

2.3 Είσοδος στο σύστημα

2.3.1 welcome.php

After the successful login of the user, the page will be redirected welcome.php of the application (for all user roles). There will be the possibility for logout by redirecting to index.php and the user name and the role (in parentheses) at the top right of the page. Through a suitable menu (bar on the left or at the top of the page), the user will be able to access every page of the application. The menu should appears only when the user places the mouse cursor over specific area.Επιπλέον, από το μενού ο χρήστης θα μπορεί να μεταβεί στις σελίδες movies.php (για τους Users), owner.php (για τους CinemaOwners) και administration.php (για τους Admins). In case the role of a user does not allow access to some of them above pages, should go to a special page where it will be updated to deny access and there will be a special button to return to the original page.

2.3.2 movies.php

View all movies and search by cinema (username owner), category, release date or title. Ability to add any movie to favorites. Link to access the favorite page from which will eventually be deleted if the user so wishes.

2.3.3 owner.php

View all movie entries and add, delete and update movie details. Make sure each user can see and manage only his own movies.

2.3.4 administration.php

Display all users in table format and ability to delete and update of their data in the database.

NOTE:

●Make the application enjoyable and easy to use (images, easy to use menus, fonts, colors, input data validation forms, etc.).

● Assume that there is an initial Admin on the system (registered at initialization of the application).

● You can create additional pages of the application where it suits you serves as long as the right of access to them is controlled according to the role users and prevent access to them if they have not previously authenticate users to the system (login procedure).

● The implementation of the exercise requires the installation and use of Apache Server & MySQL Database (useful link for all in one solution: https://www.apachefriends.org/index.html) . You are free to use any Web Framework that suits you both in the creation of the graphical environment (frontend) and in the implementation of the logic of the application (backend).

DELIVERED:

● Code scripts with explanatory comments

●Half page report with remarks about your work, technologies and frameworks used and possible application flaws that do not implemented or partially implemented.

●Sql file (backup file of mysql database)

HAPPY CODING!!