# ТЕМА: Drop-out студенти

## 1. Условие

Условието на текущия проект е да се създаде инструмент за вкарване на сурови данни и визуализация за студенти, с опастност от изпадане

## 2. Въведение

Важно е лекторът да знае кои са студентите с опасност от изпадане в даден курс. Ако реши може да им предложи допъ лнителни задания, чрез които да повишат успеха си . Също така е нужно студентът да бъде наясно какви са резултатите по различните компоненти от дадения предмет и дали той е е с опасност от изпадане от курса.

## 3. Теория

В разработения проект студнетите с опасност от изпадане са тези, които имат средно аритметично на домашните по-малко от 50% или резултат на ревюто по-малкък то 30%.

Ако потребителят е лекторът, то след въвеждане на факултетен моер и парола, той има опцията да види всички рискови студенти или да въведе факултетен номер на стуеднта, чийто данни иска да види.

Ако потребителят е студент, той може да види единствено информация за себе си след въвеждането на правилен факултетен номер и парола.

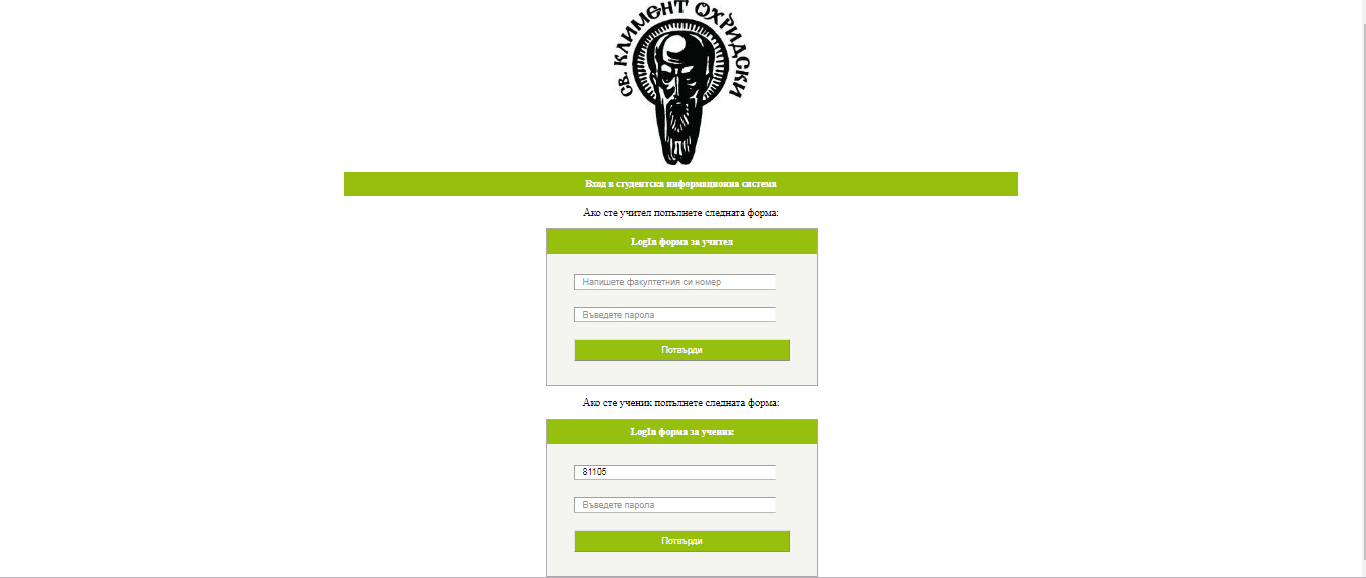
## 4. Използвани технологии

За реализацията на проекта са използвани програмните езици HTML5, PHP(7.2.10), CSS 3, и MySQL. Базата данни и сървърът са предоставени от phpMyAdmin. Добавена е допълнителна функционалност за добавяне на студенти или добавяне на информация за студентите с цел разширяване на базата данни. Тази функционалност е реализирана, чрез използване на HTML за визуализация на потребителския интерфейс, PHP за изпращане на информацията до базата данни и MySQL за модифициране на базата данни.

## 5. Инсталация и настройки

Нужно е потребителя да има инсталиран phpMyAdmin. При първоначално пускане на проекта, трябва да се добавят примерни данни в базата чрез изпълнението на addDatabase.php скрипта. Това ще доведе до създаването на базата данни и таблицата, ако вече не съществуват.

## 6. Кратко ръководство на потребителя



Фиг.1- „Изглед на формите използвани за влизане в системата.”

В зависимост от това дали потребителят е лектора на курса или студент в него той трябва да избере коя форма да попълни. Ако е лектор попълва първата форма с факултетен номер – 0 и парола – teach12. Ако потребителят е студент, то той трябва да знае факултетния номер и паролата си. Само студенти, които са записани в базата данни могат да имат достъп до информационната система.

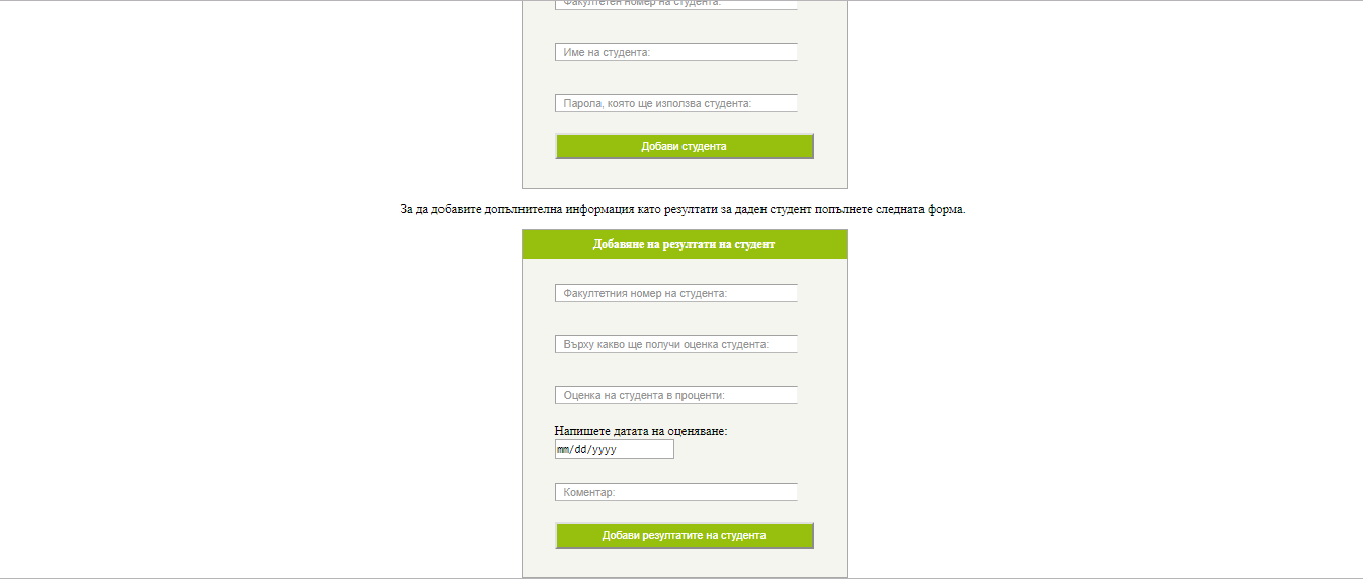


Фиг. 2 – „Изглед на резултата след влизането на студент в системата.”

Ако даден студент е въвел коректно факултетния си номер и паролата то той ще види резултатите от предадените и оценени досега домашни и ревю, както и дали е с опасност от отпадане спрямо зададените критерии.



Фиг. 3– „Изглед на формите след влизането на лектор в системата.”



Фиг. 4– „Изглед на формите след влизането на лектор в системата.”

След като лекторът влезе в системата той има няколко опции. Една от тях е да види информация за това кои са студентите с опасност от изпадане и какви са техните резултати чрез натискане на бутона потърси. Друга опция е да напише факултетния номер на конкретен студент, чийто резултати иска да види и дали той е с опасност от отпадане. Също така лекторът има възможността ако желае да добави нов студент чрез формата “Добавяне на студент” или да добави информация за даден студент чрез формата “Добавяне на резултати на студент”.



Фиг. 4– „Изглед на страницата след като лекторът е избрал да види всички студенти с опасност от отпадане.”



Фиг. 5– „Изглед на страницата след като лекторът е избрал да види информация за конкретен студент.”

## 7. Примерни данни

Примерните данни за студентите се намират в web.sql файла. Също така ги има и в addDatabase.php. При изпълнение на този скрипт примерните данни ще бъдат добавени в базата данни и след това могат да бъдат прегледани и използвани.

## 8. Описание на програмния код

Ако решим да достъпим примерните данни чрез addDatabase.php , първо се проверява дали базата данни съществува и ако това не е така, се създава. След това се проверява дали таблиците с информация съществуват и отново, ако не е така се създават. Чрез connection.php се свързваме с базата данни.

В startingPage.php (файла, който отваряме първо) в HTML кода имаме две форми с възможност за въвеждане на парола и число – факултетен номер. В php кода се проверява дали въведената информация съществува в базата данни и дали е правилна – ако не е се появява съобщение за грешка.

Ако потребителят е студент се създава обект от класа DBTable във файла database\_table.php и се вика функцията show\_info($fn) Тази функция има за цел да визуализира информацията за дадения студент и да определи дали той е с опасност от изпадане и заради кой от компонентите е с опасност от изпадане.

Ако потребителят е лектор той бива пренасочен към index.php. В HTML кода имаме три форми. В зависимост от това дали лекторът натисне бутона ‘потърси’ с попълнен или с непопълнен факултетен номер той бива пренасочен към show\_dr\_students.php, където се създава обект от класа DBTable във файла database\_table.php и се вика функцията show\_info($fn) или съответно dr\_students\_course(). Вече обяснихме какво прави функцията show\_info($fn). dr\_students\_course()има за цел да визуализира информацията за студентите с опасност от изпадане Тя пази факултетните номера заедно със средно аритметичния резултат от домашните на студентите с тези факултетни номера в асоциативен масив $students\_to\_be\_dropped\_out.

Чрез другите две форми лекторът може да добави нов студент или да добави информация за даден студент –създава се нов обект от класа addStudent и се извикват функциите add($fn, $name, $psw) или съответно addInfo($fn, $req\_sc, $score, $score\_date, $notes), чрез които се добавя нова информация в таблиците.

## 9. Приноси на студента, ограничения и възможности за бъдещо разширение

Възможностите за бъдещо разширение са доста. Могат да се добавят допълнителни компоненти спрямо, които да се решавадали студентът е с опасност от отпадане, като присъствие и наличие например на проект. Може да се добави функционалност, която да позволява на студентите да се регистрират в системата. Също така може да се подобри визуализацията на информацията за студентите.

## 10. Какво научих

Работейки по този проект, на първо място си усъвършенствах знанията по WEB програмиране. Подобрих разбиранията си за това как да достъпвам база данни и да работя с получената информация. Научих колко е удобно различните логически части на архитектурата да са добре структурирани и подредени, за лесен достъп и ориентиране.

## 11. Използвани източници

<https://www.w3schools.com/>

<https://www.afterhoursprogramming.com/tutorial/php/php-overview/>

Изготвил: Елена Радойчева