

Имена: Елена Георгиева фн: 81555
Начална година: 2016 Програма: бакалавър, (КН) Курс: 4
Тема: Генериране на разписание (Scheduler)
Дата: 2020-02-10 Предмет: wwwTech2019_20_13ed_KN_winter
имейл: elenageorgieva10@yahoo.com

преподавател: доц. д-р Милен Петров

Предаване: Задачата се предава в архив с попълнен настоящия документ, проекта/проектите с кодовете, README.txt файл, който описва съдържанието на архива; папка с допълнителни компоненти и използвани ресурси, **архива да се казва 81555_project_final.zip. (Успех!).** (Редактирайте **маркираните зони** в жълто с коректната информация)

ТЕМА: Генериране на разписание (Scheduler)

1. Условие

Тема: Генериране на разписание

Проектът реализира:

- Цялостно разписание на презентациите по “WEB технологии”
- Лично разписание – според проявен ‘интерес’ – трябва да отида, мисля да отида, интересно ми е, може да е интересно
- Разпечатване на разписанието като CSV файл, който лесно може да бъде преобразуван от потребителя до различни формати като Excel или PDF.

2. Въведение

Уеб приложението предоставя на потребителите следните услуги: възможност за регистрация и създаване на account (Sign up), влизане с данните от акаунта (Log in), при което се създава сесия на потребителя, преглед на цялостното разписание на всички презентации (тема, презентатор, дата и час), възможност за задаване на предпочитания (No value, Interesting, Maybe going, Going) и съставяне на лично разписание според тях, преглед на личното разписание, актуализация на личното разписание, разпечатване на разписанието, изход и прекратяване на сесията на потребителя (Log out).

3. Теория

Сайтът съдържа две различни менюта – за нерегистрирани потребители или невлезли с данните от акаунта си и потребители, влезли с данните от акаунта си (logged in). Първото меню съдържа линковете Sign up и Log in, отвеждащи съответно до страниците **signup.php**, съдържаща форма за регистрация, и **login.php**, съдържаща форма за влизане и създаване на сесия. Щом потребителят влезе с данните си за регистрация, се създава сесия (инициализират се сесийните променливи `$_SESSION['user']` и `$_SESSION['pass']`), която се поддържа и при следващи влизания в сайта от същия браузър, докато потребителят не натисне Log out, което води до прекратяване на сесията му. Докато сесията е активна, името на потребителя се изписва в скоби след името на приложението – например Scheduler (elena), докато ако потребителят не е регистриран или няма сесия, се изписва Scheduler (Guest).

В случай, че потребителят е регистриран и влязъл в сайта, **header.php** изобразява второто меню. Изборът кое от двете менюта да бъде изобразено се прави въз основа на стойността на булевата променлива \$loggedin. Второто меню съдържа линковете Schedule, My schedule и Print съответно към страниците **table.php**, **myTable.php**, **print.php**.

table.php предоставя информация от базата данни за цялостното разписание на презентациите, текущите предпочитания на потребителя, както и възможност за динамичното им актуализиране чрез AJAX calls.

myTable.php предоставя информация само за презентациите, към които потребителят има интерес (лично разписание) – т.е отбелязал е предпочитания, различна от No value.

print.php разпечатва/export-ва разписанието на презентациите в CSV формат и известява потребителя в коя директория се намира генерираният .csv файл.

4. Използвани технологии

- ❖ PHP – за реализиране на бизнес логиката на приложението, динамично генериране на HTML съдържание и осъществяване на връзката с базата данни
- ❖ HTML, CSS, JavaScript – за реализиране на презентационния слой на приложението
- ❖ jQuery, AJAX – за динамично актуализиране на предпочитанията на потребителя в базата данни
- ❖ MySQL – Database Management System
- ❖ XAMPP - софтуерен пакет, предоставящ PHP interpreter, Apache HTTP Server, MySQL база от данни

5. Инсталация и настройки

Всички файлове на приложението се намират в поддиректория scheduler\ на директорията C:\xampp\htdocs\ . Преди първото стартиране на приложението в браузъра, трябва да се създаде базата данни **scheduler**, която то използва. В Command Prompt на Windows въвеждаме командите:

```
> C:\xampp\mysql\bin\mysql -u root
```

```
> CREATE DATABASE scheduler;
```

```
> USE scheduler;
```

След изпълнението на тези команди еднократно се извиква в браузъра конфигурационният файл **config.php**, който създава таблиците members, presentations и preferences (<http://localhost/scheduler/config.php>). Релацията members съдържа всички регистрирани потребители и паролите им, presentations съдържа информацията за презентациите (номер, тема, презентиращ) и разписанието им (дата и час), а в preferences автоматично се записва информацията за потребител, презентация и 'интерес' (preference) всеки път, когато потребител отбележи конкретно предпочитание

(Interesting, Maybe, Going) за някои презентация на страницата с разписанието на всички презентации.

След създаването на базата и таблиците в нея се попълват данните в таблицата presentations, с които приложението ще борави при стартиране в брауъра.

Описаните стъпки се изпълняват еднократно, например от сървърния администратор.

6. Кратко ръководство на потребителя

Потребителят стартира приложението в брауъра, въвеждайки URL <http://localhost/scheduler/>, който го отвежда на началната страница (home page) на приложението index.php. Там е необходимо потребителят да се регистрира в системата, последвайки линка **Sign up**, след което да влезе в системата с входните си данни от регистрацията (**Log in**). След влизането си в системата потребителят може да разглежда разписанието на всички презентации (**Schedule**), да избира презентации чрез предпоченции и така да състави свое собствено разписание според интересите си (**My schedule**), да разпечата разписанието на презентациите (**Print**) или да излезе от системата (**Log out**).

7. Screenshots от работата на приложението с примерни данни

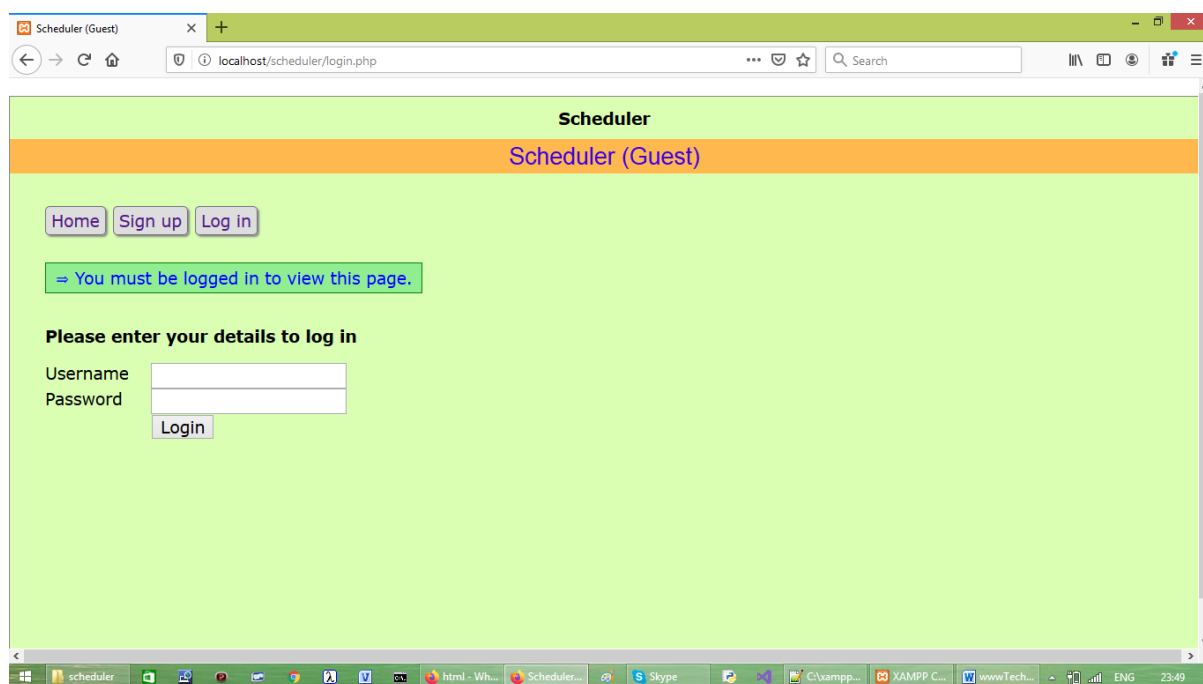


Figure 1 - Log in

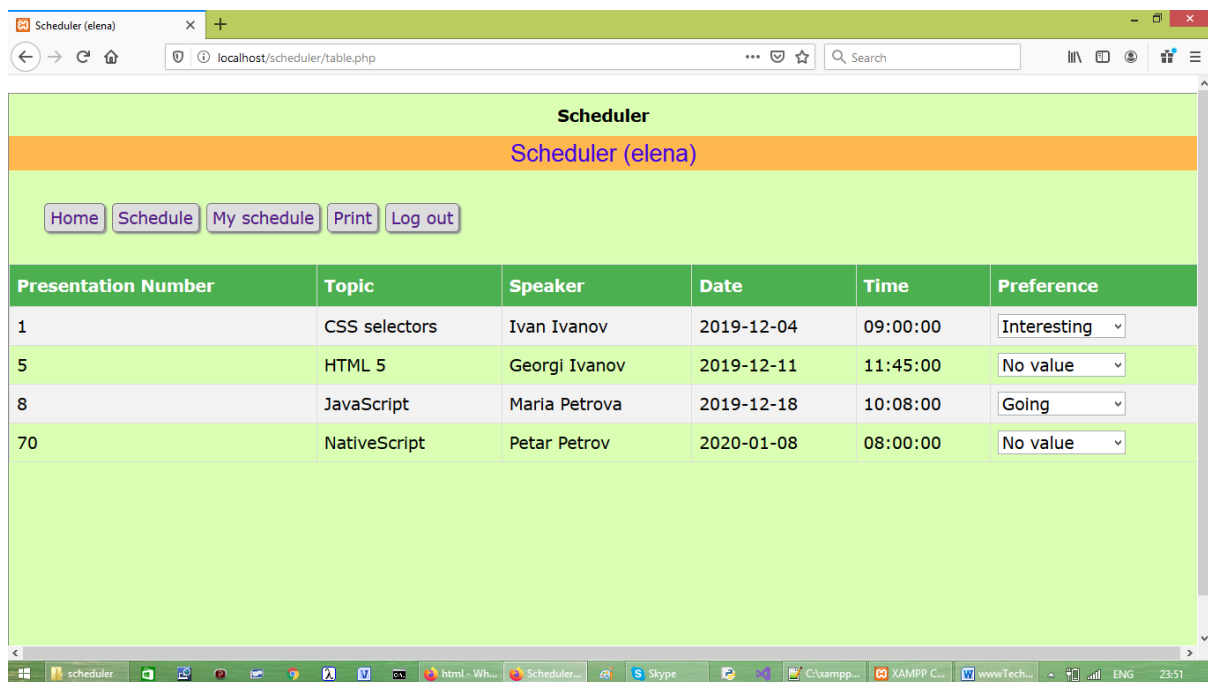


Figure 2 – Schedule

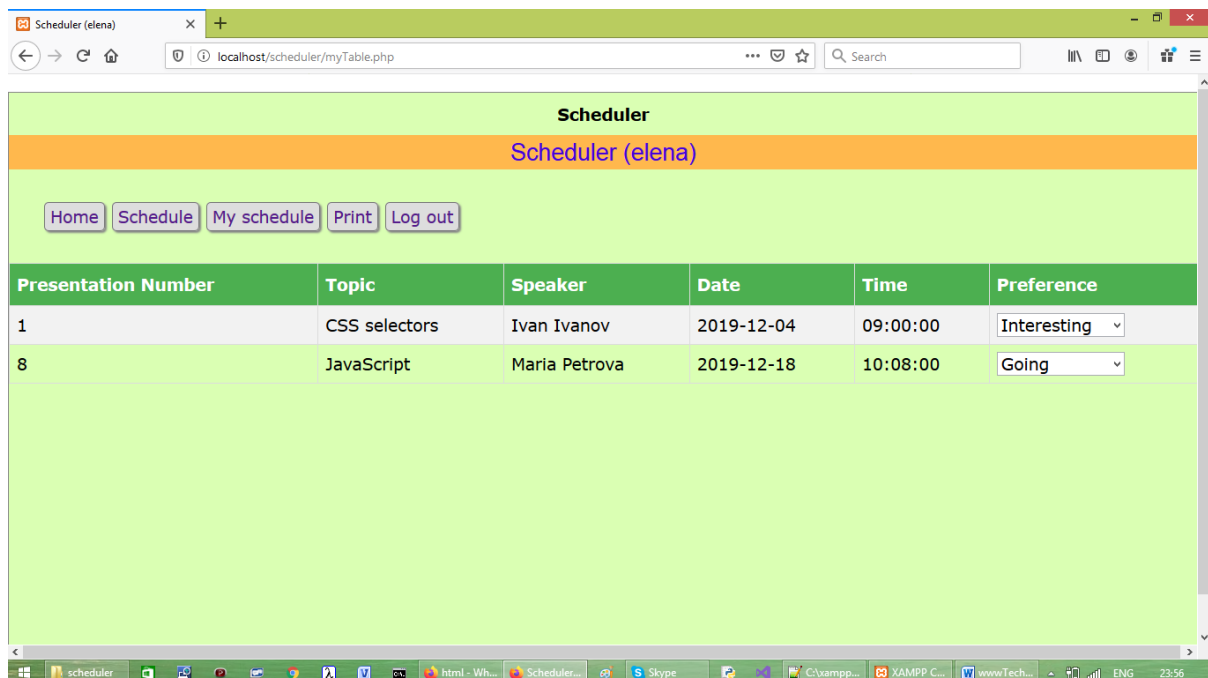


Figure 3 - My schedule

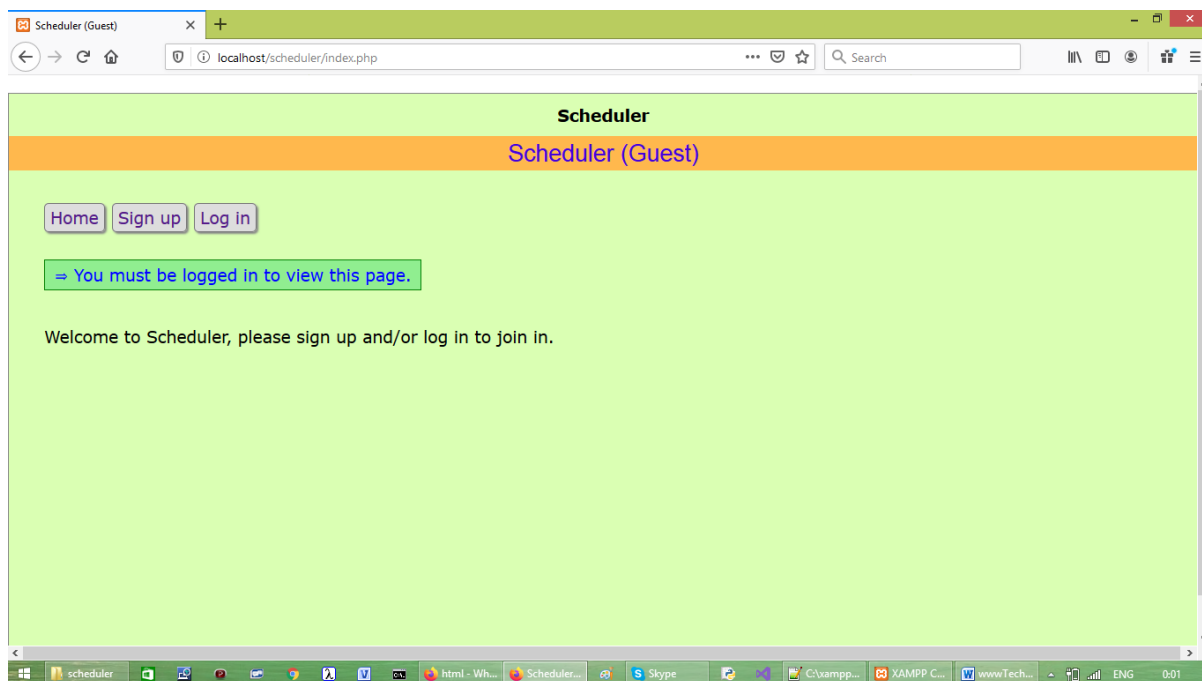


Figure 4 - Log in or Sign up

8. Описание на програмния код

Файлът **header.php** е общ за всички страници на приложението. Неговата работа е първо да генерира <head> частта от HTML документа, задаваща името на приложението и на потребителя и включваща файла със стилизацията на сайта **styles.css** и библиотеката jQuery. След това в зависимост от това дали потребителят е влязъл (logged in) с потребителското си име и парола и стартирал сесия (сесийните променливи `$_SESSION['user']` и `$_SESSION['pass']` пазят информация за потребителя, която е обща за всички страници на приложението в текущата сесия), се генерира едно от двете менюта, описани по-горе. **header.php** включва файла **functions.php**, който установява mysqli конекция с базата данни и съдържа няколко общи функции, използвани от всички PHP скриптове на приложението.

index.php е началната страница (home page) на приложението. Тя съдържа поздрав и евентуално подкана за sign up и/или log in.

signup.php генерира регистрационна форма с полета за потребителско име и парола и прави проверка дали полетата в нея са попълнени. Ако са попълнени и двете полета, се прави проверка в базата данни (таблица members) дали въведеното потребителско име не е заето. Ако не е заето, се прави нов запис в таблицата в базата с данните на новорегистрирания потребител (създава му се акаунт) и скриптът приключва работата си, генерирайки съобщение, че акаунтът е създаден (die(msg) функция). В противен случай се генерира съответното съобщение за грешка и формата за регистрация се изобразява отново.

login.php работи на същия принцип, но проверката се състои в това дали въведените username и password съществуват в таблицата с потребителите в базата данни. Ако е

така, се инициализират сесийните променливи `$_SESSION['user']` и `$_SESSION['pass']` с въведените стойности и тази информация е достъпна за всички PHP скриптове на приложението до прекратяване на сесията на потребителя. В противен случай се генерира съобщение за невалидни потребителско име и парола.

logout.php прекратява текущата сесия и заличава сесийните променливи.

table.php генерира цялостното разписание на презентациите с информацията за съответните предпочитания на текущия потребител. За всяка презентация от базата данни (релация `presentations`) скриптът прави справка за съответната предпочитаност на текущия потребител относно тази презентация в релацията `preferences` в базата, която съдържа информацията за потребител, номер на презентацията и предпочитания всеки път, когато потребител има интерес към някоя презентация, тоест предпочитаност, различна от `No value`. За всяка презентация потребителят може да избере предпочитаност от dropdown меню, като съответният `<select>` елемент има атрибут `class`, еднакъв за `<select>` елементите на всички презентации, и атрибут `id`, равен на `presentation_id` на конкретната презентация, за която се отнася. По тези атрибути JavaScript идентифицира еднозначно презентацията, за която се е случила промяна всеки път, когато потребителят промени предпочитаността си за дадена презентация. Съответната информация за новата предпочитаност, номер на презентацията и потребителското име на текущия потребител се изпраща на сървъра чрез AJAX call под формата на `key-value pairs`. На сървъра информацията за новата предпочитаност се обработва от скрипта **saveChoice.php**, който проверява дали за тези потребител и презентация вече има зададена предпочитаност в релацията `preferences` в базата данни. Ако има такава и новата предпочитаност не е `'none'` (т.е. `No value`), актуализираме старата предпочитаност в таблицата. Ако новата предпочитаност е `'none'`, изтриваме реда от таблицата `preferences` в базата данни, тъй като този потребител вече няма интерес (предпочитаност) относно дадената презентация. В случай, че новата предпочитаност не е `'none'` и няма ред в базата данни за този потребител и презентация, то добавяме новата информация в `preferences`.

myTable.php генерира лично разписание на потребителя. Скриптът прави заявки към таблиците `presentations` и `preferences` в базата данни и филтрира само презентациите, към които текущият потребител има интерес (отбелязана предпочитаност в базата данни).

9. Какво научих

Усъвършенствах уменията си в уеб програмирането с PHP, MySQL, HTML, CSS и JavaScript. Обогатих познанията си относно PHP сесии, AJAX и jQuery.

10. Използвани източници

[1]<https://www.w3schools.com/>

[2]<https://www.php.net/manual/en/langref.php>

[3]<https://jquery.com/>

[4]<https://mariadb.com/kb/en/mariadb-basics/>

Предал:

/Елена Валентинова Георгиева, ФН:81555, Компютърни науки, курс 4, група 1 /

Приел:

/доц. Милен Петров/