**Имена:**  Дафина Караманолева **фн:**61695*.***Начална година:** *2013* **Програма:** бакалавър, (СИ) **Курс: 3**   
**Тема: Система за комуникация между участници в курс  
Дата: 2016-06-19 Предмет: wwwTech2015\_16\_6ed\_SI\_summer имейл: dafina.kar@gmail.com**

**преподавател:** доц. д-р Милен Петров

# ТЕМА: Система за комуникация между участниците в курс

## 1. Условие

Условието на задачата е да се разработи уеб базирано приложение с помощта на PHP, HTML, CSS, JavaScript и MySQL. Приложението трябва да бъде базирано на трислойната архитектура MVC(Model-View-Controler). Логиката трябва да бъде разделена от изгледите и конекцията с базата данни. Целта на приложението е да предостави средство за комуникация на студенти в университет, чрез което студентите да могат лесно да комуникират по между си. Функционалност за различни по вид чатове, на различна тематика трябва да бъде разработена чрез възможността за създаване на различни чат стаи. Потребителите влязли в определна стая трябва да могат да се включват или не в определен разговор като влизат/излизат от него и вътре в него трябва да могат свободно да комуникират и да прехвърлят различни видове файлове.

## 2. Въведение

Web Chat application е уеб-базирана система за комуникация между съученици/колеги. Тя има за цел подпомагането на учебната дейност на потребителите си. Чрез нея те свободно могат да комуникират, групирани по собствените си предмети/интереси. Имат възможността сами да създават нови чат стаи с различни теми на разговори, което подпомага тяхната академична кариера.

Приложението е качено в гитхъб на този адрес: <https://github.com/dafinakaramanoleva/php_chat> и при интерес има възможност свободно да се разгледа и тества нашата имплементацията.

## 3. Теория

Основата нужна, за да може свободно да се разработи приложението е доста обширна. Включва HTML, CSS, JavaScript, PHP, MySQL. Аз ще обърна внимание на най-важните според мен компоненти, тези които най-много са ме затруднили или по точно AJAX и частта от PHP, свързана с обработката на идващи от клиента заявки.

* **AJAX** - AJAX е съкращение от Asynchronous JavaScript And XML (асинхронен JavaScript и XML – не че като се преведе внася яснота).

„Асинхронен JavaScript“ означава, че кодът се изпълнява в отделен процес на браузъра, което не кара страницата ни да замръзне. По време на изпълнение на синхронен код, браузърът не позволява никаква интеракция със съдържанието като скролване на страницата или кликване на линк. Прост пример за синхронен код е викането на alert() с JavaScript. Popup-ът, който излиза, блокира всичко останало на страницата, включително и изпълнението на по-нататъшни скриптове, докато Popup-ът не се затвори.

Ето един прост пример на AJAX завка с jQuery:

$.ajax({  
 url: "any/url/for/the/request",  
 type: "get",  
 success: function(data){  
 alert(data);  
 }  
});

* **Обработване на заявки с PHP** - за обработване на завките в PHP се използват така наречените свръхглобални променливи

Това са променливи, които могат да бъдат достъпени от всяка функция, всеки клас или файл, без да правите нищо специално. Свръхглобалните са:

* $GLOBALS
* $\_SERVER
* $\_REQUEST
* $\_POST
* $\_GET
* $\_FILES
* $\_ENV
* $\_COOKIE
* $\_SESSION
* **$\_POST**
* $\_POST се използва, за достъпване на информация изпратена от форма, със метод POST.
* **if ($\_SERVER[“REQUEST\_METHOD”] == “POST”) {**
* **// collect value of input field**
* **$name = $\_POST[‘fname’];**
* **if (empty($name)) {**
* **echo “Name is empty”;**
* **} else {**
* **echo $name;**
* **}**
* **}**
* **$GET**
* $\_GET се използва, за достъпване на информация изпратена от форма, със метод GET. Използва се и за достъпване и на информация през URL.
* **<html>**
* **<body><a href=”test\_get.php?subject=PHP&web=W3schools.com”>Test $GET</a></body>**
* **</html>**

Когато потребител кликне върху връзката, subject и web се изпращат на php файла, който ги обработва. Самия php скрипт, може да видите в долния код:

* **<html>**
* **<body>**
* **<?php**
* **echo “Study ” . $\_GET[‘subject’] . ” at ” . $\_GET[‘web’];**
* **?>**
* **</body>**
* **</html>**

## 4. Използвани технологии

* PHP
* JavaScript
* jQuery
* AJAX
* CSS
* Bootstrap
* HTML
* MySQL database
* Apache Server
* XAMPP Control Panel
* Sublime Text Editor
* Google Chrome, Firefox, Edge

## 5. Инсталация и настройки

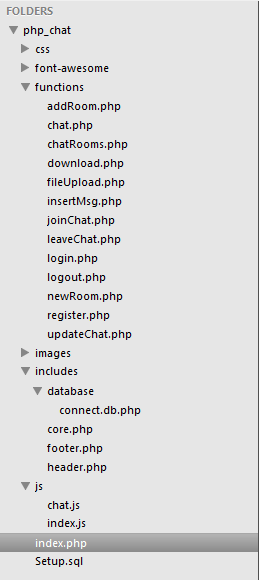
Както вече казахме за Control panel използваме XAMPP, така че той е най-необходим.

* Свали от тук - <https://www.apachefriends.org/index.html>
* Упътвания - <https://www.apachefriends.org/faq_windows.html>
* За настройка на порт и хост, при проблеми - http://stackoverflow.com/questions/14073985/xampp-apache-server-is-not-starting-after-skype-installation
* Setup.sql файлът в главната папка на проекта съдържа скриптове за създаване на базата данни, които се изпълняват на <http://localhost:8080/phpmyadmin>. Кратко ръководство на потребителя
* Домейн на системата -<http://localhost:8080/php_chat>
* Sign Up форма за регистраци, след която директно бивате навигиран до главната страница на системата, от където избирате желана чат стая или създавате такава и комуникацията Ви може да започне.

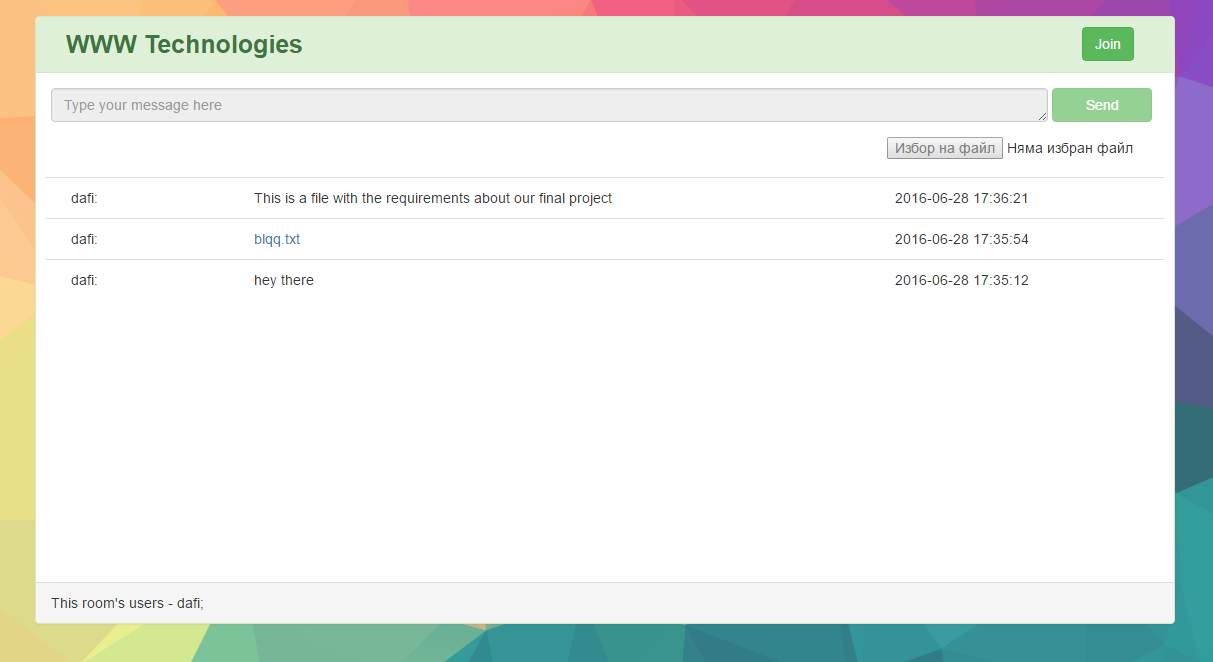
## 7. Примерни данни

Тъй като нашето уеб приложение поддържа регистрация на нови потребители, примерни данни не са нужни, тъй като всеки нов потребител има възможността да се регистрира и да използва самото приложение.

## 8. Описание на програмния код



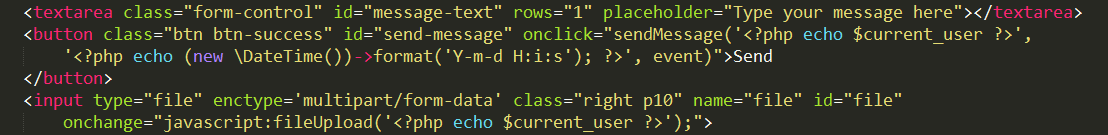
Фиг.1: Главна структура

На фиг. 1 е показана нашата файлова структура. Както се вижда състои се от поддиректории с js, css, images, където съответно се намират JavaScript, CSS фуйнкционалностите и прилежащите към приложението изображения. Директорията includes съдържа в себе си поддиректория за конекцията с базата данни и header-a и footer-a на нашето приложение, които се добавят в всеки един html файл, тъй като те съдържат линкове към използваните css и javaScript файлове. Най-интересна е functions директорията, тъй като там се съдържа главната логика на нашата система. В тази документация ще разгледаме втората част от самото приложение а именно: chat.php, joinChat.php, leaveChat.php, insertMsg.php, updateChat.php, fileUpload.php, download.php и накрая, но не последно chat.js. Няма да разглеждаме CSS файловете, тъй като от визуална гледна точка сме целяли простота и изчистеност и съответно сме се държали към bootstrap концепцията. Това, за което имам честта да ви разкажа аз е следното: На Фиг. 2 виждаме една чат стая – WWW Technologies. Така в заглавната част на стаята, в която се намираме виждаме в десния ъгъл бутон ‘Join. Точно този бутон изпълнява функционалностите присъедини се към стая или я напусни съответно, ако вече си член. Всички стаи са публични, съответно всеки потребител може да вижда какво се пише в стаята, но няма право да изпраща съобщения или да прикачва файлове (joinChat.php и leaveChat.php).

Фиг.2: Чат функционалност

Фиг.2: Чат функционалност

След като вече се е присъединил даден оптребител може да използва функционалностите на нашата система, а именно да изпраща съобщения или файлове. Функционалността е разработена с помощта на Ajax заявки и съответно PHP, за обработване на тези заявки на сървъра. Тук ще демонстрирам например как се осъществява изпращане на съобщение:



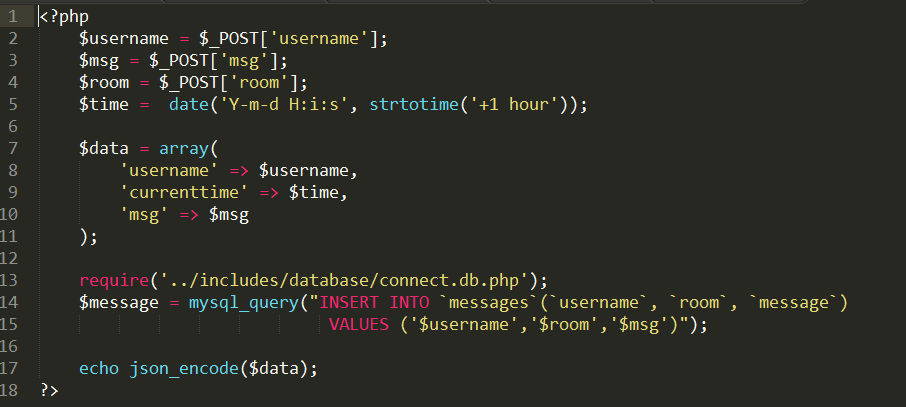
Фиг.3: Полетата в клиентската част

На фиг.3 виждаме текстовото поле, в което потребителя пише своето съобщение и двата бутона за изпращане на съобщението или прикачване на файл. Тук също можем да покажем и bootstrap класовете, които използваме за дизайн на системата и responsive структура.

На фиг. 4 виждаме функцията, която се изпълнява при натискане на бутона ‘Send’.

Фиг. 4 jQuery функция с ajax

На функцията подаваме потребителсно име и още няколко атрибута, които ще са ни нужни при изготвянето на заявката и записваме ги в един JSON обект - data. След това изпълняваме POST заявка, на която пращаме обекта data към insertMsg.php. A ето го и него:

На фиг. 5 виждаме логиката зад изпращането на съобщение. Записваме данните дошли от клиентската заявка в променливи, които чрез SQL заявка записваме в базата и по подходящ начин ги връщаме по заявката, където в success частта на заявката, върнатите резултати се обработват на страницата на системата.

Фиг.5: insertMsg.php

Аналогично се справяме с изпращането на файлове, презареждането на съобщения, заявките за присъединяване към чат стая или напускането и. Всичко имплементираме чрез ajax заявки, които пускат или чакат данни към сървъра и при успешно преминаване визуализират резултата при клиента. По подробната им имплементация, можете да намерите във файловете, които споменах в началото на тази секция, в директорията на проекта.

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

По време на планирането на проекта избрахме методологията за разработка Pair programming, тъй като с Адела Нанева работим лесно в екип, дори бихме казали, че работата върви изключително бързо и плавно по този начин. Дори използвахме работната обстановка на офис през уикендите, което ускори процеса и ефективността.Тъй като сме студентки и може би това ни е вродено ограничение ни беше времето. Въпреки това крайната равносметка е, че сме вложили около 60 човеко/часа труд и разработване на системата. Също така липсата на наш собствен сървър ни ограничи да качим проекта си онлайн.

Идеи за бъдещо развитие – Както всеки ценим много собствения си труд и се стремим към бъдещата му реализация. Със сигурност тази версия не е финална и предстоят още много промени и оптимизации. Краткосрочният ни план е да изгладим проблемите в програмния код и да качим системата си онлайн, за да разберем дали ще се хареса тя на потребителите. След адекватна обратна връзка ще издигне приложението си на следвашото ниво.

## 10. Какво научих

С разработването на този курс научих изключително много за уеб програмирането и цялостната интеграция на един уеб базиран сайт. Упражнението над PHP, където нямах особен опит освен с упражненията през семестъра, ми беше много полезно. Както в теорията споменах, обработването на заявките идващи от клиентската част бяха голямо предизвикателство за мен и научих как точно се случва. Особено полезно ми беше самото свързване на клиентска част със сървърна. Може би най-много се въодущевих разработвайки точно връзките, тъй като и в работата си и по време на другите курсове правим проекти и се учим на една определена част или front-end или back-end или бази от данни. В този проект имахме възможността да обединим знанията си от всичко научено до сега и да ги приложим :).

## 11. Използвани източници

* <http://getbootstrap.com/>
* <http://api.jquery.com/jquery.ajax/>
* <https://developer.mozilla.org/en-US/docs/Web/API/FormData/Using_FormData_Objects>
* <http://www.w3schools.com/js/js_timing.asp>
* <https://www.youtube.com/watch?v=imCTMz_XW5A&list=PL97E5F049784B72C4&index=5>
* <http://www.w3schools.com/sql/default.asp>
* <http://stackoverflow.com/questions/6326696/connecting-to-mysql-using-php>

Предал: ………………………….

/*фн, имена*/

Приел: ………………………….

/доц. *Милен Петров*/