**Имена:**  Мерлин Мехмед, Полина Тодорова **фн:** *61890, 61931***Начална година:** *2018* **Програма:** бакалавър, (СИ) **Курс:**  3   
**Тема:** Система за групова работа **Дата:** 2018-06-30  **Предмет:** wwwTech2017\_18\_10ed\_SI\_summer **имейл:** merlinmehmed17@gmail.com, p.v.todorova@abv.bg

**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 използваниресурси**, архива да се казва 9999\_project\_final.zip. (Успех!). (Редактирайте маркираните зони в жълто с коректната информация)**

# ТЕМА: Система за групова работа

## 1. Условие

Система за групова работа:

* позволява качване, създаване, редактиране и сваляне на документи
* всеки документ си има собственик, който решава кои потребители имат достъп до файла и могат да го редактират
* само собственикът на файла може да го изтрие
* груповото редактиране се извършва чрез уеб сокети

## 2. Въведение

Съвместната работа представлява подход за преподаване и учене, при който групи от обучаеми работят заедно за решаване на поставената задача. В такава ситуация се очаква членовете на групата да се организират и работят продуктивно като разпределят задачите и натоварването помежду си. Обучаемите общуват, обменят идеи и задават въпроси. При обучението в университет груповата работа е честа практика. Заедно с нея обаче идват и някои неудобства като съгласуването на груповите срещи с ангажиментите на всеки участник в екипа, непрекъснатото препращане на файлове и сливането на документи, ако двама участници са редактирали някаква част от файла едновременно. Тук на помощ идва системата за групова работа, която позволява достъп до един файл по всяко време и от всяко място, не изисква изпращане на файлове и разрешава само на един потребител да редактира документа в даден момент, останалите могат само да го преглеждат.

## 3. Теория

**Трислойна архитектура на приложение**

В софтуерното инженерство, многослойната архитектура е архитектура от тип клиент-сървър, в която интерфейсът, обработката на приложения и съхранението и обработката на данни са логически разделени на отделни модули. Най-разпространената форма на многослойна архитектура е трислойната архитектура.

Трислойната архитектура се състои от следните три слоя:

* **Презентационен слой**

Презентационният слой е на най-високо ниво в приложението и потребителят има директен достъп до него. Освен, че служи комуникира със останалите слоеве, презентационният слой предоставя различни видове информация на потребителя.

* **Слой за бизнес логика (междинен слой, слой за обработка на данни)**

Този слой е изтеглен от презентационния слой, и като отделен такъв, контролира функционалността на приложението като извършва различни процеси по обработката на данните.

* **Слой за данните**

Този слой се състой от сървър база данни. Тук информацията се съхранява и достъпва. В слоя за бази данни информацията се съхранява независима от бизнес логиката или сървърът за приложения. Когато данните се съхраняват в отделен слой се увеличава мащабируемостта и се подобрява производителността.

**Употреба в уеб разработката**

В областта на уеб разработката трислойната архитектура често се използва в уеб сайтове, които се състоят от 3 слоя:

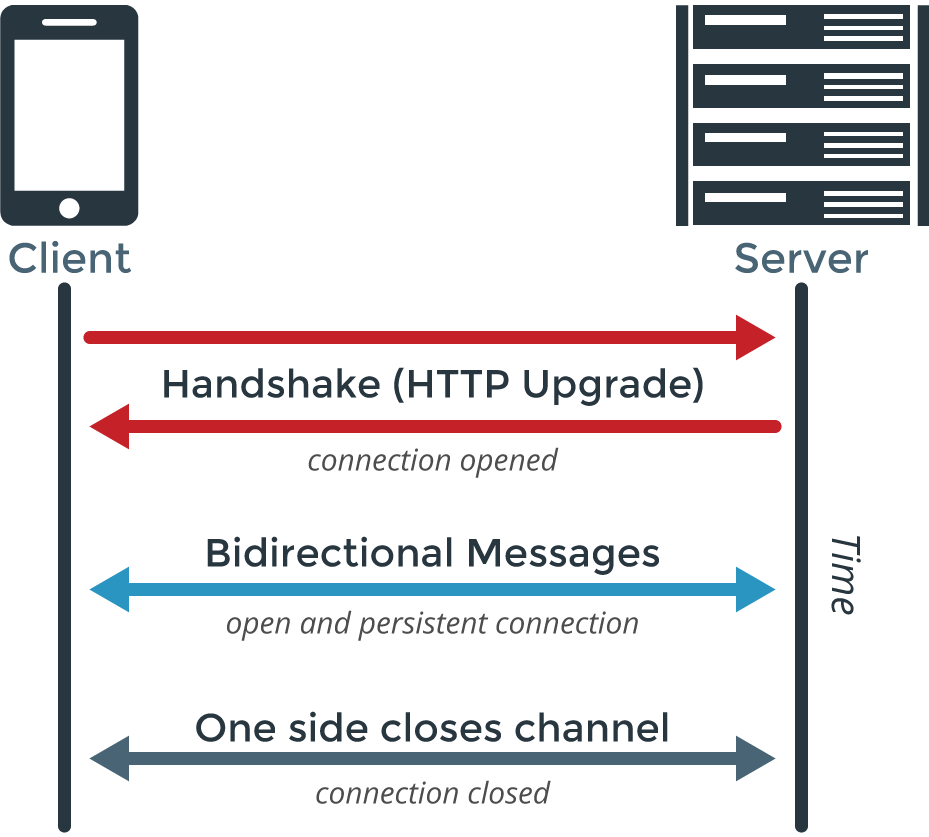
* Front-end слой (Клиентски слой) - този слой, е съдържанието, което се възпроизвежда от уеб браузърът и достига до крайния потребител
* Среден слой – той реализира логиката на самото уеб приложение. Това се осъществява със сървър, който обработва и генерира динамично съдържание
* Back-end слой – той представлява база данни, която се състои, както от самата информация в таблици, така и от система за управлението ѝ.



**Уеб сокети**

Уеб сокетите имат двупосочна, напълно дуплексна (full-duplex), постоянна връзка от уеб браузър към сървър. След като се установи връзка чрез уеб сокет, връзката остава отворена, докато клиентът или сървърът не реши да затвори тази връзка. При тази отворена връзка клиентът или сървърът могат да изпращат съобщения по всяко време до другия. Това прави уеб програмирането изцяло задвижвано, а не (само) инициирано от потребителя. Това е състояние. Освен това, понастоящем едно единствено сървърно приложение е наясно с всички връзки, което позволява комуникация с произволен брой отворени връзки по всяко време. Най-често уеб сокети се използват при:

* Социални приложения
* Групови игри в браузър
* Групово редактиране на файлове в реално време
* Чат приложения
* Излъчване на събития на живо



## 4. Използвани технологии

Разработената система е под формата на уеб сайт с трислойна архитектура, като за отделните слоеве са използвани следните технологии:

* презентационен слой (Front-end) – CSS, JavaScript и HTML
* база от данни (Back-end) – MySQL
* бизнес логика (Server) – PHP

## 5. Инсталация и настройки

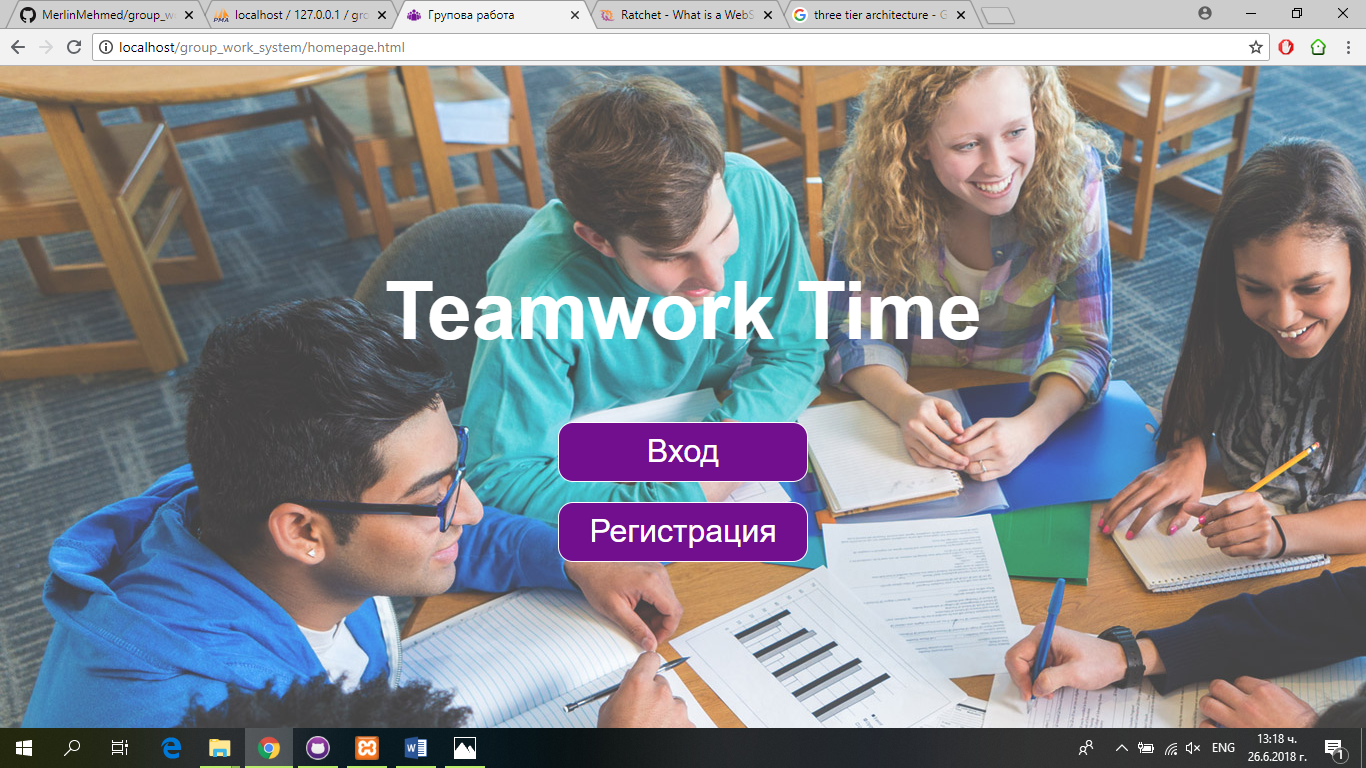
За изпълнението на уеб приложението използваме XAMPP, който осигурява необходимите сървър и база от данни. Тъй като приложението използва уеб сокети, в конфигурационния файл **php.ini** променяме **;extension=sockets** на **extension=sockets**, за да разрешим използването им.

Сървърът, който създава сокети и изпълнява заявките, направени чрез тях се стартира чрез командата **php -q …\php-socket.php**, като файлът се намира в папка различна от htdocs.

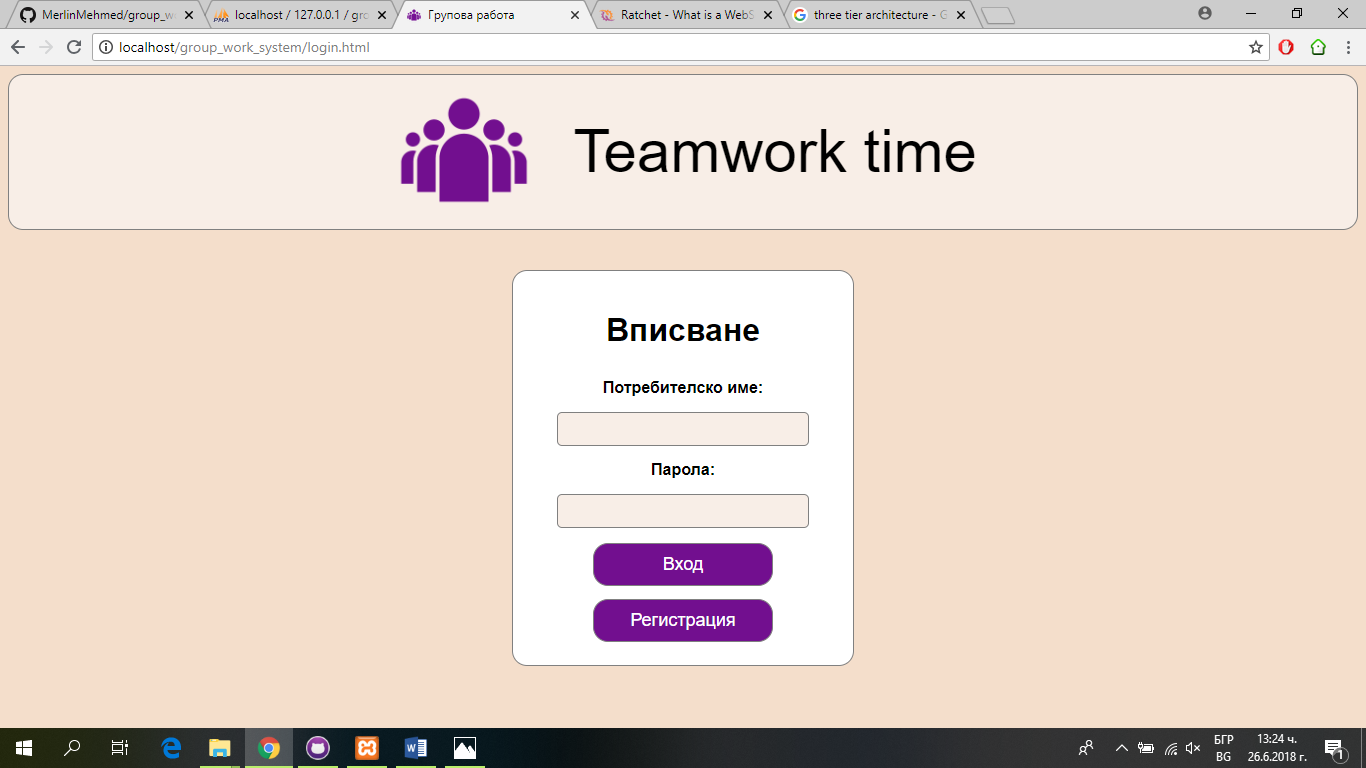
При стартиране изпълняваме PHP скрипт, който създава базата от данни и въвежда начални данни в нея. URL за зареждане на началната страница е <http://localhost/group_work_system/homepage.html>. Възможни потребителско име и парола за достъп са съответно fmi и fmipass.

## 6. Кратко ръководство на потребителя

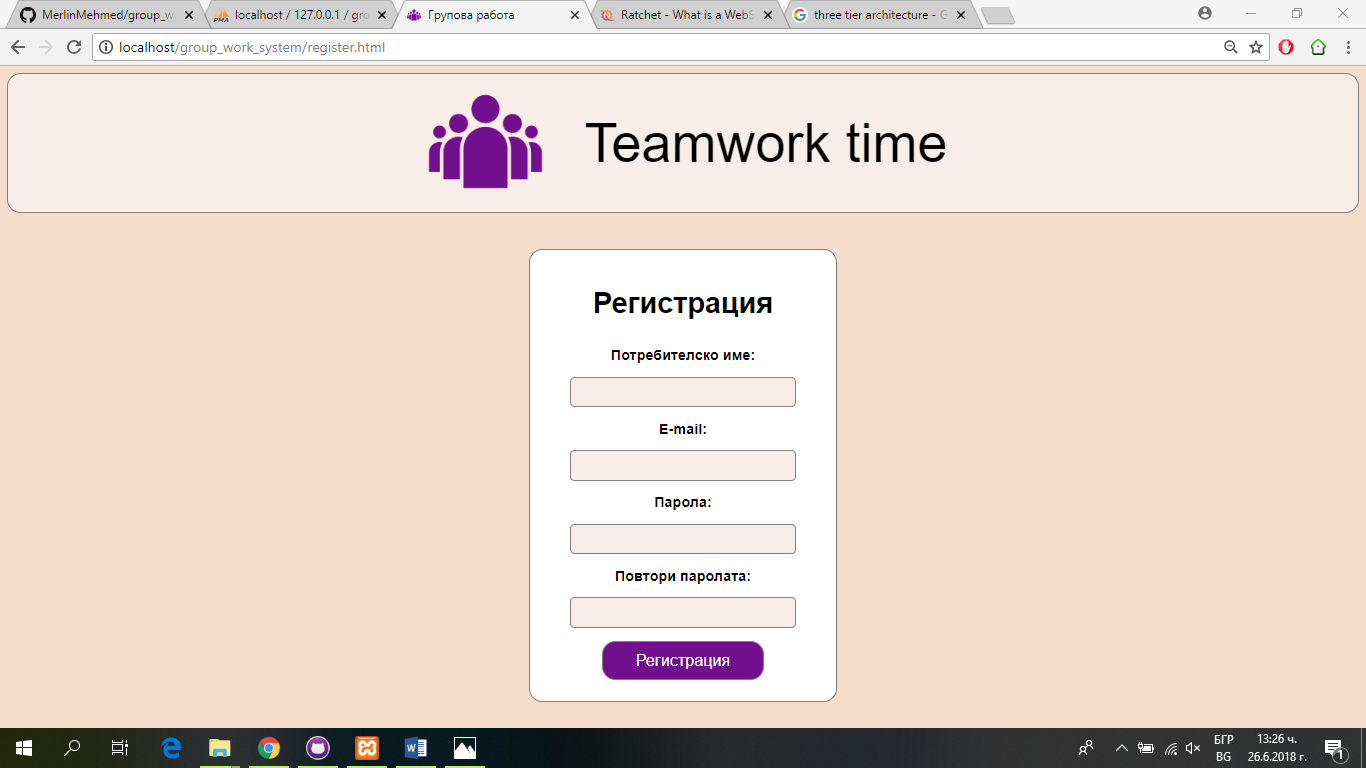
Начална страница: първата страница, която вижда потребителят след отваряне на уеб сайта. Има 2 възможности за избор – вход и регистрация, тъй като системата може да се използва само от регистрирани потребители.



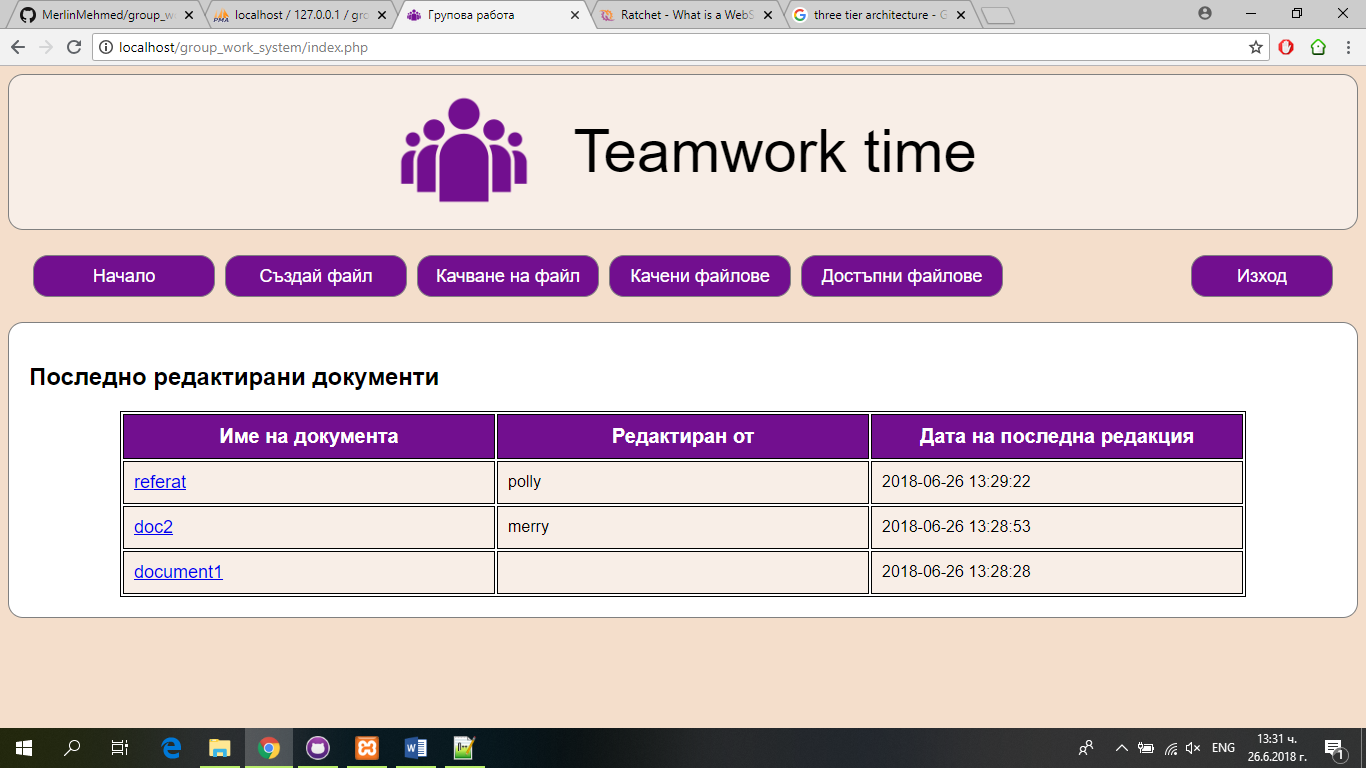
Форма за вход: потребителят има възможност да въведе потребителско име и парола за вписване в сайта или да се регистрира, ако все още не го е направил



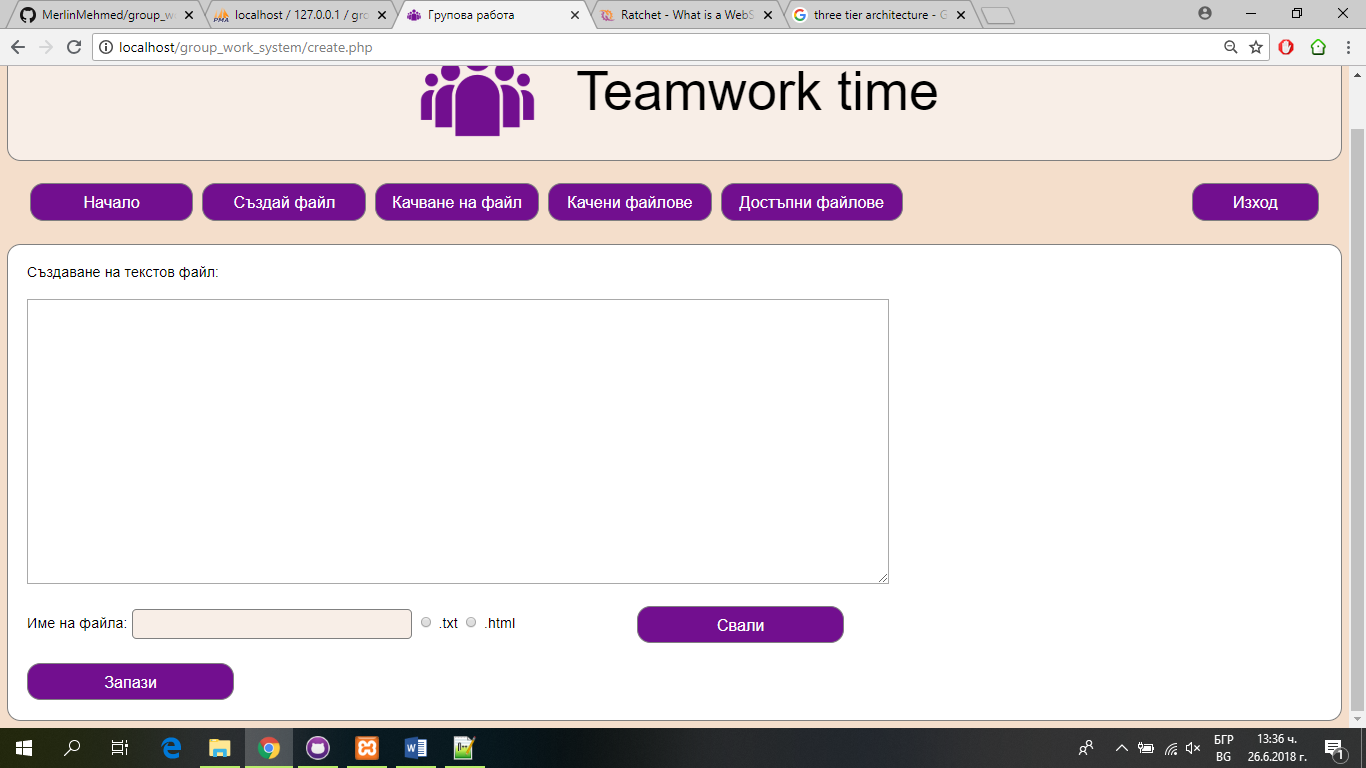
Форма за регистрация: потребителят има възможност да се регистрира в сайта чрез попълване на полетата за потребителско име, email и парола



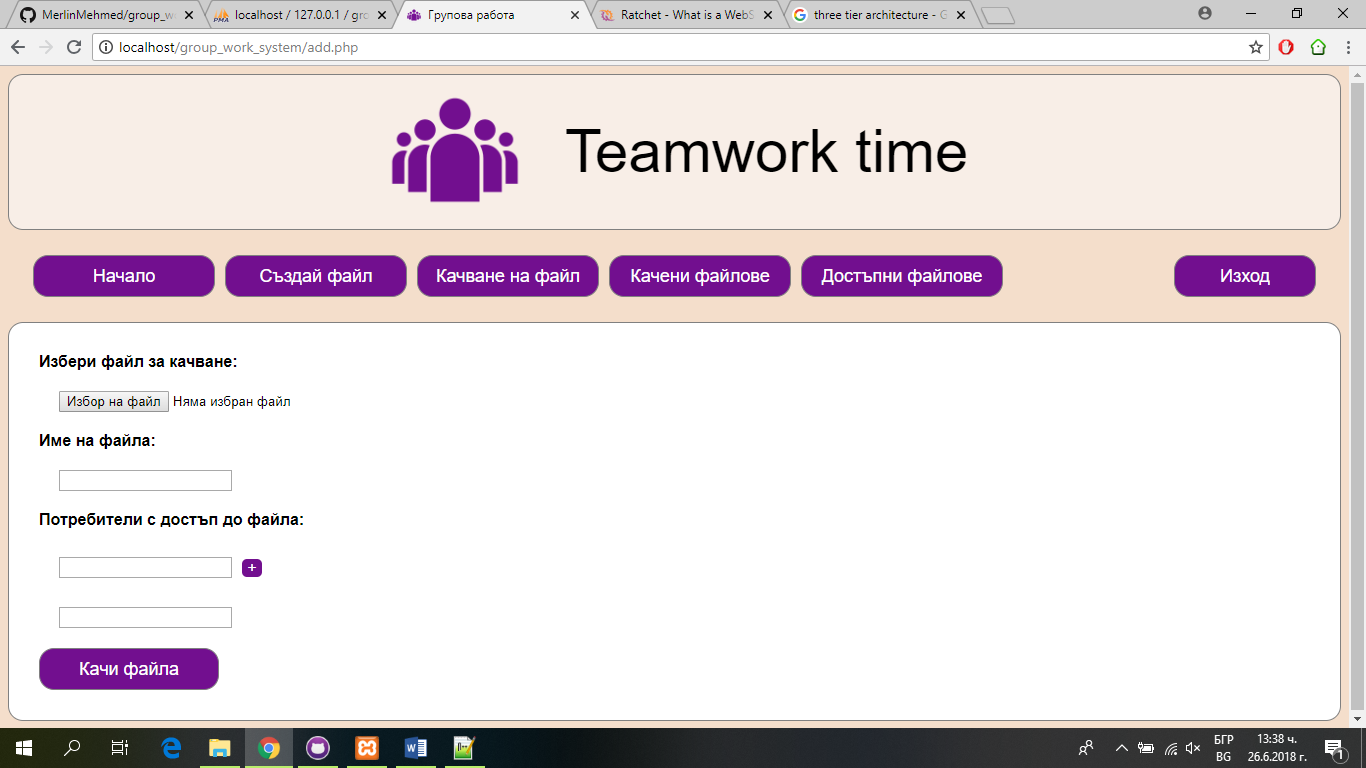
Начална страница на регистриран потребител: включва таблица с документите на съответния потребител, които последно са били редактирани, като са подредени така, че последно редактираният да е най-отгоре



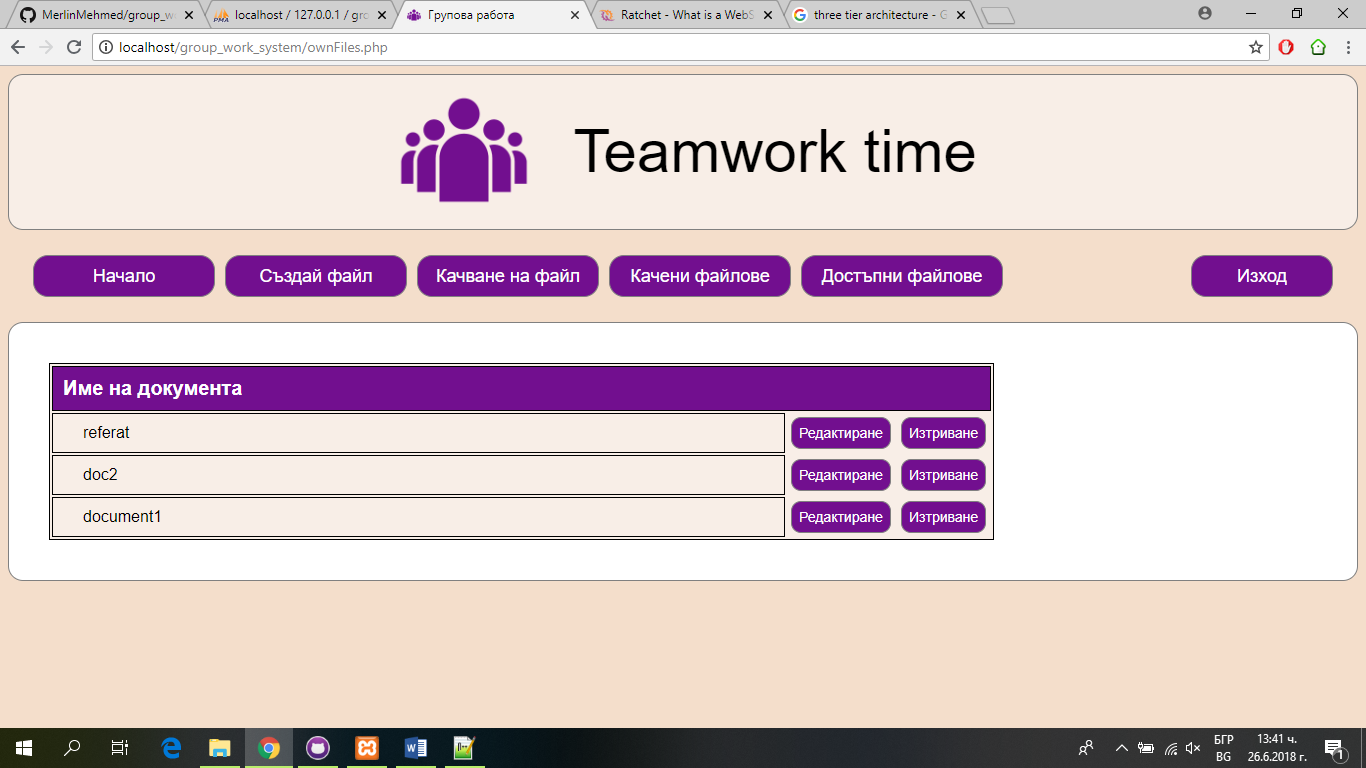
Създай файл: възможност за създаване на текстов файл, който след това може да бъде изтеглен като .txt или .html и/или качен на сървъра



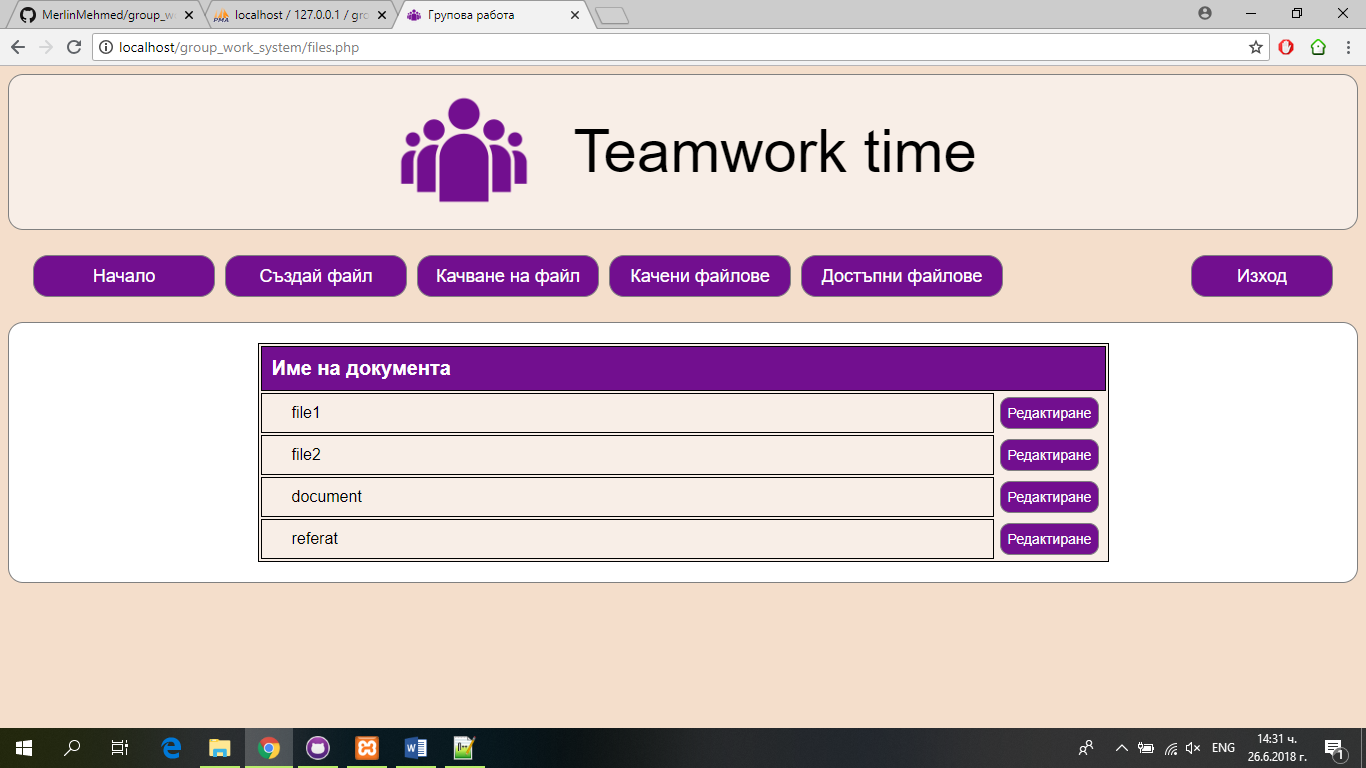
Качване на файл: възможност за качване на файл, като му се зададе име и потребители, които могат да го редактират (чрез клик върху бутона „+“ текстовите полета за въвеждане на потребители могат да бъдат неограничен брой)



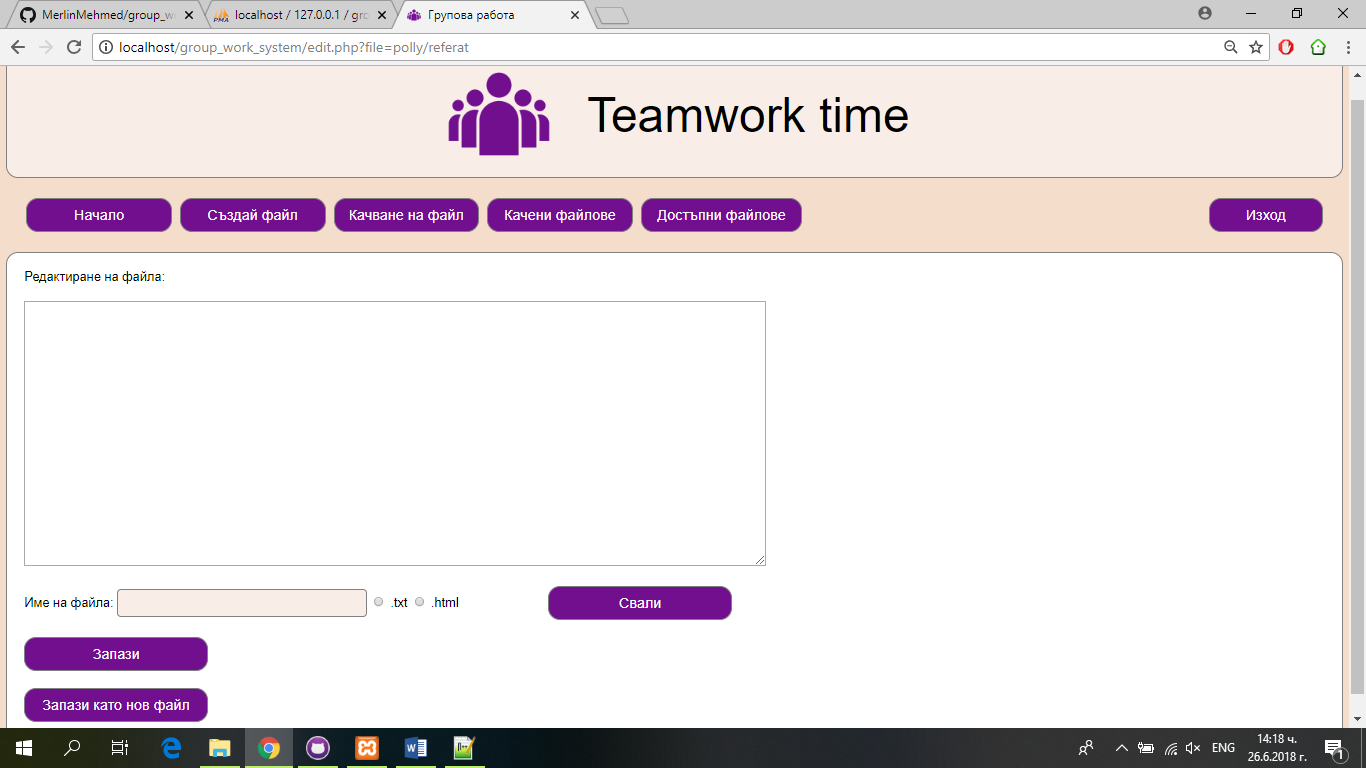
Качени файлове: таблица с файловете качени от вписания потребител с възможност за редактиране и изтриване чрез клик върху съответния бутон след името на файла



Достъпни файлове: таблица с файловете, които вписания потребител има възможност да редактира след клик върху бутона след името на файла



Редактиране на файл: възможност за редактиране и форматиране на файл, 3 възможности след приключване на редакцията – запис на промените върху същия файл на сървъра, запис като нов файл и сваляне като .txt или .html



## 7. Примерни данни

* За вход в системата:

потребителско име: fmi

парола: fmipass

## 8. Описание на програмния код

Основните обекти в системата са потребителят и документите, които той създава. За целта сме реализирали класовете – Db, User, DocumentInfo, Document.

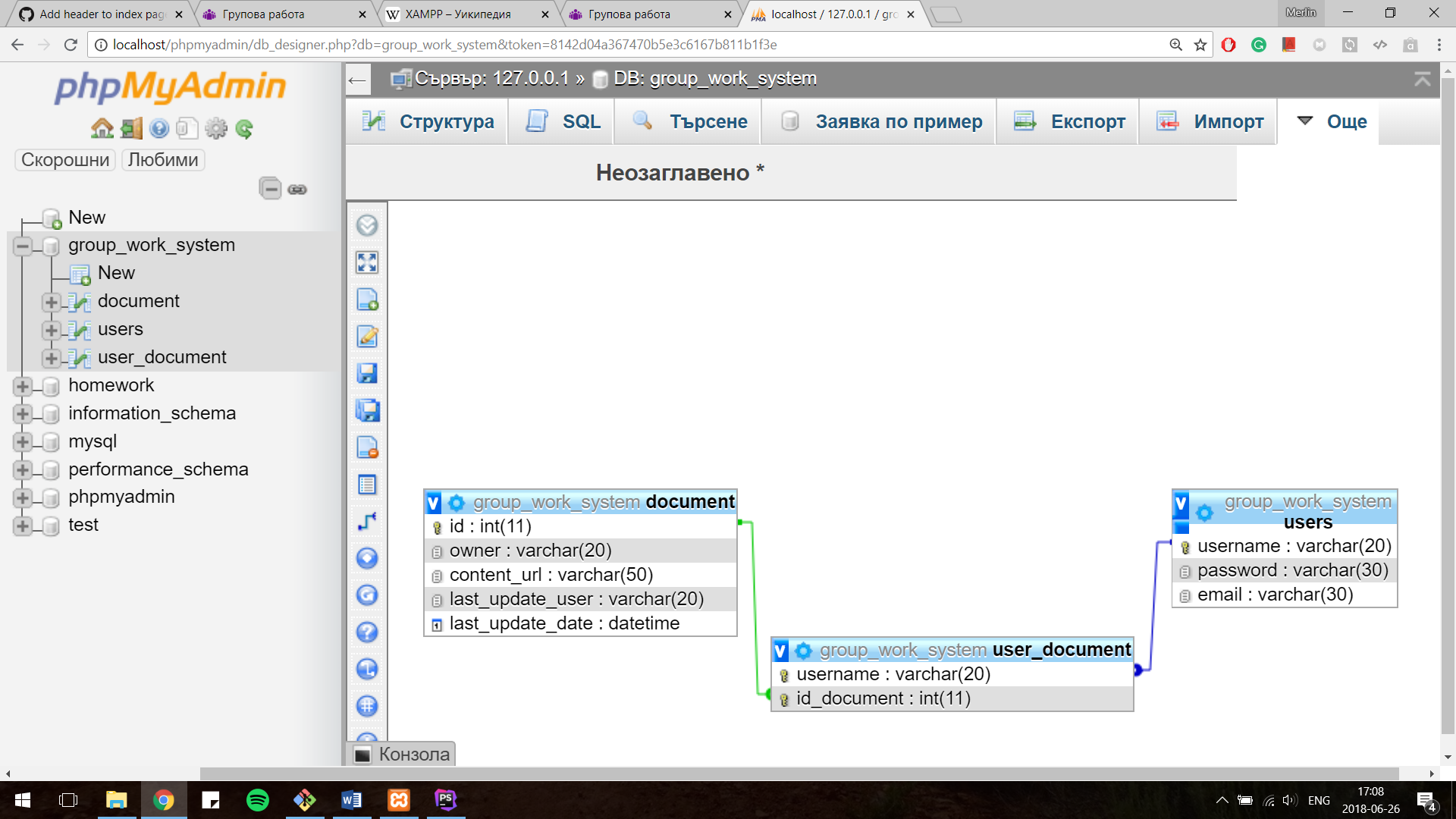
Класът **Db** осъществява връзката с базата от данни и има метод **getConn()**, която връща тази връзка така, че другите класове да я използват свободно.

Класът **User** съдържа данните за потребителя, като те могат да се записват в базата от данни чрез метода **insert(),** зареждат чрез **load()** или **fetchAll()** в зависимост дали искаме да заредим един или всички потребители.

Класът **DocumentInfo** съдържа данните за документа, който потребителят е добавил. По подобен начин на User той може да бъде зареден от базата чрез метода **load()**, добавен в нея чрез **insert()**. Освен това чрез методите **addRight()** и **removeRight()** да се добавят и премахват права за достъп до документа на други потребители. Това е позволено само ако потребитеят, който желае да изпълни съответните действия е собственик на документа. Тази проверка става чрез метода **isOwner().**

Класът **Document** се състои от DocumentInfo и съдържанието на файла, който ще бъде добавен или редактиран. Методът на класа loadDocument() служи за извличане на съдържанието на документа, като това става чрез четене на файла, пътят, към който се намира в DocumentInfo. **UpdateDocument()** се използва при редактиране на вече създаден документ, като съдържанието му се презаписва и в базата от данни се обновяват часът на последна редакция и потребителското име на човека, който я е извършил. Целта на метода **writeFile()** е да презапише съдържанието на документа.

Структурата на базата от данни е следната:



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

**Приноси на студента:**

**Ограничения:**

* функционалностите на системата могат да се използват единствено след регистрация;
* възможност за редактиране само на .txt и .html файлове;
* възможностите за редактиране са удебелен, наклонен и подчертан текст, форматиране на линк, заглавия, цитат и код, добавяне на табулация и изображение;
* споделеният документ може да бъде изтрит само от собственика му;
* за споделяне на документ потребителят трябва задължително да знае потребителското име на човека, с когото иска да го сподели.

**Възможности за бъдещо разширение:**

* възможност за групово редактиране и на други типове файлове;
* добавяне на още възможности за форматиране;
* създаване на групов проект, в който се съхраняват множество данни;
* възможност за добавяне на индивидуални задачи на потребителите, участващи в проект.

## 10. Какво научих

Научихме как се прилага MVC архитектурата и как трябва да бъдат разпределени файловете между различните й слоеве. Също така как работят уеб сокетите и приложението им при групово редактиране на файлове. По-важното обаче е знанието за прилагане на ООП с PHP. Нови знания са и начините за конфигуриране на сървъра.

## 11. Използвани източници

1. <https://en.wikipedia.org/wiki/Multitier_architecture>
2. <http://socketo.me/docs>

Предал (подпис): ………………………….

/*фн, имена, спец., група*/

Приел (подпис): ………………………….

/доц. *Милен Петров*/

## Препоръки за предаване на проекта (изтрийте при предаване на проекта)

Заб. *Спазвайки препоръките по-долу биха спомогнали да направите добри проекти по Уеб технологии.*

## Финален проект (инструкции към 7 издание)

2. Изпитни проекти: (настоящият документ може да съхраните като .docx . За хората, ползващи редактори, различни от MS Office – освен docx/rtf да качат и pdf версия на документацията – за по-сигурно.)

2.1. Темите за изпитните проекти трябва да са съгласувани с мен на място (вече няколко хора го направиха); като тема си запишете ф.н. в гугъл докса - там пише как се записвате (в коя колона и какъв разделител да ползвате); Обем на проектите: 30 човекочаса на човек на проект. Това е доста относително, но все пак е нещо.

2.2. За документация на проекта ползвайте шаблона (ще гледам да го кача скоро); Задължително потребителска документация (а.к.а. userguide) - няколко скрийншота с кратко разяснение; условие на проекта (т.е. какво сте разтълкували сме се разбрали да правите - то може и да се различава от описанието дадено в гугъл докс-а към момента); както и инструкции за инсталиране; за защитата - ще дам няколко дати през сесията/преди сесията + официалните дати, за който не успее да мине преди това; за защитата - кода, документацията/необходими библиотеки/среди, инсталации - за инсталирането се предават на DVD (ако има още такова нещо като CD-може и на CD). Може по изключение да сложа И форма за качване само на документацията и програмен код и в мудъл, НО идеята е, че ако няма интернет (да речем е паднал мудъла, спрял е тока и т.н. - само по съдържанието на диска, който сте предали да може да се инсталира и подкара проекта); Също така разпечатвате първа страница от проекта (с името ви и заданието), 1 страница от userguide-а и последната страница, където пише предал/приел (там пише вашите и моите имена), т.е. не е необходимо да печатите цялата документация - така или иначе ще я има в електронен формат; За хората, които не ползват MS Office - ще помоля освен изходният документ в docx/rtf, да качат нещата и е pdf формат, т.к. често такъв тип документация се размества и не се чете. В кода сложете и MySql sql скрипт със създаване на на таблицата и скриптове, задаващи примерни данни (т.е. може да изтестваме приложението дали работи без данни, и ако за да се види пълната прелест на проекта е нуждно да се вкарат предварително данни - sql и/или снимки/звуци и т.н. може да ги подготвите на диска).

2.3. Срок за защита на изпитните проекти - до изпита. За съгласуване на теми - ми пишете да се разберем за час за консултации за проектите.

Вижте и инструкциите към 6 издание – които не противоречат с инструкциите за настоящото издание – са валидни и сега.

## Финален проект (инструкции към 6 издание)

Заб. *Спазвайки препоръките по-долу биха спомогнали да направите добри проекти по Уеб технологии.*

**Задължително:** Реферата да е в zip файл с име на зип файла: **fakNo\_final\_project.zip** където вместо **fanNo** пишете факултетният си номер (според инструкциите, зададени в форума на курса).

## Още няколко упътвания:

I. АРХИТЕКТУРА НА УЕБ СИСТЕМАТА: Да има три-слойна архитектура (Препоръки: 1. презентационен слой - css/js/html, 2. БД: MySql и като допълнение по желание може да имате импорт/експорт към XML; 3. Бизнес логика - Php)

II. ФУНКЦИОНАЛНИ ХАРАКТЕРИСТИКИ: Съгласуват се с преподавателя - избира се темата, а каква да е функционалността - питате на лекции или се явявате на предварителна защита, където съгласувате обхвата на изискванията;Да се познават и спазват добрите практики, завършеност - според сложността на приложението – 30 човеко-часа);

III. НЕ-Функционални характеристики:

- Конфигурируемост (лесно да може да се инсталира - например: смяна на едно место ако се смени физически папката на сървъра - да е в под-папка, смяна на адреса - IP/URL, смяна на име/парола/

- Разширяемост - лестно да може да се разширява функционалността на различните слоеве;

- Документация - без да се пуска проекта - да може да се ориентира в неговата функционалност, как се настройва, точки на разширение на отделните слоеве- ако има особенност, примерни данни за тестване - администраторски акаунти, ръководство на потребителя за различните роли - скрийншоти с кратко заглавие).

IV. Други изисквания, зададени на лекция (следете лекциите и форумите).

**Заб.** *Има вероятност да пусна нова инстанция на системата за рефератите, но този път за проекти, където в html да сложите документацията си (т.е. подобен на този шаблон, но в html формат)*.

За изискването конфигурируемост (описвате в документацията), се очаква нещо от вида:

***<?php***

*//*install.php *- can called once, initiating database, creating configuration file of doesnt exists, etc.*

*//Optional::* ConfigPanel.php *-> Configures/changes settings of app*

*//Required:* config.php, db\_sql.txt

*//required: help.php (can be part of documentation)...*

*//or config.ini -> със същите настройки;*

*class Config {*

*$SITE\_FN = 61999; //can be used bellow*

*$SITE\_CREATOR = "Your Name(s)";*

*$SITE\_ADMIN\_EMAIL = "your@email.com";*

*$SITE\_INFO = "This project was created during ...year, on Web Technologies, Sofia University, FMI, lead by:*

*Name of Instructor, assistant: Name-Of-Assistant";*

$SITE\_URL=“<http://loremipsum.fmi.uni-sofia.bg/WEBTECH/www_6ed_prj/61999_alg_animation>”;

*$ROOT\_FOLDER="c:\xampp\htdocs\www\_6ed\_prj/61999\_alg\_animation"*

*$DB\_USER="61999\_user";*

*$DB\_PASS="61999\_pass";*

*$DB\_NAME="www\_6ed\_61999\_alg\_animation";*

*$SITE\_DESCRIPTION="What is ready, and what can be improved for future";*

*$PROJECT\_REQ="...(from documentation)";*

*}*

***?>***

История на версиите

* Последна модификация 0.2/2016-05-13/by MP

Успех!