**jИмена:**  Галин Симеонов, Александър Витков **фн:**  *81635, 81618***Начална година:** *2020* **Програма:** бакалавър, (KN) **Курс: 4**   
**Тема: filemeup  
Дата: 2020-06-01 Предмет: Web Технологии имейл: gtsimeonov@uni-sofia.bg**

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

# ТЕМА: Filemeup

## 1. Условие

Система за качване, преглед и споделяне на различни видове ресурси.

## 2. Въведение

Fileup е система за качване, организиране и споделяне на файлове.

## 3. Теория

Имплементацията ни е базирана върху виртуална файлова система - всеки потребител притежава своя собствена коренна директория, в която има право да качва и трие файлове, както и да създава потдиректории. Качените файлове се записват в преопределена директория в сървъра, а метаданните за тези файлове се държат в базата от данни.

Потребителите могат да споделят файлове чрез генериране на линкове, като всеки може линк има определени свойства, служещи за защита на данните – линкът може да иска парола за изтегляне, може и достъпът да е ограничен, така че само определени потребители да могат да го ползват.

## 4. Използвани технологии

PHP 7.3 + MariaDB за сървърната част

JavaScript/HTML/CSS за клиента

Кафе Nova Brasilia, както за клиента, така и за сървъра

## 5. Инсталация и настройки

Конфигурацията на системата става чрез конфигурационния файл php/configuration.php.

Трябва да се направят следните неща:

* Да се създаде папка, в която сървърът да записва файловете.   
  По подразбиране тази папка е C:\fileup\_storage, може да се промени в php/configuration.php
* Да се конфигурира MySQL потребител/парола/адрес на сървъра. По подразбиране се ползват “root”/празна парола/127.0.0.1
* Да се подкара скрипта INIT\_DATABASE.sql, който създава базата данни “fileup” и нужните таблици в нея

## 6. Кратко ръководство на потребителя

Първо трябва да се регистрирате и логнете в системата.

При успешно влизане в системата се показва прозорец с главната ви директория. В нея може да създавате нови директории чрез бутона **New Folder** и да качвате файлове чрез бутона **Upload**.

Ако дадете десен бутон на някой файл, се отравя меню с възможни действия, например Open/Download/Share/Delete.

## 7. Примерни данни

При регистрация:  
*Username: Alice  
Email:* [*alice@wonderland.com*](mailto:alice@wonderland.com) *Password: rabbit*

Качването на файлове става изцяло чрез уеб интерфейса.

## 8. Описание на програмния код

### 8.1 Структура на базата от данни

Метаданните за файловете и директориите се представят чрез node-ове (nodes таблицата във fileshare.sql)

Съдържанието на директориите се описва чрез връзките (node\_links) към node-а на директорията (node\_links таблицата във fileshare)

Достъпът до даден ресурс се описва чрез node\_access таблицата и може да бъде комбинация от (can\_read – има право да чете и копира файлът) и (can\_write има право да променя и трие файлът).

### 8.2 Структура на PHP частта от backend-а

Главната част от php кода е в php/database.php, където се реализира връзката между PHP и базата от данни.

Логиката на симулираната файлова система се реализира в php/node.php

Освен това има редица API endpoint-ове, които JavaScript кодът извиква.

* Заявка за изтриване на файл се прави чрез php/delete.php
* Заявка за местене на файл се прави чрез php/move.php
* Заявка за списък с файловете в директория се прави чрез php/readdir.php
* Заявка за четене съдържанието на файл се прави чрез php/readfile.php
* Заявка за преименуване на файл се прави чрез php/rename.php
* Заявка за създаване на директория се прави чрез php/mkdir.php
* Заявка за качване на файл в системата се прави чрез php/upload.php
* Заявка за регистриране се прави чрез php/register.php
* Заявка за влизане в системата се прави чрез php/login.php

### 8.2 Структура на frontend-а

Index.php страницата, въз основа на това дали потребителя е логнат или не, ще зареди една от тези две страници:

loginregister.php и loginregister.js – Landing страницата, в която се съдържат формите за регистрация и логване.

loggedin.php и loggedin.js – loggedin.php e страницата, която се зарежда, като потребителят е логнат.

След като потребителят успешно се логне, програмата е single page application - заявките към сървъра, описани в списъка горе, стават асинхронно чрез XMLHttpRequest/fetch, а изгледите за различните елементи се генерират динамично в монолитичния скрипт loggedin.js.

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

Галин Симеонов – работи главно по backend-а. Разработи файловата система, структурата на базата данни, както и част от API endpoint-овете.

Александър Витков – работи главно по frontend-а, и по част от PHP API-то, с което frontend-а директно комуникира.

## 10. Какво научих

Галин Симеонов – ползата от съставянето на добра архитектура, преди да станат проблемите.

Александър Витков – научих, че мога да напиша 300 реда JS за 25 минути, което по мой сметки се приравнява на 120 хиляди реда седнично. Шегата настрана, разбрах доста от проблемите с Browser API-тата за динамично генериране на HTML, и защо хората избират да ползват frontend frameworks.

## 11. Използвани източници

<https://www.php.net/docs.php> - документация на PHP

<https://dev.mysql.com/doc/> - документация на MySQL

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects> - документация на различни Browser API-та

<https://www.novabrasilia.bg/>

Предал (подпис): ………………………….………………………….

………………………….………………………….

/*фн, имена, спец., група*/

Приел (подпис): ………………………….

/доц. *Милен Петров*/