

Имена: Димитър Николаев Татарски	ФН: 81818	
Начална година: 2023	Програма: бакалавър, КН	Курс: 4
Тема: 7.1 Filemeup		
Дата: 24.01.2023	Предмет: w19prj_KN_final	имейл: tatarski.dim@gmail.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 9999_project_final.zip.** (Успех!). (Редактирайте **маркираните зони** в жълто с коректната информация)

ТЕМА: 7.1 Filemeup

1. Условие

Система за качване и преглед на различни видове ресурси.

2. Въведение

Системата позволява създаването на акаунт и качването в лична директория. Потребителя може да създава директории, да качва файлове и да разглежда вече записаните директории и файлове.

3. Теория

Системата работи на базата на архитектурата Model View Controller.

Моделите (Model) са интерфейси за достъп до базата данни/ записаните файлове. През тях се извършват действия като извличане на съдържание на поискана от директория/ създаване на потребител/ извличане на информация за потребителите/ извличане на информация за файловете.

Контролерите (Controller) управляват обработката на заявки към системата. Това са действия като: валидация на полета при попълване на форма, качване на файл, регистрация, логин, преглед на файл.

Изгледите (View) предоставят информация към потребителя в html формат. В тях се генерират html файлове за различни цели: форма за регистрация/ изглед към файлове в една директория/ меню за навигация.

Всичко това бива комбинирано от файла index.php, който в зависимост от параметъра на заявката ?route се избира, които изгледи да бъдат включени.

4. Използвани технологии

1. Използван е XAMPP за основна технология.
2. **СУБД:** MySQL
3. **PHP Server:** Apache
4. **Front end:** html страници, стилизирани със **css**. Без **javascript**.

5. Инсталация и настройки

1. Архивът се разархивира и директорията filemeup се копира в директорията C:\\xampp\\htdocs

2. Конфигурационните файлове за хатрр се намират в filemeup\setup
3. SQL script за създаване и запълване с тестови данни на базата данни: filemeup\setup\db_create.sql
4. Директорията filemeup\tmp_init съдържа началното съдържание на директориите на двата тестови потребителя. Тя трябва да бъде копирана в директорията filemeup\tmp
5. Във файла filemeup\config.php се намират конфигурационни променливи за:
 - Базов url на уебсайта
 - Базова директория за уебсайта
 - Информация за създателя, за уебсайта
 - Потребителско име и парола за създаване връзка с базата данни

6. Кратко ръководство на потребителя

1. Регистрация

Попълнете формата за регистрация. При наличието на грешки – изпълнете инструкциите, които ще се покажат на екрана за да бъдат валидни попълнените данни.

Filemeup

Регистрация

Потребителско име
ADMIN

Имейл
Imeyl

Имена
Имено

Парола

Повторете паролата
Повторете паролата

Register

Логин

Потребителско име
ADMIN

Парола

Login

Пример за невалидно попълнена форма

Filemeup

Регистрация

Потребителско име
ADMIN

Имейл
qin@qin.com

Имена
qin

Парола

Повторете паролата

Въведете валиден имейл адрес.

Паролата трябва да съдържа поне една малка латинска буква, главна латинска буква и цифра.

Повторете паролата

Паролите трябва да съпадат.

2.>Login

Попълнете формата за логин. При наличието на грешки – оправеете входните данни.

The screenshot shows two yellow forms on a light gray background. The left form is titled 'Регистрация' (Registration) and contains fields for 'Потребителско име' (Username) with 'ADMIN' entered, 'Имейл' (Email) with 'me@me', 'Имена' (Names) with 'me@me', 'Парола' (Password) with '*****', and 'Повторете паролата' (Repeat password) with '*****'. A 'Register' button is at the bottom. The right form is titled 'Логин' (Login) and contains fields for 'Потребителско име' (Username) with 'ADMIN' entered and 'Парола' (Password) with '*****'. A 'Login' button is at the bottom. Two red arrows point to the password fields in both forms.

Грешка при логин:

The screenshot shows the 'Логин' (Login) form with the 'Потребителско име' (Username) field containing 'ADMIN' and the 'Парола' (Password) field containing '***'. A red error message 'Неправилна парола.' (Incorrect password.) is displayed below the password field. The 'Login' button is at the bottom.

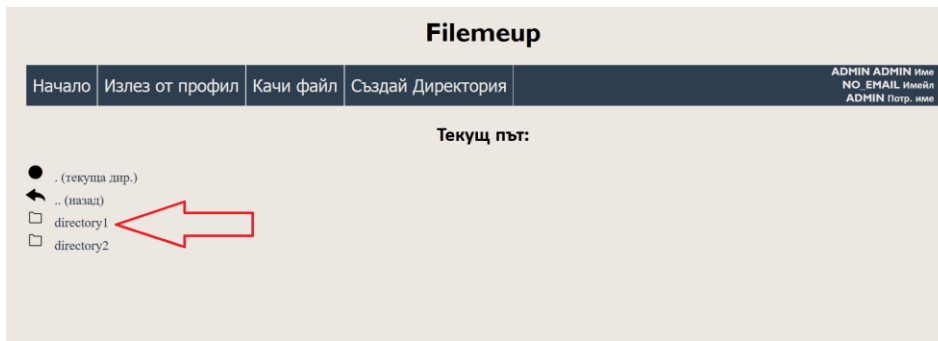
3.Изглед на директория

При успешно влизане в профила – показва се изгледа на директория. Там ще видите всички качени до момента файлове. След регистрация тази директория ще е празна.

The screenshot shows the 'Filemeup' directory view. At the top, there is a navigation bar with links: 'Начало' (Home), 'Излез от профил' (Logout), 'Качи файл' (Upload file), and 'Създай Директория' (Create Directory). On the right, there is a table with columns: 'ADMIN', 'ADMIN', 'Имейл', 'NO', 'EMAIL', 'Имейл', 'ADMIN', 'Потр.', 'Имейл'. Below the navigation bar, there is a section titled 'Текущ път:' (Current path:). Under this section, there is a list of items: a black dot representing the current directory, a back arrow representing the parent directory, and two directories named 'directory1' and 'directory2'.

4. Навигация в директория

- Натиснете върху директория за да влезете в нея.

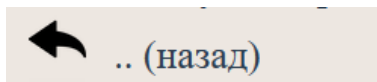


Резултат:

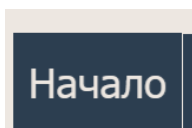


Различните видове файлове имат икони, представящи типа им.

- Избор на родителска директория

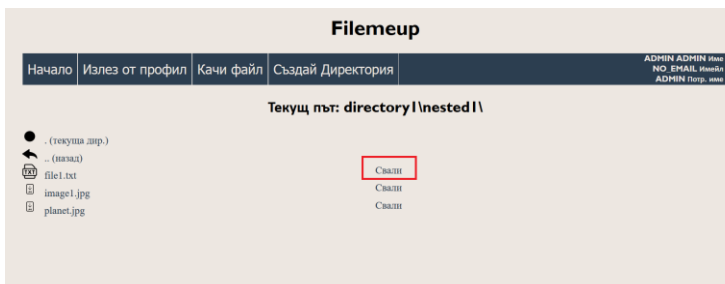


- Избор на начална директория



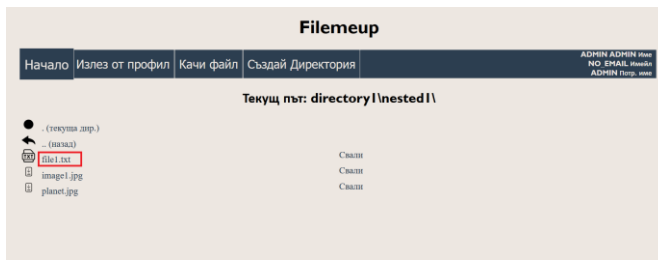
- Сваляне на файл

При наличието на файл, който не е директория – има възможност за неговото изтегляне.

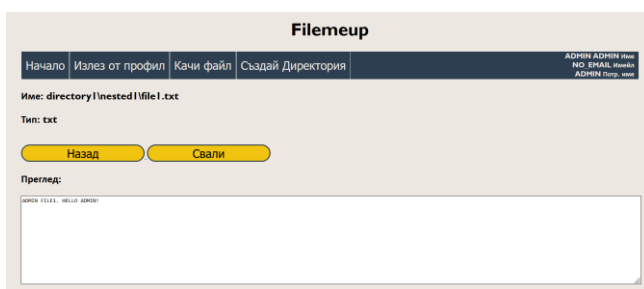


5. Изглед към файл

При избор на името на файл – показва се изглед към конкретния файл. Прави се опит за преглед на файла.



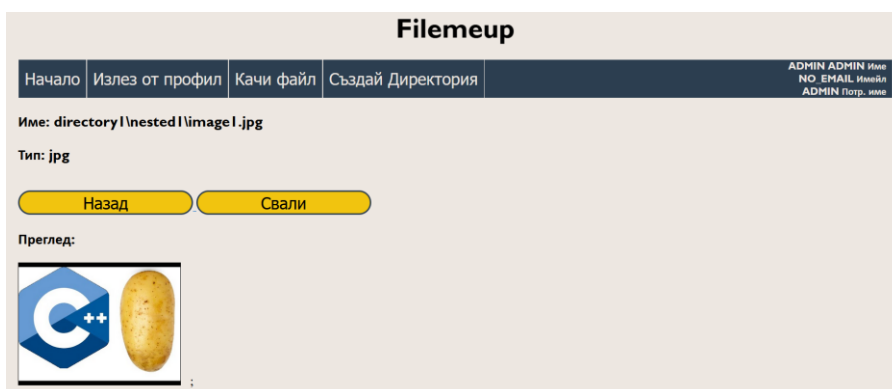
Резултат:



- Пример 2:

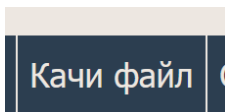


Резултат:

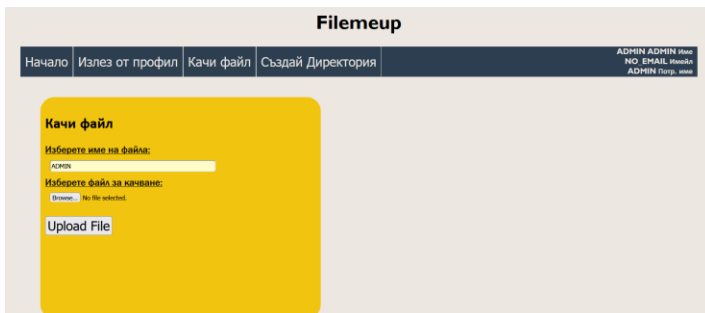


6. Качване на файл

- Изгледа за качване на файл се отваря със следния бутон от навигацията:



Резултат:

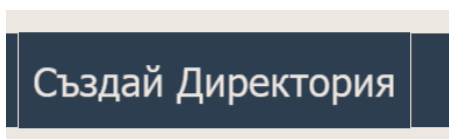


Тук потребителя може да избере файл, който да качи и неговото име. Ако полето за името остане празно, то файлът ще се запази с името, с което е качен.

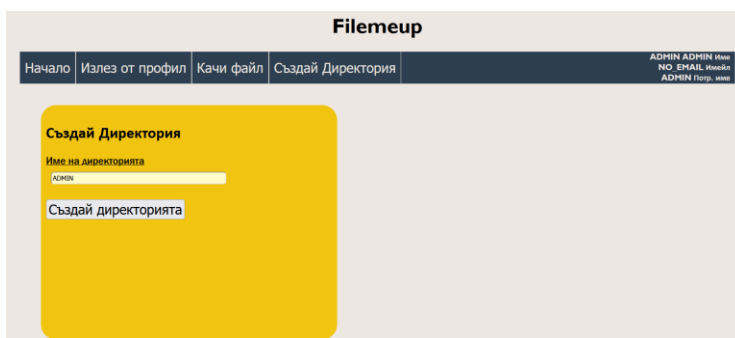
След качване – потребителя бива пренасочен в директорията, в която се намира новия файл.

7. Създаване на директория

- Изгледа за създаване на директория се достъпва чрез следния бутон

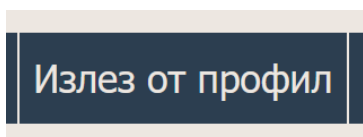


Резултат:



При попълване на валидно име – създава се директория в директорията, в която потребителя се е намирал, при натискане на бутона от менюто за навигация

8. Излизане от профила – потребител може да излезе от профила си със следния бутон:



7. Примерни данни

Създадени са 2 тестови потребителя:

- Username: ADMIN
- Password: ADMIN123

Има роля Admin и роля User.

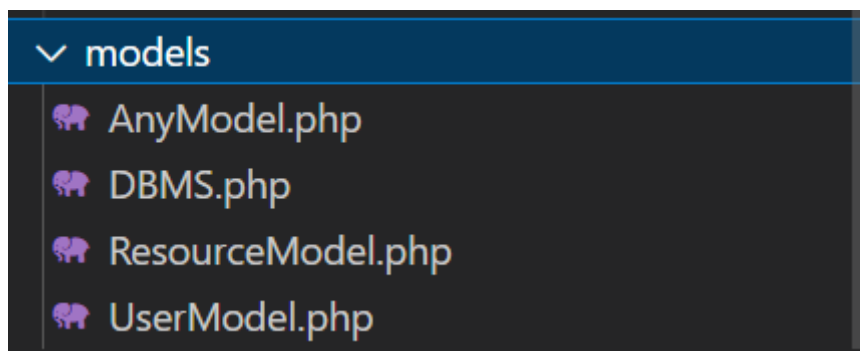
- Username: TEST
- Password: TEST123

Има роля User.

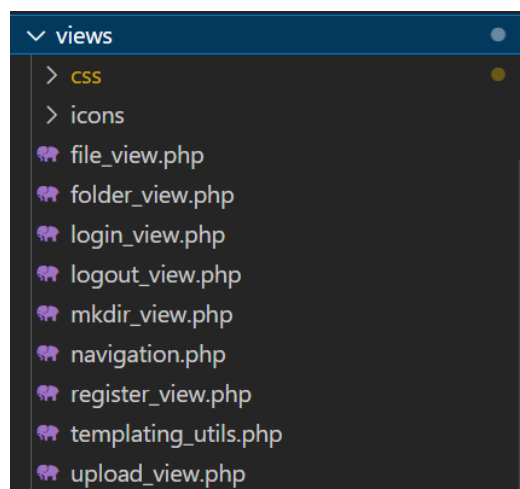
8. Описание на програмния код

3 основни директории:

- Model
- View
- Controller

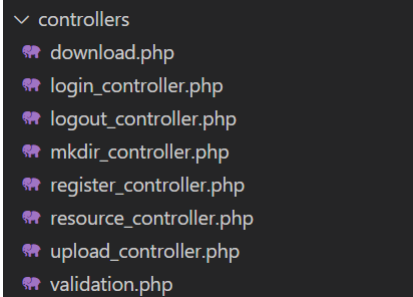


- **AnyModel.php** – абстрактен статичен клас, който бива наследяван от другите модели. Има базови функции за заявки към базата данни.
- **DBMS.php** – осъществява връзка с базата данни и предоставя достъп до нея
- **ResourceModel.php** – функции, работещи с файлове и директории (ресурси).
- **UserModel.php** – функции, свързани с извличане и запазване на информация, свързана с потребителите



- **register_view/ login_view** – изгледи за регистрация и логин
- **file_view** – изглед към единичен файл
- **folder_view** – изглед към директория

- **logout_view** – бутон за излизане от профила
- **mkdir_view** – изглед за създаване на директория
- **navigation** – навигационния панел, съдържащ главните бутони
- **templating_utils** – помощни функции за създаване на форми
- **upload_view** – изглед за качване на файл



- **download.php** – php script за изтегляне на файл от потребителя
- **login_controller/ register_controller** – логиката за регистрация и login
- **logout_controller** – логика за logout
- **mkdir_controller** – контролер за създаване на директория
- **resource_controller** – контролер за изгледите, които боравят с директории/ файлове
- **upload_controller** – качване на файл от потребителя
- **validation.php** – помощни функции за валидация на данни от форми

- **config.php** – конфигурационни данни
- **index.php** – индексен файл

При зареждането на индексен файл, в зависимост от параметъра на заявката **?route** се включват съответните изгледи към основната html страница.

Запазването на път към файл, когато се превключва между страници, става чрез **?path** параметъра на заявката. В зависимост от това каква е стойността на **?path**, при качване на файл – този файл ще бъде качен различна директория.

При наличието на грешки при валидация на форма – съобщенията за грешки автоматично се събират и се изпращат в **?errors** параметъра на заявката.

Генерирането на форми става автоматично, чрез наличието на **form_fields** конфигурационни променлива, която съдържа информация за полетата на формата и техните атрибути, описания, стойности. Базовата функционалност е имплементирана в **views/templating_utils.php**

Валидацията на форми става автоматично, чрез **validation_data**, променлива, която пази имената на полетата и списък от функции за валидация и съобщения за грешка. Основната функционалност е в **controllers/validation.php**

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

- Възможност за избиране на индексен файл (избира се html файл и се получава линк за неговото достъпване). При посещения на линка – автоматично се зареждат файловете, реферирани от **html** файла.
- Възможност за администрация на потребители
- Възможност за създаване на шаблони за проекти

10. Какво научих

Много в областта на разработката на уеб приложения с **xampp**. Запознах се с **php**.

11. Използвани източници

- <https://www.php.net/manual/en/index.php>
- <https://www.w3schools.com>

12. Добавена функционалност след първата версия

- **автоматично разархивиране на zip архиви**

Потребителите имат възможно да изберат качен архив да бъде разархивиран автоматично и да бъде включен в структурата на виртуалната му директория.



- **допълнително описание за всеки файл при изгледа на файловете**

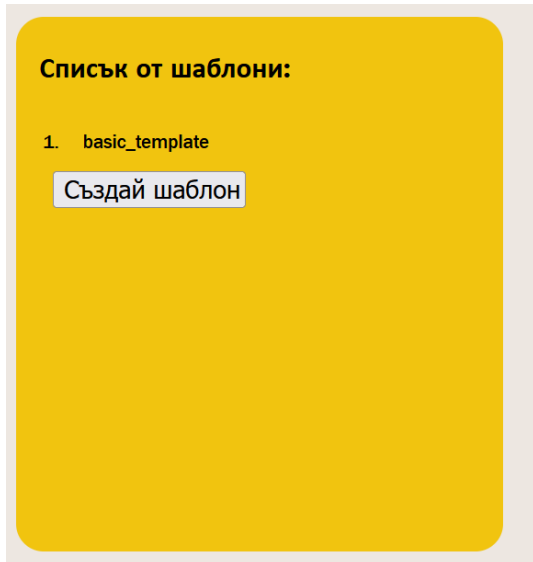
Добавена е информация за размера и типа на файловете. Освен това е добавено меню за избор на шаблон, спрямо който да се тества някоя директория. Ако подструктурата на тази директория пасва на шаблона, то се изписва съответно съобщение на екрана.



Име	Размер	Тип	Шаблон
• (текуща дир.)			
← .. (назад)			
basedir			basic_template Go ПАСВА Изтрий
directory1			Няма шаблон Go Изтрий
directory2			Няма шаблон Go Изтрий

- **възможност за създаване на шаблони**

Създадена е функционалност за списък от всички създадени шаблони. Шаблон описва как да трябва да бъде структурирана една директория. Предоставена е функционалност за тестване дали една директория пасва на шаблон.



В менюто за създаване на шаблон се въвежда неговото име и освен това се въвежда и JSON текст, който описва шаблона в следния формат:

- на ниво директория/ файл има **regex**, който описва какво може да бъде името на файла/директорията (**regex**)
- всяката директория може да съдържа много обекти за директории, които съответно си имат в масива (**children**). Всяко дете се получава по описаната схема
- Ако един обект има **regex**, но няма **children**, то той описва файл, а не директория

Примерна структура:

```
{
  "regex": "(root|basedir)",
  "children": [
    {
      "regex": "subdir[0-9]",
      "children": [
        {
          "regex": "files[0-9]\\.(png|jpeg)"
        }
      ]
    },
    {
      "regex": "README.txt"
    }
  ]
}
```

Меню за създаване на шаблон:

Създай шаблон

Име на шаблона

Име на шаблона

JSON структура на шаблона

JSON структура на шаблона

Създай Шаблон

Меню за разглеждане на шаблон:

Име на шаблон: basic_template

Структура:

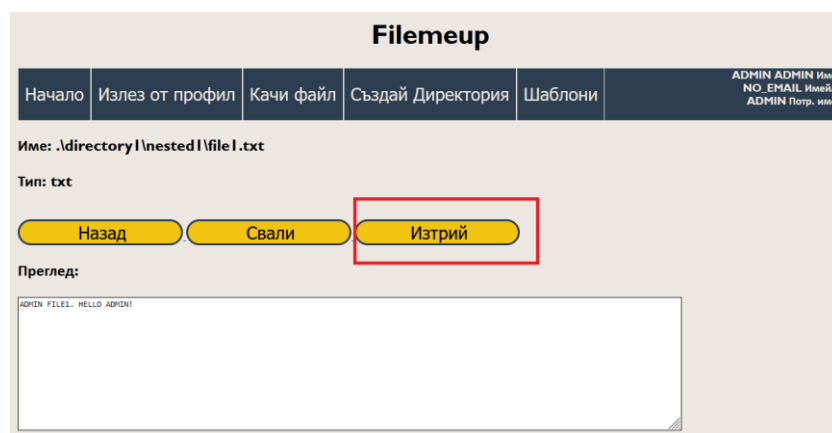
```
(root|basedir)
  subdir[0-9]
    files[0-9].(png|jpeg)
  README.txt
```

Назад

Съдържание:

```
{
  "regex": "(root|basedir)",
  "children": [
    {
      "regex": "subdir[0-9]",
      "children": [
        {
          "regex": "files[0-9]\\.(png|jpeg)"
        }
      ]
    },
    {
      "regex": "README.txt"
    }
  ]
}
```

- функционалност за изтриване на файлове – възможно е файл или директория да бъде изтрит при натискане на бутона **Изтрий** през изгледа на директории или изглед на конкретен файл



- config_validation.php с конфигурация на валидациите на форми
- функционалност за валидация на подструктура на директория според някакъв шаблон. След избиране на съответен шаблон и натискане на Go се извършва проверка върху избраната директория. Проверява се дали подструктурата на директорията пасва на шаблона (дали всички файлове, които са в директорията имат описание в шаблона).

basedir	basic_template	Go	ПАСВА	Изтрий
directory1	basic_template	Go	НЕ ПАСВА	Изтрий

Предал (подпис):

/Димитър Татарски, КН, 4курс, 81818, 7 група/

Приел (подпис):

/доц. Милен Петров/