**Имена: Боряна Начева, ФН: 62105Начална година: 2018/2019, Програма: бакалавър(СИ), Курс: 3   
Тема: FileMeUp  
Дата: 2019-07-06, Предмет: wwwTech2018\_19\_12ed\_SI\_summer, имейл: bori97@abv.bg**

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

# ТЕМА: FileMeUp

## Условие

Уеб базиран сървър за споделяне на файлове (документи, музика, филми). Евентуално със защитено REST API.

## Въведение

Уеб базиран система за споделяне на различни видове файлове. Приложението е базирано на арх. стил REST. Приложението е Single Page Application. Като основни функционалности то съдържа регистрация (като администратор и обикновен потребител), вход, изход, промяна на потребителските данни, качване на файлове в различни формати, достъп до собствените файлове на потребителя и до файлове, до които потребителят има достъп, преглед на файлове. За правата и достъпа на потребителите в системата използвах т.нар. Json Web Token, което си го обяснявам като еквивалента на сесиите, но в случая, тъй като приложението ми е на базата на REST арх. стил, трябва да е stateless. Използва се просто една допълнителна библиотека и другите неща са стандартни, дори в базата никъде не се записва token, той се съдържа в header-a на заявката като метаданни. Приложението може да използва локална база, или да използва база, която е хостната на AWS машина. За тази цел разбира се инстанцията трябва да е работеща.

## Теория

**REST** е стил софтуерна архитектура за реализация на уеб услуги. Основната идея е да се определи системен ресурс, който се променя в резултат на взаимодействието между доставчика на услуги и потребителя. Архитектурният модел REST включва взаимодействията между сървър и клиент, осъществени по време на трансфера на данни. Концепцията беше въведена за пръв път от Рой Филдинг през 2000 г. като част от неговата докторска дисертация. Филдинг е един от основните автори на HTTP протокола, под който се изпълняват REST имплементациите в повечето случаи.

**SON Web Token** (JWT) е JSON-базиран отворен стандарт за създаване на „жетони“, които съдържат определен брой твърдения. Например, един сървър може да генерира жетон, който съдържа твърдението „влязъл като администратор“ и достави тази информация до клиент. Клиентът тогава може да използва това означение да докаже, че е влязъл като администратор. Жетоните са подписани от ключа на сървъра, така че сървърът да е в състояние да удостовери, че подписът е законен. Жетоните са проектирани да бъдат компактни, URL-безопасни и използваеми най-вече в уеб браузър единично влизане (SSO) контекст. Твърденията JWT обикновено се използват, за да се предаде самоличността на регистрирани потребители между доставчик на идентичност и доставчик на услуга, или всякакъв друг вид твърдения, както се изисква от бизнес процесите. Жетоните могат да бъдат потвърдени и криптирани.

## Използвани технологии

За реализиране на системата се използва:

- **XAMPP v3.2.2** – пакет за вдигане на локален сървър и база данни. **XAMPP** е безплатен и свободен софтуерен пакет, който съдържа Apache HTTP Server, MySQL база от данни и други необходими инструменти за използване на езиците за програмиране PHP, Perl.

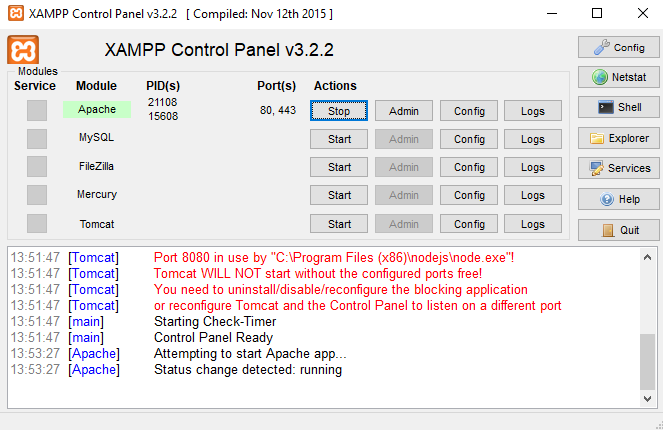
- **Visual Studio Code 2019** – среда за разработка на приложението. Използвала съм тази среда, защото е доста лека, в сравнение с други, лесно се инсталира и лесно се конфигурира да работи с различни езици за програмиране.

- **Postman 7.2.0** – тестване на endpoints на приложението. Тъй като приложението ми е базирано на REST архитектурния стил, съм използвала Postman за тестване на всички ендпойнти в приложението.

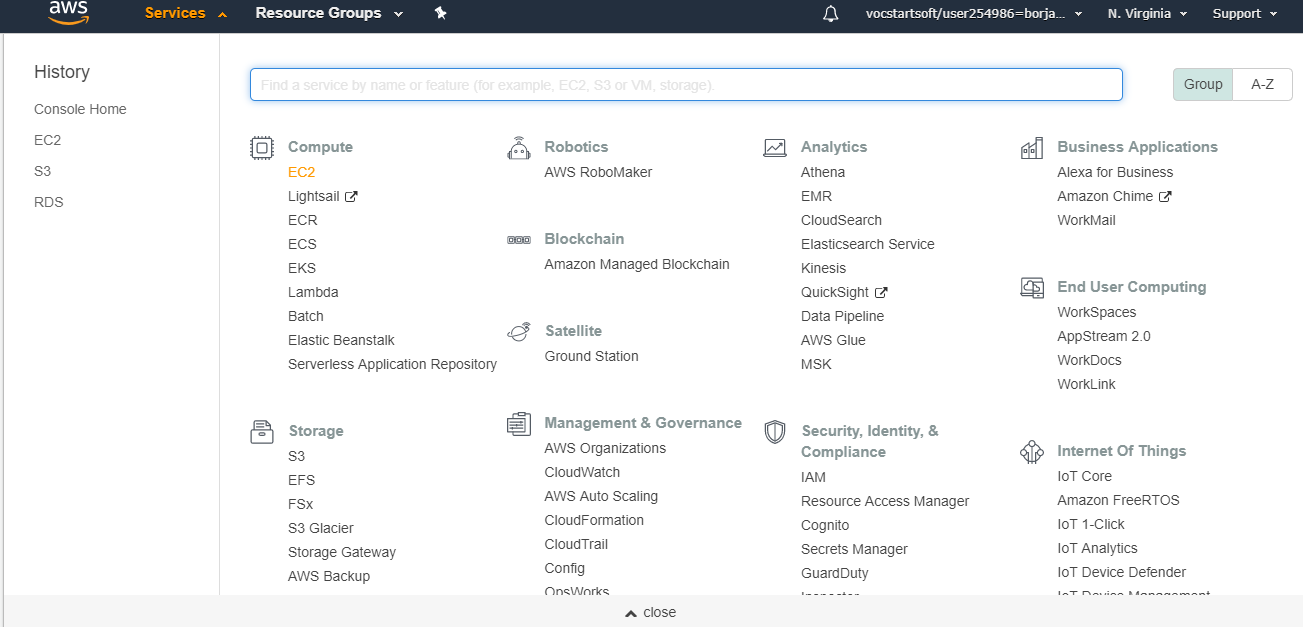
- **AWS EC2** – хостване на база, която ще се използва от приложението. От профилът ми в AWS съм създала нова инстанция под Linux. За да обясня по-разбираемо какво се случва с хостването на база на тази машина, ще започна с обяснение на създаване на база на локалната машина. Както вече обясних, използването на XAMPP ни дава възможност да стартираме база на локалната ни машина, като я завържем за определен порт, на който да работи. Същото нещо трябва да направим и на машината на AWS. За тази цел трябва да инсталираме mysql там. В следващата точка има подробно обяснение за това. Тук трудностите, които срещнах са: първоначално се опитах да инсталирам целия XAMPP пакет на машината на AWS, след това да стартирам базата и да се вържа към приложението ми, но базата не успяваше да се върже на някакъв порт и да слуша на него, съответно не можех да си вържа и системата към тази база.

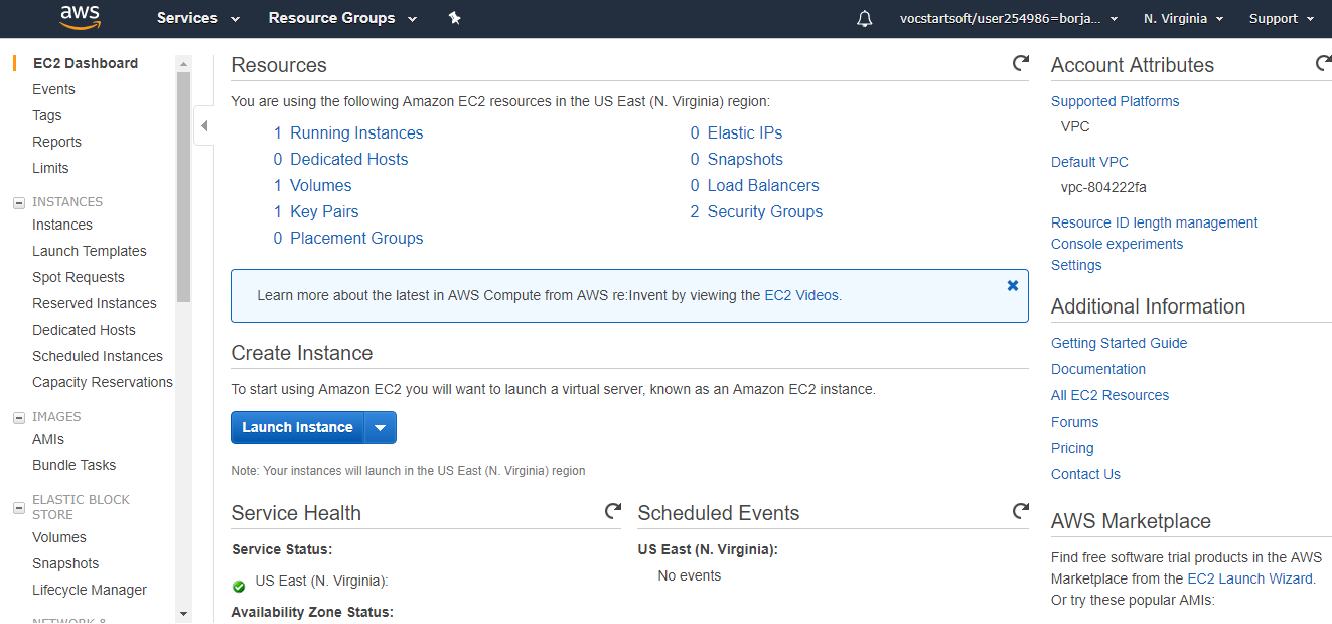
## Инсталация и настройки

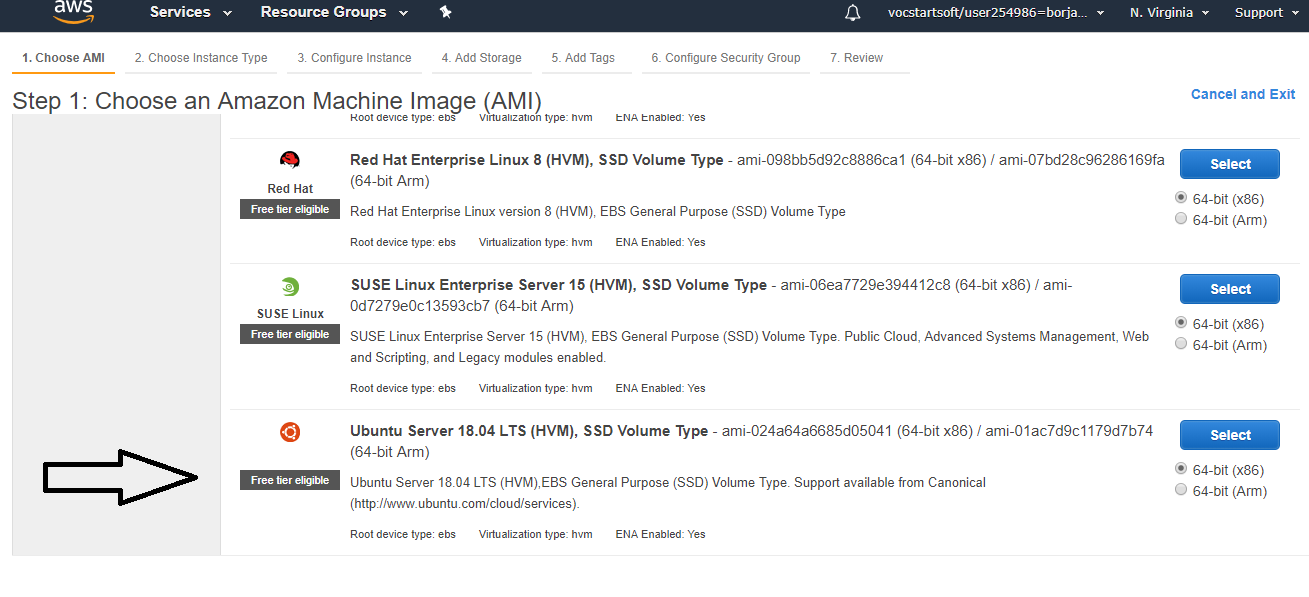
1. Инсталация на XAMPP: <https://www.apachefriends.org/index.html> - от тук може да се инсталира XAMPP, като стъпките за инсталацията са доста лесни, просто next -> next -> next. Важно е само програмата да се сложи в папка C:\ на вашата машина, за да се избегнат бъдещи грешки. Както обясних по-нагоре, приложението може да използва локална база и база, която е хостната на AWS машина. При всички положения, за да се вдигне локален сървър папката с файлове трябва да се разархивира и да се сложи в папката htdocs на XAMPP. След това се включва XAMPP и се стартира Apache Server.

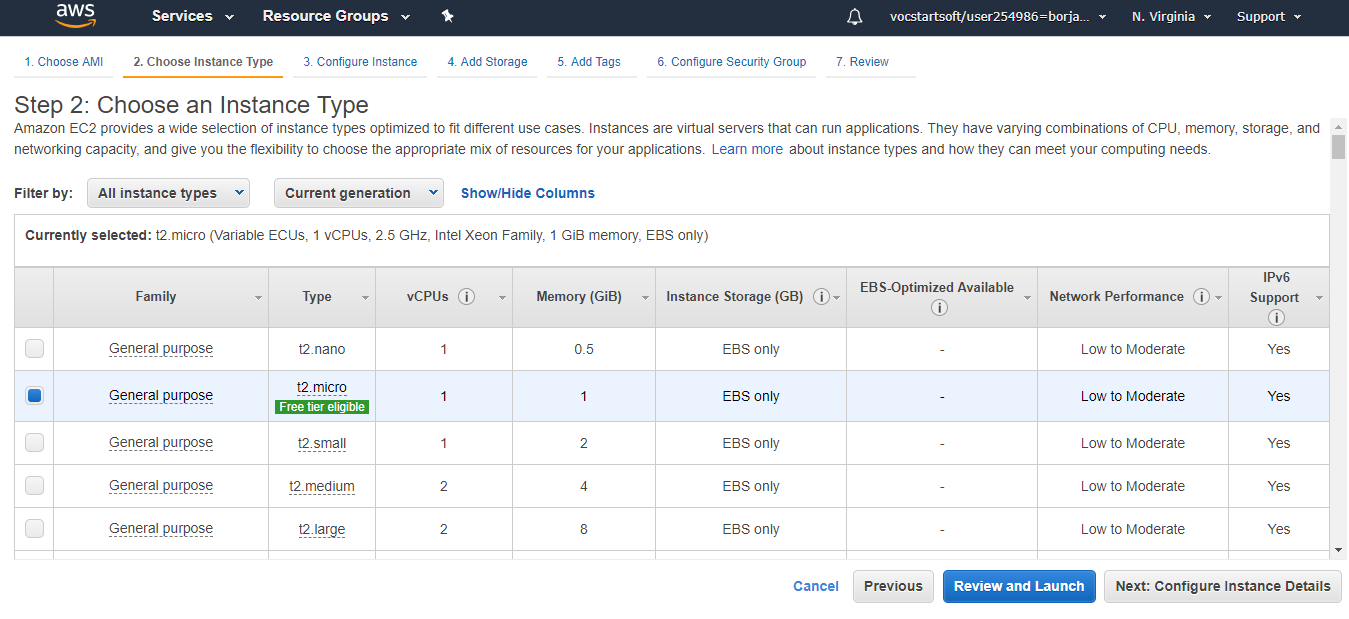


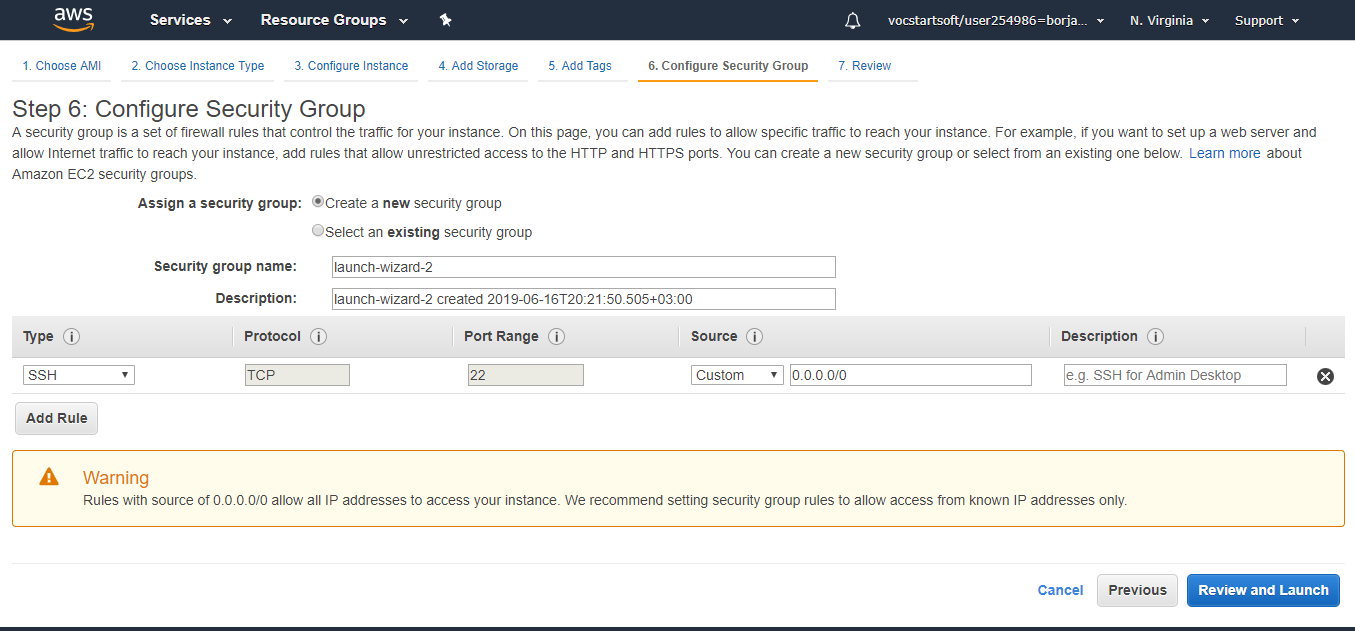
1. Инсталация на Visual Studio Code: <https://code.visualstudio.com/download> - от тук може да се инсталира Visual Studio Code.
2. Инсталация на Postman - <https://www.getpostman.com/downloads/> - от тук може да се инсталира Postman.
3. AWS EC2 – За хостване на базата на машина на AWS съм използвала EC2 инстанция. Следват стъпки за създаването на тази инстанция.











Преди да създадем инстанцията е важно да отбележим, че можем да използваме вече създаден ключ, чрез който ще се вържем към самата инстанция през терминала, или да създадем нов. Ако създадем нов ключ, трябва да го свалим. За да се вържа през терминала към съответната инстанция, аз лично използвам PuТТy.

След като се влезне в машината на AWS през терминала се изпълняват последователно следните команди:

1. sudo su – за root права
2. sudo apt-get install mysql-server – за инсталиране на mysql на машината на AWS
3. cd /ect/mysql/mysql.conf.d – за да се променят данните в следващия файл
4. sudo nano mysqld.cnf – тук трябва да се промени bind-adress = 0.0.0.0, за да слуша на всички портове
5. sudo service mysql restart – рестартиране
6. sudo mysql – за влизане в базата, тук може да се манипулират таблици, права на потребители и т.н., с подходящите заявки
7. sudo netstat -tulpn | grep LISTEN – проверка на слушащи портове

## Кратко ръководство на потребителя

Както обясних по-нагоре, приложението може да използва локална база и база, която е хостната на AWS машина. Базата се вдига от самото приложение. За всеки случай, в папката на проекта има и прикачен скрипт, който може директно да се импортне и да се използва наготово. При всички положения, за да се вдигне локален сървър папката с файлове трябва да се разархивира и да се сложи в папката htdocs на XAMPP. Стартира се XAMPP. Това дали приложението да използва локална база, или базата хостната на AWS машината се прави през кода на приложението. От там се вдига и самата база + таблиците в нея. Тоест от потребителя се изисква само да сложи папката с проекта в папката htdocs на xampp, да стартира XAMPP и да използва приложението. В отделен файл от проекта ще бъдат записани и данните на потребителя на локалната машина и на потребителя на машината на AWS.

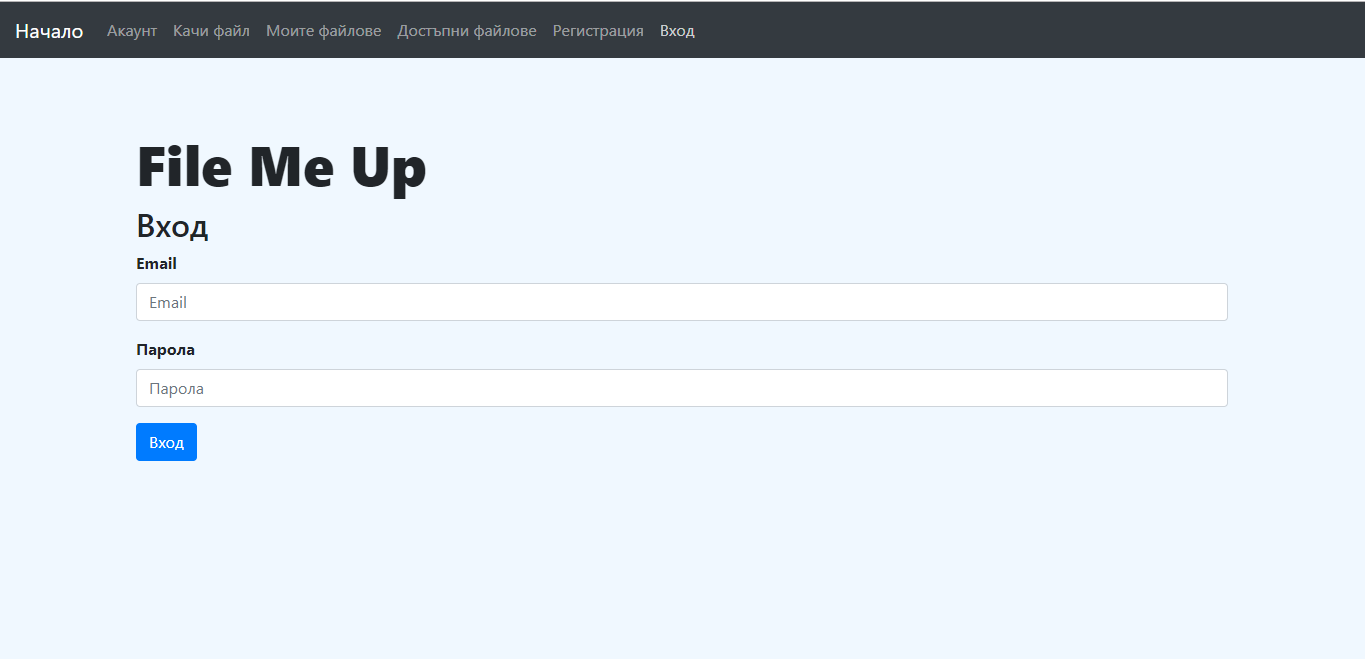
## Примерни данни

За да използвате приложението, трябва първо да се регистрирате като потребител и след това да влезнете в системата. Без да сте влезнали в системата не може да използвате нито една от функционалностите й. Приложението работи и с предварително създаден администраторски профил, който се създава всеки път при вдигане на приложението. Данни – потребителско име: [bori97@abv.bg](mailto:bori97@abv.bg), парола: 123456. Тук е мястото да кажа, че приложението работи с криптирани пароли, т.е. паролите на потребителите се запазват в базата в криптиран вид.

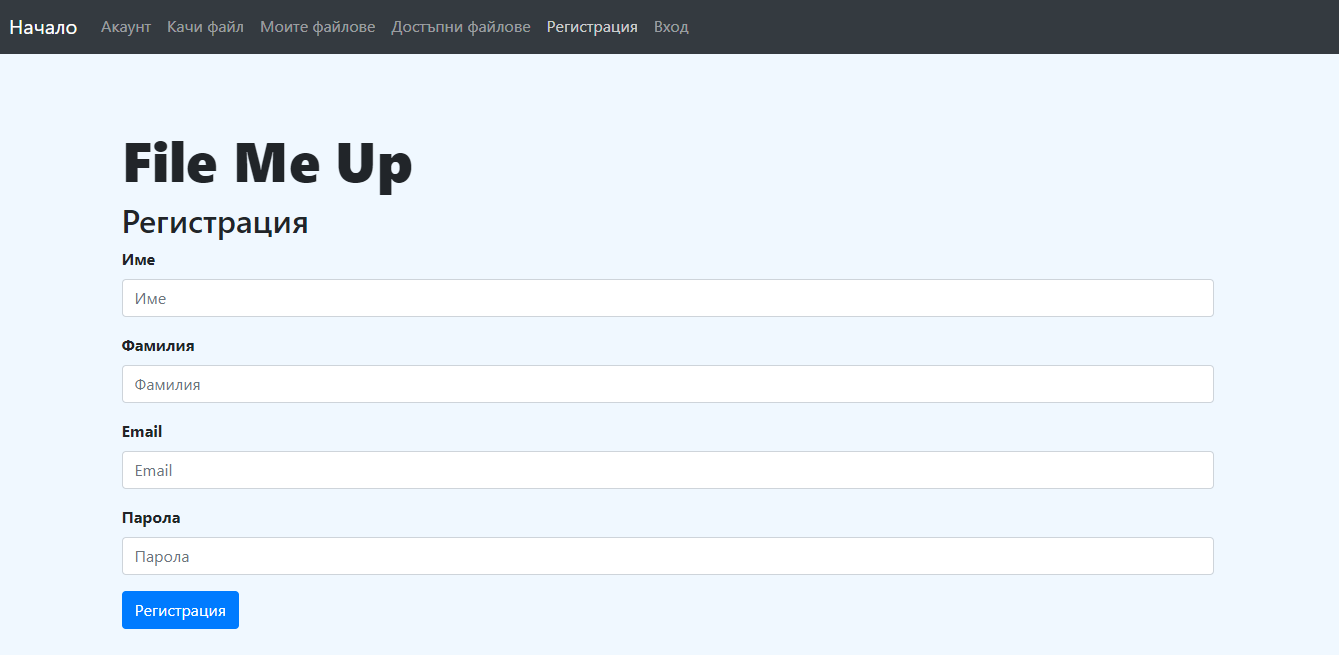
## Описание на програмния код

Кодът се разделя в няколко php скрипта, като всеки изпълнява определена функция, JavaScript скрипт за комуникацията между frontend-а и backend-а на приложението, css файл с код, който описва стилизирането на потребителския интерфейс и html файл, който описва страницата, в която ще се рендерира цялата информация. Следват и няколко user-guide снимки, които нагледно представят какво се случва след интерпретирането на кода:

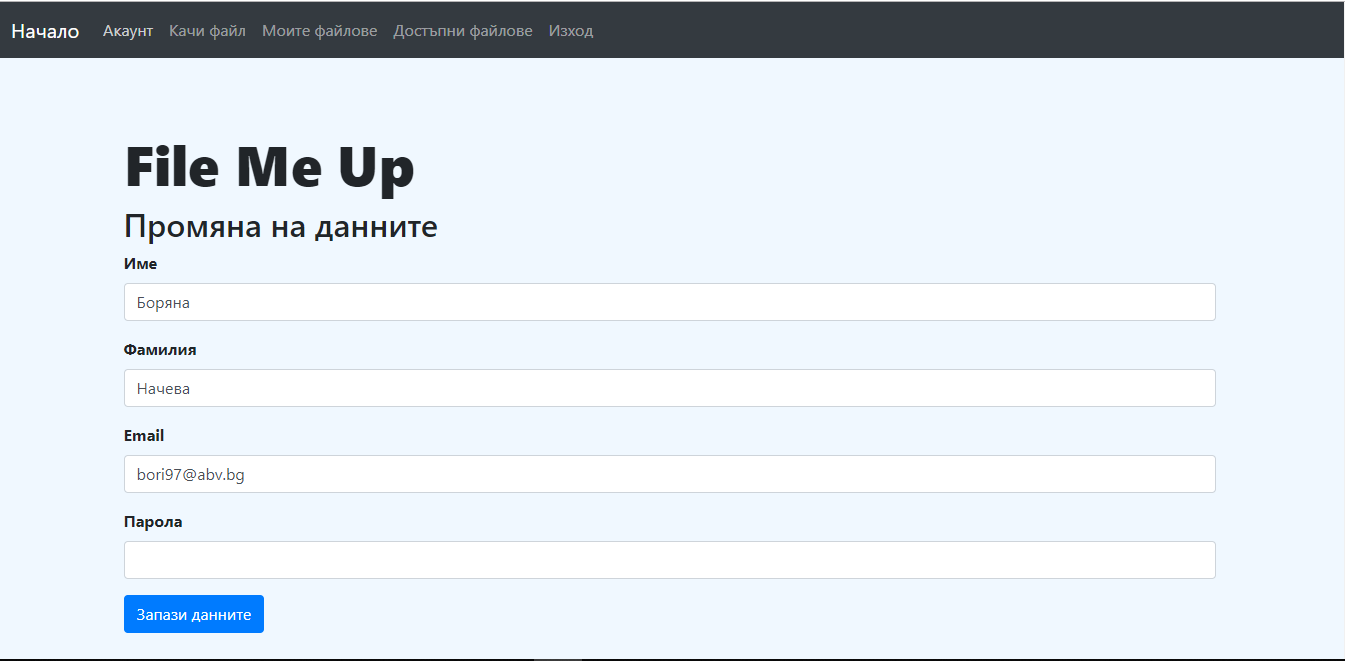
1. Login



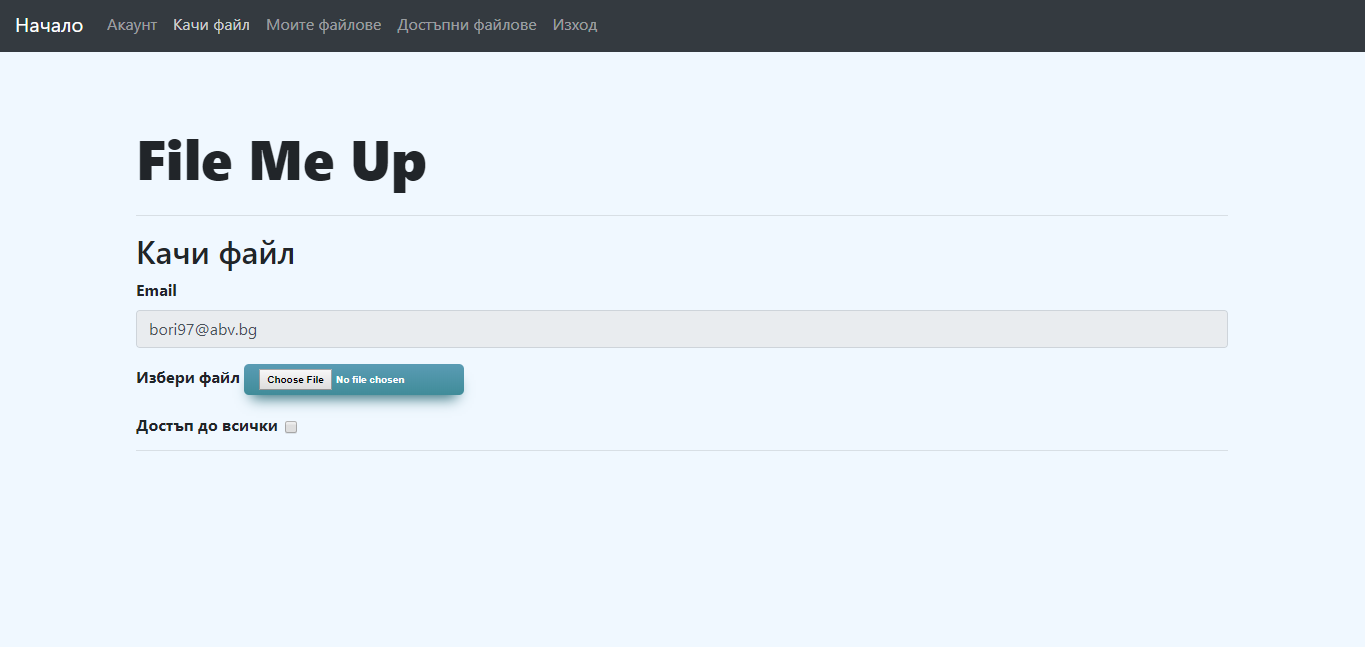
1. Register



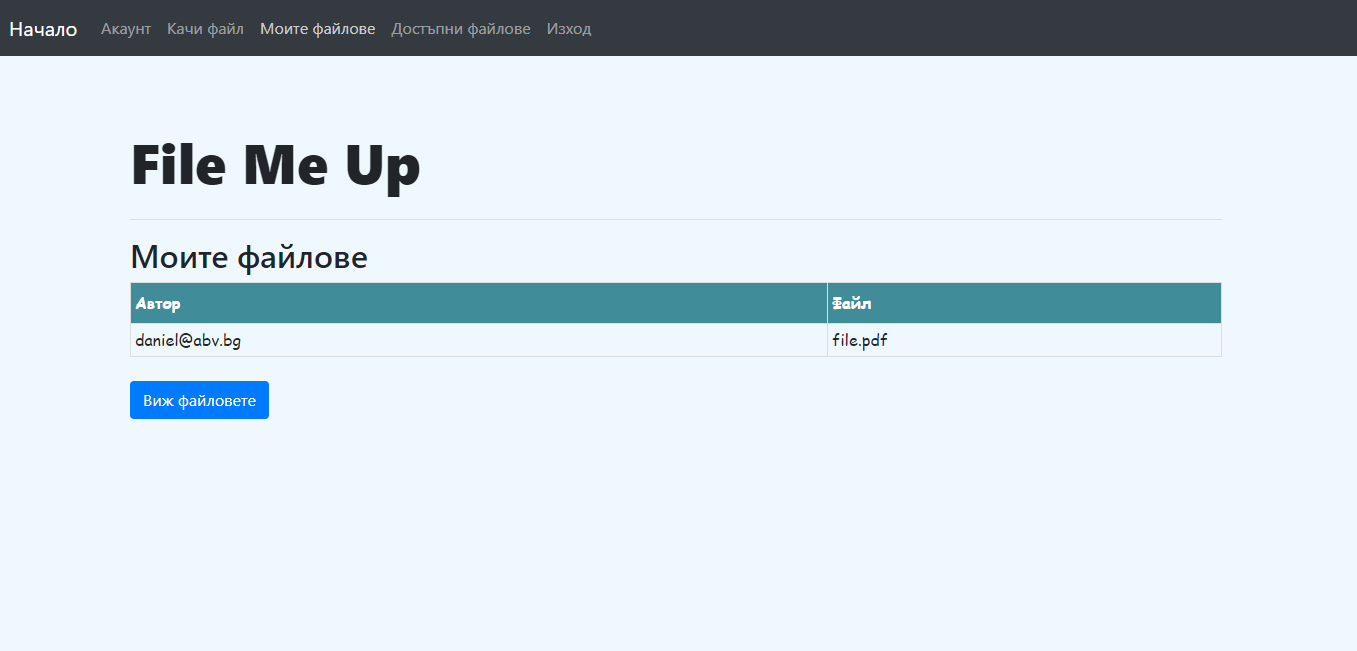
1. Update profile



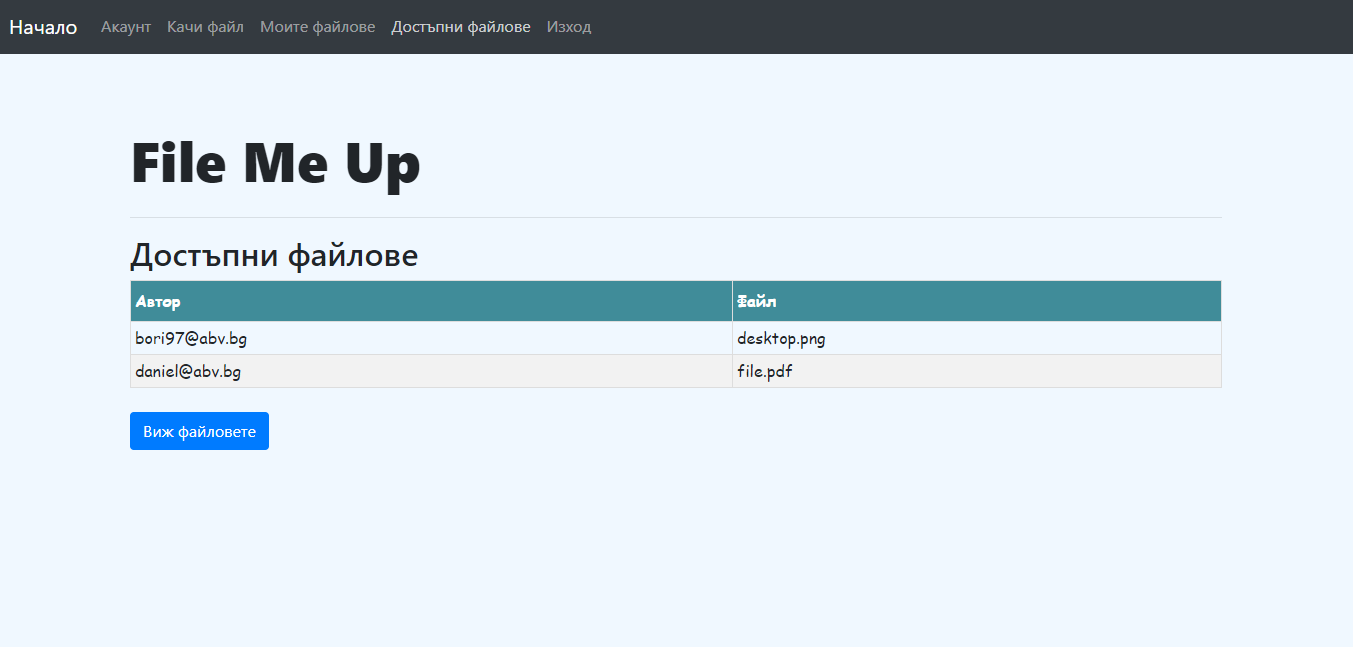
1. Upload file



1. Show my files



1. Show all files



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

В момента приложението може да използва локална база и база, която е хостната на AWS машина. За бъдеще това може да се доразшири, ако цялото приложение се хостне на тази машина, или още по-добре – ако се използва и локална база, и база, хостната на AWS машина, като при ъпдейт на данни в локалната база, базата на AWS машината също се ъпдейтва и обратно. Също така, може да се добави функционалност за сваляне на файлове в различни формати.

## Какво научих

След завършването на проекта мога да кажа, че научих доста неща, използвах нови технологии, които ми бяха доста интересни, задълбах в нови и модерни методологии за разработването на софтуер с уеб насоченост и прочетох доста литература, която ми беше полезна и интересна. Смятам, че най-вече придобих познания в разработката на REST APIs, Single Page Applications, използването на AWS console.

## Използвани източници

[https://console.aws.amazon.com](https://console.aws.amazon.com/) – конзолата на AWS

<https://www.digitalocean.com/community/tutorials/how-to-create-a-new-user-and-grant-permissions-in-mysql> - създаване на права на потребител за използване на mysql

<https://coolestguidesontheplanet.com/start-stop-mysql-from-the-command-line-terminal-osx-linux/> - стартиране/спиране/рестартиране на mysql на машина

<https://jwt.io/introduction/> - официален сайт и подробна информация за Json Web Token

<https://bg.wikipedia.org/wiki/JSON_Web_Token> - теория за JWT

<https://bg.wikipedia.org/wiki/REST> - теория за REST арх. стил

Предал (подпис): ………………………….

/*фн, имена, спец., група*/

Приел (подпис): ………………………….

/доц. *Милен Петров*/