**Имена:** Мая Бораджиева, Петя Ляскалиева, Ралица Вунцова

**Фн:** 62335, 62297, 62312 **Начална година**: 2018

**Програма:** бакалавър, (СИ) **Дата:** 2021-06-16

**Курс:** 3  **Предмет**: WEB технологии

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

**Имейли**: [maya.boradzhieva@gmail.com](mailto:maya.boradzhieva@gmail.com), [petq667@gmail.com](mailto:petq667@gmail.com), [ralitsa.vun@gmail.com](mailto:ralitsa.vun@gmail.com)

**Тема:** Алумни клуб на СУ (4.2)

## **1. Условие**

Създаване на система за генериране на покани за участие в клуб на завършилите (алумни клуб) и автоматична регистрация (импорт/експорт) – по списъци от 2.1.3.

Потребителите трябва да могат:

* да редактират на своите данни – постоянни или временни GPS координати;
* да имат контакти;
* да определят област на видимост (от колеги от групата, от специалността, от всички останали).

Системата трябва да включва визуализация и комуникация между алумни, модератор и статистика (колко контакта има в системата, колко организирани срещи, семинари, компании, работни групи, курсове).

## **2. Въведение**

Една от най-разпространените и използвани функции на онлайн мрежата е комуникацията между хората. Това е причината за създаването на различни приложения и сайтове именно с тази цел. С оглед на настоящата ситуация на глобална пандемия, смятаме, че идеята за разработване на система, която да позволява на студентите да продължат комуникацията помежду си след завършването, е много добра.

Потребителите на системата автоматично получават регистрация. Като се логнат, те оставят своите координати, за да покажат на своите колеги къде се намират в момента. Именно това е главната идея - осигуряване на по-лесен и по-бърз начин на комуникация между завършили колеги.

**3. Теория**Системата “Алумни клуб на СУ” разработихме, приемайки, че завършилите студенти са получили вече потребителско име и парола. След влизане в системата те могат да променят тези данни, както и да добавят GPS координати и други лични данни.

Потребителите в системата имат възможността да публикуват пост, който да представлява покана за среща на випуска, годишнина или някакво друго събитие. Този пост има опция за избиране на област на видимост (с опции: колеги от групата, колеги от специалността, колеги от факултета, всички потребители на системата). Постът ще излезе в списък със събития в началната страница на другите потребители, ако те са в съответната област на видимост. Също така, потребителите могат да виждат своите постове в отделна секция - „Моите покани“, като имат опцията да изтриват.

Потребителят може да отбележи, че ще присъства на събитието. След като го е отбелязал, може и да откаже, ако размисли. Когато потребител се отбележи, че ще присъства, той се появява в списъка с потребители, които ще присъстват на събитието, който списък се визуализира в самия пост.

Друга функционалност на системата е визуализация на статистика (колко потребители има в системата, колко поста са публикувани в системата, графики за брой потребители по факултет, специалност и година на завършване). Тази функционалност е достъпна само от администратора. Също така, той има достъп до списък с потребителите на системата.

Освен това, потребителите могат да оставят своите координати, след което им се визуализира кои алумни са наблизо (само тези, които също са си оставили координатите). Визуализира се информация за алумните, които са на 5 км разстояние и на карта се посочва къде се намират те. Достъпът до Google Maps се осъществява посредством API key като в случая не ни е необходим.

## **4. Използвани технологии**

## Използваната архитектура за разработването на приложението е MVC (Model-View-Controller).

## За база данни се използва релационната база данни MySQL. Таблиците, които сме създали, съхраняват информация за потребителите и постовете.

## Потребителският интерфейс е разработен чрез HTML, CSS и JavaScript.

## Функционалностите на системата са разработени чрез PHP. Връзката с базата данни също се осъществява чрез PHP.

Controller частта от архитектурата осъществява извличането и обработката на данните в системата.

## Приложението е писано във Visual Studio Code.

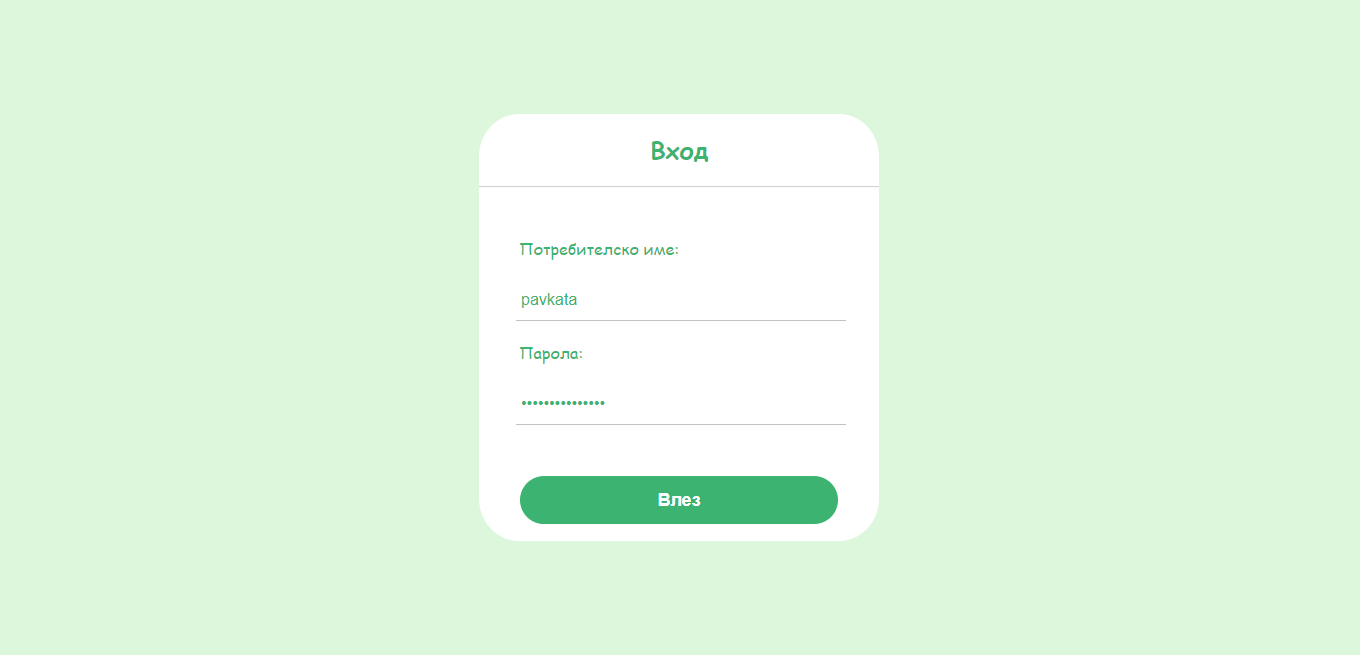
## За да участват всички в писането на кода, използвахме git и github.

## **5. Инсталация и настройки**

За да се достъпи системата и да се тества, е необходимо да се инсталира XAMPP (3.2.4). XAMPP е платформа, която позволява изграждането на сайт офлайн, на локален уеб сървър на нашия компютър. Чрез XAMPP могат да се пуснат сървъра и MySQL база от данни. Не е нужно да се инсталират други програми. phpMyAdmin е удобен UI за създаване на базата инеобходимите таблици. Файловете на проекта трябва да са поставени в директорията htdocs на XAMPP. Приложението се стартира в браузъра чрез изписване на localhost и път на файла спрямо htdocs директорията.

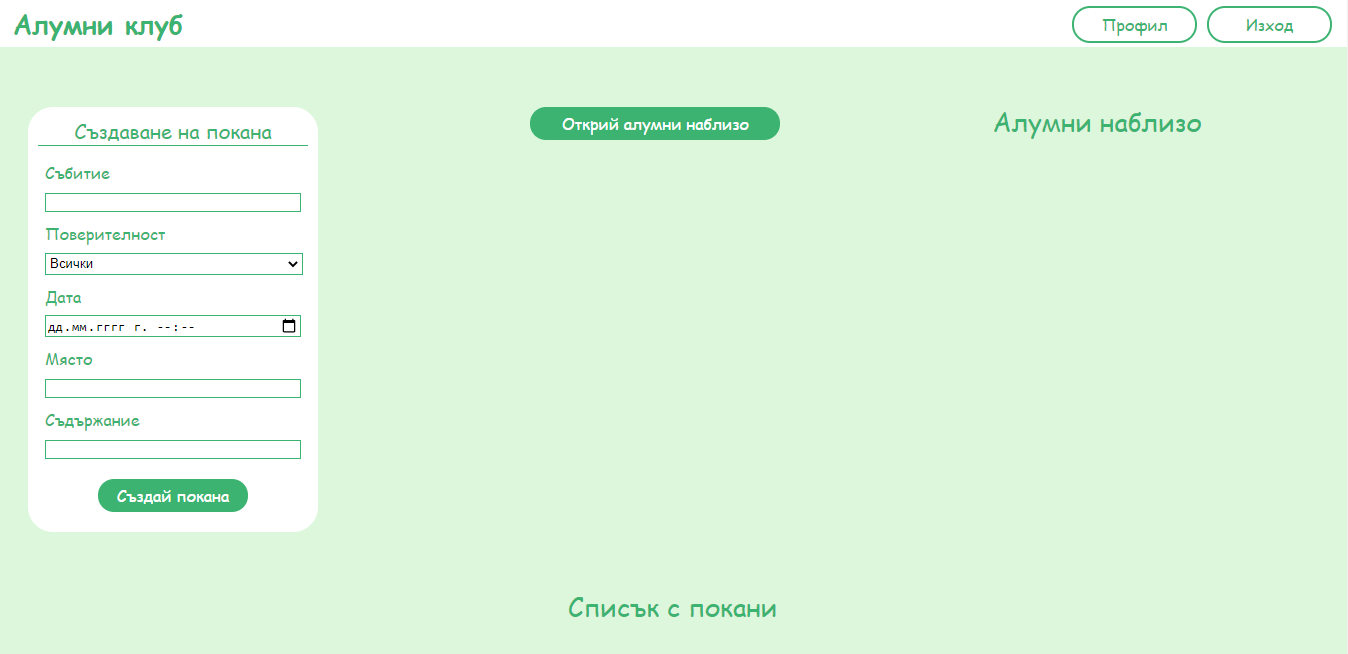
## **6. Кратко ръководство на потребителя**

При при стартиране на приложението се появява форма за вход в системата. Потребителят трябва да въведе своето потребителско име и парола и да натисне бутона „Влез“.

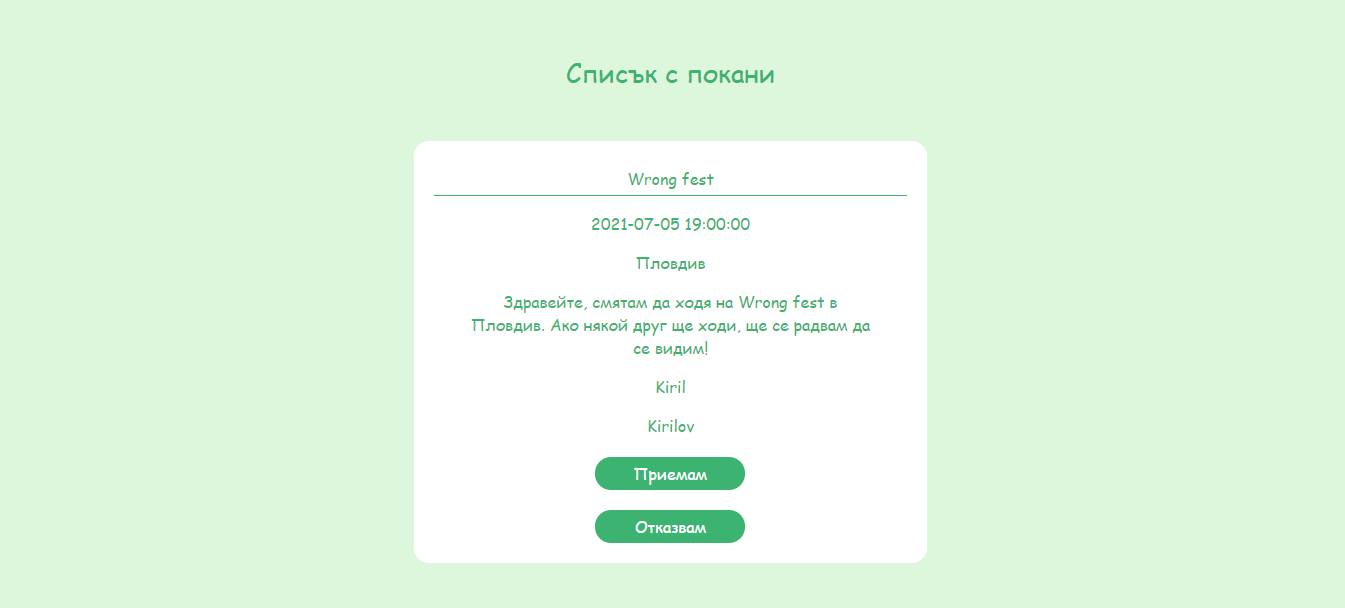


Фигура 1: Страница “Вход”

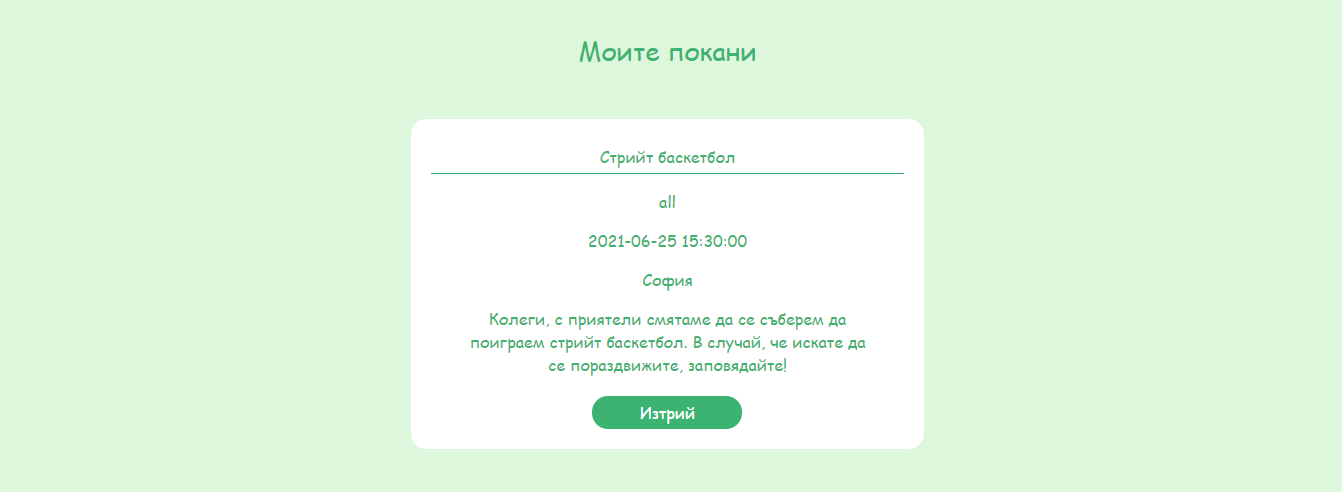
Основната страница на приложението съдържа функционалността за създаване на покана, откриване на близки алумнита и визуализиране на списък с покани (за потребителя - в секцията “Списък с покани”, и такива, създадени от него - в секцията „Моите покани“). В поканите от “Списък с покани”, потребителят може да отбележи “Приемам” или “Отказвам”, а в поканите от “Моите покани”, той може да изтрие своя собствена покана.



Фигура 2: Начална страница, представяща логото, навигацията и секциите за създаване на покана и алумни наблизо

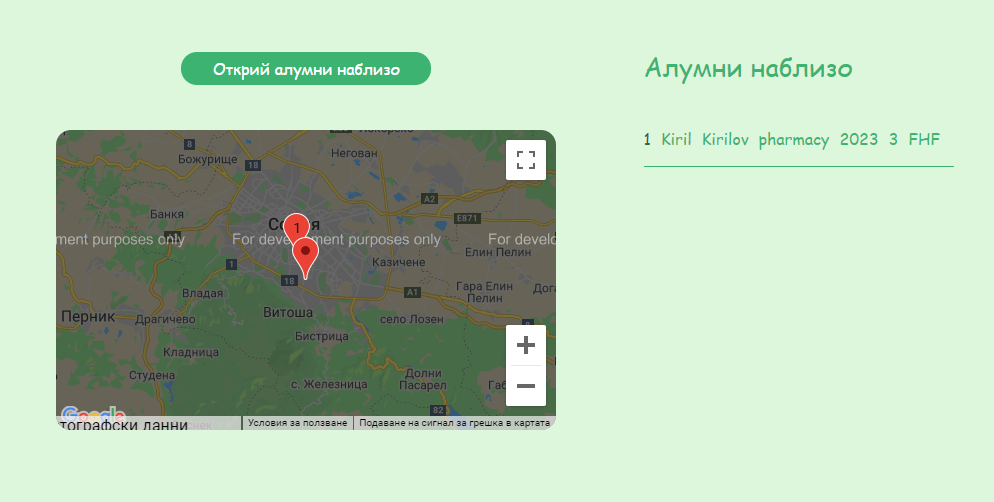


Фигура 3: Начална страница, представяща секцията “Списък с покани”



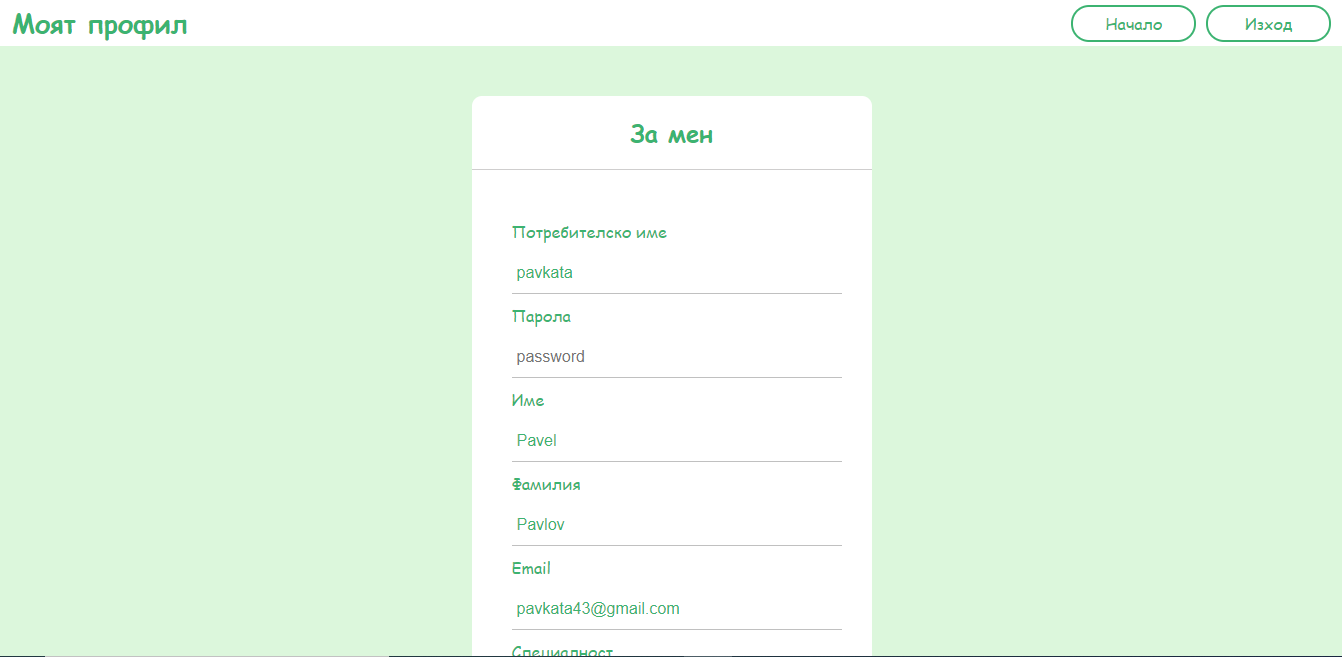
Фигура 4: Начална страница, представяща секцията “Моите покани”

След натискане на бутона „Открий алумни наблизо“, се визуализира карта, показваща местоположението на текущия потребител, както и на тези в близост до него. Всички близки потребители имат номер на маркера на картата и отдясно (в секцията „Алумни наблизо“) е изписана информация за това кои са те.

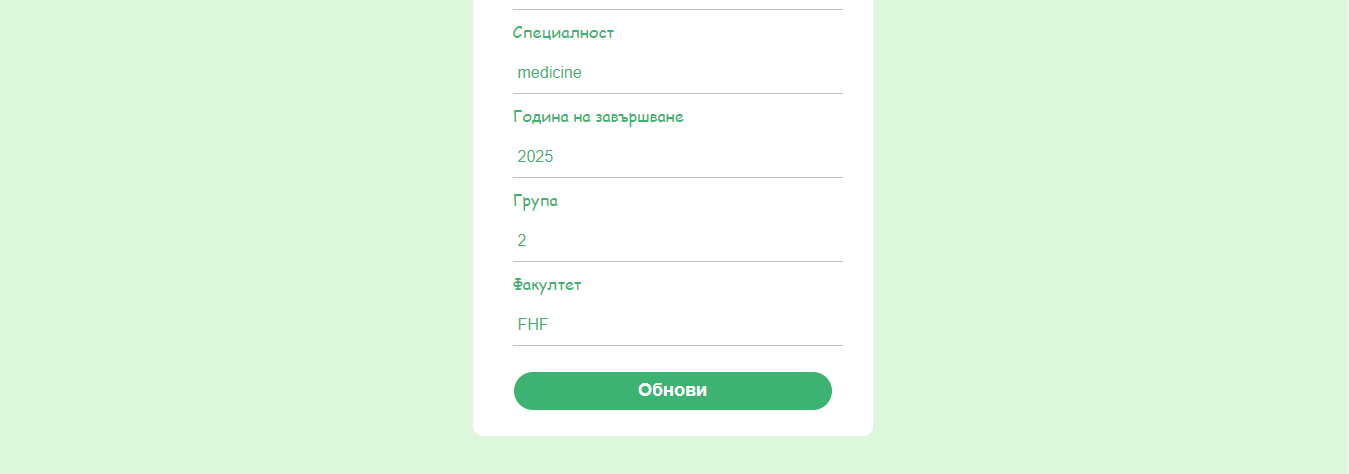


Фигура 5: Начална страница, представяща секцията “Алумни наблизо” и визуализиране на карта

От навигационното меню може да се отиде на страницата за Промяна на профил, в която потребителят може да промени своето име, фамилия, парола и/или имейл. Всички други данни за завършилия студент са фиксирани и не могат да се променят. Промяната става чрез натискане на бутона „Обнови“, след като потребителят е въвел своите нови данни.



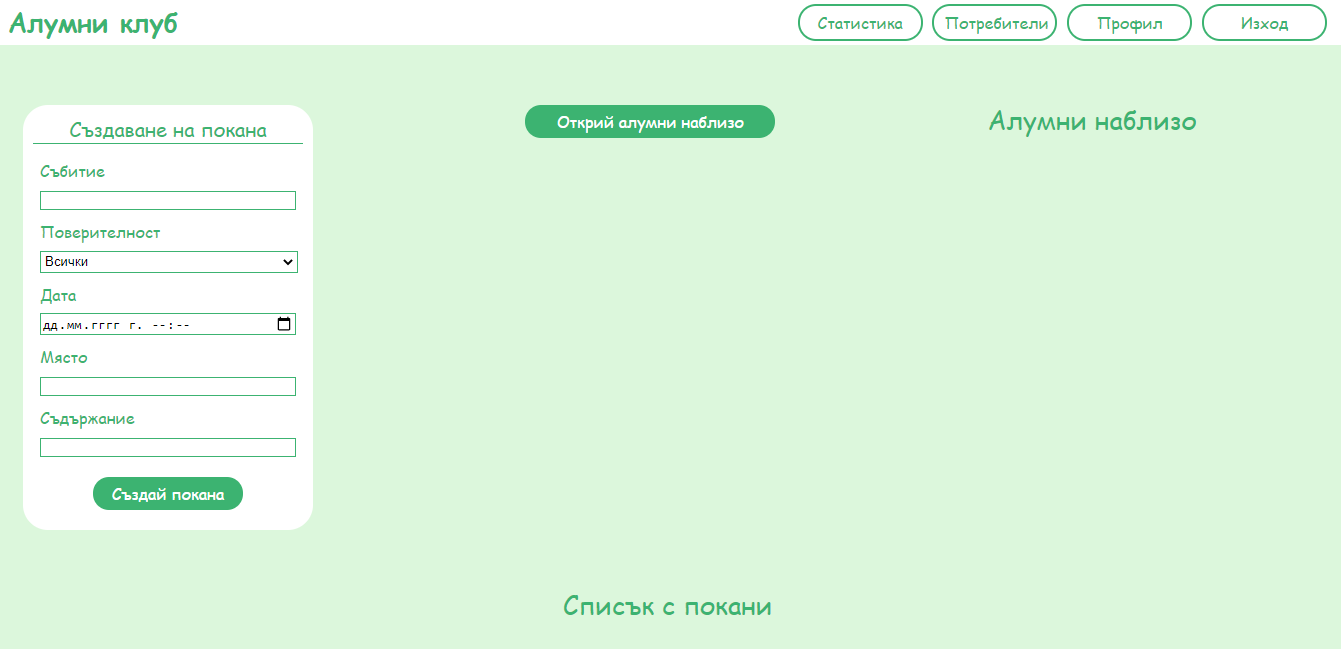
Фигура 6: Страница “Моят профил” 1/2



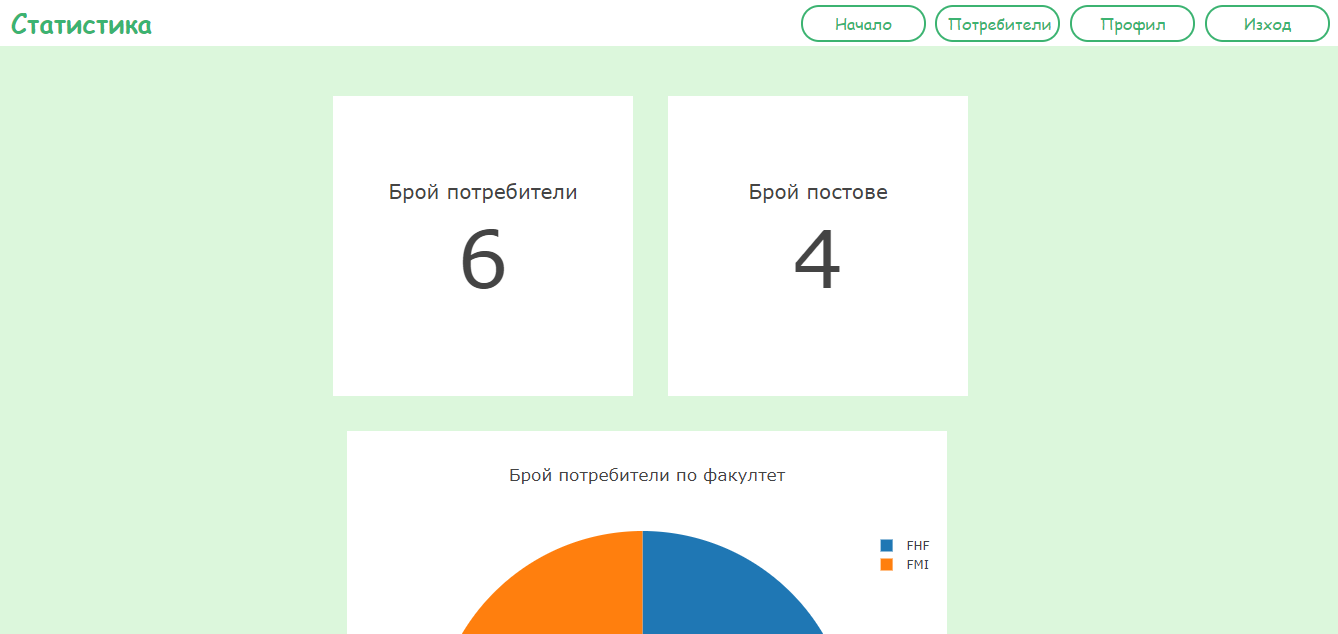
Фигура 7: Страница “Моят профил” 2/2

От навигационното меню потребителят може да отиде отново на началната страница или да излезе от системата. При натискане на бутона „Изход“ се зарежда страницата за вход в системата.

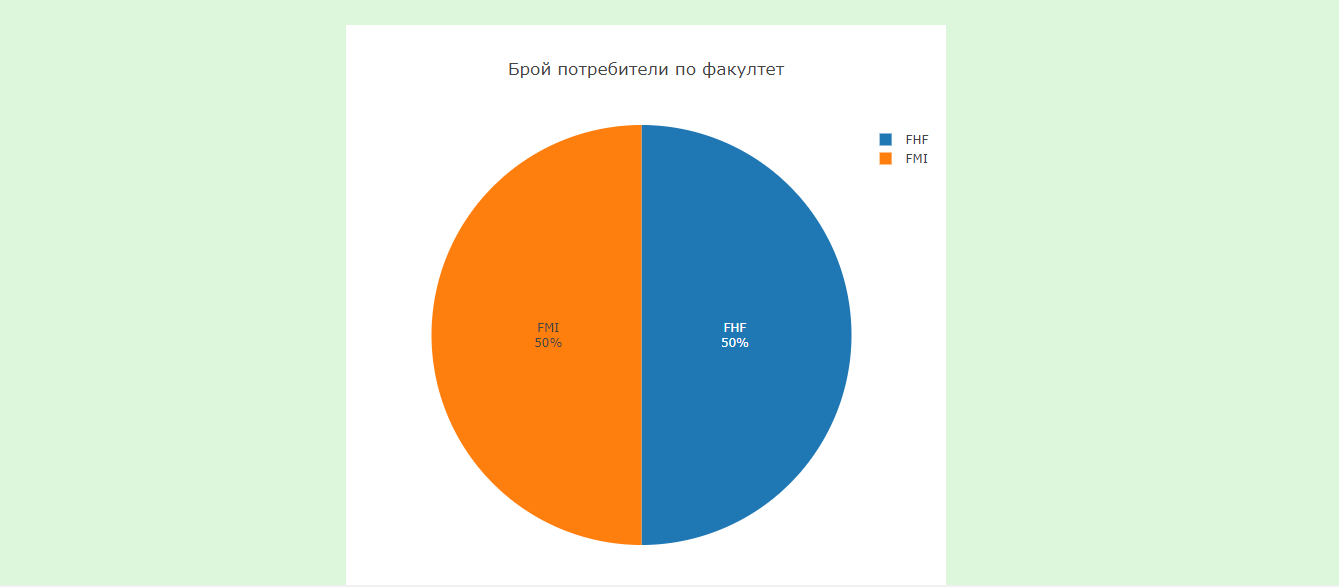
Админът е друг тип потребител на нашата система. Той има достъп освен до горепосочените функционалности, също и до страници, съдържащи справки за потребителите на системата (бутонът „Потребители“ от навигационното меню) и статистики за потребителите и направените постове в системата (бутонът “Статистика”).



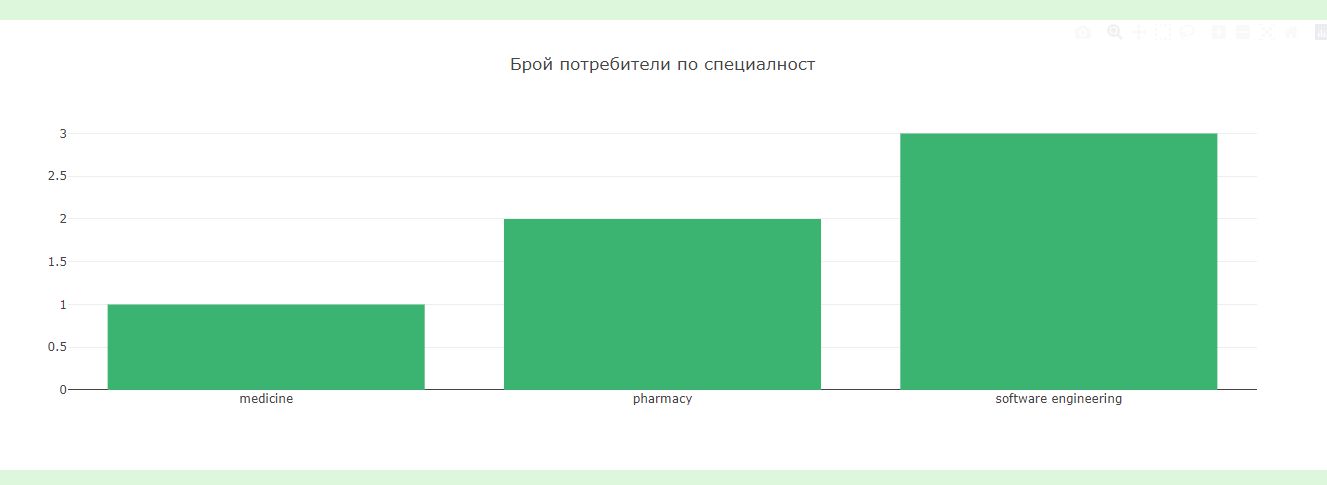
Фигура 8: Начална страница на администратора



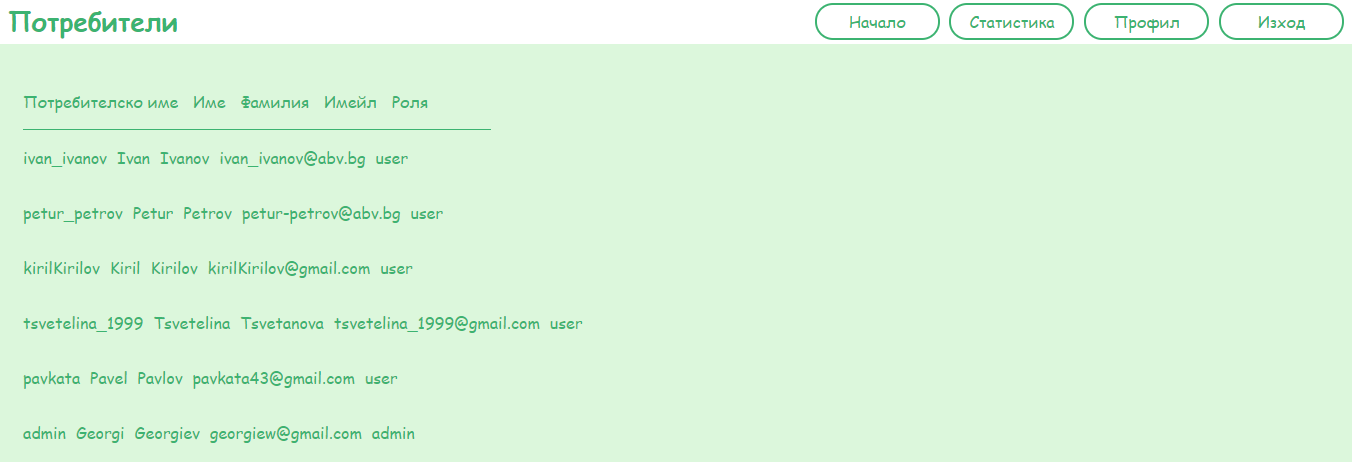
Фигура 9: Страница “Статистика”, представяща брой потребители и брой постове



Фигура 10: Страница „Статистика“, представяща брой потребители по факултет



Фигура 11: Страница „Статистика“, представяща брой потребители по специалност



Фигура 12: Страница “Потребители”, представяща списък с всички потребители на системата

## **7. Примерни данни**

Примерни данни за влизане в системата като администратор:

{

“username”: “admin”

“password”: “Admin\_359”

}

Примерни данни за влизане в системата като потребител:

{

“username”: “ivan\_ivanov”

“password”: “Ivan\_ivanov123”

}

Примерни данни за промяна на профила на потребител:

{

“password”: “Ivan\_ivanov123-new-password”

“firstName”: “Ivan”

“lastName”: “Ivanov”

“email”: “[ivan\_ivanov@abv.bg](mailto:ivan_ivanov@abv.bg)”

}

Примерни данни за създаване на пост:

{

“occasion”: “Имен ден”

“privacy”: “all”

“occasionDate”: “2021-09-09 18:00:00”

“location”: “Плевен”

“content”: “Здравейте, колеги. Заповядайте на именния ми ден. Ще го празнувам вкъщи в Плевен. Ако някой желае, може да донесе настолни игри.”

}

## **8. Описание на програмния код**

Структурата на проекта е следната:

alumniClub

config

config.ini

frontend

index.html

home

home.html

home.js

home.css

login

login.html

login.js

login.css

profile

profile.html

profile.js

profile.css

statistics

statistics.html

statistics.js

statistics.css

users

users.html

user.js

user.css

backend

db

repositories

userRepository.php

postRepository.php

dbConnection.php

endpoints

allUsers.php

answerPost.php

createPostEndpoint.php

deletePost.php

....

entities

nearbyUserInfo.php

post.php

privacy.php

user.php

userPost.php

services

postService.php

statisticsService.php

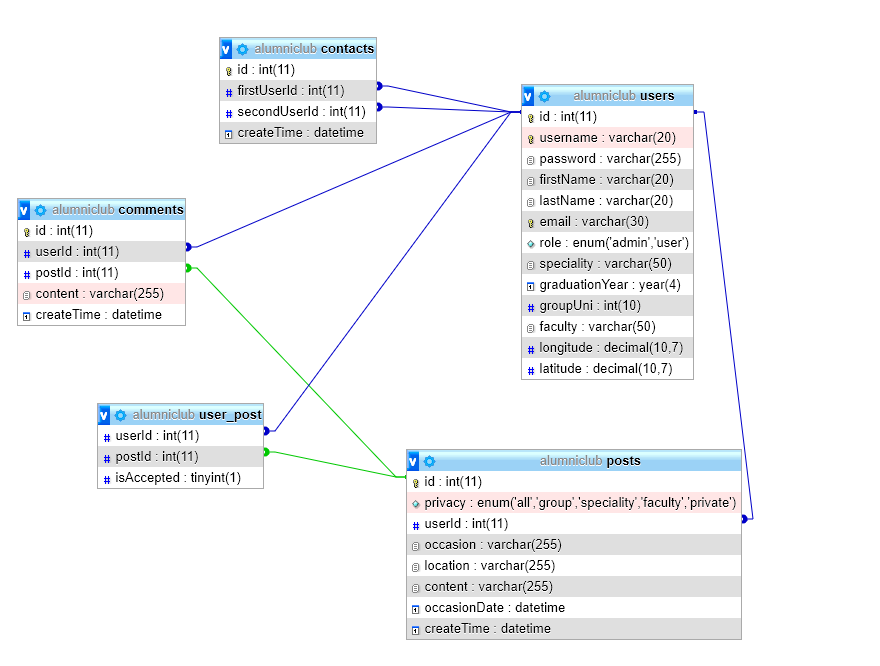
userService.php

database

create-tables.sql

insert-data.sql

В директорията database се намират SQL файловете, които трябва да се заредят в phpMyAdmin. create-tables.sql съдържа заявките за създаването на таблиците, a insert-data.sql съдържа заявките, които добавят примерни данни.



Фигура 13: Диаграма на таблиците от базата данни

В директорията config се намира config.ini файла, който пази информацията, която ни трябва за свързването с базата данни, като и радиуса, по който намираме потребители наблизо. Идеята тази информация да е в конфигурационен файл е по-лесно да могат да се правят промени.

В папката repositories са всички SQL заявки, свързани с базата от данни. UserRepository и PostRepository съдържат съответно statements за потребителите и постовете (поканите).

В папката services се намира основната част от PHP кода. В класовете PostService, StatisticsService и UserService е бизнес логиката на нашето приложение, която взима и обработва данните от заявките в repositories.

Следващата съществена част от backend-кода са endpoint-ите. Имаме ендпойнт за всяка функционалност на системата.

Front-end частта на системата е разпределена по папки в зависимост от това какъв изглед ни е предложен - дали начална страница, профил, статистики. В JavaScript файловете е валидацията на входните потребителски данни и също връзката с backend-a.

За да бъде по-добро потребителското изживяване, сметнахме, че би било добре да не бъдат визуализирани постовете с минала дата.

Хеширането на паролата в едно приложение също е критично за сигурността на потребителите. Освен валидации в back-end частта, има и валидации във front-end с помощта на JavaScript.

Приложението е responsive и е използван само CSS за стилизацията на различните компоненти.

Покрити са всички необходими функционалности приложението да служи за комуникация между завършили студенти. Има два основни типа потребители - обикновен потребител (алумни) и администратор.

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

Работата по проекта беше изцяло обща. Всички функционалности измислихме и разработихме заедно. Всяка една от нас участваше в писането на код и в изготвянето на документацията.

Една от бъдещите насоки за развитие на системата е да добавим възможност на потребител да коментира събитие. Друга насока е създаването на контакти, т.е. един потребител да може да добавя друг потребител като “приятел”. Това ще означава, че ще може да добавим още една област на видимост на постовете, която да е само за приятелите на потребителя.

## **10. Какво научих**

Едно от нещата, които научихме е, трислойната архитектура MVC, приложена на практика. Удобството, което дава е, че няма смесване на логиката. Това ни помогна по-лесно да си оправим проблемите, които срещнахме. Също, и за бъдещо развитие на системата, ще ни е по-лесно разработването.

Успяхме да научим основите на PHP, JavaScript и CSS, използвайки информацията от лекциите, упражненията и статии от интернет.

Най-важното качество, което усвоихме, беше работата в екип. Това е едно от основите неща, необходими за изграждането на добре работещо приложение.

## **11. Използвани източници**

[1] JavaScript Open Source Graphing Library, <https://plotly.com/javascript/>

[2] Примерни кодове на различните уеб технологии, <https://www.w3schools.com/>

[3] Достъпни решения на възникналите проблеми по време на разработване, <https://stackoverflow.com/>

[4] Хеширане на потребителските пароли директно в базата от данни, <https://phppasswordhash.com/>

[5] Google maps API,

<https://developers.google.com/maps/documentation/javascript/overview>

Предал (подпис): ………………………….

/*фн, имена, спец., група*/

Предал (подпис): ………………………….

/*фн, имена, спец., група*/

Предал (подпис): ………………………….

/*фн, имена, спец., група*/ Приел (подпис):………………………….

/доц. *Милен Петров*/