**Имена:**  Петър Петров, Стефан Чолаков **Начална година:** *2020* **Програма:** бакалавър, (КН) **Курс:** 4 **фн:** *81612, 81722*  
**Тема:** Статистики за натоварване на сгради **Дата: 2021-01-26 Предмет: w15prj\_KN \_final**

**имейл**: [petar.petrov220998@gmail.com](mailto:petar.petrov220998@gmail.com)**,** stcholakov98@gmail.com

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

# ТЕМА: Статистики за натоварване на сгради

## 1. Условие

Целта на проекта е да се създаде система, която да показва статистики за натоварването на сгради. Тези статистики да бъдат визуализирани с различни графики – по етажи за конкретна дата и сграда, за сграда по зададена дата и за зала по зададени сграда, име на зала и дата. Системата трябва да позволява импорт и експорт на данни за сгради, зали и резервации. Също така трябва да предоставя начин за добавяне на сграда и зала, като при добавянето на сграда трябва да е възможно специфицирането на GPS координати и капацитет. Трябва да е възможно извикването на системата отвън, с помощта на REST API call, като по подадени име на сграда и дата, системата трябва да показва графики за съответната сграда за тази дата.

## 2. Въведение

При разработването на разпис на занятията за дадено учебна заведение, трябва да се има предвид доколко дадена сграда, етаж или зала са заети, преди да може да се запази зала за даден времеви интервал. Затова правим система, която да може да улесни този процес.

## 3. Теория

Приложението е разделено на две части – frontend и backend. Backend частта е разделена на 4 основни части – database, domain, controllers и services. Database модулът е отговорен за това да прави връзка към базата, като информацията, нужна за тази връзка е изнесена в отделен файл и е лесно конфигурируема. Domain модулът съдържа класове, които изграждат домейна на приложението. Модулът “services” съдържа php страници, които се достъпват от frontend частта. Те обработват информацията от идващите заявки и я предават на контролерите, които от своя страна създават инстанции на класовете в “domain” модула и ги записват в базата. Frontend логиката е разделена в 2 модула – “components” и “pages”. В “components” модула се намират класове, които ни помагат по лесен начин да създаваме графики. Също така има класове, отговорни за създаватено на navigation bar-a. В “pages” се намират класове, които визуализират основните изгледи на приложението – форми за добавяне на сгради, зали, импорт и експорт на данните и визуализации на графиките.

## 4. Използвани технологии

**Презентационен слой** - разработен на HTML 5, CSS и JavaScript. Използвахме React js  
**База данни** - създадена на MySQL, като за DB клиент е използван phpMyAdmin.  
**Бизнес логика** - разработена на PHP

## 5. Инсталация и настройки

Първо, zip файлът, съдържащ проекта, трябва да бъде разархивиран в папка htdocs, която се намира в инсталационната папка на XAMPP. Също така, XAMPP Apache сървисът трябва да бъде конфигуриран да използва порт 8888 за HTTP комуникация. Файлът config.php, който се намира в /backend/database/ трябва да бъде променен с името и паролата на потребителя, от чието име ще се достъпва базата. Пускане на проекта:

1. BackEnd – за пускането на тази част от проекта е нужно да се пуснат Apache и MySQL сървисите от XAMPP Control Panel-a.
2. За FrontEnd частта има два основни начина:
   1. С помощтта на Docker, като се изпълнява следната команда:  
      **docker run -it --rm -p 8888:8888 -p 3000:3000 stefancholakov/web-frontend**
   2. Изпълняват се последователно командите **npm install** и **npm start** в директорията **/frontend**, като предварително трябва да имате инсталиран Node.js runtime (може да бъде свален и инсталиран оттук: <https://nodejs.org/en/download/> )

След пускането на FrontEnd и BackEnd, можете да достъпите уеб приложението на адрес http://localhost:3000

## 6. Кратко ръководство на потребителя

## 

Ще опишем функциите една по една:

1. При натискане на home,ни връща в началната страница.
2. Бутона add building,ни вкарва в нова страница,в която има форма за създаване на нова сграда по подадени характерситки.
3. Бутона add hall,ни вкарва в нова страница,в която има форма за създаване на нова зала по подадени характерситки.
4. Statistics per building – ни препраща в страница където при избиране на сграда и дата ни визуализира графики за натовареност за тази сграда.
5. Statistics per floor - ни препраща в страница където при избиране на сграда и етаж ни визуализира графики за натовареност за този етаж в сградата.
6. Statistics per hall - ни препраща в страница където при избиране на сграда и зала ни визуализира графики за натовареност за тази зала в сградата.
7. Import – При избирането на даден JSON file, импортва тази конфигурация в JSON file-a в базата
8. Export – Export-ва базата в JSON файл.

## 7. Примерни данни

Примерни данни могат да бъдат намерени във файла sample\_data.json. Те могат да бъдат импортнати в приложението с помощта на “Импорт” функционалноста.

## 8. Описание на програмния код

Ще започнем първо с бекенд файловете:

***/controllers***

*BookingController.php* – в него изпълняваме основните операции за booking(добавяне и проверяване дали съществува)

*BuildingController.php* – в него изпълняваме основните операции за building(добавяне,проверяване дали съществува и листване на всички сгради)

*DBController.php* – в него имаме като фунцкионалност изтриване на всичката информация от базата

*ExportController.php* – като функционалност имаме тук,че експортваме базата в json файл

*HallController.php* - в него изпълняваме основните операции за hall(добавяне ,проверяване дали съществува и листване на всички зали)

*ImportController.php* – по подаден json импортваме данни в базата

***/database***

*Config.php* – конфигурацията на базата

*DBConnector.php* – създава нова конекция към базата

**/domain**

*Booking.php* – клас, репрезентиращ резервация на зала

*Building.php* - клас, репрезентиращ сграда

*Hall.php* - клас, репрезентиращ зала

**/services**

*buildings.php* – страница, към която в зависимост от това дали правим GET или POST рекуест можем съответно да вземем всички сгради или да добавим сграда

*dropDB.php* - страница, към която правейки DELETE рекуест можем да вземем изтрием съдържанието на базата

*export.php* - страница, към която правейки GET рекуест, получаваме съдържанието на цялата база в JSON формат

*halls.php* - страница, към която в зависимост от това дали правим GET или POST рекуест можем съответно да вземем всички зали или да добавим зала

*import.php* - страница, към която правейки POST рекуест можем да добавим информация за сградир зали или резервации на зали в базата. Тази информация се предава в body частта на рекуеста.

Фронтенд файлове:

*Barchart.js* – Създава ни графика от вид BarChart

*DoughnutChart.js* –Създава ни графика от вида DoughnutChart

*LineChart.js* –Създава ни графикат от вида LineChart

*Navbar.css* - Задава изгледа на navigation bar-a ни

*Navbar.js* – Създава и менажира navigation bar-a

*SidebarData.js* – съдържа данните на navigation bar-a

*App.js* –Съдържа основните route-ове на фронтенд-а

*App.module.css* – css файла за App.js

*Index.js* – Началната сраница на нашето уеб приложение

*AddBuilding.js –* форма за добавяне на сграда

*AddBuilding.module.css –* CSS файл за формата за добавяне на сграда

*AddHall.js -* форма за добавяне на зала

*AddHall.module.css -* CSS файл за формата за добавяне на зала

*BuildingStatistics.js – изобразява* графики за натовареност на дадена сграда по зададени име на сградата и дата

*BuildingStatistics.module.css –* CSS файл за BuildingStatistics.js

*Export.js –* форма, позволяваща експорт на данните от базата в JSON формат

*Export.module.css -* CSS файл за *Export*.js

*FloorStatistics.js - изобразява* графика за натовареност по етажи на дадена сграда по зададени име на сградата и дата

*FloorStatistics.module.css -* CSS файл за *FloorStatistics*.js

*HallStatistics.js - изобразява* графика за натовареност по етажи по зададени име на сграда и дата

*HallStatistics.module.css -* CSS файл за *HallStatistics*.js

*Home.js – начална страница на приложението*

*Home.module.css* - CSS файл за *Home*.js

*Import.js – форма, позволяваща импорт на данни за сгради, зали и резервации на зали в базата, като те трябва да са в JSON формат*

*Import.module.css -* CSS файл за *Import*.js

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

Проектирахме заедно схемата на базата и си разделихме писането на backend частта. Относно графиките, първоначално направихме графиките за натовареността по сгради заедно, а после всеки едит от нас създаде по още една графика самостоятелно. По останалата част от frontend-a, си разделихме работата така, че един от нас направи UI частта, а другият направи свързването ѝ с backend-a. Относно възможностите за бъдещо развитие е възможно да се добави още допълнителна информация за всяка сграда в базата, както и да се добавят още REST API ендпойнти, така че системата да може да се достъпва отвън и да показва графики спрямо параметрите, подадени в заявката.

## 10. Какво научих

Научихме как се разработва уеб приложение от нулата,как се свързва бекенд-а със фронт-енд и как всички компоненти да работят като едно. Научихме доста неща за JavaScript, PHP, HTML и CSS.

## 11. Използвани източници

[1] Страници от официалната документация на езика PHP:   
<https://www.php.net/docs.php>  
Автори: The PHP group  
Последно посетен: 15.01.2021

[2] Документация на Chart.js  
<https://www.chartjs.org/docs/latest/>  
Последно посетен: 15.01.2021

[3] Страници от официалната документация на MySQL:  
<https://dev.mysql.com/doc/>  
Автори: Oracle Inc.  
Последно посетен: 15.01.2021

[4] Създаване на REST API с PHP   
<https://codeofaninja.com/2017/02/create-simple-rest-api-in-php.html>  
Автори: Mike Dalisay  
Публикувано: 17.02.2017

[5] Въведение в React js  
<https://reactjs.org/tutorial/tutorial.html>  
Автори: Mike Dalisay  
Последно посетен: 15.01.2021

[6] Превръщане на PHP обект в JSON   
<https://appdividend.com/2019/04/02/php-json_encode-example-encode-json-data-in-php-tutorial/>  
Автори: Krunal Lathiya  
Последно обновен: 10.11.2020

Предал (подпис): ………………………….

/*81612, Петър Петров, КН, 2/*

Предал (подпис): ………………………….

/*81722,Стефан Чолаков, КН,2*/

Приел (подпис): ………………………….  
  
/доц. *Милен Петров*/