

Ръководители: доц. д-р Милен Петров			
Година: 2022		Програма: бакалавър, (СИ)	Курс: 3
Предмет: WEB Технологии		Издание: 18-то	Код: w18prj_SI_final
Име	Фамилия	Факултетен номер	e-mail
Андрея	Дяксова	62455	aidjaksova@uni-sofia.bg
Мария	Велева	62445	mimi.veleva1608@gmai.com
Светослава	Димов	62477	svetoslav.dimov21@gmail.com



**СУ „СВ. КЛИМЕНТ ОХРИДСКИ“ ГРАД
СОФИЯ**

Система за паркиране във ФМИ



Съдържание

Условие	4
Въведение	4
Теория	4
База данни	4
Структурата на базата от данни се състои от следните таблици:	4
Връзки в базата от данни:	6
Ключове	6
Ключове:	6
Чужди ключове:	6
Използвани технологии	6
Инсталация и настройки	6
Ръководство на потребителя	6
„Регистрация“	6
„Вход в системата“	7
Начална страница „Моят акаунт“	8
Моят график	9
QR код	10
Паркинг зони	12
Запази паркомясто	13
Изход	15
Примерни данни	15
Описание на програмния код	16
Описание на функционалностите	21
Регистрация	21
Функционално изискване	21
Резултат	21
Начин на реализация	21
Вход	21
Функционално изискване	21
Резултат	21
Начин на реализация	21
Преглед на резервириани паркоместа	22
Функционално изискване	22
Резултат	22
Начин на реализация	22
Качване на график във CSV формат	22
Функционално изискване	22

Резултат	22
Начин на реализация	22
График на предстоящите занятия	23
Функционално изискване	23
Резултат	23
Начин на реализация	23
Генериране на QR код	24
Функционално изискване	24
Резултат	24
Начин на реализация	24
Справка за свободни паркоместа в определен интервал от време	24
Функционално изискване	24
Резултат	24
Начин на реализация	24
Резервиране на паркомясто в определен интервал от време	25
Функционално изискване	25
Резултат	25
Начин на реализация	25
Принос на студента	26
Ограничения и възможности за бъдещо развитие	26
Какво научихме	27
Използвани източници	27
Фигури и таблици	28
Фигури	28
Таблици	28

1. Условие

Да се създаде система за паркиране във ФМИ, която включва:

- Вход в системата и управление на видовете регистрирани потребители: „Щатен преподавател“ и „Хоноруван преподавател“.
- Начална страница за всеки потребител, включваща информация за неговите данни и календар с график за деня.
- Прочитане на CSV файл, съдържащ графика на занятията на щатен или хоноруван преподавател, и запаметяване на информацията от него в базата данни на системата.
- Генериране на постоянен QR код за щатните преподаватели и временен QR код за хоноруваните. Кодът се асоциира с номер на колата, имена на потребителя и вид потребител.
- Глобална карта на парко зоните, където може да се паркира.
- Възможност за търсене на свободни паркоместа по зададена дата и времеви интервал.
- Възможност за избиране на свободно паркомясто от картата на съответната зона.
- Възможност за резервиране на паркомясто по зададена дата и времеви интервал.
- Възможност за преглед на всички запазени паркоместа, направени от съответния потребител.
- Възможност за преглед на всички предстоящи занятия на потребителя.

2. Въведение

Система за паркиране във ФМИ има за цел да въведе ред в използването на паркоместата на територията на ФМИ и близките до него факултети – ФЗФ и ФХФ. Като собственост на СУ, за тях трябва да бъде предоставена „справедлива“ логика за използване от оторизираните за това потребители. До този момент такава система няма и всички потребители, които използват паркоместата, успяват да паркират там единствено ако има свободно място, когато пристигнат в университета. Система за паркиране във ФМИ ще промени тази практика, като обосobi допълнителна логика, която ще внесе ред и последователност.

3. Теория

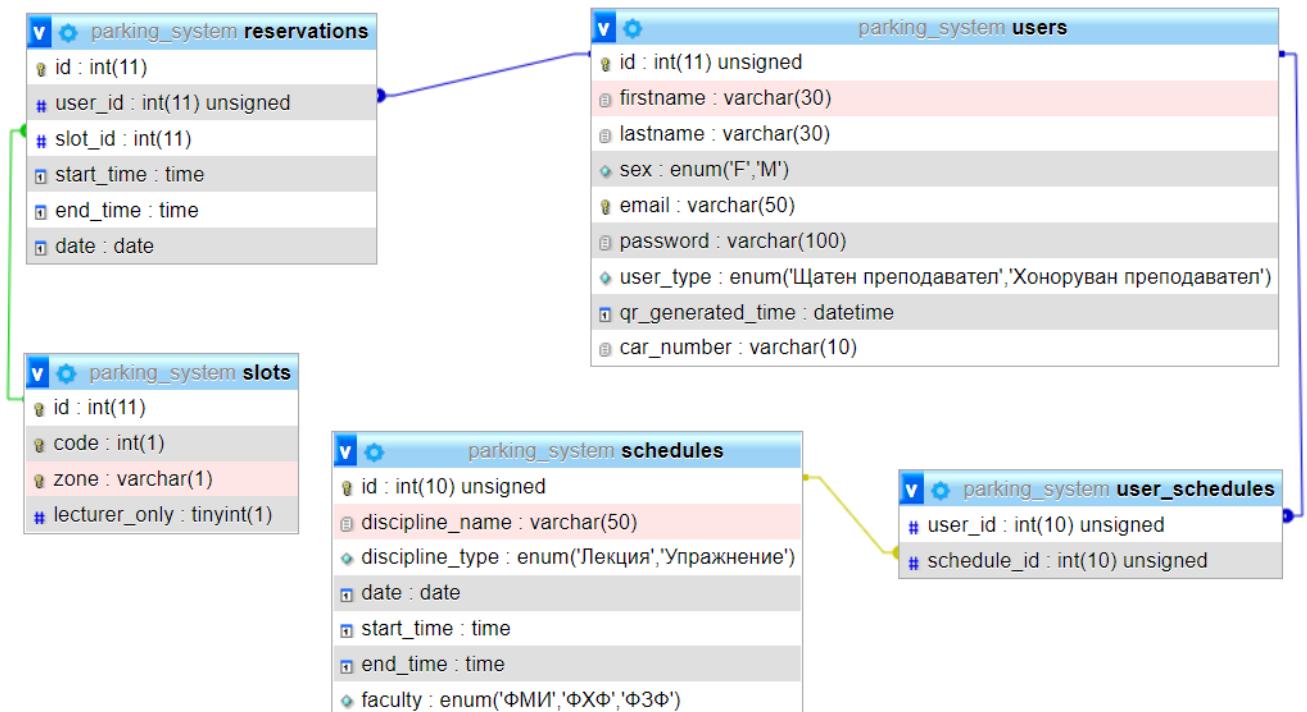
3.1. База данни

3.1.1. Структурата на базата от данни се състои от следните таблици:

- **users**: съдържа информация за потребителите на системата (име, фамилия, пол, имейл, хеширана парола, вид потребител, дата на генериране на QR код, номер на кола). Когато потребител се регистрира в системата, се добавя запис в таблицата. Когато

потребител влиза в системата, се автентицира посредством съдържанието в тази таблица.

- **schedules**: съдържа информация за занятията от графиците на всички преподаватели (име на дисциплина, вид на дисциплина, начало и край на часа и сграда, в която се провежда). Когато преподавател качва в системата CSV файл, съдържанието му се добавя под формата на записи в таблицата. Служи за валидация на времеви интервали при правенето на резервации на паркоместа.
- **user_SCHEDULES**: съдържа идентификационния номер на преподавателя за всеки запис от schedules. Служи като свързваща таблица между users и schedules, тъй като връзката между тях трябва да бъде много:много.
- **slots**: съдържа информацията за всяко паркомясто и се попълва от конфигурационния файл "create_db.sql" (код на мястото, което е число, зона - А, В или С, и булева променлива за това дали мястото е предназначено само за щатни преподаватели). Използва се за валидация при резервация на паркомясто.
- **reservations**: съдържа информация за всички резервации на паркоместа (id на паркомястото и id на потребител, който го е запазил, дата, начален и краен час на резервацията). При създаване на нова резервация се добавя нов запис в таблицата. Използва се за валидация при резервация на паркомясто.



Фиг. 1. Диаграма на Базата данни

3.2. Връзки в базата от данни:

- users – user_schedules: връзка от тип „1 : много“
- user_schedules – schedules: връзка от тип „много : 1“
- users – reservations: връзка от тип „1 : много“
- reservations – slots: връзка от тип „много : 1“

3.3. Ключове

3.3.1. Ключове:

- users: id;
- schedules: id;
- reservations: id;
- slots: id.

3.3.2. Чужди ключове:

- user_schedules: user_id, schedule_id;
- reservations: user_id, slot_id.

4. Използвани технологии

Системата за паркиране е със стандартна трислойна архитектура, състояща се от презентационен слой (Presentation Tier), логически слой (Logic Tier) и слой „Данни“ (Data Tier).

За реализацията на презентационния слой сме използвали маркиращ език HTML, стилизиращ език CSS и програмен език JavaScript. При разработката са следвани добри практики за семантичен HTML5 синтаксис.

Логическият слой е базиран на програмния език PHP и чрез него е реализирана сървърната логика.

За слоя „Данни“ сме използвали релационна база данни – MySQL.

При разработването на системата се използва мултиплатформения софтуер XAMPP за поддръжка на базата данни и изпълнението на PHP скриптовете.

5. Инсталация и настройки

За създаването на базата данни е необходимо да се импортира SQL скрипта "backend/db/create_db/create_db.sql".

За добавяне на примерни данни в базата данни е необходимо да се импортира SQL скрипта "backend/db/create_db/test_data_db.sql".

6. Ръководство на потребителя

6.1. „Регистрация“

Потребител се регистрира в системата, като попълни форма със следните полета:

- Име
- Фамилия
- Пол (Мъж или Жена)

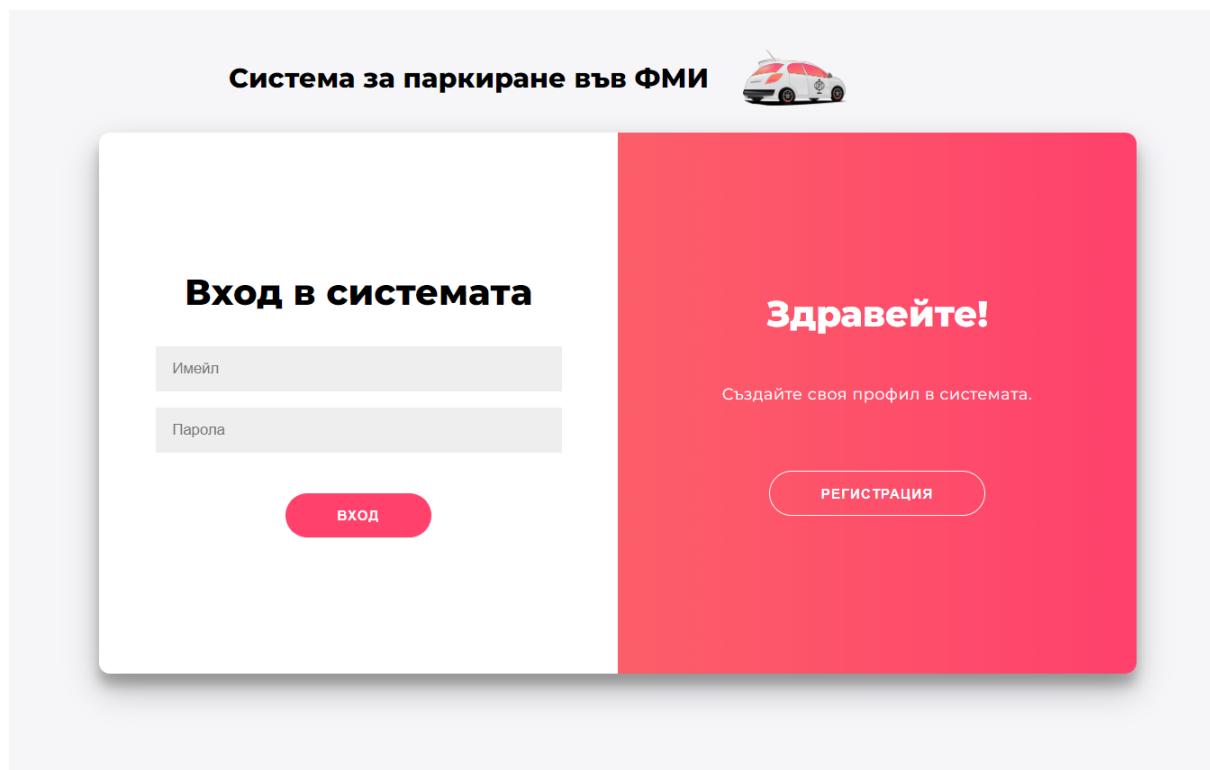
- Номер на кола
- Имейл
- Парола
- Повторно въвеждане на паролата
- Тип на потребител (Щатен преподавател или Хоноруван преподавател).

The screenshot shows the registration interface for the parking system. At the top, there's a header with the text "Система за паркиране във ФМИ" and a small icon of a white car with red and blue details. Below the header, the main title "Създай акаунт" is displayed. The form consists of several input fields and dropdown menus. The first row contains "Име" (Name) and "Фамилия" (Surname). The second row contains a dropdown menu set to "Мъж" (Male) and a field for "Номер на кола" (Car number). The third row contains a field for "Имейл" (Email). The fourth row contains a field for "Парола" (Password). The fifth row contains a field for "Повторете паролата" (Repeat password). The sixth row contains a dropdown menu set to "Щатен преподавател" (Tutor). A large orange button labeled "РЕГИСТРИРАНЕ" (Register) is located at the bottom right of the form area.

Фиг. 2. Регистрация

6.2. „Вход в системата“

Потребител влиза във вече съществуващия си профил в системата чрез въвеждане на имейл и парола.



Фиг.3. Вход в системата

6.3. Начална страница „Моят акаунт“

В горната част на началната страница е разположено навигационно меню с връзка към следните функционалности: „Моят акаунт“, „Моят график“, „QR код“, „Паркинг зони“, „Запази паркомясто“, „Изход“.

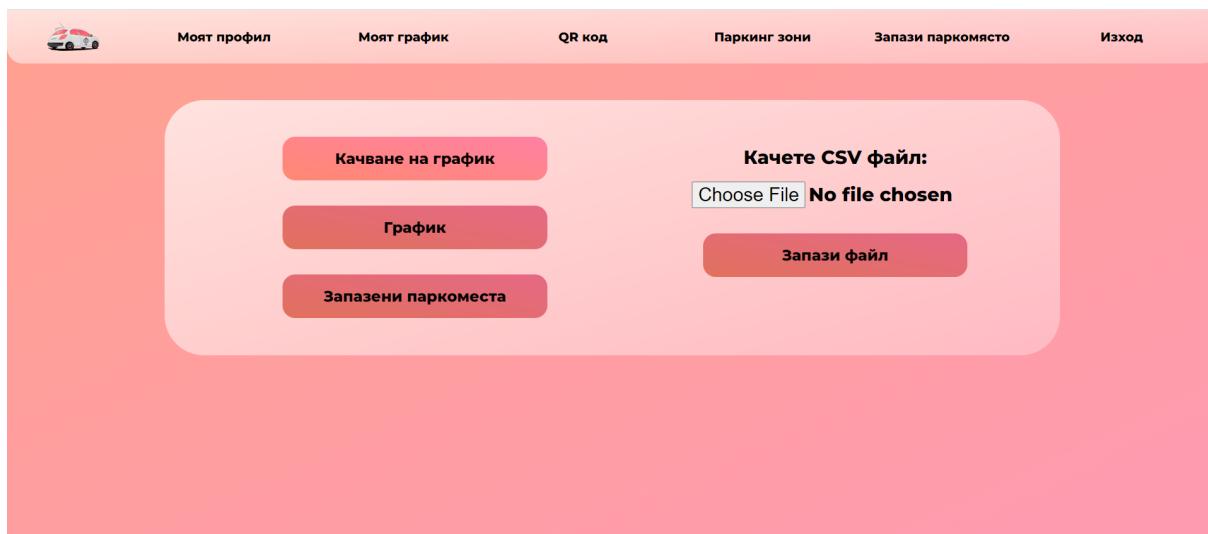
В средата на страницата е обособена секция с информация относно акаунта на потребителя. Секцията е разделена на две части, като от лявата страна се намира информацията на потребителя, а в дясната - графикът за деня.

The image shows the "Моят акаунт" (My Account) screen. At the top, there is a navigation bar with icons for a car, "Моят профил" (My Profile), "Моят график" (My Schedule), "QR код" (QR Code), "Паркинг зони" (Parking Zones), "Запази паркомясто" (Reserve Parking Space), and "Изход" (Logout). The main content area has a pink background. On the left, there is a rounded rectangle containing a user profile picture, the name "Ирина", and other details: "Фамилия: Павлова", "Имейл: irenap@gmail.com", "Тип потребител: Хоноруван преподавател", and "Номер на кола: CA8682PB". On the right, there is a box titled "График за дненя:" (Daily Schedule) with the word "Неделя" (Sunday) inside.

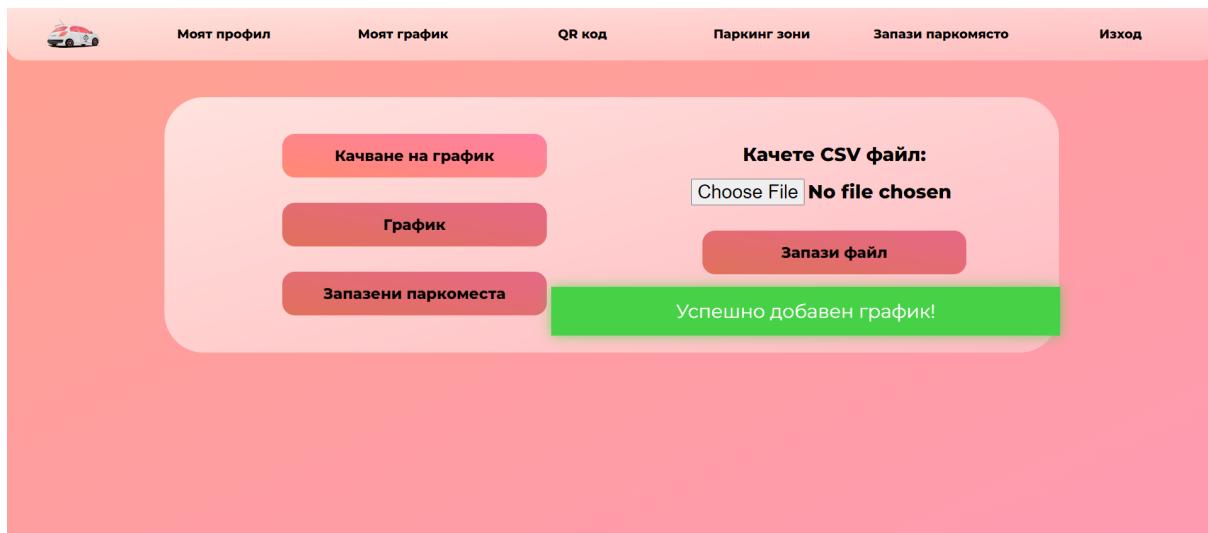
Фиг. 4. Моят акаунт

6.4. Моят график

Потребителите могат да качат своя график чрез бутона „Качване на график“ и прикачване на CSV файл. Предаването става с бутона „Запази файл“.

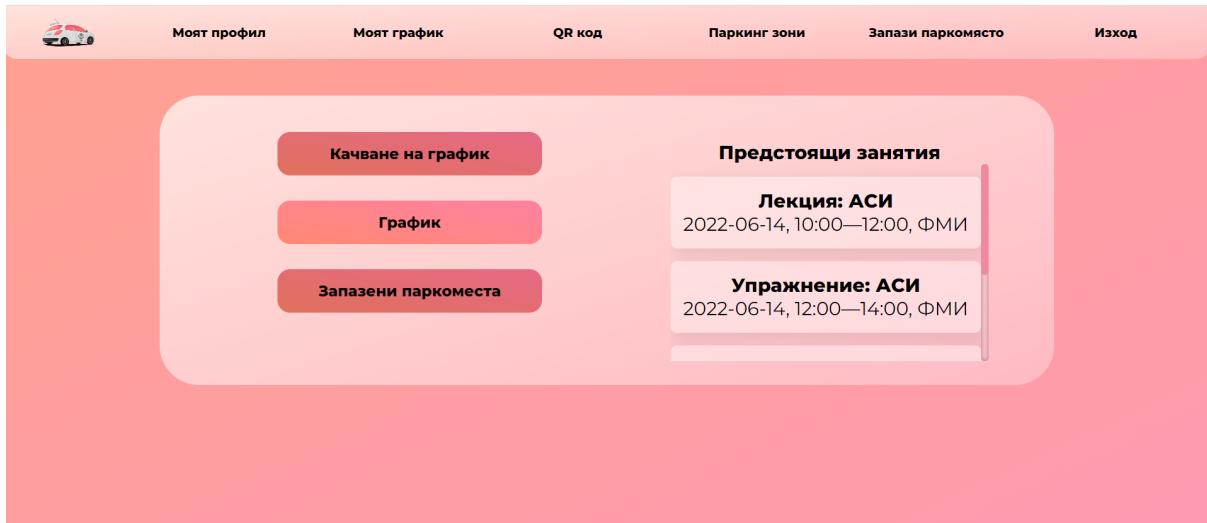


Фиг. 5. Качване на график



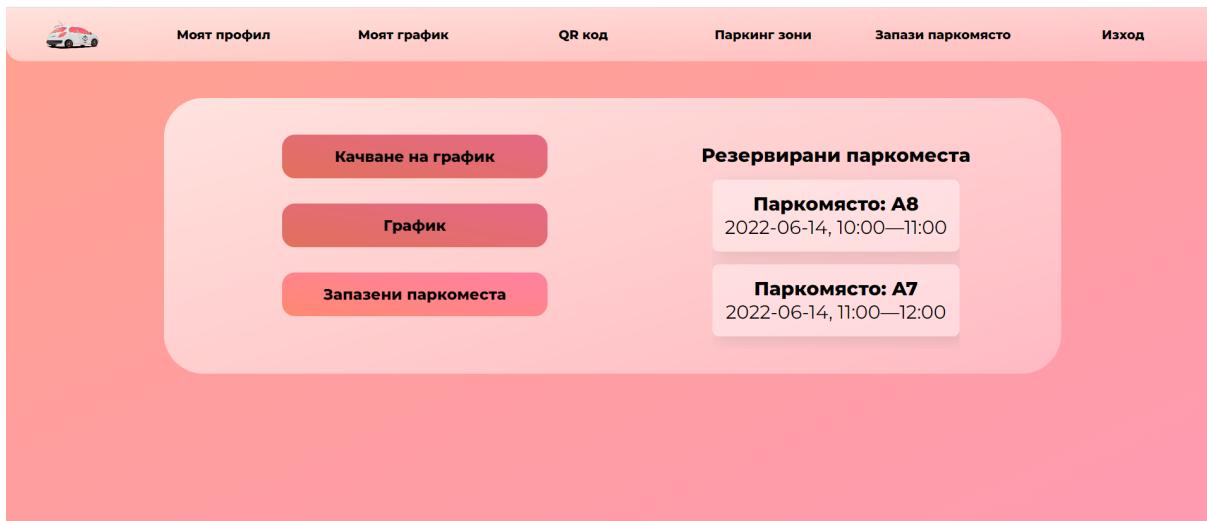
Фиг. 6. Успешно качен график

В секцията „График“, потребителите проверяват предстоящите си занятия сортирани по време.



Фиг. 7. График

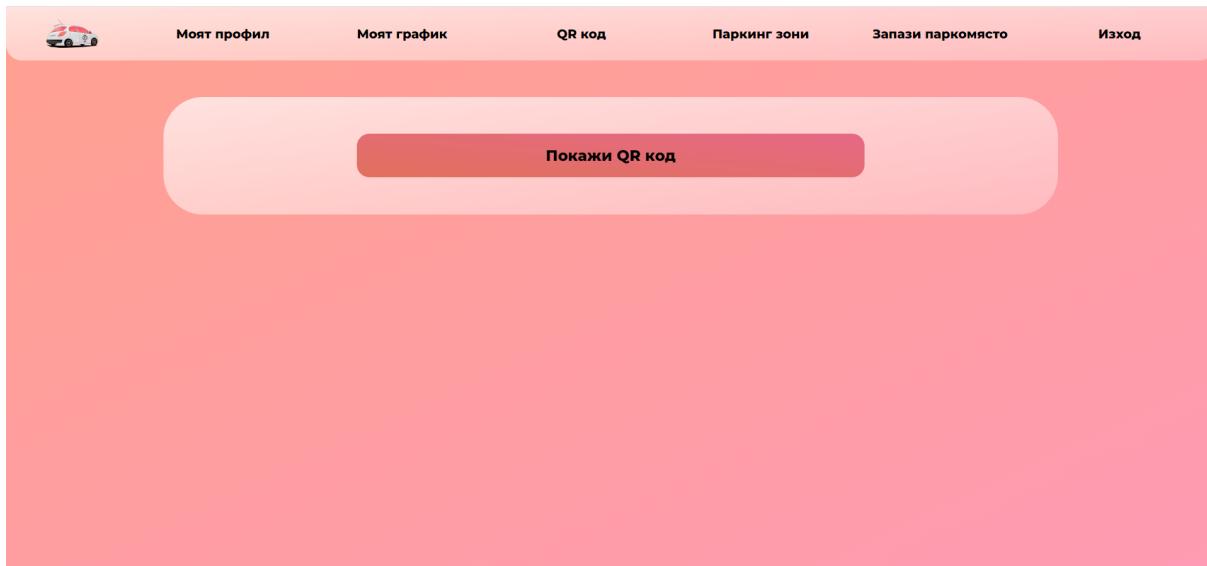
В секция „Запазени паркоместа“, всички потребители на системата могат да проверят предстоящите си резервации на паркоместа сортирани по време.



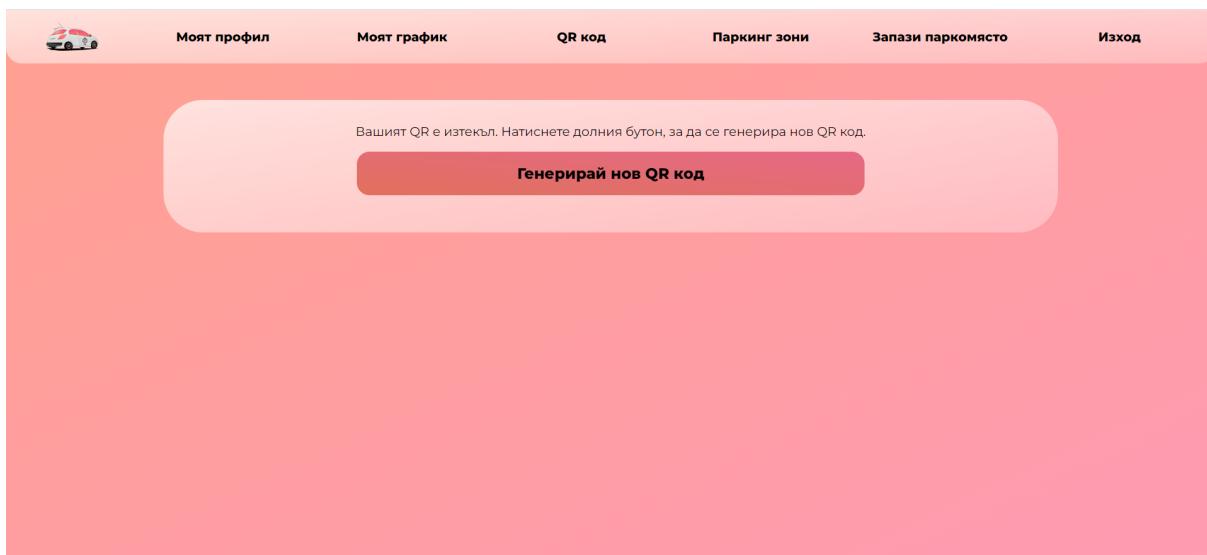
Фиг. 8. Запазени паркоместа

6.5.QR код

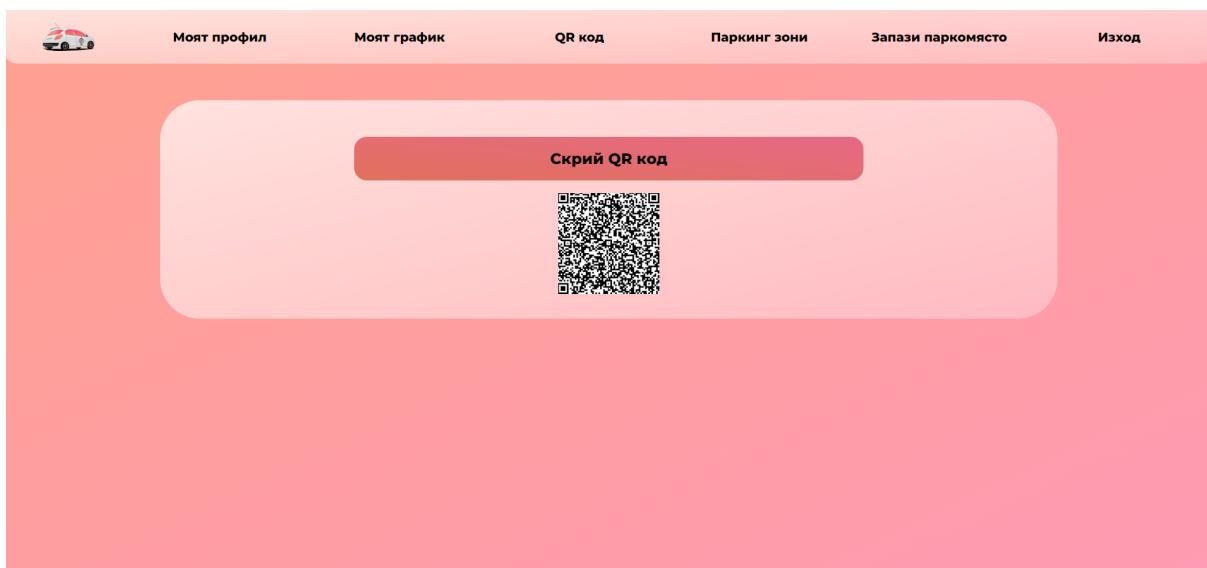
В секцията „QR код“, при натискане на бутона „Покажи QR код“, се визуализира QR код на потребителя, който е постоянен за потребителите от тип „Щатен преподавател“ и временен с валидност 1 час от генерирането за потребителите от тип „Хоноруван преподавател“.



Фиг. 9.1 Бутон за визуализация на QR код



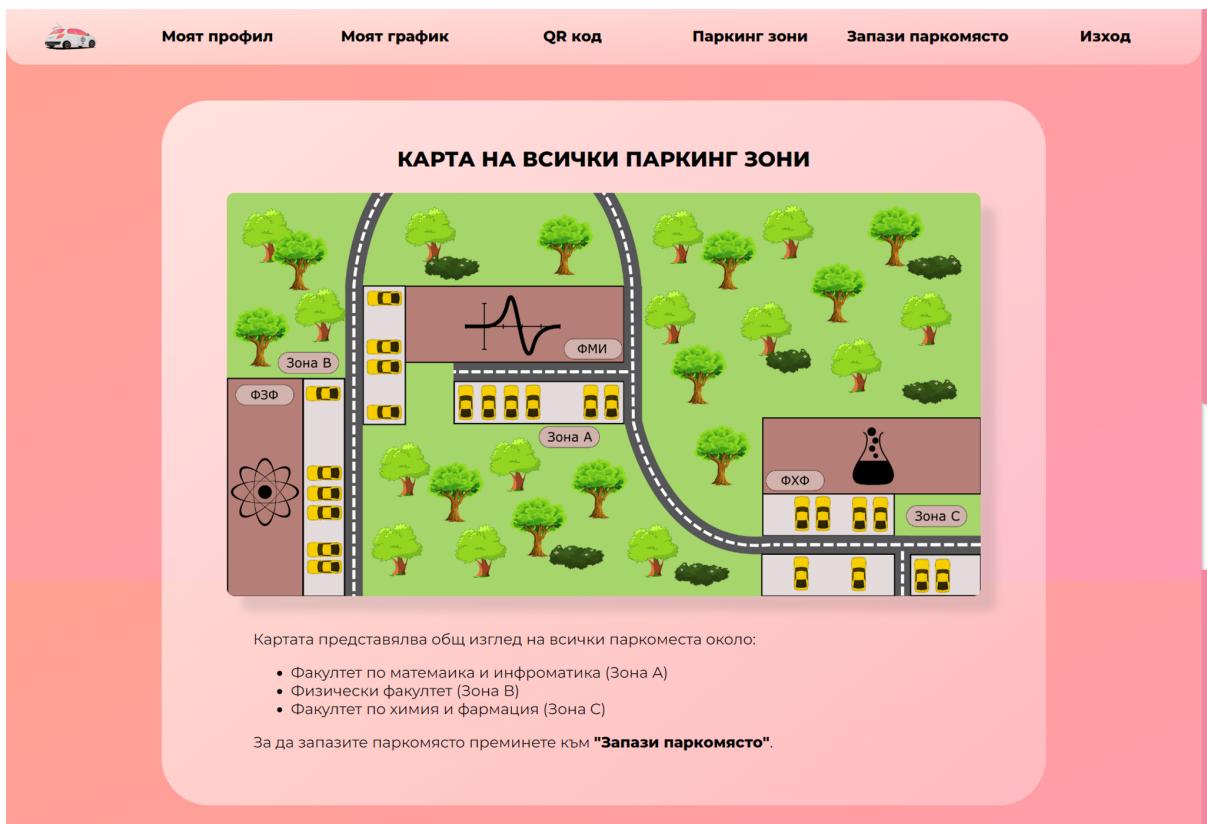
Фиг. 9.2 Генериране на QR код при изтекла валидност



Фиг. 10. Визуализиран QR код

6.6. Паркинг зони

Представена е общ карта с всички паркинг зони, които се администрират от системата.



Картата представя общ изглед на всички паркоместа около:

- Факултет по математика и информатика (Зона A)
- Физически факултет (Зона B)
- Факултет по химия и фармация (Зона C)

За да запазите паркомясто преминете към "Запази паркомясто".

Фиг. 11. Карта на всички зони за паркиране

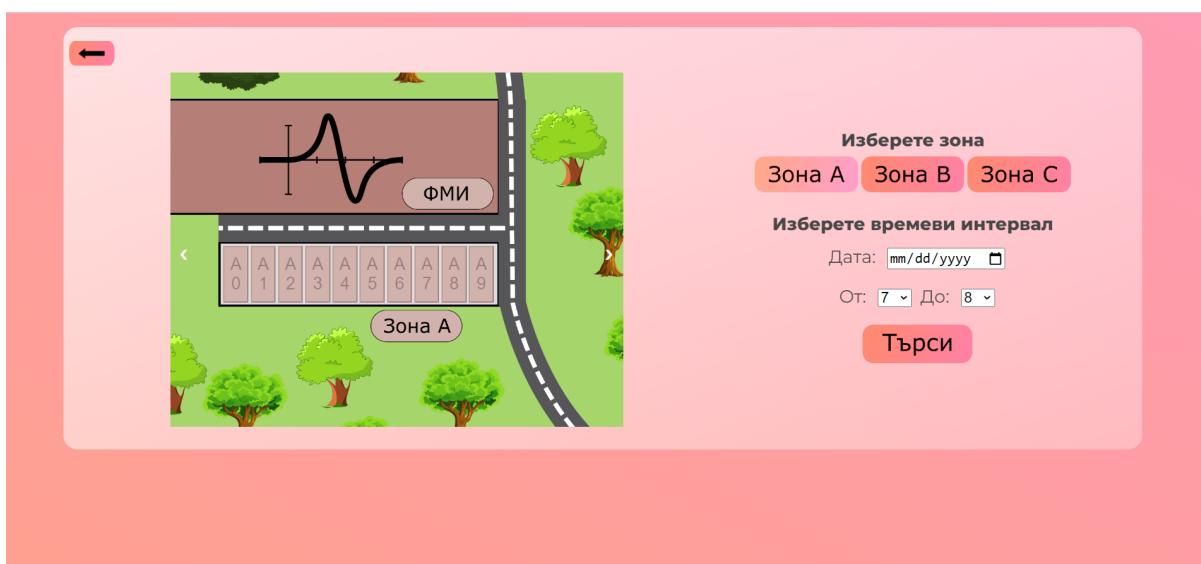
6.7. Запази паркомясто

Включена е галерия, която предоставя визуална информация на потребителя за разпределението на местата по зони. Потребител избира дата и времеви интервал посредством специализирана за това форма. След изпращане на формата до базата данни се представя информация за статуса на всяко паркомясто в избраната от потребителя зона.

Описание на паркоместата:

- **Недопустимо** - Паркомясто е маркирано в розов цвят, ако попада в един от два случая:
 - о Потребителят все още не е направил заявка;
 - о Мястото не е достъпно за дадения вид потребител. За хоноруван преподавател не са достъпни всички места, предназначени само за щатен преподавател (в случая местата от зона А с номера < 5, за зона В места с номер > 9, за зона С места с номер < 7);
- **Свободно** - паркомястото е маркирано в зелен цвят, ако е направена заявка за даден интервал и мястото не е заето от друг потребител по същото време и потребителят има достъп до него (т.е. ако е хоноруван, мястото е различно от споменатите в горната подточка).
- **Резервирано** - паркомястото е маркирано в червен цвят, ако вече е запазено от друг потребител.
- **Избрано за резервиране** - паркомястото е маркирано в светло зелен цвят, когато потребителят го е избрали на картата, за да го резервира.

Потребител може да избере само паркоместата маркирани със зелен цвят. При натискане на дадено място изскача меню за потвърждение на избора на паркомястото. След успешно резервиране на паркомясто се извежда съобщение.



Фиг. 12. Преди избиране на времеви интервал



Фиг. 13. След избиране на времеви интервал



Фиг. 14. Избиране на паркомясто на картата



Фиг. 15. Успешно запазено паркомясто

6.8.Изход

Потребителят излиза от своя профил в системата. Тогава той е препратен към началната страница за вход/рестиграция.

7. Примерни данни

Импортирането на SQL скрипта с тестови данни, ще създаде следните два акаунта на потребители, техните графици и направени резервации за паркоместа:

	Щатен преподавател	Хоноруван преподавател
Име	Боян	Ирена
Фамилия	Бончев	Павлова
Имейл	boyanb@fmi.uni-sofia.bg	irenap@gmail.com
Парола	boyanb	irenap

Таблица 1. Данни на потребителите

Дисциплина	Тип занятие	Дата	Начален час	Краен час	Сграда
АСИ	Упражнение	13/06/2022	10:00:00	12:00:00	ФМИ
УСИ	Упражнение	14/06/2022	9:00:00	11:00:00	ФМИ
АСИ	Лекция	15/06/2022	9:00:00	12:00:00	ФХФ
УСИ	Лекция	15/06/2022	13:00:00	15:00:00	ФМИ

Таблица 2. Данни на хоноруван преподавател

Таблица с направени резервации на паркоместа

Потребител	Номер	Дата	Начален час	Краен час
Ирена Павлова	16	2022-06-13	10:00:00	13:00:00
Ирена Павлова	11	2022-06-15	9:00:00	12:00:00

Иrena Павлова	15	2022-06-15	13:00:00	15:00:00
---------------	----	------------	----------	----------

Таблица 3. Данни за направени резервации на паркоместа

Примерен CSV файл, съдържащ допълнителни занятия за преподавател, който да се въведе в базата данни на системата чрез функционалността „Качване на файл“.

Дисциплина	Тип занятие	Дата	Начален час	Краен час	Сграда
XML	Лекция	13/06/2022	10:00:00	13:00:00	ФМИ
СШП	Лекция	13/06/2022	15:00:00	17:00:00	ФХФ
XML	Упражнение	15/06/2022	9:00:00	11:00:00	ФМИ
СШП	Упражнение	15/06/2022	14:00:00	16:00:00	ФХФ
ПКВ	Лекция	17/06/2022	12:00:00	14:00:00	ФзФ
ПКВ	Упражнение	17/06/2022	15:00:00	16:00:00	ФзФ

Таблица 4. Тестови данни - график на щатен преподавател

8. Описание на програмния код

0. index.html

Стартира заглавната страница на приложението, която препраща потребителя към страницата за вход/регистрация (1.1.1 login_form.html).

1. frontend

1.1. login

1.1.1. **login_form.html** - HTML код за вход в системата.

1.1.2. css

1.1.2.1. **login_form_styles.css** - Стилизация на страницата за вход в системата.

1.1.3. javascript

1.1.3.1. **send_user_data.js** - Изпраща въведените данни от формата за вход в системата към endpoint 2.1.6.1. на сървъра или към endpoint 2.1.8.1. на сървъра.

1.1.4. images

1.1.4.1. error_response.png

1.1.4.2. Logo.png

1.2. account

1.2.1. **account_view.html** - HTML код за главната страница на потребителя.

1.2.2. css

1.2.2.1. **account_view_styles.css** - Стилизация на главната страница на потребителя.

1.2.3. javascript

1.2.3.1. **generate-qr.js** - Дефинира реакцията на бутоните “Покажи QR код”, “Скрий QR код” и “Генерирай QR код”. При изтекъл QR код прави POST заявка към

endpoint 2.1.10.1, за да запише новото време на генерирания код. При валиден QR код прави заявка към endpoint 2.1.4.2, за да вземе от базата нужните данни, които да се пазят в QR кода.

- 1.2.3.2. **get_schedule.js** - Дефинира реакциите на бутоните от секция „Моят график“, проверявайки типа на потребителя. При натискане на бутон „График“ извлича графика за съответния преподавател с предстоящите му занятия (от endpoint 2.1.5.1) и ги визуализира в сортиран по време ред. При натискане на бутон „Запазени паркоместа“ извлича предстоящите резервации на паркоместа за съответния потребител (от endpoint 2.1.5.2.) и ги визуализира в сортиран по време ред. При натискане на бутон „Качване на график“ се открива подсекцията за качване на CSV файл.
- 1.2.3.3. **load_today_schedule.js** - Зарежда графика със занятия за съответния ден на преподавателя (чрез endpoint 2.1.4.1.), сортирани по два критерия: първо по дисциплина, след което по време.
- 1.2.3.4. **load_user_data.js** - Зарежда данните от регистрацията на потребителя (чрез endpoint 2.1.4.2.) и ги визуализира в заглавната му страница „Моят акаунт“.
- 1.2.3.5. **navbar_buttons.js** - Дефинира реакцията на бутоните от навигационния бар. При натискане на „Моят профил“, „Моят график“, „QR код“ или „Паркинг Зони“ се показват съответните секции от html-a. При натискане на „Запази паркомясто“, се препраща към страницата zones.html, а при натискане на бутон „Изход“ се прави заявка към endpoint 2.1.7.1, при което потребителят излиза от профила си.
- 1.2.3.6. **submit_csv_file.js** - Изпраща се въведения от преподавателя CSV файл към endpoint 2.1.10.1 на сървъра.

1.2.4. **images**

- 1.2.4.1. Logo.png
- 1.2.4.2. map.png
- 1.2.4.3. male photo.png
- 1.2.4.4. female photo.png

1.3. **map**

- 1.3.1. **zones.html** - HTML код за секция „Запази паркомясто“.

- 1.3.2. **css**

1.3.2.1. **zones.css** - Стилизация на страницата за секция „Запази паркомясто“.

1.3.3. **javascript**

1.3.3.1. **buttons_click.js** - Създават се слушател за събитие 'click' върху всеки от бутоните на паркоместата. При настъпване на такова събитие се проверява дали

съответният бутон е на свободно паркомясто и се изпраща заявка към endpoint 2.1.9.1. на сървъра. В зависимост от отговора на сървъра – ще се изведе съобщение за успешна резервация или ще

визуализира самозатварящ се прозорец, индикиращ за грешка.

1.3.3.2. **handle_select_option.js** - Задава логика за рестрикция на опциите за „Край“ в зависимост от избора на „Начало“ на времевия интервал от формата за търсене на свободно паркомясто.

1.3.3.3. **load_map.js** - Създава динамично бутона за всички паркоместа. Създава динамично опциите за „Начало“ и „Край“ от формата за търсене. Дефинира поведението на слайдшоуто със стрелките в ляво и дясно на снимките, както и на бутоните под него.

1.3.3.4. **send_form_search.js** - Изпраща въведените от потребителя данни във формата за търсене на свободно паркомясто до endpoint 2.1.1.1. на сървъра. В зависимост от отговора на сървъра или всеки бутон ще се оцвети в цвета, отговарящ на статуса му, или самозатварящ се прозорец ще индицира за грешка.

1.3.4. **images**

1.3.4.1. back.png

1.3.4.2. buttonA.png

1.3.4.3. buttonB.png

1.3.4.4. buttonC.png

1.3.4.5. fmi1.png

1.3.4.6. fzf1.png

1.3.4.7. fhf1.png

1.3.4.8. search.png

2. **backend**

2.1. **api**

2.1.1. **check_slots**

2.1.1.1. **check_slots.php** - Проверява данните за въведения времеви интервал от потребителя и го валидира. Ако е некоректен връща грешка, а ако е коректен прилага връща два масива от асоциативни масиви, които

съдържат информация за резервираните и недостъпните за потребителя бутони.

2.1.2. check_user_type

- 2.1.2.1. **check_user_type.php** - Проверява и връща типа на потребителя.

2.1.3. functions

- 2.1.3.1. **check_JSON_validity.php** - Валидира подаден обект с JSON формат.
- 2.1.3.2. **login.php** - При успешно автентифициране на потребител създава негова сесия и бисквитки.

2.1.4. load_account_page

- 2.1.4.1. **get_today_schedule.php** - Връща графика със занятия на съответния преподавател за деня, извлечени от базата данни.
- 2.1.4.2. **get_user_data.php** - Връща име, фамилия, имейл и типа на потребителя, извлечени от базата данни.

2.1.5. load_user_schedule_data

- 2.1.5.1. **load_user_schedule.php** - Връща графика с предстоящите занятия на съответния преподавател, извлечени от базата данни.
- 2.1.5.2. **load_user_taken_slots.php** - Връща графика с предстоящите резервации на паркоместа за съответния потребител, извлечени от базата данни.

2.1.6. login

- 2.1.6.1. **login_user.php** - Получава и проверява формата на данните от 1.1.3.1. чрез endpoint 2.1.3.1. и ги препраща на endpoint 2.1.3.2. на сървъра. Върнатият отговор се предава към 1.1.3.1.

2.1.7. logout_user

- 2.1.7.1. **delete_user_session.php** - Изтрива сесията и бисквитките на съответния потребител.

2.1.8. registration

- 2.1.8.1. **register_user.php** - Проверява дали въведените данни от 1.1.3.1. за регистрация са валидни и ако да – ги запаметява в базата данни и създава сесия и бисквитки на съответния потребител.

2.1.9. reserve_slots

- 2.1.9.1. **reserve_slot.php** - Изпраща заявка до базата данни, за да провери дали потребителят вече има съществуваща резервация за (част от) избрания времеви интервал. Проверява дали избраното от потребител място е "lecturer_only" и на база на това определя дали паркомястото е достъпно за съответния потребител. Ако потребителят е

преподавател и мястото е "lecturer_only" се проверява дали преподавателят има занятие в рамките на часовия интервал. Ако сме достигнали до тук след всички проверки и не сме върнали съобщение за грешка, то единственото нещо което остава да проверим е дали заявленото паркомясто е свободно или заето. Това става чрез заявка до базата данни. Ако е свободно (няма сечение с други резервации) се изпраща нова заявка до базата данни, с която паркомястото се резервира.

2.1.10. update_qr

- 2.1.10.1. **update_qr_datetime.php** - На входа прочита имейл и време, в което е бил генериран новият QR код, и изпраща заявка към базата данни за промяна на стойността на колоната qr_generated_time от таблицата users, където имейлът съответства на този, който е подаден към endpoint-a.

2.1.11. upload_csv

- 2.1.11.1. **send_csv_to_db.php** - Проверява дали в асоциативния масив от файлове, които се получават чрез POST заявка, има качен файл в CSV формат. След проверката, съдържанието на файла се разделя по нов ред ('\n') и се записва като масив от редове. Проверява за наличието на заглавен ред (header) и в случай, че има го пропуска. Всеки ред се разделя по запетая (','), преформатират се данните както базата данни очаква и всеки ред се записва в масив от асоциативни масиви. Всеки ред от новия масив се подава на заявка, която го записва в базата данни.

2.2. db

2.2.1. create_db

- 2.2.1.1. **create_db.sql** - Създава празни таблици от базата данни и релациите между тях. Този файл служи за конфигуриране на базата данни на системата. След неговото импортиране, системата ще бъде напълно функционална.
- 2.2.1.2. **test_data_db.sql** - Добавя примерни записи в таблиците на базата данни, които служат за тестването на системата.

2.2.2. db_connection

- 2.2.2.1. **connect_to_db.php** - Създава клас, който има метод за свързване с базата данни.
- 2.2.2.2. **db_info.ini** - Конфигурационен файл за осъществяване на връзка с базата данни.

9. Описание на функционалностите

9.1. Регистрация

9.1.1. Функционално изискване

Потребител иска да създаде свой профил в системата.

9.1.2. Резултат

Потребителят е успешно регистриран и се е отворила страницата „Моят профил“.

9.1.3. Начин на реализация

1. Ако потребителят няма регистрация в системата, натиска бутона „РЕГИСТРАЦИЯ“ и се показва формата за регистрация.
2. Потребителят попълва всички необходими полета във формата за регистрация (Име, Фамилия, Пол, Номер на кола, Имейл, Парола, Повторно въвеждане на парола и тип) и натиска бутона „РЕГИСТРИРАНЕ“.
3. JavaScript функцията register() от файла send_user_data.js, взема всички въведени данни от формата за регистрация.
4. При изпращане на попълнената форма се извиква JavaScript функцията sendFormData() от файла send_user_data.js, която прави POST заявка до endpoint "register_user.php" на сървъра.
5. В register_user.php се прави проверка дали двете въведени пароли съвпадат.
6. Прави се заявка до базата данни дали няма вече съществуващ потребител с въведеният имейл.
7. При валидни данни те се запаметяват в базата данни.
8. JavaScript функцията register() от файла send_user_data.js препраща потребителя към account_view.html.

9.2. Вход

9.2.1. Функционално изискване

Регистриран потребител иска да влезе в профила си в системата.

9.2.2. Резултат

Потребителят е влязъл успешно в системата и се е отворила страницата „Моят профил“.

9.2.3. Начин на реализация

1. Потребителят попълва всички необходими полета във формата за вход в системата (Имейл и Парола) и натиска бутона „ВХОД“.
2. JavaScript функцията login() от файла send_user_data.js, взема всички въведени данни от формата за вход.
3. При изпращане на попълнената форма се извиква JavaScript функцията checkLoginData () от файла send_user_data.js, която прави POST заявка до endpoint "login_user.php" на сървъра.

4. Във файла "login_user.php" се извиква "check_JSON_validity.php", в който се проверява дали входният JSON е валиден.
5. Във файла "login_user.php" се извиква "login.php", в който се прави заявка до базата до базата данни, проверява се дали съществува регистриран потребител с въведения имейл, ако да дали паролата съвпада.
6. JavaScript функцията login() от файла send_user_data.js препраща потребителя към account_view.html.

9.3. Преглед на резервирани паркоместа

9.3.1. Функционално изискване

Регистриран потребител прави справка за предстоящите си резервации на паркоместа.

9.3.2. Резултат

Системата визуализира списък на предстоящите резервации на паркоместа на потребителя.

9.3.3. Начин на реализация

1. Потребител избира подсекция „Запазени паркоместа“ от „Моят график“.
2. JavaScript функция в "get_schedule.js" слуша за това събитие и след като го улови, изпраща GET заявка до endpoint "load_user_taken_slots.php" на сървъра.
3. Прави се заявка до базата данни, която обединява таблиците за резервации и паркоместа и извлича всички предстоящи резервации (с кода и зоната) на паркоместа на съответния потребител, сортирани по време.
4. Форматира часовете, получени от базата данни (hh:mm:ss hh:mm).
5. След всяко форматиране се записва реда в масив "taken_slots".
6. Масивът "taken_slots" се връща в JSON формат към "get_schedule.js".
7. За всяка резервация от масива се създават динамично HTML елементи, които я описват и след това се стилизират чрез CSS.

9.4. Качване на график във CSV формат

9.4.1. Функционално изискване

Регистриран потребител въвежда график на занятията си в CSV формат.

9.4.2. Резултат

Системата записва информацията от CSV файла в базата данни.

9.4.3. Начин на реализация

1. Преподавател избира подсекция „Качване на график“ от „Моят график“.
2. Преподавател избира CSV файл, който да бъде качен чрез бутона „Choose file“.

3. Преподавател финализира изпращането на файла към системата посредством бутона „Запази файл“.
4. JavaScript функция от файла "submit_csv_file.js" улавя за това събитие и създава асоциативен масив, в който добавя формата на файла като ключ и файла като стойност.
5. Изпраща POST заявка към endpoint "send_csv_to_db.php" на сървъра;
6. Проверява дали в асоциативния масив от файлове, които се получават чрез POST заявка, има качен файл в CSV формат.
7. Съдържанието на файла се разделя по нов ред ("\n") и се записва като масив от редове.
8. Проверява за наличието на заглавен ред (header) и в случай, че има го пропуска.
9. Всеки ред се разделя по запетая (','), преформатират се данните както базата данни очаква и всеки ред се записва в масив от асоциативни масиви.
10. Всеки ред от новия масив се подава на заявка, която го записва в базата данни.
11. Връща се статуса на заявката за добавяне на съдържанието на файла в базата данни.
12. JavaScript функция от файла "submit_csv_file.js" визуализира самозатварящ се прозорец със съобщение за статуса на изпълнената операция.

9.5. График на предстоящите занятия

9.5.1. Функционално изискване

Регистриран потребител прави справка за графика си със занятия.

9.5.2. Резултат

Системата визуализира списък на предстоящите занятия на преподавателя.

9.5.3. Начин на реализация

1. Потребител избира подсекция „График“ от „Моят график“.
2. JavaScript функция в "get_schedule.js" слуша за това събитие и след като го улови, изпраща GET заявка до endpoint "load_user_schedule.php" на сървъра.
3. Прави се заявка до базата данни, която обединява таблиците за графики и свързващата таблица между графики и преподаватели и извлича всички предстоящи занятия на съответния преподавател, сортирани по време.
4. Форматира часовете, получени от базата данни (hh:mm:ss hh:mm).
5. След всяко форматиране се записва реда в масив "schedules".
6. Масивът "schedules" се връща в JSON формат към "get_schedule.js".
7. За всяко занятие от масива се създават динамично HTML елементи, които го описват и след това се стерилизират чрез CSS.

9.6. Генериране на QR код

9.6.1. Функционално изискване

Регистриран потребител иска да му бъде генериран QR код за влизане в паркинга.

9.6.2. Резултат

Генерира се QR код и в зависимост от типа на потребителя има различен период на валидност.

9.6.3. Начин на реализация

1. Потребител избира бутона „Покажи QR код“ от секцията „QR код“.

2.1 Ако потребителят е „щатен“ или хоноруван, но с валиден QR код, при натискане на „Покажи QR код“ директно ще бъде извикана JavaScript функцията `displayQR()`, която след като вземе данните за потребителя от базата данни, генерира QR код, който се визуализира на екрана.

2.2 Ако потребителят е хоноруван и неговият QR код е вече изтекъл, при натискане на „Покажи QR код“ ще се изведе съобщение, че кодът е изтекъл и трябва да бъде генериран нов.

2.2.1 На мястото на „Покажи QR код“ се появява бутон „Генерирай нов QR код“, който, ако бъде кликнат, ще изпълни JavaScript функцията `get_new_qr()`, посредством която ще се обнови времето на създаване на QR кода в базата, като изпрати POST заявка към endpoint `update_qr_datetime.php` и оттам се промени стойността в базата данни. След това потребителят може да пристъпи към стъпка 1.

3. След визуализацията на QR кода се появява бутон „Скрий QR код“.

Ако потребител кликне върху него се извиква JavaScript функцията `hideQR()`, която скрива кода и текущия бутон, а на негово място се показва бутон „Покажи QR код“. Сега вече потребителят може отново да направи стъпка 1, ако желае.

9.7. Справка за свободни паркоместа в определен интервал от време

9.7.1. Функционално изискване

Регистриран в системата потребител прави справка за свободни паркоместа за дадена дата и интервал от време.

9.7.2. Резултат

Система маркира всички паркоместа в зависимост от техния статус.

9.7.3. Начин на реализация

1. Потребителят въвежда дата, час за начало и час за край на времевия интервал. При избиране на начален час, чрез JavaScript функция се определят допустимите часове съответно за край или начало на

времевия интервал. По този начин се ограничава възможността от потребителска грешка, при която се въвежда невалиден интервал от време.

2. След натискане на бутона „ТЪРСИ“, JavaScript функция съхранява въведените от потребителя данни в обект и връща формата в първоначалното ѝ състояние.
3. Системата изпраща данните от обекта в end-point "check_slots.php" на сървъра.
4. Проверява се дали в обекта има валидна дата.
5. Извличат се интервалите на всички занятия, които базата данни съдържа за потребителя за деня от обекта.
6. Създаваме и попълваме масива hours[].
7. Проверяваме дали лекторът има право да запази място в посочения от него времеви интервал чрез масива hours[].
8. Извличаме и записваме заетите паркоместа в избрания от потребителя времеви интервал в масива taken_slots[].
9. В зависимост от типа на потребителя се извличат и съхраняват недостъпните за него места в масив unavailable_slots[].
10. json форматирано съобщение със заетите и недостъпните за потребителя места.
11. Чрез JavaScript функция маркираме съответните паркоместа в цветовете съответстващи на техния статус.
12. Създаваме секция показваща избраните от потребителя дата и времеви интервал, за който е визуализирана информацията за паркоместата.

9.8. Резервиране на паркомясто в определен интервал от време

9.8.1. Функционално изискване

Регистриран в системата потребител резервира паркомясто в избрания времеви интервал от справката за търсене.

Предварително условие е да бъде изпълнена функционалност 9.7.

9.8.2. Резултат

Система запазва в базата данни резервацията, направена от потребителя.

9.8.3. Начин на реализация

1. JavaScript функция долавя събитието 'click' върху свободно паркомясто за резервация.
2. JavaScript функция от "buttons_click.js" добавя клас към HTML елемент, чрез който да се стилизира и визуализира меню за потвърждение.
3. JavaScript функция "waitForUserInput()" от "buttons_click.js" долавя отговор на потребителя.

4. JavaScript функция "sendReservationData (button)" комплектова данните за избрания бутон (код и номер на паркомясто) в асоциативна структура "reservationInfo", след което извиква функцията "makeReservation (reservationInfo)".
5. Изпраща POST заявка към endpoint "reserve_slot.php" на сървъра.
6. Изпраща заявка до базата данни, за да провери дали потребителят вече има съществуваща резервация за (част от) избрания времеви интервал.
7. Проверява дали избраното от потребител място е "lecturer_only" и на база на това определя дали паркомястото е достъпно за съответния потребител.
8. Ако потребителят е преподавател и мястото е "lecturer_only" се проверява дали преподавателят има занятие в рамките на часовия интервал.
9. Проверява се дали заявленото паркомясто е свободно или заето, чрез заявка до базата данни.
10. Ако е свободно се изпраща нова заявка до базата данни, с която паркомястото се резервира.
11. Връща JSON код с резултата от процеса по запазване на паркомясто.
12. Визуализира се самозатварящ се прозорец със съобщение за резултата от операцията.
13. При успех се извежда съобщение за успешно запазено паркомясто.

10. Принос на студента

Усилията по разработването на FMI Parking System са разпределени равномерно между разработчиците от екипа. Стилизацията и дизайна на приложението е резултат от групово взимане и имплементиране на решения, идеи и екипна работа. Въпреки формалното разделение, и в реализацията на отделните функционалности има принос на всеки един от членовете на разработващия екип.

Име	Главни функционалности
Мария Велева	9.1, 9.2
Светослав Димов	9.3, 9.4, 9.5
Андрея Дяксова	9.6, 9.7, 9.8

Таблица 5. Принос на участниците в проекта

11. Ограничения и възможности за бъдещо развитие

Идеи за бъдещо развитие на проекта:

- Възможност за промяна на данните на потребител.
- Специализиране на места за електрически коли.
- Изтриване на занятие от програмата.
- Изтриване на резервация на паркомясто.

- Игнориране на специализирането на паркоместата за „Щатни преподаватели“ и „Хонорувани преподаватели“ през неучебни периоди.

12. Какво научихме

В рамките на проекта, ние успяхме да обогатим знанията си по WEB технологии както и да подобрим работата си в екип. Някои по-конкретни неща, които научихме, са следните:

- съвместна работа с Github Desktop;
- обработка на файлове с PHP;
- употреба на JavaScript за манипулация на DOM дърво и използване на fetch API за създаване на асинхронни заявки;
- работа с XAMPP.

Като цяло екипът осъзна как разработката на макар и малка откъм функционалности система изисква много усилия, ако искаме тя да бъде напълно функционираща. Важно е задачите да бъдат приоритизирани, комуникацията да бъде постоянна и работата добре разпределена, за да се постигне успех.

13. Използвани източници

- [1] W3Schools, обновен 2022, [<https://www.w3schools.com/>], последно посетен на: 11-06-2022
- [2] JavaScript.com, обновен 2022, [<https://www.javascript.com/>], последно посетен на: 11-06-2022
- [3] The PHP Group, обновен 2022, [<https://www.php.net/>], последно посетен на: 11-06-2022
- [4] Stackoverflow, обновен 2022, [<https://stackoverflow.com/>], последно посетен на: 11-06-2022
- [5] GeeksForGeeks, обновен 2022, [<https://www.geeksforgeeks.org/>], последно посетен на: 11-06-2022
- [6] Google, обновен 2022, [<https://www.google.com/maps/>], последно посетен на: 02-06-2022
- [7] ROverhate, "male female icon", публикуван 2019, [<https://pixabay.com/>], последно посетен на 02-06-2022
- [8] Презентация "CSS-1" от "Курс по Web технологии 2021/22 (летен семестър)" в Мудъл, 18 издание, публикувана на 2020-11-17, [<http://webtech.w3c.fmi.uni-sofia.bg/w15labs/CSS-1.html#1>], последно посетена на: 02-06-2022.
- [9] Презентация "JS-1" от "Курс по Web технологии 2021/22 (летен семестър)" в Мудъл, 18 издание, публикувана на 2021-03-02, [<http://webtech.w3c.fmi.uni-sofia.bg/w16labs/pdf/JS-1-light.pdf>], последно посетена на: 02-06-2022.
- [10] Презентация "Forms" от "Курс по Web технологии 2021/22 (летен семестър)" в Мудъл, 18 издание, публикувана на 2021-03-02,

[<http://webtech.w3c.fmi.uni-sofia.bg/w16labs/pdf/Forms-light.pdf>], последно посетена на: 02-06-2022.

[11] Презентация "HTML" от "Курс по Web технологии 2021/22 (летен семестър)" в Мудъл, 18 издание, публикувана на 2021-03-01, [<http://webtech.w3c.fmi.uni-sofia.bg/w16labs/pdf/HTML-light.pdf>], последно посетена на: 02-06-2022.

14. Фигури и таблици

14.1. Фигури

Фигура 1. Диаграма на Базата данни

Фигура 2. Регистрация

Фигура 3. Вход в системата

Фигура 4. Моят акаунт

Фигура 5. Качване на график

Фигура 6. Успешно качен график

Фигура 7. График

Фигура 8. Запазени паркоместа

Фигура 9.1 Бутон за визуализация на QR код

Фигура 9.2 Генериране на QR код при изтекла валидност

Фигура 10. Генериран QR код

Фигура 11. Карта на всички зони за паркиране

Фигура 12. Преди избиране на времеви интервал

Фигура 13. След избиране на времеви интервал

Фигура 14. Избиране на паркомясто на картата

Фигура 15. Успешно запазено паркомясто

14.2. Таблици

Таблица 1. Данни на потребителите

Таблица 2. Данни на хоноруван преподавател

Таблица 3. Данни за направени резервации на паркоместа

Таблица 4. Тестови данни - график на щатен преподавател

Таблица 5. Принос на участниците в проекта