**Имена:**  Роберт Борисов **фн:**  *62529*

Владимир Радев **фн:** *62530*

**Начална година:** *2023* **Програма:** бакалавър, (СИ) **Курс: 3**   
 **Дата: 2023-06-27 Предмет: Web технологии**

**имейли:** [**robert.georgiev.borisov@gmail.com**](mailto:robert.georgiev.borisov@gmail.com)[**vladimiradev@gmail.com**](mailto:vladimiradev@gmail.com)

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

# *ТЕМА: 21.1 ФМИ паркинг*



Софийски университет “Св. Климент Охридски”

Факултет по математика и информатика

2023

Предал (подпис): ………………………….

/*фн, имена, спец., група*/

Приел (подпис): ………………………….

/доц. *Милен Петров*/

**Съдържание**

## 1. Условие………………………………………………………………………………………………………………………3

## 2. Въведение………………………………………………………………………………………………………………....4

## 3. Теория………………………………………………………………………………………………………………………..5

## 4. Използвани технологии……………………………………………………………………………………………6

## 5. Инсталация и настройки………………………………………………………………………………………….7

## 6. Кратко ръководство на потребител………………………………………………………………………...8

## 7. Примерни данни……………………………………………………………………………………………………….9

## 8. Описание на програмния код…………………………………………………………………………………..10

## 9. База данни………………………………………………………………………………….,......................................11

## 10. Приноси на студента, ограничения и възможности за бъдещо разширение……..12

## 11. Какво научих…………………………………………………………………………………………………………..13

## 12. Използвани източници…………………………………………………………………………………………..14

1. **Условие**

Да се създаде паркосистема на ФМИ и околните факултети ( ФХФ , ФЗФ) , която включва в себе си:

* Вход и регистрация на потребители в системата , като при регистрация потребител се вписва в системата с роля “Regular user” или “teacher”
* При вход потребител трябва да се препраща към login страницата.
* Администраторът на системата е само един и той не се вписва в системата, а предварително е вкаран в базата данни от разработчиците на системата.
* Системата трябва да позволява на потребител с роля “Admin” да оразмерява съответна матрица, съдържаща парко клетки , представляваща конкретен факултет от трите ( ФМИ, ФХФ, ФЗФ).
* Системата трябва да позволява на потребител с роля “Admin” да създава ново паркомясто и да му задава номер , който трябва да е уникален за всички паркоместа в цялата система
* Системата трябва да позволява на потребител с роля “Admin” да импорт/експорт-ва сумарно всички данни за създадени паркоместа, паркоклетки, размери на матрици в .csv файл.
* Системата трябва да позволява на потребител с роля “Regular user”,както и на потребител с роля “teacher” да резервират парко място на конкретен ден от час X до час Y, като X<Y;
* Системата трябва да позволява на потребител с роля “Regular user”,както и на потребител с роля “teacher” да преглеждат своите резервирани паркоместа
* Системата трябва да позволява на потребител с роля “Regular user”,както и на потребител с роля “teacher” да изтриват резервациите си.

2. Въведение

ФМИ паркинг има за цел да улесни и автоматизира процеса на запазване на паркоместа в района на факултета , като за целта ще предостави на своите потребители възможността да резервират конкретно място в конкретен диапазон от време , като така ще спести време на потребителите в търсене на свободно паркомясто и също така ще елиминира фактора на конкурентност за място като всичко ще се случва прозрачно и все видимо. Администраторът на системата пък ще има възможността да редактира и конструира нови паркоместа при евентуално разширяване на свободното място за паркиране в района на факултета.

3. Теория

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 добрите практики за семантичен HTML5 синтаксис, както и ECMAScript 6 синтаксис за JavaScript.

Логическия 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 (version 3.3.0 с включени модули *Apache* на порт 80 , както и включен модул MySQL на порт 3307 ) за поддръжка на базата данни и изпълнението на PHP скриптовете.

4.Използвани технологии

За системата са използвани технологиите , взети на занятията по време на провеждането на курса, като не са използвани никакви външни библиотеки или алгоритми за подпомагане на по-бързата и по-лесна разработка . За изграждане на системата са използвани:

1. **JavaScript**:
   1. JavaScript е скриптов език, който се използва за добавяне на интерактивност и динамично поведение на уеб страници.
   2. Той се използва за манипулиране на HTML елементи, взаимодействие с потребителя и изпращане на заявки към сървъра.
   3. JavaScript може да бъде вграден директно в HTML или да се зареди от външен файл.
2. **HTML**:
   1. HTML (Hypertext Markup Language) е стандартен маркиращ език, използван за създаване на структурата и съдържанието на уеб страници.
   2. Той използва тагове за дефиниране на различните елементи и техните взаимоотношения в уеб документа.
   3. HTML се използва главно за дефиниране на семантичната структура на уеб страниците.
3. **CSS**:
   1. CSS (Cascading Style Sheets) е език за стилови множества, използван за описание на представянето и визуалния стил на уеб страниците, написани в HTML.
   2. Той позволява контрол върху оформлението, цветовете, шрифтовете и други визуални аспекти на уеб страниците.
   3. CSS работи като асоциира стилови правила с HTML елементите, определяйки начина, по който те трябва да се показват в браузъра.
4. **PHP**:
   1. PHP е език за сървърно странично скриптиране, предназначен за уеб разработка.
   2. Той може да бъде вграден в HTML код и да се изпълни на сървъра, за да се генерира динамично съдържание на уеб страниците.
   3. PHP често се използва за задачи като достъп до база данни, обработка на формуляри и управление на сесии.
5. **MySQL**:
   1. MySQL е отворена релационна система за управление на бази данни (RDBMS), която се използва широко в уеб разработката.
   2. Тя предоставя надеждна и мащабируема платформа за съхранение и управление на данни.
   3. MySQL използва SQL (Structured Query Language), за да взаимодейства с базата данни, позволявайки изпълнение на операции като съхраняване, извличане, актуализиране и изтриване на данни. [[1]](#bookmark=id.7iiden1djl1h)

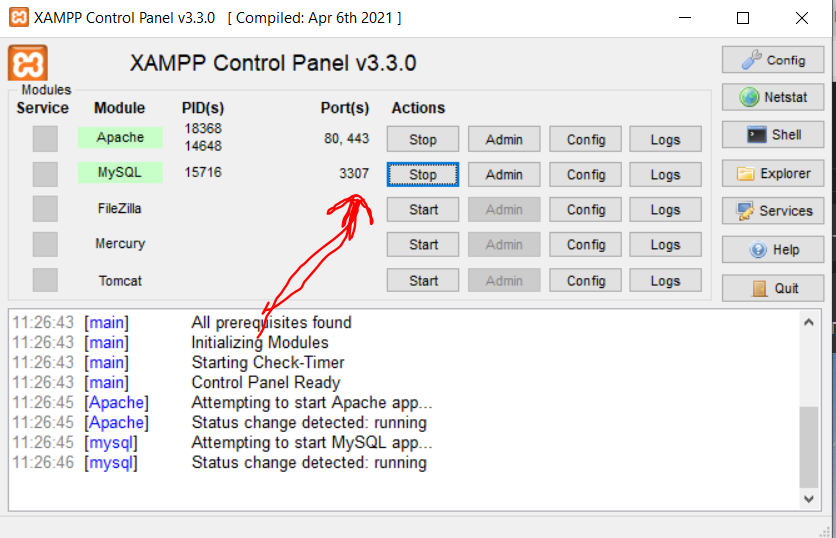
5.Инсталация и настройки

**За XAMPP v.3.3.0 (подкаран на Windows 10)** : необходимо е да стартирате модул Apache от контролния панел като стартирания порт не би трябвало да е проблемен за подкарването на проекта при вас ( при нас е порт 80) , също така трябва да стартирате модул MySQL като тук порта е важен. Ако ви се стартира на порт различен от 3307 , който е дефаултният и нашия проект е именно на него, вие трябва да замените ред номер 8 във файла:

*WEB-PROJECT-2023/backend/db/db.php* със следния ред:

$this->connection = new PDO("mysql:host=localhost:X;dbname=web\_project", "root", "");

(Където X е портът който ви е стартирал XAMPP-a. )

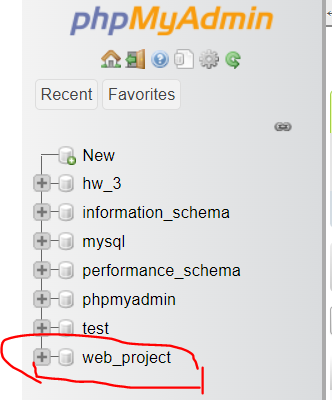


Фиг.1 Показване на порт, който трябва да вземете

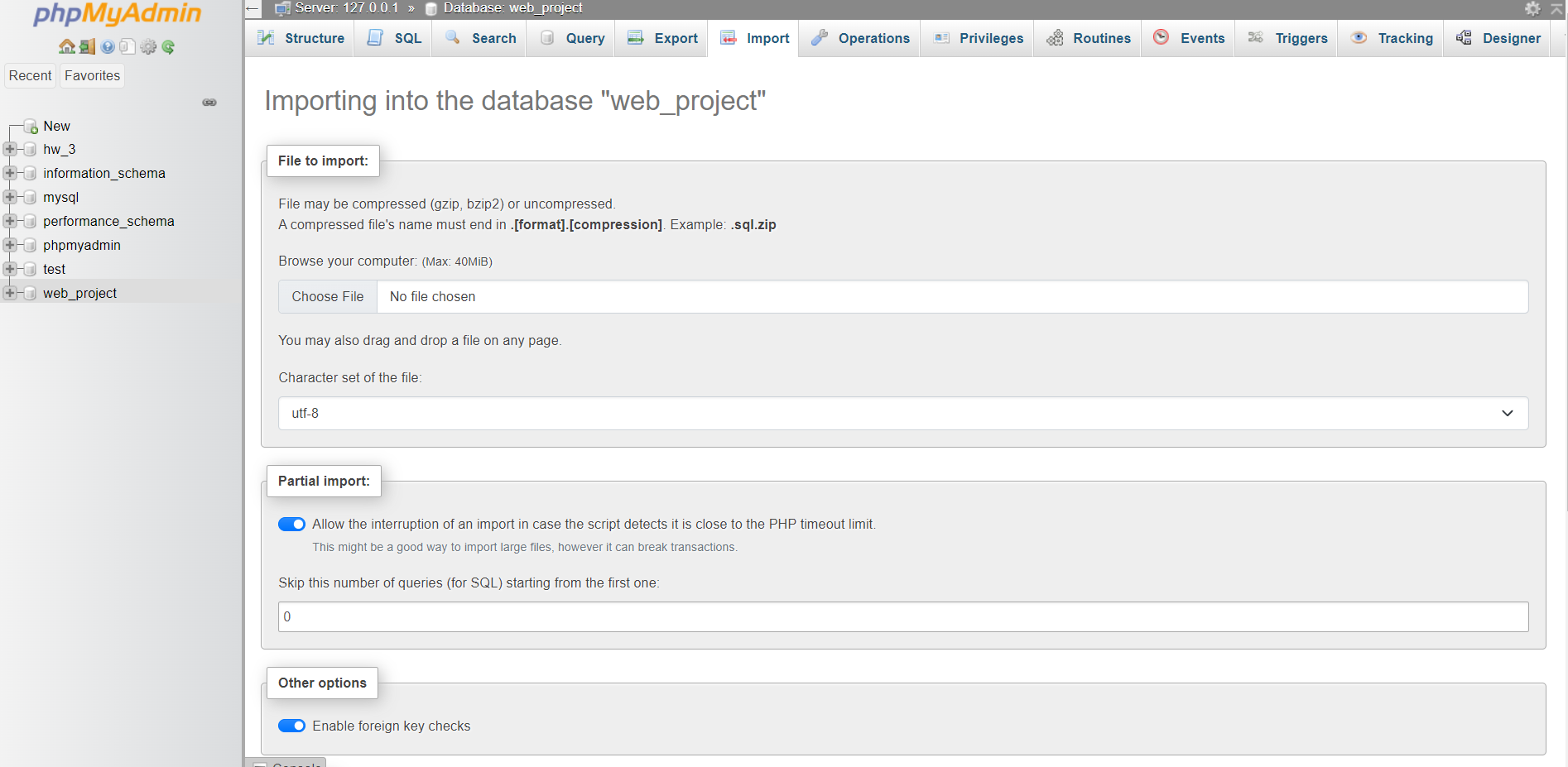
**За вмъкване на базата данни:** След като имате настроен XAMPP , трябва да отиде в phpmyadmin. Това се случва като цъкнете на ред MySQL - бутона Admin. Ще ви се зареди страницата на phpMyAdmin на дефаултния ви браузер.

От там трябва да създадете нова база данни с името *“web\_project”* ( името е важно да е написано така защото се свързваме с базата данни с това име , ако има вече съществуваща база данни с същото име , трябва да я изтриете ) .

След като имате създадена базата данни *“web\_project”* , трябва да кликнете на нея и да цъкнете прозореца “Import” от горното меню и да вкарате файл : “*web\_project.sql*” намиращ се в root директорията на проекта: “*WEB-PROJECT-2023/*” ( не трябва да променяте никакви настройки при импортването, оставете ги дефаултни).



Фиг.2 Създадена база данни в phpMyAdmin



Фиг.3 Импорт меню

**За подкарване на системата:** Трябва да сложите root папката на паркинг системата ни : “*WEB-PROJECT-2023/”* в папката htdocs/ на xamppa-a. Обикновено се намира тук : C:\xampp\htdocs , ако не е там трябва да потърсите къде сте я сложили локално на вашият компютър и да направите същата операция.

Когато всичко е сложено , може да напишете localhost в url секцията и да отидете чрез навигация и да влезете в нашата root папка, след влизане в нея ще ви се зареди началната страница на системата ни . ( страница Вход)

6.Кратко ръководство на потребител

6.1 Регистрация в системата

Потребител се регистрира в системата чрез потребителско име , имейл и парола ( за ограниченията и изискванията при попълване на полетата трябва да видите файл constraints.txt ) . При регистрация потребител може да избира между роля “ Regular user” (студент) и „Teacher” ( преподавател).

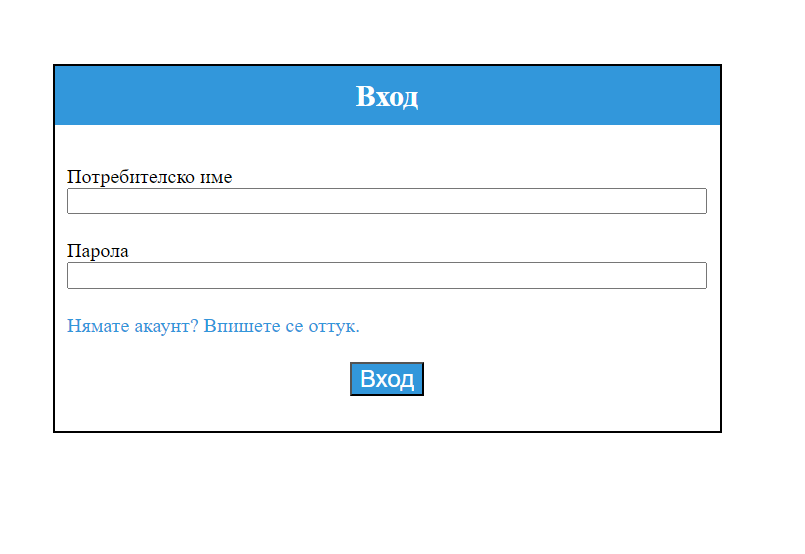
A picture containing text, screenshot, number, font

Description automatically generated

Фиг.4 Регистрация на нов потребител в системата

6.2 Вход в системата

Потребител влиза в системата с потребителско име и парола. Това е началната страница на системата.



Фиг.5 Вход на регистриран потребител в системата

6.3 Начална страница на потребител с роля “Admin”

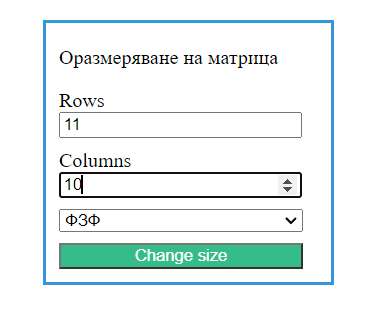
A screen shot of a parking system

Description automatically generated with medium confidence

Фиг.6 Начална страница на потребител с роля “Admin”

6.3.1 Оразмеряване на матрица съответстваща на избран факултет ( ФМИ, ФХФ, ФЗФ )

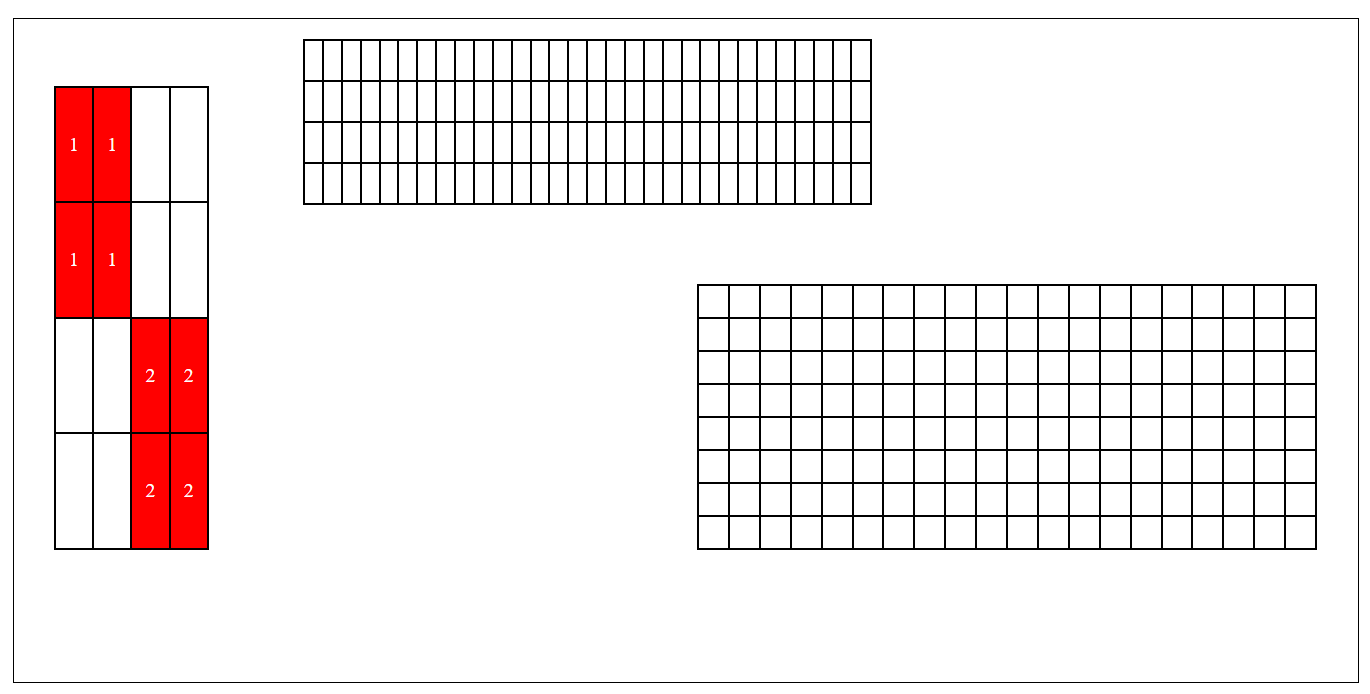
Потребител с роля “Admin’ има възможността да оразмерява матриците на факултетите като избира съответно стойност за броят на редове и броят на колоните и факултета за който иска да направи промяната. При оразмеряване с матрица по-малка от конкретната, т.е въведен размер по-малък от текущия , съответно по редове и/или по колони - информацията която е била въведена вече върху нея ще бъде изтрита.,както и всички резервации свързани с дадените паркоместа на конкретния факултет.При въвеждане на размери по-големи или равни от текущия , загуба на информация няма да има.



Фиг.7 Оразмеряване на ФЗФ до матрица с 11 редове и 10 колони

6.3.2 Създаване на ново паркомясто на избран факултет ( ФМИ, ФХФ, ФЗФ )

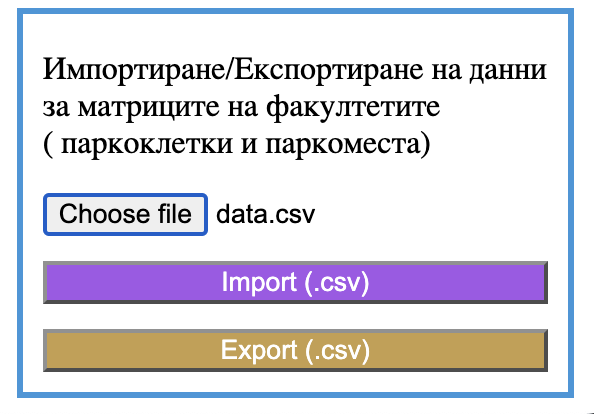
Потребител с роля “Admin” ще може да създава ново паркомясто след като е натиснал бутона “Create new park place” за съответния факултет и чрез натискане на парко клетки върху съответния факултет да очертае новото паркомясто.

След това запазва промените с бутона “Save new park place” като трябва да избере и номер на паркомястото. 

Фиг.8 Създаване на паркоместа за ФЗФ с номера 1 и 2

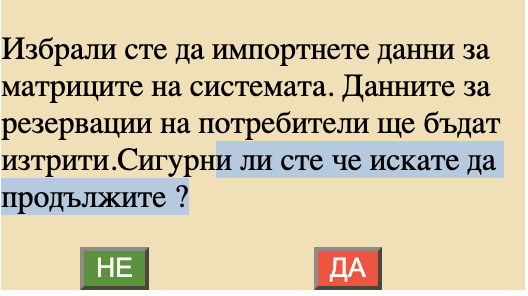
6.3.3 Export/Import CSV

Потребител с роля “Admin” ще може да импортва и експортва към/от **CSV. CSV** файлът съдържа информация за факултетите,колко колонки, колко редове да съдържат, както и информация за паркоместата и техните разположения върху матриците на дадените факултети.При импортване на **CSV** файл се генерират новите паркоместа и се трият съществуващите резервации в базата данни.



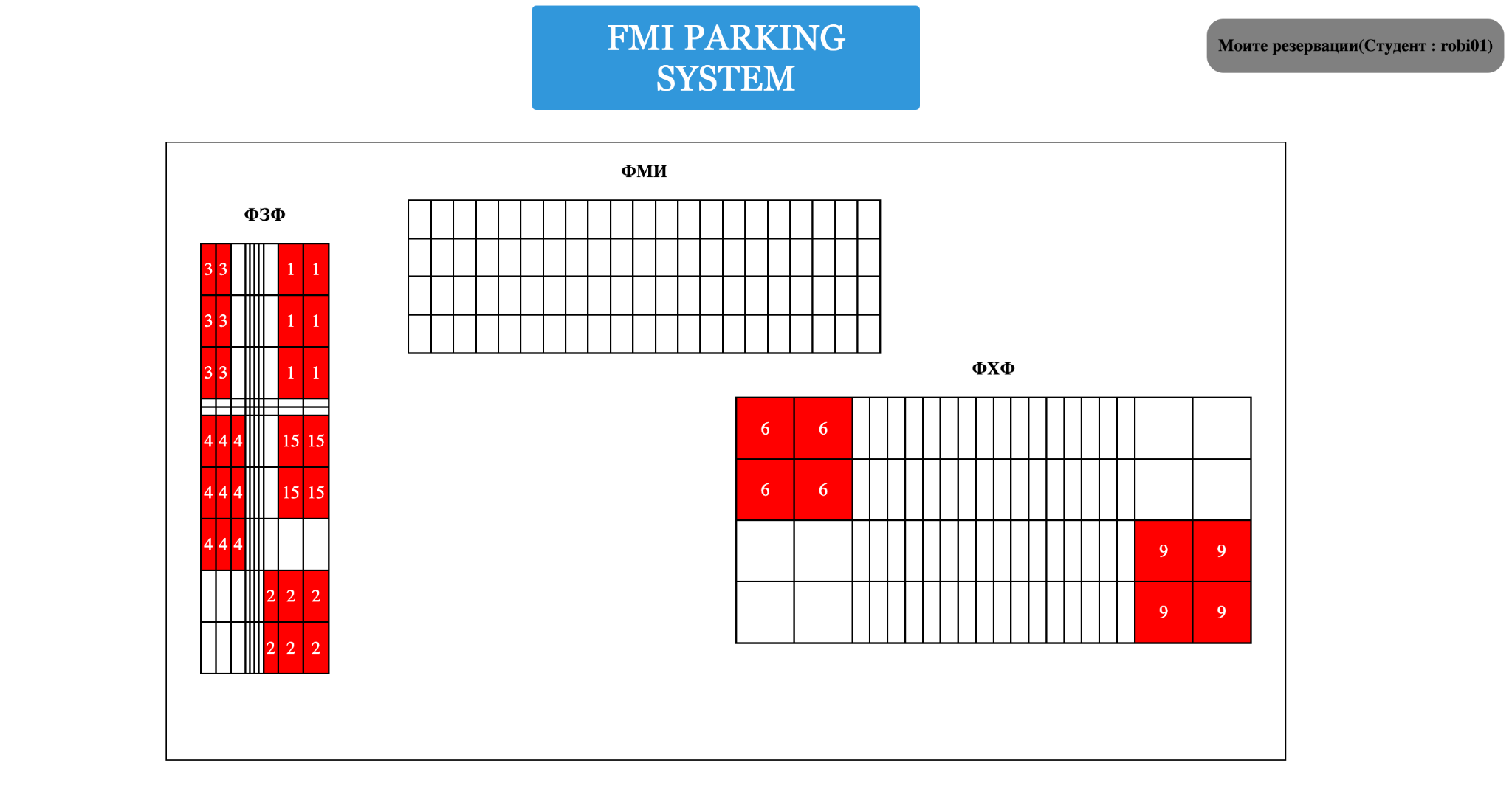
Фиг.9 Импортване на файл **data.csv**

След като натиснем бутонът **Import(.csv)** ни излиза съобщение ,което ни уведомява,че данните за съществуващите резервации на паркоместа ще бъдат изтрити от базата данни и ни пита дали искаме да продължим.



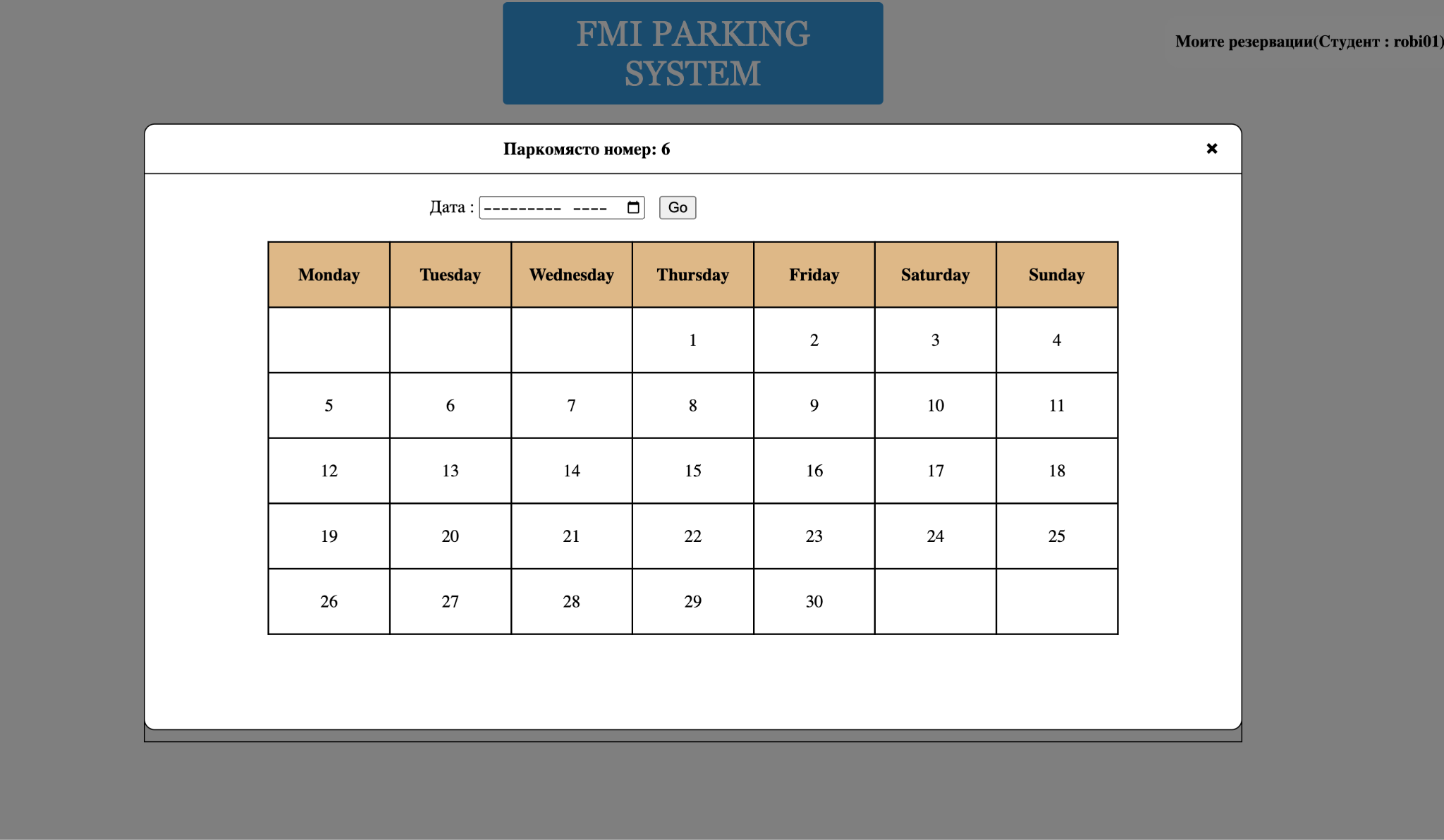
Фиг.10 Уведомпление при импортване на **CSV** файл

6.4 Начална страница на потребители с роля “Regular user” и “Teacher”



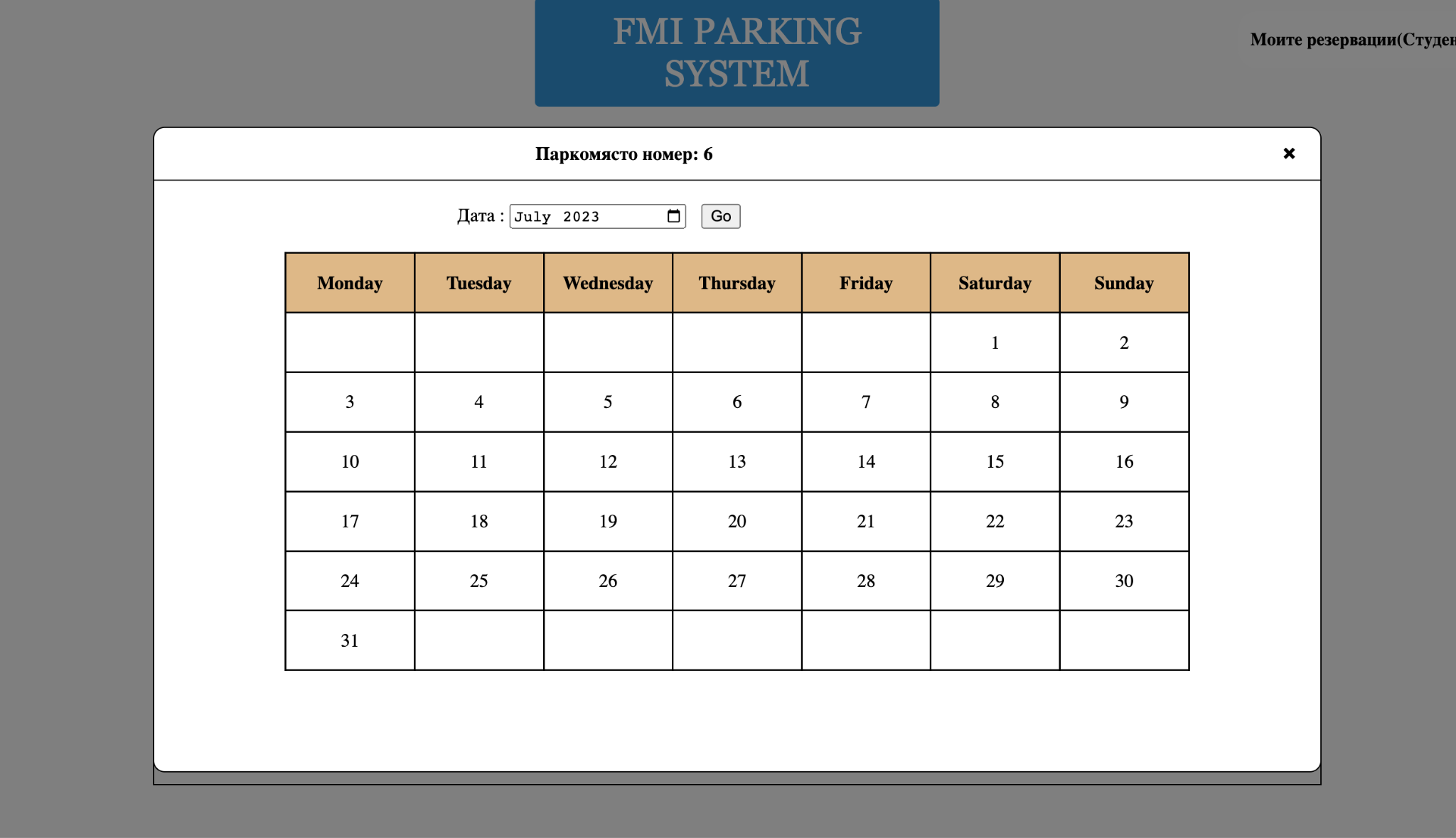
Фиг.11 Начална страница на потребители с роля “**Regular user**” и “**Teacher**”

6.4.1 Показване на календар

При поставяне на мишката върху някое паркомясто, всички “cell”-чета или всички квадратчета асоциирани с него се оцветяват в синьо.При натискане на дадено парко място,излиза pop-up прозорче на което е визуализиран актуален календар на текущия месец под формата на таблица.

Фиг.12 Календар за текущия месец при отваряне на конкретно паркомясто

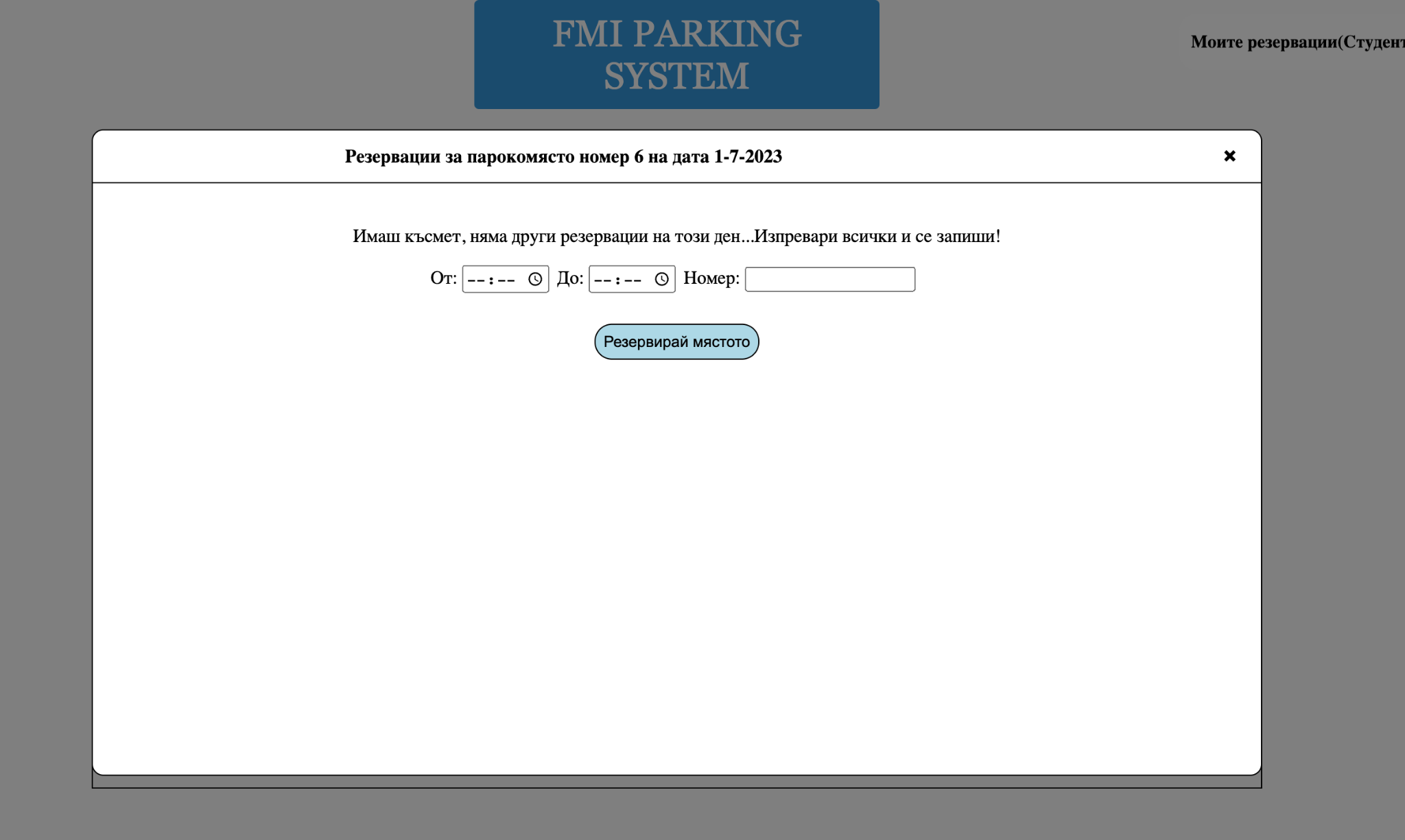
Над календара има поле в което може да въведем съответно месец и година и при натискане на бутонът “Go” ни се генерира календар на съответния Месец и Година



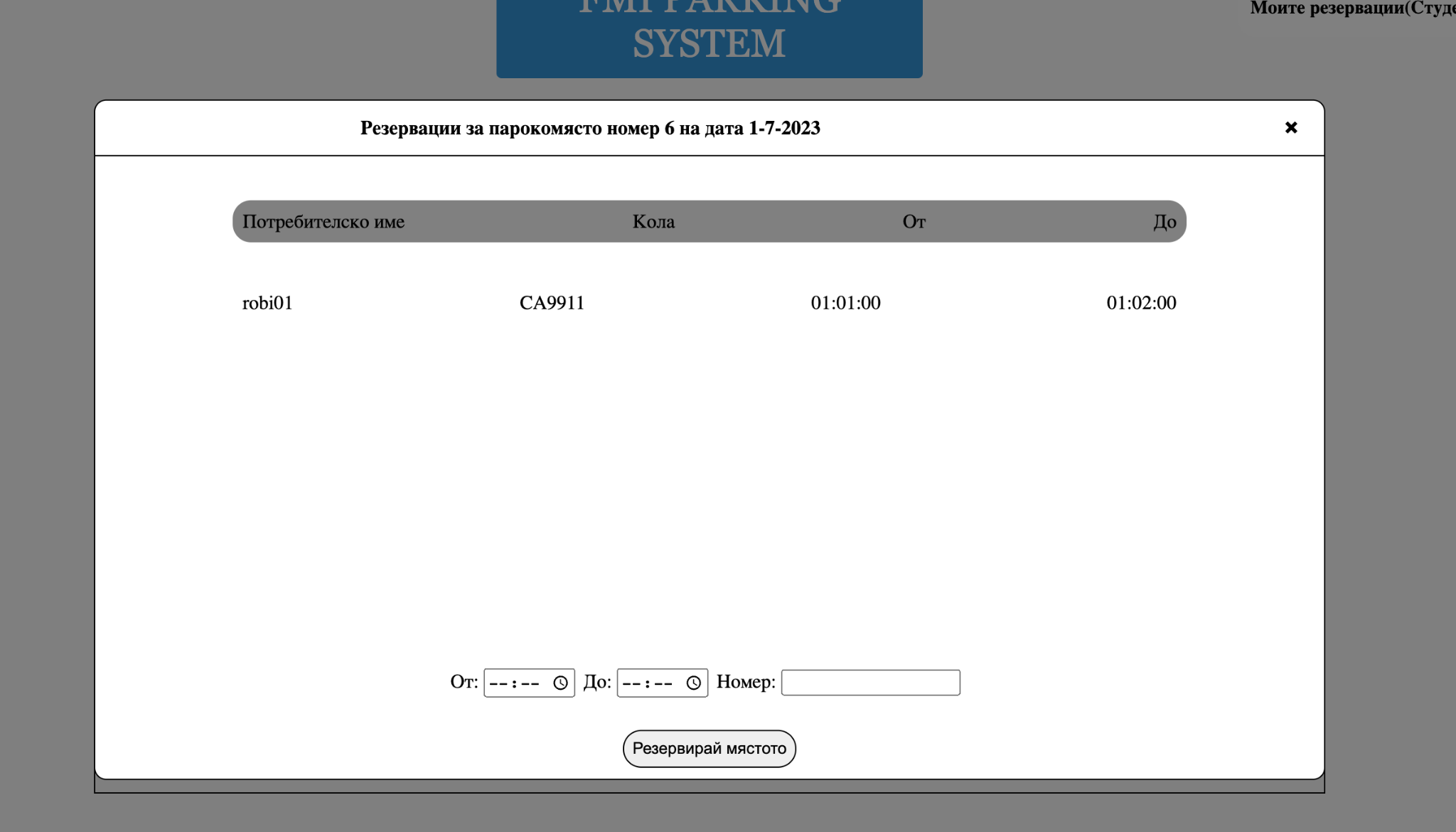
Фиг.13 Календар на месец юли,2023 година

6.4.2 Резервиране на паркомясто

При избиране на някое от датите на месеца,например 1 юли 2023 година ни се визуализират резервациите на конкретното парко място за конкретната дата.Ако няма такива ни излиза съобщението - “Имаш късмет, няма други резервации на този ден...Изпревари всички и се запиши!”.

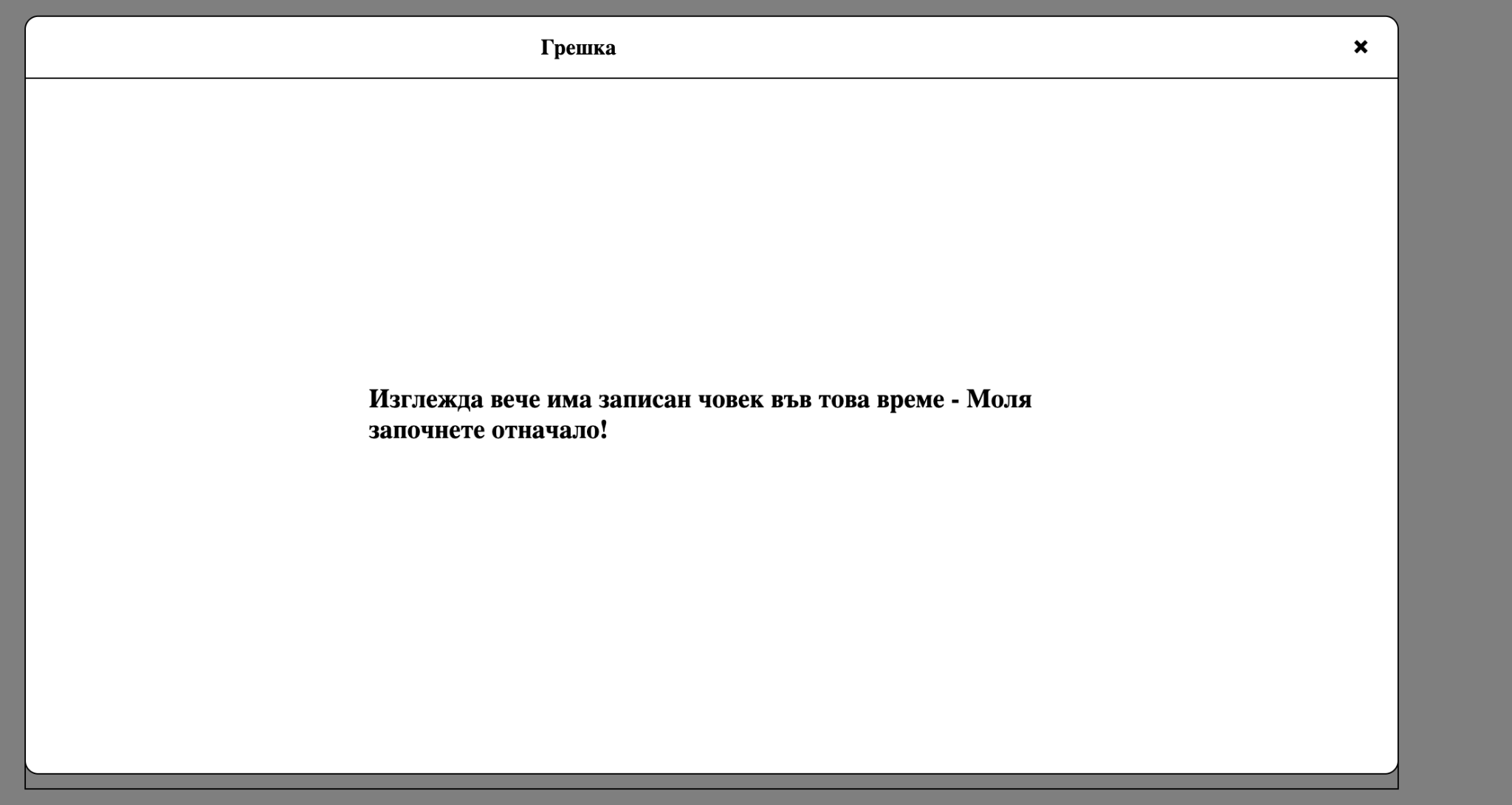


Фиг14.Паркомясто ,за което няма резервации на конкретна дата



Фиг15. Паркомясто ,за което има резервации на конкретна дата

И в двата случая имаме формичка с данни “От”,”До” - тип време(например 9:30) и Номер на кола.След натискане на бутонът “Резервирай мястото”, ако данните са валидни се записва в базата данни и резервацията на мястото е успешно.(Под валидни данни се има предвид времето-ден,месец,година,време са по-големи или равни на текущото,дали има регистриран човек за този времеви период и други).



Фиг16.Примерно съобщение за грешка при некоректни данни.

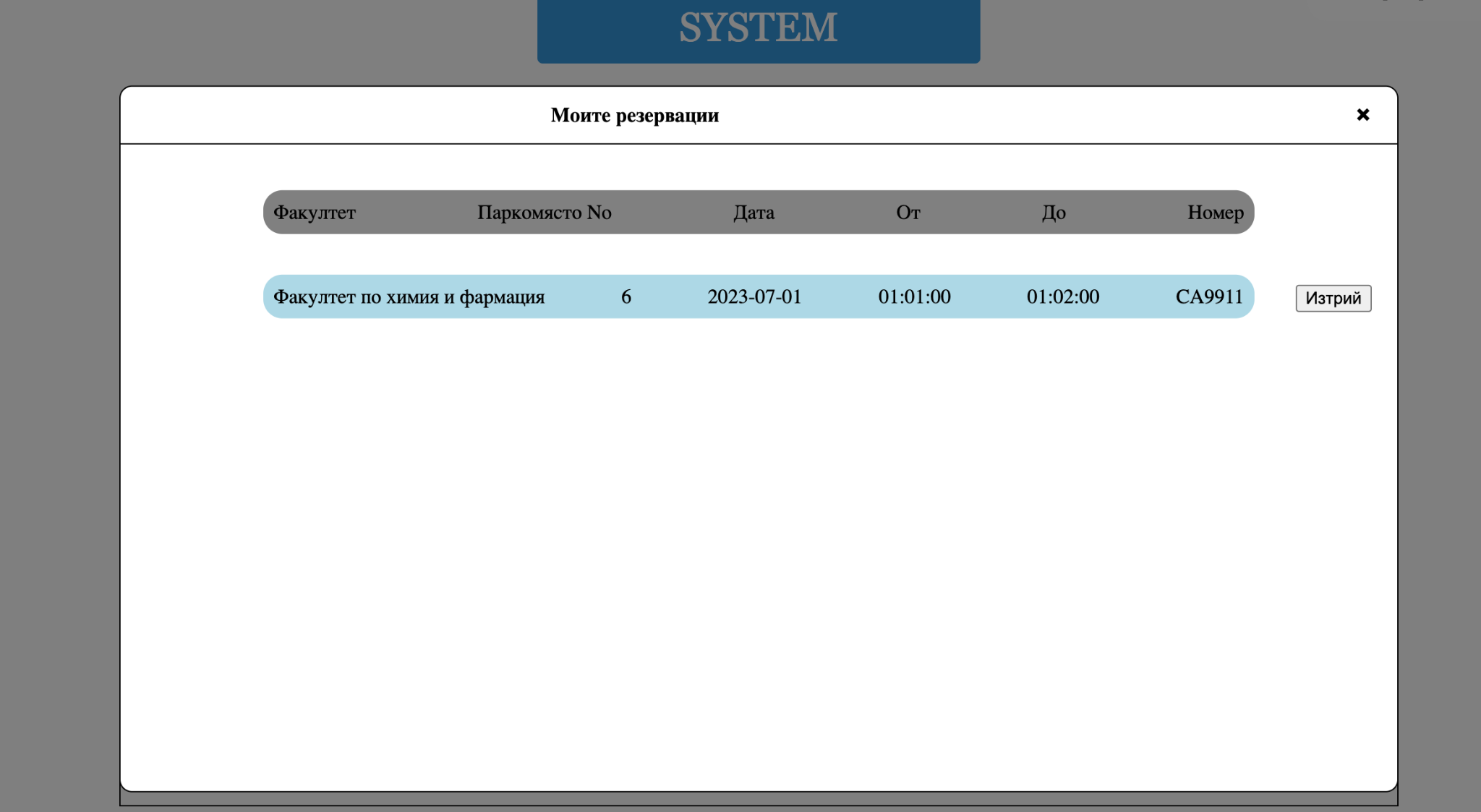
При грешка за некоректни входни данни, потребителят започва отначало.

6.4.2 Показване на “Моите резервации”



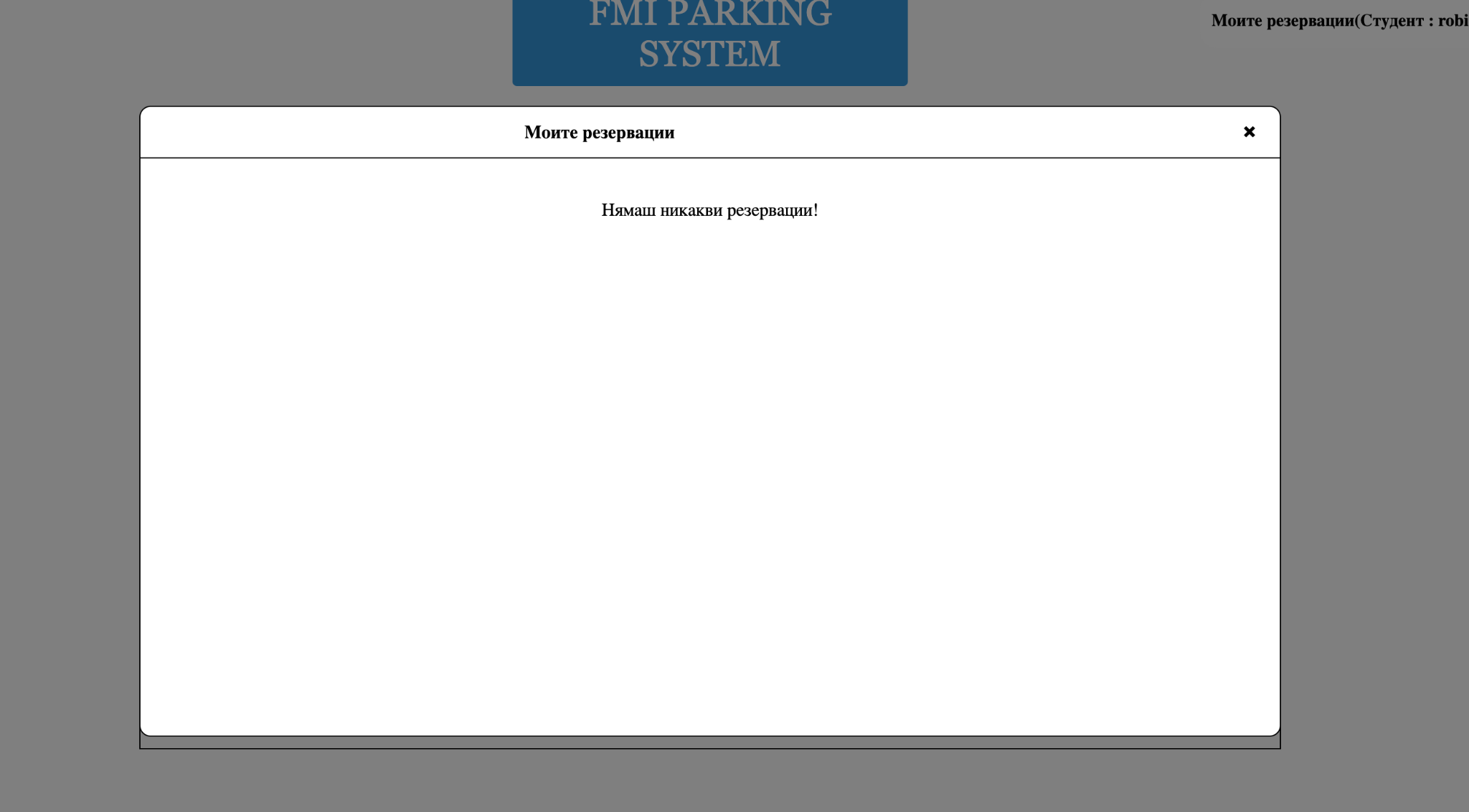
Фиг16.”**Моите резервации**”

При натискане на “Моите резервации(Роля:Потребиелско\_име) , което се намира горе в дясно ни се отваря pop-ъпче с резервациите на текущият потребител.



Фиг17.Резервации на текущия потребител

На тази страничка текущият потребител може да преглежда своите резервации, както и да отменя(трие) всяка една от тях.



Фиг18.Страница “Моите резервации” на потребител с 0 резервации

7. Примерни данни

7.1. Админ

Потребителско име : admin

Имейл : admin@abv.g

Парола : 12345

7.2. Regular потребител

Потребителско име : regular\_student

Имейл : regular\_student@abv.bg

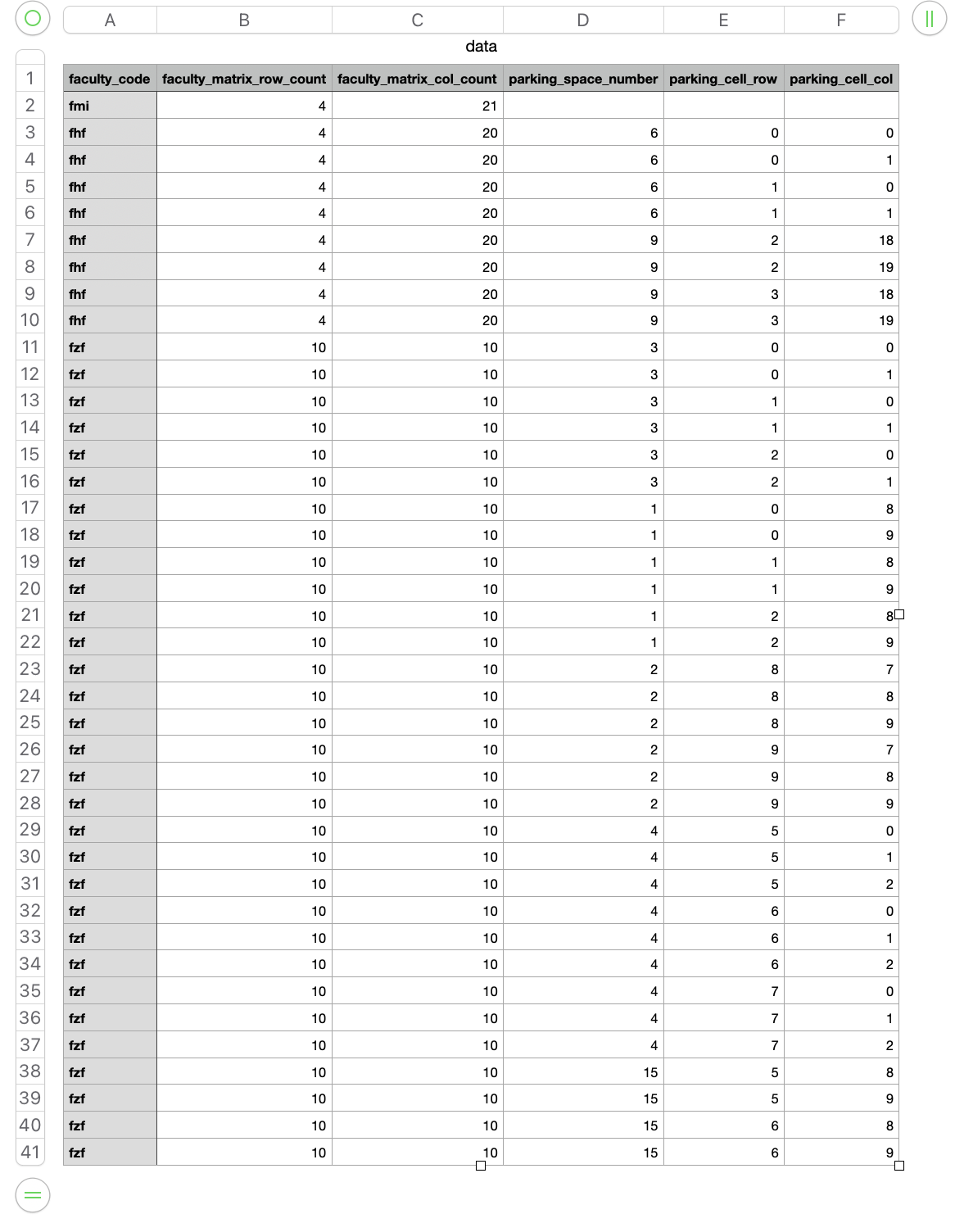
Парола : 12345

7.3. Teacher потребител

Потребителско име : teacher

Имейл : teacher@abv.bg

Парола : 12345



Фиг18.Примерен файл **data.csv**

Примерният файл **data.csv** включва факултетен код,брой на редове и колони в даден факултет,номер на парко място,както и колона и ред в което се намира конкретното cell-че.

8. Описание на програмния код

Система ФМИ Паркинг съдържащата:

0. index.html

Това е началната страница на системата. Представлява входна форма, в която потребителите да въведат потребителско име и парола и да влезнат. Имат опция и да цъкнат на текстов бутон , който да ги отведе в регистрационната форма.

**1.backend**

**1.1 api**

1.1.1 deleteAllParkCelssOfCertainFaculty.php

Изтрива всички парко клетки от базата данни първо които реферират към парко местата , които ги съдържат.

1.1.2 deleteAllParkSpacesOfCertainFaculty.php

Изтрива всички паркоместа от базата данни. Това се случва при оразмеряване на матрицата с по-малки размери.

1.1.3 **getCellsInfo.php**

Взима информация от базата данни за клетките , когато се зареждата основните страници на потребителите.

1.1.4 getMatrixSize.php

Взима информация за размерите на матриците от базата данни.

1.1.5 login.php

Проверя дали съществува потребител с такива входни данни в базата данни.

1.1.6 register.php

Записва в базата данни новорегистриран потребител след проверки за коректност.

1.1.7 **savingParkPlace.php**

Записва ново паркомясто в базата данни като запазва и паркоклетките, които го изграждат

1.1.8 **updateMatrixSize.php**

Записва данните за ново оразмерена матрица в базата данни.

1.1.9 **deleteAllReservationsOfCertainFaculty.php**

Изтрива всички резервации свързани с даден факултет

1.1.10 **Calendar.php**

Клас ,чрез който се генерира календарът

1.1.11 **deleteReservation.php**

Изтрива резервация по подадено “id” на регистрация.

1.1.12 **getCalendarInfo.php**

Генерира календар използвайки Calendar.php

1.1.13 **getExportData.php**

Събира информация при експортване на данни.

1.1.14 **getReservationInformation.php**

Събира информация за резервациите на конкретно паркомясто

за конкретна дата.

1.1.15 **getReservationsOfCurrentUser.php**

Събира информация за всички резервации на текущия потребител.

1.1.16 getUserData**.php**

Събира информация за текущия потребител - username и роля

1.1.17 **reserveSpace.php**

Резервира парко място след проверка на коректност на данните.

**1.2 db**

1.2.1 db.php

PHP файл съдържащ класа DB , който ни връща конекция към базата данни.

**2. frontend**

**2.1 admin**

2.1.1 home.css

Стилови множества за началната страница на потребител с роля “Admin”.

2.1.2 home.html

HTML файл ,в който се намира html елементите изграждащи началната страница на потребител с роля “Admin”

2.1.3 home.js

Java script файл, в който се намират eventListeners на всички основни бутони и html елементи ,с които се изгражда логиката зад основните функционалности на админа : оразмеряване на матрица, създаване на ново паркомясто , канселиране на създаване на ново паркомясто , ограждане и определяне на конкретно паркомясто , резервиране на паркомясто , както и import/export. В файла се извикват асинхорнни заявки(fetch с get и post методи, както и json обекти в body-то при post заявка) към сървъра , от които взимаме отговори и боравим с получената информация.

**2.2 login**

2.2.1 login.js

Тук изпращаме към login.php конструиран json обект за валидация на въведените данни от index.html и при успех препращаме потребителя към съответната начална страница спрямо ролята му.

**2.3 registration**

2.3.1 registration.html

HTML файл , съдържащ регистрационна форма с полета: потребителско име , имейл, парола

2.3.2 registration.js

Тук изпращаме json обект с въведените данни към registration.php , където се обработва и ни връща информация съответно за валидна регистрация или не , ако е невалидна регистрацията се изписва съобщение до потребителя къде трябва да поправи данните си , а ако е валидна се препраща към regular/home.html.

**2.4 regular**

2.4.1 home.html

Основна страница на потребител с роля “Regular User”

2.4.2 home.js

Тук се намира цялата логика за eventListener-ите,както и

изпращане и получаване на данни към backend api-тата чрез

асинхронни fetch заявки. Динамично променя структурата на HTML

файла.

2.4.3 home.css

Стилов лист свързан със “**home.html**” на **regular user**

**2.5 styles**

2.5.1 login-registration-styles.css

Oбщи стилови множества за index.html и registration.html

**3. web\_project.sql**

Sql файл за създаване на таблиците от базата данни, връзките между таблиците, както и информацията складирана в тях. Този файл се използва при конфигуриране на системата локално при вас ( начинът на използване е описан в т. 5. Инсталация и настройки )

9. База данни

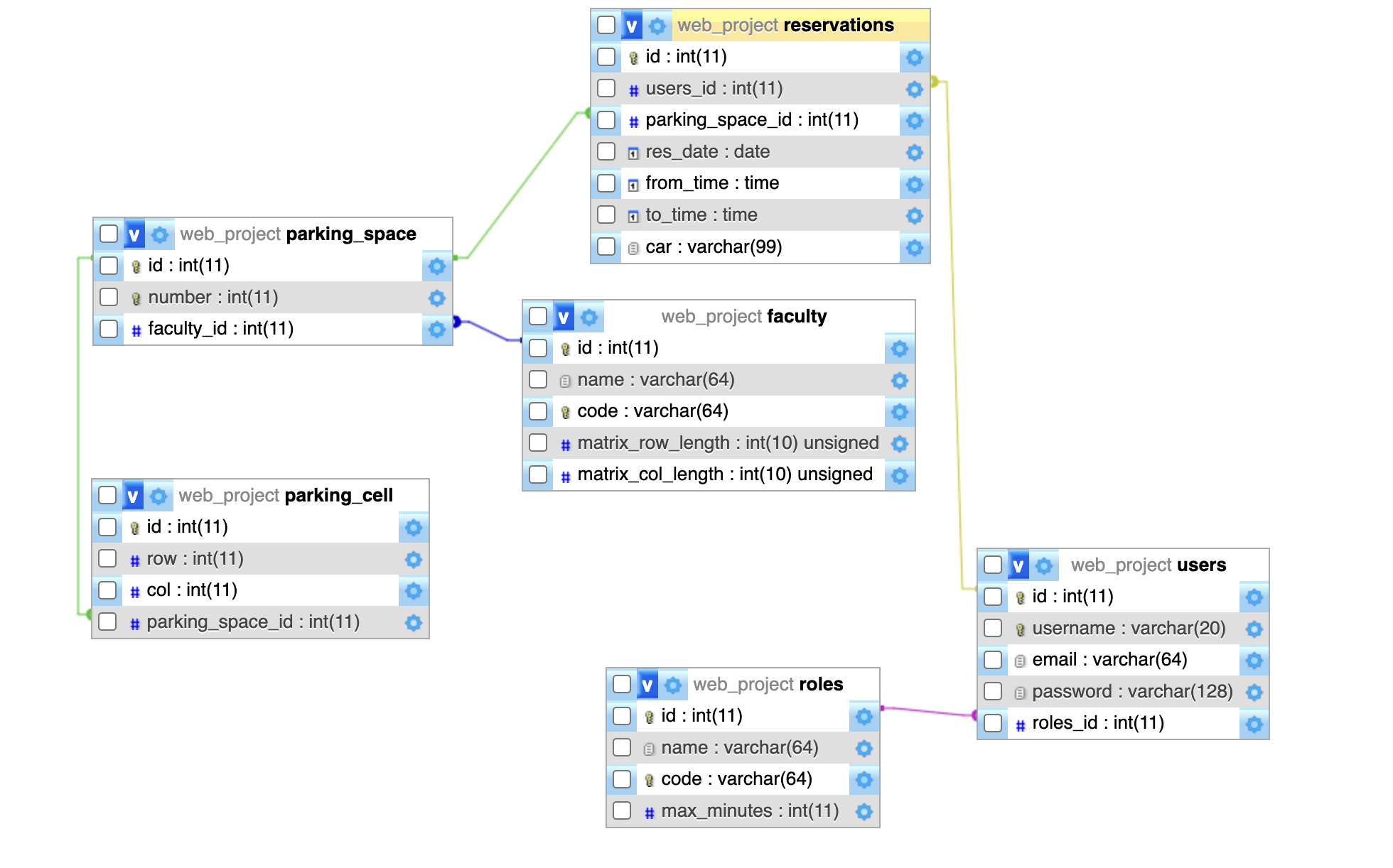
9.1 Структура

* roles - таблица която съдържа всички роли, които поддържа системата
* users- таблица , която съдържа информация за потребителско име , имейл , хеширана парола на потребител , както и foreign key към таблицата roles , като така можем да разберем всеки потребител с каква роля е.
* faculty- таблица съдържата информация за факултети , които поддържа нашата система както и информация за размерите на матрицата на този факултет
* parking\_cell - таблица съдържата информация за паркоклетка, ред и колона от матрицата , както и foreign key към таблицата parking\_space , за да знаем кои парко клетки се съдържат в едно паркомясто
* parking\_space - таблица с колона за номер на паркомясто, като този номер трябва да е уникален за цялата система като цяло , не само за факултета , върху който се намира това паркомясто
* reservations - таблица обвързваща потребителите със запазените от тях паркоместа , както и още налична инфформация свързана с резервацията.

9.2 Ключове

|  |  |  |
| --- | --- | --- |
| Таблица | Primary key | Foreign keys |
| roles | id |  |
| users | id | roles\_id |
| faculty | id |  |
| parking\_cell | id | parking\_space\_id |
| parking\_space | id | faculty\_id |
| reservations | id | users\_id, parking\_space\_id |

9.3 Дизайн



Фиг.8 Дизайн на базата данни

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

* **Приноси на студента**

Ако разделим проекта на няколко основни функционалности : оразмеряване на матрица, създаване на паркомясто, резервиране на паркомясто, импорт/експорт , то разпределението на работата е следното:

* оразмеряване на матрица, създаване на паркомясто ,импорт/експорт - Владимир
* резервиране на паркомясто,преглед на моите резервации,, импорт/експорт - Роберт
* **Възможности за бъдещо разширение**

За следващо разширение на проекта ние бихме предложили :

* да има възможност за редактиране на паркомясто
* да може да се създаде алгоритъм за запазване на максимално много паркоместа при оразмеряване с по-малка матрица
* да има възможност за следене на история на запазени места.
* да има опция за промяна на парола на потребителски акаунт.
* по-добра визуализация на паркоместата

## 11. Какво научихме

В рамките на ФМИ паркинг, екипа по разработването на проекта успя да обогати познанията си в WEB технологиите както и екипната си работа. Някои по-конкретни неща, които научихме са следните:

1. Манипулация на DOM дърво с елементи от HTML-a чрез javascript.
2. Взаимна работа между html - javascript- php файлове чрез асихронни заявки (fetch с get, post методи) и обработка на данните.
3. Асихронност на javascript и справяне с това
4. Извличане, обработка и вмъкване на информация в база данни посредством php.

**12. Използвани източници**

[1] Генериран отговор на запитването : “can u tell me information to fill theory bullet of documentation of creating a web project consists java-script,html,css,php,mysql and translate to bulgarian” в онлайн платформата [Chat-GPT by OpenAI](https://openai.com/blog/chatgpt), последно посетен: 2023-06-22.

[2] Генериран отговор на запитването : “how to add navigation to other part of word document” в онлайн платформата [Chat-GPT by OpenAI](https://openai.com/blog/chatgpt), последно посетен: 2023-06-22.