

Имена: Мария Иванова
Николай Костандиев

фн: 9MI0600045
фн: 6MI0600046

Начална година: 2021 **Програма:** бакалавър, (СИ) **Курс:** 3

Тема: Система за медицински прегледи

Дата: 2024-06-14 **Предмет:** w24prj_SI_final **имейл:** mariami@uni-sofia.bg
nkostandie@uni-sofia.bg

преподавател: проф. д-р Милен Петров

ТЕМА: w24/53.1 Система за медицински прегледи

1. Условие

Целта на проекта е да се създадена система за резервация на медицински прегледи.

Потребителят може да търси лекар с опция за филтрация по специалност или регион. При избран лекар от показания списък, потребителя вижда информация за него. Това включва обща информация, както и отзиви и оценки, написани от други пациенти. Изобразяват се също свободните часове при този лекар, налични за резервация.

От показания списък със свободни часове, потребителя може да запази час за преглед само ако е регистриран в системата като пациент и е влязъл в акаунта си. Ако потребителя е влязъл в профила си с роля на пациент, то той може да избере един от показаните часове и да го запази. При направена резервация, тя се записва в базата данни, и се изобразява на пациента и на лекаря.

Пациентът може да преглежда всички свои минали и бъдещи прегледи. След като е преминал преглед, пациента може да остави оценка и отзив за него. Ако прегледът не е минал, то пациентът има опция за отказване на прегледа.

Потребителят с роля на лекар, може да добавя свободни часове, които се изобразят на другите потребители и могат да бъдат запазвани. Той може да види своите резервирани часове, както и тези, които са все още незаети. Лекаря има възможност да остави бележка към даден преглед, която ще бъде показана на пациента. Ако прегледът не е минал, то лекарят има опция за отказване на прегледа.

2. Въведение – извличане на изисквания

Системата ще поддържа 3 типа потребители:

1. Гост, който ще може да се регистрира като пациент или лекар, да влиза в профила си, да търси лекари с опция за филтриране по специалист и регион. Също ще може да отравя страницата на лекар, където ще бъде представена информация за него: обща информация, отзиви и оценки от пациенти, свободни часове за преглед.
2. Пациент, който ще има всички права на гост потребителя, както и ще може да запазва някой от свободните часове, представени в страницата на лекар. Освен това пациента ще може да преглежда всички свои резервации, за всяка отминала ще може да оставя оценка и кратко ревю, и ще може да отменя някоя от тях, ако още не е минала.
3. Лекар, който ще може да предоставя свободни часове за прегледи, които пациентите ще могат да запазват. Към всяка една резервация за преглед, лекарите ще могат да оставят допълнителна бележка, която да е от помощ на пациентите, и ще могат да отменят резервация, ако още не е минала.

Функционални изисквания:

1. Гост потребителя може да се регистрира като пациент или лекар в системата.
2. Гост потребителя може да влиза в профила си.
3. Пациент или лекар, влязъл в профила си, има опция да излезе от него.
4. Пациентът и гост потребителят могат да видят списък с всички лекари.
5. Пациентът и гост потребителят могат да търсят лекари по име.
6. Пациентът и гост потребителят могат да филтрират търсенето по регион и/или специалност.
7. Пациентът и гост потребителят могат да отворят страница на лекар от предоставения списък.
8. Пациентът и гост потребителят могат да преглеждат обща информация, оценки и отзиви, и списък със свободни часове на лекар от страницата му.
9. Пациентът може да резервира преглед при лекар, избран от списъка със свободни часове от страницата на лекаря.
10. Пациентът може да прегледа историята от всички прегледи, които е посетил или му предстоят.
11. Пациентът може да оставя отзив и/или оценка за вече отминал преглед.
12. Пациентът може да вижда бележка към даден предстоящ или изминал негов преглед, дадена от лекаря.
13. Пациентът може да отменя преглед, който е резервирал, но все още не е преминал.
14. Лекарят може да преглежда своя график с резервирани часове.
15. Лекарят може да оставя бележка за негов, резервиран от пациент, преглед.

16. Лекарят може вижда отзив и/или оценка към даден изминал преглед, дадени от пациента.
17. Лекарят може да отменя негов преглед, който е резервиран, но все още не е преминал.
18. Лекарят може да предоставя свободни часове за прегледи.
19. Лекарят може да преглежда часовете от графика му, които са все още свободни.
20. Лекарят може да отменя преглед, който не е резервиран.

Нефункционални изисквания:

1. Сигурност – Системата трябва да предоставя 99% ниво на сигурност, за да защити личните данни на потребителите и медицинските им записи:
 - a. Паролите на потребителите трябва да се съхранява в криптиран вид.
 - b. Правата за достъп до различните функционалности и страници на системата е ограничена, спрямо това дали потребителят е влязъл в профил и ролята на профила му (пациент или лекар).
2. Надеждност – Системата трябва да бъде надеждна и да осигурява безпроблемно функциониране, за да гарантира удобен достъп на потребителите:
 - a. Системата трябва да бъде достъпна за потребителите 95% от времето на месечна база. Допустимо е да се извършват прекъсвания за поддръжка веднъж в месеца между 02:00 – 03:00, след като потребителите са били своевременно предупредени не по-късно от 48 часа преди това.
3. Производителност – системата трябва да бъде бърза и ефективна, особено по времена резервации, за да се осигури гладко и безпроблемно потребителско изживяване:
 - a. Системата трябва да може да обработва до 10,000 едновременни потребители без забележимо забавяне или загуба на данни.
 - b. Средното време за отговор на заявка от потребителски интерфейс трябва да бъде под 1 секунда при нормално натоварване и под 3 секунди при пиково натоварване.
4. Лесно използване и интерфейс:
 - a. Дизайнът на интерфейса трябва да бъде консистентен във всички страници и модули, използвайки единни цветови схеми, шрифтове и икони.
 - b. Навигационните елементи трябва да бъдат поставени на същите места във всички страници, за да се улесни използването от потребителите.
 - c. Системата трябва да предоставя незабавна обратна връзка за действията на потребителя (напр. потвърждение при успешна резервация, съобщение за грешка при неуспешно действие).
 - d. Всички форми и полета трябва да включват ясни съобщения за грешки и предложения за корекции, когато въведените данни са невалидни.
5. Скалабилност:

- a. Системата трябва да поддържа хоризонтална скалируемост, което означава, че може да се добавят допълнителни сървъри или виртуални машини при увеличаване на натоварването, без да се налагат значителни промени в архитектурата на системата.
6. Съвместимост:
- a. Системата трябва да бъде съвместима с основните операционни системи, включително Windows, macOS и Linux.
 - b. Уеб базираното приложение трябва да бъде съвместимо с всички модерни уеб браузъри, включително Google Chrome, Mozilla Firefox, Microsoft Edge и Safari.
 - c. Интерфейсът трябва да бъде тестван и оптимизиран за различни резолюции и устройства (десктоп, таблети, смартфони).
7. Издръжливост:
- a. Системата трябва да има механизми за автоматично възстановяване при грешки, които да възстановяват работоспособността на услугата до 15 минути след отказ
 - b. При отказ на някоя услуга, това не трябва да влияе върху работата на останалите услуги, които не са свързани с нея.
 - c. Системата трябва да бъде редовно тествана за издръжливост, включително симулация на високи натоварвания.
8. Съвместимост с регулаторни изисквания:
- a. Системата трябва да съответства на всички регулаторни изисквания и стандарти за сигурност и защита на личните данни, които са приложими за медицинските системи.

Ползи от реализацията:

- 1. Улесняване на управлението на графика на лекарите, регистрирани в системата.
- 2. Улесняване на процеса по резервирането на час при медицински специалист.
- 3. Улесняване на потребителите при техните посещения при медицински специалисти като те получават бележка от лекаря, свързана с прегледа им.
- 4. Подпомагане на потребителите в намирането на подходящ специалист, като предоставя информация за всички ревюта и оценки за наличните специалисти.

3. Теория – анализ и проектиране на решението

Архитектура:

Системата следва три-слойна архитектура. Тя включва презентационен слой, слой на бизнес логика и слой на база данни. Това осигурява ясно разграничение между различните функционалности на приложението, което улеснява поддръжката, разширяемостта и тестването на системата.

1. Презентационен слой:

Този слой е отговорен за визуалното представяне и взаимодействията с потребителя и включва PHP файлове с HTML и PHP код; CSS файлове за стайлинг и оформление; JavaScript файлове за динамично поведение на функционалностите; снимки, използвани в системата. HTML е използван за структуриране на уеб страниците, което включва оформяне на хедъри, създаване на формуляри, изобразяване на данни в табличен вид и други. PHP е използван за проверки и валидации, извикване на функции и обработка на данните, върнати от тези функции. С CSS е създаден удобен и привлекателен потребителски интерфейс, като стиловете включват оформяне на формуляри, бутони, таблици и други. JavaScript е добавен за интерактивност на управление на събития и динамично обновяване на съдържанието без презареждане на страницата.

2. Слой на бизнес логиката:

Бизнес логиката съдържа обработка на данни и изпълнение на процеси. Той включва PHP файлове за обработка на входящите заявки от потребителите, взаимодействие с базата данни, връщане на резултат към презентационния слой. Към това се отнасят обработка на регистрация и вход, резервация и отмяна на прегледи, добавяне на бележки, оценки и ревюта и други. PHP също осигурява валидации за сигурност на потребителските данни преди да се изпратят към базата данни.

3. Слой на базата данни:

Слоят на базата данни е отговорен за съхранение и управление на данните. Това включва файл с MySQL база данни, съдържаща всички таблици, необходими за приложението; MySQL импорти на данни за таблиците; PHP файл с конфигурационни настройки и функции за свързване с базата данни, това включва данни като хост, потребителско име, парола и име на базата данни, както и логика за установяване на връзка с базата данни.

Таблици на базата данни:

1. patients – съхранява информацията за потребителите, имаща връзка едно към много към appointments, който те са резервирали.
2. doctors – съхранява информацията за лекарите, имаща връзка едно към много към appointments, който те предлагат.
3. appointments – съхранява информацията за прегледите.

Декомпозиция на приложението:

1. Модул за управление на потребители:

- a. Регистрация на потребители.
- b. Вход и изход на потребители.

2. Модул за управление на часове:

- a. Добавяне на свободни часове от лекари.
- b. Резервация на часове от пациенти.
- c. Отмяна на часове от пациенти и от лекари.
- d. Преглед на налични часове.

3. Модул за ревюта, оценки и бележки

- a. Добавяне на ревюта и оценки от пациенти след преглед.
- b. Добавяне на бележки от лекар за преглед.
- c. Преглед на ревюта и оценки от лекарите за техните прегледи.
- d. Преглед на ревюта и оценки от пациентите за даден лекар.
- e. Преглед на бележки от пациенти за техните прегледи.

4. Модул за управление на базата данни:

- a. Създаване.
- b. Четене.
- c. Актуализиране.

4. Използвани технологии

За реализирането на приложението за използвани следните технологии:

1. PHP 8.2.12 – Използвани са стандартните възможности на езика, без да са включени допълнителни библиотеки. Чрез `php` се моделира бизнес логиката на нашето приложение.
2. MySQL – базата данни, в която съхраняваме информацията за потребители, лекари, резервирани часове.
3. HTML – Служи за имплементирането на презентационния слой на приложението.
4. CSS – Служи за създаването на привлекателен интерфейс на приложението
5. JS – Служи за вкарването на динамични елементи в нашите страници.

5. Инсталация, настройки и DevOps

За стартиране на приложението е необходимо първоначално да инсталирате PHP и MySQL. За удобство можете да инсталирате директно XAMP – <https://www.apachefriends.org/>.

Той предоставя директно PHP и MySQL, без да се налага да се инсталират и конфигурират поотделно.

След инсталирането на XAMP е нужно да стартираме Apache web server и MySQL. Портовете по подразбиране биха ни свършили работа, но в случай, че на някой от тях има вече стартиран друг ресурс, трябва да ги променим от конфигурационните файлове.



Фиг: Стартиране на Apache и MySQL

Следващата стъпка е да настроите връзката към базата данни през приложението:

- Отворете файла **DB.php**, който се намира в директорията **data** и въведете вашите данни, за да се осъществи връзка с базата:
 - \$dbhost** – трябва да е **localhost**, но ако сте с различна инсталация може да е различно.
 - \$dbName** – името на базата е **medappoint**, но може да се промени.
 - \$port** – ако използвате порта по подразбиране - 3306, не е нужно да се оказва. При различен порт се поставя тук.
 - \$userName** – потребителското име на потребител, който има достъп до базата данни (по подразбиране е **root**).
 - \$userPassword** – паролата на потребителя, който има достъп до базата данни (по подразбиране е празна).
- Трябва да създадете база данни в MySQL, чието име да стойността на променливата: **\$dbName**
- След това трябва да изпълните следните два скрипта, които можете да намерите в папката **data**:
 - createDbStructure.sql** – създава таблиците.
 - initDatabase.sql** – попълва примерни данните в таблиците.

За стартиране на приложението, трябва да поставите приложението в папката **xamp/htdocs**, която се намира там, където сте оказали при инсталиране. След това в удобен за Вас браузър (напр. Chrome) въведете следния адрес: <http://localhost/WebProject/index.php>, където **WebProject** е директорията, в която се намира проекта.

6. Кратко ръководство на потребителя

1. Начална страница:

- a. Представят всички налични доктори в системата.
- b. Потребителите могат да филтрират докторите по име, специалност и регион.
- c. Потребителите да видят информация за рейтинга на доктора, който се изчислява като средно аритметично на всички негови оценки.
- d. Потребителите могат да прегледат профила на всеки доктор.

Регистрация за пациент

Вход за пациент

Регистрация за доктор

Вход за доктор

Търсене

Специалност

Регион

Търсене

Резултати от търсенето:

Доктор	Специалност	Регион	Рейтинг	
Петър Петров	кардиолог	София	★★★★★	<div>Виж профил</div>
Никола Стоянов	УНГ	Пловдив	★★★★☆	<div>Виж профил</div>
Димитър Димитров	невролог	Бургас	★★★★☆	<div>Виж профил</div>

2. Преглед на профил на доктор:

- a. Показва се основната информация на доктора.
- b. Показват се всички отзиви и оценки от други пациенти, вече посетили дадения специалист.
- c. Показват се всички свободни часове на доктора.
- d. Пациентът може да запази съответния час, в случай че е регистриран и е влязъл в профила си. При опит за запазване на час от Гост потребител, се извежда съобщение за неуспех.

Назад

Информация за доктора

Име: Петър Петров

Специалност: кардиолог

Работен адрес: София, жк Манастирски Ливади

Регион: София

Рейтинг:★★★★★

Отзиви:

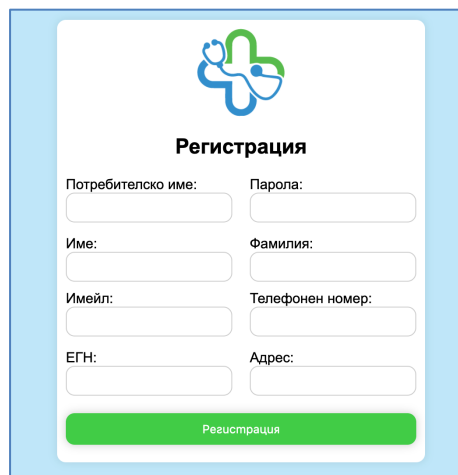
Пациент	Ревю	Оценка
Николай Костандиев	Много добро отношение. Препоръчвам!	★★★★★

Свободни часове:

Доктор	Локация	Дата и час	
Петър Петров	кабинет 666	2024-06-17 09:00:00	<div>Запази час</div>
Петър Петров	кабинет 666	2024-06-17 08:00:00	<div>Запази час</div>
Петър Петров	кабинет 123	2024-06-15 09:30:00	<div>Запази час</div>

3. Регистрация на пациент:

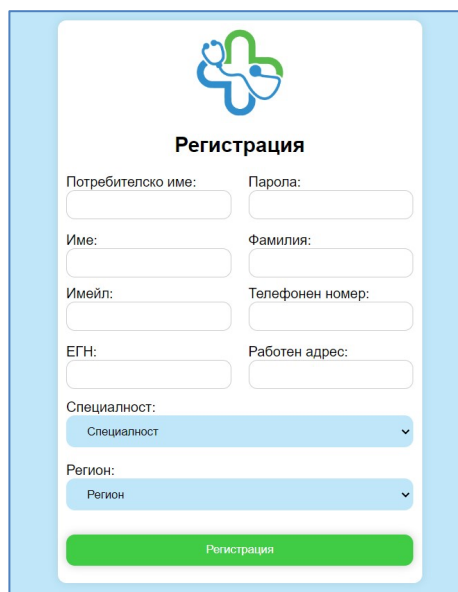
- a. Пациентът се регистрира в системата, въвеждайки следните данни: Потребителско име; Парола; Име; Фамилия; Имейл; Телефон; Адрес; ЕГН.
- b. При въвеждане на невалидни данни по време на регистрацията се извеждат съобщения за грешка за невалидните полета.



The registration form for a patient is titled "Регистрация" and features a medical logo at the top. It contains input fields for: Потребителско име (Username), Парола (Password), Име (First Name), Фамилия (Last Name), Имейл (Email), Телефонен номер (Phone Number), ЕГН (VAT Number), and Адрес (Address). A green "Регистрация" button is at the bottom.

4. Регистрация на доктор:

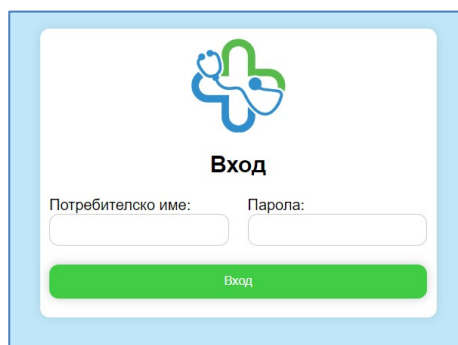
- a. Докторът се регистрира в системата, въвеждайки следната информация: Потребителско име; Парола; Име; Фамилия; Имейл; Телефон; ЕГН; Работен адрес; Специалност; Регион.
- b. При въвеждане на невалидни данни по време на регистрацията се извежда съобщение с грешка за невалидното поле.



The registration form for a doctor is titled "Регистрация" and features a medical logo at the top. It contains input fields for: Потребителско име (Username), Парола (Password), Име (First Name), Фамилия (Last Name), Имейл (Email), Телефонен номер (Phone Number), ЕГН (VAT Number), Работен адрес (Work Address), Специалност (Specialty) with a dropdown menu, and Регион (Region) with a dropdown menu. A green "Регистрация" button is at the bottom.

5. Вход в системата:

- a. Пациентите и докторите влизат в системата въвеждайки следната информация: Потребителско име; Парола.
- b. При въвеждане на невалидни данни по време на вход се извежда съобщение с грешка за невалидното поле.




The login form is titled "Вход" and features a medical logo at the top. It contains input fields for: Потребителско име (Username) and Парола (Password). A green "Вход" button is at the bottom.

6. Начална страница на пациент, след вход в системата:

- a. Представят всички налични доктори в системата.
- b. Потребителите могат да филтрират докторите по име, специалност и регион.
- c. Потребителите да видят информация за рейтинга на доктора, който се изчислява като средно аритметично на всички негови оценки.
- d. Потребителите могат да прегледат профила на всеки доктор.
- e. Потребителят има възможност да прегледа историята на всички свои прегледи.

- f. Потребителят може да прегледа бележки, които са оставени от специалиста за съответния преглед.
- g. Потребителят може да остави оценка и/или ревю на вече посетен преглед. Задължително условие е прегледа да е преминал.
- h. Потребителят може да отменя негов преглед, който все още не е минал.



nkostandiev
Николай Костандиев

Излизане

Търсене

Специалност

Регион

Търсене

Резултати от търсенето:


Доктор	Специалност	Регион	Рейтинг	
Петър Петров	кардиолог	София	★★★★★	Запази час
Никола Стоянов	УНГ	Пловдив	★★★★☆	Запази час
Димитър Димитров	невролог	Бургас	★★★★☆	Запази час

Резервирани часове:

Доктор	Локация	Ден и час	Бележки	Ревю	Оценка
Петър Петров	кабинет 123	2024-12-06 10:00:00	Моля, донесете направените медицински снимки!	Много добро отношение. Препоръчвам!	★★★★★
Никола Стоянов	кабинет 123	2024-10-06 11:30:00	Направете предварителни изследвания на кръв и урина	Докторът е голям специалист. Без забележки.	★★★★☆
Петър Петров	кабинет 123	2024-06-25 08:00:00			Откажи час
Петър Петров	кабинет 235	2024-03-03 10:00:00	Моля, изпратете информация от предишни лечения.		Запази

7. Начална страница на доктор, след вход в системата:

- a. Докторът има възможност да преглежда всички свои резервирани часове.
- b. Докторът има възможност да добавя бележки към вече резервирани часове, които пациентът да получи и прегледа.
- c. Докторът има възможност да преглежда оставени ревюта и оценки от пациентите.
- d. Докторът може да отменя резервиран преглед, който все още не е минал.
- e. Докторът има възможност да добави нов час за преглед. Не се допуска дублиране на часове.
- f. Докторът има възможност да прегледа всички свои свободни часове.
- g. Докторът може да отменя преглед, който все още не е резервиран.



resho
Петър Петров

Излизане

Резервирани часове:

Пациент	Локация	Ден и час	Бележки	Ревю	Рейтинг
Николай Костандиев	кабинет 123	2024-12-06 10:00:00	Моля, донесете направените медицински снимки!	Много добро отношение. Препоръчвам!	★★★★★
Николай Костандиев	кабинет 123	2024-06-25 08:00:00			Откажи час

Добави час за преглед

Дата:

mm / dd / yyyy

Час:

-- : --

Локация:

Добави часа

Свободни часове:

Локация	Ден и час	
кабинет 666	2024-06-27 09:00:00	Откажи час
кабинет 666	2024-06-27 08:00:00	Откажи час
кабинет 123	2024-06-25 09:30:00	Откажи час

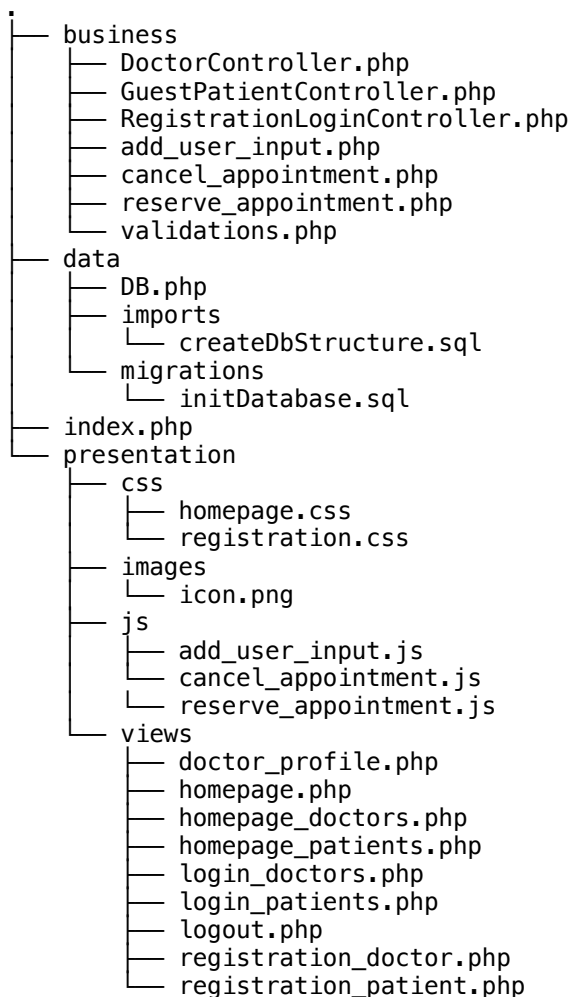
8. Примерни данни

За да вкарате примерни данни в таблиците на базата, изпълнете SQL скрипта, който се намира в `data/migrations/initDatabase.sql`

```
1  -- Add sample data for patients
2
3  INSERT INTO `patients` (`username`, `password`, `email`, `first_name`, `last_name`, `egn`, `address`, `phone_number`) VALUES
4  ("nkostandie", "6038a086464502c0b30a2fe89490348dee2ef020", "nkostandie@uni-sofia.bg", "Николай", "Костандиев", "1234567890", "София, жк Студентски град
5  ("ivn-maria", "561acdc636e0e0176a960dd4162192c9939bbe0c", "ivn.maria@gmail.com", "Мария", "Иванова", "0212345670", "София, жк Лозенец", '0878123456');
6
7  -- Add sample data for doctors
8  INSERT INTO `doctors` (`username`, `password`, `email`, `first_name`, `last_name`, `egn`, `work_address`, `region`, `phone_number`, `speciality`) VALUE
9  ('pesho', '6ed95172cef6020ab37165a7dce8f505c8659683', 'petar_petrov@gmail.com', 'Петър', 'Петров', '1234512345', 'София, жк Манастирски Ливади', 'София
10 ('niki', '6038a086464502c0b30a2fe89490348dee2ef020', 'nikola_stoqnow@gmail.com', 'Никола', 'Стоянов', '1234512345', 'Пловдив, жк Капана', 'Пловдив', '0
11 ('mitko', '1e9d42b13721ae13022daaad335e40e13eb1c639', 'mitko_mitkov@abv.bg', 'Димитър', 'Димитров', '1234512345', 'Бургас, жк Меден Рудник', 'Бургас',
12
13 -- Add sample data for appointments
14 INSERT INTO `appointments` (`doctor_id`, `patient_id`, `review`, `rating`, `notes`, `location`, `appointment_date`) VALUES
15 ( 1, 1, 'Много добро отношение. Препоръчвам!', 5, 'Моля, донесете направените медицински снимки!', 'кабинет 123', '2024-12-06 10:00:00'),
16 ( 2, 1, 'Докторът е голям специалист, Без забележки.', 4, 'Направете предварителни изследвания на кръв и урина', 'кабинет 123', '2024-10-06 11:30:00'),
17 ( 3, 2, 'Доста изчерпателен и подробен преглед!', 4, 'Донесете информация за съпътстващи заболявания', 'кабинет 235', '2024-10-06 14:00:00'),
18 ( 1, 1, NULL, NULL, 'Моля, изпратете информация от предишни лечения.', 'кабинет 235', '2024-03-03 10:00:00'),
19 ( 1, 1, NULL, NULL, NULL, 'кабинет 123', '2024-06-15 08:00:00'),
20 ( 1, NULL, NULL, NULL, NULL, 'кабинет 123', '2024-06-15 09:30:00'),
21 ( 1, NULL, NULL, NULL, NULL, 'кабинет 666', '2024-06-17 08:00:00'),
22 ( 1, NULL, NULL, NULL, NULL, 'кабинет 666', '2024-06-17 09:00:00');
23 |
```

9. Описание на програмния код

Структура на кода:



1. business – php файлове, отнасящи се към слоя на бизнес логиката на системата:

- a. DoctorController.php – php файл с функции за създаване, четене и актуализиране на записи от базата данни, свързани с функционалности, които се изпълняват от доктор.
- b. GuestPatientController.php – php файл с функции за създаване, четене и актуализиране на записи от базата данни, свързани с функционалности, които се изпълняват от гост потребител или пациент.
- c. RegistrationLogin.php – php файл с функции за създаване на записи от базата данни, свързани с функционалности, които се изпълняват при регистрация и вход в системата.
- d. add_user_input.php – php файл с функция за актуализиране на записи от базата данни при добавяне на информация от потребителя: бележка, оценка и отзив.
- e. cancel_appointment.php – php файл с функции за триене или актуализиране на записи от базата данни при опит за премахване на преглед.

- f. `reserve_appointment.php` – php файл с функции за актуализиране на записи от базата данни при опит за резервация на преглед.
 - g. `validations.php` – php файл с функции за валидиране на данните при създаване, четене и актуализиране на записи от базата данни.
2. `data` – файлове, отнасящи се към слоя на базата данни на системата:
- a. `DB.php` – php файл, осъществяващ връзката с базата данни.
 - b. `imports` – sql скриптове за създаване и инициализиране на базата данни:
 - `createDbStructure.sql` – sql скрипт за създаване на всички таблици, нужни на системата.
 - c. `migrations` – sql скриптове за импорт на данни в базата:
 - `initDatabase.sql` – sql скрипт с примерни данни за тестване на системата.
3. `index.php` – php файл за препращане към началната страница на системата при отваряне на проекта.
4. `presentation` – файлове, отнасящи се към презентационния слой на системата:
- a. `css` – css файлове за стайлинг и оформление:
 - `homepage.css` – css файл за началните страници и за страницата на лекар.
 - `registration.css` – css файл за страниците за регистрация и вход.
 - b. `images` – снимки, използвани в системата:
 - `icon.png` – png снимка на логото.
 - c. `js` – javascript файлове за динамично поведение на функционалностите:
 - `add_user_input.js` – javascript файл с функция за динамично поведение при добавяне на информация от потребителя: бележка, оценка и отзив.
 - `cancel_appointment.js` – javascript файл с функция за динамично поведение при опит за премахване на преглед.
 - `reserve_appointment.js` – javascript файл с функция за динамично поведение при опит за резервация на преглед.
 - d. `views` – php файлове с файлове с html и php код
 - `doctor_profile.php` – php файл за страницата на лекар с различен хедър, спрямо това дали потребителя е в профила си или не (бутони за регистрация и вход, ако не е в профил; бутон за връщане назад, за излизане, и информация за потребителя, ако е логнат); обща информация за лекаря; ревюта и оценки за лекаря, оставени от пациенти; таблица със свободните часове на лекаря с бутон за запазване на час, с различно поведение при потребител, който е влязъл в профила си, и такъв, който не е.

- homepage.php – php файл с началната страница на системата с бутони за регистрация и вход в хедъра; таблица с лекарите с опция за търсене по име, филтриране по специалност и регион.
- homepage_doctors.php – php файл с началната страница при вход на лекар с бутон за излизане и информация за потребителя в хедъра; таблица, изобразяваща запазените от пациенти прегледи с опция за оставяне на бележка към преглед; таблица със свободните часове лекаря; форма за създаване на нов свободен час за преглед, предоставяща полета за попълване на нужната информация, с проверки и изобразяване на съобщения за грешки.
- homepage_patients.php – php файл, представляващ началната страница при вход на пациент с бутон за излизане и информация за потребителя в хедъра; таблица с лекарите с опция за търсене по име, филтриране по специалност и регион; таблица, изобразяваща запазените от пациента прегледи, с опция за оставяне на отзив и бележка за отминал преглед.
- login_doctors.php – php файл с форма за вход на лекари, предоставящи полета за попълване на нужната информация, с проверки и изобразяване на съобщения за грешки.
- login_patients.php – php файл с форма за вход на пациенти, предоставящи полета за попълване на нужната информация, с проверки и изобразяване на съобщения за грешки.
- logout.php – php файл за унищожаване на сесията и препращане към началната страница на системата.
- registration_doctor.php – php файл с форма за регистрация на пациенти, предоставящи полета за попълване на нужната информация, с проверки и изобразяване на съобщения за грешки.
- registration_patient.php – php файл форми за регистрация на пациенти, предоставящи полета за попълване на нужната информация, с проверки и изобразяване на съобщения за грешки.

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

Нашето приложение не може да бъде разделено на 2 отделни спрямо извършената работа, тъй като около 90% от работата беше извършена заедно по време на онлайн разговори, където активно обменяхме идеи и съвместно решавахме възникналите проблеми. Въпреки това има някои неща, върху който всеки от нас се съсредоточи извън общата ни работа:

1. Мария Иванова: фокус върху създаването на интуитивен интерфейс, стил и разположение на компонентите, което направи приложението лесно за употреба.
2. Николай Костандиев: задълбочаване в JavaScript за създаване на функционалности за динамично запазване на резервациите, бележките, отзивите и оценките.

Ограниченията включват липсата на мобилна версия на приложението и нуждата от допълнителни тестове за сигурност (напр. удостоверяване на самоличността на лекарите).

Възможностите за бъдещо разширение включват интеграция на мобилни приложения, интеграция със система за плащане, добавяне на функционалности качване на документи, за известия и customer support.

11. Какво научих

1. Мария Иванова:
По време на курса и разработването на проекта придобих умения по HTML и CSS, чрез който се научих как да създавам интуитивни и стилни потребителски интерфейси, които улесняват потребителите, и се запознах с основните принципи на UX.
Също така, научих основите на PHP за изграждане на backend, разбрах как да свързва frontend структурата с backend, използвайки PHP.
Използвах предишните си знания по MySQL за ефективно управление на бази данни и потребителските данни, и връзката им с PHP.
Научих и основите на JavaScript за добавяне на интерактивност и динамичност към потребителския интерфейс на регистрационната система.
Придобих умения за ефективно структуриране на проекти, за работа в екип и комуникация на идеи по време на онлайн срещи.
2. Николай Костандиев:
В процеса на учене и работа по проекта, научих как да използвам HTML и CSS за създаване на прости, но ефективни потребителски интерфейси.
Също така, разбрах важността на структурирането на цялостния проект и свързването на frontend и backend чрез JavaScript и PHP.
Усвоих принципите на работа с бази данни и интеграцията им с уеб интерфейси, чрез подобряване на досегашните ми знания по MySQL бази данни.
Друго нещо, което научих по време на курса, беше задълбочаването в JavaScript и използването на Fetch API за създаване на динамични уеб приложения.
Също така, придобих ценен опит в управлението на проекти и съвместната работа по време на онлайн срещи. Разбрах важността на добрата комуникация и координация в екипната работа.

12. Използвани източници

1. PHP Документация - <https://www.php.net/manual/en/>
2. MySQL Документация - <https://www.mysql.com/>
3. W3Schools PHP примери - <https://www.w3schools.com/php/>
4. Презентации от лекции и упражнения в модул - <https://learn.fmi.uni-sofia.bg/course/view.php?id=10001>

Предал (подпис):

/9MI0600045, Мария Иванова, СИ, група 1/

/6MI0600046, Николай Костандиев, СИ, група 3/

Приел (подпис):

/проф. д-р Милен Петров/