ФМИ проект Exam Browser

Към курс „WEB технологии“, зимен семестър

Документация

Димитър Николаев Колев ФН: 81998 Група 2, Курс IV

Мартин Йорданов Пенев ФН: 81990 Група 2, Курс IV

Специалност: Компютърни науки



1) Уводна част:

Проектът представлява реализирана система – уеб приложение за провеждане на електронни изпити, която поддържа информация за потребители (учители и ученици), електронни тестове с възможни отговори, електронно автоматизирано оценяване. Програмата съхранява информацията си в база данни като за Database Server е използван MariaDB, а за Web Server ***Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/8.2.0*.**

Основната цел на проекта е да осигурява възможност на потребителите да провеждат изпитни сесии под формата на тест. Чрез уеб приложението - учителите имат възможност да създават тестове, чрез прикачване на файлове във формат json, след което да проверят как са се справили техните ученици със съответния тест и да видят какъв е средният успех (по шестобалната система) на направените изпитни сесии до момента. Учениците от своя страна имат бърз и лесен достъп до тестовете, които техният учител е създал и след приключване на съответната изпитна сесия получават оценка на база процент верни отговори, които са дали на тестовите въпроси. След приключване на теста, ученикът или студентът може да провери кои въпроси са правилни и кои грешни по цвета на отговорите, които е дал (зелен за правилен и червен за грешен). Освен автоматизираната система за оценяване, приложението мониторира и броя на правилно отговорените въпроси. Проектът би бил приложим в множество учебни заведения като университети, училища и други предприятия, които имат потребност от провеждане на изпитване под формата на тест.

Структурата на документацията ще бъде следната:  
(1) Уводна част, в която се описва основната идея на проекта, за какво и по какъв начин биха го използвали хората. Съдържа също така кратко описание на основните функции и кратка информация за начина, по който приложението обработва своите данни и ги запазва в базата. (2) Следва преглед на предметната област, в който се разглеждат подходите и методите при реализацията на съответната система, както и концепция. (3) Трета част – кратка информация – какво се запазва в базата данни, по какъв начин и поради каква причина. (4) В четвъртата част са разгледани конкретни фрагменти от кода и подходи, използвани при създаването на уеб приложението при Front-end частта. (5) Следващата част съдържа фрагменти от код от Back-end частта на уеб проекта. (6) Кратка информация за страниците и CSS оформлението се разглеждат в 6та точка, след което в (7) са представени екранни заснемания на уеб приложението в действие, демонстрация на функционалността. (8) Направено е заключение, в което се коментира възможност за бъдещо развитие на проекта и по какъв начин той може да бъде оптимизиран.

2) Преглед на предметната област:

Основните проблеми, които трябва да бъдат решени при реализацията на приложението са: правилно подбиране на данните, осигуряване на комуникация между клиента и сървъра, осигуряване на коректна работа с базата данни, възможности за редакция, покриване на основните изисквания и взаимодействия . Проектът коректно се справя със следните функционалности:

* Създава акаунти с потребителско име, парола и роля (ученик или учител)
* Запазва съответните профили в базата данни MariaDB
* Възможност за директен лог при вече създаден акаунт в системата
* В случай, че акаунтът е учителски имаме следните възможности:
  + - Създаване на тестове, чрез прикачване на файл във формат json (тестовете автоматично се запазват в базата данни и могат да бъдат достъпени от учениците). – tab CREATE TESTS
    - Проверка на създадените тестове и възможност съответният тест да бъде премахнат. – tab MY PROFILE
    - Проверка на оценките на учениците, които са работили по съществуващите тестове – tab MY PROFILE
    - Възможност за приключване на сесията (излизане от текущия активен профил) – tab LOGOUT
* В случай, че акаунтът е ученически имаме следните възможности:
  + - Показване на възможните тестове, които могат да бъдат започнати (тест, който веднъж е направен, няма възможност да бъде направен отново) – tab SHOW TESTS
    - Показване на вече завършените тестове със съответната оценка и средноаритметична оценка на всички приключени изпитни сесии - tab MY PROFILE
    - Възможност за приключване на сесията (излизане от текущия активен профил) – tab LOGOUT
* При стартиран тест:
  + - Възможност за маркиране на 1 от 4 възможни отговора
    - Възможност за преминаване към следващ въпрос (ако има такъв)
    - Възможност за връщане към предишен въпрос (ако има такъв)
    - Възможност за връщане към основната страница (в случай, че тестът е завършен)
    - При завършване на теста – възможност за проверка на правилно отговорени и грешно отговорени въпроси
    - Автоматично генериране на оценка на база процент верни отговори

При реализацията на проекта е използвана добре познатата Client – Server технология, която представлява модел на разпределени изчисления, при който клиентът прави заявки към сървър за ресурси, а сървърът отговаря на тези заявки. В този модел клиентът и сървърът комуникират през мрежа, използвайки стандартизиран протокол.

Клиентът предоставя потребителски интерфейс или приложение, с което потребителят взаимодейства, докато сървърът осигурява основната логика и ресурси, необходими за изпълнение на заявките.

Този модел клиент-сървър позволява по-голяма мащабируемост, надеждност и сигурност, тъй като сървърът може да бъде оптимизиран за производителност и сигурност, докато клиентът може да се съсредоточи върху осигуряването на добро потребителско изживяване.

3) Database Server (MariaDB):

За запазване на данните, от които имаме нужда за коректна работа с уеб приложението е използван базата данни – MariaDB. Тя представлява релационна база данни, която е създадена като алтернатива на MySQL, разработена и поддържана от Монтy Програми AB, който е основател на MySQL. MariaDB използва същия SQL език като MySQL и поддържа много от същите функции и команди като MySQL.

Въпреки това, MariaDB включва някои нови функции, като например поддръжка на NoSQL бази данни и усъвършенствана скалируемост и производителност.

Базата съдържа две основни таблици от потребители и тестове.

Всеки потребител от първата таблица разполага със следните атрибути:

* Уникален идентификационен номер
* Първо име (изисква се при създаване на нов акаунт)
* Фамилия (изисква се при създаване на нов акаунт)
* Потребителско име (използва се при влизане в съответен акаунт)
* Криптирана парола (използва се при влизане в съответен акаунт)
* Роля – 1 или 2 (съответно ученик или учител)
* Списък от имена на завършени тестове със съответните оценки (отделени с -)

(валиден само за ученици) (тип text)

* Списък от създадени тестове (валиден само за учители) (тип text)

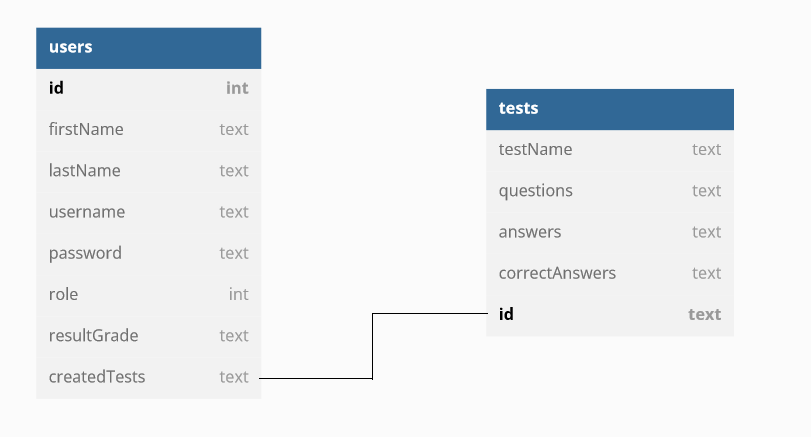
Картина, която съдържа текст

Описанието е генерирано автоматично

Всеки създаден тест във втората таблица разполага със следните атрибути:

* Име на теста
* Списък от въпроси разделени с уникален стрингов разделител (тип text)
* Списък от отговори, разделени с уникален стрингов разделител (тип text)
* Списък от правилни отговори, разделени с уникален стрингов разделител (тип text)
* Уникален идентификационен номер на теста



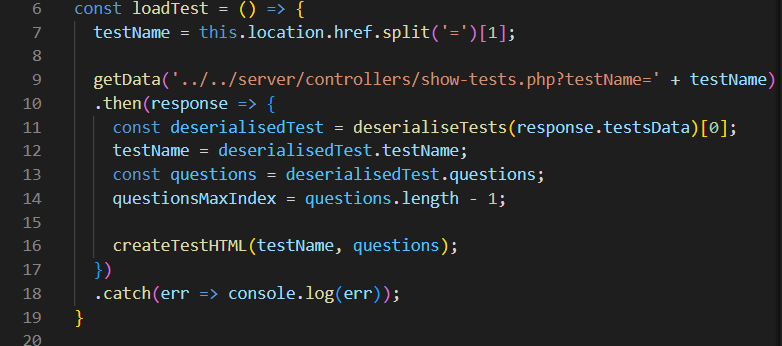


4) Front-end (Javascript):

JavaScript е мощен език за програмиране, който може да се използва за множество задачи във фронт-енд уеб разработката. Един от най-важните аспекти на JavaScript е, че той е интерпретируем език, което означава, че може да се изпълнява директно в браузъра на клиента, без да се изисква специален компилатор или други инструменти. Поради тази причина това е нашият избор за разработване на frond-end частта на приложението. JS скриптовете на уеб апликацията се намират в папка client/scripts в дървовидната структура на проекта.

Ето и някои фрагменти код, които реализират основните функционалности на приложението:

Създаване и зареждане на тестови изпит:  

На екранната снимка е представено по какъв начин се обновява страницата чрез добавяне към кода на html страницата.

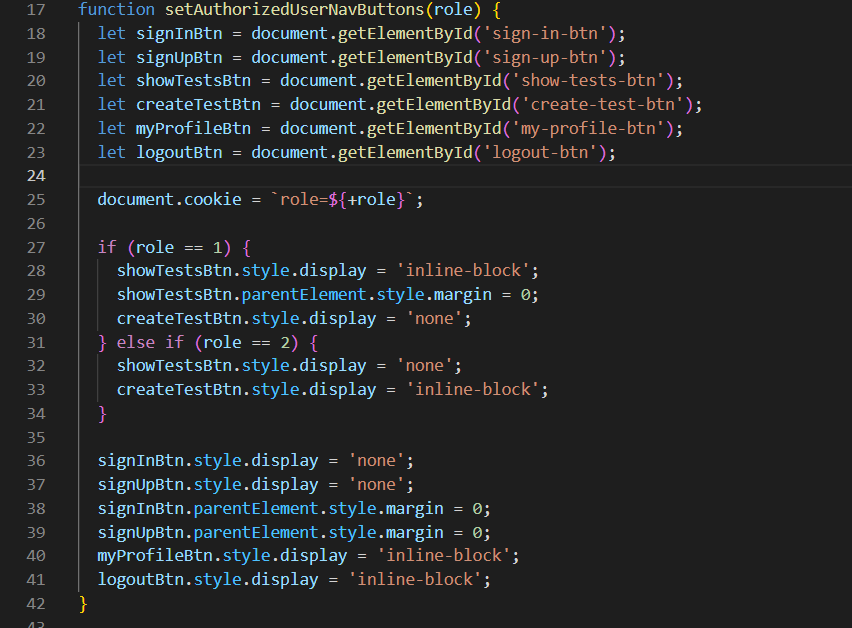
Изчисляване на оценка:

Продължение на функцията за изчисление и обновяване на оценката на съответния ученик:



Изпращане на POST заявка за зареждане на потребителска сесия:

Реализация на функцията, която настройва навигационните бутони при стартирана потребителска сесия:

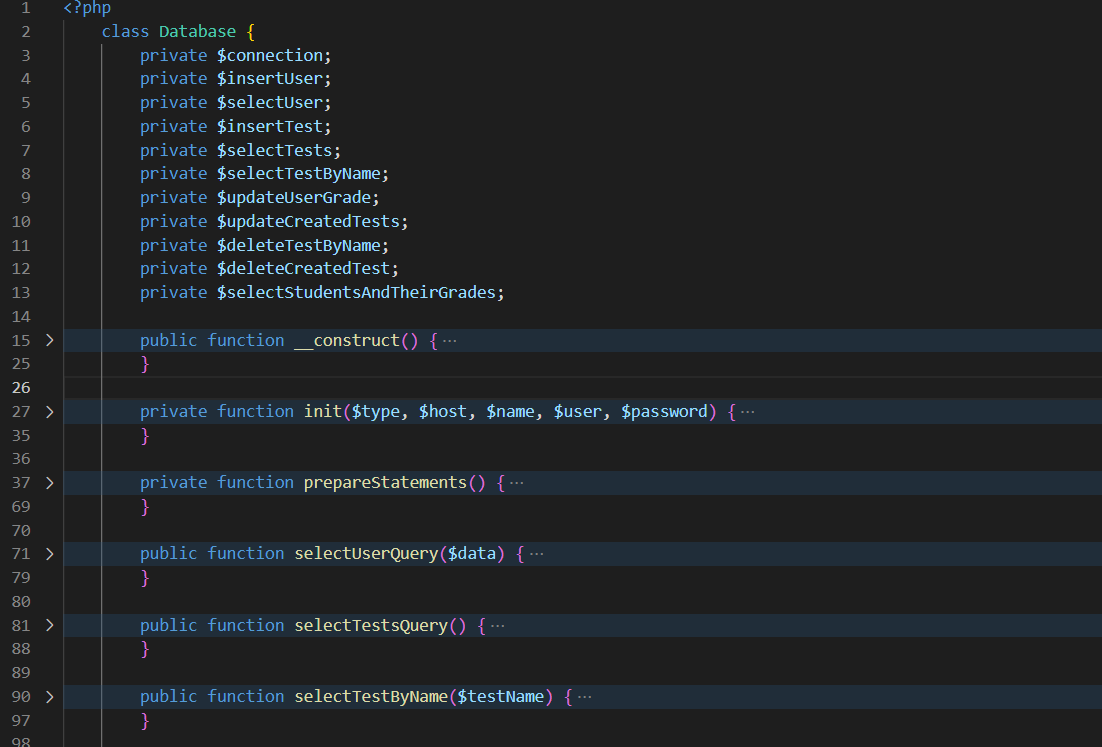


Вградените функции на javascript позволяват лесно настройване на навигацията според страницата, на която се намираме. Ролята (ученик или учител) на потребителя се запазва в COOKIE, откъдето лесно може да бъде достъпена при нужда.

5) Back-end (PHP):

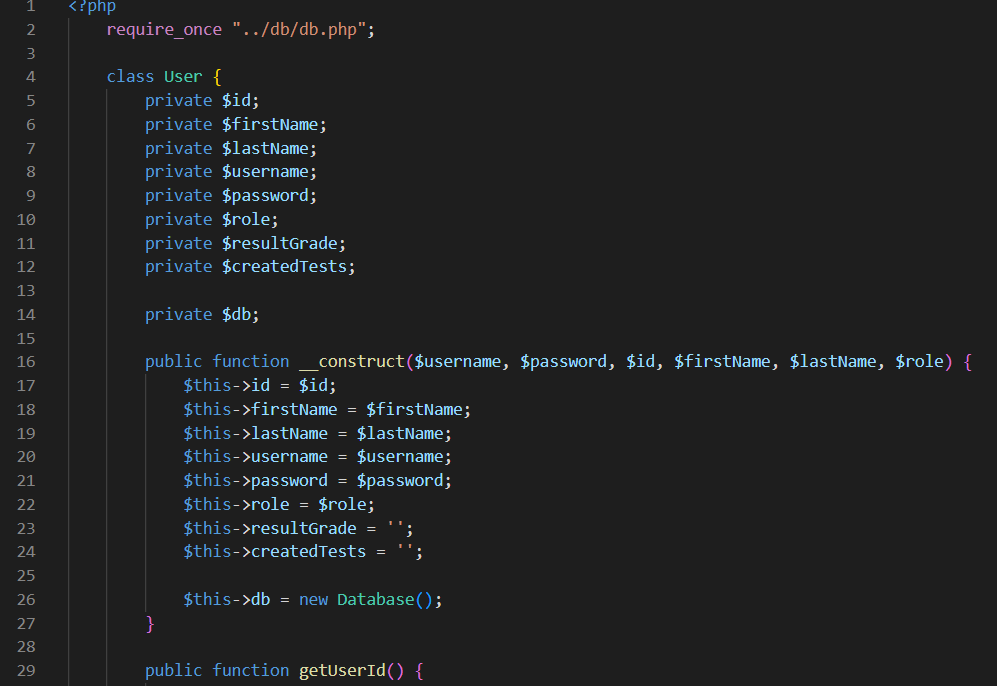
Логиката на уеб проекта е написана на PHP 8.2 и се намира в папка server в дървовидната структура на проекта. Тя от своя страна съдържа други подпапки: controllers, db, models и utils. Db -> основна папка съдържаща имплементацията, която осигурява връзката с базата данни. Там се намира основният клас -> Database. Той разполага с член данни и методи, които позволяват обновлението на информацията за потребителите и тестовете в базата.

Картина, която съдържа текст

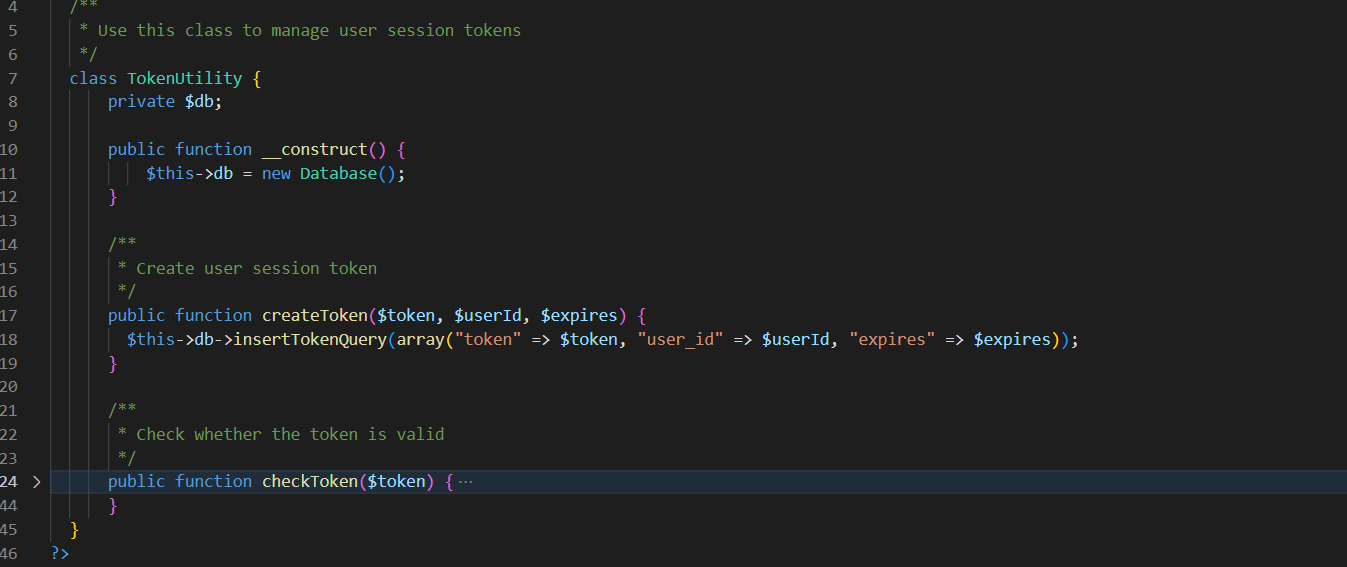
Описанието е генерирано автоматичноClass Database:Останалите класове използвани в проекта са следните:   
Class Тест:

Реализира член данни и методи нужни за поддръжката и коректната работа с тестовете.

Class User:



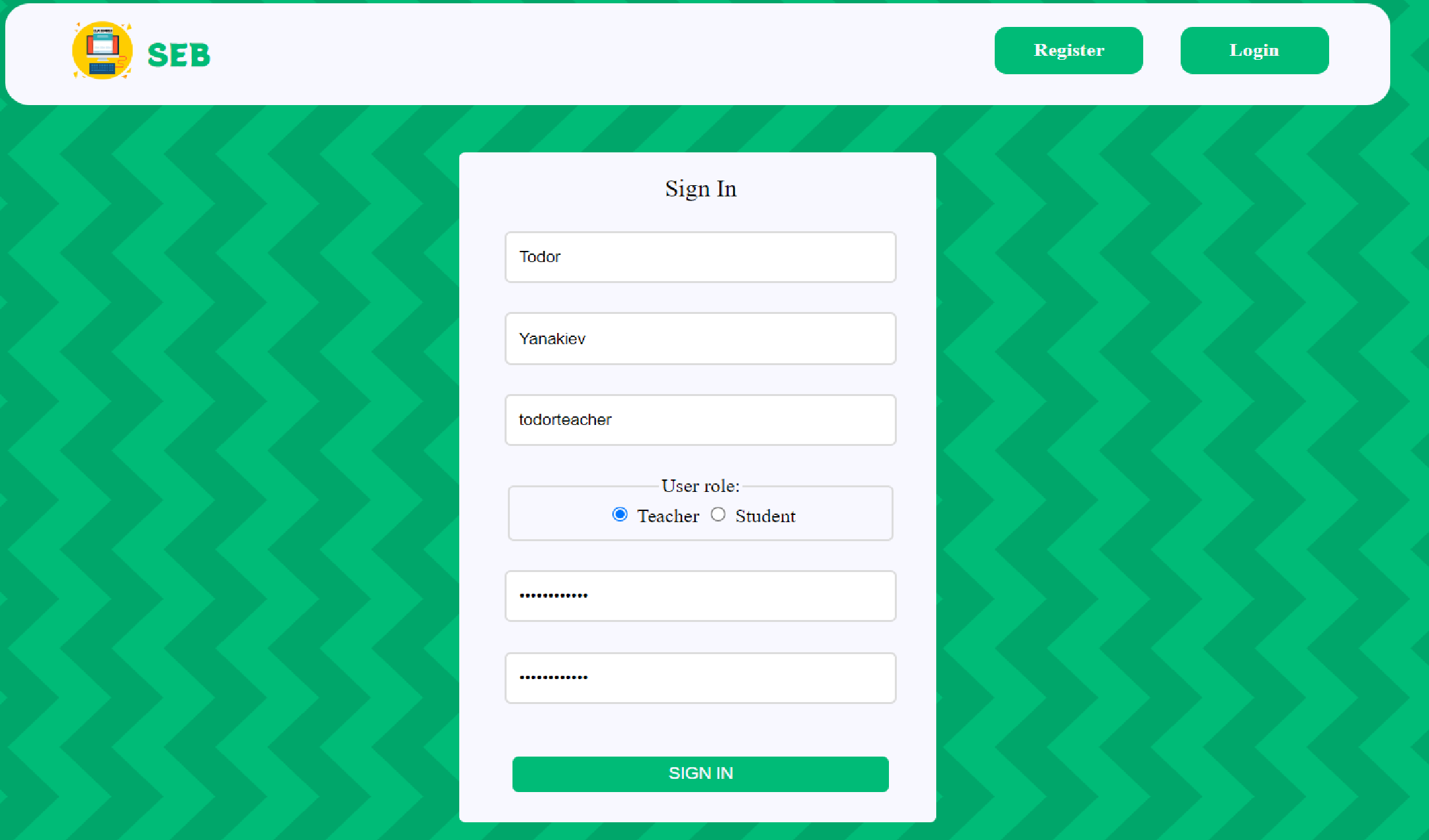
Реализира член данни и методи нужни за поддръжката и коректната работа с потребителите в системата.

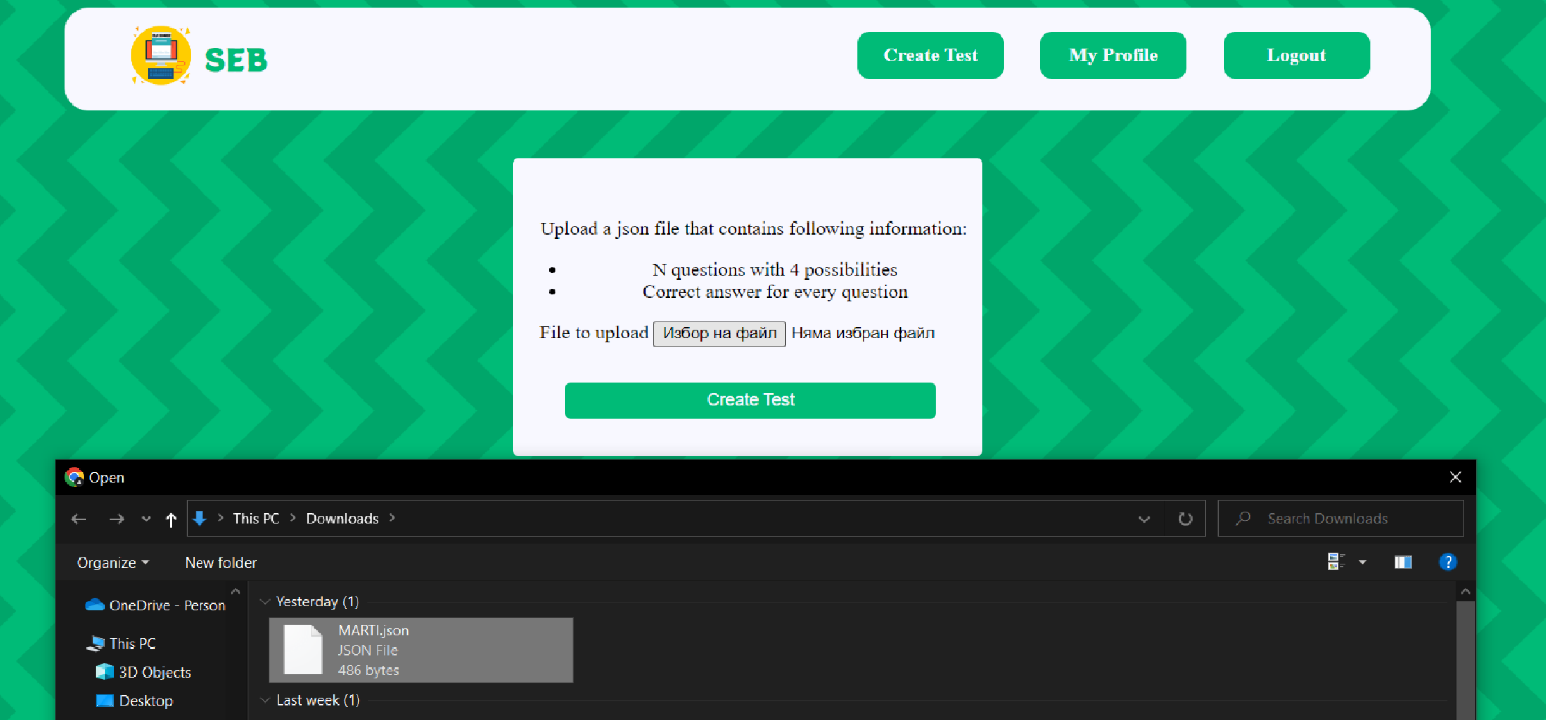
Class TokenUtility:

Реализира член данни и методи нужни за поддръжката и коректната работа с потребителски токени, които стартират сесия.

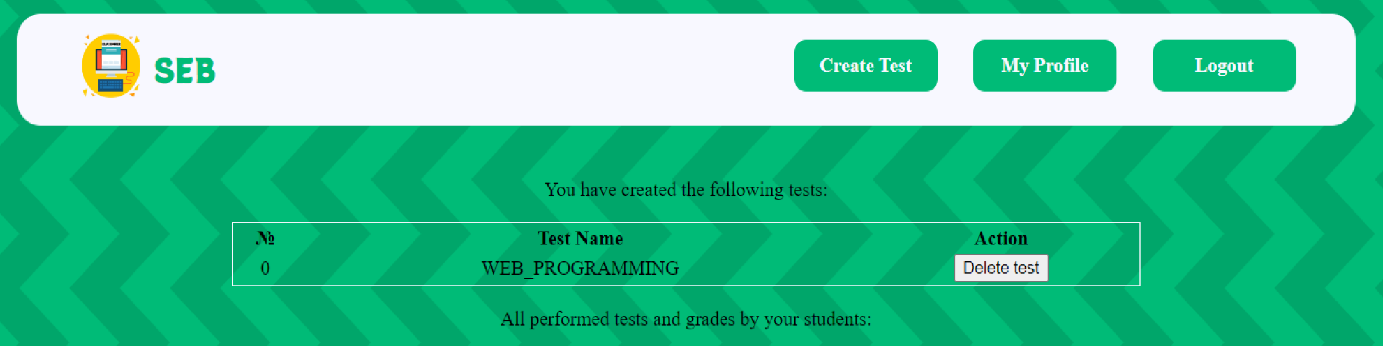
6) Реализация, тестване:

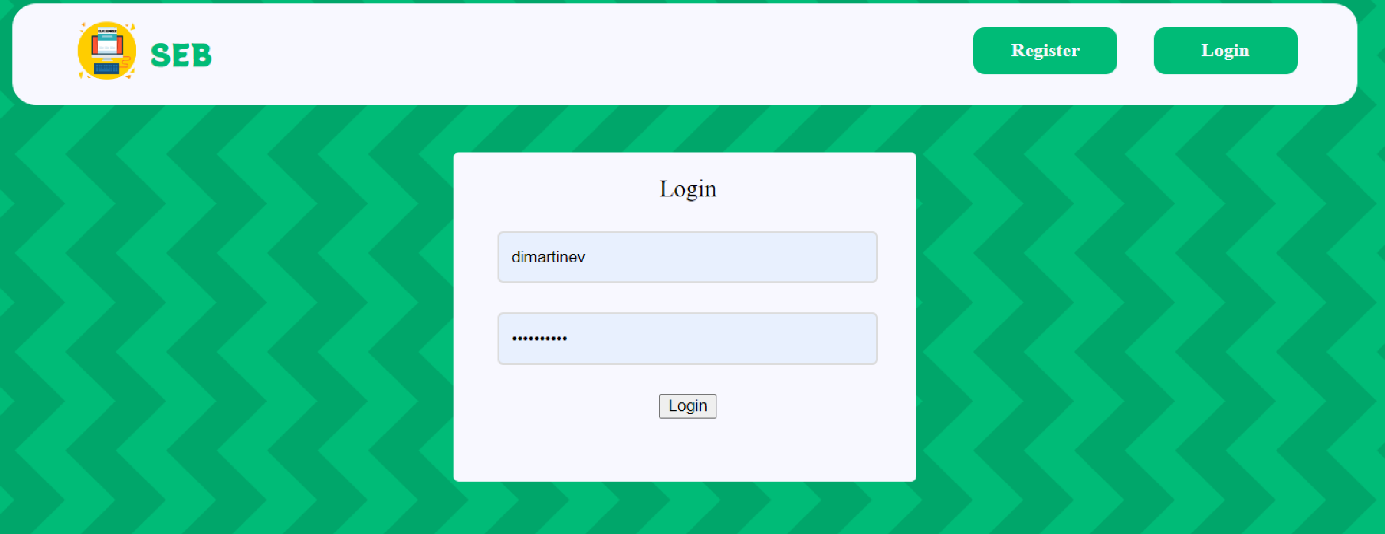
Примерен тестов сценарии за уеб приложението в илюстрирани стъпки:

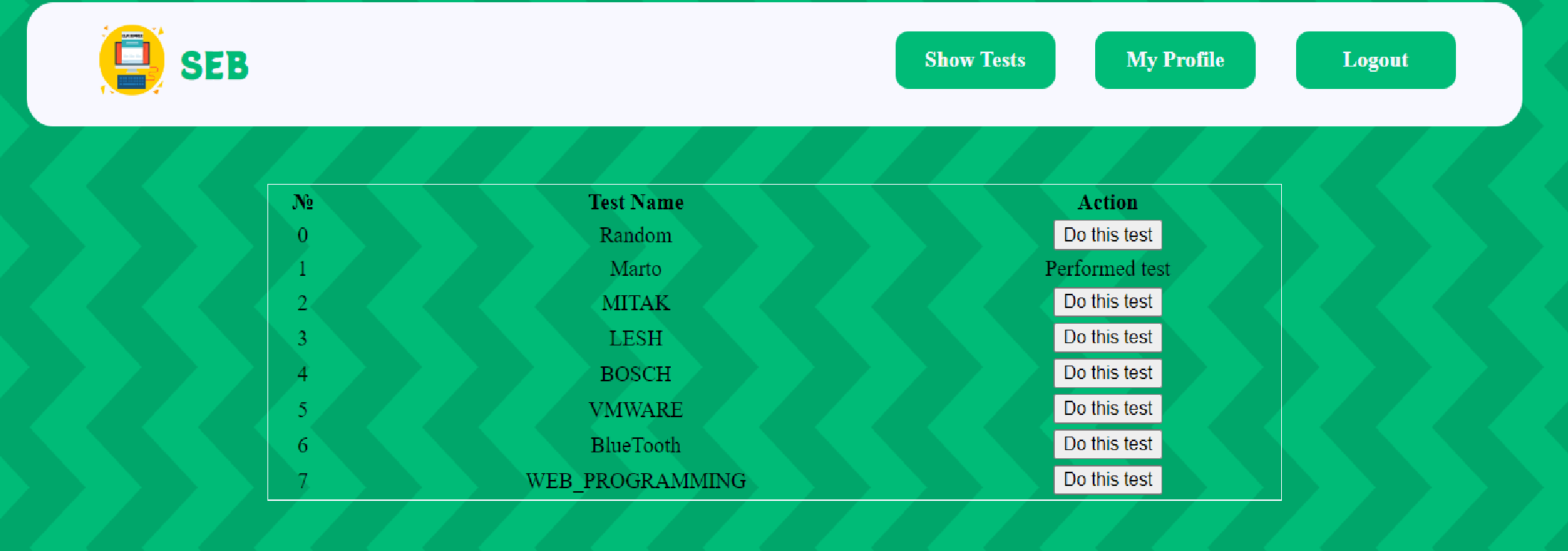
1) Създаване на нов потребителски профил на учител:

2) Автоматично логване с новия учителски акаунт и създаване на нов тест, чрез прикачване на предварително подготвен json файл с въпроси, възможни отговори и правилни отговори на съответните въпроси:

3) Проверка на създадения тест -> мониториране в страница MY PROFILE:

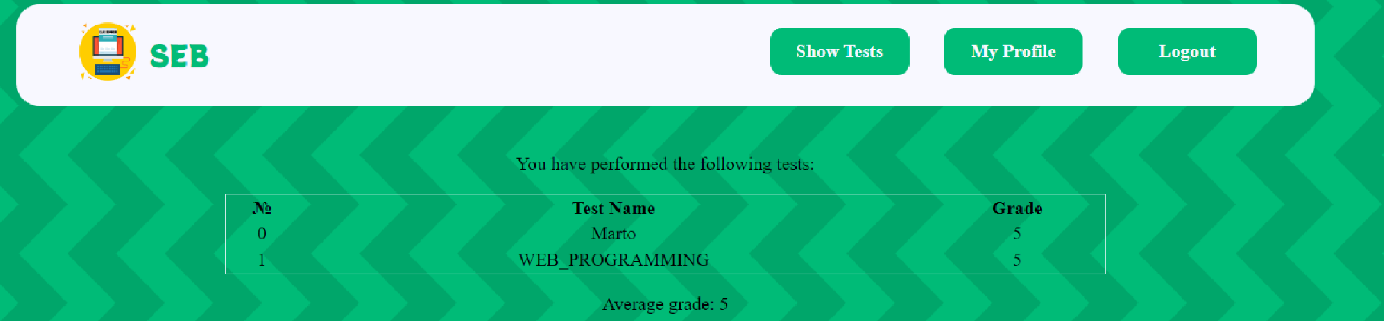


4) Излизане от учителски профил и логване в съществуващ ученически профил:  


5) Проверка на възможните изпитни сесии, които могат да бъдат стартирани:  


6) Стартиране на изпитна сесия:  


7) Приключване на изпитна сесия и проверка на оценка:

8) Връщане назад към основна страница и проверка на MY PROFILE и средноаритемтична оценка на направените тестове:

9)Излизане от ученическия профил:



8) Заключение:

При разработката на това приложение е използвана клиент-сървър методологията за създаване, като тя предоставя ефективен начин за обработка на заявките от клиентите и осигурява по-голяма сигурност на данните. PHP, JavaScript и MariaDB са избрани технологии, които предоставят мощни функционалности за изграждане на мащабируеми и персонализирани уеб приложения.   
 Като възможност за развитие на този проект може да се добавят функционалности за провеждане на изпитни сесии със свободни отговори, които да се проверяват от учителите, също така възможност за стартиране на изпит в конфигуриран прозорец, който ограничава достъпа на учениците до странични материали, приложения и сайтове.