**Имена:**  Румен Тошев **фн:**  *2MI0600038*

Цветан Тошев **фн:**  *0MI0600039*

Дейвид Барух **фн:**  *0MI0600128***Начална година:** *2024* **Програма:** бакалавър, (СИ) **Курс: 3**   
**Тема:** *Редактор - текстов/визуален на JSON файл за създаване на ескейп стаи*

**Дата: 12.6.2024г. Предмет:** *Web технологии*

**имейл:** [**rumen.tihomir.toshev@gmail.com**](mailto:rumen.tihomir.toshev@gmail.com)**,** [**cvetan.toshev@gmail.com**](mailto:cvetan.toshev@gmail.com)**, davidbaruchreal@gmail.com**

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

# ТЕМА: 50.1 Редактор - текстов/визуален на JSON файл за създаване на ескейп стаи

## 1. Условие

Да се създаде система, която позволява създаването/редактирането на JSON файлове, които се използват за ескейп стаи, като те съдържат ключалки от различен тип и самият ключ от съответния тип. Също така, съдържат и допълнителна информация като подсказки и полезни линкове и ресурси свързани с въпросите и ключалките.

## 2. Въведение – извличане на изисквания

**Роли**: регистриран потребител, който има достъп до функционалностите.

## Функционални и нефункционални изисквания:

1. Системата да може да зарежда JSON файлове, съдържащи ескейп стаи.
2. Системата да може да зарежда за преглед JSON файлове, съдържащи ескейп стаи.
3. Системата да може да редактира JSON файлове.
4. Системата да поддържа регистрация на потребители.
5. Всеки потребител да може да преглежда своите ескейп стаи.
6. Всеки потребител да може да редактира своите ескейп стаи.
7. Всеки потребител да може да изтегли дадена стая, която е в неговия профил.
8. Всеки потребител да може да създава нови ескейп стаи.
9. При създаването на ескейп стая, да може да се въвежда броя на ключалките, които той ще съдържа.
10. При създаването на ескейп стая, да може да се въвежда броя ресурсите за съответна ключалка.
11. Потребителя да може да вкарва в системата вече готови ескейп стаи.
12. Потребителя да може да променя името на негова ескейп стая.
13. Потребителя да може да се вписва в системата чрез имейл и парола.
14. Системата да запазва ескейп стаите на потребителя в базата данни.
15. Системата да зарежда ескейп стаите от базата данни, когато потребителя поиска съответната стая.
16. Потребителя да може да излезе (log out-не) от системата.
17. Системата да поддържа панел за навигация в приложението.
18. Панелът за навигация да е винаги видим и достъпен за потребителя.
19. При редакция на JSON файл, да се показват вече въведените стойности в него.
20. Интерфейса на приложението да бъде интуитивен.
21. Интерфейса на приложението да бъде опростен и лесен за разбиране.

**Ползи от реализацията:** Потребителите, използващи системата, ще могат да редактират и създават свои ескейп стаи. Системата ще помогне на потребителя, лесно и прецизно да създаде своята ескейп стая, като след това може да я запази и впоследствие достъпи в своя профил. Потребителя ще има пълен достъп до неговите ескейп стаи, като ще може да ги преглежда, редактира и изтегли като JSON файлове, за да може да ги използва наготово. След реализация на системата, потребителите, които я използват, ще могат ефективно и бързо да достъпват и обработват своите ескейп стаи.

## 3. Теория – анализ и проектиране на решението

Приложението е разделено на шест големи модула: controllers, css, models, scripts и validating.

Модулът controllers съдържа подмодулите : deleteGameController.php, importGameJsonController.php, listGamesJsonController.php, loginController.php, logoutController.php, registerControlle.php, updateGameJsonController.php.

Модулът css съдържа подмодулите : createEditJson.css

Модулът models съдържа подмодулите : Database.php, Game.php, User.php

Модулът pages съдържа подмодулите : createJsonFrom.php, editGame.php, editJsonForm.php, gamesJson.php, home.php, importGameJson.php, login.php, logout.php, register.php, viewOrDownloadJson.php

Модулът scripts съдържа подмодулите : createEditJson.js, script.js

Модулът validating съдържа подмодулите : validate.php

## 4. Използвани технологии

За работата на системата е нужно приложението XAMPP като версията не би трябвало да има значение, но ние сме използвали Version 8.2.12. По-конкретно версиите на следните са: Apache - 2.4.58, phpMyAdmin - 5.2.1, PHP - 8.2.12.

За операционна система няма строги изисквания, защото XAMPP може да бъде инсталиран на Windows, macOS и Linux.

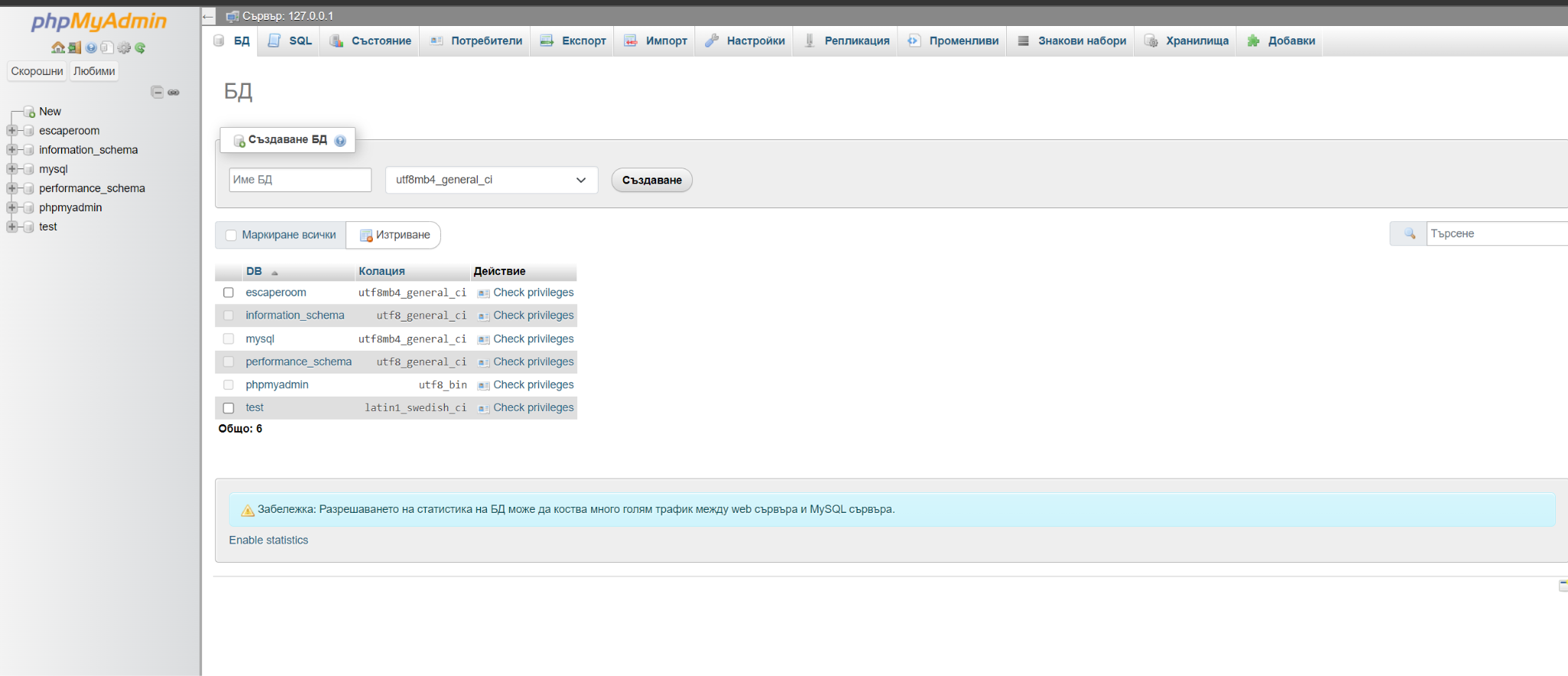
Системата не изисква други модули или приложения за своята работа.

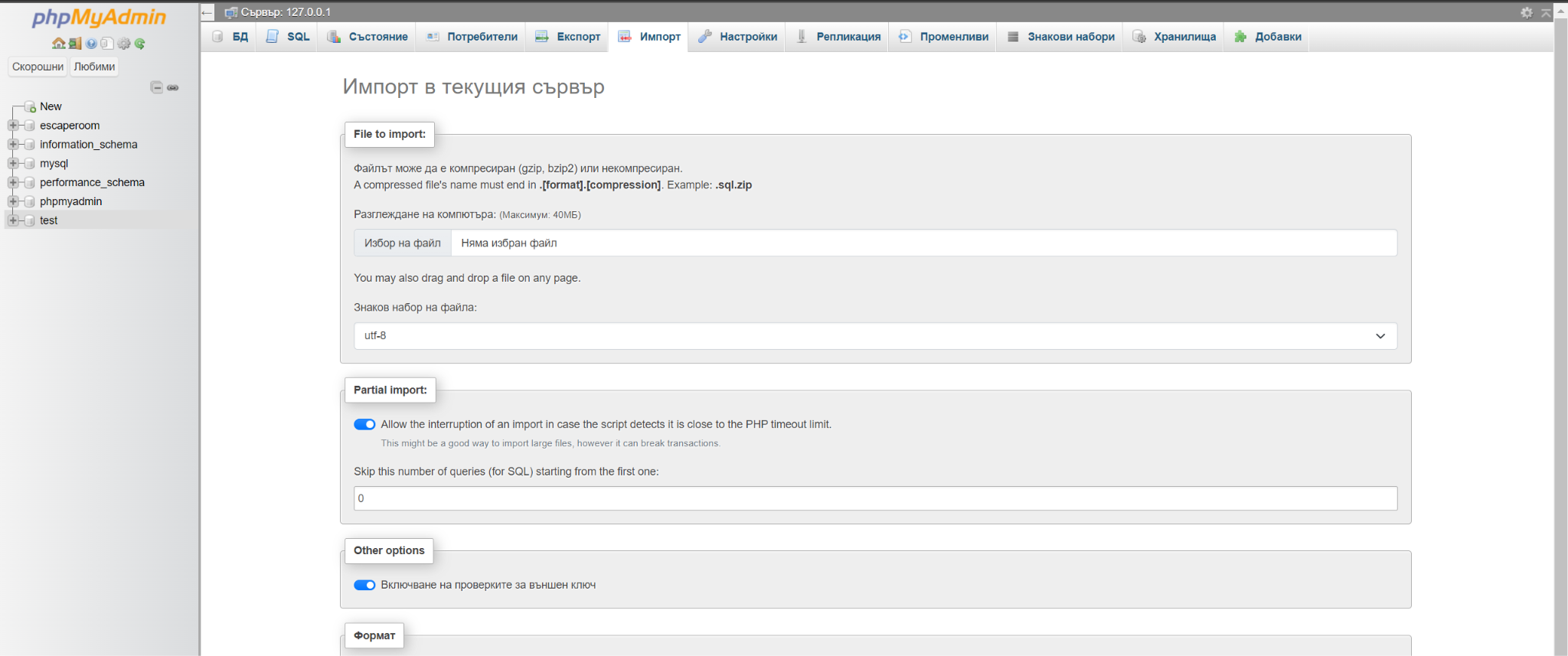
## 5. Инсталация, настройки и DevOps

За да можем да използваме системата трябва да изтеглим всички файлове със сорс код от качения архив в папката PROJECT. Също така, ще е нужно да се изтегли и приложението XAMPP, което съдържа Apache HTTP Server, MySQL база от данни и други необходими инструменти за използване на езика за програмиране PHP. Когато XAMPP е инсталирано изтеглената папка PROJECT трябва да бъде преместена в папката hdocs в папката на XAMPP. След успешното изпълнение на горните стъпки сме готови да заредим примерната база данни и да използваме приложението.

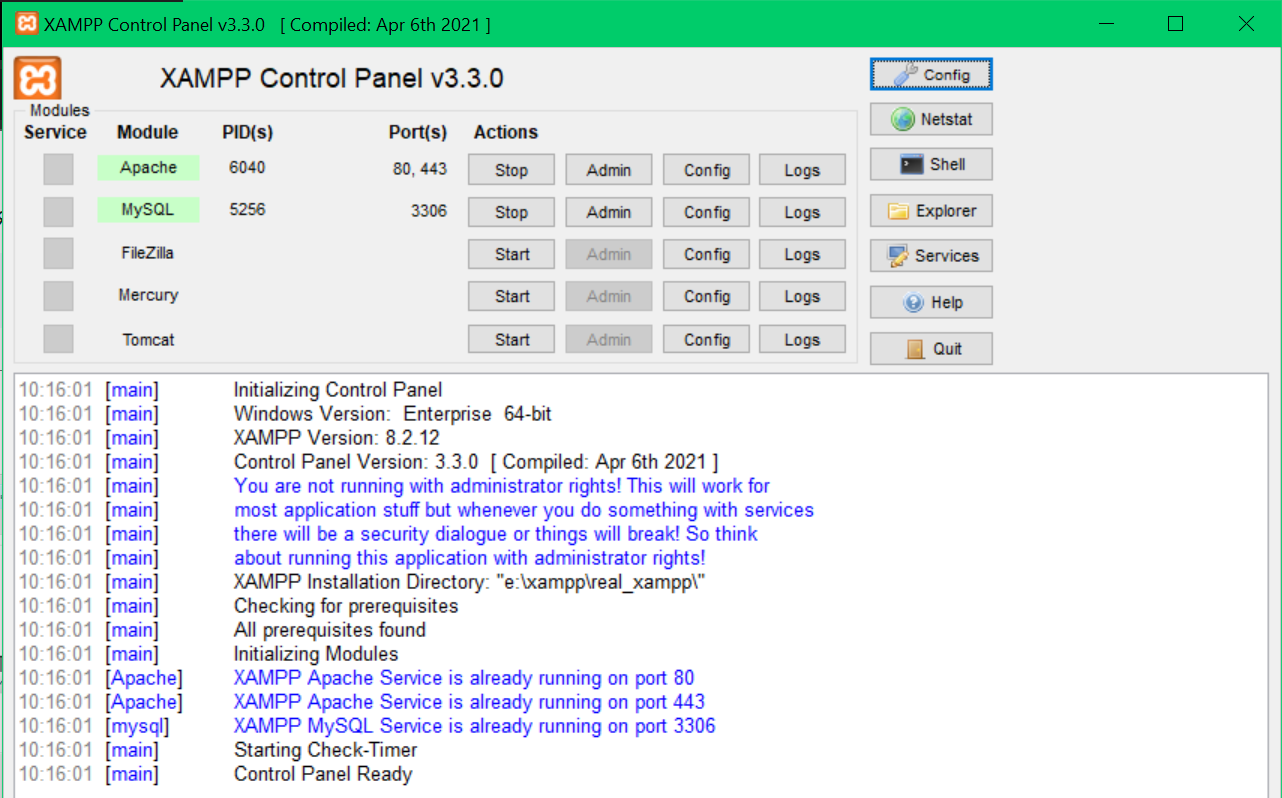
1.**Зареждане на базата данни.**

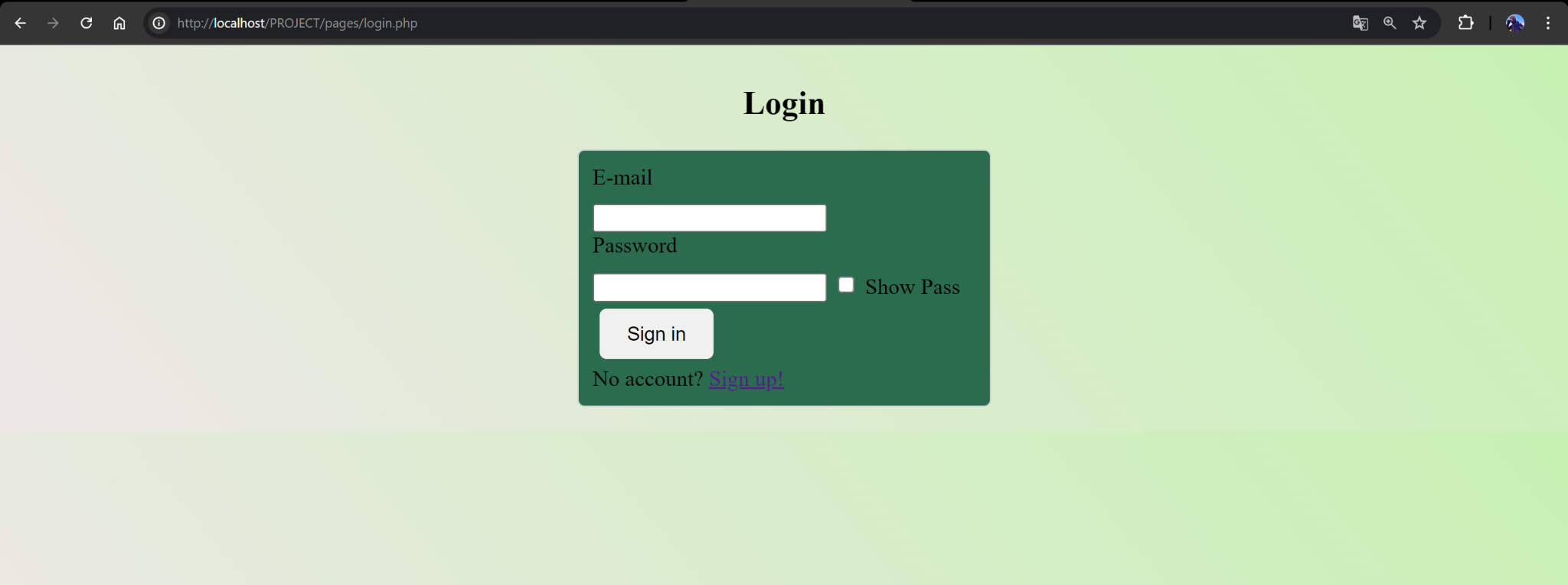
Трябва да отворим браузъра при пуснат сървър от XAMPP и да достъпим адреса localhost/phpmyadmin/. След това трябва да натиснем навигационния бутон “БД” и създадем база данни с име escaperoom. Когато направим това успешно, следващата стъпка е да се натисне бутона “Импорт”, където да се избере файл - SQL скрипт, чрез който да се зареди базата данни и да се натисне бутона “Импорт”. Този скрипт може да бъде намерен в папката SQL Script, която е подпапка на папката PROJECT. По този начин вече ще имаме заредена база данни и ще може да тестваме/използваме системата.





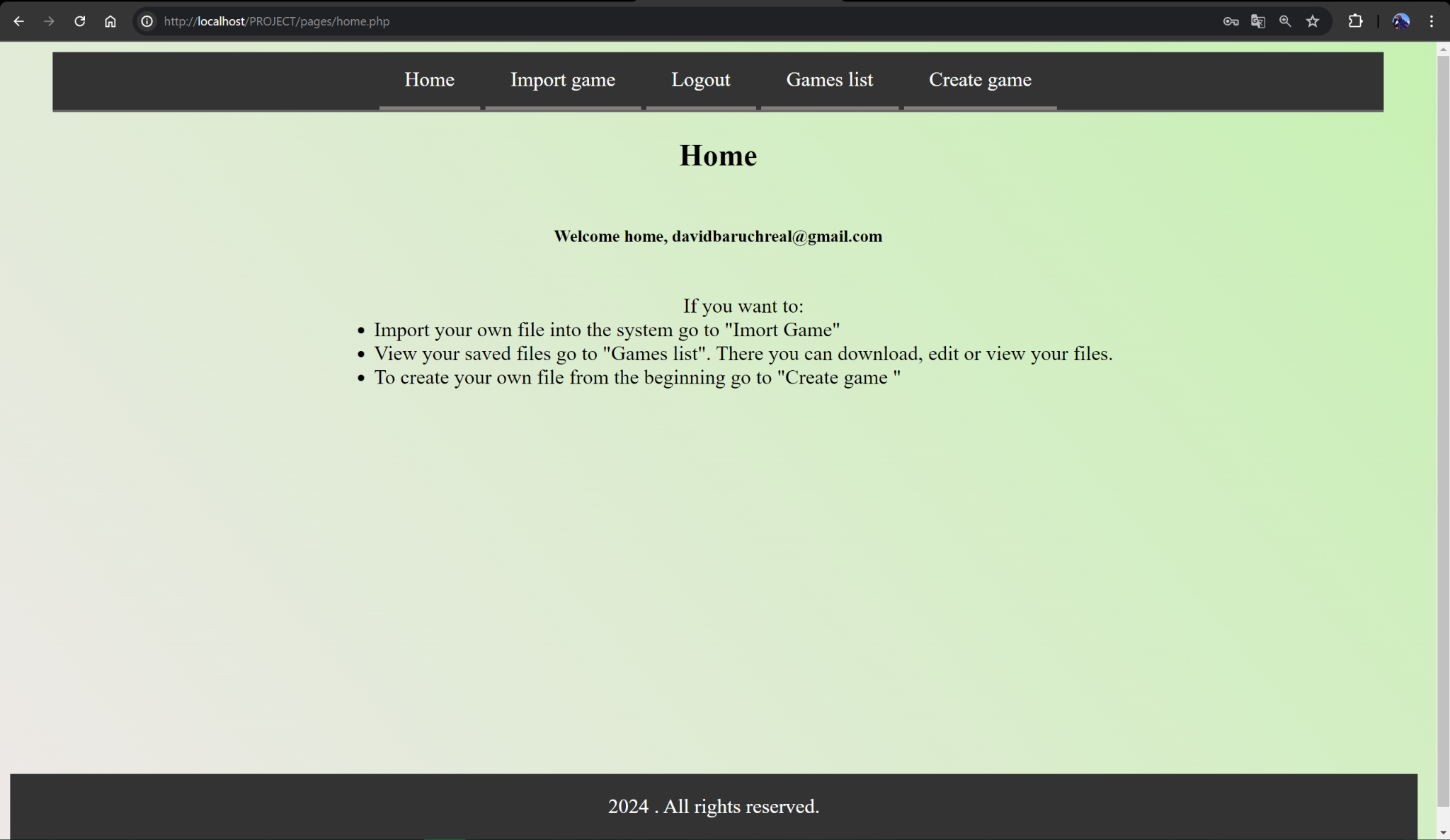
2.**Използване/тест на системата**

След като има заредена база данни вече може да преминем към влизане и използване/тест на системата. За да постигнем тази задача е необходимо да достъпим адреса localhost/PrOJECT/pages/login.php при пуснат сървър от XAMPP.

Достъпвайки горния адрес ще достигнете до страницата за влизане, където може да създадете ваш акаунт или да ползвате създаден такъв. Примерен акаунт с вече създадени и запазени стой/JSON файлове е акаунта с email: [davidbaruchreal@gmail.com](mailto:davidbaruchreal@gmail.com) и парола: 1234. След като се регистрирате и влизите чрез вашия или тестови акаунт, може да използвате функционалностите на системата.

## 6. Кратко ръководство на потребителя

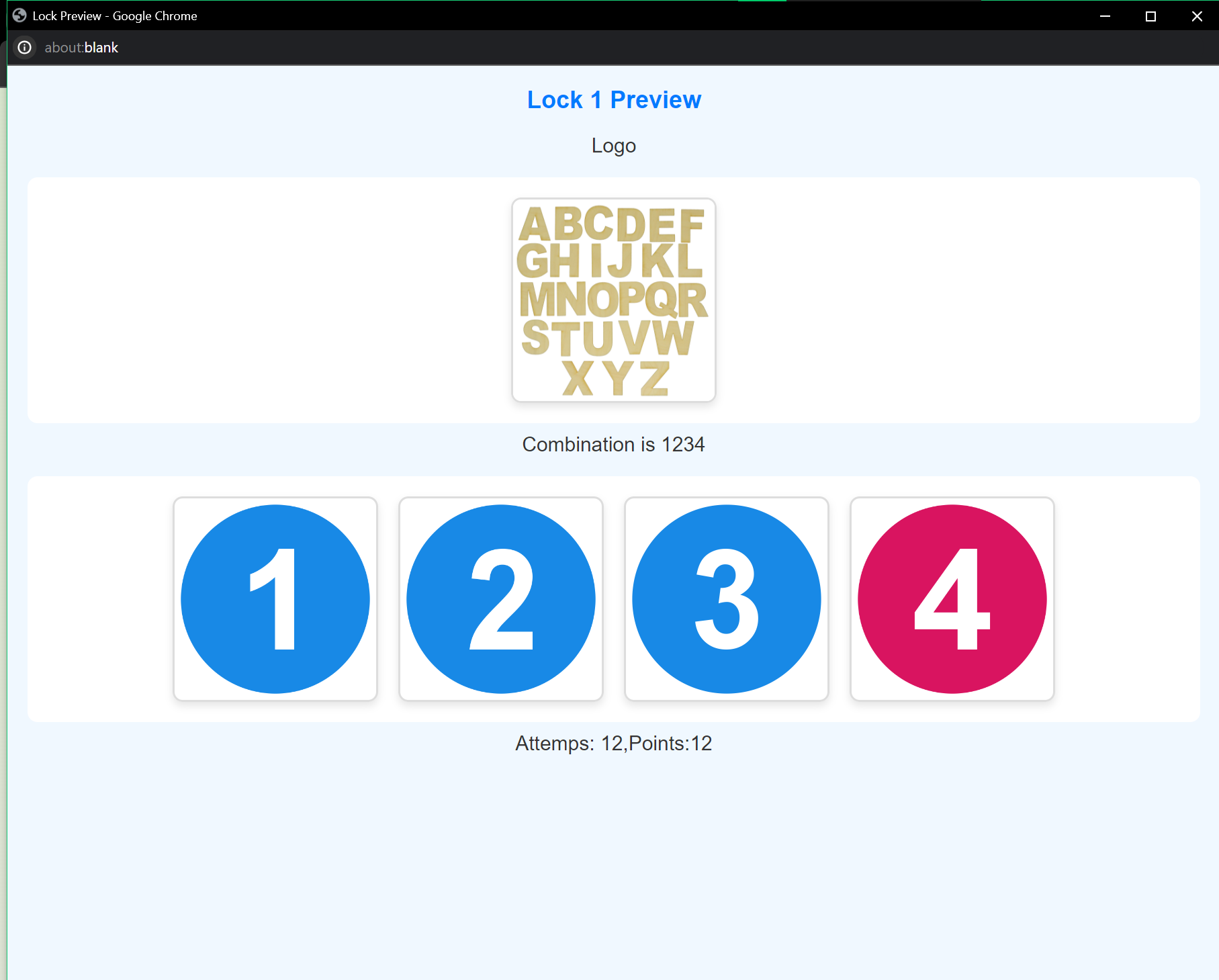
Когато влезем в системата чрез създаден от нас акаунт, ще се озовем на главната страница (Home page). В горната част ще намерим навигация из системата както и към функционалностите на системата.



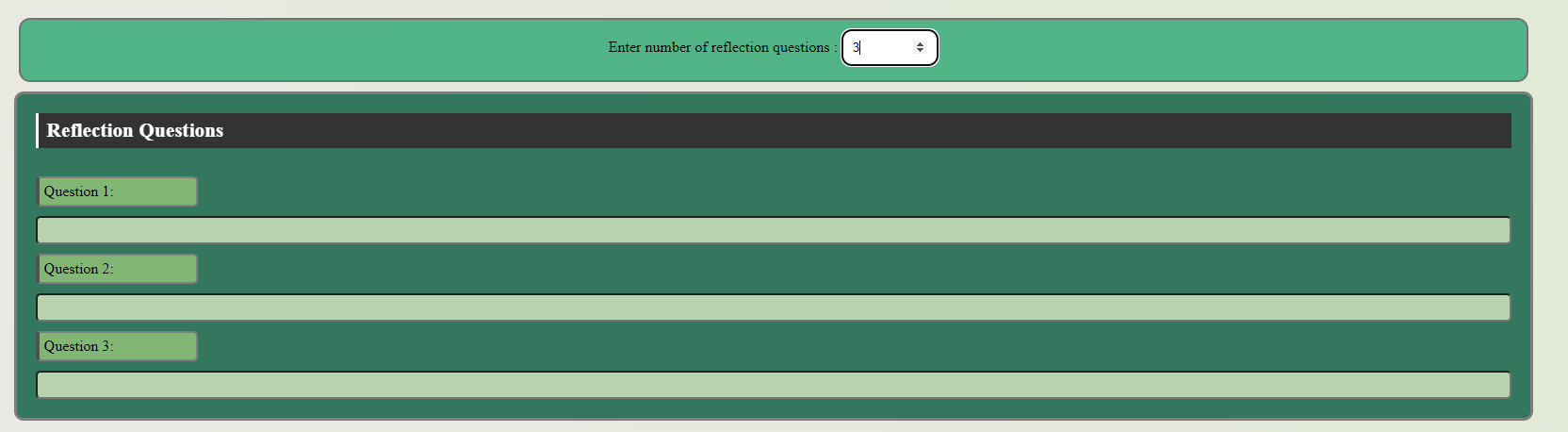
1. **Създаване на json файл за ескейп стая**

За да създадем нов json файл за ескейп стая трябва да натиснем бутона “Create game”.

След това ще бъдем пренасочени към нова страница, в която ще можем да въведем желания от нас брой ключалки (locks).

Когато сме въвели желания брой ключалки, можем да преминем към тяхното попълване. Ако изберем даден тип за ключ, ще ни се заредят предварително дефинирани стойности, които ако желаем можем да променим. Също така, можем да прегледаме ресурсите за дадена ключалка като натиснем бутона “Preview lock”. Като в прозореца, който се отвори ще видим броя точки, опити ,отговора и снимка за всяка част от отговора.

Също така можем да окажем и броя на въпросите за размисъл (reflection questions).



Когато приключим със създаването на файла най-отдолу ще открием и бутонът за изтегляне на файл, чрез който да изтеглим създадения от нас файл.

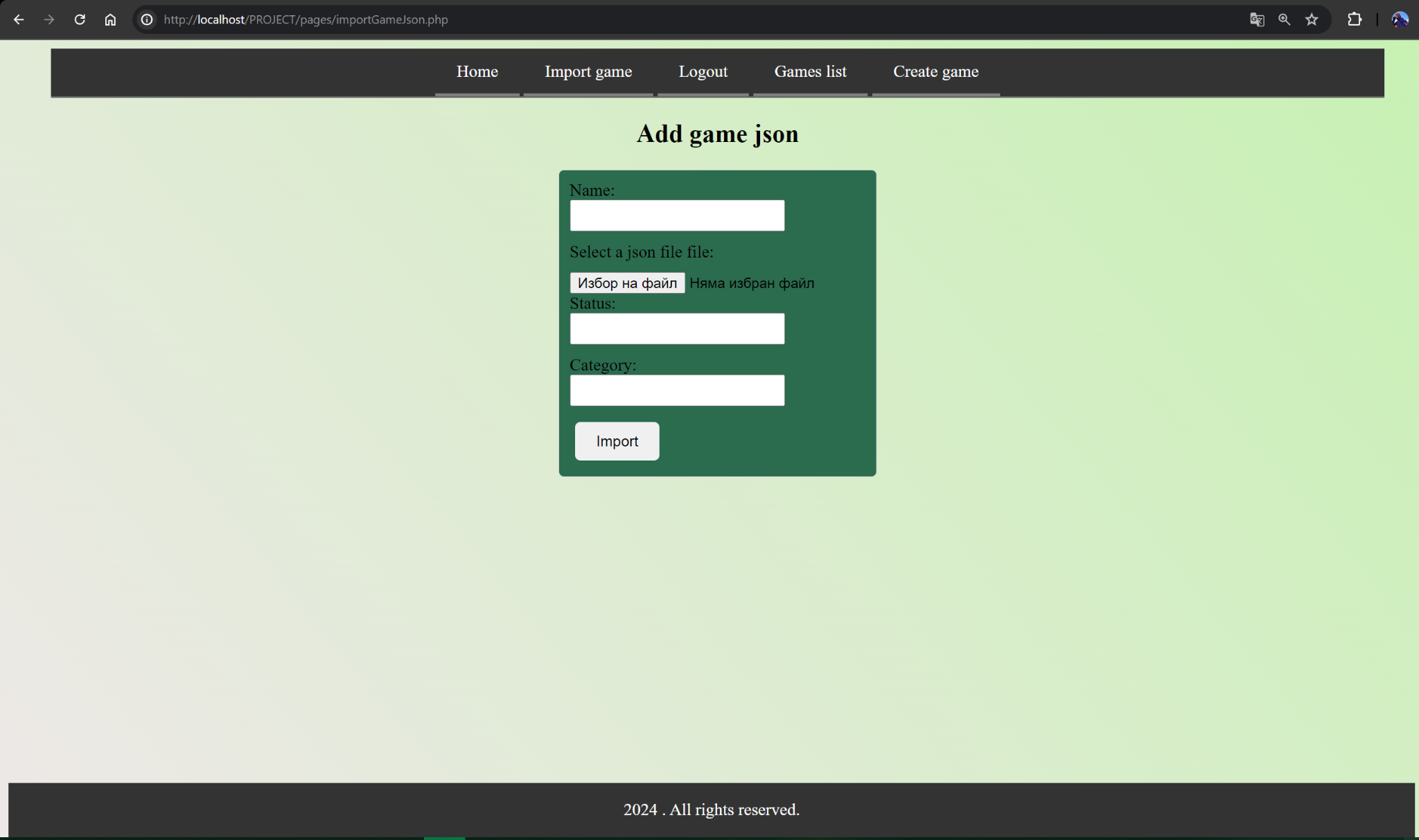


Аналогично на горната редакция, ако желаем можем да редактираме JSON файла и директно като текст. Това става възможно чрез бутона “Create without formatting”, който отваря текстово поле, където директно можем да създадем нашия файл или да го редактираме. Тази възможност я има и в секцията за редакция на вече качен файл.



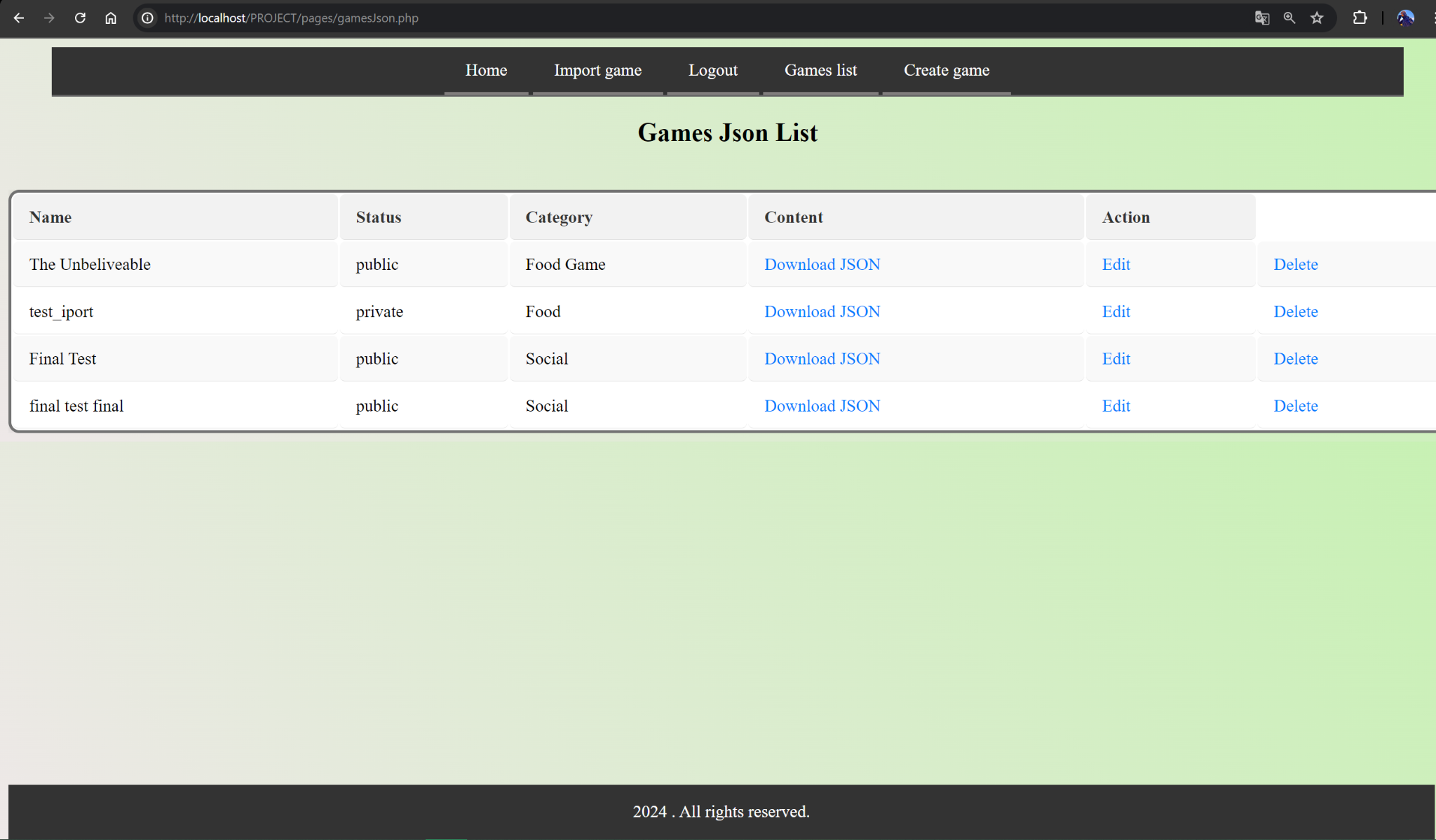
1. **Импортване на готов json файл**

Когато искаме да импортнем съществуващ готов json файл в системата, от главната страница, която вече показахме, трябва да изберем бутона “Import game”. Това ще ни пренасочи към нова страница, където ще трябва попълните необходимата информация за вкарване на файла в системата, за да имате достъп до него. Тази информация е име на файла/име на ескейп стаята, самият json файл, статуса на ескейп стаята и нейната категория. След като направите това при натискането на бутона “Submit”, вашият json файл ще се запази в системата и ще може да го редактирате, премахвате, теглите или просто да го прегледате.

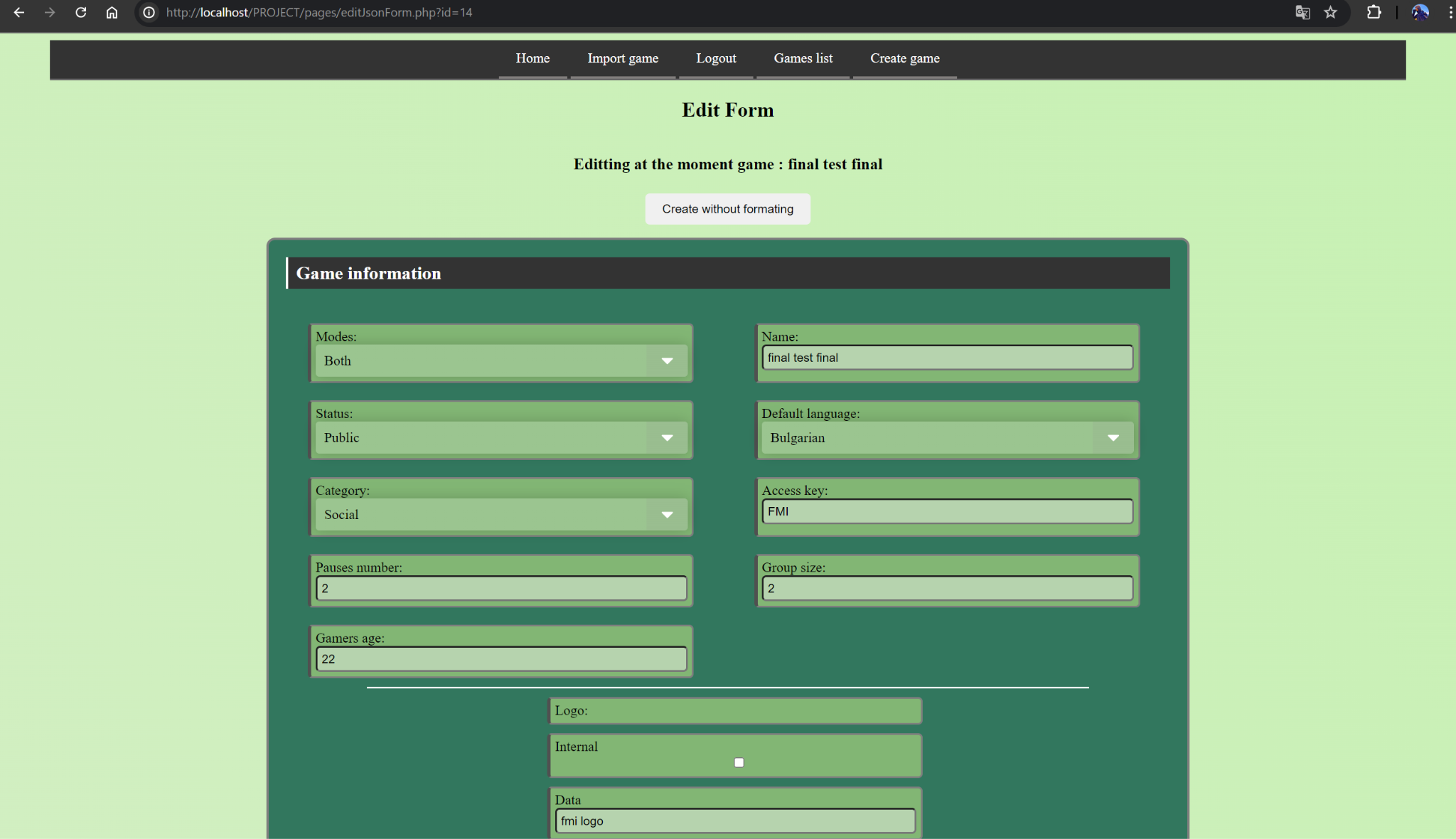


1. **Преглед на вашите json файлове/ескейп стаи**

За да прегледаме нашите json файлове/ескейп стаи в системата, след като сме влезли в нея, трябва от главната страница да изберем бутона “Game list”. След като го натиснете ще ви се отвори страница, в която ще можете да видите всички ваши ескейп стае/json файлове в системата. От там чрез различните бутони в дясно от файловете можете да изтеглите файл, да го редактирате, да го премахнете от системата или напълно да го замените с нов такъв директно. Също така ще виждате името на стаята/файла, статуса и категорията на всеки файл.



При едитването на файл, ще Ви се показват вече съществуващите стойности от json файла.



Допълнително можем да редактираме направо json файла и директно чрез текстовото поле посредством бутона - .

## 7. Примерни данни

Акаунт за тестване: email - [davidbaruchreal@gmail.com](mailto:davidbaruchreal@gmail.com), парола - 1234

SQL Script за базата данни - escaperoom.sql, намиращ се в папката Test Materials в папката PROJECT от архива

Примерен JSON файл - food\_game\_mod.json, намиращ се в папката Test Materials в папката PROJECT от архива

## 8. Описание на програмния код

1. Модулът **controllers** съдържа бизнес логиката на приложението съставена чрез езика PHP, като в него има подмодули за всяка функционалност на системата. Следните подмодули на модула **controllers**  са :
   1. **deleteGameController.php** - съдържа логика, която се свързва с базата данни, за да се осъществи триене на създадена игра. Картина, която съдържа текст, екранна снимка, Шрифт, номер

      Описанието е генерирано автоматично
   2. **importGameJsonController.php** - съдържа логика, която се свързва с базата данни, за да се осъществи добавяне на нова игра. Картина, която съдържа текст, екранна снимка, Шрифт, документ

      Описанието е генерирано автоматично
   3. **listGamesJsonController.php** - съдържа логика, която се свързва с базата данни, за да се изберат всики игри. Картина, която съдържа текст, екранна снимка, Шрифт, номер

      Описанието е генерирано автоматично
   4. **loginController.php** - съдържа логика, която се свързва с базата данни,за да верифицира дали съществуващ потребител си въвежда данните в системата, като проверява за съществуващ емайл и парола, асоциирана с акаунта на потребителя. Създава се сесия за влезлия в системата потребител.

Картина, която съдържа текст, екранна снимка, Шрифт

Описанието е генерирано автоматично

* 1. **logoutController.php** - приключва се сесиято на текущия потребител. Картина, която съдържа текст, екранна снимка, Шрифт, линия

     Описанието е генерирано автоматично
  2. **registerController.php** - съдържа логика за създаване на нов потребител, който да се запише в базата данни, като за данните на потребителя се използват емайл, парола,име и фамиля. Картина, която съдържа текст, екранна снимка, Шрифт, документ

     Описанието е генерирано автоматично
  3. **updateGameJsonController.php** - съдържа логика, която се свързва с базата данни, за да се обнови информацията за избрана от потребителя игра, която е редактирал. Картина, която съдържа текст, екранна снимка, Шрифт, номер

     Описанието е генерирано автоматично
  4. **viewOrDownloadJson.php** - съдържа логика за теглене или само преглеждане на избран json- файл на игра от страницата за представяне на всички игри.

Картина, която съдържа текст, екранна снимка, Шрифт, номер

Описанието е генерирано автоматично

1. Модулът **css**  съдържа css файлове за придаване на формат на уеб страниците на приложението . Следните подмодули на модула **css**  са :
   1. **createEditJson.css** - съдържа css код за оформление на страницата за създаване и редактиране на json-файл за игра.
2. Модулът **models** съдържа класове на обекти написани на PHP , които се използват в главните функционалности на приложението.Класовете са Database, User и Game . Следните подмодули на модула module са:
   1. **Database.php** - този клас осъществява свързването с базара данни. В него има и методи, с които се модифицират данните на обекти от клас User и Game. Тези методи позволяват редактиране, добавяне, триене на данни и селектиране на обекти от тип Game, и добавяне на данни на обекти от клас User

Картина, която съдържа текст, екранна снимка, Шрифт, номер

Описанието е генерирано автоматично

* 1. **User.php** - този клас описва информацията необходима на един потребител. Има параметри име,фамилия,емайл. Член данните са private и се използват get-ри за достъпването им. Картина, която съдържа текст, екранна снимка, Шрифт, номер

     Описанието е генерирано автоматично
  2. **Game.php** - този клас описва информацията необходима на една игра. Има параметри име , категория, статус и съдържанието на json-файл, като един голям низ. Член данните са private и се използват get-ри за достъпването им.

**Картина, която съдържа текст, екранна снимка, Шрифт

Описанието е генерирано автоматично**

1. Модулът **pages** представлява интерфейса на приложението , като е изплозване HTML и съответно PHP , за сесията на текущия потребител.Всеки файл съдържа меню,чрез което да се превижва потребителя през различните страници. Следните подмодули на модула **pages** са:
   1. **createJsonFrom.php** - базов файл, в който се зарежда createEditJson.js скрипта, за генериране на форма за създаване на нов json файл за игра.
   2. **editGame.php** - в този модул, се представя форма, в която може да се редактира избра игра от страницата за представяне на всички игри. Съответно има полета за редактиране на име,статус и категория на играта, и вкарване на нов json-файл, който да съответства на играта. Всички полета задължително се попълват, без избирането на нов json-файл.
   3. **editJsonForm.php** - базов файл, в който се зарежда createEditJson.js скрипта, за генериране на форма, като данните на избрания json-файл се вписват в полетата на формата, за да бъдат редактирани. json-файл на игра се избира от страницата за представяне на всички игри.
   4. **gamesJson.php** - в този модул се представят всики игри, като към всяка игра и са показват данните и има опции за редакци и триене на играта.
   5. **home.php** - в този модул се представя входната страница на системата , съпроводена с текст, който посреща текущия потребител.
   6. **importGameJson.php** - този модул представява форма за създаване на игра,в която се въжда информация за име,статус, категория и имортиране на json - файл.
   7. **login.php** - модул, представяващ форма за вписване на емайл и парола за същестуващ потребител, за да влезе в системата.
   8. **logout.php** - модул, който дава възможност на текущия потребител да прекрати своята сесия и да излезе от приложението.
   9. **register.php** - модул, представяващ форма за вписване на емайл и парола, име и фамилия на несъщестуващ потребител, за да се регистрира в системата.
2. Модулът **scripts** представлява javascript логика, която се използва за предаване на динамика на приложението.Следните подмодули на модула **scripts** са:
   1. **createEditJson.js** - този файл генерира форма за създаване или редактиране на json-файл. В зависимост от операцията при редакция се взима json-файла избран от страницата за предствавяне на всички игри и данните на файла се насят върху полетата от формата за редакция, ако се създава нов json-файл се използва празен обект, на когото се нанасят данните, вкарани от потребителя.
   2. **script.js** - този файл служи за показване или криене на парола на потребител при влизане в системата.
3. Модулът **validating** представява логика за валидиране на данни при създаване или редактиране.Следните подмодули на модула **validating** са:
   1. **validate.php** - съдържа логика за валидиране на форми при създаване на потребител(валидация за правилен мейл,проверяване дали този мейл е зает, и задължаване на попълване на всички полета ) и създаване и редактиране на игра(проверяване дали името е заето, и задължаване на попълване на всички полета ).

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

**Цветан** - Разработих функционалността, която ти дава възможност за редактиране на JSON файлове, съдържащи информация за ескейп стая. Помогнах на колегите с визуализирането на JSON файловете. Разработих и обновяването към базата данни.

**Румен** - изготвих JavaScript-a, който се грижи за визуализирането на JSON файла и попълването на неговите данни, като може да се определят броя на ключалките, ресурсите и въпросите, за да може да се създаде или редактира файл, който да отговаря на желанието на потребителя.

**Дейвид -**  разработих интерфейсите на програмата като създаване на игра , страница с лист от всички създадени игри към които са предоставени и възможности за редакция и триене.Разработих и бизнес логиката съответно върху тези интерфейси ,както и меню за придвижване между страниците.

## 10. Какво научих (най-важните неща, които сте научили по време на курса и при разработването на проекта-за всеки студент)

**Цветан** - По време на курса, лично аз научих много за основните похвати и практики при създаването на уеб приложения. Чрез реферата, разбрах и приложих наученото, през упражненията, за създаването на уеб страница, използвайки основно HTML и CSS. Разбрах и как да добавим различни интерактивности и да придадем динамичност на нашите уеб страници чрез PHP и JavaScript, които приложихме във финалния проект. Курсът ми даде една цялостна представа за това как се създава и поддържа уеб приложение.

**Румен** - Най-важните неща, които научих по време на курса са как се съчетават различните езици и технологии, за да се получи цялостен продукт. Също така за първи път разработвах проект с други хора, а не самостоятелно както досега и това отново ми показа как съвкупност от различни файлове написани от различни хора, могат да се обединят и да дадат система, която да бъде използвана. Също така, научих и основните на уеб програмирането, HTML, CSS, JavaScript и PHP, както и да използвам XAMPP и да настройвам портове.

**Дейвид -** Благодарение на курса навлязох по-надълбоко в уеб програмирането. От правене на статични HTML страници, достигнах до умение за разработване на динамично уеб приложение. Научавайки JS и PHP, успях да придобия идея и умения, за създаване на едно full stack приложение.

## 11. Използвани източници

Всички презентации от курса (лекции и упражнения).

Гит хъб репото на асистента на 5-та група - Атанас Куртаков: <https://github.com/Nasko-Kurtakov/web-si> - дата на посещение 28.05.2024

Предал (подпис): ………………………….

/*фн, имена, спец., група*/

Приел (подпис): ………………………….

/проф. д-р *Милен Петров*/