

Имена: Стефан Шиваров (фн: 9MI0600175); Анастасия Маджарова (фн: 0MI0600156); Никола Илиев (фн: 1MI0600159)

Начална година: Летен семестър 2025 **Програма:** бакалавър, (СИ) **Курс:** 3

Тема: 28.1

Дата: 10.06.2025

Предмет: w25prj_SI_final

Имейл: stefan.shivarov.jr@gmail.com, a_madjarova@abv.bg, ndiyanov@uni-sofia.bg

преподавател: доц. д-р Милен Петров

ТЕМА: w25/28.1

1. Условие

Проектът представлява уеб-базирана платформа на име **QuoteShare**, предназначена за събиране, споделяне и организиране на вдъхновяващи или запомнящи се цитати от известни личности. Всеки регистриран потребител може да добавя собствени цитати, да създава лични колекции, да аотира избрани цитати, да ги харесва, да ги докладва при злоупотреба и да експортира избрани колекции във файл.

2. Въведение – извличане на изисквания

Роли:

- **Потребител**
 - Регистрация и вход
 - Добавяне на нови цитати
 - Харесване на цитати
 - Запазване на любими цитати
 - Докладване на неподходящи цитати
 - Създаване на колекции
 - Лични аотации по цитати
 - Добавяне на цитати към колекция
 - Експорт на колекцията към PDF, CSV, HTML, BibTeX формати
 - Промяна на парола
 - Споделяне на цитати във Facebook, Twitter(X) или e-mail
- **Администратор**
 - Преглед на общи статистики за системата
 - Преглед на най-харесвани цитати
 - Преглед на всички докладвани цитати
 - Изтриване на цитати
 - Изтриване на потребители
 - Търсене на потребители

- Промяна на роля на потребител
- Визуализация на всички предприети действия в системата под формата на логове
- Търсене (филтриране) на логове
- Изтриване на логове

Функционални изисквания:

1. Системата трябва да позволява на всеки потребител да се регистрира чрез уникално потребителско име и парола.
2. Системата трябва да валидира въведените данни при регистрация – потребителското име не може да се повтаря, а паролата трябва да бъде с минимална дължина от 6 символа.
3. След регистрация потребителят трябва да може да влезе в системата чрез формуляр за вход.
4. След успешен вход системата трябва да предоставя достъп до пълната функционалност на потребителската роля.
5. Всеки потребител трябва да може да достъпва страницата на своя профил
6. Всеки потребител трябва да може да променя паролата си
5. Системата трябва да позволява на потребителя да добавя нов цитат, като въведе следните задължителни данни: заглавие, съдържание и автор.
6. Системата трябва да визуализира списък с всички цитати, подредени по дата на добавяне (най-нови най-отгоре).
7. Всеки потребител трябва да може да харесва цитати, като всеки потребител има право на едно харесване на даден цитат.
8. Всеки потребител трябва да може да докладва даден цитат чрез натискане на бутон за докладване.
9. Системата трябва да позволява на всеки потребител да създава неограничен брой лични колекции от цитати.
10. Всеки потребител трябва да може да добавя и премахва цитати от собствените си колекции.
11. Всеки потребител трябва да може да копира/споделя цитат в социалните мрежи.
12. Системата трябва да позволява на всеки потребител да може да се добавя анотация – бележка под формата на текст към всеки цитат.
13. Системата трябва да позволява на всеки потребител да може да разглежда вече добавените анотации към цитат.
14. Всеки потребител трябва да може да експортира избрана колекция във PDF, CSV, HTML, ViTeX формати, като файлът съдържа списък с цитати и съответните анотации.
15. Всеки потребител трябва да може да импортира цитати, чрез качване на CSV файл.

16. Системата трябва да предоставя отделен административен панел, достъпен само за администратор.
17. Администраторът трябва да може да вижда всички докладвани цитати.
18. Администраторът трябва да може да изтрива цитати.
19. Администраторът трябва да може да променя ролите на потребител
20. Администраторът трябва да може да изтрива потребителски акаунти
21. Администраторът трябва да може да вижда всички направени действия в системата под формата на логове
22. Администраторът трябва да може да изтрива логове
23. Администраторът трябва да може да филтрира логове
24. Системата трябва да показва на администратора статистика с:
 - a. общ брой активни потребители и цитати
 - b. най-харесвани цитати

Нефункционални изисквания:

- Добра разделимост на логика, изглед и данни.
- Конфигурируемост и разширяемост на архитектурата.
- Интуитивен потребителски интерфейс.

Ползи от реализацията:

а) Ползи за разработчиците

- **Придобиване на практически опит в изграждането на уеб приложения с трислойна архитектура MVC (модел - изглед - контролер).**
- **Изучаване и използване на PHP сесии.**
- **Управление на зависимости чрез Composer**, което отговаря на реалните практики в съвременната PHP екосистема.
- **Изграждане на система с потребителски роли**, което включва автентикация и оторизация.
- **Работа с релационна база от данни (MariaDB)**, включваща свързани таблици.
- **Използване на добри практики за локално разработване** - използване на XAMPP като лесно и достъпно решение за стартиране на уеб сървър и база от данни в локална среда.
- **Управление на проект** - проектът е документиран изцяло, лесно се конфигурира и пуска.
- **Работа в екип и разпределяне на задачи.**

а) Ползи за крайния потребител

1. **Интуитивен и лесен за използване интерфейс** за събиране и организиране на любими цитати.

2. **Персонализирани колекции с анотации**, които превръщат платформата в личен архив на вдъхновения и размисли.
3. **Функционалност за експортиране** на колекциите във формат PDF– подходящо за личен архив, разпечатване или споделяне.
4. **Възможност за социално взаимодействие чрез харесвания**, което дава индиректна оценка на популярността и въздействието на цитатите.
5. **Сигурност и качество на съдържанието**, осигурени чрез система за докладване и модериране от администратор.

3. Теория – анализ и проектиране на решението

В хода на разработване, беше предприето имплементирането на следните шаблони:

1. Архитектурен шаблон: MVC (Model-View-Controller)

Приложението е организирано според **MVC (Model-View-Controller)** архитектурен шаблон, който предлага ясно разделение на отговорностите и улеснява поддръжката и разширяемостта на системата.

Предимства от използването на MVC:

- Улеснява тестването и модулното разширяване на функционалности.
- Осигурява разделяне на отговорностите (separation of concerns).
- Подобрява възможностите за работа в екип – всеки може да работи независимо по слой.

2. *Composer – управление на зависимости*

- Автоматично зареждане на класове чрез PSR-4 autoloading.
- Организация на конфигурацията и разработката според съвременните PHP практики.

3. *XAMPP (Apache web server + MariaDB) – среда за разработка*

За локална разработка в проекта е използван XAMPP, който предоставя бърза и лесна настройка на локален уеб сървър с Apache и база данни MariaDB. Това позволява:

- Стартиране на проекта без нужда от допълнителни зависимости или конфигурации.
- Унифицирана локална среда за всички участници в екипа.
- Улеснено тестване и дебъгване чрез директна интеграция с htdocs директорията и XAMPP Control Panel.

4. Dependency Injection (DI) контейнер за контролери

Контролерите в приложението не създават сами своите зависимости, а ги получават отвън чрез **DI контейнер**. Това позволява централизирано управление на свързани обекти и улеснява промени в логиката без нужда от пренаписване на самите контролери.

4. Използвани технологии

Компонент	Технология
Език за програмиране	PHP 8.4
Среда за разработка	PhpStorm, Visual Studio Code
Уеб сървър	Xampp + Apache
Бази данни	MariaDB

5. Инсталация, настройки и пускане на проекта

1. Системни изисквания:
 - XAMPP
2. Пускане на проекта стъпка по стъпка
 - Стартираме Apache и MySQL от XAMPP Control Panel
 - Копираме проекта в htdocs директорията на XAMPP
 - Създаваме нова база данни чрез phpMyAdmin
 - Конфигурираме връзката към базата данни в съответния конфигурационен файл на проекта - config/database.php

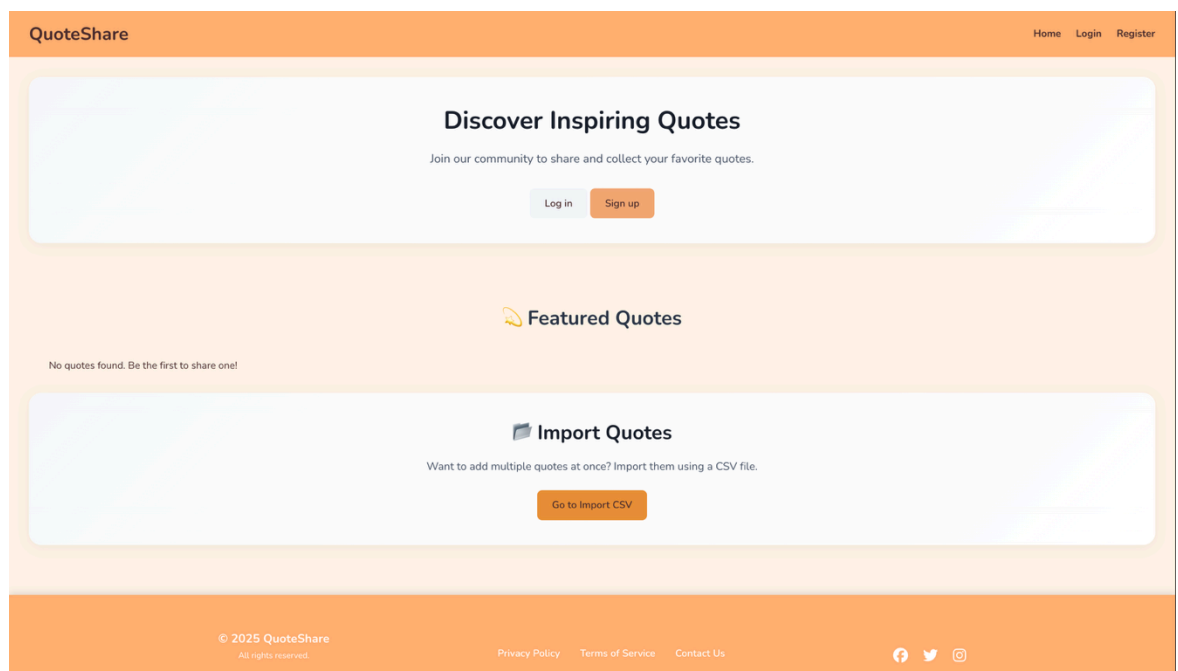
- Пускаме миграциите (migrations/db.sql), което ще създаде нашите таблици.
 - Пускаме сийдовете (seeds/seed.sql), което ще ни създаде админ акаунта.
3. Отваряме проекта на адрес: <http://localhost/<име на проекта>>

Акаунт за администратор:

- **потребителско име:** admin@quoteshare.com
- **парола:** admin123

6. Кратко ръководство на потребителя

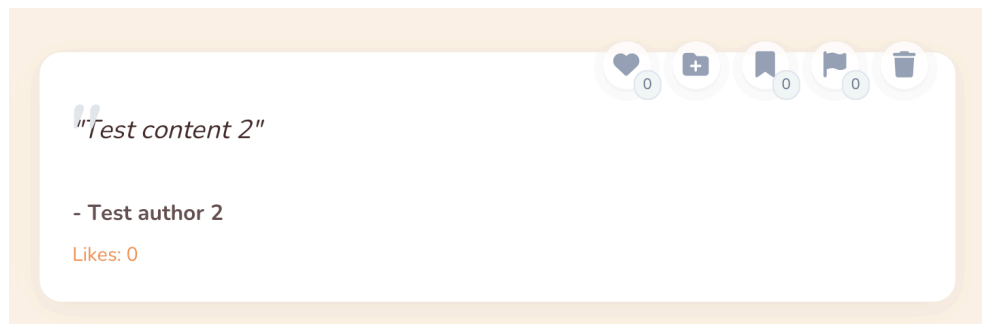
- Начална страница - тя има два типа в зависимост от това дали потребителят се е вписал в системата или не
 - Ако потребителят не е вписан в системата, той вижда:



- Имаме меню най-отгоре. В лявата част на менюто виждаме името на платформата QuoteShare. Когато се натисне, то ни отвежда към началната страница. В дясната част на менюто имаме Home, Login, Register. Когато потребител натисне Home, бива отведен до началната страница. При натискане на Login зареждаме страница за

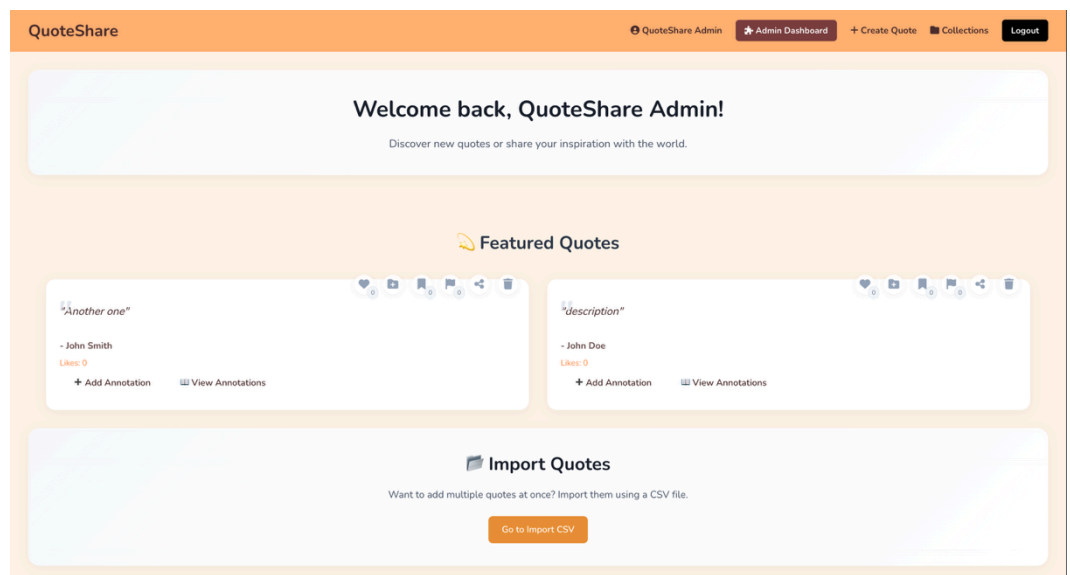
вход, а при натискане на Register зареждаме страница за създаване на акаунт в нашата система.

- Под навигацията имаме банер (hero section), който ясно обяснява на потребителя каква е целта на платформата. Имаме два бутона под нея за вход и за регистрация.
- Под тази секция се зареждат всички цитати, добавени от различни потребители. Всеки цитат може да бъде харесан, добавен към любими, докладван на администратора или добавен към лична колекция. Изтриването на цитат е възможно само за администратора или за потребителя, който го е създал. Можем да го добавим към наша колекция. Най-отдолу на цитата се показва общия брой харесвания.



Когато не е вписан потребителя в системата и натисне тези действия - бива прехвърлен към страницата за вход.

- Ако потребителят е вписан в системата, той вижда:



- Единствената разлика тук е, че горе дясно в менюто се премахват опциите за вход и регистрация. Сега се показва името на логнатия потребител, което при натискане води до профила на логнатия потребител, бутон, който води към админ панела (ако сме админ),

линк към страницата за създаване на цитат, линк към страницата за създаване на колекция и бутон за изход от системата.

- Банерът вече показва лично съобщение на потребителя “Добре дошъл обратно!”. В момента потребителят е логнат и може да натиска и изпълнява бутоните, които са над всеки цитат - “Харесване/Добавяне към колекция/Запазване/Докладване/Споделяне”. Допълнително има бутон за изтриване, който може да бъде извикан само от администратор или ако потребителят е собственик на цитата.

- Страница за регистрация

QuoteShare

Home Login Register

Create Your Account

Full Name

Email Address

Password

Confirm Password

Register

Already have an account? Login here

© 2025 MyApp. All rights reserved.

- Потребителят въвежда своите имена, имейл адрес, парола и потвърждение на паролата. Когато се натисне бутона за регистрация формата се изпълнява и потребителят бива насочен към страницата за вход само ако данните са валидни.

- Страница за вход

QuoteShare

Home Login Register

Login to QuoteShare

Email

Password

Login

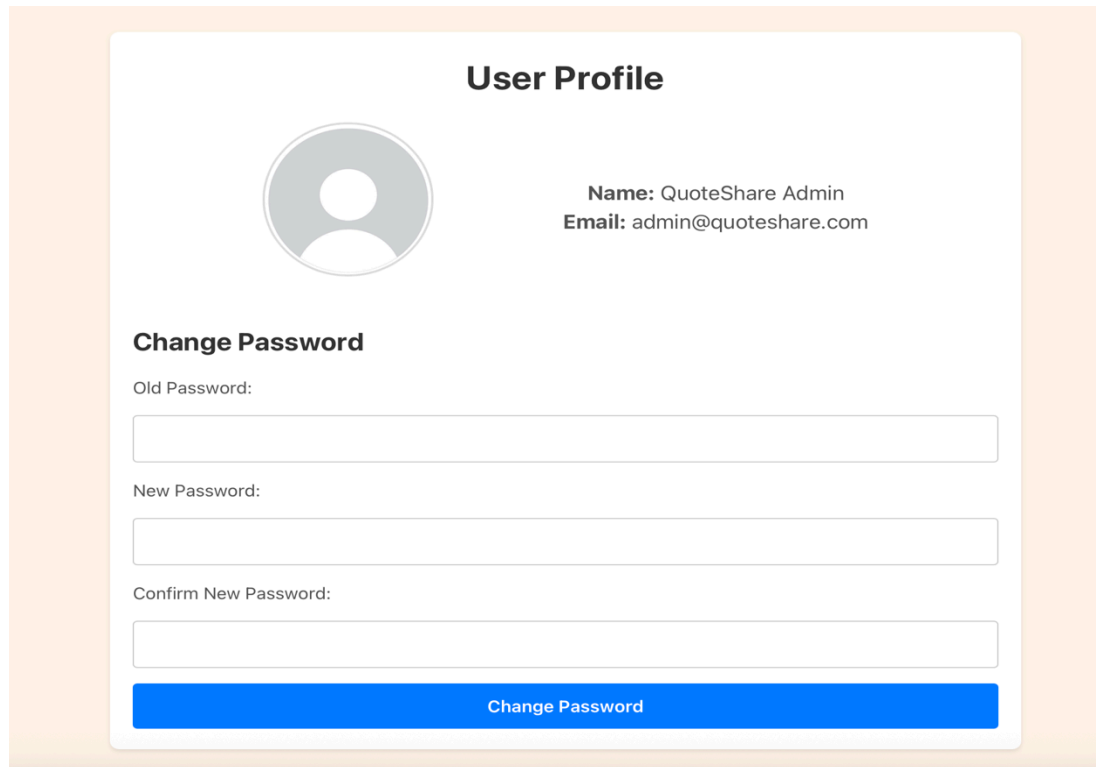
Don't have an account? Register here

© 2025 MyApp. All rights reserved.

[Privacy Policy](#) [Terms of Service](#) [Contact Us](#)

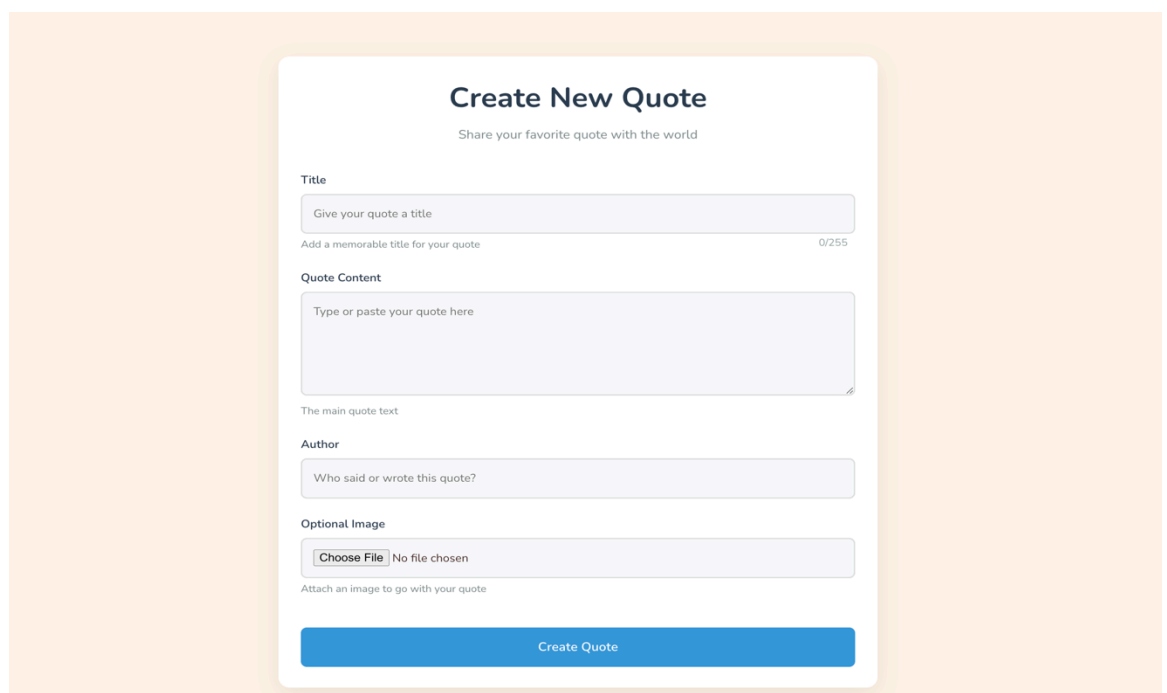
- Потребителят въвежда своя имейл адрес и парола с които се е регистрирал в системата. След като натисне “Login” ако данните са правилни ще му бъде създадена PHP сесия и ще бъде вкаран в системата успешно.

- Изглед на потребителски профил
 - При натискане върху линка с името на логнатия потребител в навигацията, системата отваря изглед с информация за потребителя.



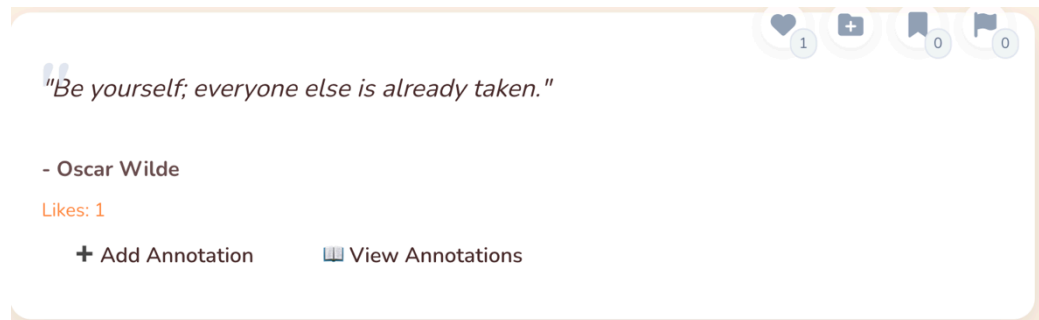
The image shows a 'User Profile' form. At the top, there is a circular placeholder for a profile picture. To the right of the picture, the user's name and email are displayed: 'Name: QuoteShare Admin' and 'Email: admin@quoteshare.com'. Below this, the section is titled 'Change Password'. It contains three input fields: 'Old Password:', 'New Password:', and 'Confirm New Password:'. At the bottom of the form is a blue button labeled 'Change Password'.

- В този изглед също така има форма, която позволява промяна на паролата, като първо трябва да се въведе старата парола и след това новата. Достъп до този изглед има само потребителя, за чийто профил е този изглед.
- Създаване на цитат



The image shows a 'Create New Quote' form. At the top, the title 'Create New Quote' is followed by the subtitle 'Share your favorite quote with the world'. The form has four main sections: 1. 'Title' with a text input field and a character count '0/255'. 2. 'Quote Content' with a large text area and a character count '0/255'. 3. 'Author' with a text input field. 4. 'Optional Image' with a 'Choose File' button and a 'No file chosen' status. At the bottom is a blue button labeled 'Create Quote'.

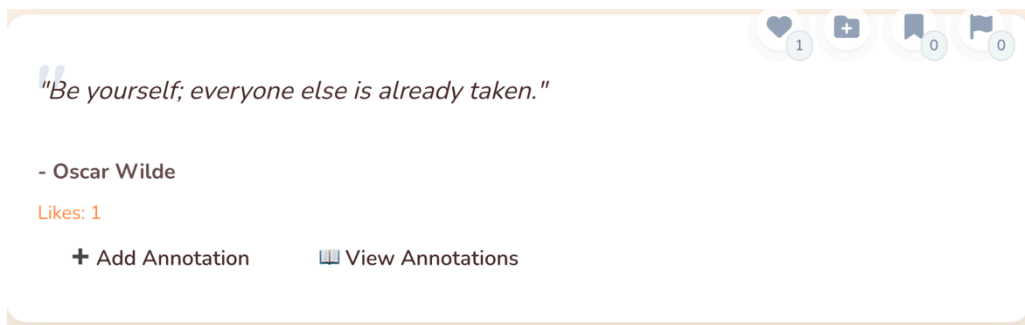
- За създаването на цитат е необходимо потребителят да въведе заглавие (title), съдържание на цитата (content) и автор на цитата (на кого е), както и изображение (по избор). След като бъде натиснат бутона Създай цитат ако всички данни са коректни ще бъде прехвърлен потребителя към началната страница, където ще види своя нов качен цитат.
- Добавяне на анотация към вече съществуващ цитат



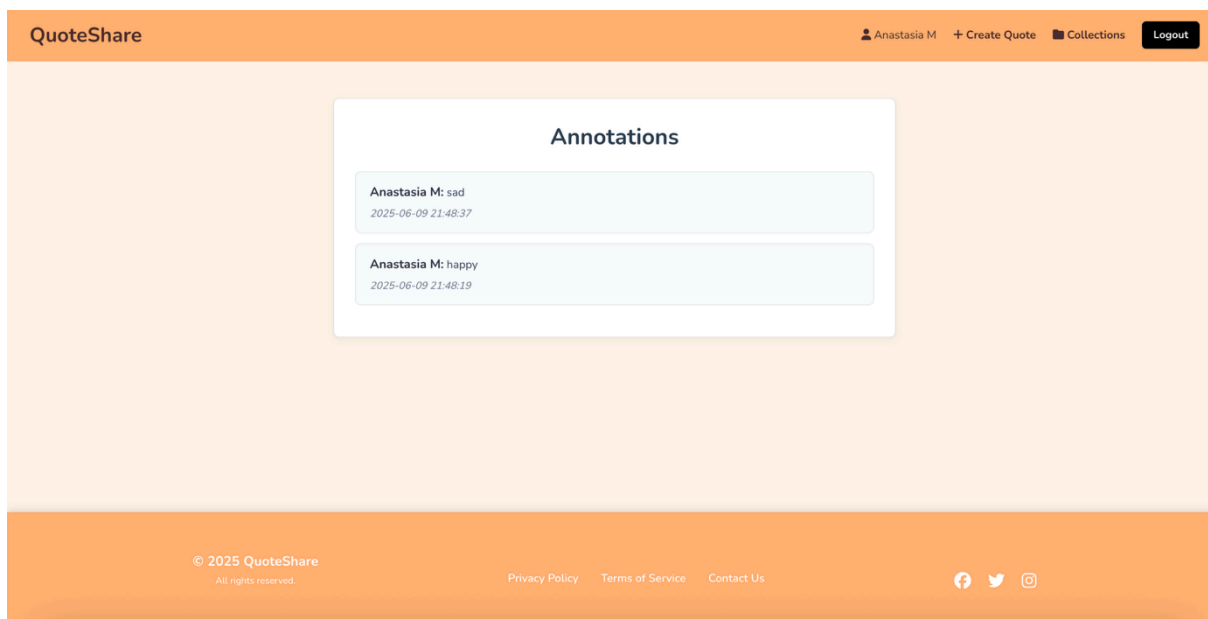
- След натискане на бутона „Add Annotation“, който е наличен за всеки цитат в началната страница, потребителят бива препратен към страница където трябва да въведе бележка за анотацията, която иска да добави

A screenshot of the "Add new annotation" form on the QuoteShare website. The form has a title "Add new annotation", a subtitle "How do you feel while reading this quote?", a "Note" label, a text input field with placeholder "Add a note", and a "Create Annotation" button.

- За създаването на анотация е необходимо потребителят да въведе бележка (note). След като бъде натиснат бутона за създаване на анотация „Create Annotation“, ако всички данни са коректни, анотацията ще бъде създадена успешно и потребителят ще бъде пренасочен към началната страница.
- Разглеждане на добавени анотации към цитат



- След натискане на бутона „View Annotations“ който е наличен за всеки цитат в началната страница, потребителят бива препратен към страница където може да види всички анотации добавени към даден цитат

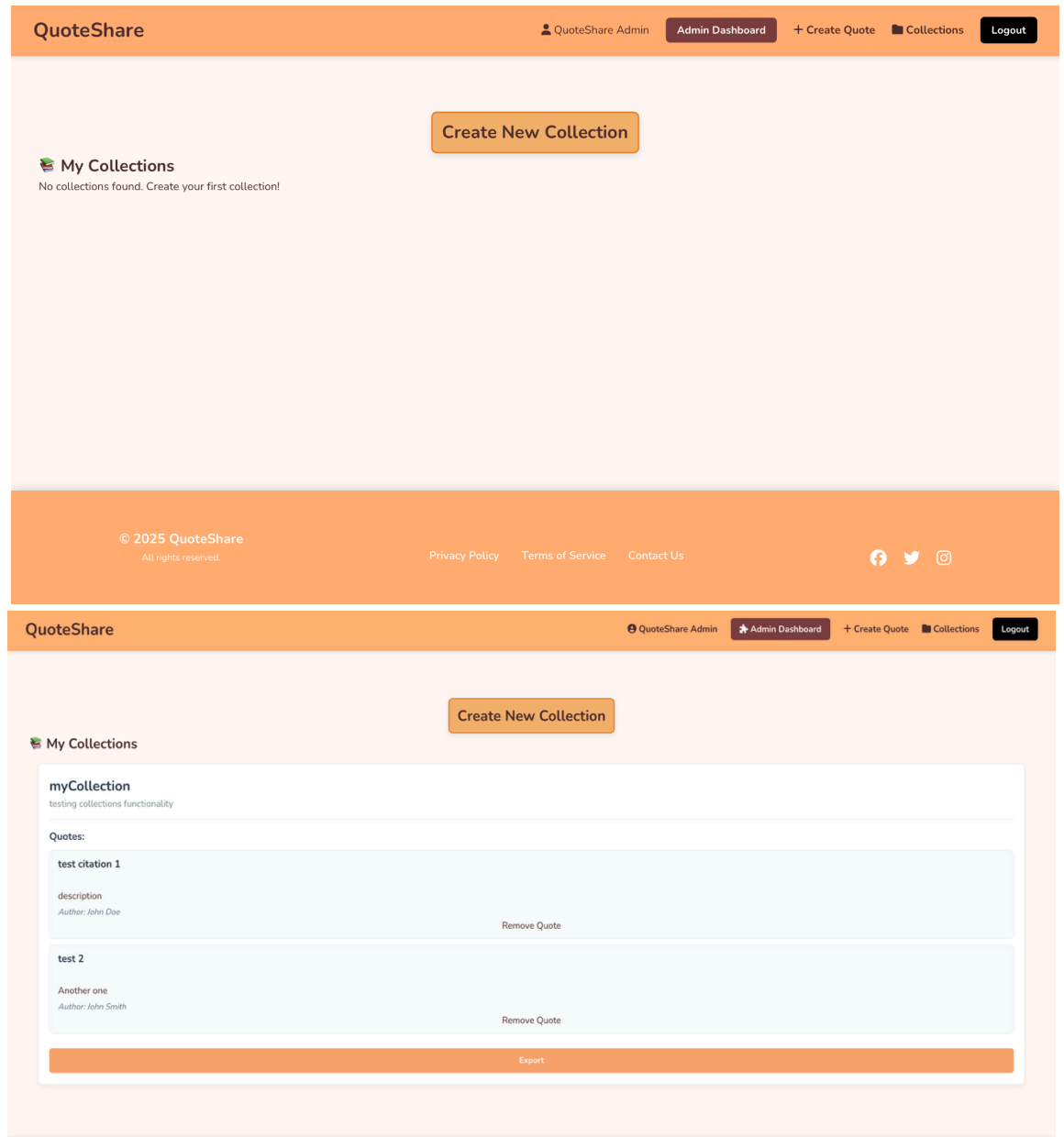


- Създаване на колекция

The screenshot shows the 'Create Collection' page on the QuoteShare website. The header is orange and contains the 'QuoteShare' logo, a user profile icon labeled 'QuoteShare Admin', an 'Admin Dashboard' button, a '+ Create Quote' button, a 'Collections' link with a folder icon, and a 'Logout' button. The main content area has a light orange background and is titled 'Organize your favorite quotes into a collection'. It features two text input fields: 'Collection Name' with a placeholder 'Give your collection a name' and a character count '0/255', and 'Collection Description' with a placeholder 'Add a description for the collection.'. Below these fields is an orange 'Create Collection' button. The footer is orange and contains copyright information '© 2025 QuoteShare All rights reserved.', links for 'Privacy Policy', 'Terms of Service', and 'Contact Us', and social media icons for Facebook, Twitter, and Instagram.

- За създаването на колекция е необходимо потребителят да въведе име (name) и описание (description) на колекцията. След като бъде натиснат бутона за създаване на колекцията (Create Collection), ако всички данни са коректни, колекцията ще бъде създадена успешно и потребителят ще бъде препратен към страницата с колекции /collections, където може да види новосъздадената си колекция.

- Страница „Collections“



- Най-отгоре на страницата се вижда бутонът „Create Collection“ за създаване на колекция, след натискането му потребителят ще бъде препратен към страницата /collections/create, и може да създаде колекция по начина описан в предишната точка.
- В секцията „My collections“ могат да се видят всички колекции на текущия потребител като всяка колекция съдържа име, описание и цитати, които са добавени.

- Потребителят може да премахва цитат от колекцията чрез натискане бутона „Remove Quote“, който се намира под съответния цитат.
- Всяка колекция може да се експортира в PDF, CSV, HTML или BibTeX след натискане на бутона „Export“, който се намира под съответната колекция.

Test Collection

Test Collection

Quotes:

- **Test title**
Test content
Author: Test author

myCollection		
Title	Content	Author
test citation 1	description	John Doe
test 2	Another one	John Smith

myCollection

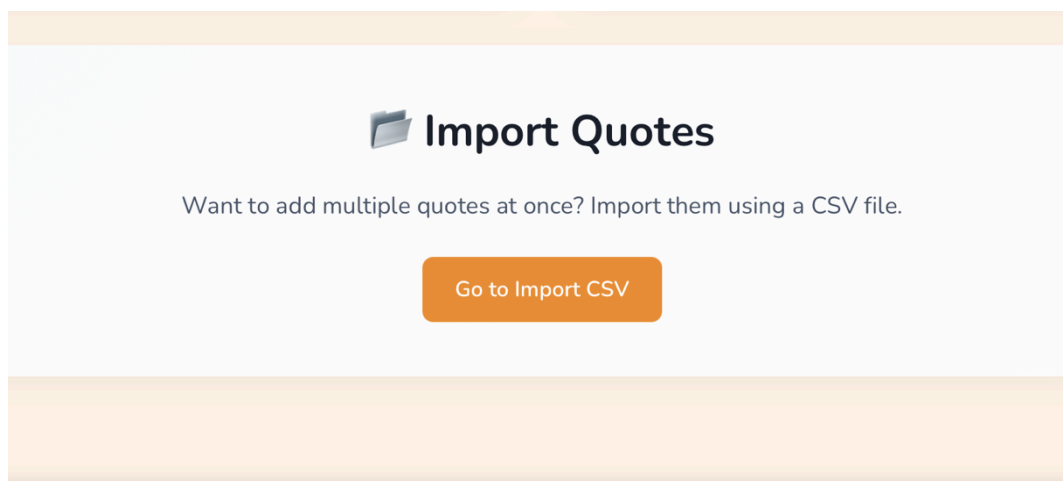
testing collections functionality

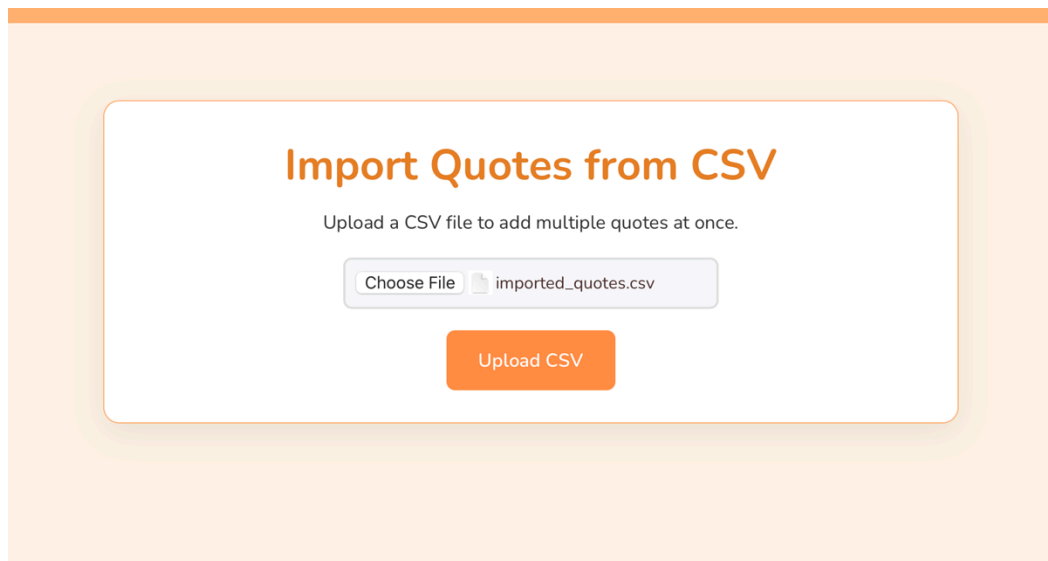
Quotes:

- **test citation 1**
description
Author: John Doe
- **test 2**
Another one
Author: John Smith

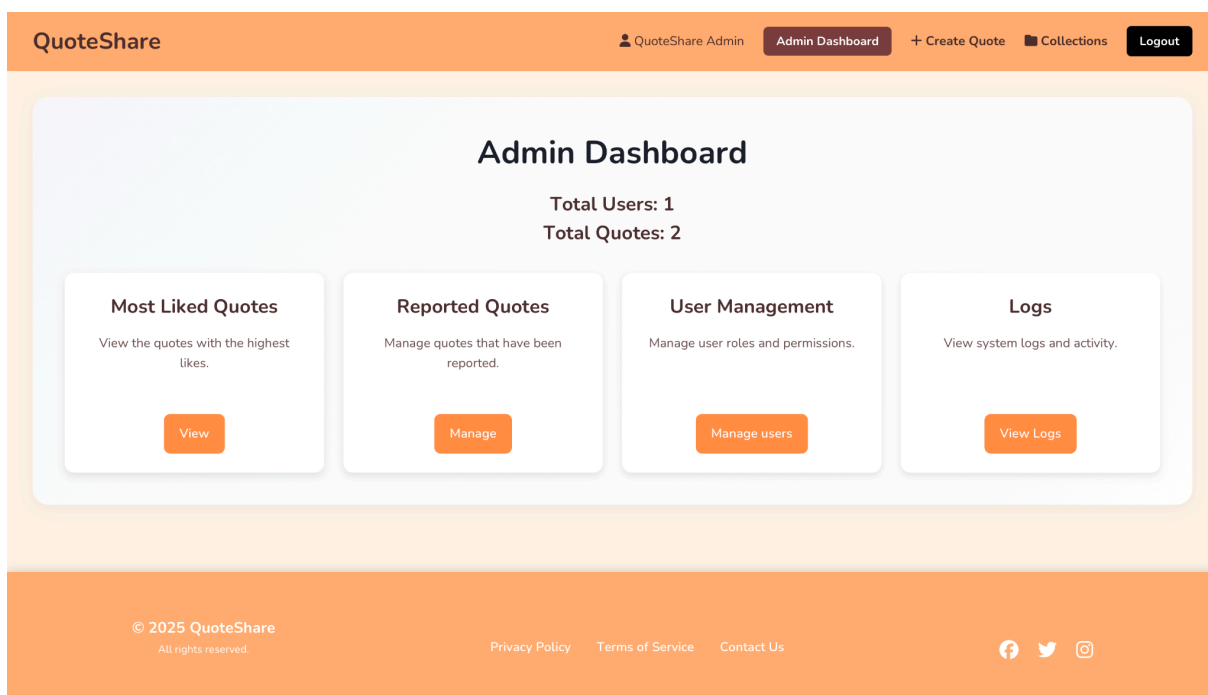
```
@misc{quote1,  
  title = {test citation 1},  
  note = {description},  
  author = {John Doe}  
}  
  
@misc{quote2,  
  title = {test 2},  
  note = {Another one},  
  author = {John Smith}  
}
```

- Имортиране на цитати от CSV файл
 - На home страницата имаме секция, от която можем да импортираме цитати от CSV файл. CSV файлът трябва да съдържа 3 колони (title, content, author).



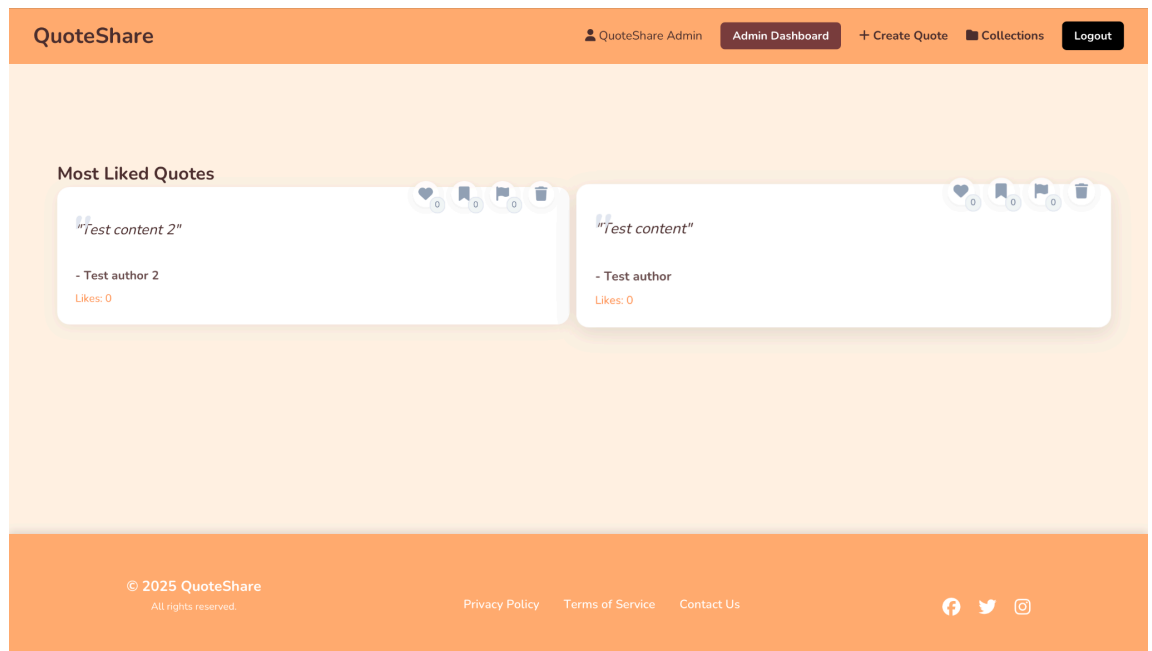


- При качване на валиден CSV файл, и натискане на бутона “Upload CSV”, цитатите биват създадени автоматично. Това позволява бързо създаване на множество цитати наведнъж.
- Админ панел - начална страница

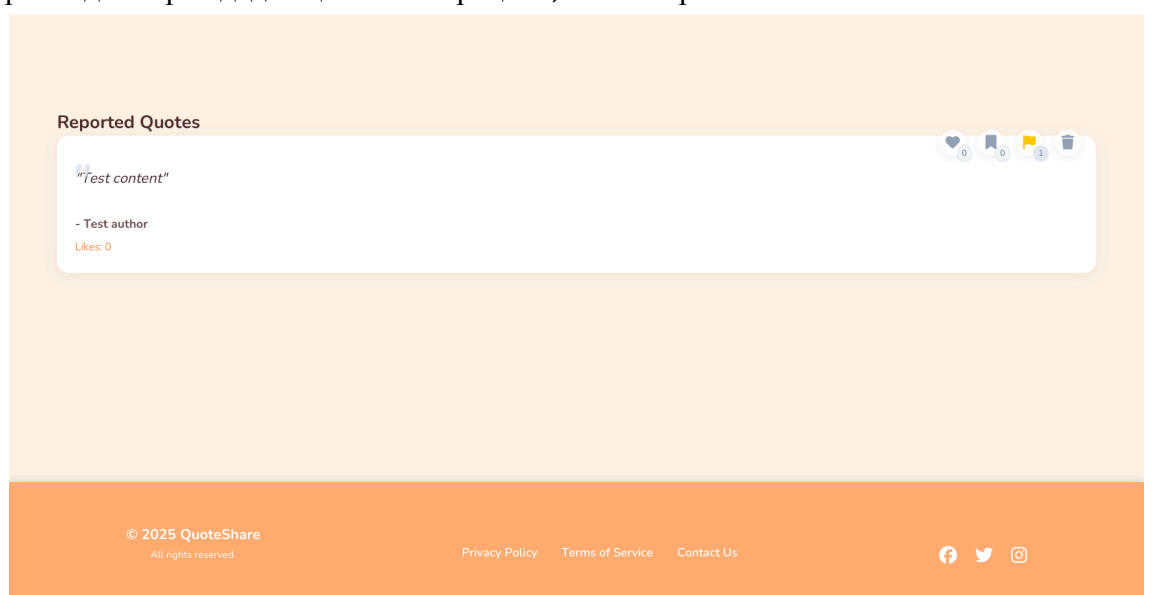


- Админ панелът може да бъде достъпен единствено от потребители, които имат ролята “admin”.

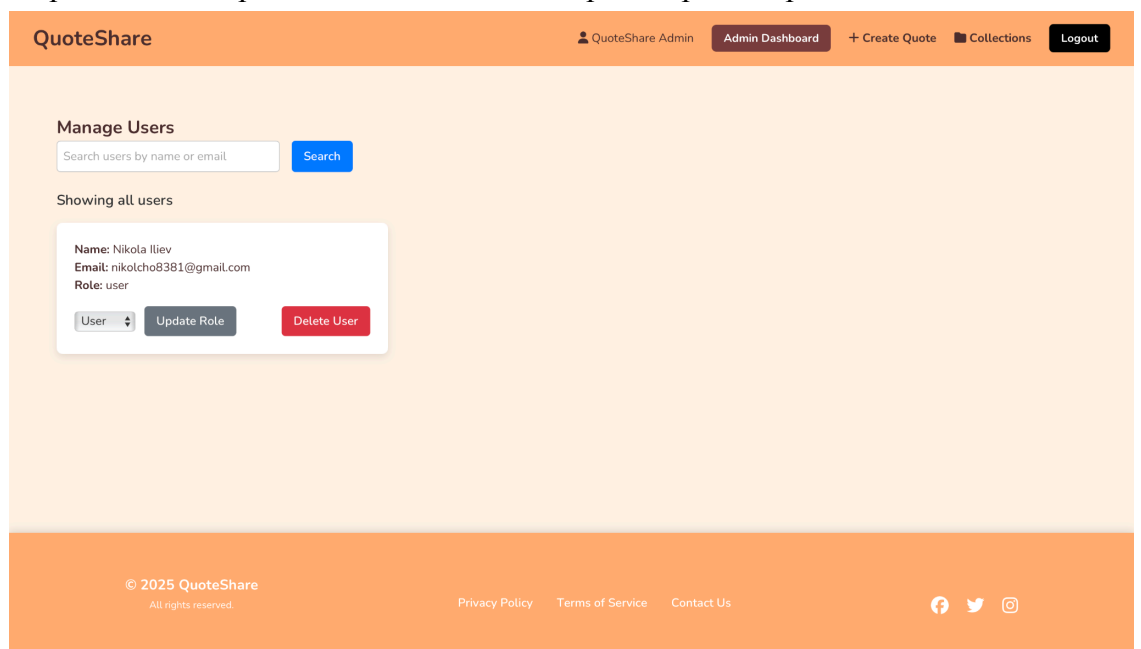
- В админ панела се показва общия брой потребители общия брой цитати в системата.
- Отдолу имаме четири секции - най-харесвани цитати, докладвани цитати, управление на потребители, логове.
- Админ панел - най-харесвани цитати
 - В тази страница показваме най-харесвани цитати, сортирани по брой харесвания



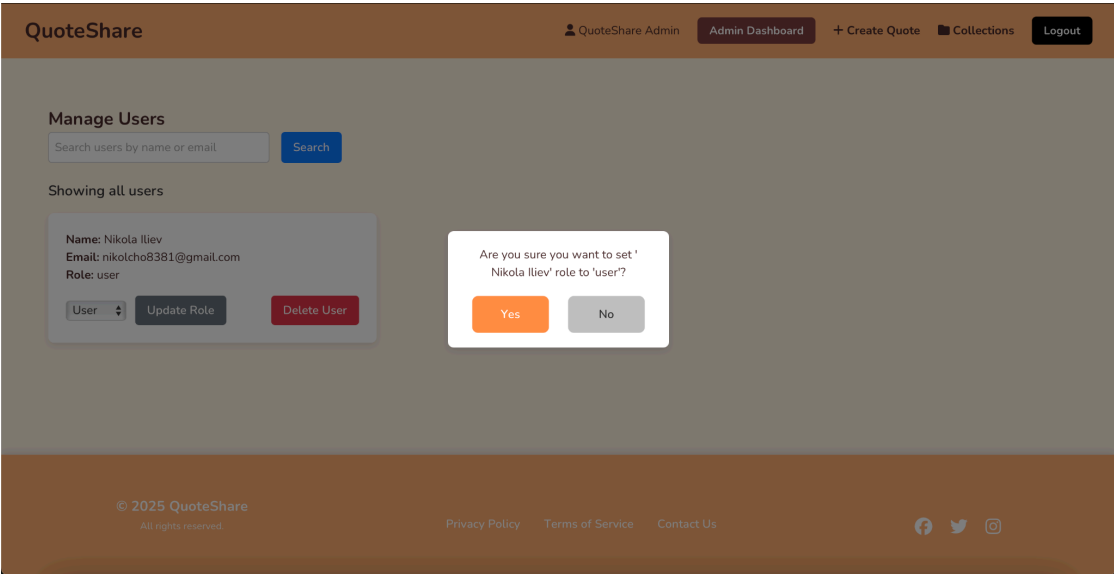
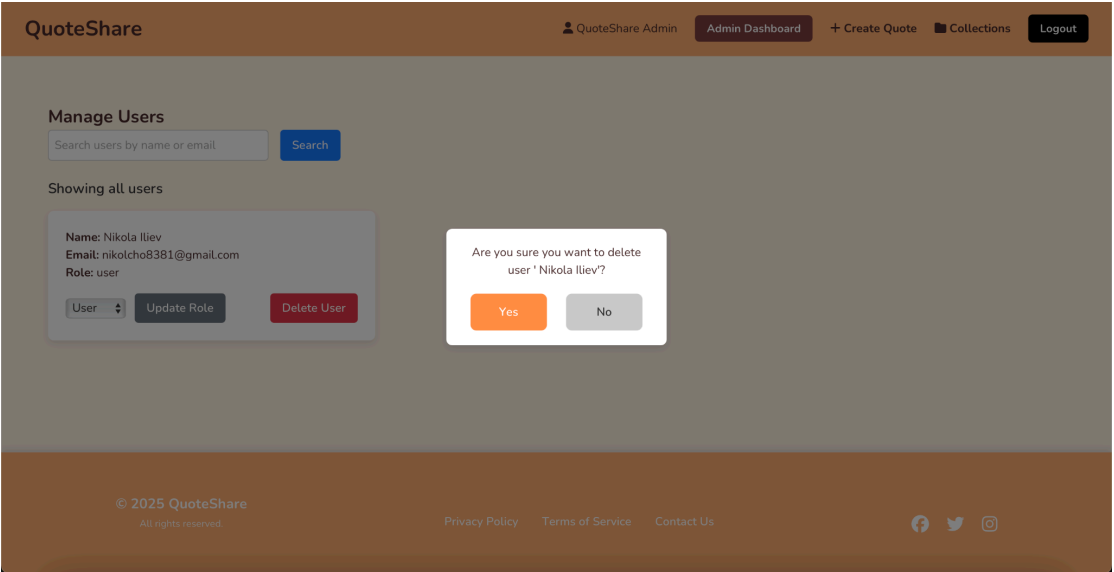
- Админ панел - докладвани цитати
 - В тази страница показваме всички докладвани цитати и админът може да реши да изтрие даден цитат ако прецени, че има причина.



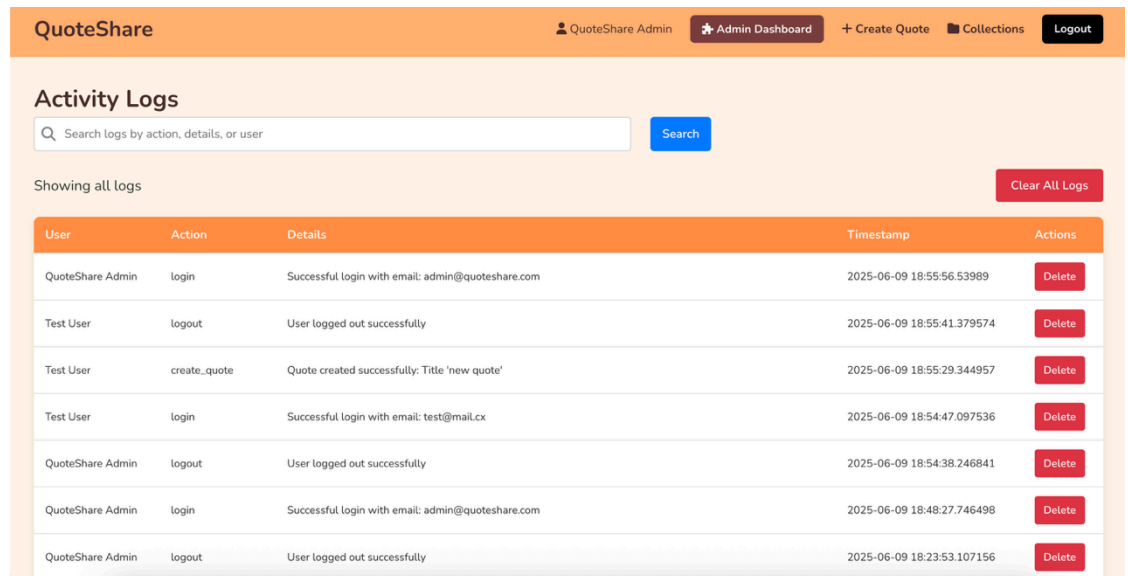
- Админ панел - управление на потребители
 - В тази страница показваме всички регистрирани потребители в системата. Можем да изтриваме даден акаунт и да модифицираме неговата роля - дали да е потребител или администратор. Също така можем да търсим потребители по име или имейл адрес. Това е много удобно, когато имаме твърде много потребители и искаме да се ориентираме бързо.



- Когато изтриваме потребител или променяме неговата роля системата отваря прозорец за потвърждение:



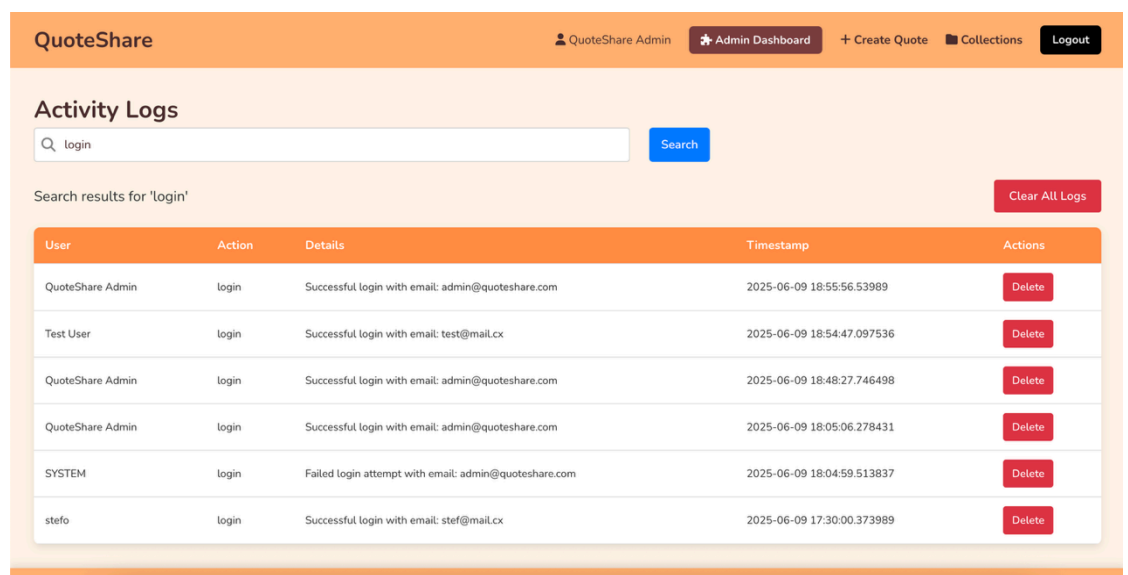
- Админ панел – логове на дейностите
 - В тази страница показваме всички дейности, които са били изпълнени в нашата система. За всяка дейност получаваме информация от кой потребител е била извършена, тип на дейността, описание и време на извършване.



The screenshot shows the 'Activity Logs' page in the QuoteShare Admin Dashboard. At the top, there's a navigation bar with 'QuoteShare Admin', 'Admin Dashboard', '+ Create Quote', 'Collections', and 'Logout'. Below the navigation bar, the 'Activity Logs' section has a search input field with the placeholder 'Search logs by action, details, or user' and a 'Search' button. Below the search bar, it says 'Showing all logs' and a 'Clear All Logs' button. The main content is a table with 5 columns: User, Action, Details, Timestamp, and Actions. The table contains 7 rows of log entries.

User	Action	Details	Timestamp	Actions
QuoteShare Admin	login	Successful login with email: admin@quoteshare.com	2025-06-09 18:55:56.53989	Delete
Test User	logout	User logged out successfully	2025-06-09 18:55:41.379574	Delete
Test User	create_quote	Quote created successfully: Title 'new quote'	2025-06-09 18:55:29.344957	Delete
Test User	login	Successful login with email: test@mail.cx	2025-06-09 18:54:47.097536	Delete
QuoteShare Admin	logout	User logged out successfully	2025-06-09 18:54:38.246841	Delete
QuoteShare Admin	login	Successful login with email: admin@quoteshare.com	2025-06-09 18:48:27.746498	Delete
QuoteShare Admin	logout	User logged out successfully	2025-06-09 18:23:53.107156	Delete

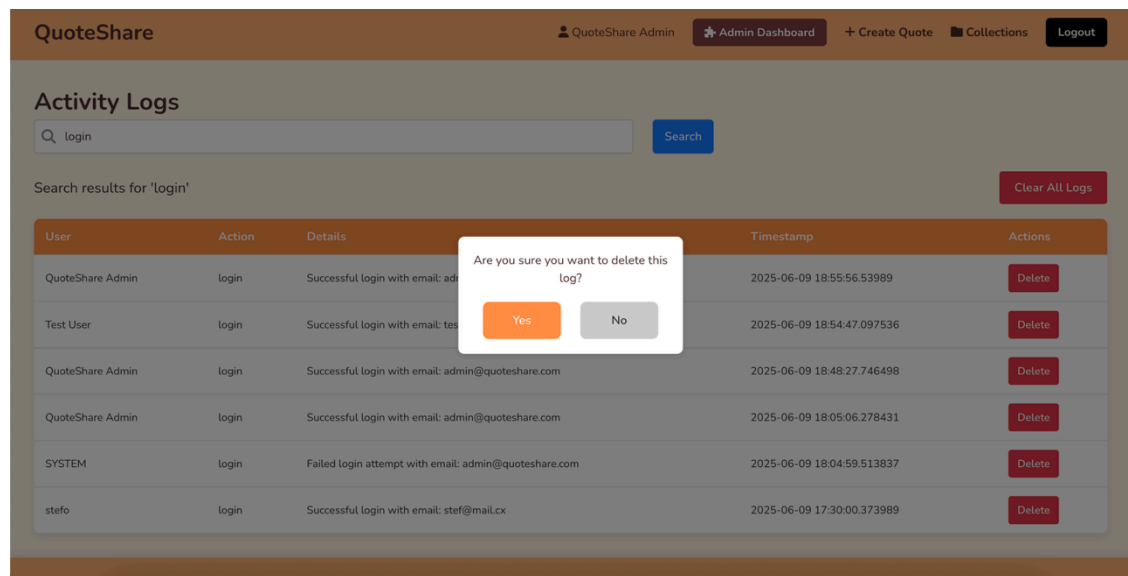
- Аналогично на потребителите, тук отново можем да търсим (филтрираме), използвайки търсачката над таблицата с логовете.



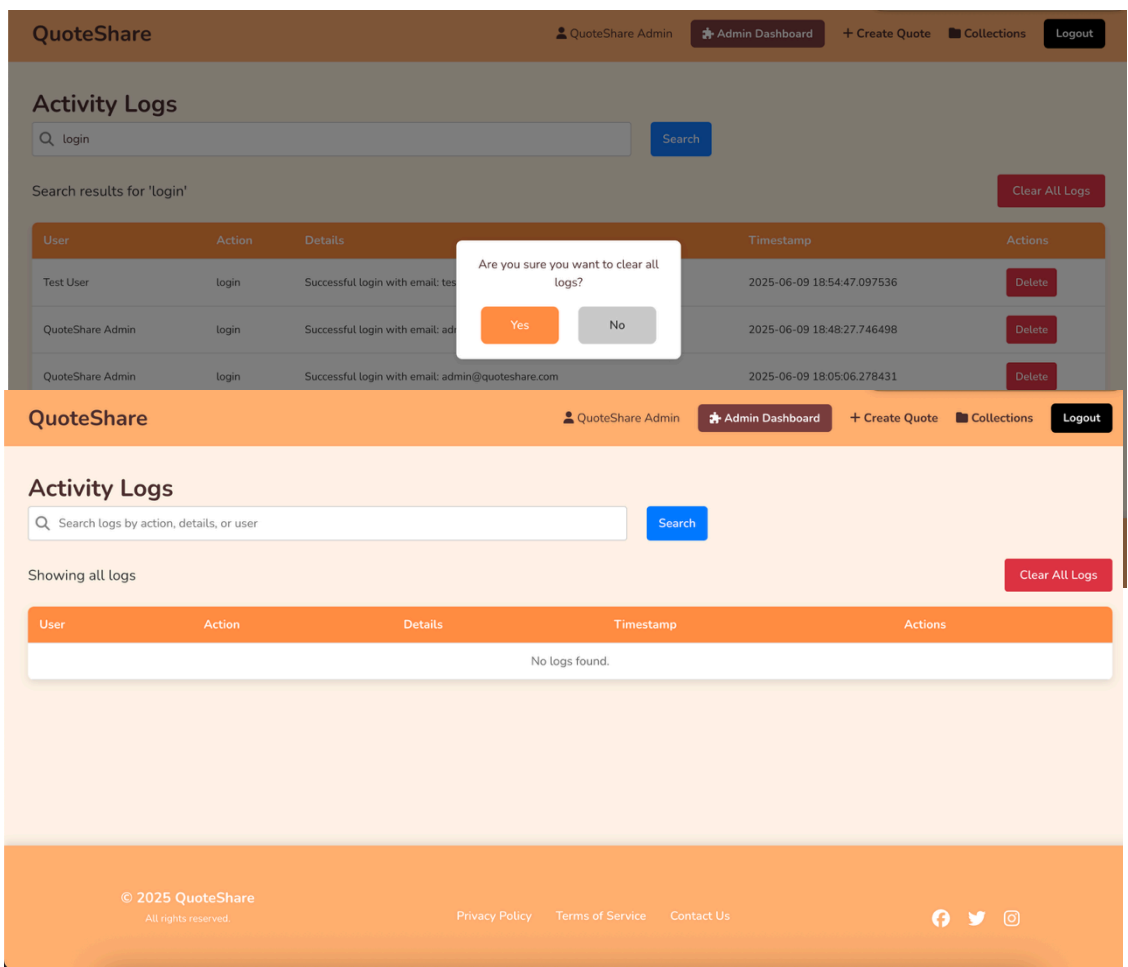
The screenshot shows the 'Activity Logs' page in the QuoteShare Admin Dashboard with the search filter 'login' applied. The search bar contains 'login' and the 'Search' button is active. Below the search bar, it says 'Search results for 'login'' and a 'Clear All Logs' button. The table now displays only the login-related entries, which are 6 in total.

User	Action	Details	Timestamp	Actions
QuoteShare Admin	login	Successful login with email: admin@quoteshare.com	2025-06-09 18:55:56.53989	Delete
Test User	login	Successful login with email: test@mail.cx	2025-06-09 18:54:47.097536	Delete
QuoteShare Admin	login	Successful login with email: admin@quoteshare.com	2025-06-09 18:48:27.746498	Delete
QuoteShare Admin	login	Successful login with email: admin@quoteshare.com	2025-06-09 18:05:06.278431	Delete
SYSTEM	login	Failed login attempt with email: admin@quoteshare.com	2025-06-09 18:04:59.513837	Delete
stefo	login	Successful login with email: stef@mail.cx	2025-06-09 17:30:00.373989	Delete

- При опит за изтриване на лог, системата изкарва диалог за потвърждение. При потвърждение, логът бива изтрит.



- Системата предоставя и опция за изчистване на всички логове



- Споделяне на цитат
 - Натискайки бутона за споделяне на цитат, системата изкарва прозорец, с опции за копиране към clipboard, споделяне към Facebook, Twitter или e-mail.



- Споделяне в Twitter (X):

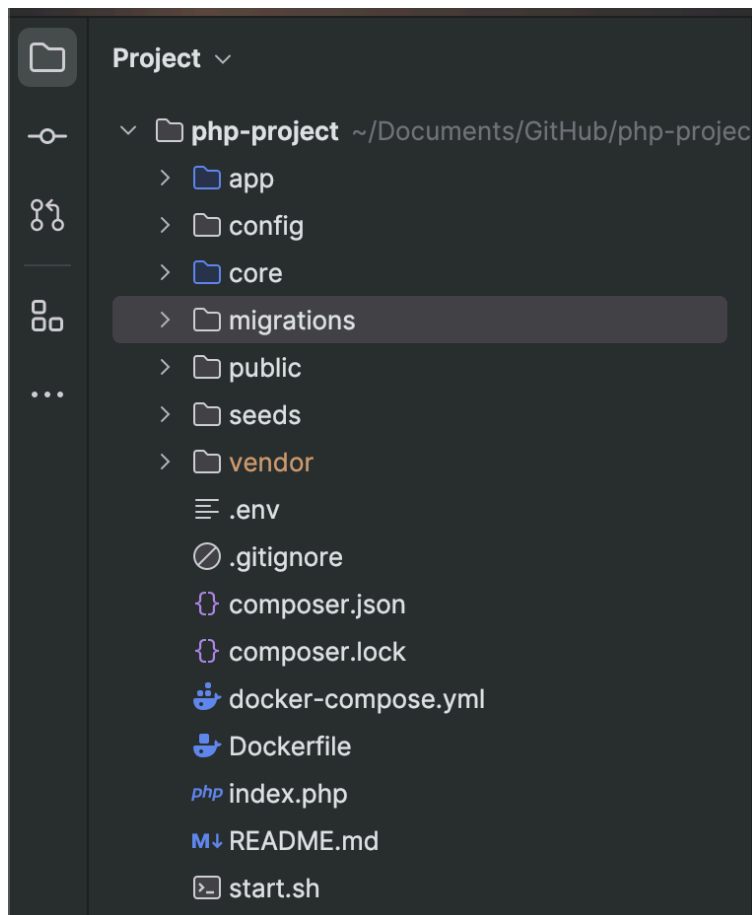


7. Примерни данни

Акаунт за администратор, генериран при стартиране на системата:

- потребителско име: admin@quoteshare.com
- парола: admin123

8. Описание на програмния код



- **app:** съдържа моделите, контролерите, изгледите на проекта и middleware, който защитава частни ресурси - изисква автентикация/сесия.
 - **Controllers:** AdminController, HomeController, UserController, CollectionController, QuoteController и т.н. Всеки контролер има в себе си метод, който бива извикан, когато се зареди даден URL адрес и се използва специфичен HTTP method (GET/POST/DELETE/PATCH).
 - **Middlewares:** AuthMiddleware, AdminMiddleware: Съдържа логиката, която се изпълнява преди зареждането на всяка заявка. Ако дадената заявка/ресурс е защитена с този middleware и ако потребителят не е вписан в системата ще възникне грешка за потребителя и ще бъде пренасочен към страницата за вход. AdminMiddleware проверява дали потребителят изпращащ заявката има администраторски права.
 - **Models:** UserModel/QuoteModel/CollectionModel/LogModel. В себе си съдържат функции, които общуват с базата дани (четене/създаване/триене и т.н.)
 - **Views:** Съдържа php файлове, които са самите изгледи, които се зареждат на даден адрес. Всеки контролер може да зареди даден изглед от тази папка. Тъй като футърът и навигацията са общи сме създали папка /partials, където имаме nav.php и footer.php. Те биват импортвани от

различни страници. Така ако променим навигацията или футъра - ще бъде направена промяната и във всички други страници автоматично.

- **config:** съдържа файл, в който лесно конфигурираме уеб приложението какви параметри за използва при създаване на връзка с базите данни
- **core:** главни класове, които са необходими за работата на приложението
 - **App:** Стартовата точка на приложението, която извиква Router и обработва глобални грешки.
 - **Router:** Отговаря за маршрутирането на HTTP заявки към съответните контролери и методи. Поддържа GET/POST маршрути, динамични параметри и middleware-и.
 - **Request:** Абстрахира входната заявка – метод, път, query параметри, body, параметри от URL и достъп до сесията.
 - **Response:** Използва се за изграждане на отговор – текст, JSON, HTTP статус, пренасочване или зареждане на view файл.
 - **Middleware:** Абстрактен клас, който се използва за междинна логика като автентикация, логове и достъп, преди контролерът да бъде изпълнен.
 - **Database:** Управлява връзката с MariaDB базата данни и предоставя методи за изпълнение на заявки (SELECT, INSERT и др.).
 - **Session:** Обработва PHP сесиите – set, get, remove, destroy. Използва се за вход, съобщения и други състояния между заявки.
 - **Container: Dependency Injection контейнер,** който съхранява и създава нужните обекти при поискване (напр. контролери, бази и др.).
- **migrations:**
 - db.sql - съдържа всички нужни таблици, които са необходими, за да работи приложението коректно.
 - setup.sh - това е скрипт, който се извиква от start.sh и се използва, за да пусне миграциите в базата данни.
- **public/assets:**
 - съдържа всички нужни CSS файлове, които се импортват от различни страници
- **public/js:**
 - съдържа всички нужни JavaScript скриптове, които се използват от различни страници, с цел по – добро потребителско преживяване

- **seeds:**
 - съдържа скрипт, който като се пусне регистрира един нормален потребител в системата и един администратор с цел по-лесно тестване.
- **index.php:** Главният входен файл на PHP приложението. Създава dependency injection контейнер, регистрира модели, контролери и маршрути, и стартира Router-a. Това е централната точка, от която тръгва цялото приложение.

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

- **Измисляне и създаване на MVC архитектурата и структурата на проекта:** Всички заедно
- **Сетъпване на уеб сървър с база данни в XAMPP:** Никола
- **Регистрация и вход:** Никола
- **Добавяне на нов цитат:** Никола
- **Харесване, Запазване и Докладване на цитати:** Никола
- **Оправяне на проекта да работи с релативни пътища:** Никола
- **Лесно конфигуриране на проекта:** Никола
- **Pagination на логовете в админ панела:** Никола
- **Добавяне на снимка към цитат:** Никола
- **Добавяне на анотации към цитати:** Анастасия
- **Добавяне на цитати към колекция:** Анастасия
- **Създаване и анотиране на лични колекции:** Анастасия
- **Експорт на дадена колекция в PDF/HTML/CSV/BIBTEX:** Анастасия
- **Преглед на общи статистики за системата в админ панела:** Стефан
- **Преглед на всички докладвани цитати в админ панела:** Стефан
- **Изтриване на цитати през админ панела:** Стефан
- **Преглед и изтриване на логове през админ панела:** Стефан
- **Изглед на потребителски профил:** Стефан

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

Никола:

По време на разработката на проекта придобих опит в създаването на уеб приложения с помощта на MVC архитектура. Научих как се реализира регистрация и вход чрез PHP сесии, както и как се имплементират действия като харесване, запазване и докладване на съдържание. Усъвършенствах уменията си за работа с контролери, модели и изгледи.

Допълнително научих как да конфигурирам и използвам Docker, като контейнеризирах базата данни и уеб приложението, така че целият проект да може да се стартира с една команда - бързо и лесно. Това ми даде реална представа за добрите DevOps практики и изграждането на преносима и лесна за настройка среда за разработка.

Разбира се, също така съм направил проектът да бъде лесно конфигурируем и да работи в ХАМРР в произволна директория, използвайки релативни пътища. Научих се как се инсталират и използват локални библиотеки (например tcpdf за генериране на PDF-и).

Надградих уменията си по JavaScript, включително работа с fetch, асинхронни заявки, динамично обновяване на съдържание, както и HTML/CSS за изграждане на интуитивен и отзивчив потребителски интерфейс. Проектът ми помогна също така да подобра уменията си за работа в екип.

Анастасия:

Разработването на проекта ми помогна да придобия множество нови знания и практически умения. Научих се да създавам уеб приложения с използване на PHP и архитектурата MVC, която включва работа с контролери, изгледи и модели.

По време на процеса успях значително да надградя вече усвоените знания от курса по HTML, CSS и JavaScript. Проектът беше отлична възможност да обединя и приложа всички придобити умения на практика, създавайки работещо приложение, готово за реална употреба от крайни потребители.

Не на последно място, подобрих и уменията си за работа в екип. Работата по проекта изискваше постоянна комуникация и съгласуване с колегите, което допринесе за по-ефективен и успешен екипен процес.

Стефан:

По време на разработката на този PHP проект научих много практически неща, които допълниха теоретичните ми познания. За първи път използвах PHP в реален проект и ми беше изключително интересно да разбера как работи неговата структура – от изграждането на контролери и модели до работата с шаблони за изгледи. Също така се запознах с основни концепции около организацията на MVC архитектурата.

Работата по проекта ми помогна да усъвършенствам уменията си за работа в екип чрез използването на Git и GitHub – от създаване на отделни branch-ове и pull request-и до разрешаване на конфликти и водене на ясна комуникация чрез commit съобщения и

кодови ревюта. Освен това ми беше полезно да автоматизирам процеси чрез shell скриптове и SQL миграции.

Целият процес ми даде реална представа как се изгражда и поддържа пълноценна уеб система в екипна среда.

11. Използвани източници

[1]: [Как се използва Composer](#)

[2]: [Как се създава примерно MVC с Composer](#)

[3]: [Какво е PDO и защо ни е нужно?](#)

[4]: [Как се работи с MariaDB в PHP](#)

[5]: [Сесии в PHP](#)

[6]: [Как се пуска MariaDB в Docker](#)

[7]: [PHP образ в Docker](#)

Предал (подпис):

/1M10600159, Никола Илиев, СИ, 3, 3/

Предал (подпис):

/0M10600156, Анастасия Маджарова, СИ, 3, 4/

Предал (подпис):

/9M10600175, Стефан Шиваров, СИ, 3, 3/

Приел (подпис):

/проф. д-р *Милен Петров*/