## horizontal line



Генериране на галерии със снимки

21.07.2020

**─**

Александра Йовкова 62229, email: alex.yovkova@gmail.com

Елена Тупарова 62196, email: elena.tuparova@gmail.com

ФМИ 2019/2020

СИ, 3 курс

# Условие

Тема 35 (17.1): Генериране на галерии със снимки и възможност за обединяване/разделяне на галерии (merge functionality).

# Въведение

Във фокуса на настоящия проект стои обработването на изображения, извличането на техните метаданни и групирането им по различни критерии в обособени галерии (албуми). Основната идея е потребителят на системата лесно и удобно да може да класифицира своите снимки според определени признаци.

В хода на разработка на проекта бяха идентифицирани следните потребителски случаи и произтичащите от тях функционални изисквания към системата:

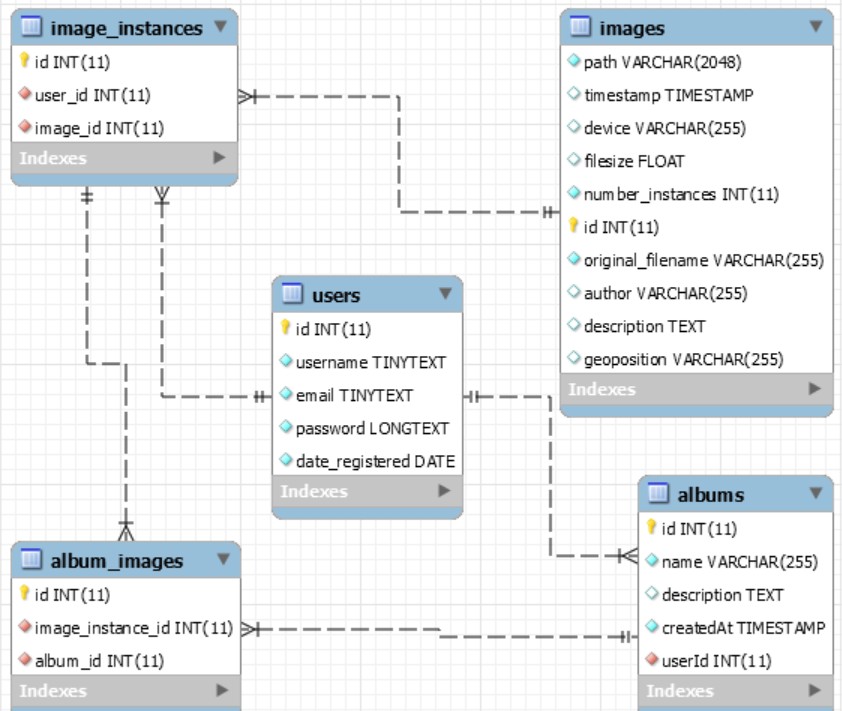
1. Потребител може да се регистрира в системата.
2. Регистриран потребител може да влезе в системата с валидни данни.
3. Регистриран потребител може да излезе от системата.
4. Регистриран потребител може да качва снимки в системата.
5. Системата може да генерира албум от снимки на база зададен от регистриран потребител времеви интервал и посочено име на албума.
6. Системата може да генерира албум от снимки, като обединява вече съществуващи албуми, посочени от потребителя. Опционално може да се посочи времеви интервал, от който да бъдат снимките в резултантния албум.
7. Регистриран потребител може да разглежда всичките си качени снимки.
8. Регистриран потребител може да вижда списък на всичките си съществуващи албуми.
9. Регистриран потребител може да разглежда снимките си в определен албум.
10. Системата може да показва на регистриран потребител спомените му от днешния ден преди Х години.
11. Регистриран потребител може да види информация за своя профил (потребителско име, имейл, дата на регистрация, дата на последно влизане в системата, брой снимки, брой албуми).

# Теория

Системата трябва да бъде реализирана като клиент-сървър. При избора на вид архитектура другият възможен вариант, който обмисляхме, беше MVC, но тъй като основният му недостатък е, че не е приложим за сравнително неголеми системи, защото води до неоправдан overhead, го отхвърлихме като решение.

За оптималност при запазване на изображенията, всяко качено изображение се съхранява точно веднъж в оригиналния си вид. За целта, при качване се хешира изображението и се проверява дали изображение със същия хеш съществува вече на сървъра. Това е постигнато като в JSON файл се държи информация във вида {хеш-на-снимката: път-до-снимката}. По този начин проверката за съществуване се извършва с константна сложност и не се допуска дублиране на изображения. Освен че пести място, този метод подсигурява и авторските права на човека, който пръв е качил снимката като автор, защото мета данните се обработват и запазват само при първото качване. Ако изображението вече съществува, се създава само нова инстанция (копие), която в базата данни се характеризира само с id на оригиналното изображение и на потребителя, който я е качил.

На Фигура 1 е представена схемата на проектираната за целите на проекта база данни.



**Фигура 1.** Схема на базата данни

# Използвани технологии

Основната технология, която е използвана в проекта, е PHP, като сървърната част на приложението е изцяло написана на езика. За клиентската част са използвани още HTML, CSS и JavaScript. Базата данни е MySQL. За обработката на мета данни и exif информация се използват специални функции, вградени в PHP.

# Инсталация и настройки

За да стартирате проекта локално, е необходимо да имате инсталиран XAMPP и стартирани Apache и MySQL сървъри в него. За да може XAMPP да прочете правилно проекта и да го стартира на localhost, е необходимо да се извърши една от следните стъпки:

## Добавяне на проекта в коренната директория на сървъра

Папката на проекта се добавя в коренната директория на XAMPP (напр. htdocs).

## Добавяне на пътя до проекта в конфигурационния файл на XAMPP (httpd.conf)

Отваряме xampp/apache/conf/httpd.conf и добавяме следните настройки:

|  |
| --- |
| **<Directory "C:/Users/alexandra/Documents/WEB-Gallery/">**  **Options Indexes FollowSymLinks**  **AllowOverride None**  **Require all granted**  **</Directory>**  **<IfModule alias\_module>**  **ScriptAlias /cgi-bin/ "${SRVROOT}/cgi-bin/"**  **Alias "/gallery" "C:/Users/alexandra/Documents/WEB-Gallery/"**  **</IfModule>** |

Съответно на мястото на `**C:/Users/alexandra/Documents/WEB-Gallery/**` се слага пътя до папката с проекта. Това създава alias в настройките, който казва на xampp да добави и съответната папка

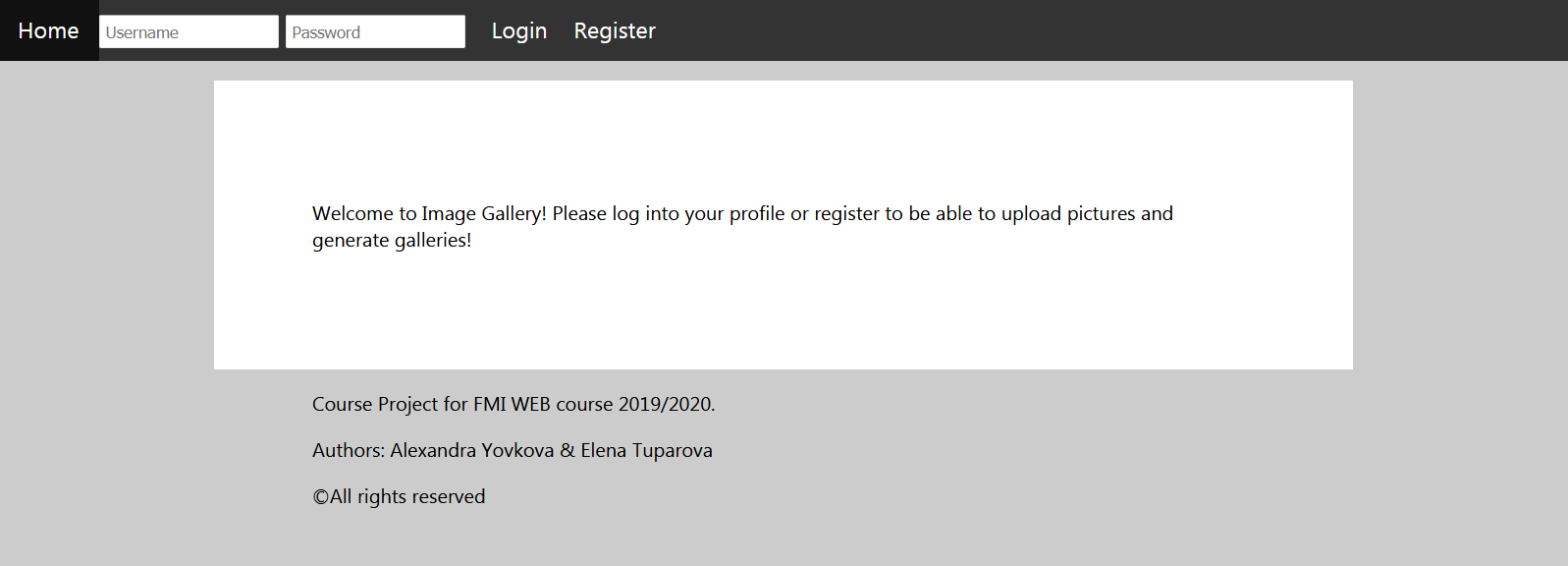
За създаване на базата данни и зареждане на тестовите данни е необходимо в конзолата на xampp/mysql/bin директорията да се изпълни web\_gallery.sql скриптът, намиращ се в главната папка на проекта, чрез следната команда (трябва да се упомене коректно текущото местоположение на скрипта спрямо bin директорията):

**mysql -u root -p < web\_gallery.sql**

След изпълнение на всички описани в тази точка стъпки, проектът може да се достъпи като в браузъра се отваря `<http://localhost/gallery/client/index.php>`.

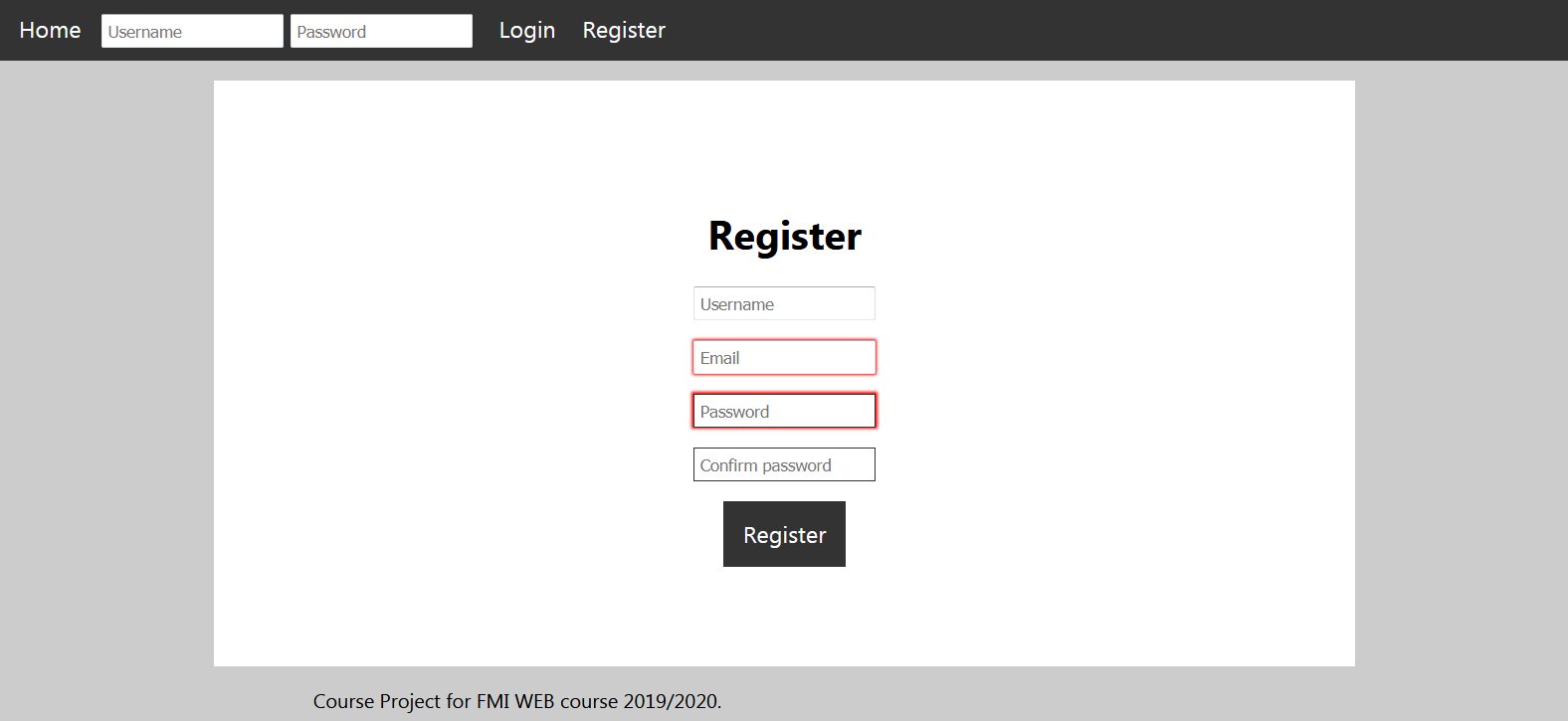
# Кратко ръководство на потребителя

При зареждане на началната страница на системата в браузъра, потребителят е посрещнат от следния екран:



**Фигура 2.** Начална страница на системата

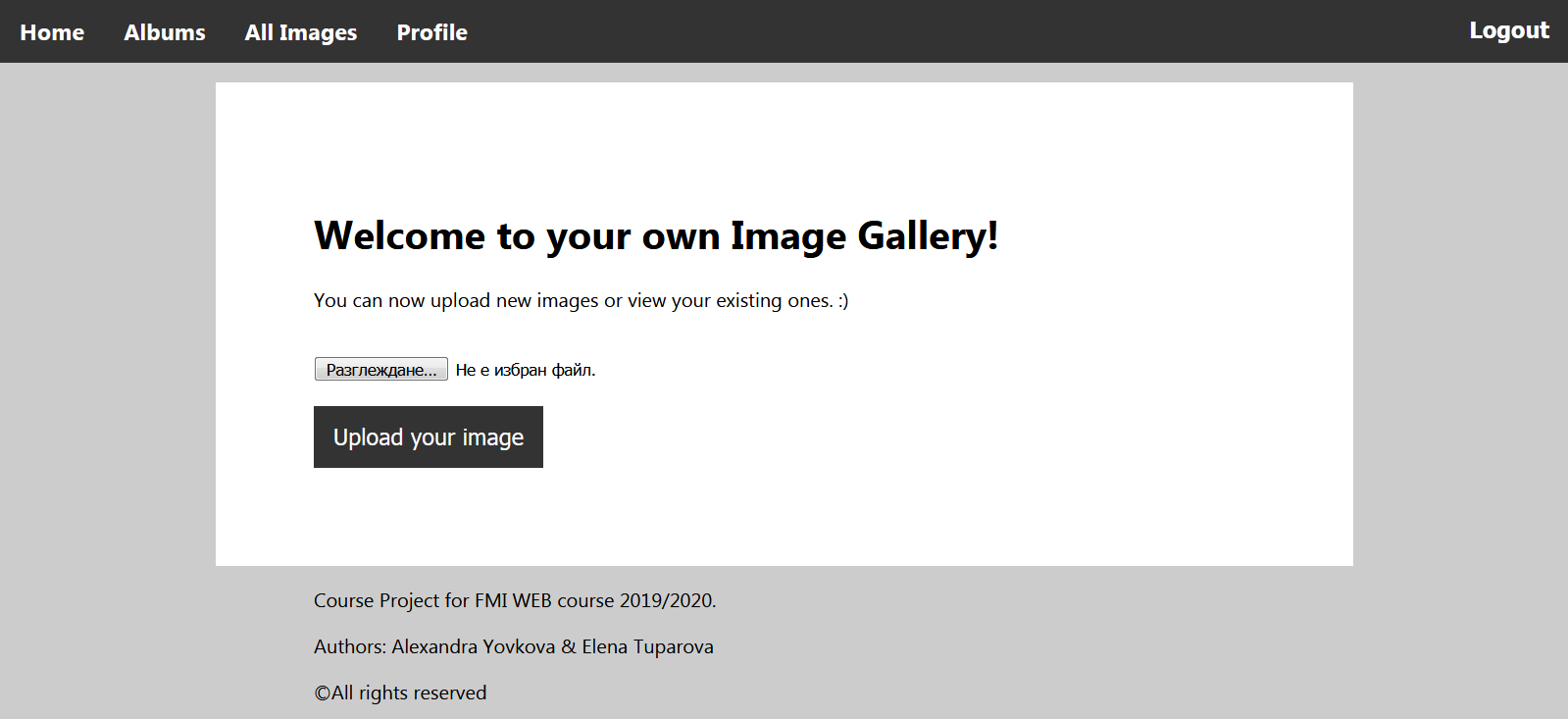
За да използва функционалностите на системата, потребителят трябва да влезе в профила си с потребителското си име и парола, които въвежда във формите в горния ляв ъгъл. Ако все още няма създаден акаунт, то може да се насочи към регистрация чрез натискане на бутона Register. Ще бъде пренасочен към следния екран:



**Фигура 3.** Форма за регистрация в системата

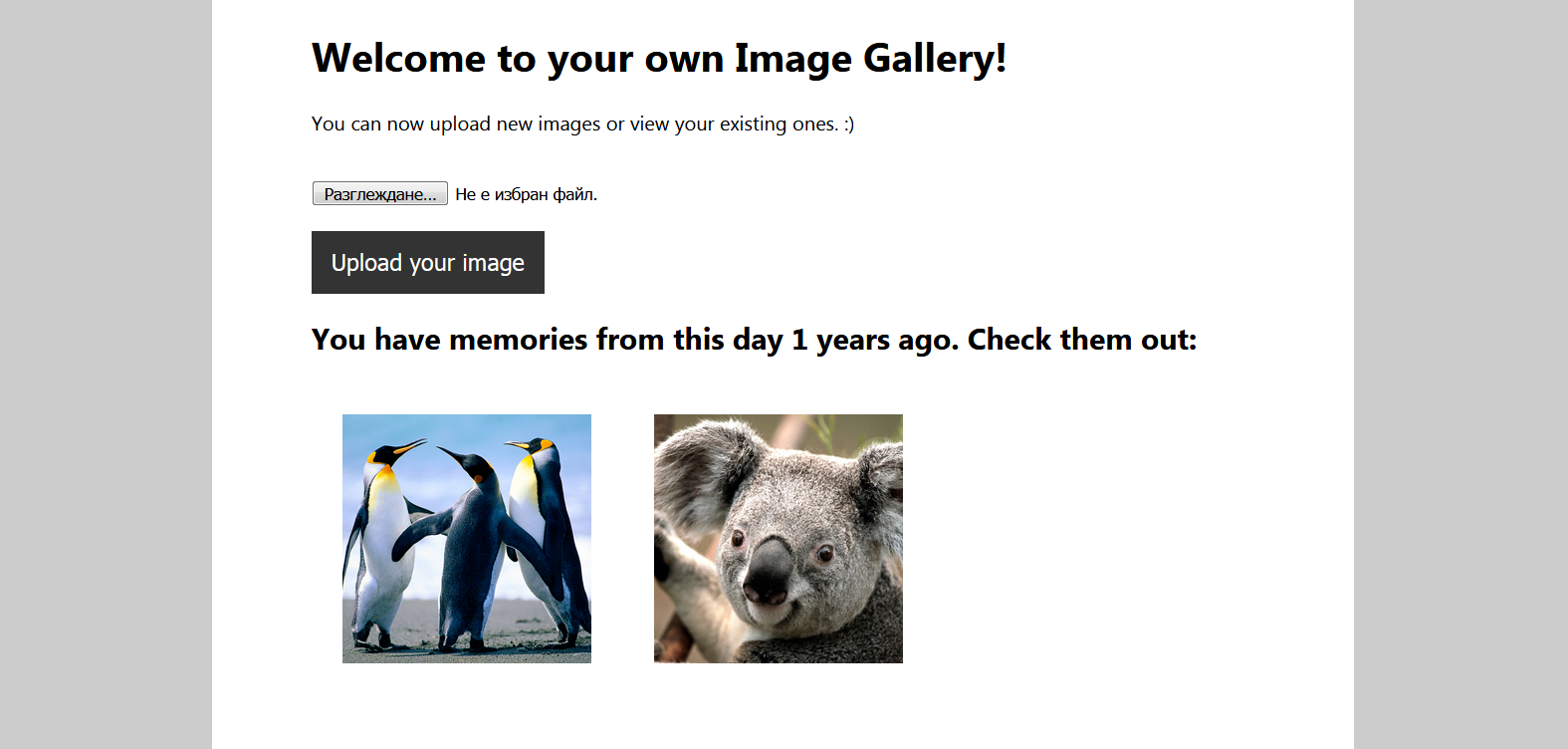
Необходимите данни за регистрация са потребителско име, имейл и парола, която трябва да бъде потвърдена. Системата не поддържа опция за смяна на паролата.

След като потребителят влезе в профила си, вече може да използва пълния набор от функционалности на системата. Началната страница изглежда по следния начин:



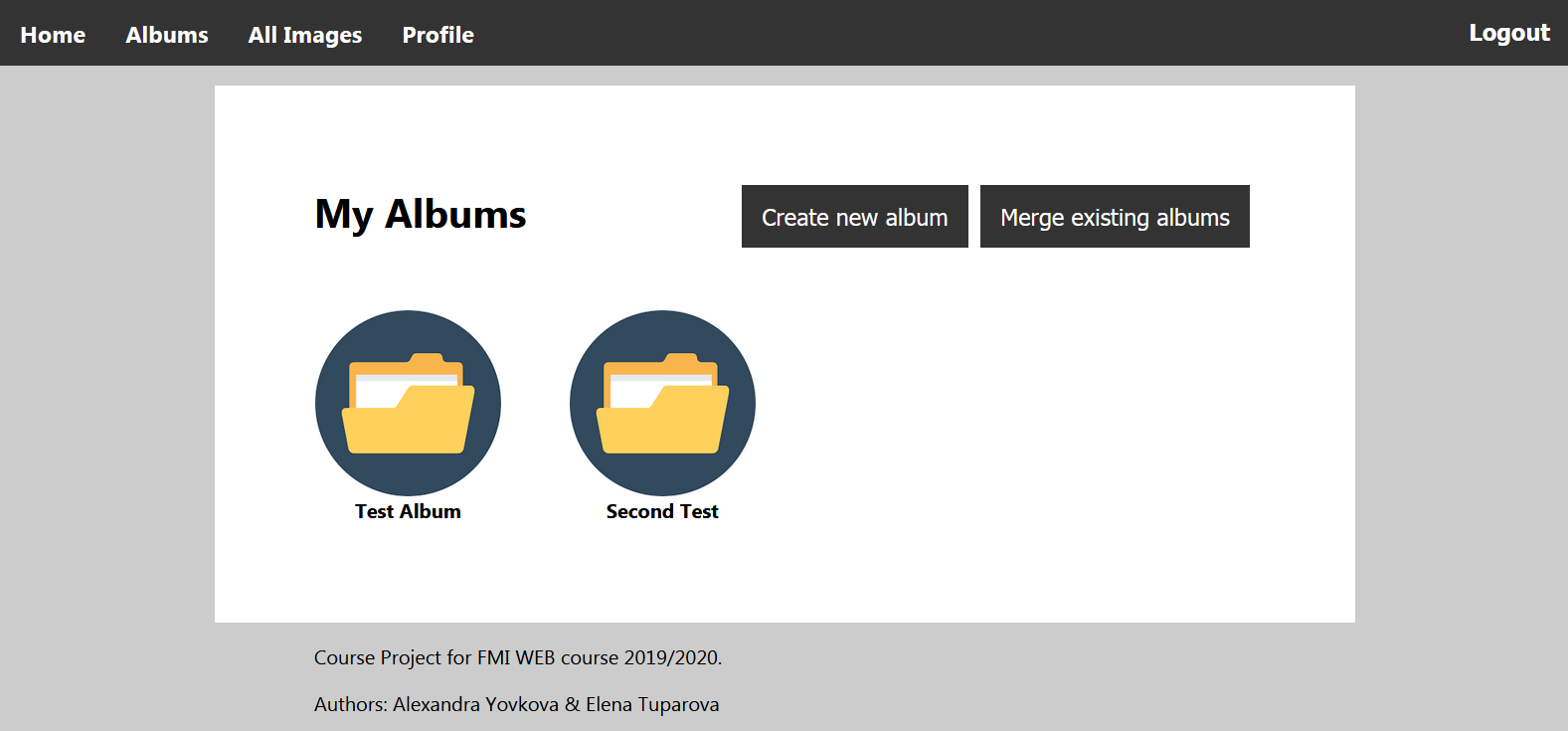
**Фигура 4.** Начална страница на системата за влязъл в профила си потребител

От нея потребителят може да качва своите снимки. Също така ако потребителят има вече качени снимки от днешния ден преди 1, 2, 3 или повече години, то той ще бъде посрещнат с тях.

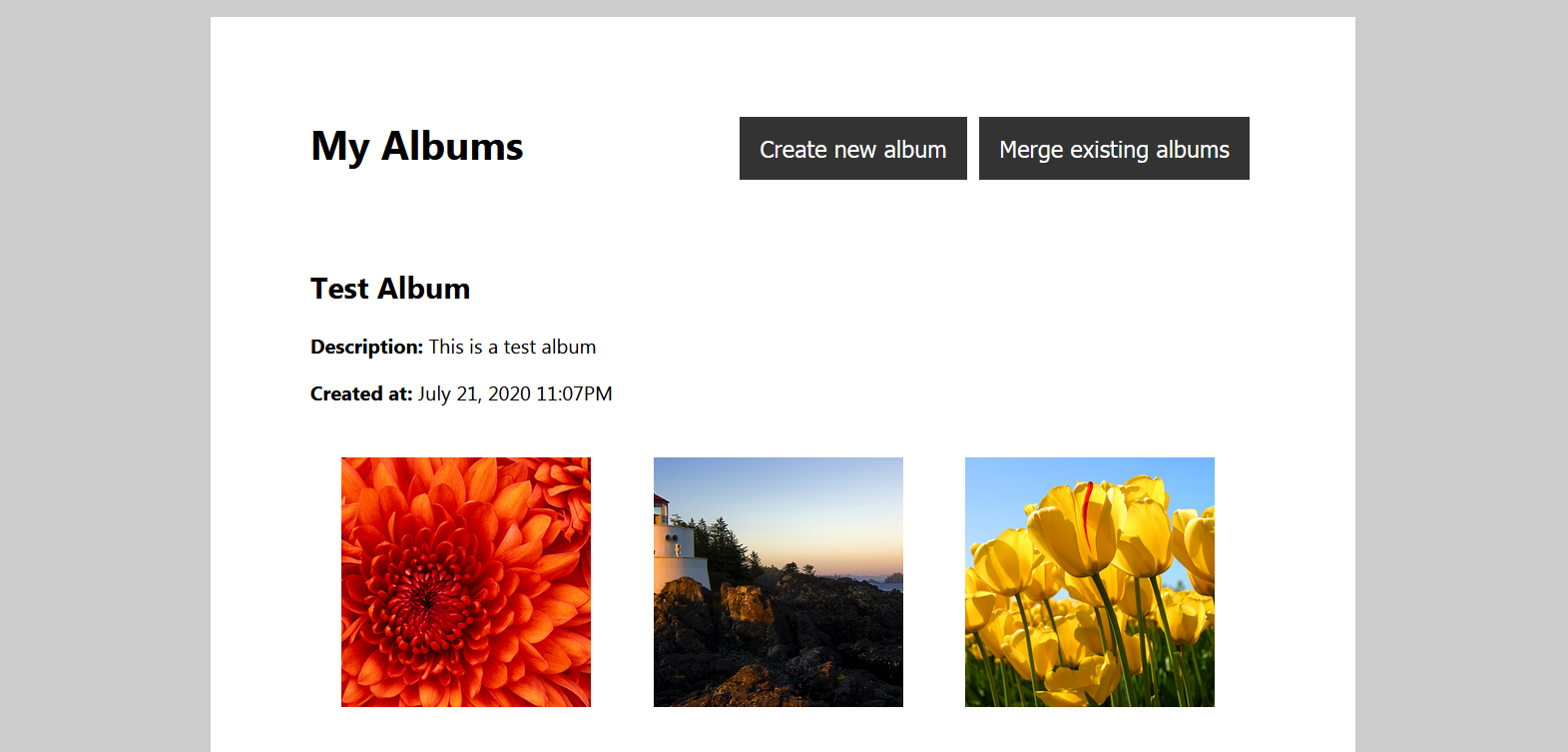


**Фигура 5.** Изглед на спомените на потребителя

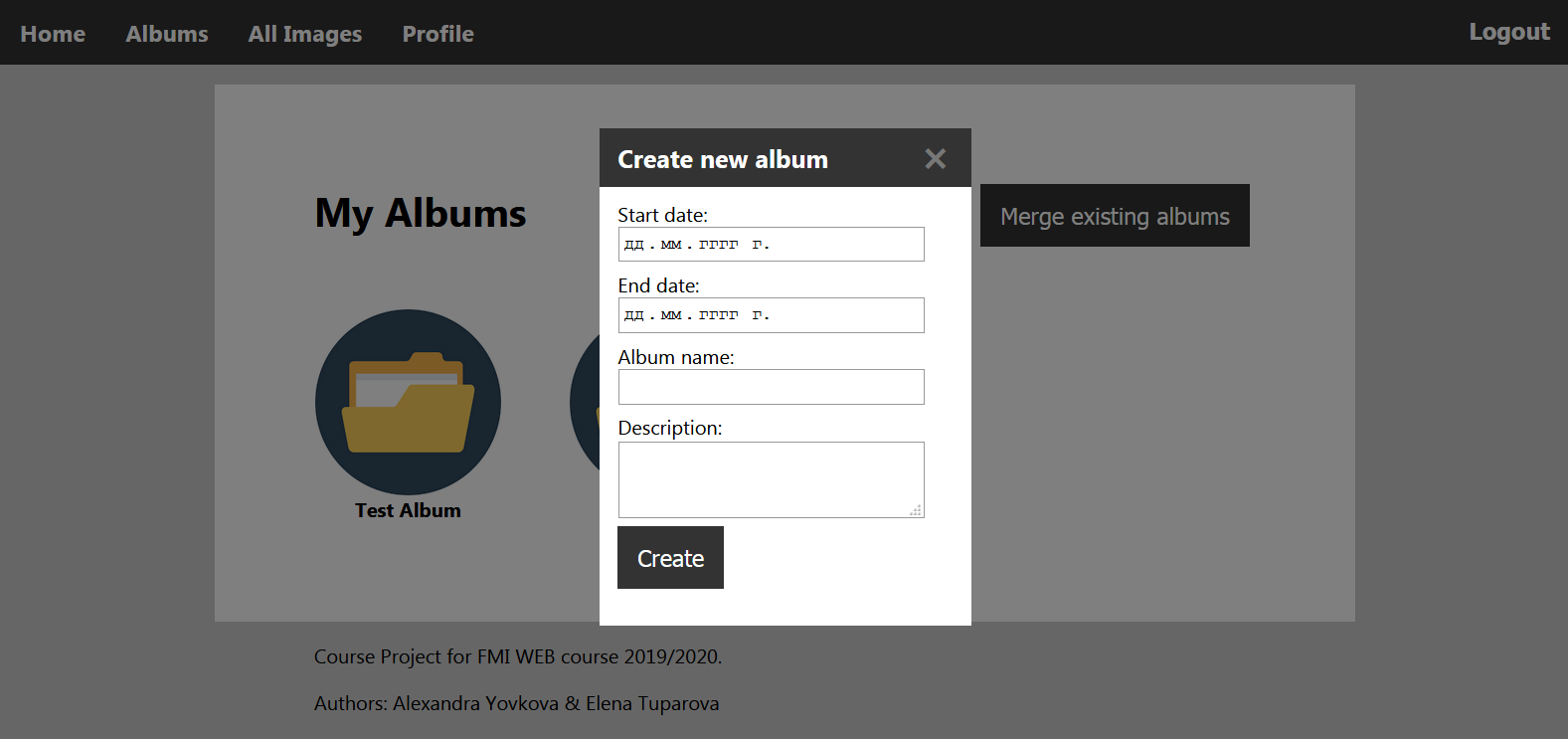
Страницата Albums предоставя на потребителя възможност да види всичките си албуми и да избере някой за разглеждане, както и да създаде нов албум или да обедини вече съществуващи в нов.



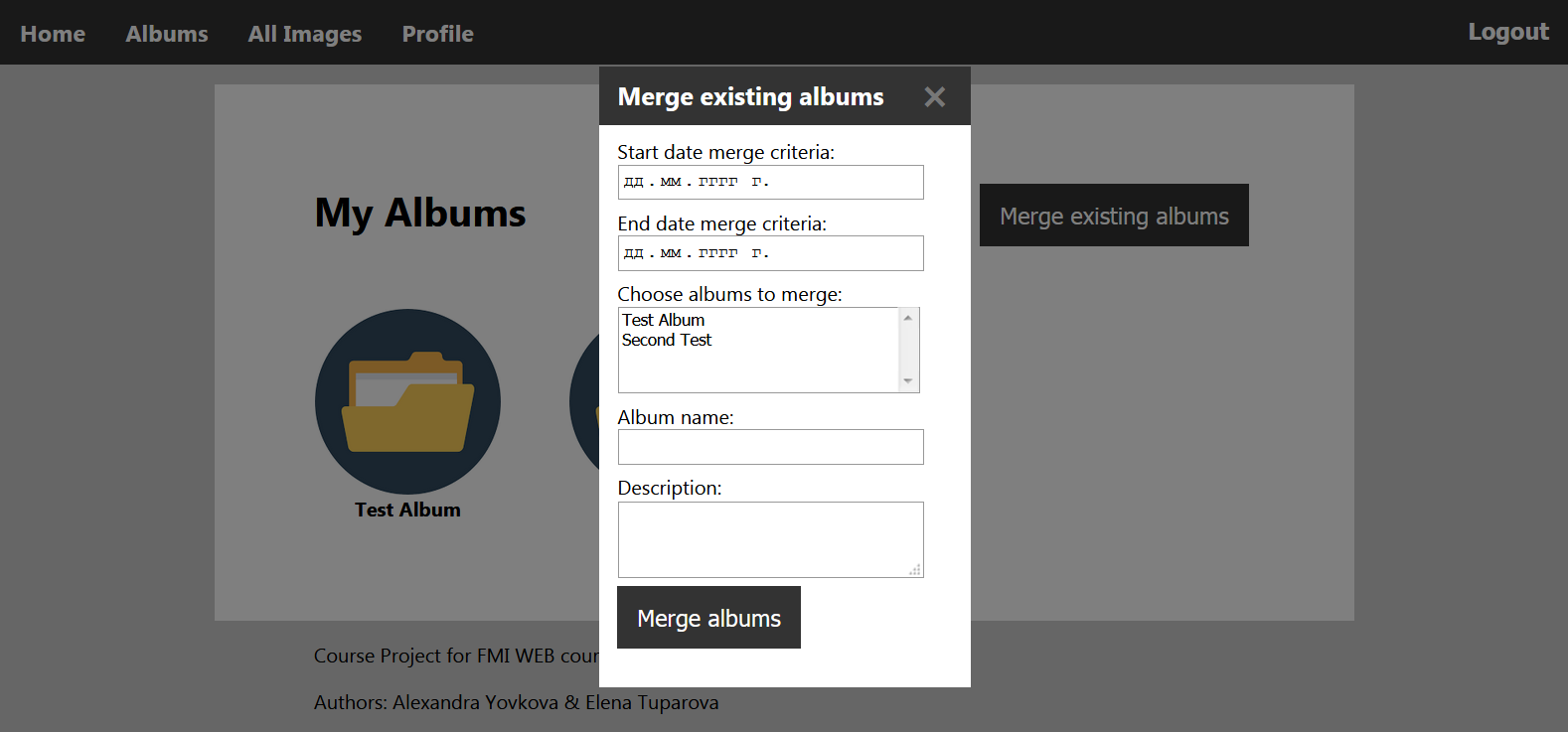
**Фигура 6.** Изглед на всички съществуващи албуми на потребителя



**Фигура 7**. Изглед на отделен албум

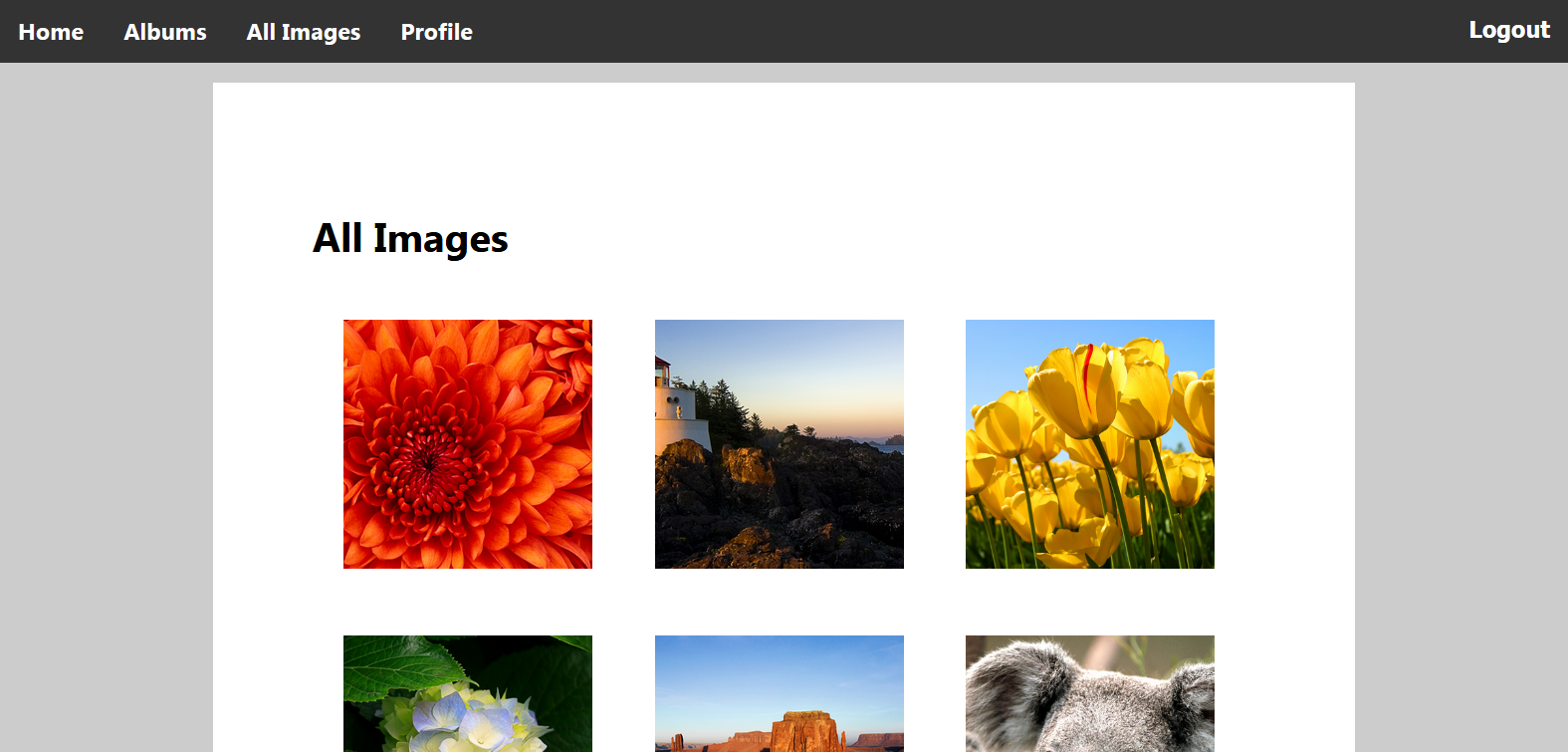


**Фигура 8.** Изглед на модалния екран за създаване на албум

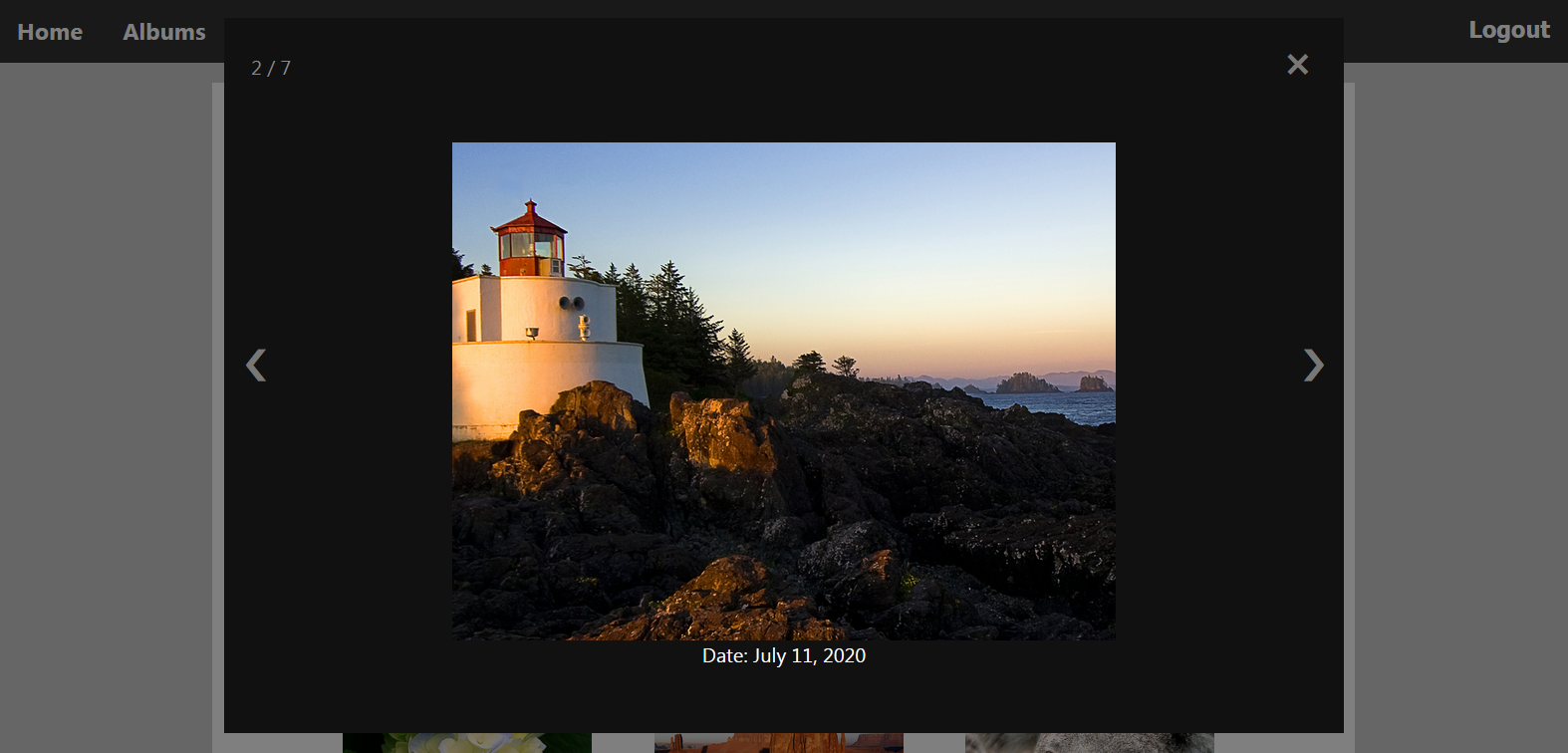


**Фигура 9.** Изглед на модалния екран за обединяване на албуми в нов албум

На страницата All Images потребителят може да види всичките си снимки, сортирани по дата. Може да ги разгледа в галерия тип “lightbox”.

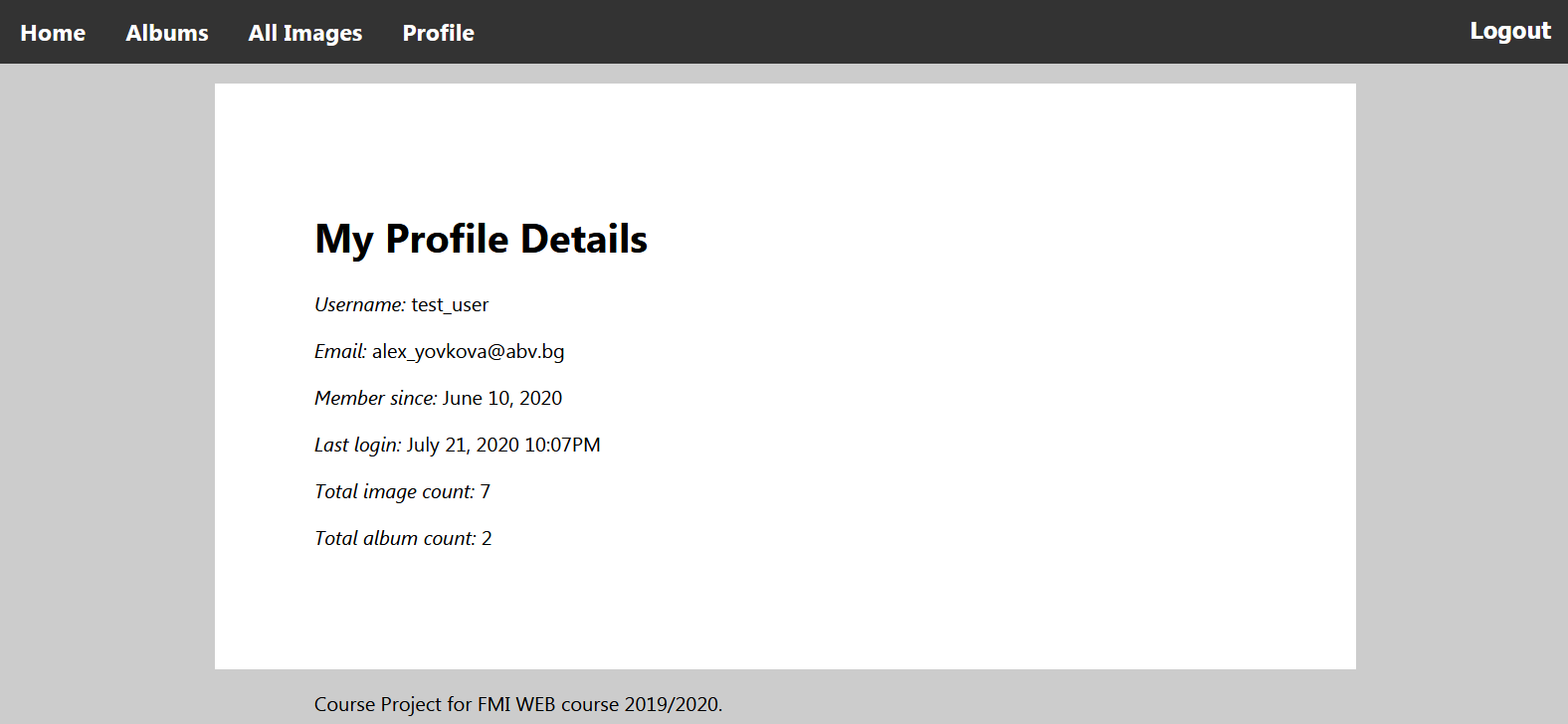


**Фигура 10.** Изглед на всички снимки на потребителя



**Фигура 11.** Изглед на снимка от галерията

В страницата Profile потребителят може да види информация за профила си.



**Фигура 12.** Изглед на детайлите за потребителския профил

# Примерни данни

Примерни данни, които могат да влязат в приложението и да бъдат обработени са всякакъв вид снимкови изображения. За оптимална функционалност е добре те да са лично заснети и непреминали през друг софтуер, който може да е заличил техните метаданни.

За илюстрация на функционалностите на системата, в web\_gallery.sql скрипта се добавят няколко снимки и албуми с подходящи за целта дати.

# Описание на програмния код

Кодът е разделен в два отделни модула - файловете в server съдържат бизнес логиката на системата (операциите по регистрация, логин, качване на изображения, създаване и обединяване на галерии), а в client се намират скриптовете, които рендерират показваните в браузера страници.

В клиентската част са всички страници, които потребителя трябва да види като форми, менюта и показване на снимки. Те са в отделни php файлове, а js и css файловете, както и изображенията, използвани за икони, са в отделни папки. За визуализация на снимките на потребителя са реализирани галерии тип “lightbox” (без използване на самата библиотека), които се отварят в модал прозорци. Създадени са модал прозорци и за формите за създаване и обединяване на албуми.

В сървърната част съответно се намира бизнес логиката - тук се обработват данните, изпратени от клиентите под формата на POST заявки, създават се албумите и т.н. За изпълнението на заданието се използват сесии, които държат id на потребителя и го използват, когато е необходимо и за да удостоверят съответно, че потребителят в момента е влязъл в профила си. POST заявките към сървърната част имат допълнителни проверки за това дали заявката е извършена посредством реалното натискане на submit бутон от формата. По този начин се избягват умишлени скриптови атаки към приложението. Също така всички заявки към базата данни са реализирани под формата на prepared statements, за да се защити системата от SQL injections. За да се запази атомарността на някои операции, които пишат в различни таблици от базата (като например създаването и обединяването на албуми), са използвани транзакции.

# Приноси на студентите, ограничения и възможности за бъдещо развитие Заедно екипът ни взе много важни архитектурни решения, основно за структурата на системата, проектирането на базата данни (в това число и процедирането по качване на изображения), идентифицирането на потребителските случаи на системата и специфицирането на изискванията към нея. Това беше първоначално осъществено на брейнсторминг сесии, по време на които дискутирахме възможностите и взехме решенията, на базата на които е изграден текущият проект. По време на същинската разработка на системата направихме още няколко сесии, на които обсъждахме промени по изискванията, както и планирахме задачите, покриващи необходимите функционалности. Колаборативната ни работа беше силно подпомогната от системата за контрол на версиите GitHub.

**Индивидуален принос:**

## Александра

Работата на студента включваше началното сетъпване на проекта, създаването на базата данни, регистрация, логин, логаут, поддържането на сесии, качване на изображения, листване на съществуващите албуми на потребителя, визуализация на данните за потребителския профил.

## Елена

Работата на студента включваше визуализация на изображенията в галерии тип “lightbox”, показване на изображения, които са качени на този ден преди Х години под формата на спомени на началната страница на потребителя, създаване на нов албум по избран времеви интервал, обединяване на съществуващи албуми в нов такъв.

Възможностите за бъдещо развитие на системата се отнасят до добавяне на още критерии за разделяне по албуми като автор, локация и други. Евентуалният проблем с тези критерии е, че дори снимките да не са минавали през система, която премахва EXIF метаданните им (като Facebook или Instagram например, които го правят с цел непроследимост на снимките, теоретично), много често за тях е налична само дата.

# Какво научихме

По време на разработването на проекта придобиваме ценен опит за проектиране и реализиране на цялостно приложение от нулата, с помощта на технологичния стек, изучаван по Web технологии, а именно PHP, HTML, CSS, JavaScript и MySQL.

Темата на проекта изискваше още да направим проучване за различните стандарти за метаданни за изображения и начините, по които можем да ги извлечем и обработим с помощта на PHP.