**Имена:**  **Бойко Красимиров Борисов** **ФН:**  ***82033*****Начална година:** ***2019*   
Програма: бакалавър, КН** **Курс: 4**   
**Тема: 11.1, Система за награди 1\*10^6   
Предмет: Web технологии КН, зимен семестър 2022/2023  
Имейл:** [**bojkob@uni-sofia.bg**](mailto:bojkob@uni-sofia.bg) **Дата: 2023-02-07**

ТЕМА: 11.1 - Система за награди 1\*10^6

## Условие

Оригинално условие на темата, върху което се основава проекта:

* *По зададени оценки – се задава количество ‚награди‘ – по различни канали и статистика*
* *Million pixel сайт – задаване на собствена картинка – според ‚приноса‘ – като администратор (и) за курс – одобряват – автоматично или след преглед; (примерна тема за участник 1)*
* *Анимирана визуализация- таймлайн за даване на наградите; (примерна тема за участник 2)*

Настоящият проект е фокусиран предимно върху реализирането на “Million pixel сайт” и спомагателни функционалности, които да предоставят възможности за управление от страна на администраторите и удобен интерфейс за използване от страна на потребителите.

## Въведение

Сайтът “[The Million Dollar Homepage](http://www.milliondollarhomepage.com/)”, създаден през 2005-та година, бързо се е превърнал в интернет феномен. Той позволява на личности да закупят определено количество пиксели, където могат да сложат собствена картина и хиперлинк. Оттогава са възникнали множество негови копия, които целят да пресъздадат успеха му. В някои случаи базовата идея е преминала и през редица вариации. Например популярният събредит [r/place](https://www.reddit.com/r/place/) надгражда идеята като добавя и допълнително ниво на интерактивност от страна на потребителите.

Сегашният проект цели да създаде версия на този уебсайт за студенти от ФМИ. Проектът има чисто развлекателен характер и се стреми да реализира лесен за използване уебсайт, в който множество студенти имат възможност да добавят съдържание към 1000x1000 пиксела картина и да виждат крайния резултат. Допълнително се реализира и ролята на администратор, който може да регулира публикуваното съдържание и да управлява списъка от студенти, които могат да участват.

## Теория

За реализирането на проекта се използва класическа трислойна архитектура.

Първият слой е презентационният слой, чиято основа е реализирана чрез HTML5, стилът на страниците е реализиран чрез CSS, а също така има и динамично съдържание, което е реализирано чрез JavaScript.

Вторият слой е бизнес слоят. За неговата реализация е използван PHP.

Третият слой е слоят за данни. Той е реализиран чрез СУРБД (система за управление на релационна база данни) MariaDB, която е вариация на често използваната MySQL. Тя се достъпва посредством phpMyAdmin. Освен това на сървърната машина има отделна директория ‘pictures’, в която се съхраняван изображения, необходими за работата на уебсайта.

## Използвани технологии

За реализирането на проекта е използвана средата за разработка XAMPP. Включените технологии и версиите, при които е изпробван проекта, са дадени по-долу:

|  |  |
| --- | --- |
| Технология | Версия |
| XAMPP | 8.1.10 |
| Система | Windows NT DESKTOP-N4NGQ0R 10.0 Build 19045 (Windows 10) AMD64 |
| Mozilla Firefox браузър | 109.0 (64-bit) |
| Apache сървър | 2.4.54 (Win64) |
| PHP | 8.1.10 |
| MariaDB | 10.4.25 |

Освен това е използвана и PHP библиотеката GD за обработка на изображения.

|  |  |
| --- | --- |
| GD библиотека | bundled (2.1.0 compatible) |

## Инсталация и настройки

## Инсталация

* + 1. За пускането на проекта е нужно да има инсталиран браузър на машината. Освен това е нужно да се инсталира XAMPP или друго приложение, позволяващо достъп до сървъра, ако все още не е инсталирано. Приложението XAMPP може да бъде инсталирано от [тук](https://www.apachefriends.org/download.html).
    2. Разархивирайте предоставения архив. В него има папка “**FN82033**”, която съдържа проекта. Можете да преименувате папката, ако искате.
    3. Поставете папката със съдържанието в директорията “**C:\xampp\htdocs**”, ако използвате ОС Windows. За Linux аналогичната директория е “**/opt/lampp/htdocs**”. При други операционни системи ще трябва да проверите къде е инсталиран XAMPP и да намерите съответната папка “**htdocs**”.
    4. Отворете приложението XAMPP и пуснете модулите Apache и MySQL. След това в браузъра отворете <http://localhost/phpmyadmin/>. От тук ще можете да управлявате базите данни, които са достъпни за XAMPP. След това натиснете бутона **Import** и заредете файла “**millionpixels.sql**”, Който се намира директно в папката, с проекта.
    5. В папка на проекта отворете файла “**configuration\serverConfig.ini**” с текстов редактор. От там можете да промените името на използваната база данни, което трябва да съвпада с това, показано в phpMyAdmin. По подразбиране то е “**millionpixels**” и не би трябвало да е нужно да променяте нещо, за да потръгне проекта. Освен това по подразбиране базата данни ще се достъпва от потребител с име “root” с празна парола, който трябва да има достъп за писане и четене от тази база данни. Ако искате да използвате друг потребител, трябва да промените съответните полета във файла.
    6. Инсталирайте библиотеката GD за PHP, ако още не е инсталирана. Тя е необходима, за да се освободят пълните функционалности на проекта. Можете да намерите подробно упътване как да се инсталира [тук](https://www.geeksforgeeks.org/how-to-install-php-gd-in-windows/) за Windows, но накратко:
       - В XAMPP панела натиснете на Admin срещу Apache модула.
       - От появилото се меню натиснете PHPInfo
       - Проверете дали имате таблица със заглавие “gd”, която съдържа информация за библиотеката. Ако да, то вече е инсталирана. Ако не, то продължете с упътването.
       - От менюто на XAMPP натиснете Config срещу Apache модула и отворете php.ini
       - Намерете “**;extension=gd” реда и изтрийте точката и запетаята в началото. След това запаметете файла.**
       - **В инсталационната папка на XAMPP намерете файла “php\ext\php\_gd.dll” и поставете негово копие в “C:\Windows\System32**”
       - Вече трябва библиотеката да е инсталирана. Можете да проверите това като отново отидете на PHPInfo и потърсите таблицата “gd”
    7. Вече сте готови да пуснете проекта. Отворете браузъра и отидете на “[**localhost/FN82033/client/**](http://localhost/FN82033/client/)” и автоматично трябва да се зареди началната страница на проекта. Ако сте преименували папката, трябва вместо “**FN82033**” да пише новото име на папката.

## Настройки

За настройката на проекта отворете файла “**configuration/serverConfig.ini**” с текстов редактор. В него ще намерите следните настойки:

* **host** – Хост, на който да се установи връзка с базата данни. По подразбиране е “**localhost**”.
* **db\_name** – Име на базата данни. По подразбиране е “millionpixels”
* **db\_type** – Вид на базата данни. По подразбиране е “mysql”
* **user** и **password** – Потребителско име и парола на потребителя в СУБД. По подразбиране е “root” и “”. Трябва съответния потребител да има право за писане и четене в съответната база данни.

Освен това във файла “**configuration/clientConfig.js**” има настойка за помощна функционалност при добавяне на нови картини от страна на участниците.

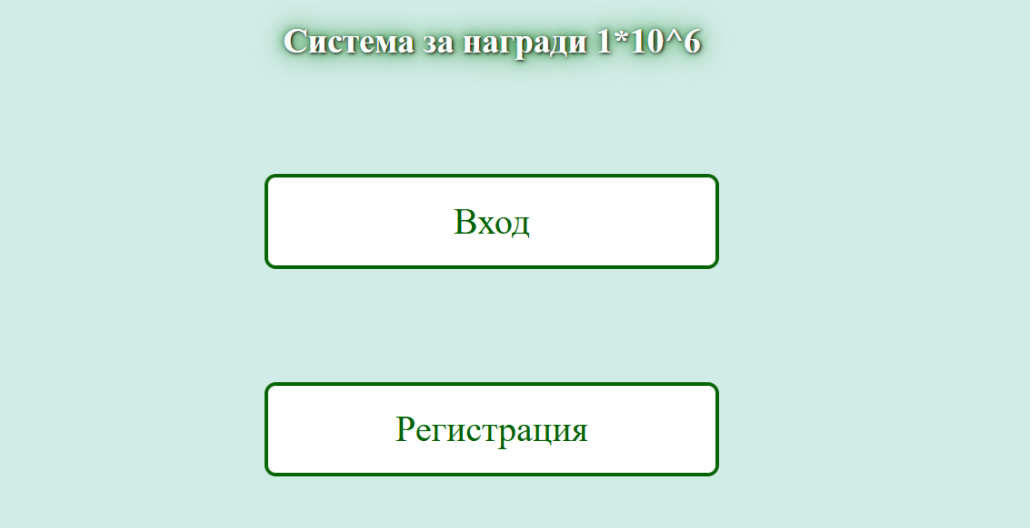
## Файлово разпределение на проекта

В основната папка на проекта има няколко директории и файлове, в които е разпределено съдържанието. Те имат следното предназначение:

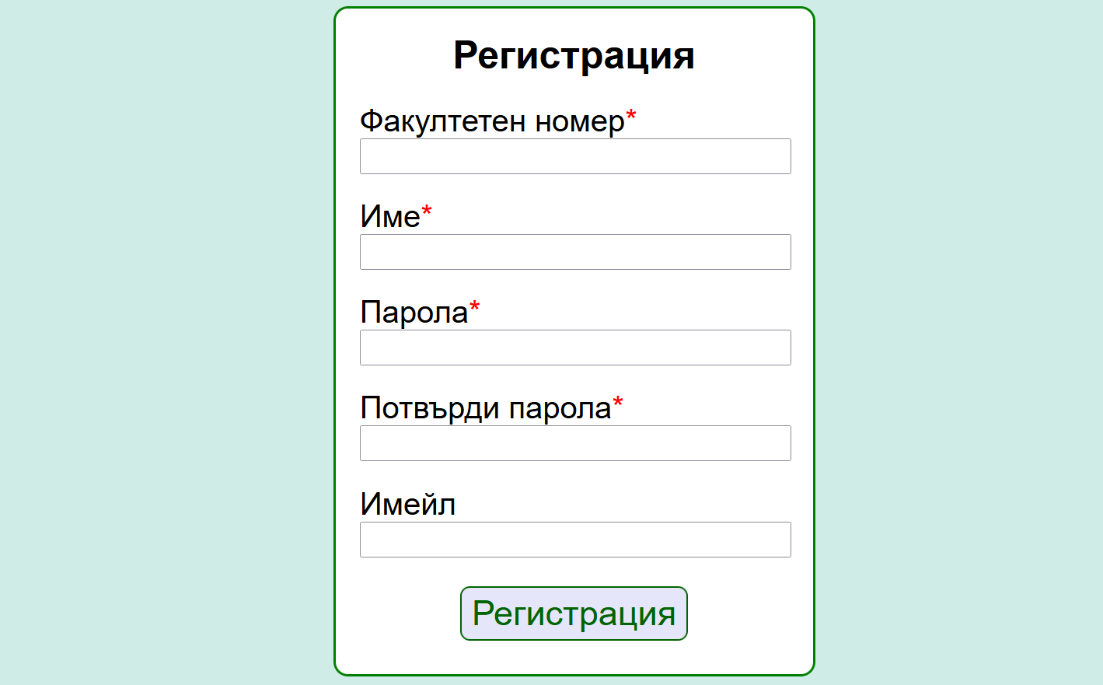
* **client** – Съдържа клиентската част на кода, реализираща частично презентационния слой.
* **configuration** – Съдържа конфигурационните файлове.
* **demo** – Съдържа примерни материали за тестване на проекта.
* **pictures** – Съдържа текущата версия на основната 1000x1000 пиксела картина в поддиректорията “**main**” заедно с няколко помощни картини. В поддиректорията “**part**” се пазят картините, които потребителите са добавяли с времето.
* **server** – Съдържа сървърната част от кода, реализираща бизнес слоя.
* **styles** – Съдържа стиловете за клиентската част от кода, с което се допълва презентационния слой.
* **Documentation.docx** – Текущата документация на проекта.
* **millionpixels.sql** – База данни с подготвена структура на таблиците и примерни данни. Нужна е само при инсталацията на проекта. След това може да се ползва като backup.

## Кратко ръководство на потребителя

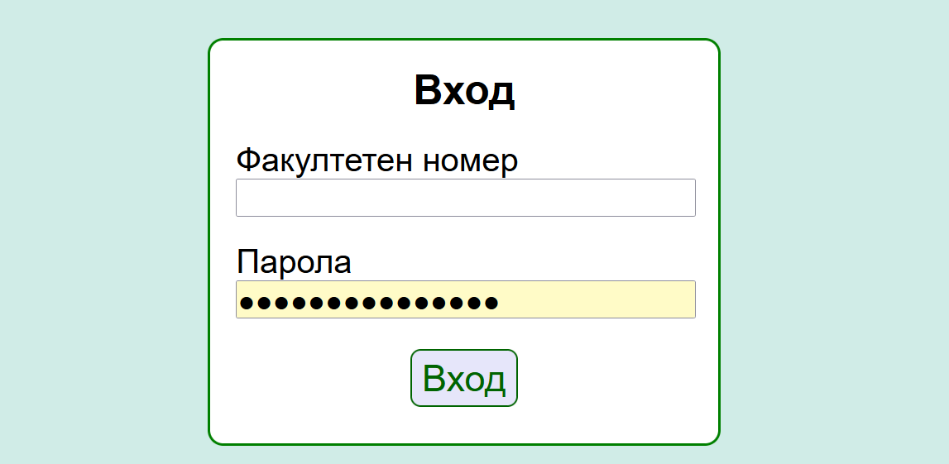
1. След пускане на проекта в браузъра се показва началното меню, от което може да се избере дали да се влезе в потребителски акаунт или да се направи нова регистрация.



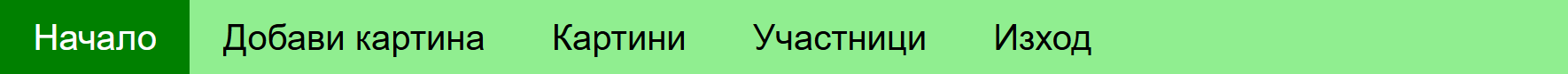
1. При избор на опцията за регистрация се появява формуляр, с който клиентите могат да си направят нова регистрация. С червена звезда са означени задължителните полета. При неуспешна регистрация се показват грешки във формуляра, за да може да бъдат поправени. При успешна регистрация се пренасочва към страницата за вход.



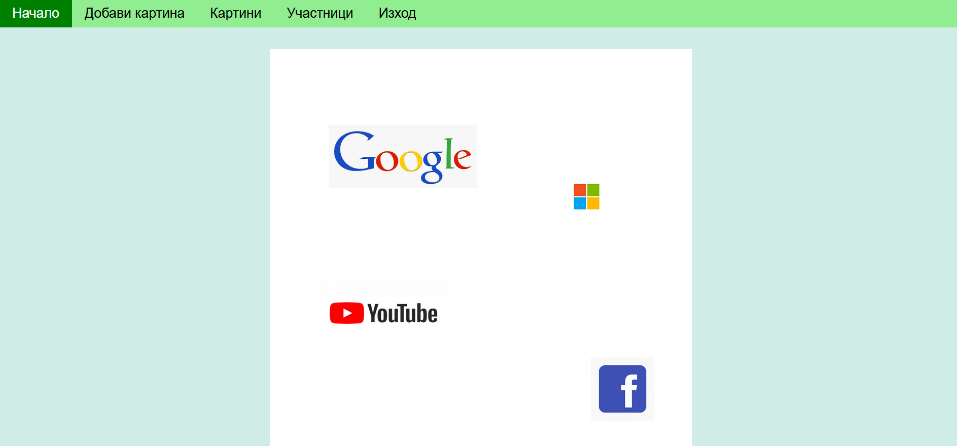
1. При избор на опцията за вход се появява формуляр за вход. При неуспешно влизане се показват грешки във формуляра, за да бъдат поправени. При успешна регистрация се пренасочва към началния раздел за потребители.



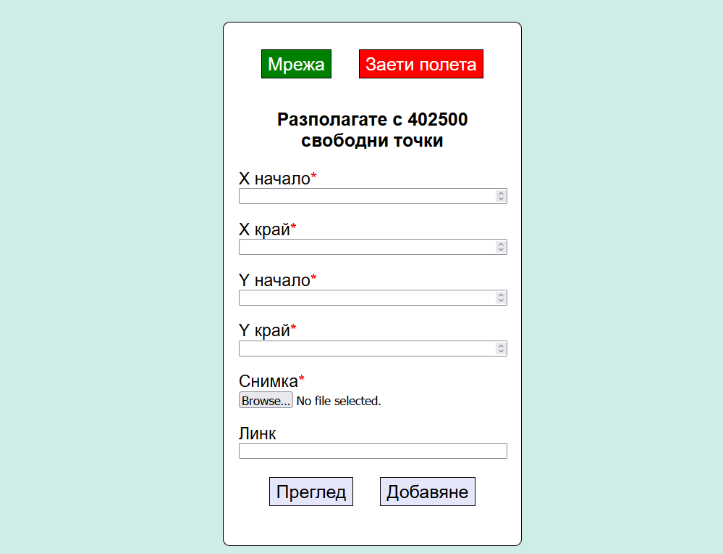
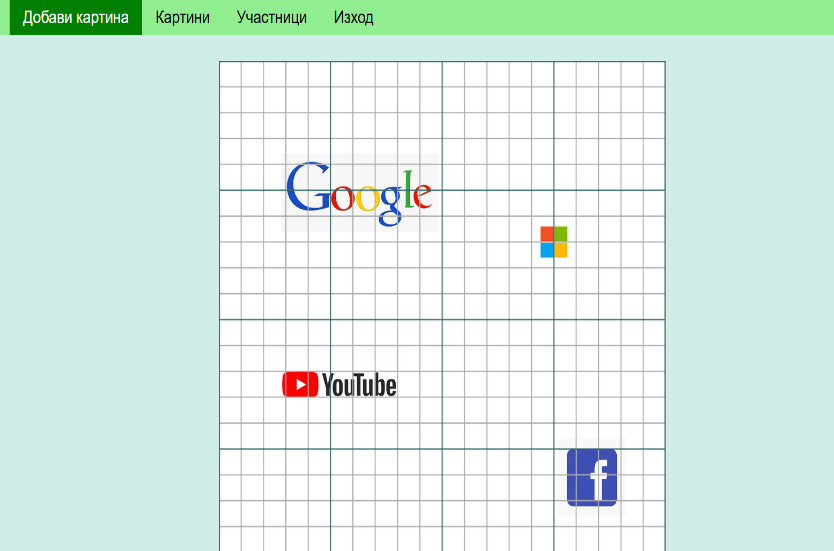
1. Във всеки раздел за потребителите има меню в горната част на страницата, в което е показан активния раздел. Разделите са следните:
   * + **Начало** – Начален раздел в който се вижда текущата версия на основната картина. Достъпен е за всички потребители.
     + **Добави картина** – Раздел за добавяне на картини. Достъпен е само за потребителите, които имат статус на участници.
     + **Картини** – Раздел за управление на картините. Достъпен е само за администратори.
     + **Участници** – Раздел за управление на участниците. Достъпен е само за администратори.
     + **Изход** – Изход от акаунта и връщане към началното меню. Достъпен е за всички потребители.



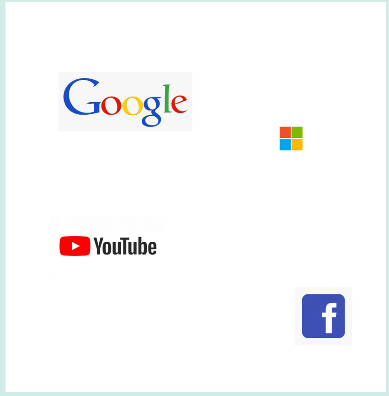
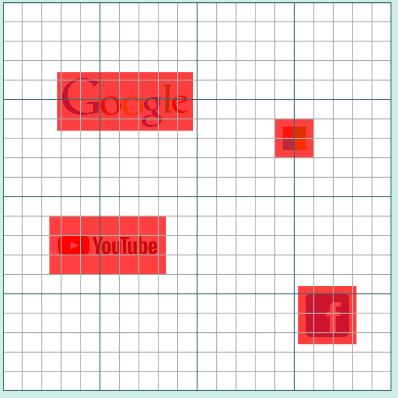
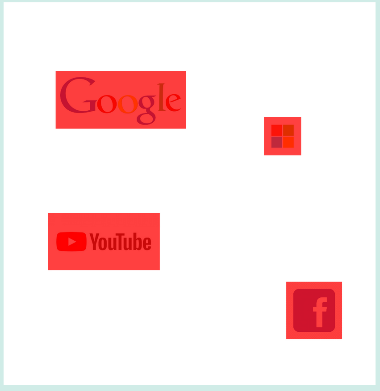
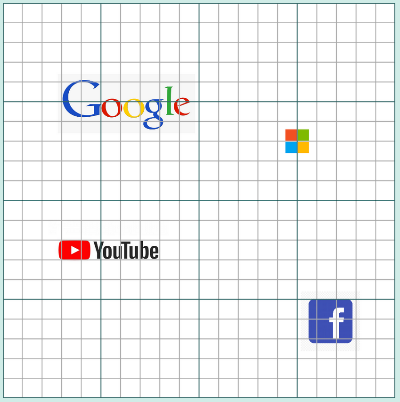
1. В началния раздел се вижда текущата версия на основната картина. Може да се кликне върху картините, към които има добавени хиперлинкове.



1. В раздела за добавяне на картина се показва текущата версия на основната картина и формуляр за добавяне на нова картина.

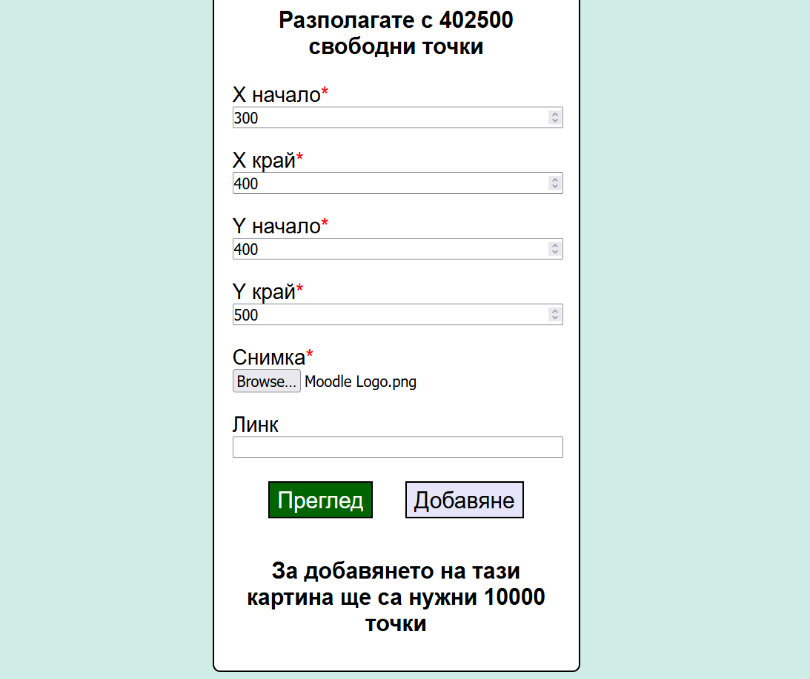


При натискане на горните бутони може да се активират и деактивират съответните режими.

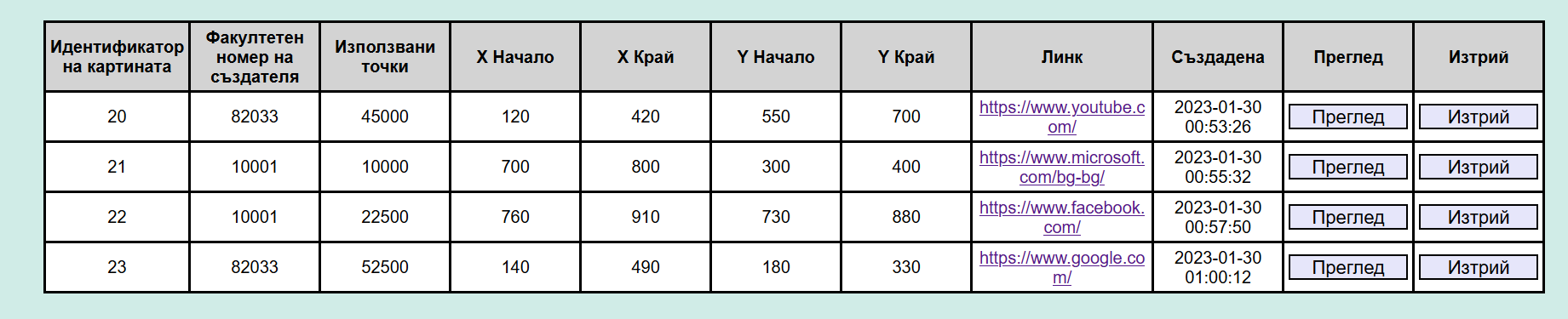
Ако е активиран режима за мрежа, при цъкане върху мрежата автоматично ще бъдат попълнени координатите на съответното квадратче.

При попълване на формуляра и натискане на **Преглед** се изписва нужния брой точки за добавянето на картината и върху основната картина се показва със зелена сянка къде ще се намира добавената картина. След махане на мрежата и заетите полета може да се натисне върху картината за пренасочване към хиперлинка, ако е попълнен. Картината заема пикселите с координати в полуотворените интервали [x\_start,x\_end) и [y\_start,y\_end).



След натискане на бутона за добавяне се изпраща заявка за добавяне. Ако има грешки като застъпване с други картини и недостатъчен брой точки, те се показват във формуляра. При успешно добавяне страницата се презарежда и се вижда новата версия на основната картина.

1. В раздела за картини се показва основната картина, а под нея има таблица с възможните действия.



При натискане на бутона за преглед се появява червена сянка около съответната картина, за да се види къде се намира.

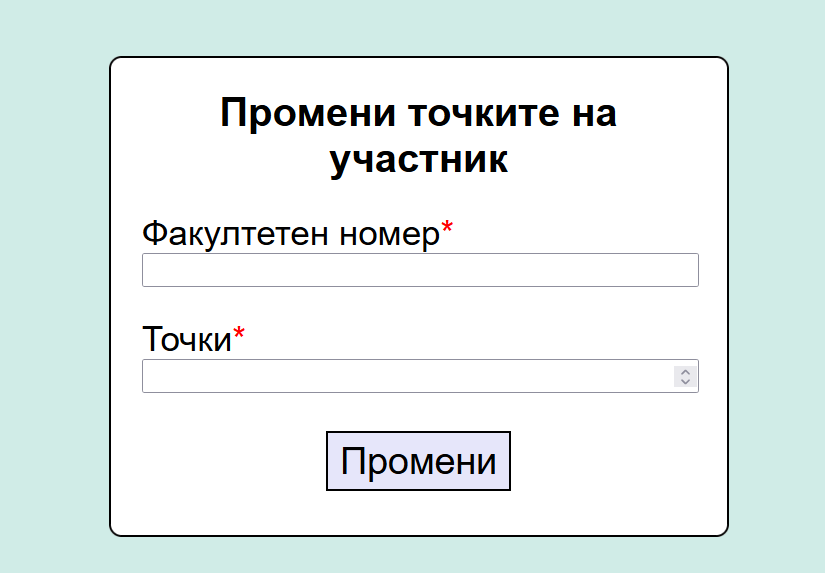
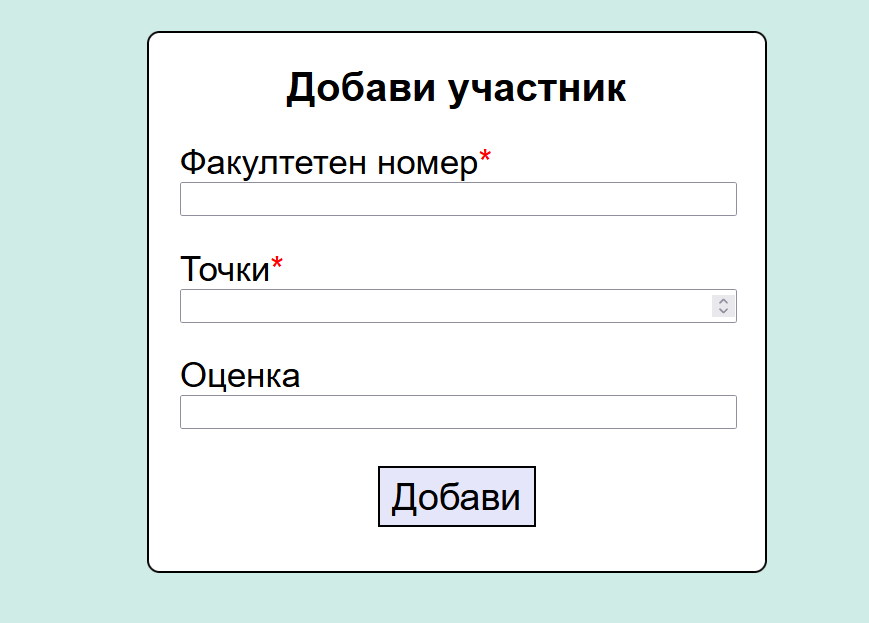


При натискане на бутона за изтриване се появява промпт дали сме сигурни, че искаме да изтрием картината. При потвърждаване картината се изтрива от основната и от таблицата. Използваните точки се възвръщат на участника, добавил картината.

1. В раздела за участници се появява таблица, в която са показани участниците и възможните действия върху тях. При бутоните за промяна на участието може да се позволи или забрани на участниците да добавят картини. При натискане на бутона за изтриване се показва промпт дали сме сигурни, че искаме да изтрием участника. Само участници, които никога не са добавяли картини могат да бъдат изтрити.



Освен това има две форми съответно за добавяне на нов участник и за промяна на точките на съществуващ участник. При изпращане на формулярите в тях се появяват съобщения за грешки, ако има такива. В противен случай операциите са успешни и страницата се презарежда.



## Примерни данни

В базата данни вече има добавени готови данни, за да се тества проекта. Те са следните:

**Регистрирани потребители:**

|  |  |  |  |
| --- | --- | --- | --- |
| Факултетен номер | Парола | Статус на участник | Статус на администратор |
| 82033 | abc | ДА | ДА |
| 10003 | 000 | ДА, но има забрана да добавя картини | НЕ |
| 10002 | 789 | НЕ | ДА |
| 10001 | 456 | ДА | НЕ |
| 10000 | 123 | НЕ | НЕ |

**Участници:**

|  |  |  |
| --- | --- | --- |
| Факултетен номер | Налични точки | Позволено участие |
| 99999 | 50000 | ДА |
| 82033 | 500000 | ДА |
| 10003 | 20000 | НЕ |
| 10001 | 40000 | ДА |

Може да забележим, че факултетни номера 10000 и 10002 имат регистрация, но не са сред наградените участници и не могат да добавят картини, а само да гледат текущата основна картина. Факултетен номер 99999 пък е сред наградените, но все още не си е направил регистрация, с която да добавя картини.

Освен тях има и 4 добавени картини към основната. Те автоматично се запазват в “**pictures\parts**” и информацията за тях е следната:

**Картини**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Име на картината | Факултетен номер на създателя | X координати | Y координати | Хиперлинк |
| i2132096988.png | 82033 | [120,420) | [550,700) | <https://www.youtube.com/> |
| i361919509.png | 10001 | [700,800) | [300,400) | <https://www.microsoft.com/bg-bg/> |
| i283098394.png | 10001 | [760,910) | [730,880) | <https://www.facebook.com/> |
| i484849009.png | 82033 | [140,490) | [180,330) | <https://www.google.com/> |

Допълнително в “**demo\pictures**” има 5 примерни картини, които може да се ползват за тестване на добавянето. 4 от тях са същите както вече добавените по-горе, но това не пречи да се добавят отново на свободни места.

## Описание на базата данни

В базата данни има 3 таблици:

* **participants** – Таблица с участниците в добавянето на картини
* **users** – Таблица с регистрираните потребители
* **pictures** – Таблица с добавените картини

Важно е да се прави разлика между участник и потребител. Участниците са всички, които са наградени от администратор с определен брой точки, с които могат да добавят картини към цялостното изображение. Някои от тях може все още да не са се регистрирали в системата и да не са потребители, но възможността за добавяне на картини ще им бъде достъпна след регистрация. Потребителите пък са всички, които са си направили регистрация. Някои от тях може да имат статут на участници и да могат да добавят картини, но други може да не са сред участниците и да нямат тази възможност. Допълнително всеки потребител може да има и статут на администратор, което да му дава достъп до функционалности за администриране на системата.

В таблицата **participants** имаме следните корони:

* **fn** – Факултетен номер на наградения участник. Полето е първичен ключ и е уникално.
* **points** – Общ брой точки, които са дадени на участника.
* **grade** – Оценка на участника. Незадължително поле, което е чисто информативно и към момента няма функционално значение.
* **forbidden\_to\_participate** – Поле, което указва дали към момента участникът има временна забрана за добавя картини.
* Допълнително имаме и два тригера при добавяне и при изтриване на редове. Те обновяват статуса на участник на регистрираните потребители в таблицата **users**.

В таблицата **users** има следните колони:

* **fn** – Факултетен номер на потребителя. Полето е първичен ключ и е уникално.
* **name** – Име на потребителя. Задължително, но няма функционално значение.
* **email** – Имейл на потребителя. Незадължително, в бъдеще може да се използва за изпращане на съобщения до потребителите.
* **password** – Хеширана парола за автентикация на потребителя.
* **is\_participant** – Дали потребителят е сред участниците и може да добавя картини.
* **is\_administrator** – Дали потребителят има административни права.

В таблицата **pictures** има следните колони:

* **id** – Уникален идентификатор на картините, които се определя автоматично и инкрементално. Това поле е основен ключ и е уникално.
* **creator\_fn** – Факултетен номер на създателя на картината. Освен това полето има външен ключ към факултетните номера в таблицата participants. Това означава, че само участници могат да добавят картини и един участник не може да бъде изтрит, ако някога е имал история на добавяне на картина.
* **points\_cost** – Цена в точки на съответното добавяне на картината.
* **x\_start**, **x\_end**, **y\_start** и **y\_end** – Координати на добавената картина.
* **picture\_name** – Име на добавената картина. Всички добавени картини се пазят в директорията “**picture/parts**” със съответното име. Полето е уникално.
* **link** – Хиперлинк за съответната картина. Полето е незадължително и може да бъде NULL.
* **text** – Текст, който ще се показва при местене на курсора върху картината. Полето е незадължително и може да е NULL.
* **created** – Дата и час на добавяне на съответната картина. По подразбиране се взима текущото време на машината.
* **deleted** – Дата и час, когато е изтрита картината. По подразбиране е NULL и това показва, че картината не е изтрита. При изтриване от страна на администратор редът за съответната картина не се изтрива изцяло, а само се променя това поле с датата и часа на изтриване. Това позволява в бъдеще да се реализира таймлапс, който да показва как се е променяла основната картина с времето.

## Описание на програмния код

Програмният код е разделен в множество поддиректории в зависимост от предназначението.

1. Сървърна част – директория “**server\**”
   * **database\db.php** – Реализира клас **Database**, който капсулира нужните заявки към базата данни.
   * **login\login.php** – Реализира обработка на заявката за влизане на потребител. При успешна автентикация започва нова сесия за съответния потребител. При неуспех връща отговор с подходящи съобщения за грешка.
   * **register\register.php** – Реализира обработката на заявката за регистрация на потребител. При успех добавя регистрирания потребител към базата данни. При неуспех връща отговор с подходящи съобщения за греша.
   * **navigation\navigation.php** – Създава навигация за потребителите в зависимост от техните права и текущия отворен раздел.
   * **view\createLinks.php** – Създава <map> таг с хиперлинкове за основната картина.
   * **add\addPicture.php** – Реализира заявка за добавяне на нова картина към основната.
   * **manage\addParticipant** – Реализира заявка за добавяне на нов участник.
   * **manage\changeAllowanceOfParticipant.php** – Реализира заявка за промяна на възможността на участник да добавя картини.
   * **manage\createParticipantTable.php** – Създава таблица с участниците, която да се използва за обратна връзка за администратора и като интерфейс за други операции.
   * **manage\createPictureTable.php** – Създава таблица с текущо добавените картини, която да се използва за обратна връзка за администратора и като интерфейс за други операции.
   * **manage\deleteParticipant.php** – Реализира заявка за изтриване на участник.
   * **manage\deletePicture.php** – Реализира заявка за изтриване на добавена картина.
   * **manage\updateParticipant.php** – Реализира заявка за обновяване на наличните точки на участник.
   * **utilities\destroySession.php** – Реализира заявка за прекратяване на сесията.
   * **utilities\getNumberOfFreePoints.php** – Реализира функция за получаване на броя на свободните точки за даден участник.
   * **utilities\testInput.php** – Реализира премахване на специалните символи от низове с цел повишаване на сигурността на сайта.
2. Клиентска част– директория “**client\**”
   * **Index.html** – Реализира началното меню, от което може да се избере дали да се влезе в потребителски акаунт или да се направи нова регистрация.
   * **login\login.html** и **login\login.js** – Реализират интерфейса за вход на потребителя и изпращат заявки към съответния сървърен скрипт.
   * **register\register.html** и **register\register.js** – Реализират интерфейса за регистрация и изпращат заявки към съответния сървърен скрипт.
   * **view\view.html** – Реализира началния раздел, от който може да се види текущата версия на основната картина.
   * **add\addPicture.html** и **add\addPicture.js** – Реализират интерфейса за добавяне на нова картина към основната и изпращат заявка към съответния скрипт.
   * **manage\pictures.html** и **manage\pictures.js** – Реализират интерфейс за преглед на добавените картини и изтриването им, както и изпращане на заявка за изтриване към съответния скрипт.
   * **manage\participants.html** и **manage\participants.js** – Реализират интерфейс за преглед на участниците както и забраняването им за участие и изтриването им, а също така и интерфейс за добавяне на нов участник и промяна на точките на съществуващ участник. Изпраща заявки към съответните скриптове.
   * **utilities\logout.js** – Реализира функция за изпращане на заявка за прекратяване на текущата сесия, което се използва за изход на потребителя и връщане към началното меню.
   * **utilities\sendRequest.js** – Пакетира изпращането на заявки към сървъра във функция.
   * **redirect\noPermission.html** – Страница, към която се препраща, ако клиент се опита за достъпи страници, за които сесията му няма нужните права.
3. Стилове – директория “**styles\**”
   * **main.css** – Задава основния фон на страниците.
   * **index.css** – Задава стила на началното меню *client\index.html*.
   * **errors.css** – Задава стила на грешките, даващи обратна връзка.
   * **authForm.css** – Задава стила на формите за вход и регистрация в *client\login\login.html* и *client\register\register.html*.
   * **navigation.css** – Задава стила на навигацията за потребителите.
   * **pictures.css** – Задава стила на картините, които се показват в страниците *client\view\view.html, client\add\addPicture.html, client\manage\pictures.html*.
   * **requestForm.css** – Задава стила на формите за заявки в *client\add\addPicture.html* и *client\manage\participants.html*.
   * **table.css** – Задава стила на таблиците, които се показват в *client\manage\pictures.html* и *client\manage\participants.html*.

## Принос на студента, ограничения и възможности за бъдещо развитие

Проектът е реализиран изцяло от мен, тъй като съм единствен в екипа.

Няколко идеи за бъдещо развитие на проекта:

* **Добавяне на таймлапс как се е променяла основната картина** – Първоначално мислех да реализирам тази функционалност, но се отказах, защото отделих твърде много време на останалите неща в проекта. Структурата на данните е направена така, че възстановяването на основната картина в даден момент от време да е напълно възможно без да се пази допълнително копие на основната картина.
* **Добавяне на меню, от което участниците могат да следят собствените си картини** – За момента само администраторите могат да следят кои картини от кои участници са добавени. Ще е добра идея да се даде подобна възможност и на участниците, които не са администратори. Също може да се добави и възможност в това меню участниците да трият собствените си картини, за да си възвърнат използваните точки.
* **Добавяне на функции за автоматично изчисляване на точките на участниците** – Към момента характеристики като оценката на участниците са декоративни и са независими от наличните точки. Би могло да се добави функционалност, която автоматично разпределя точки в зависимост от оценките например.

Едно ограничение на проекта е, че използва една база данни за управление на всички заявки. Тя често изисква блокиране на таблици в случаи на добавяне, промяна или триене на данни, за да се гарантира консистентност между данните и картините. Ако броят на заявките в даден момент от време стане твърде голям, базата от данни може да бъде претоварена и да отказва да изпълнява заявки на някои потребители.

## Какво научих

По време на проекта научих множество неща. Ето някои от по-важните:

* **Как се работи с PHP** – Досега не съм писал на PHP и трябваше повечето неща да ги уча в движение. Дебъгването на кода се оказа сложно, но по време на разработката на проекта открих бързи начини как може да дебъгвам.
* **Как се работи с библиотеката GD на PHP** – Наложи ми се обработвам изображения и за тази цел библиотеката GD се оказа изключително полезна. Научих за доста от функционалностите, но ми беше трудно да намеря точно това, което ми трябва.
* **Как се работи с JavaScript** – За последно писах на JavaScript преди 4 години и то на базово ниво. Наложи ми се да си припомня какво знам за него и да науча доста нови неща за езика.
* **Как да изпращам заявки от клиент към сървър** – Не знаех как точно става изпращането на заявка чрез JavaScript към PHP сървър и трябваше да го проуча.

Бих казал, че се сдобих с доста по-ясна представя как се случват нещата в сферата на уеб технологиите и как работят голяма част от уеб сайтовете.

Надявам се наученото да ми бъде от полза в бъдеще.

## Източници

Като цяло не съм следил от кои точно страници съм черпил информация, но ги помня като уеб сайтове:

1. <https://github.com/ngadzheva/WebTechnologies-labs-KN22>
2. <https://github.com/alexanderStefanovS/web-technologies-course-fmi>
3. <https://www.w3schools.com/>
4. <https://www.php.net/manual/en/>
5. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
6. <https://mariadb.org/>
7. <https://stackoverflow.com/>
8. <https://www.geeksforgeeks.org/how-to-install-php-gd-in-windows/>

Предал (подпис): ………………………….

/ФН *82033, Бойко Красимиров Борисов, спец. КН, група 5*/

Приел (подпис): ………………………….

/доц. *Милен Петров*/