

# Тема: *Meme Generator*

Имена: Никол Светозарова Емануилова

ФН: 61987

Начална година: 2016

Програма: бакалавър, (СИ)

Курс: 3

Тема: Meme Generator

Предмет: wwwTech2018\_19\_12ed\_SI\_summer

Имейл: [nicolesvemanuilova@gmail.com](mailto:nicolesvemanuilova@gmail.com)

Дата: 2019-06-20

Преподавател: доц. д-р Милен Петров

# Съдържание

<b>Условие</b>	<b>3</b>
<b>Въведение</b>	<b>3</b>
<b>Теория</b>	<b>4</b>
<b>Архитектура</b>	<b>4</b>
<b>Използвани технологии</b>	<b>4</b>
<b>Инсталация и настройки</b>	<b>5</b>
<b>Кратко ръководство на потребителя</b>	<b>5</b>
<b>Примерни данни</b>	<b>8</b>
Примерни данни за регистрация в системата	8
Примерни данни за вход в системата	8
<b>Описание на програмния код</b>	<b>8</b>
<b>Бъдещо развитие</b>	<b>10</b>
<b>Какво научих</b>	<b>10</b>
<b>Използвани източници</b>	<b>11</b>

# ТЕМА: Meme Generator

## 1. Условие

Условието на текущия проект е да се реализира система за създаване на мемета. Зададени са теми за реферати и за всяка тема трябва да се генерира мемета. Всеки студент избира шаблон, или качва своя картинка, като може да добавя текст и да го форматира. След това има възможност да запази резултата като чернова или да публикува своето мемета в системата, както и да го изтегли и запази локално. Всеки студент може да разглежда меметата на своите колеги в отделна страница, където се визуализират всички, които са отбелязани като публични.

## 2. Въведение

Приложението представлява уебсайт за генериране на мемета. Целта е да помогне за създаването на персонализирана покана от студентите в курса “Web технологии” за представянето на техните реферати, което е част от оценката.

Всеки студент се регистрира в системата като посочва своя факултетен номер и темата на реферат. При успешна регистрация, получава достъп до всички функционалности. Възможностите, които предоставя системата са следните:

- Използване на шаблон за мемета.
- Качване на собствена картинка.
- Добавяне, редактиране и форматиране на текст върху мемета, като могат да бъдат добавяни брой текстови полета по желание и да бъдат поставени на конкретно място.
- Изтегляне на картинка локално.
- Запазване на картинка в базата данни.
- Избиране дали запазената картинка да бъде чернова или публична за останалите потребители.
- Разглеждане на публикувани мемета от всички потребители, като за всяко е означено името на студент и темата на неговия реферат.

### 3. Теория

Системата поддържа един тип потребители - студентите, които трябва да направят своите покани за презентации на реферати. По тази причина със създаването на приложението са добавени част от темите, които могат да бъдат избрани. Всеки студент трябва да се регистрира в системата, за да има достъп до нейната функционалност. При създаване на профил освен личната информация, той трябва да попълни и темата на реферат, както и факултетния си номер. След валидиране на данните, той бива допуснат в уеб системата.

В страница "Meme Generator" потребител може да създаде своето меме, като има право да избира между предварително дефинирани шаблони или да качи своя картинка. След което може да добавя безброй текстови полета, да променя цвят, шрифт и местоположение на текста. Когато прецени, че е готов, може да запази резултата в базата или да го изтегли, за да може да публикува поканата си в социалните мрежи. Също така, ако отбележи мемето като публично, то ще бъде показано в страницата "Collection". В нея всички студенти могат да разгледат поканите, които са публични.

### 4. Архитектура

Уеб системата има трислойна архитектура, която е осъществена чрез принципите на архитектурния стил Model-View-Controller (MVC). Презентационният слой, който представлява изгледите на приложението, е изграден чрез HTML, CSS и JavaScript. Информацията за потребителите и рефератите се съхранява в базата данни. Бизнес логиката е осъществена чрез PHP и е разделена на модели, контролери и изгледи. Моделите служат за комуникация с базата - извличане, добавяне и редактиране на данни. Контролерите осъществяват получаването и изпращането на данни към презентационния слой, като връща данни в JSON формат, за да бъдат използвани, или директно изпраща изгледи.

### 5. Използвани технологии

За реализирането на системата са използвани следните технологии:

- *HTML5, CSS3* - за създаване на структурата на страниците и тяхното стилизиране
- *JavaScript ES6*
  - динамично добавяне на текстови полета;
  - зареждане на избран шаблон за меме;
  - зареждане на картинка, качена от потребител;
  - динамично рисуване в canvas елемент за обработка на картинка;
  - запазване на картинка на локален компютър;
  - изпращане на картинка към базата от данни;
  - динамично визуализиране на публичните мемета и информацията за тях, които са получени в JSON формат от бизнес слоя.
- *PHP 7.3.2.* - за осъществяване на бизнес логиката и комуникацията между презентационния слой и базата от данни.
- *XAMPP v3.2.3.* предоставя *Apache HTTP Server* и *MySQL* база от данни.
- Проектът е разработен с помощта на редактора *Visual Studio Code v1.35.*

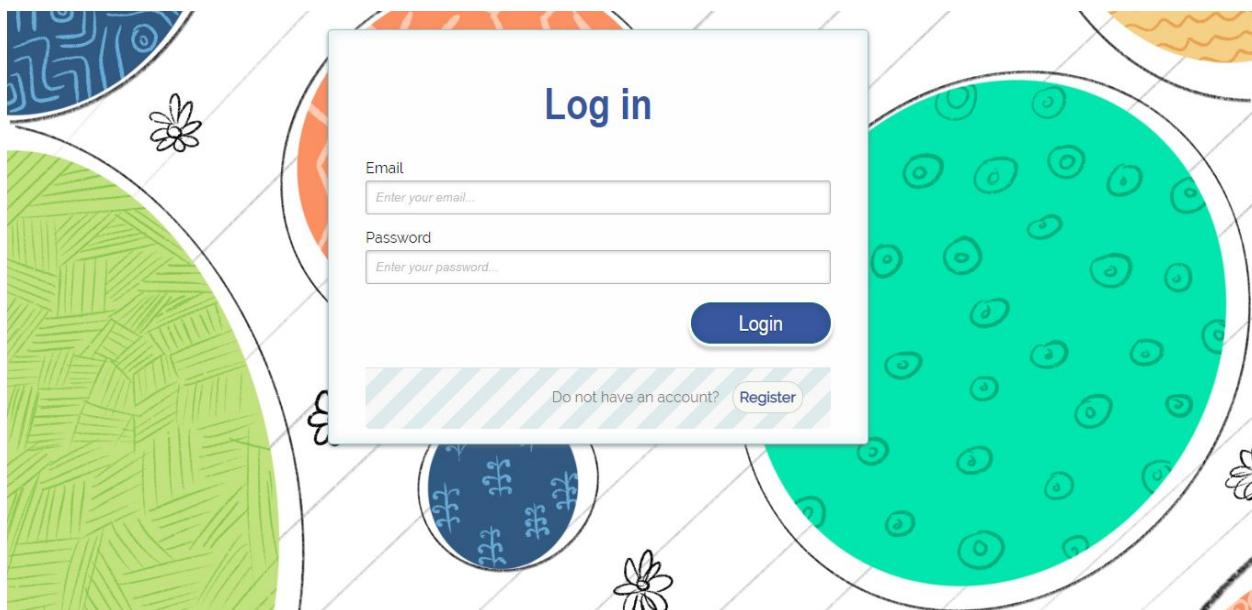
## 6. Инсталация и настройки

Стартирането на уеб приложението се осъществява с помощта на софтуерния пакет XAMPP, като трябва да е налична актуална версия. Чрез него се пускат HTTP сървър и MySQL база от данни. С помощта на phpMyAdmin могат да се създадат и редактират нужните таблици в базата, като нужните скриптове са добавени в папката на проекта. Съответно, за успешното стартиране всички файлове трябва да са в папката htdocs на XAMPP и проектът се достъпва в браузър чрез изписването на localhost:port/Memelt, като по подразбиране портът е 80.

Осъществяването на връзка с базата от данни и настройките, както и всички нужни пътища, са изнесени във файла *Memelt\config\config.php*. По този начин проектът е лесно приспособим към промени, тъй като корекциите се правят единствено в този файл.

## 7. Кратко ръководство на потребителя

При стартиране на приложението се отваря Login страницата за вход в системата. Това се случва чрез попълване на имейл и парола. Ако студент все още няма акаунт, то натиска бутона Register и бива отведен до страницата за регистрация, където попълва съответната форма. Нужните данни са имена, имейл, факултетен номер, тема на реферат и парола. Ако въведените данни са коректни, то той бива допуснат до системата.



Фиг. 1 Страница за вход в системата

**Register**

Name

Email

Faculty Number

Password

Topic

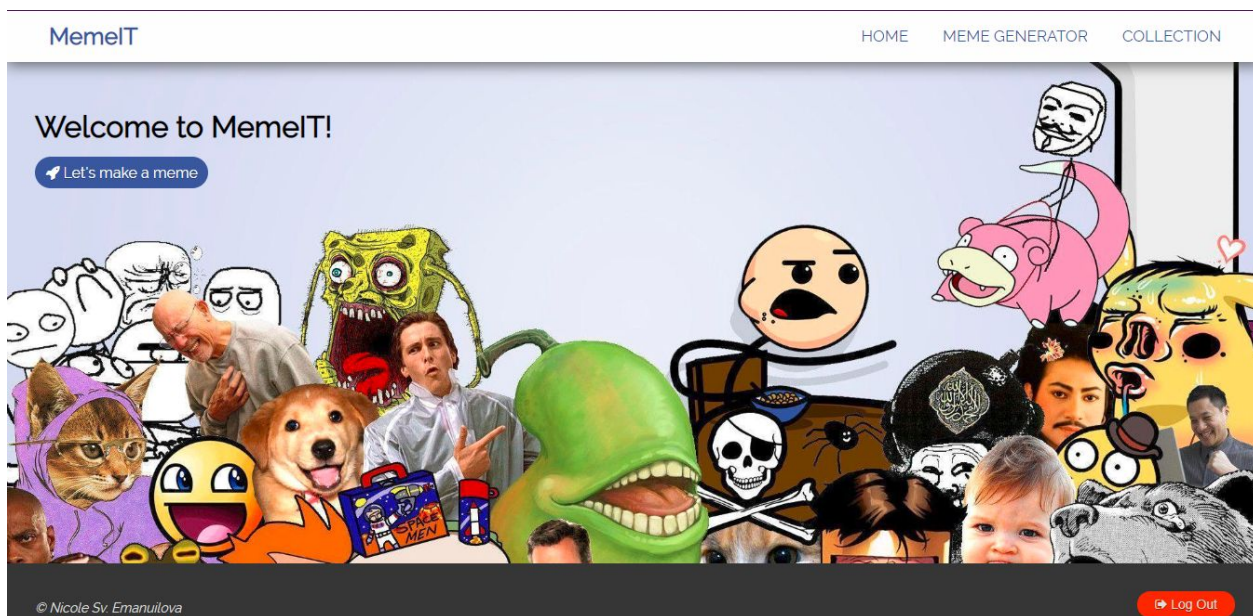
Confirm your password

[Sign up](#)

[Already have an account? Log In](#)

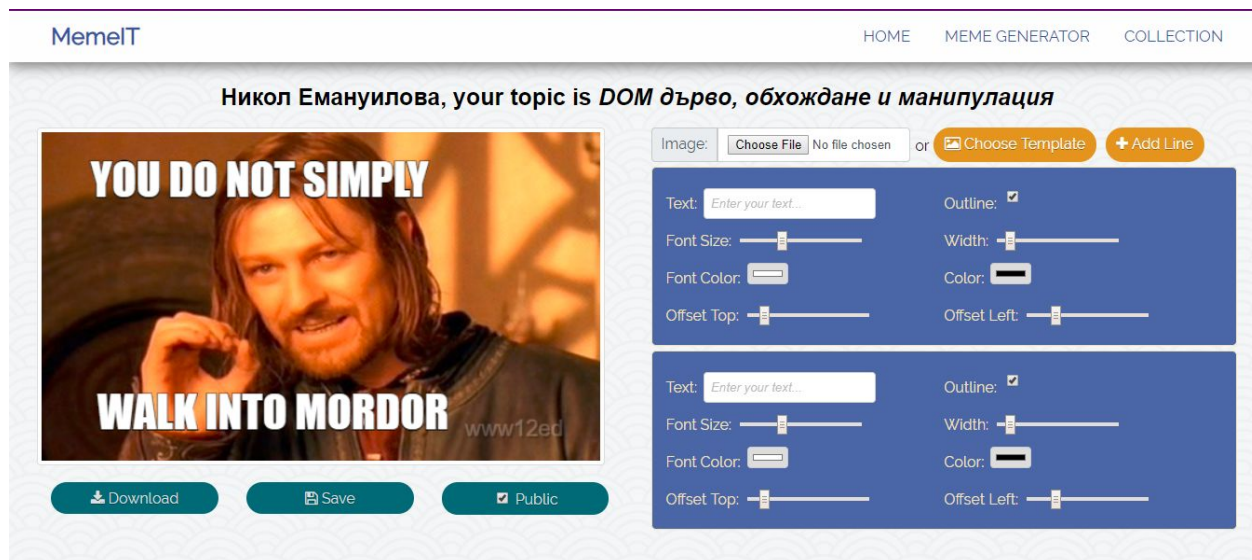
Фиг. 2 Страница за регистрация в системата

След успешен вход в системата, се визуализира началната страница на приложението. Чрез бутона "Let's make a meme" потребителят директно може да отиде на страницата за генериране на мемета, или да избере от навигацията към коя страница да отиде.



Фиг. 3 Начална страница



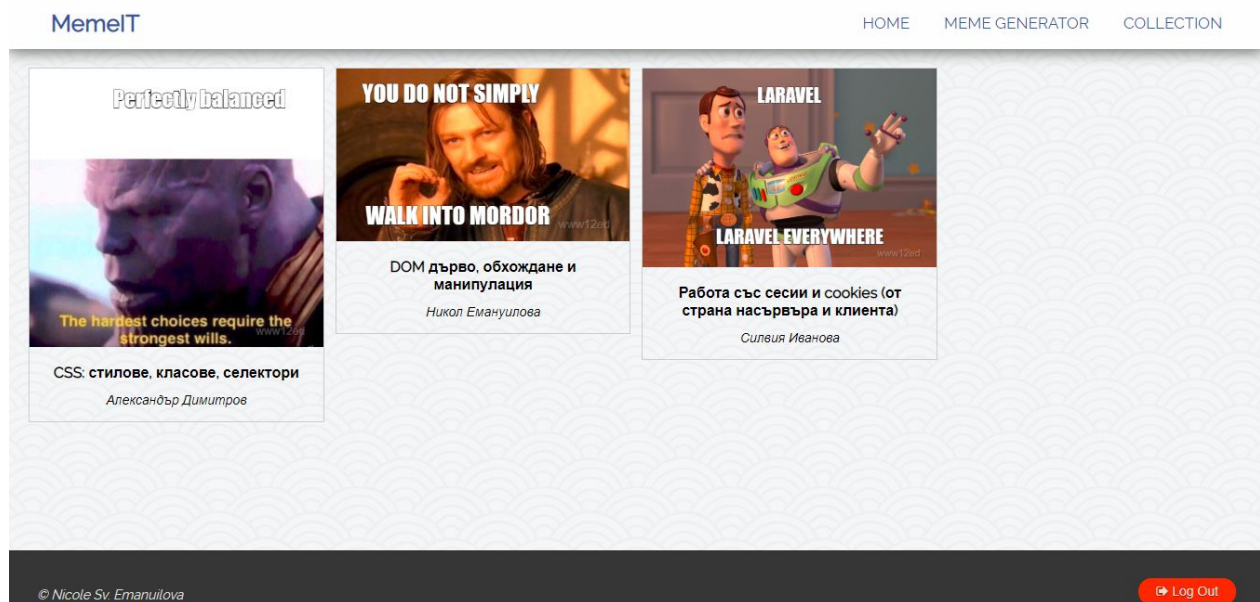


Фиг. 4 Страница за генериране на мемета

След като потребител влезе в страницата за генериране на мемета, той може да види темата на своя реферат и да започне работа върху своята покана. Дава се възможност за качване на картинка от компютъра или използването на шаблони, които са визуализирани в галерия чрез допълнителен модал. Всяко мемета създадено чрез системата има воден знак (watermark), който показва, че е направено за предмета по Web технологии.

По начални настройки са дадени опции за добавяне на два реда текст, като за всеки се редактира съдържанието, големината, цвета, границата, както и местоположението на полето. При желание могат да се добавят още текстови полета и всяко да се персонализира отделно. Чрез бутона "Download" мемето, което се вижда в canvas полето в момента, се изтегля на локалната машина. При отбелязване на "Public" мемето се отбелязва като публично и ще се визуализира в колекцията, но преди това трябва да бъде запазено в базата данни чрез бутона "Save". Ако резултатът не се отбележи като публичен, той се запазва като чернова и може да бъде довършен в по-късен момент.

Ако потребителят иска да разгледа меметата на своите колеги, то трябва да отиде в страница "Collection". Там се визуализират всички покани, които са посочени като публични. Под всяка пише темата на реферата и името на студента, който я е направил. Те са наредени в сравнително малък размер и при кликуване на конкретно мемета, то се отваря в нов раздел и може да се види в пълен размер.



Фиг. 5 Страница с колекция от публични мемета

## 8. Примерни данни

### 8.1. Примерни данни за регистрация в системата

*Name:* Руслан Емануилов

*Faculty Number:* 61000

*Topic:* 1 - Google - Web Performance Best Practices

*Email:* ruslanemanuilov@gmail.com

*Password:* 123456

### 8.2. Примерни данни за вход в системата

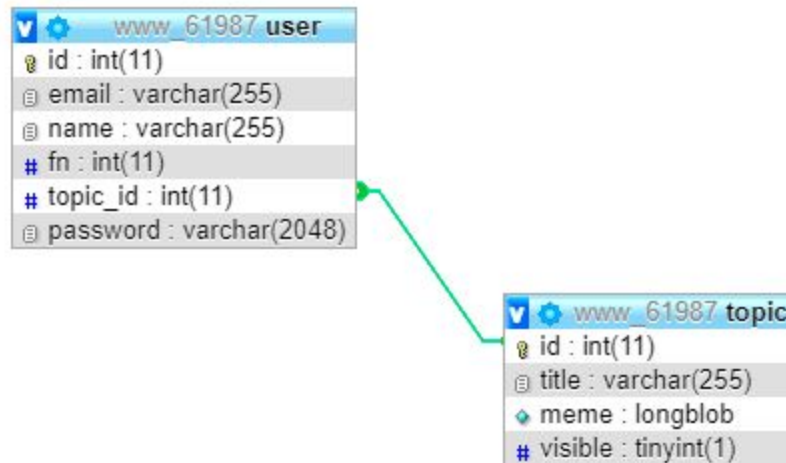
*Email:* [nicolesvemanuilova@gmail.com](mailto:nicolesvemanuilova@gmail.com)

*Password:* 123456

## 9. Описание на програмния код

Базата данни се състои от подходящо именувани таблици, като в тях се съхранява информацията за потребителите, както и за темите на реферати. Постигната е коректната връзка между тях, защото на всеки реферат трябва да отговаря точно едно мемe и да се знае дали то е публично, или не.





Фиг. 6 Схема на базата данни

Входна точка за системата е файлът *index.php*. Той отговаря за включването на всички файлове, които са нужни за конфигурацията и стартирането на приложението.

Следвайки MVC модела е изградена следната йерархия от директории:

- **config**

Тази директория съдържа всички файлове, чрез които се конфигурира приложението. Така по-лесно се добавят необходимите контролери, модели, изгледи и глобални константи.

- **controllers**

В тази директория се намират контролерите, които осъществяват връзката между моделите и изгледите, като връщат съответните изгледи или данни в JSON формат. Тези данни са резултат от AJAX заявки, които се получават от презентационния слой.

- **models**

Тук са всички модели, които представят таблиците от базите данни, и съответно осъществяват комуникацията с тях. Функцията им е да изпълняват нужните заявки към базата и да връщат резултата към съответния контролер, за да бъде обработен.

- **views**

В тази директория се намират всички изгледи на системата. При нужда те се зареждат от базовия контролер. Тук са поместени и нужните файлове за стилизация и интерактивност на страниците като css и js файлове, снимки и шрифтове.

- **css**
- **fonts**
- **images**
- **js**

- **routes**

Тази директория съдържа класът Router, който реализира маршрутизацията в системата, а в *api.php* са дефинирани всички възможни пътища в системата, които да водят към съответните контролери и функционалности.

## 10. Бъдещо развитие

Уеб системата е реализирана, така че лесно да могат да бъдат внедрени промени и нови функционалности. MVC архитектурата е интуитивна и човек, който има основни познания, би могъл бързо да се ориентира в структурата на проекта.

Самото приложение има доста възможности за подобрене и развиване, като например добавяне на възможност за директно споделяне в социалните мрежи. Също така, ще е добре да се добавят повече възможности за обработване на самите картинки, като рисуване върху тях, добавяне на стикери и други картинки, и т.н.

Като минус на системата може да се отбележи и липсата на различни роли потребители. Би било удобно да има администраторски акаунт, чрез който през системата да се добавят темите за реферати, тъй като в момента това се случва директно през базата данни. Такъв акаунт би имал и отговорността за съдържанието на публикуваните мемеа и дали те са подходящи, тъй като всички студенти ще имат достъп до тях.

## 11. Какво научих

Благодарение на работата по този проект научих много неща Web програмирането, тъй като темата ми беше интересна и с удоволствие разработих системата. Разбрах колко е важно и полезно кодът да е добре структуриран и самата логика да е разделена между презентационния слой, бизнес логиката и базата данни. Така те са независими и комуникират помежду си само когато се налага. Чрез прилагането на архитектурния стил MVC системата е много по-адаптируема и лесно може да се пригоди към промени, тъй като те се случват само в съответния компонент. Научих се и как да създам подходяща база данни, така че по-ефективен начин да се съхранява информацията и съответно лесно да се достъпва. Познанията ми по езика PHP доста се обогатиха, тъй като те бяха доста базови в началото на разработването на този проект. Също така се запознах повече и с езикът JavaScript. Научих как да обработвам и рисувам върху canvas елемент, както и да създавам динамично компоненти от презентационния слой.

## 12. Използвани източници

[1] Meme Generator

<https://imgflip.com/memegenerator>

[2] Meme Generator with JavaScript and Canvas

<https://www.anexia-it.com/blog/en/meme-generator-with-jquery-and-canvas/>

[3] User registration system with PHP and MySQL

<https://codewithawa.com/posts/complete-user-registration-system-using-php-and-mysql-database>

Предал (подпис): .....

/61987, Никол Емануилова,СИ, група 2/

Приел (подпис): .....

/доц. Милен Петров/

### История на версиите:

- Първа версия - 31.05.2019
- Втора и финална версия - 20.06.2019