**Имена:**  Таня Сейкова **фн:**  *62472* **Имейл: tanya.seykova@gmail.com**

**Имена:**  Георги Чобанов **фн:**  *62475* **Имейл: fn62475@g.fmi.uni-sofia.bg**

**Имена:**  Катерина Павлова **фн:**  *62486* **Имейл: katerinapavlova28@gmail.com**

**Начална година:** *2019* **Програма:** бакалавър, (СИ) **Курс: 3**   
**Тема: 11.2 Система, визуализираща кредитс (финални надписи)  
Дата: 2022-06-08 Предмет: w18prj\_SI \_final**

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

**Предаване:** Задачата се предава в архив с попълнен настоящия документ, проекта/проектите с кодовете, README.txt файл, който описва съдържанието на архива; папка с допълнителни компоненти и използваниресурси**, архива да се казва 9999\_project\_final.zip. (Успех!). (Редактирайте маркираните зони в жълто с коректната информация)**

# ТЕМА: 11.2 Система, визуализираща кредитс (финални надписи)

## 1. Условие

Да се направи система, която по зададен JSON файл в указан формат, в който са имената на избрани реферати, да визуализира екран, в който имената на тези реферати да излизат във вида като началните надписи на Междузвездни войни. Тази анимация да може да се спира, пуска и превърта отначало с помощта на бутони, както и да се пуска на цял екран.

Стиловете на тази анимация да могат да се променят както с config.json файл, така и с визуален редактор, който се намира на страницата.

За анимацията да може да се задава звук, който да върви с нея, и да може да се намалява и усилва.

Да могат да се добавят лични коментари на използващия системата, своеобразни бележки, в които потребителят да може да записва мнението си за рефератите, които вижда, или каквото друго пожелае.

Цялата конфигурация на една анимация - файлът с имената на рефератите, конфигурацията на стиловете и аудио файлът да могат да се записват в база данни, от където по-късно да се прочитат и анимациите да могат да се отварят отново от потребителя.

## 2. Въведение

Настоящият проект представлява система, която по зададен data.json файл с имената на избрани реферати, да визуализира тези имена като началните надписи във филма “Междузвездни войни”. Стиловете на тази анимация могат да се контролират от потребителя, тя може да се спира, пуска и визуализира на голям екран. Може да се добави аудио файл, както и коментари от потребителя, в които той записва своето мнение.

Анимациите се запазват под формата на файлове, като по-късно могат да се достъпят чрез помощта на база данни.

## 3. Теория

За реализацията на анимацията се използва единствено CSS, като се използват възможностите му за анимация и така наречените keyframes - опорни точки за моменти от анимацията (в нашия случай това са 0% и 100% - началото и края на анимацията). Техните атрибути са написани изрично, като браузърът попълва всичко между тях, което в нашия случай е изключително удобно.

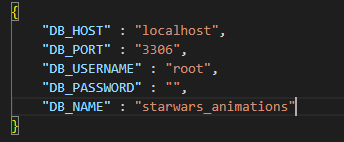
Запазването на анимации става, като стиловете за анимацията и съдържанието на анимация се записват в съответни json файлове и се изтеглят заедно със звуковия файл, използван за анимацията. Имената на всички тези файлове се запазват в база данни, заедно с името на анимацията, за да може да се достъпват после.

## 4. Използвани технологии

За реализация на проекта са използвани HTML5, CSS3, JavaScript ES6 и PHP Version 8.1.5, XAMPP Version 8.1.5, Apache 2.4.53, като за базата данни се използва MySQL.

## 5. Инсталация и настройки

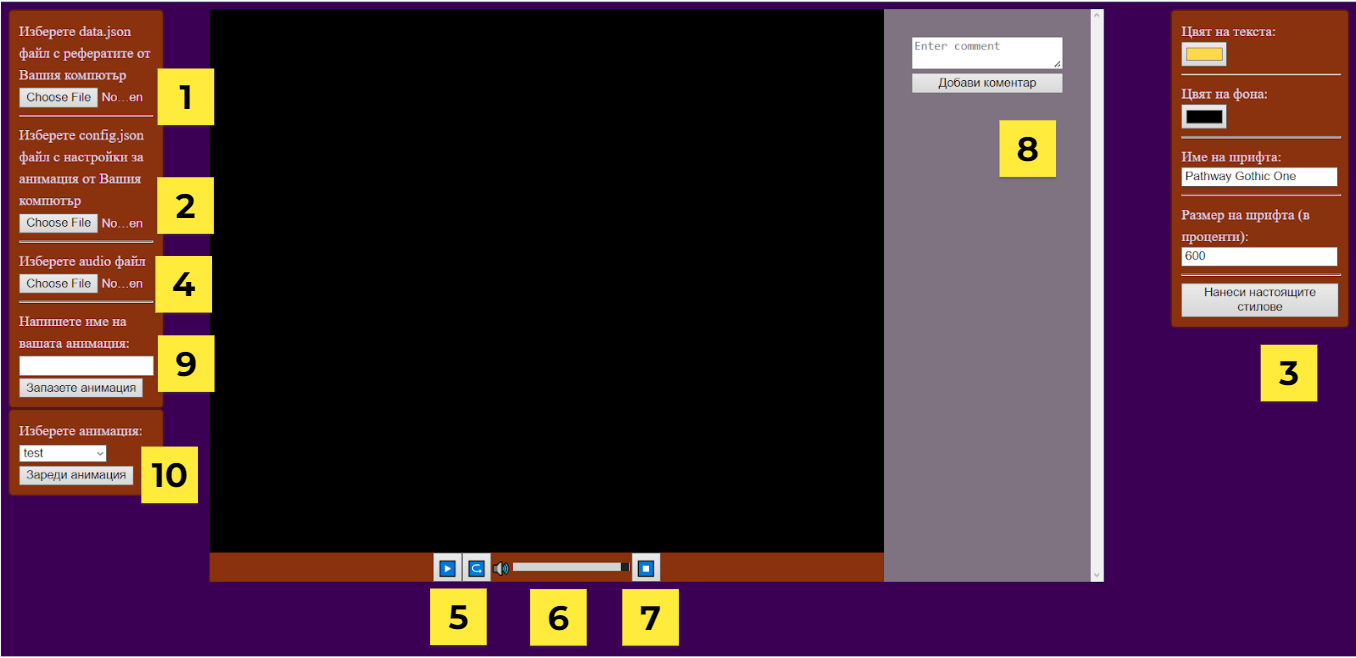
1. Създаване на файл configs-consts.json , намиращ се в root директорията на проекта, който съдържа следната информация:



1. Зареждане на съдържанието от файл *backend/db/db-util.sql* и зареждането му в локалната база.
2. Стартиране на сървърите.

## 6. Кратко ръководство на потребителя

Проектът има само една страница, на която се намира всичко. В следващите редове ще дадем кратко описание на всички достъпни от нея функционалности.



С бележките е отбелязан примерен поток на действията.

1. Потребителят може да избере data.json файл, в който са имената на рефератите. Ако потребителят е свързан към сървър или localhost и файлът json/data.json съществува, той автоматично ще бъде прочетен и анимацията ще може да бъде пусната. В противен случай е задължително да се избере такъв файл, за да може системата да работи.
2. Потребителят може да избере конфигурационен файл, подобен на показания примерен.
3. Потребителят може да редактира стиловете и от предложената форма за редакция. Ако е избран конфигурационен файл на 2), данните във формата ще се променят спрямо информацията във файла. И в двата случая, трябва да се натисне бутон “Нанеси настоящите стилове”, за да бъдат нанесени стиловете.
4. Потребителят може да добави аудио файл, който да бъде пуснат с изпълнението на анимацията.
5. Това са основните контроли на анимацията. Тя може да бъде пускана или слагана на пауза, както и да бъде пускана отначало. За да тръгне анимацията, трябва да има зареден data.json файл. С пускането на анимацията се пуска и аудио файлът, ако има такъв.
6. Това е контролът на звука. Звукът може да се усилва или намалява.
7. Това е бутонът за цял екран. При пускане на цял екран, останалите контроли се запазват.
8. Това е частта с коментарите. Потребителят може да пише коментар в текстовото поле, а с натискането на бутона, коментарът ще се визуализира в сивото поле.
9. Това е мястото за записване на анимация. Всички настоящи настройки - файлът с данните, конфигурацията на стиловете от формата за стиловете, както и аудио файлът (той също задължително трябва да се подаде) се записват във файлове и имената им заедно с името на анимацията се запазват в базата данни. Трябва да се напише име на анимацията и да се натисне бутона.
10. Системата чете записаните анимации от базата данни, чете файловете, и ги предоставя на потребителя. Той може да си избере някоя от тях от падащото меню, след което да натисне бутона и да я зареди.

Във файла json/linesConfig.json може да се попълни броят на редовете, които излизат на итерация на анимацията. Ако не се попълни друго, стойността по подразбиране е 5.

## 7. Примерни данни

В папките json/ и audio/ са дадени примерни файлове с данни, конфигурационен файл и аудио файл.

В базата данни са създадени примерни анимации, които могат да бъдат отворени.

## 8. Описание на програмния код

Основната функционалност на системата, тоест - самата анимация, е направена единствено чрез CSS и информацията за нея е описана в styles.css файла, в класа .crawl и keyframe анимацията crawl, като самото прилагане на анимацията се случва във файла index.js. В този файл са още основните контроли за пускане/спиране, пускане отначало, както и необходимите функции за четене от файл и създаване на параграфи от текст, които на свой ред да бъдат анимирани. Четенето на файл с имената на рефератите работи посредством XMLHttpRequest - GET, към локален файл с релативен път ./json/data.json. Това, разбира се, работи единствено когато системата е пусната на сървър, а не локално. Приема се като поведение по подразбиране. Другият вариант за четене на файл е чрез избран от потребителя файл и FileReader.

За опцията “Голям екран” се грижи файлът fullscreen.js, като също в него се поема грижата за това контролите да могат да бъдат използвани дори на голям екран, както и големината на шрифта да се уголеми, за да може да съответства на по-големият екран.

За аудио файлът и контролите за аудио отговаря файла audio.js. Аудио файлът се избира от потребителя и бива прочетен чрез FileReader.

За прилагането на стилове върху текста на анимацията отговаря файлът animation-styles.js. Той прочита данните от config.json файл, ако такъв има, и ги нанася върху формата за стилове. Отговаря за прилагането на стиловете от формата върху текста от анимацията след натискане на съответния бутон.

Файлът comments.js отговаря за добавянето на коментари от потребителя.

Файлът save-animation.js служи за взимането на информацията за анимацията, което включва data.json, стила на анимациите и сложения аудио файл и я подготвя да бъде обработена от save-animation.php. Преди да се осъществи връзката с php файла, се проверява дали споменатите данни са въведени.

Файлът save-animation.php има работата да създаде папка, в която да се съхраняват файлове с гореспоменатите данни и да добави нов ред в базата данни, който съдържа името на анимацията и имената на новите файлове. За добавянето в базата данни, кода използва функция създадена във файла db\_save\_animation.php.

Файлът db\_save\_animation.php съдържа функция, която осъществява връзката с базата данни и изпраща заявка за добавяне на нов ред. Използва се файлът db\_util.php в папка db за създаването на клас DB.

Файлът db\_util.php съдържа class db, с който се създава интерфейс за достъпването на база данни.

Файлът list-animations.js служи за попълване на списък със запазени анимации, като се свързва с list-animations.php за да вземе имената на запазените анимации.

Файлът list-animations.php изпраща заявка към базата данни, с която се вземат имената на всички запазени анимации.

Файлът load-animation.js служи за това да приложи данните в папката на избрана анимация върху страницата. Тя го прави като се свърза с php файловете в папка load-animations в папка backend.

Папка load-animations:

Файлът get-animation-data.php служи за изпращането на заявка към базата данни, с която се взима информация за избраната анимация.

Останалите get файлове за четене на данните в запазените файлове в папката на анимация.

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

**9.1 Таня Сейкова**

* Визуализация на основната анимацията.
* Контролни бутони за анимацията.
* Форма за редактиране на стиловете на анимацията.
* Оформление и на страницата.

**9.2 Георги Чобанов**

* Работа по създаването, свързването и използването на базата данни.
* Работа с конфигурационния файл.
* Работа по опцията за пълен екран.

**9.3 Катерина Павлова**

* Работа с аудио аспекта на системата.
* Работа по секцията с коментарите.
* Работа по корекция на грешки и съобщения към потребителя.

**9.4 Ограничения**

Чрез визуализиране на анимацията единствено посредством CSS, направата на контроли за преминаване напред / назад е изключително трудно и неефективно.

**9.5 Бъдещо разширение**

* Добавяне на коментарите в базата данни
* Добавяне на различни видове анимации
* Интеграция с Puffin, за да се вземат рефератите с най-високи оценки и да се представят.

## 10. Какво научих

Сложни анимации могат да се правят единствено чрез CSS.

Много bash команди са интегрирани в PHP.

Как работят eventListener-и възможностите, които предоставят.

Как се използват темплейти.

Използването на jQuery доста улеснява навигирането в документа.

## 11. Използвани източници

<https://www.w3schools.com/> - множество статии за CSS, PHP и JavaScript

<https://developer.mozilla.org/en-US/> - множество статии за CSS и JavaScript

<https://css-tricks.com/snippets/css/star-wars-crawl-text/> - основно вдъхновение за направата на анимацията

Предал (подпис): ………………………….

/*62472, Таня Сейкова, СИ, група 4*/

Предал (подпис): ………………………….

/*62475, Георги Чобанов, СИ, група 5*/

Предал (подпис): ………………………….

/*62486, Катерина Павлова, СИ, група 2*/

Приел (подпис): ………………………….

/доц. *Милен Петров*/