**Имена:**  Мартин Танчев **фн:**  *81401***Начална година:** *20159* **Програма:** бакалавър, (КН) **Курс: 4**   
**Тема: Учител на акорди за песни  
Дата: 2020-01-06 Предмет: wwwTech2019\_20\_13ed\_KN\_winter имейл:** [**m1tanchev@gmail.com**](mailto:m1tanchev@gmail.com)

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

# ТЕМА: 29 Привю на акорди

## 1. Условие

## 2. Въведение

## 3. Теория

## 4. Използвани технологии

## 5. Инсталация и настройки

## 6. Кратко ръководство на потребителя

## 7. Примерни данни

## 8. Описание на програмния код

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

## 10. Какво научих

## 11. Използвани източници

Предал (подпис): ………………………….

/*фн, имена, спец., група*/

Приел (подпис): ………………………….

/доц. *Милен Петров*/

## 1. Условие

Привю на акорди представлява

## 2. Въведение

Условието на проекта беше интерпретирано по следния начин: основната идея е изграждане на приложение подобно на вече съществуващото Chordify, което позволява възпроизвеждане на видео от youtube, заедно с което се движат акордите съответстващи на музиката от видеото. Проектът предлага нещо подобно, но с доста лимитирана функционалност.

## 3. Теория

Привю(на англ. Privew) е възможността да се види нещо предварително, преди да бъде изисквано или преди да е достъпно. Така привю на акорди означава възможността да видим кой акорд трябва да изсвирим преди да е дошъл момента в който трябва да го изсвирим или като цяло да видим последователността от акорди за дадена песен.

## 4. Използвани технологии

PHP

MySQL

Първите две са използвани за да се рализира сървърната част от проекта – базите данни, които служат за запаметяване на линковете на песните и информацията към тях, които биват добавени от потребителите.

Plain JavaScript

HTML

CSS

SASS

YouTube Player API

Технологиите изброени по-горе са използвани за релизиране на визуалната част на проекта – потребителския интерфейс с, който потребителят си служи при използването на сайта.

## 5. Инсталация и настройки

Начин за достъп до системата е чрез инсталиране на актуалната версия на XAMP. Чрез тази програма могат да се пуснат сървър и MySQL база от данни, без необходимост от инсталация на други програми. Чрез phpMyAdmin могат да се създадат и редактират необходимите таблици за базата от данни. Файловете на проекта трябва да са поставени в конкретната папка на XAMP (htdocs) или да се конфигурира apache сървъра. Като начална страница да се използва index.php.

За да се подкара правилно платформата е нужно да бъде създадена нова база данни с името songsData. В директорията database се намира SQL скрипт с име songsData.sql който при импортване в базата данни създава нова таблица с информация за песни. Във файла db\_setup.php в директорията config се намират настройките за свързване с базата данни.

## 6. Кратко ръководство на потребителя

Кратко ръководство на потребителя е налично в отделен файл, който е приложен заедно с тази документация.

## 7. Примерни данни

Данните които потребителят ще въвежда в системата са: Имена на песни като като форматът им е като заглавия на песни в Ютуб. Най-често на първо място стои името на изпълнителя следвано от тире и след това самото име на песента: <изпълнител> - <песен>. Възможно е търсене само по част от името на изпълнителя или само по част от името на песента. Търсачката е чувствителна на малки и големи букви (case sensitive).

## 8. Описание на програмния код

Спазени са добрите практики за писане на код – функции, променливи и класове ясно описват значението си. Няма излишни и подвеждащи коментари, а само там където се налага да се въведе повече контекст за работата/значението на кода. Файловете са структурирани в поддиректории. Всяка страница включва хедър и футър които се намират в директорията templates. Стилизацията на всички файлове се намира във файла style.css който е генериран чрез sass файла style.scss като се използва sass компилаторът. Music.js е self-contained файл който предоставя класа Synthesizer за генериране на звук. Проектът има 3 страници – index.php представлява board страница визуализираща всичи песни от базата данни, search.php поддържа логиката и визуализирането на search функцията като са използвани функциите от php които имат известна защита от атака вурху базата данни и play.php – проложението за привю на акорди. При избиране на песен се сменя локацията на документа (с JavaScript) като чрез url се подава youtube id(11 символа 64 бита). Това съответства на GET метод. В play.php се проверява дали съществува ред в базата с това id. При налична информация за песента се извлича реда и се подава на JavaScript, който зарежда информацията за песента – необходими са само bpm и chords. При зареждане на страницата първо се изисква скрипътъ UI.js – там е основният клас контролиращ изгледа на страницата и функционалността за аутоскрол на акордите. След това се изисква youtube API-то. При успешно заредено видео се създава нов обект от класа UI – който пази youtube player-a и информация за песента. Име на песента и дължина на песента се взимат от youtube API-то. Ако съществува масив chords от базата и броя акорди съвпада с броя акорди изчислен на база дължината на клипчето и bpm (т.е акордите са валидни), то се тогава директно се визуализират. В противен случай в конзолата се извежда съобщение за грешка и се генерират произволни акорди (това е направено за да се тества системата с dummy данни. Функционалността за аутоскрол е имплементирана в метода handleReelScroll. На този метод се подава id на акорд изчислено на база текущото време на песента. Ако разликата на текущото id и предходното се разминава с повече от 4 – то тогава сме превъртели видеото с толкова че smooth-scroll-а вече да е много бавен. На всеки 4 бийта превъртаме скрола с толкова пиксела колкото заемат 4 reels. Функцията playingHandler се извиква когато статуса на youtube клипчето е playing. В тази функция извикваме setInterval с период от 10мс (достатъчен за най-бързите песни) а в setInterval пресмятаме индекса на текущия акорд.

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

Приносите на студента се изразяват в под формата на по-голямата функционалност реализирана в проекта и която е описана в точка 6. Единствено е използвано Youtube Player API, което предоставя възможността за лесно вграждане на видео в iframe таг, получаване на заглавие на видео, както и информация за текущото времето, което стои и в основата на приложението. Ограничение на системата е това, че не могат да се променят акордите от потребител, но това ще бъде разгледано във възможностите за бъдещо разширение. Приложението разчита на правилно темпо съответстващо на песента, както и песента да започва с отброяване на темпото. Това естествено налага ограничение на клипчетата които могат да се използват. Например изпълнения на живо не са подходящи тъй като началото на клипа не винаги е началото на самата песен. Друго ограничение наложено от тези допускания е, че хората не могат да поддържат перфектно темпо, което води до разминаване в показания акорд от приложението и реалния акорд.

Възможностите за бъдещо разширение са многобройни, като ще споменем само някои от тях. Естествено, може да се подобри изгледа на сайта, използваната стилизация е на доста базово ниво, защото е заложено на функционалност, отколкото на външен вид. Подобряване на поддържаните акорди – да се добавят не само минорни и мажорни акорди, а всички популярни в уестърн музиката – sus2, sus4, 7, 9, dim и т.н. За това може да се използва вече създадения формат – разделителя да е +. Много полезно разширение би било със всеки акорд да се показва как се свири на китара или пиано.

Съществува софтуер за автоматично извиличане на темпо, хармония и мелодия от звук, така че този софтуер може да се обвърже със текущия сайт. Могат да се въведат потребители, така че всеки потребител да създава и споделя плейлисти, както и да маркира песни като любими. Друга идея е подобно на субтири(караоке) заедно с акордите да се движи и текста на песента.

## 10. Какво научих

Преди да започна да се занимавам с с разработването на темата имах главно теоретични знания по някои от технологии които са ползвани – PHP, Javascript, и много малко опит със HTML, CSS, MySQL придобит на упражненията във ФМИ. Ползвайки източници от интернет и код за справка от упражнения се научих как да боравя с бази от данни – запазване и визуализиране на информация. Полезен опит беше и това че използвах API, което е добре документирано. Това ми показа колко важно е всъщност да пишем интуитивни и добре замислени интерфейси. Процесът на разработка беше доста интересен и приятен за мен, не само защото научавах нови неща, но и защото изпитвах удоволетворение от това да създам работещ продукт. Ако мога да променя нещо, то това би било процеса на разработка на архитектурата на проекта – ясна визия и връзки между елементите биха направили имплементацията и бъдещите подобрения по-лесни. При разработка на алгоритъма за автоскрол и показване на текущия акорд, се запознах със системи за автоматично извличане на темпо, мелодия и хармония от песните, което може би е следващата стъпка в разработка на проекта.

## 11. Използвани източници

<https://www.sitepoint.com/use-json-data-fields-mysql-databases/>

<https://developers.google.com/youtube/iframe_api_reference>

<https://marcgg.com/blog/2016/11/01/javascript-audio/>

<https://www.php.net/manual/en/>

<https://stackoverflow.com/>