

Имена: Захари Маринов, Богдан Миронов, Владимир Кръстев

фн: 1MI0800059, 0MI0800002, 2MI0800068

Начална година: 2025

Програма: бакалавър, (КН)

Курс: 4

Тема: ASCII акорди - редактор и проиграване

Дата: 2025.01.29

Предмет: w25prj_KN_final

имейл:

zamarinov@uni-sofia.bg

vkrastev@uni-sofia.bg

0mi0800002@g.fmi.uni-sofia.bg

преподавател: доц. д-р Милен Петров

ТЕМА: w25/ 45.1

1. Условие

Създаване на акорди (за различни инструменти), импорт и експорт в ASCII (и друг формати csv, json) видове

<https://tabs.ultimate-guitar.com/tab/deep-purple/mistreated-chords-993596> F#m E/G# D/A

Bm F#m E/G# e|---2---2---2---2---|---2---2---2---2---|---2---2---2---2---|

B|---2---2---5---5---|---3---3---3---3---|---2---2---5---5---|

G|---2---2---4---4---|---2---2---4---4---|---2---2---4---4---|

D|---4---4---2---2---|---4---4---4---4---|---4---4---2---2---|

A|---4---4---2---2---|---0---0---2---2---|---4---4---2---2---|

E|---2---2---4---4---|-----|---2---2---4---4---|

Редактор, плейър, управление на файловете (импорт/експорт, архивиране, споделяне с външна система, и четене на данните от външна система

2. Въведение – извличане на изисквания

Проектът „Редактор и плейър на китарни таблатури“ е разработен с цел да осигури удобен и функционален инструмент за музиканти и любители на китарата.

Приложението предлага възможност за създаване, редактиране и възпроизвеждане на таблатури, което го прави полезно за начинаещи музиканти, търсещи интуитивен начин за работа с музикални ноти.

Роли

Приложението обслужва само един тип потребител StandardUser, който има право да създава и редактира таблатури, да експортира таблатурите си в различни формати, да импортира таблатури и да проиграва таблатурите.

Функционални изисквания

- Създаване на таблатури чрез визуален редактор.
- Възможност за въвеждане и редактиране на акорди
- Възпроизвеждане на таблатури с реалистични звуци на китара.
- Синхронизация между визуализацията и звуковото възпроизвеждане.
- Опция за запазване и експортиране на таблатурите във формат, удобен за споделяне или разпечатване.
- Интуитивен потребителски интерфейс, който да улесни навигацията и работата с таблатури.

Нефункционални изисквания

- Приложението трябва да бъде съвместимо с най-новите версии на популярни уеб браузъри (Chrome, Firefox, Edge).
- Осигуряване на бързо зареждане и безпроблемна работа дори при по-големи таблатури.
- Интуитивен и адаптивен интерфейс, който да работи на устройства с различни резолюции (настолни компютри, лаптопи, планшети).
- Възможност за лесно разширяване и надграждане на функционалността.
- Приложението трябва да бъде устойчиво на често срещани грешки, като например невалидни входни данни или загуба на връзка със сървъра.

Ползи от реализацията

- **Образователна стойност:** Помага на начинаещи китаристи да усвоят по-добре техниките за свирене, като им предоставя ясна визуализация на таблатурите и възможност да чуят как звучат акордите.
- **Практичност:** Предоставя лесен и достъпен начин за създаване и тестване на таблатури, без нужда от специализиран софтуер.
- **Гъвкавост:** Приложението може да се използва както за обучение, така и за създаване на професионални таблатури.
- **Интерактивност:** Възпроизвеждането на таблатури с реалистични звуци помага на музикантите да проверят как точно звучат техните акорди, което повишава качеството на тяхната работа.

С този проект се поставя основата за създаване на иновативен инструмент, който може да бъде надграждан в бъдеще, за да предложи още повече функционалности и да отговори на нуждите на различни типове потребители.

3. Теория – анализ и проектиране на решението

Архитектура на приложението

Приложението следва модел на разделение на отговорностите между Frontend, Backend и база данни. Това гарантира яснота и лесно разширяване на функционалността.

1. Frontend:

Frontend частта е изградена с HTML, CSS и JavaScript. Основните файлове включват:

- **HTML файлове:**
 - `index.html`: Главната страница на приложението.
 - `login.html` и `register.html`: Страници за вход и регистрация.
- **CSS файлове:**
 - `auth.css`: Стили за страниците за вход и регистрация.
 - `editor.css`: Специфични стилове за редактора на таблатури.
 - `styles.css`: Глобални стилове за цялото приложение.
- **JavaScript файлове:**
 - `main.js`: Управлява взаимодействието между потребителя и приложението.
 - `editor.js`: Реализира функционалностите на редактора на таблатури.
 - `export.js`: Отговаря за експортиране на таблатури във външни формати.
 - `auth.js`: Отговаря за автентикиране на потребителя
 - `logout.js`: Отговаря за излизането от профила на потребителя
 - `save.js`: Отговаря за запазване на таблатури в профила на потребителя
 - `load.js`: Отговаря за зареждане на таблатури от профила на потребителя

2. Backend:

Backend частта е реализирана с PHP и комуникира с Frontend чрез REST API. Основните файлове са:

- `db.php`: Управлява връзката с базата данни.
- `user_handler.php`: Съдържа логика за управление на потребителите (регистрация, вход).
- `melody_handler.php`: Обработва данни, свързани с таблатурите (създаване, промяна, изтриване).
- `endpoints.php`: Дефинира API маршрутите и обработва заявките от Frontend.

3. База данни:

Базата данни е създадена с MySQL. Основни таблици:

- **users:** Съдържа данни за потребителите (ID, имейл, хеширана парола и др.).
 - **tabs:** Запазва информация за таблатурите (име, автор, съдържание).
 - **chords:** Съдържа информация за конкретните акорди в таблатурите.
-

Модули и компоненти

1. Регистрация и вход:

- HTML формуляри за вход и регистрация (`login.html`, `register.html`).
- Силове, специфични за тези страници (`auth.css`).
- PHP скриптове за управление на потребителските сесии (`user_handler.php`).

2. Редактор на таблатури:

- Създаване на акорди и визуализация чрез интерактивен интерфейс (`editor.js`, `editor.css`).
- Записване на таблатурите в базата данни чрез API заявка.

3. Възпроизвеждане на звуци:

- Използване на JavaScript за възпроизвеждане на предварително записани аудио файлове, базирани на въведените акорди.

4. Експортиране и споделяне:

- Скрипт за експортиране на таблатурите в текстови формат (`export.js`).
- Споделяне на таблатури чрез уникален линк.

Процес на декомпозиция

Приложението е разделено на следните основни части:

1. Потребителски интерфейс (UI):

- HTML за структурата на страниците.
- CSS за стилове и визуални ефекти.

2. Функционална логика:

- JavaScript за обработка на потребителски действия (редактиране на таблатури, възпроизвеждане на звуци).
- PHP за обработка на заявки и връзка с базата данни.

3. Данни и съхранение:

- MySQL база данни за съхранение на потребителски данни и таблатури.

4. Използвани технологии

Езици за програмиране

1. **HTML5:**
 - Използва се за структурата на интерфейса (страници като `index.html`, `login.html`, `register.html`).
 2. **CSS3:**
 - Отговорен за дизайна и стиловете на приложението (`auth.css`, `editor.css`, `styles.css`).
 3. **JavaScript (ES6):**
 - Логика на фронтенда, включваща манипулация на DOM, обработка на събития и комуникация с бекенд (`auth.js`, `editor.js`, `export.js`, `main.js`).
 4. **PHP:**
 - Логика на бекенда, обработка на заявки и взаимодействие с базата данни (`db.php`, `endpoints.php`, `melody_handler.php`, `user_handler.php`).
-

База данни

1. **MySQL:**
 - Релационна база данни, използвана за съхранение на потребителски данни, акорди и мелодии.
 - Структурата на базата е зададена в SQL скрипта `ASCII.sql`.
-

Фреймуърк и библиотеки

1. **Native JavaScript:**
 - Използва се за обработка на логика, без да се използват външни библиотеки.
 2. **Fetch API:**
 - За изпращане на HTTP заявки от JavaScript към PHP бекенда.
-

Среда за разработка

1. **Локален сървър:**
 - Може да се използват:
 - **XAMPP** (Apache + PHP + MySQL).
 - **WAMP** или **MAMP** за алтернативи.
-

Операционна система

1. **Windows** или **macOS** (локално тестване и разработка).
-

Версии

- **HTML5, CSS3 и ES6** – Поддържани от съвременни браузъри.
 - **PHP >= 7.4** (за съвместимост с нови функции и библиотеки).
 - **MySQL >= 5.7.**
-

Технологични решения

1. **Визуализация и интерактивност:**
 - `index.html` е свързан с JavaScript модули за динамично създаване на акорди и таблатури.
 - CSS добавя семпли и функционални дизайни.
2. **Обработка на данни:**
 - PHP осигурява API за регистрация, вход, управление на акорди и мелодии.
 - MySQL съхранява данните.
3. **Експорт на таблатури:**
 - Реализира се с JavaScript (създаване на текстов файл директно във фронтенда).

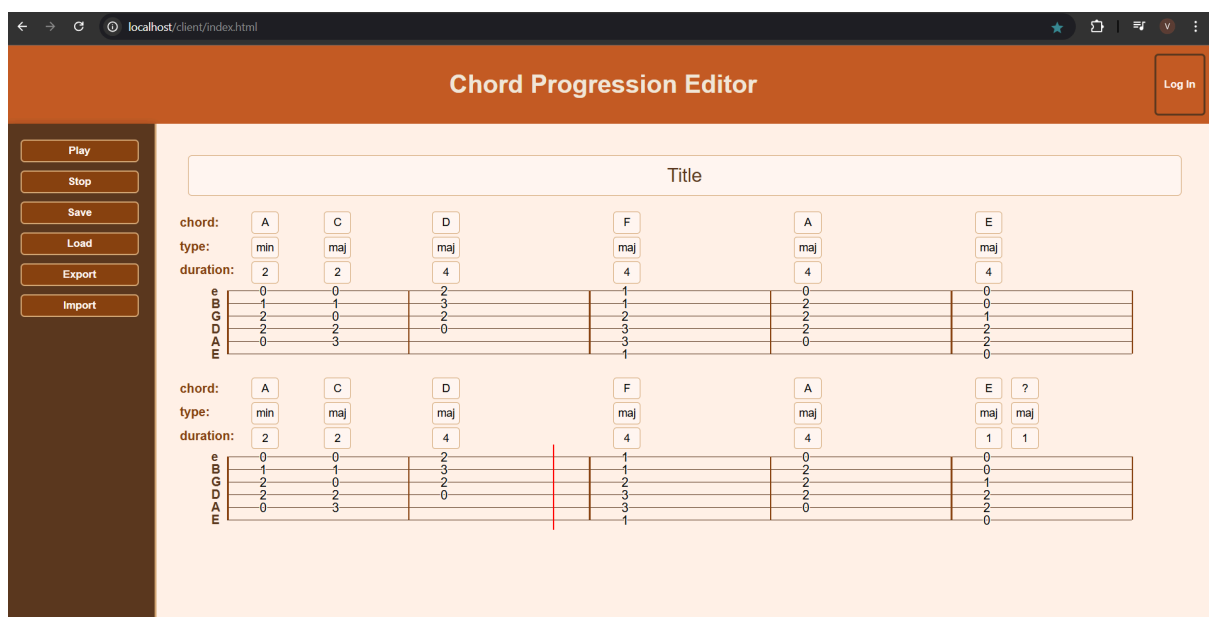
5. Инсталация, настройки и DevOps

1. **Инсталиране на XAMPP**
 - Свалете XAMPP от apachefriends.org и го инсталирайте.
2. **Стартиране на сървърите**
 - Отворете XAMPP Control Panel и стартирайте **Apache** и **MySQL**.
3. **Импорт на базата данни**
 - В phpMyAdmin (бутон **Admin** до MySQL) отидете на **Import**.
 - Качете [server/resources/ASCII.sql](#) и натиснете **Go**.
4. **Поставяне на уеб файловете**
 - Копирайте файловете на уеб приложението в [xampp/htdocs](#).
5. **Достъп до приложението**
 - Отворете браузър и въведете <http://localhost:80>.
6. **Примерни данни за вход**
 - **Email:** [test@test.com](#)
 - **Парола:** [123456abc](#)

6. Кратко ръководство на потребителя

Основни функционалности

- **Без регистрация:** Добавяне, проиграване и експортиране на акорди.
- **Добавяне на акорд:**
 - Изберете основния тон на акорда.
 - Автоматично се появява поле за следващия акорд.
 - Задайте тип (мажорен/минорен) и продължителност.
- **Проиграване на таблатура:**
 - Натиснете **Play**, за да чуете акорди.
 - Натиснете **Stop**, за да паузирате.

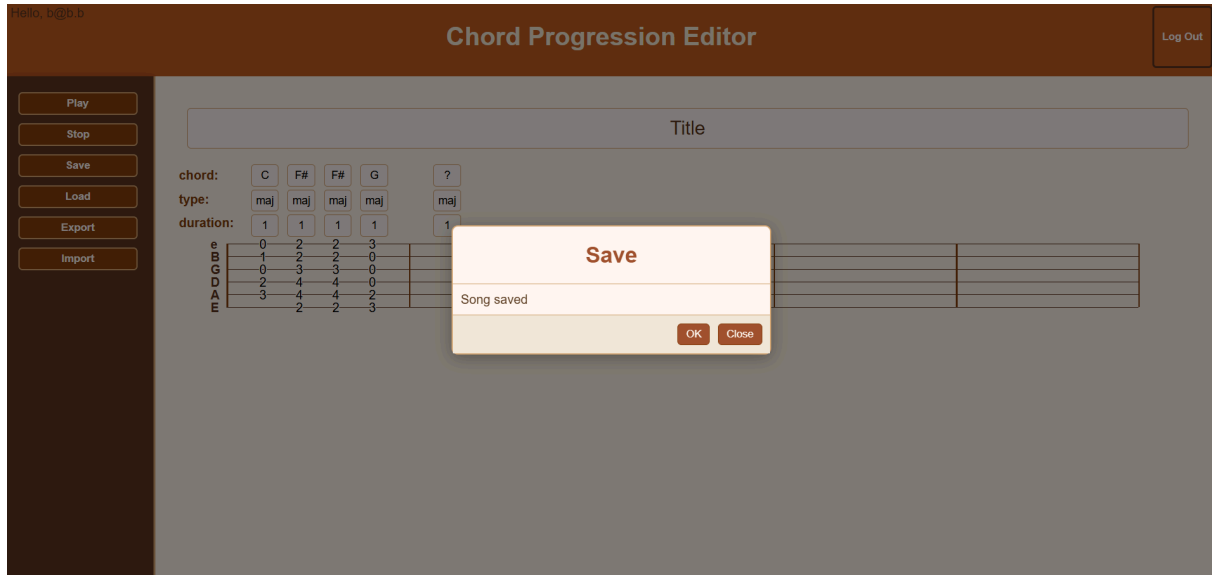


- Редакция не е позволена по време на проиграването.
- **Експортиране:**
 - Натиснете **Export**.
 - Потвърдете изтеглянето в поп-уп прозорец.

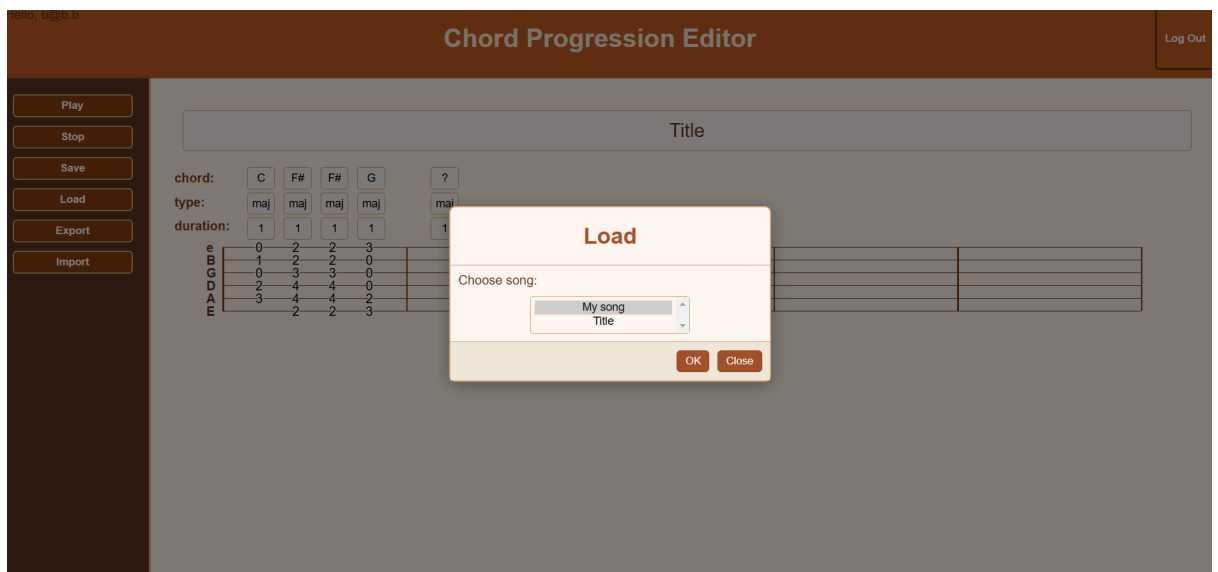
Запазване и зареждане на таблатури

(Изисква влизане в профил)

- **Save** – запазва таблатурата в профила.

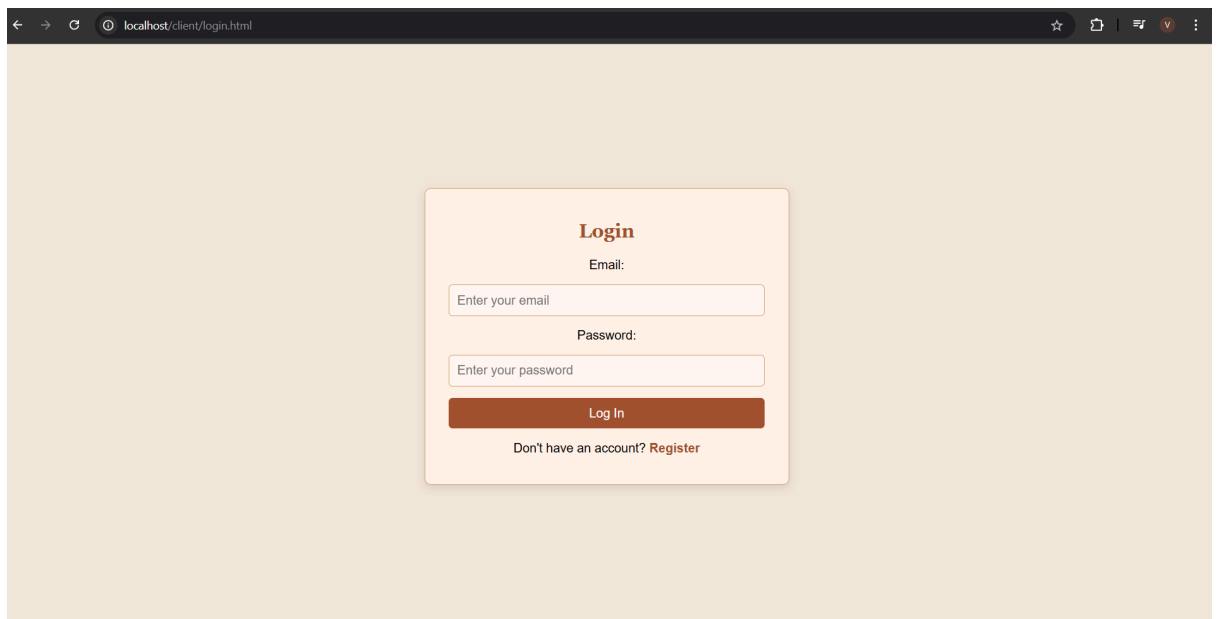


- **Load** – зарежда запазена таблатура.



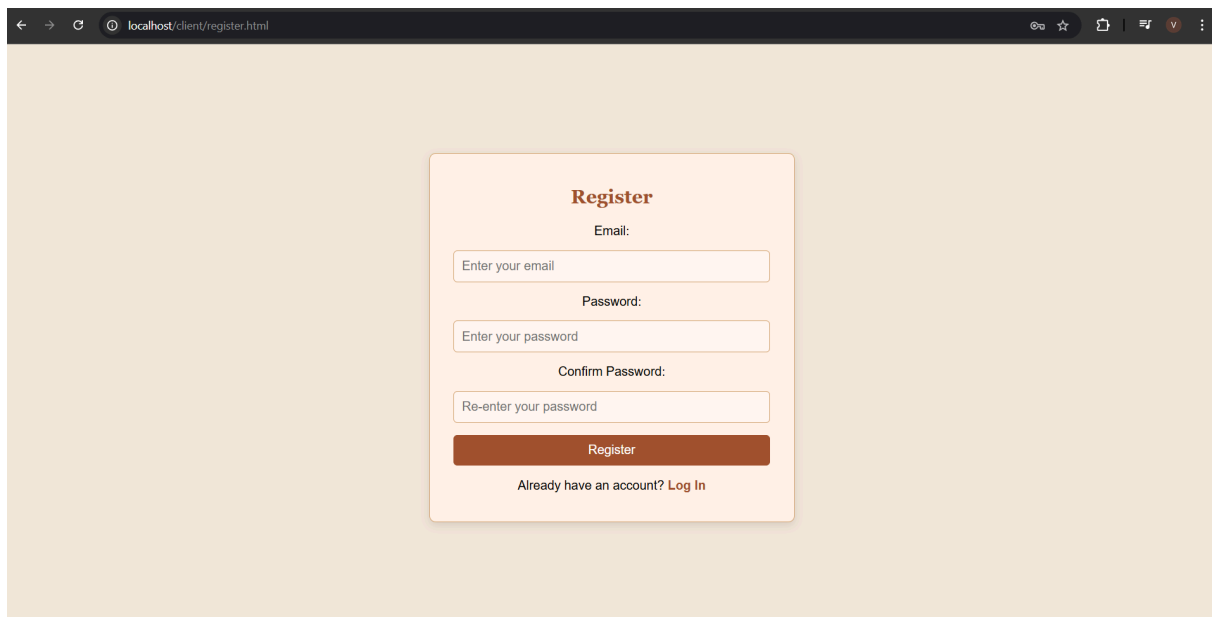
Вход и регистрация

- **Log in:** Въведете имейл и парола.



The screenshot shows a web browser window with the address bar displaying 'localhost/client/login.html'. The main content area features a light beige background with a central white login form. The form is titled 'Login' in a bold, dark brown font. Below the title, there are two input fields: 'Email:' with a placeholder 'Enter your email' and 'Password:' with a placeholder 'Enter your password'. A dark brown 'Log In' button is positioned below the password field. At the bottom of the form, there is a link that says 'Don't have an account? Register'.

- **Register:** Въведете имейл и парола два пъти за регистрация.



The screenshot shows a web browser window with the address bar displaying 'localhost/client/register.html'. The main content area features a light beige background with a central white register form. The form is titled 'Register' in a bold, dark brown font. Below the title, there are three input fields: 'Email:' with a placeholder 'Enter your email', 'Password:' with a placeholder 'Enter your password', and 'Confirm Password:' with a placeholder 'Re-enter your password'. A dark brown 'Register' button is positioned below the confirm password field. At the bottom of the form, there is a link that says 'Already have an account? Log In'.

- След успешен вход/регистрация се връщате в главната страница.

7. Примерни данни

1. Примерни потребителски акаунти

- i. Email: test@test.com
- ii. Парола: 123456abc

2. Примерни акорди за таблатури

При редактиране на таблатури можете да използвате следните настройки:

Акорд	Тип	Продължителност
A	min	2
C	maj	2
D	maj	4
F	maj	4

8. Описание на програмния код

1. **auth.js** (Модул за автентикация)

Този файл обработва регистрацията и входа на потребителите, изпращайки заявки към бекенд API.

Основни функции:

- **onsubmit** обработка за вход:
 - Събира email и парола от формата.
 - Изпраща POST заявка към API-то за вход.

```
loginForm.onsubmit = async (e) => {
  e.preventDefault();
  const email = document.getElementById('loginEmail').value;
  const password = document.getElementById('loginPassword').value;

  const requestData = {
    username: email,
    password: password
  };

  try {
    const response = await fetch(`${API_URL}?action=login`, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify(requestData),
    });
  }
```

- **onsubmit** обработка за регистрация:
 - Събира email и парола от формата.
 - Изпраща POST заявка към API-то за вход

2. **editor.js** (Модул за редактиране на таблатури)

Това е сърцето на приложението, което предоставя интерфейс за създаване и редактиране на акорди.

Основни класове и функции:

- **Chord**: Представява един акорд с неговите свойства (тип, продължителност и др.).
- **Data**: Управлява всички акорди и предоставя данни за експорт.
- **displayEditor**: Рендира акорди и таблатури на екрана.

```

export function displayEditor() {
  editor.innerHTML = ''
  for (let i = data.chords.length - 1; i >= 0; i--) {
    if (data.chords[i].chord === -1) {
      data.chords.splice(i, 1)
    }
  }
  data.chords.push(new Chord())

  resetLine();

  let offset = 0, startIndex = 0, startOffset = 0;
  const totalPositions = divisionsPerRow * 4
  for (let i = 0; i < data.chords.length; i++) {
    offset += data.chords[i].duration
    if (offset >= totalPositions) {
      editor.appendChild(Row(startOffset, startIndex, i))
      offset -= totalPositions;
      startOffset = offset;
      startIndex = i + 1;
    }
  }
  if (startIndex < data.chords.length) {
    editor.appendChild(Row(startOffset, startIndex, data.chords.length - 1))
  }
}

```

- **playChord**: Възпроизвежда звука на акордите при натискане на бутона **Play**

3. **export.js** (Модул за експортиране)

Този файл отговаря за създаването на ASCII файл с таблатури.

Основни функции:

- **toASCIICHords**:
 - Превръща данните за акорди в ASCII формат.
 - Създава текстов файл за изтегляне.

```

export function toACIChords({chords, title}) {
  file = title + '\n';
  let position = 0, startIndex = 0, startPosition = 0;
  for (let i = 0; i < chords.length; i++) {
    position += chords[i].duration
    if (position >= positions) {
      for (let j = 0; j < 6; j++) {
        writeLine(startPosition, chords.slice(startIndex, i + 1), j)
      }
      file += '\n'
      position -= positions;
      startPosition = position;
      startIndex = i + 1;
    }
  }
  if (startIndex < data.chords.length) {
    for (let j = 0; j < 6; j++) {
      writeLine(startPosition, chords.slice(startIndex), j)
    }
  }
}

```

- **exportPopupContent**: Показва изскачащ прозорец за потвърждение на експорта.

4. main.js (Главен модул)

Централно място за управление на събития и взаимодействия между модулите.

Основни функционалности:

- Управление на бутони за възпроизвеждане, експортиране и импорт.
- Свързва **editor.js** и **export.js**.

```

exportButton.onclick = () => {
  displayPopup(exportPopupDisplay, exportConfirm)
}

```

- Обработка действията на потребителя като стартиране на възпроизвеждане и затваряне на изскачащи прозорци

5. db.php (Модул за връзка с базата данни)

Файлът управлява свързването с MySQL базата данни.

Основни функции:

- Свързване с базата данни:
- Изпълнение на SQL заявки и обработка на грешки.

6. `endpoints.php` (REST API)

Този файл маршрутизира заявки към правилните обработващи модули.

Основни функции:

- Разпределя действия като "login", "register", "add_chord":

7. `melody_handler.php` и `user_handler.php`

- `melody_handler.php`:
 - Управлява CRUD операции за мелодии и акорди.
- `user_handler.php`:
 - Обработка регистрация и вход.

8. `ASCII.sql` (База данни)

- Създава таблици като:
 - `users` – Съхранява потребителски данни.
 - `melodies` – Съхранява мелодии и акорди.

9. CSS файлове

- `auth.css`: Дизайн за страниците за вход и регистрация (топли тонове, лесен за четене интерфейс).
- `editor.css`: Стиллове за акордите и редактора.
- `styles.css`: Общи стиллове за структурата на приложението

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

Ограничения на текущата версия

1. **Липса на роля "Администратор":**
 - В момента няма функционалност за управление на потребители от страна на администратор. Например, изтриване на потребители или одобрение на акаунти.
2. **Ограничени функционалности за таблатури:**
 - Редакторът позволява създаване само на основни акорди (мажорни/минорни). Няма поддръжка за сложни акорди като септакорди (7th) или добавени ноти.
3. **Липса на мобилна оптимизация:**
 - Интерфейсът не е напълно адаптивен за мобилни устройства, което ограничава потребителите, които искат да използват приложението на телефони или таблети.
4. **Липса на вградени аудио ефекти:**

- В момента звукът на акордите се възпроизвежда чрез предварително записани аудио файлове. Няма възможност за динамично генериране на звук в реално време.
 - 5. **Ограничени тестови функционалности:**
 - Няма вградена система за автоматизирано тестване на бекенд функционалностите или таблатурите.
 - 6. **Сигурност:**
 - Вход и регистрация обработват паролите на клиентската страна без SSL криптиране (изисква HTTPS за по-висока сигурност).
-

Възможности за бъдещо развитие

1. **Добавяне на администраторски функции:**
 - Разработване на администраторски интерфейс за управление на потребители (блокиране, изтриване, преглед на активност).
2. **Разширение на редактора:**
 - Поддръжка на по-сложни акорди като:
 - Септакорди (7th), увеличени/намалени акорди (augmented/diminished).
 - Възможност за промяна на темпо и метрика.
3. **Мобилна адаптация:**
 - Оптимизиране на интерфейса за мобилни устройства чрез адаптивни стилове (CSS Media Queries).
4. **Интеграция на уроци:**
 - Добавяне на готови уроци за начинаещи, които да демонстрират основни акорди и таблатури.
5. **Реално време и колаборация:**
 - Разработване на система за работа в реално време, където множество потребители могат да създават таблатури заедно (подобно на Google Docs).
6. **По-добро управление на звука:**
 - Интегриране на библиотека за синтезиране на звук (например Tone.js), което ще позволи динамично възпроизвеждане на акорди без предварително записани файлове.
7. **Подобряване на сигурността:**
 - Прилагане на HTTPS и хеширане на пароли с модерни алгоритми (например bcrypt).
 - Защита срещу SQL инжекции чрез използване на подготвени заявки в PHP.
8. **Импортиране и експортиране в различни формати:**
 - Поддръжка на формати като MIDI за по-лесна интеграция с музикални програми.
 - Възможност за директно споделяне на таблатури чрез линкове.

9. Интеграция с облачни услуги:

- Съхранение на таблатури в облака, за да бъдат достъпни от всяко устройство.
- Интеграция с платформи като Google Drive или Dropbox.

Принос на студентите

- **Богдан Миронов:** Разработване на бекенд логиката (файловете `db.php`, `user_handler.php`, `melody_handler.php`), `login`, `register` и помощ със стилизацията
- **Захари Маринов:** Реализиране на редактора, експорт, запазване и зареждане в базата данни(`editor.js`, `main.js`).
- **Владимир Кръстев:** Проиграване на акорди и `logout(editor.js, main.js, logout.js)`.

10. Какво научих (най-важните неща, които сте научили по време на курса и при разработването на проекта-за всеки студент)

Владимир Кръстев

По време на курса и разработването на проекта разширих значително познанията си в уеб разработката и придобих увереност в изграждането на функционални и интерактивни приложения.

Усъвършенствани технологии и подходи

- Подобрени умения в **HTML, CSS и JavaScript**, особено при изграждането на динамични интерфейси.
- Работа с **възпроизвеждане на звуци в уеб приложения** и интеграция на аудио с визуални елементи.
- Разбиране на **синхронизацията на звукови ефекти с таблатури** за по-добро потребителско изживяване.

Технически предизвикателства и решения

- Разработване на **алгоритъм за синхронизация** между визуализацията на таблатурите и тяхното възпроизвеждане.
- Оптимизиране на кода чрез **разделение на логиката и визуализацията** за по-лесна поддръжка и разширяване.

Умения за работа в екип и обратна връзка

- Развитие на **структуриран и модулен код**, който да бъде лесен за четене и поддръжка.
- Разбиране на значението на **конструктивната обратна връзка** и подобряване на решенията въз основа на нея.

Този проект беше ценен опит, който ми позволи да приложа и усъвършенствам както техническите си знания, така и уменията за работа в екип и ефективна разработка на уеб приложения.

Захари Маринов

Проектът ми помогна да затвърдя знанията си по **уеб разработка**, комуникация между клиент и сървър по **HTTP** протокол и използване на **програмния интерфейс на брауъра**.

Усъвършенствани технологии и подходи

- Прилагане на **компонентен стил** при разработката на уеб приложения.
- Подобрени умения в **CSS стилизиране** и **структуриране на семантичен HTML**.

Придобити практически умения

- Опит в **екипна и индивидуална работа**, което спомогна за по-добра организация на кода.
- Разбиране на принципите на **професионалната разработка** и приложението им в реални проекти.

Този проект ми даде ценен практически опит, който ще ми бъде от полза при бъдещо участие в разработването на уеб приложения в професионална среда.

Богдан Миронов

Проектът ми помогна да развия знанията си по **уеб разработка**. В писането му затвърдих **уменията си по PHP, взаимодействия между клиент и сървър по HTTP и стилизация чрез CSS**. Научих как се използват **бисквитки и сесии**.

Усъвършенствани технологии и подходи

- Подобрени умения в **PHP**, включително обработка на **логин информация и управление на сесии**.
- Научих как да **използвам и манипулирам бисквитки (cookies)** за съхраняване на информация за потребителите.

Придобити практически умения

- Опит в **екипна и индивидуална работа**, което спомогна за по-добра организация на кода.
- Опит в работа с **автентикация и управление на потребителски данни**.

11. Използвани източници

<https://github.com/mtgeorgiev/www-kn-2024>

<https://github.com/alexanderStefanovS/web-technologies-course-fmi>

<https://learn.fmi.uni-sofia.bg/course/view.php?id=10491>

<https://developer.mozilla.org/en-US/>