**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**„Київський політехнічний інститут імені Ігоря Сікорського”**

**Факультет прикладної математики**

“ЗАТВЕРДЖЕНО”

Керівник роботи

\_\_\_\_\_\_\_\_\_\_ Є.С.Сулема

“\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_ 2020 р.

**ВЕБ-ДОДАТОК ДЛЯ СТВОРЕННЯ МЕЛОДІЙ ЗА ДОПОМОГОЮ ВІРТУАЛЬНОГО ПІАНІНО**

**Пояснювальна записка**

ПЗКС.045440-03-81

|  |  |
| --- | --- |
| \_\_” \_\_\_\_\_\_\_\_\_\_\_\_ 2011 р. | Виконавець:  \_\_\_\_\_\_\_\_\_\_ Городченко А.В. |

2020

**ЗМІСТ**

[СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ 3](#_Toc58520479)

[ВСТУП 4](#_Toc58520480)

[1. АНАЛІЗ МОВ ПРОГРАМУВАННЯ ТА ТЕХНОЛОГІЙ 5](#_Toc58520481)

[1.1 Мова програмування JavaScript 5](#_Toc58520482)

[1.2 Мова програмування Python 6](#_Toc58520483)

[1.3 Опис фреймворку ReactJS 7](#_Toc58520484)

[1.4 Опис фреймворку Python Flask 8](#_Toc58520485)

[2. РОЗРОБЛЕННЯ ВЕБ-ДОДАТКУ СТВОРЕННЯ МЕЛОДІЙ ЗА ДОПОМОГОЮ ВІРТУАЛЬНОГО ПІАНІНО 9](#_Toc58520486)

[2.1 Аналіз вимог до розробки веб-сервісу 9](#_Toc58520487)

[2.2 Опис вимог до розробки веб-сервісу 10](#_Toc58520488)

[2.3 Архітектура веб-додатку 11](#_Toc58520489)

[2.4 Вміст та дизайн сторінок веб-додатку 12](#_Toc58520490)

[3. ОПИС РОЗРОБЛЕНИХ АЛГОРИТМІВ ТА ПІДПРОГРАМ 13](#_Toc58520491)

[3.1 Особливості реалізації модулю сервера 13](#_Toc58520492)

[3.2 Особливості реалізації модулю клієнтського інтерфейсу 14](#_Toc58520493)

[АНАЛІЗ РОЗРОБЛЕНОГО WEB-РЕСУРСУ 15](#_Toc58520494)

[4.1 Процес тестування 15](#_Toc58520495)

[4.2 Рекомендації щодо подальшого вдосконалення 16](#_Toc58520496)

[ВИСНОВКИ 17](#_Toc58520497)

[СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ 18](#_Toc58520498)

**СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ**

**HTTP** – Hyper Text Transfer Protocol (протокол передачі гіпертекстових документів);

**API** – прикладний програмний інтерфейс, набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення;

**HTML** – HyperText Markup Language, мова розмітки документів у всесвітній павутині

**JS** – JavaScript, мова програмування, найбільш широке застосування знаходить в браузерах як мова сценаріїв для додання інтерактивності веб-сторінок

`

**ВСТУП**

Музика супроводжує людину ще з давніх часів і поступово розвивається нарівні з людством. Вона відкриває нові кордони світогляду і допомагає виражати свої почуття і доносити їх іншим людям.

Піаніно і рояль – вічні цінності. Інструмент – наш музичний друг, наш емоційний якір в стрімкому потоці часу. Жоден інший інструмент, мабуть, не володіє такою універсальністю, як фортепіано. Йому під силу будь-який стиль – від класики до джазу, року або поп-музики, в залежності від настрою і смаку виконавця.

Гра на фортепіано приносить задоволення, це корисно для душі і для тіла, вона розвиває творчі здібності і допомагає зменшити стрес. Проте, не завжди є можливість грати на справжньому інструменті.

Отже, створення веб-додатку віртуального піаніно є актуальною задачею. Віртуальне піаніно відмінно підходить, як для розваг та ігор, так і для навчання на фортепіано.

Дана курсова робота присвячена розробленню веб-додатку віртуального піаніно, що дасть змогу будь-якому користувачу мережі Інтернет відчути себе піаністом і дасть змогу вчитись грі та створювати мелодійні композиції онлайн.

**1. АНАЛІЗ МОВ ПРОГРАМУВАННЯ ТА ТЕХНОЛОГІЙ**

**РОЗРОБЛЕННЯ WEB-САЙТІВ**

* 1. **Мова програмування JavaScript**

**JavaScript** — динамічна, об'єктно-орієнтована прототипна мова програмування. Є реалізацією стандарту мови ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

JavaScript це мова з великими перевагами, які роблять його найкращим вибором серед подібних йому, особливо в деяких варіантах застосування. Кілька переваг використання JavaScript:

* Вам не потрібен компілятор, тому що веб-браузер інтерпретує його в HTML;
* Його простіше вивчати, ніж інші мови програмування;
* Помилки простіше виявити, а значить і виправити;
* Він може прив'язуватися до спеціальних елементів сторінок або події на зразок натискання (click) або наведення миші (mouseover);
* JS працює в різних браузерах і на різних платформах;
* Ви можете використовувати JavaScript для валідації вхідних даних і зниження необхідності ручної перевірки даних;
* Він робить сайт більш інтерактивним і привабливим для відвідувачів;
* Він швидше і легше, ніж інші мови програмування.
  1. **Мова програмування Python**

**Python** — інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднування наявних компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду.

Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована.

Серед основних її переваг можна назвати такі:

* чистий синтаксис (для виділення блоків слід використовувати відступи);
* переносність програм (що властиве більшості інтерпретованих мов);
* стандартний дистрибутив має велику кількість корисних модулів (включно з модулем для розробки графічного інтерфейсу);
* можливість використання Python в діалоговому режимі (дуже корисне для експериментування та розв'язання простих задач);
* стандартний дистрибутив має просте, але разом із тим досить потужне середовище розробки, яке зветься IDLE і яке написане мовою Python;
* зручний для розв'язання математичних проблем (має засоби роботи з комплексними числами, може оперувати з цілими числами довільної величини, у діалоговому режимі може використовуватися як потужний калькулятор);
* відкритий код (можливість редагувати його іншими користувачами)

Python має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис Python, динамічна обробка типів, а також те, що це інтерпретована мова, роблять її ідеальною для написання скриптів та швидкої розробки прикладних програм у багатьох галузях на більшості платформ.

* 1. **Опис фреймворку ReactJS**

**React** — відкрита JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту веб-сторінки, з якими стикаються в розробці односторінкових застосунків.

React дозволяє розробникам створювати великі веб-застосунки, які використовують дані, котрі змінюються з часом, без перезавантаження сторінки.

Одна з ключових особливостей React - універсальність. Цю бібліотеку можна використовувати на сервері і на мобільних платформах за допомогою React Native. Це принцип Learn Once, Write Anywhere або «Навчіться один раз, пишіть де завгодно».

Ще одна важлива особливість бібліотеки - декларативність. За допомогою React розробник описує, як компоненти інтерфейсу виглядають в різних станах. Декларативний підхід скорочує код і робить його зрозумілим.

React заснований на компонентах, це ще одна ключова особливість бібліотеки. Кожен компонент повертає частину призначеного для користувача інтерфейсу зі своїм станом. Об'єднуючи компоненти, програміст створює завершений інтерфейс веб-додатки.

Важлива особливість React - використання JSX. Це розширення синтаксису JavaScript, яке зручно використовувати для опису інтерфейсу.

* 1. **Опис фреймворку Python Flask**

**Flask** — мікрофреймворк для веб-додатків, створений з використанням Python.

Flask називається мікрофреймворком, оскільки він не вимагає спеціальних засобів чи бібліотек. У ньому відсутній рівень абстракції для роботи з базою даних, перевірки форм або інші компоненти, які надають широковживані функції за допомогою сторонніх бібліотек. Однак, Flask має підтримку розширень, які забезпечують додаткові властивості таким чином, наче вони були доступні у Flask із самого початку. Існують розширення для встановлення об'єктно-реляційних зв'язків, перевірки форм, контролю процесу завантаження, підтримки різноманітних відкритих технологій аутентифікації та декількох поширених засобів для фреймворку. Розширення оновлюються частіше аніж базовий код.

Властивості:

* Містить сервер для розробки та відлагоджувач
* Вбудована підтримка юніт-тестів
* Управління запитами RESTful
* Використовує шаблони Jinja2
* Має підтримку безпечних куків (сесії на стороні клієнта)
* 100% відповідність WSGI 1.0
* Підтримка Unicode
* Докладна документація
* Сумісність з Google App Engine
* Наявність розширень для забезпечення бажаної поведінки

**2. РОЗРОБЛЕННЯ ВЕБ-ДОДАТКУ СТВОРЕННЯ МЕЛОДІЙ ЗА ДОПОМОГОЮ ВІРТУАЛЬНОГО ПІАНІНО**

## **2.1 Аналіз вимог до розробки веб-сервісу**

Аналіз вимог – це частина процесу розробки програмного забезпечення, що включає в себе збір вимог до програмного забезпечення, їх систематизацію, виявлення взаємозв'язків, а також документування. В процесі збору вимог важливо брати до уваги можливі протиріччя вимог різних зацікавлених осіб, таких як замовники, так і для розробників.

В якості розробника виступають студенти, в якості замовника – викладач.

Процес аналізу вимог до розробки веб-сервісу складається з трьох видів діяльності:

* збір вимог – спілкування з клієнтами і користувачами, щоб визначити, які їхні вимоги, а також аналіз предметної області;
* аналіз вимог – визначення, чи є зібрані вимоги неясними, неповними, неоднозначними або такими, що суперечать. А також рішення проблем та виявлення взаємозв'язку вимог;
* документування вимог – вимоги можуть бути задокументовані в різних формах, таких як простий опис, сценарії використання, користувацькі історії, або специфікації процесів.

Таким чином, були створені вимоги до розробки та функціональності веб-додатку створення мелодій за допомогою віртуального піаніно, сформовані та описані під час етапу документування вимог.

**2.2 Опис вимог до розробки веб-сервісу**

Зібраним вимогам було призначено ідентифікатор - код вимоги. Після етапу документування вимог, вони прийняли вид таблиці.

Таблиця 1. Опис вимог до проекту

|  |  |
| --- | --- |
| Код вимоги | Опис вимоги |
| А1 | Відображення піаніно з білими( та чорними клавішами |
| А2 | Підсвічування клавіш піаніно при натисканні клавіши |
| А3 | Можливість одночасного натискання декількох клавіш піаніно |
| А4 | Можливість натискання клавіши за допомогою курсору |
| А5 | Можливість натискання клавіши за допомогою клавіатури |
| А6 | Можливість запису програної композиції з урахуванням затримки між натисканнями клавіш |
| А7 | Можливість прослуховування програної мелодії |
| А8 | Можливість завантаження на пристрій зіграної мелодії |
| А9 | Аудіофайл зіграної мелодії повинен генеруватись без затримки для користувача |

## **2.3 Архітектура веб-додатку**

Даний веб-додаток складається з таких модулів:

* Модуля серверу
* Модуля клієнтського інтерфейсу
  + Модуля клавіатури піаніно
  + Модуля панелі керування (кнопок)

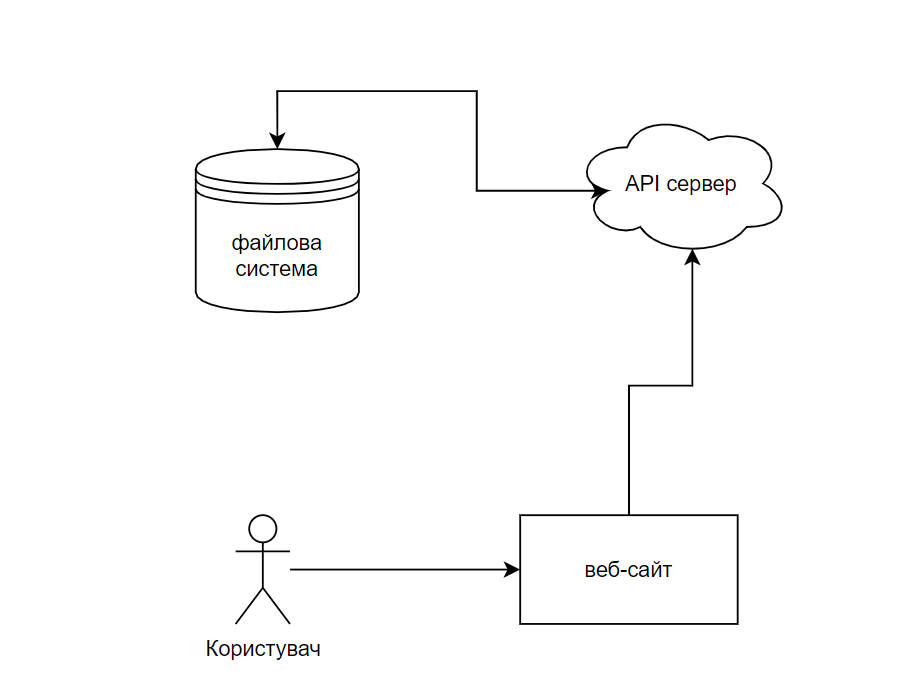


Рис. 1 Архітектура системи

Модуль серверу призначений для створення програми HTTP- серверу, який дозволяє клієнтській стороні взаємодіяти з файловою системою через його API. Модуль призначений для генерації аудіофайлу записаної мелодії користувача.

Модуль клієнтського інтерфейсу представляє собою односторінковий веб-сайт, який дозволяє користувачу через елементи керування на ньому створювати, прослуховувати та завантажувати зіграні мелодії. Цей модуль безпосередньо взаємодіє із модулем серверу через його API.

## **2.4 Вміст та дизайн сторінок веб-додатку**

Даний веб-додаток являє собою HTML сторінку, на якій відображено клавіши піаніно та панель керування зіграними мелодіями.

Віртуальне піаніно складається лише з однієї сторінки, вся логіка взаємодії з якою прописана за допомогою JS коду.

На панелі керування розташовані 3 кнопки керування зіграними мелодії. А саме: кнопка початку запису мелодії, кнопка програвання записаної мелодії, кнопка завантаження програної мелодїі.

Рис. 2 Сторінка браузерного віртуального піаніно

На даному скріншоті можна спостерігати головну сторінку інтерфейсу користувача. Тут можна спостерігати основні елементи описані на початку цього пункту, а саме піаніно та панель керування.

**3. ОПИС РОЗРОБЛЕНИХ АЛГОРИТМІВ ТА ПІДПРОГРАМ**

**3.1 Особливості реалізації модулю сервера**

Модуль серверу було створено на мові програмування Python із використанням фреймворку Python Flask.

Даний модуль підтримує взаємодію із клієнтською частиною через API побудоване за моделлю REST. Він дозволяє створювати аудіо файл на основі програних користувачем нот на віртуальному піаніно.

Для створення аудіо файлу потрібно надіслати HTTP запит на сервер із JSON-об’єктом, що містить список програних нот, який утворився при натисканні кнопки «Записати» на клієнтській частині системи. За допомогою списку програних нот сервер буде співставлювати назву ноти, що була зіграна в певний момент часу, з фізичним представленням ноти на пристрої, де знаходиться сервер. За допомогою бібліотеки **Jammer** утворюється аудіо файл, що надсилається у відповідь на HTTP запит. Аудіо файл зберігається на стороні серверу з назвою «H-M-S\_d-m-Y» у форматі .mp3, де H-години, M-хвилини, S- секунди, d-день, m-місяць, Y-рік, тобто час та дата створення мелодії.

**3.2 Особливості реалізації модулю клієнтського інтерфейсу**

Модуль клієнтського інтерфейсу було створено з використанням мови програмування JavaScript та фреймворку ReactJS у вигляді веб-додатку, який взаємодіє із API сервером. Користувач має можливість натискати на клавіші віртуального піаніно, тим самим вчитись компонувати звуки піаніно, щоб створити власну мелодію. Також між може записати свою створену мелодію та прослухати її, разом з цим він також може зберегти записані мелодії на пристрій для особистого користування використовуючи панель керування, що складається з 3 кнопок.

При натисканні кнопки «Записати» всі звуки від клавіш, що натискаються користувачем записуються в список, який містить назву клавіши, що була нажата, нота, яка відповідає натиснутій клавіші, та час, коли була натиснута відповідна клавіша.

При натисканні кнопки «Грати» список програних нот відтворюється за допомогою емуляції натискання клавіш користувачем. Тобто, відтворюється програна користувачем мелодія на піаніно з підсвіткою клавіш, для розуміння, яка клавіша була натиснута в певний момент.

При натисканні кнопки «Завантажити» надсилається запит на API серверу для створення аудіо файлу. Запит містить JSON-об’єкт з даними для створення файлу, а саме список програних нот, що утворився при натисканні кнопки «Записати». У відповідь користувач отримує посилання на файл, який можна завантажити на пристрій та прослухати.

# **АНАЛІЗ РОЗРОБЛЕНОГО WEB-РЕСУРСУ**

## **4.1 Процес тестування**

Тестування системи відбулося шляхом мануального тестування (ручного тестування).

**Ручне тестування** (Manual testing) — це процес ручної перевірки програмного забезпечення на помилки. Тестувальник має відігравати роль користувача програми й використовувати властивості програми для знаходження помилок у роботі програми. Для професійного тестування тестувальник часто користується написаним планом тестування з варіантами тестування (test cases).

Для тестування було написано та протестовані такі варіанти (test cases):

* Запис пустої мелодії ( не було зіграно ні однії ноти)
* Запис мелодії з одночасним натисканням мінімум 10 клавіш
* Одночасне натискання декількох клавіш

А також стандартні варіанти такі як:

* Натискання клавіш курсором
* Натискання клавіш клавіатурою
* Запис зіграної мелодії
* Програвання зіграної мелодії
* Завантаження зіграної мелодії

Усі варіанти тестування були успішно виконаними.

**4.2 Рекомендації щодо подальшого вдосконалення**

В процесі роботи над веб-додатком створення віртуального піаніно було виявлено ряд напрямів та функцій, які можуть покращити розроблене ПЗ, а саме:

* Створити редактор нот, за допомогою якою можна редагувати програну мелодію
* Створити базу даних відомих композицій, які будуть слугувати прикладом для вивчання нових композицій користувачем
* Додати різні види звучання клавіш піаніно, поповнивши базу даних нот
* Додати різні налаштування відтворення звуку клавіш піаніно, наприклад, гучність
* Створити підказки для користувача, наприклад, відображення ноти на клавіші

**ВИСНОВКИ**

Метою даного курсового проекту було розроблення веб-додатку для створення мелодій за допомогою віртуального піаніно.

Розроблений веб-додаток дозволяє:

* Програвання нот за допомогою курсору
* Програвання нот за допомогою клавіатури
* Запис програної мелодії
* Прослуховування програної мелодії
* Завантаження програної мелодії на пристрій

Розробка виконана у повному обсязі, всі вимоги, викладені у технічному завданні враховані, тестування продукту виконано у відповідності до затвердженої програми та методики тестування.

Використання цього веб-додатку дозволить користувачам створювати власноруч створені композиції за допомогою віртуального піаніно.

**СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ**

1. Web-програмування [Електронний ресурс]. — Режим доступу : <http://webstudio2u.net/ua/programming/96-cms.html>
2. Ручне тестування [Електронний ресурс]. — Режим доступу :

<https://uk.wikipedia.org/wiki/Ручне_тестування>

1. ReactJS: керівництво для початківців [Електронний ресурс]. — Режим доступу :

<https://habr.com/ru/company/ruvds/blog/428077/>

1. Створення веб-додатку за допомогою Flask в Python3 [Електронний ресурс]. — Режим доступу :

<https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3-ru>

1. Аналіз вимог [Електронний ресурс]. — Режим доступу: <https://habr.com/post/340956/>