

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка веб-сайту музичної групи «The Hardkiss»

Виконала: студентка IV курсу, групи СНс-42

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Панюс Н. О.

(прізвище та ініціали)

Керівник

(підпис)

Назаревич О. Б.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Шимчук Г.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Тиш Є. В.

(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)

«__» _____ 2021 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студентці Панюс Нелі Олегівній
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка веб-сайту музичної групи «The Hardkiss»

Керівник роботи кандидат технічних наук, доцент кафедри КН Назаревич Олег Богданович
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «П» березня 2021 року № 4/7-171

2. Термін подання студентом завершеної роботи 24 червня 2021р.

3. Вихідні дані до роботи Літературні та Інтернет джерела

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. Розділ 1 постановка завдання розроблення веб-сайту музичної групи The Hardkiss.

1.1 Найменування та область застосування. 1.2 Призначення розробки веб-сайту музичної групи «The Hardkiss». 1.3 Вимоги до розробки веб-сайту музичної групи «The Hardkiss».

1.4 Вибір засобів реалізації веб-сайту групи «The Hardkiss». 1.5 Висновок до першого Розділу. 2 Проектування веб-сайту музичної групи The Hardkiss. 2.1 Розробка архітектури веб-сайту музичної групи «The Hardkiss». 2.2 Реалізація архітектури веб-сайту музичної групи The Hardkiss. 2.3 Створення та обробка форм засобами HTML та Python 3.7. 2.4 Робота з базою даних веб-сайту музичної групи «The Hardkiss». 2.5 Валідація та тестування веб-сайту музичної групи «The Hardkiss». 2.6 Перенесення веб-сайту музичної групи «The Hardkiss» на хостинг. 2.7 Висновок до другого розділу. 3 Безпека життєдіяльності, основи охорони праці. 3.1 Контроль за станом охорони праці. 3.2 Загальні вимоги безпеки з охорони праці для користувачів ПК. 3.3 Критичні стани людини. Висновки. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці	Гурик О.Я., доцент кафедри МТ		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи		Виконано
2.	Підбір джерел про розробку та створення веб-сайтів		Виконано
3.	Опрацювання обраних джерел		Виконано
4.	Здійснення аналізу програмних засобів для реалізації веб-сайту музичної групи The Hardkiss		Виконано
5.	Розроблення веб-сайту музичної групи The Hardkiss		Виконано
6.	Оформлення першого розділу «Постановка завдання розроблення веб-сайту музичної групи The Hardkiss»		Виконано
	»		
7.	Оформлення другого розділу «Проектування веб-сайту музичної групи The Hardkiss»		Виконано
8.	Виконання завдання до підрозділу «Безпека життєдіяльності»		Виконано
9.	Виконання завдання до підрозділу «Основи хорони праці»		Виконано
10.	Оформлення кваліфікаційної роботи		Виконано
11.	Нормоконтроль		Виконано
12.	Перевірка на плагіат		Виконано
13.	Попередній захист кваліфікаційної роботи		Виконано
14.	Захист кваліфікаційної роботи		

Студент

(підпис)

Панюс Н. О.

(прізвище та ініціали)

Керівник роботи

(підпис)

Назаревич О. Б.

(прізвище та ініціали)

АНОТАЦІЯ

Розробка веб-сайту музичної групи «The Hardkiss» // Кваліфікаційна робота освітнього рівня «Бакалавр» // Панюс Неля Олегівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНс-42 // Тернопіль, 2021 // С. 47 , рис. – 32, табл. – 2, кресл. – 0, додат. – 29, бібліогр. – 30.

Ключові слова: веб-сайт, html, css, js, mysql, python, flask, хостинг, веб-тестування.

Метою кваліфікаційна робота є розробка веб-сайту музичної групи The Hardkiss. Під час роботи були використані наступні технології - HTML5, CSS 3 для проектування зовнішнього вигляду та розмітки сайту, JS для динамічної зміни вигляду сайту на стороні клієнта, Python 3.7 для створення сервера і MySQL 5.6 для роботи з базою даних.

Для реалізації серверної частини був використаний Python-фреймворк Flask, який займається обробкою запитів, взаємодією сайту з базою даних та обробкою HTML шаблонів. Використовуючи його, також був створений функціонал розсилки анонсів на електронні пошти занесених у базу користувачів. Для цього використовувалась бібліотека Flask-Mail як інтерфейс для веб-сайту на основі протоколу SMTP.

Також на стороні сервера була реалізована його взаємодія з базою даних MySQL, використовуючи бібліотеку Flask-SQLAlchemy, яка створює інтерфейс для веб-сайту, в основі якого працює технологія об'єктно-реляційного відображення бази даних.

ANNOTATION

Music group «The Hardkiss» website development // Qualification work of Bachelor educational degree // Panius Nelia // Ternopil Ivan Puluj Natinal Technical University, Faculty of Computer Information Systems and Software Engineering, Computer Science Department, SNs-42 group // Ternopil, 2021 // Pages – 47, figures – 32 , tables – 2 , sketches – 0, addendums – 29, references – 30.

Keywords: web-site, html, css, js, mysql, python, flask, web testing, hosting.

The purpose of the qualification work is to develop the music group's website "The Hardkiss". The following technologies were used during the work - HTML5, CSS 3 to design the appearance and layout of the site, JS to dynamically change the look of the site on the client-side, Python 3.7 to create a server, and MySQL 5.6 to work with the database.

The Python framework Flask was used to implement the server part, which deals with request processing, website interaction with the database, and HTML-templates processing. The functionality of announcements sending to e-mails entered in the users' database was also created by using it. For this purpose, the Flask-Mail library was used as an interface for a website based on the SMPT protocol.

On the server side, its interaction with the MySQL database was implemented using the Flask-SQLAlchemy library that creates an interface for the website based on the database object-relational mapping.

СПИСОК СКОРОЧЕНЬ

HTML – Hyper text markup language (мова розмітки гіпертекстових документів).

XHTML – Extensible hyper text markup language.

XML – Extensible markup language.

CSS – Cascading style sheets (каскадні таблиці стилів).

JS – JavaScript (мова програмування).

AJAX – Asynchronous JavaScript and XML.

WWW – World Wide Web (найбільше всесвітнє багатомовне сховище інформації в електронному вигляді).

COM – Component object model.

СКБД – Система керування базами даних.

SQL – Structured Query language.

ТЗ – Технічне завдання.

OS – Operating system.

MVC – Model-View-Controller (модель–вигляд–контролер).

SMTP – Simple Mail Transfer Protocol.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 ПОСТАНОВКА ЗАВДАННЯ РОЗРОБЛЕННЯ ВЕБ-САЙТУ МУЗИЧНОЇ ГРУПИ THE HARDKISS.....	8
1.1 Найменування та область застосування	8
1.2 Призначення розробки веб-сайту музичної групи «The Hardkiss»....	8
1.3 Вимоги до розробки веб-сайту музичної групи «The Hardkiss»	9
1.4 Вибір засобів реалізації веб-сайту групи «The Hardkiss».....	10
1.5 Висновок до першого розділу	14
РОЗДІЛ 2 ПРОЕКТУВАННЯ ВЕБ-САЙТУ МУЗИЧНОЇ ГРУПИ THE HARDKISS	15
2.1 Розробка архітектури веб-сайту музичної групи «The Hardkiss»	15
2.2 Реалізація архітектури веб-сайту музичної групи The Hardkiss	16
2.3 Створення та обробка форм засобами HTML та Python 3.7.....	23
2.4 Робота з базою даних веб-сайту музичної групи «The Hardkiss»	28
2.5 Валідація та тестування веб-сайту музичної групи «The Hardkiss»	29
2.6 Перенесення веб-сайту музичної групи «The Hardkiss» на хостинг	32
2.7 Висновок до другого розділу	37
РОЗДІЛ 3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ .	38
3.1 Контроль за станом охорони праці.....	38
3.2 Загальгі вимоги безпеки з охорони праці для користувачів ПК.....	39
3.3 Критичні стани людини.....	41
ВИСНОВКИ	44
ПЕРЕЛІК ДЖЕРЕЛ	45
ДОДАТКИ	

ВСТУП

Актуальність дослідження. Розвиток сучасних інформаційних систем та їх активне застосування в повсякденному житті не можливе без Інтернету, що є глобальною мережею та основою мережі WWW. Стрімкий розвиток розробки сайтів та їх застосування стало початком зародження нових напрямків, а саме веб-дизайн і веб-технологій.

Ще донедавна, сайти виступали новим явищем в світі розробки та розвитку інформаційних систем, але зараз вони стають все звичними атрибутами та невід’ємними елементами сьогодення. Оскільки, сайти стають все зручнішими та практичнішими, попит на їх розробку та використання лише зростає. Розробка простого та зручного вигляду сайту є надважливим завданням, що постає перед розробником, тому що користувач буде використовувати лише зручний та зрозумілий інтерфейс, а на складний і зовсім не звертатиме уваги.

Мета і задачі дослідження. Мета кваліфікаційної роботи є розробка веб-сайту для фанів українського музичного гурту The Hardkiss.

Щоб реалізувати поставлену мету було поставлено наступні завдання:

- здійснити аналіз області застосування;
- здійснити розробку архітектури веб-сайту;
- здійснити розроблення зовнішнього вигляду веб-сайту;
- провести валідацію та тестування веб-сайту з подальшим перенесенням веб-сайту на хостинг.

Практичне значення одержаних результатів. Результатом виконання всіх завдань є розроблений веб-сайт, який включає в себе основні функції та реалізовує основну ідею розробки – єдиний ресурс, який об’єднує всіх фанів музичного гурту «The Hardkiss».

РОЗДІЛ 1 ПОСТАНОВКА ЗАВДАННЯ РОЗРОБЛЕННЯ ВЕБ-САЙТУ МУЗИЧНОЇ ГРУПИ THE HARDKISS

1.1 Найменування та область застосування

Завданням даної кваліфікаційної роботи освітнього рівня «Бакалавр» є розробка веб-сайту для фанів української рок-групи The Hardkiss.

«The Hardkiss» — є музичним рок-гуртом України, що зародився ще в 2011 році. Кожну пісню даного гурту створила виключно солістка гурту — Саніна Юлії та її чоловік Бебко Валерій, що також виступає не тільки гітаристом, а й креативним продюсером даної групи.

Уже понад 10 років рок-гурту створює дійсно драйвову та живу музику, яка не може не привернути уваги. Це пісні, що стають необхідними кожної миті, адже їх наповнення заставляє відчувати радість, сум чи щастя — всю палітру емоцій. Гурт «The Hardkiss» об'єднує усіх своїх друзів з такими різними життєвими шляхами.

1.2 Призначення розробки веб-сайту музичної групи «The Hardkiss»

Мета даної дипломної роботи — розробка веб-сайту українського музичного гурту The Hardkiss з використанням Python 3.7, HTML5, MySQL CSS3, JS та розміщення його в мережі Інтернет та подальшого використання як інформаційного сайту.

Розроблюваний веб-сайт міститиме інформацію про гурт, його музику, відео роботи та фото галерею, також з можливістю заповнення форми для розсилки анонсів на вказану пошту, занесених у базу користувачів.

1.3 Вимоги до розробки веб-сайту музичної групи «The Hardkiss»

Розробка сайту починається з визначення суті та основної ідеї веб-сайту. Саме від цього залежатиме структура і його наповнення, а також способи просування як в мережі Інтернет так і в інших соціальних мережах, що набувають неабиякої популярності.

Основні вимоги до створення та розробки сайту є наступними:

- сайт має коректно відображатися і функціонувати в найбільш поширених браузерах таких як Firefox та Chrome;
- сайт має коректно відображатися для користувачів, в яких роздільна здатність екрана сягає 800x600 і вище;
- навігація по сайту має бути представлена у вигляді меню, яке повинно забезпечувати легкий та зрозумілий перехід до усіх сторінок сайту;
- усе текстове наповнення сайту не повинно містити як граматичних, так і стилістичних помилок;
- дизайн сайту повинен бути виконаний з дотриманням основних правил Теорії кольору та Правилам композиції [6].

Структура сайту має відповідати усім сучасним вимогам та стандартам. Необхідно також врахувати етапи користування розробленим сайтом, спроектувати процес навігації, забезпечити зручний перехід з розділу в підрозділ, швидкий спосіб пошуку необхідної інформації.

Щодо меню сайту, то воно повинно бути чітким, зрозумілим, доступним та найголовніше помітним для користувача. Якщо зробити заплутану структуру, не відсортувати наявну інформацію, приховати переходи, тоді даний сайт не придатний користування. Таким сайтом буде складно користуватися та відповідно шукати потрібну інформацію [10] [11].

Виокремлюють три моделі структури сайті, які активно використовуються, а саме:

- лінійна, що забезпечує покрокових прямий перехід з головної сторінки на відповідну під сторінку. Дана структура унеможливорює потрапляння на іншу сторінку попередньо не відкривши головної;

- деревоподібна – це структура, що відкриває сторінку, що обрана користувачем без прив’язки до головної;

- довільна – це структура, що не має обмежень, де розробник сам встановлює алгоритм переходу по сторінках, базуючись на висланих ідеях.

Розробник веб-сайту самостійно вирішує, який метод організації структури вибрати для вирішення встановлених цілей. Для реалізації веб-сайту музичного гурту The Hardkiss було обрано деревоподібну структуру організації переходів по сторінках веб-сайту [8].

1.4 Вибір засобів реалізації веб-сайту групи The Hardkiss

Вибір засобів реалізації є одним з важливих етапів створення веб-сайту. Зараз існує дуже багато технологій, мов програмування та способів розробки, проте слід почати з основи, а саме HTML (зміст і розмітка), Java Script (логіка) і CSS (зовнішній вигляд, позиціонування) [17].

Веб-сайт - це набір веб-сторінок, де вони, зазвичай, пов’язані між собою. Щоб створити веб-сайт, потрібно створити одну або більше веб-сторінок, які в свою чергу представлені у вигляді текстових файлів, який містить комбінацію тексту та коду «розмітки». Коли документ переглядається у веб-браузері, він відображається за призначенням, коли він переглядається в текстовому редакторі він відображає текст і код «розмітки».

Теги HTML представляють вихідний код веб-сторінки. Деякі теги є обов’язковими (для всі веб-сторінок повинні їх застосовувати), але більшість з них є необов’язковими. Фактичні теги, залежатимуть від вмісту, який потрібно зобразити [29].

Наведений нижче рисунок 1.1 є шаблоном для будь-якої веб-сторінки.

```

<!doctype HTML public "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Мій перший HTML-документ</title>
  </head>
  <body>
    Hello world!
  </body>
</html>

```

Рисунок 1.1 –HTML - документ

Елементи HTML вказують браузеру, як відображати вміст, позначають фрагменти вмісту, такі як кнопки, пункти списку, заголовок, абзац тексту чи посилання.

Тому для розроблення веб-сайту групи The Hardkiss прийнято рішення використовувати мову гіпертекстової розмітки тексту HTML за основу розробки сторінок даного сайту [30].

Звичайний HTML-код, що визначає структуру змісту зазвичай виступає неповноцінним контентом, завантаживши якого на сервер було б величезною помилкою. Вирішення проблеми - CSS, це технологія, яка забезпечує можливість стилізації. CSS - це неймовірно потужна технологія. Вона може взяти таку ж структуру HTML і представити її абсолютно іншим способом,

CSS описує спосіб відображення елементів HTML на екрані, папері або в інших носіях, що може зекономити досить багато роботи. Зазвичай, зовнішні таблиці стилів зберігаються у файлах з розширенням «.css». Відносно браузера, це список інструкцій, що вказують як і де потрібно відображати елементи на веб-сайті, що написано специфічним способом.

CSS працює в парі з HTML, що виконують спільну роботу, проте кожен має своє окремі функції. HTML в даному випадку відповідає за структуру, вміст та наповнення, тоді як CSS відповідає за візуальне представлення, тобто графічне оформлення [13].

Оскільки стилі CSS активно використовуються для оформлення HTML-тегів, тому було прийнято рішення, що для розроблення веб-сайту групи The Hardkiss застосовувати саме стилі CSS.

Використання мов програмування є зовсім іншим рівнем розробки, оскільки вони дають змогу комплексно виконувати роботу.

JavaScript (часто просто JS) - це досить легка, об'єктно-орієнтована мова, яка зазвичай використовується як сценарії для веб-сайту, проте також використовується і в інших програмних продуктах, таких як NODE.js або Apache CouchDB. JavaScript є мовою, що підтримує об'єктно-орієнтоване та функціональне програмування [22].

JavaScript є важливою частиною сучасної веб-розробки, надаючи розробникам перевірені і протестовані інструменти для створення масштабованих та інтерактивних веб-сайтів. Багато сучасних компаній використовують фреймворки JavaScript для своїх рішень, тому багато завдань пов'язані з розробкою клієнтської частини веб-сайтів тепер потребують навички роботи з ними. Сценарії JavaScript має бути підключений у HTML документ або створювати посилання на такий код, що буде розпізнаватися браузером [23].

Що відноситься до інших можливостей даної мови – це те, що вони залежать від середовища, в якому виконується JavaScript. У браузері JS здатен робити все, що стосується до маніпуляції із сторінкою, взаємодії з користувачем та з сервером, а саме:

- створювати та стирати HTML-теги, вносити зміни в стилі певних елементів або приховувати їх;
- реагувати на дії користувача, виступати обробником запитів, а саме опрацьовувати натискання на кнопки миші, переміщення курсора миші та нажимання по клавіатурі;
- посилати запити на сервер і загрузати дані без перезавантаження сторінки, робити запит на дані та виводити відповідні повідомлення [19] [24].

Користувач заходить на сайт і бачить певну інформацію, і тутникає питання де ж зберігаються ці дані. Щоб зберігати набір даних організованим способом потрібно використовувати, наприклад, MySQL.

MySQL позиціонується як реляційна модель, що зберігає дані у вигляді стовпців та рядків, що виступають атрибутами та записами. Це дає змогу користувачам швидко і зручно отримувати інформацію, оскільки вона вже організована у вигляді таблиці в базі даних. Для реалізації інтерфейсу взаємодії користувача та веб-сайту (користувач-додаток) використовується ПЗ – СКБД.

Модель бази даних MySQL розроблена на основі таким мов програмування як C та C ++ . MySQL може працювати на різних системних платформах. таких як: FreeBSD, BSDi, AIX , HP-UX, eComStation, , IRIX, NetBSD, NovellNetWare, OracleSolaris, Symbian, SunOS, та всі найвідоміші операційні системи. [12]

Було прийняте рішення застосовувати базу даних типу «реляційна модель» для зберігання і організації інформації, оскільки її структура базується на таблицях (зручність використання) та працює на багатьох платформах, що розширює її спектр можливостей для реалізації веб-сайту.

Нові мови програмування – це зазвичай нові можливості, які раніше не були часто застосовані. Одна з таких мов – Python. Це мова програмування, яка за останні кілька років стала посідати лідерські позиції серед найпопулярніших мов [18].

Його гнучкість дозволяє робити багато цікавих за стосунків. Використовуючи мову програмування Python можна писати основні програми та сценарії, а також створювати складні та масштабні корпоративні рішення. Створення веб та Інтернет-додатків є одною із можливостей даної мови. Наприклад, Python є лежить за основу в деяких відомих веб-сайтів, таких як: Reddit, Dropbox та YouTube. Веб-фреймворк Python Django працює як на Instagram, так і на Pinterest [28].

Flask – це фреймворк, що застосовується для розробки веб-ресурсів, базуючись на мові програмування Python, використовуючи інструменти Werkzeug, і також шаблони Jinja2. Цей фреймворк відносять до так званих «простих», оскільки, він дає ли базові можливості для розробки. Це і є перевагою, тому що це простий і швидкий інструмент для реалізації веб-сайту.[21].

SQLAlchemy - це сукупність інструментів SQL для Python та табличний переглядач, який надає розробникам повний функціонал SQL [27].

Загальну структуру можна проілюструвати так, як зображено на рисунку 1.2.

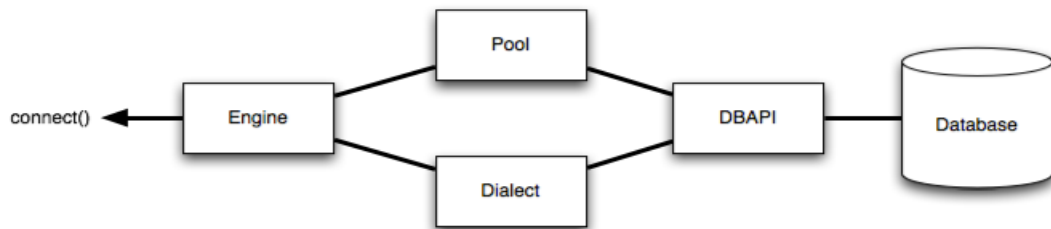


Рисунок 1.2 – Структура SQLAlchemy

Отже, Python є багато функціональна і широко застосовується для вирішення задач різних напрямків, тому для розробки серверної сторони веб-сайту групи The Hardkiss із взаємодією з базою даних прийнято рішення використовувати Python 3.7 з використанням фреймворком Flask та SQLAlchemy.

1.5 Висновок до першого розділу

У даному розділі здійснено аналіз області застосування та визначено призначення розробки веб-сайту музичної групи «The Hardkiss». Встановлено вимоги до розробки та обрано засоби реалізації.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБ-САЙТУ МУЗИЧНОЇ ГРУПИ THE HARDKISS

2.1 Розробка архітектури веб-сайту музичної групи «The Hardkiss»

Архітектура сайту – це структурно організована інформація та навігація по веб-сайту, мета якої забезпечити зручність взаємодії користувача із веб-сайтом, для пошуку необхідної інформації. Коректно розроблений UI та UX забезпечить користувачу зручність роботи з веб-сайтом та скоротило час пошуку необхідного.

Архітектура включає в себе всі рівні сайту – всі сторінки, форми та розділи. Обов’язкова наявність навігаційного меню та кнопки швидкого повернення – для зручнішого користування [1]. На рисунку 2.1 зображено структуру веб-сайту.

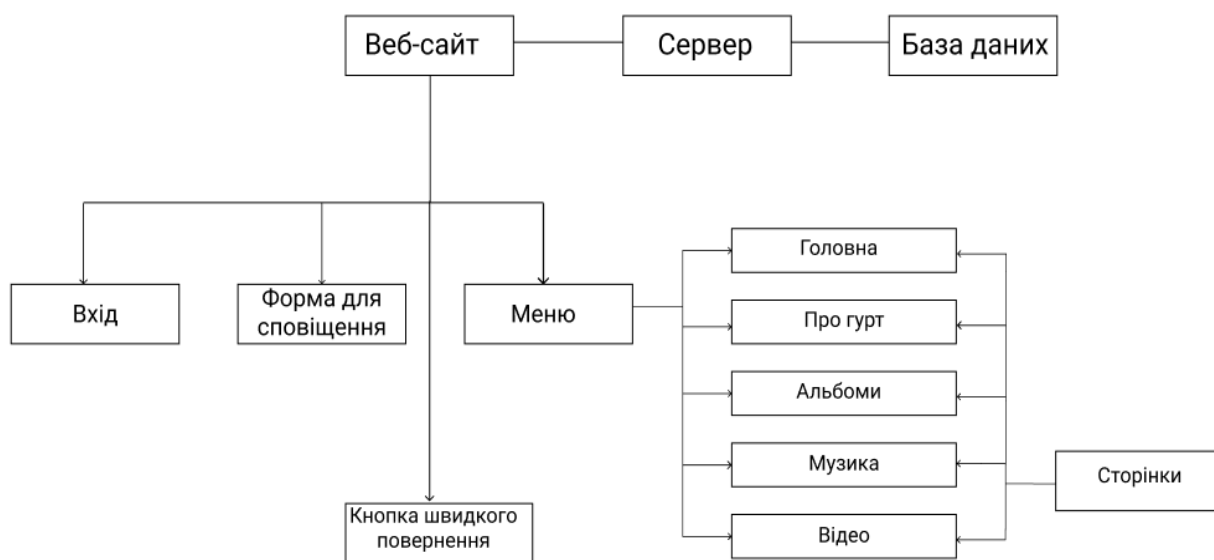


Рисунок 2.1 – Структура веб-сайту

Структура сайту, що розробляється в ході даної дипломної роботи повинна передбачати 5 рівнів (сторінок):

- «Головна»;
- «Про гурт»;
- «Альбоми»;

- «Музика»;
- «Відео».

2.2 Реалізація архітектури веб-сайту музичної групи The Hardkiss

Реалізація архітектури починається з розробки загального дизайну веб-сайту, обираючи основних кольорів та фону основної сторінки.

Початковий дизайн зображено на рисунку 2.2, що реалізовано в середовищі Figma.com [20].

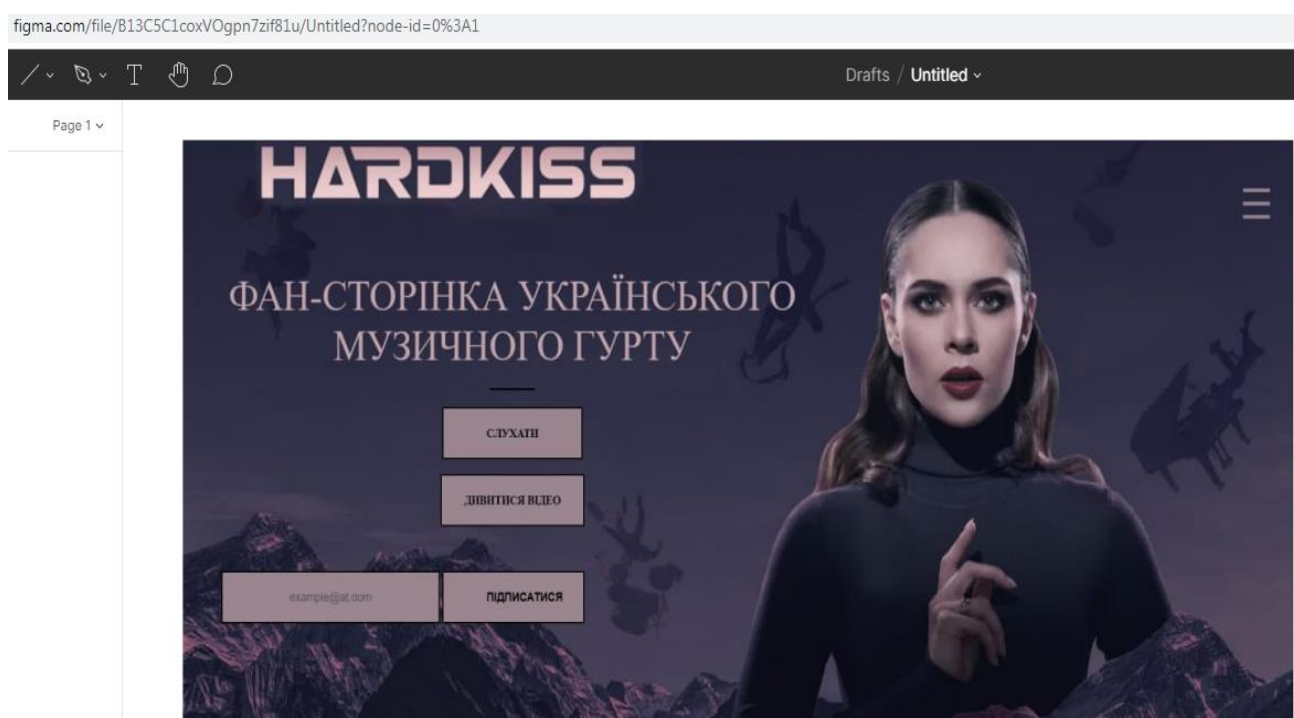


Рисунок 2.2 – Початковий дизайн головної сторінки

Головна сторінка дещо змінилася – складається з трьох рівнів:

1. Головна – основна частина, що на рисунку 2.3, що містить кнопки для швидкого переходу на Музику та Відео. Також містить форму, заповнення якої забезпечує розсилку анонсів на вказану пошту.

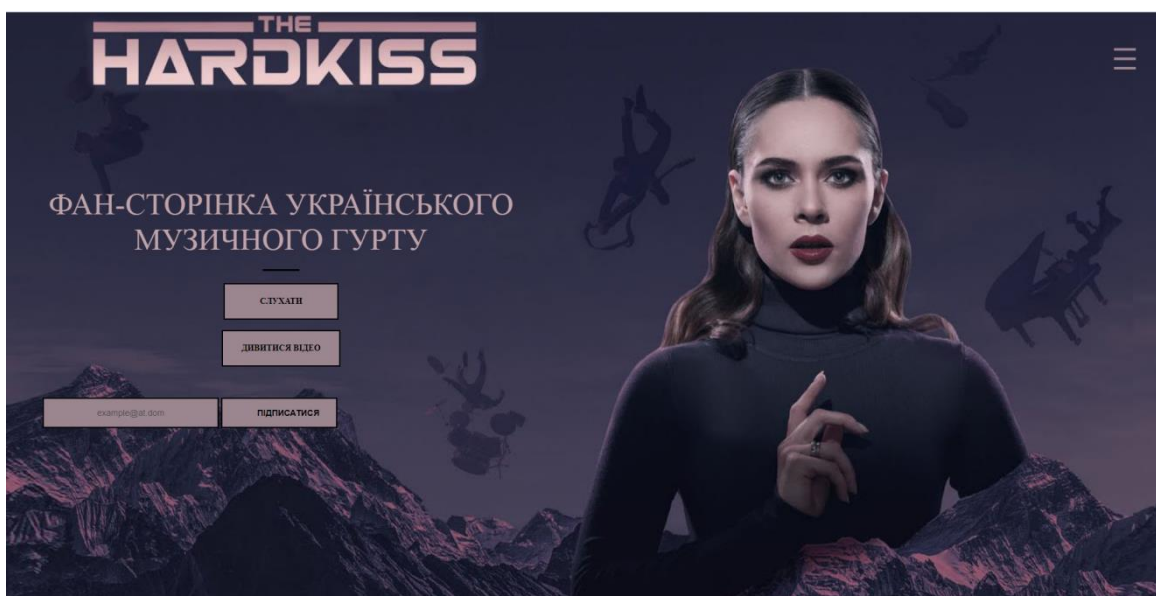


Рисунок 2.3 – Основна частина головної сторінки сайту

2. Середня частина – розділ, що на рисунку 2,4, який містить головну цитату групи. Фоном сторінки виступає фото групи.

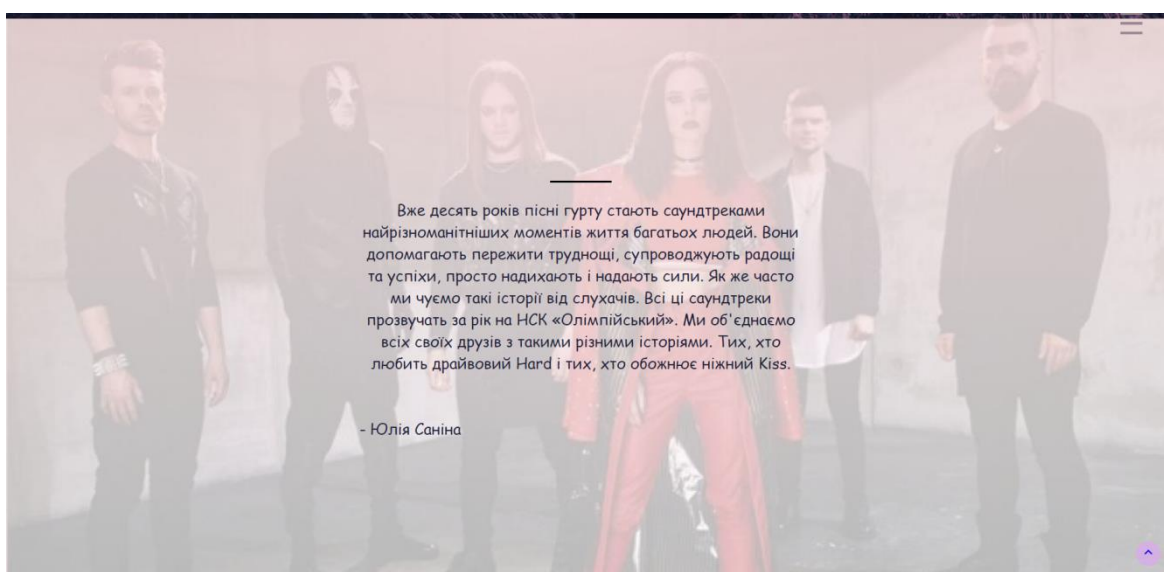


Рисунок 2.4 – Середня частина головної сторінки

3. Кінець головної сторінки – частина, що реалізована у вигляді відео-фону зображена на рисунку 2.5. При переході до даної частини сайту, вмикається відео, що слугує візитівкою Головної сторінки.



Рисунок 2.5 – Відео-фону Головної сторінки

Для даного сайту розроблено навігаційне меню, що впливає при натиску на кнопку , що зображено на рисунку 2.6. Кнопка є статична, тому меню можна викликати будь-коли і при цьому не обов'язково повертатися на початок сторінки.

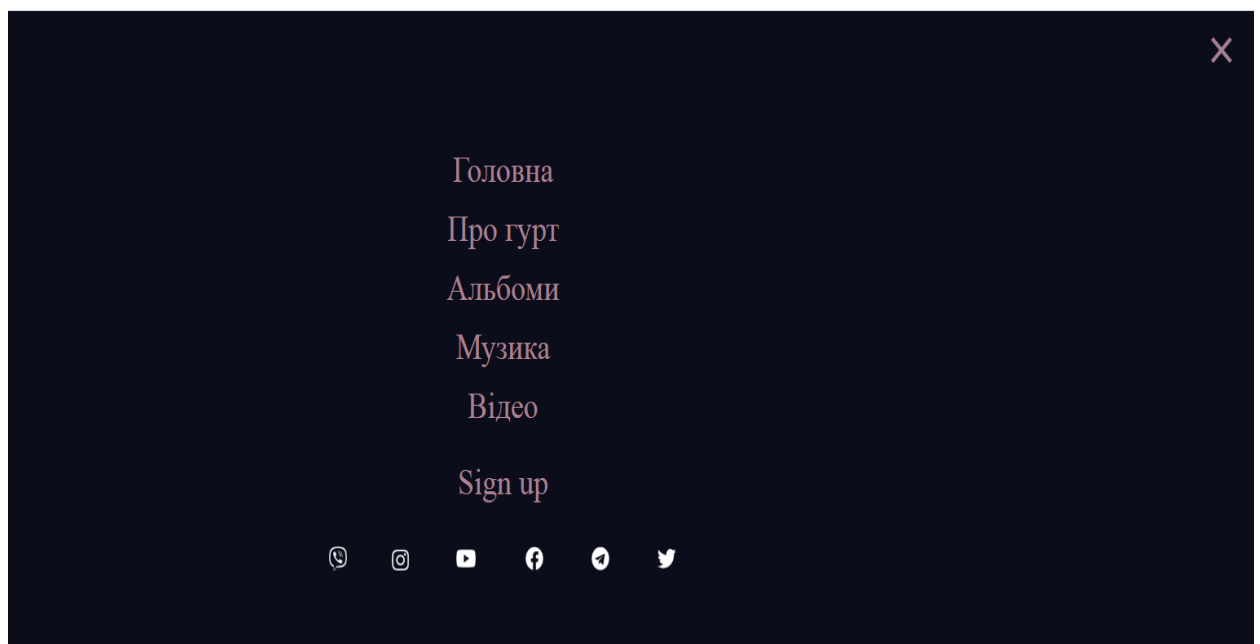


Рисунок 2.6 – Навігаційне меню

Також реалізовано кнопка швидкого повернення на вверх, що є на рисунку 2.7, оскільки Головна сторінка має властивість прокрутки.

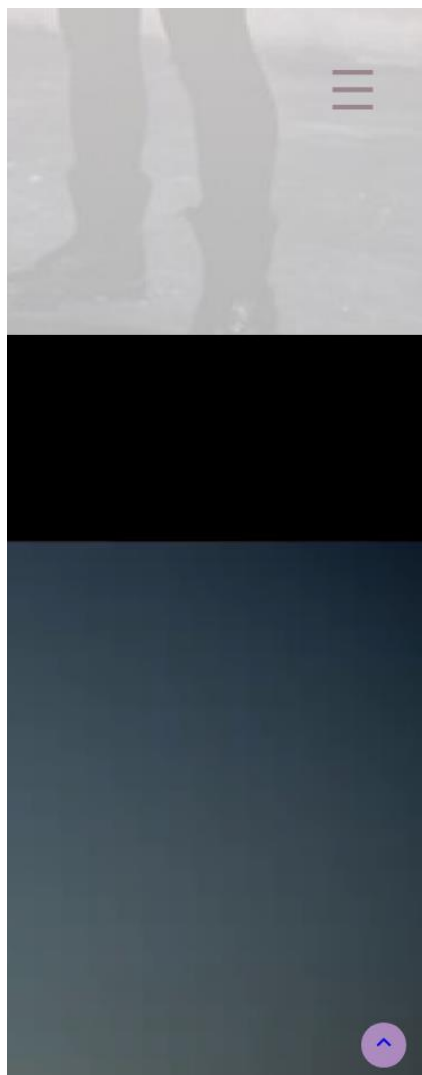


Рисунок 2.7 – Кнопка швидкого повернення

Наступна сторінка сайту – «Про гурт». Вона також реалізована частинами, а саме:

- основна інформація про групу та склад її учасників, зображено на рисунку 2.8);
- реалізована фотогалерея, що зображена на рисунку 2.9.

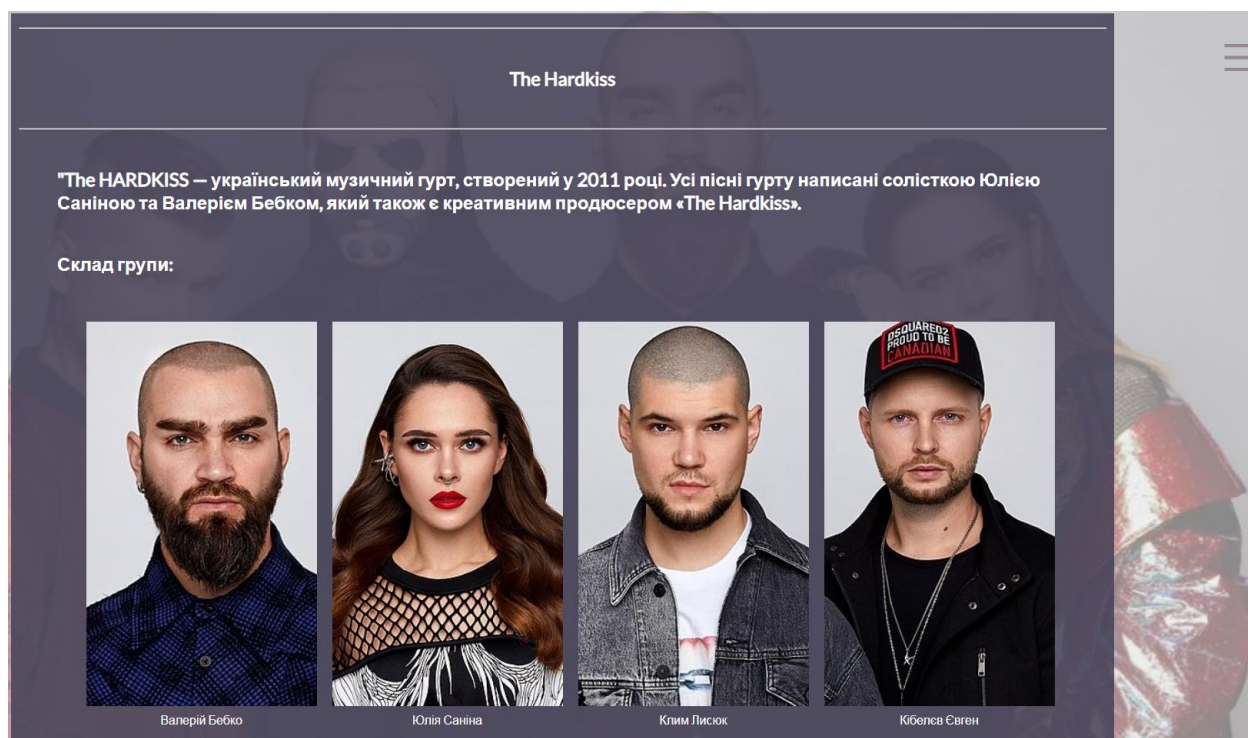


Рисунок 2.8 – Основна інформація про групу та склад її учасників

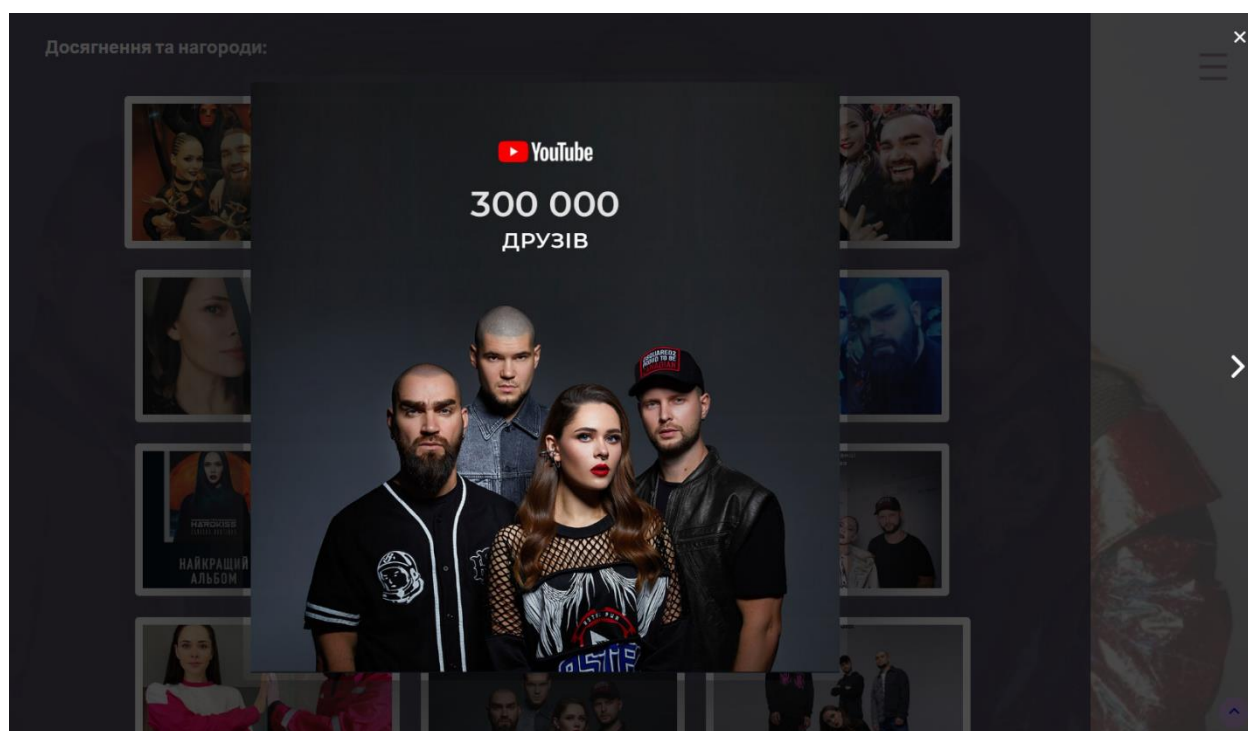


Рисунок 2.9 – Фотогалерея

Наступна сторінка – «Альбоми», що на рисунку 2.10. Тут зображено посилання на всі випущенні альбоми групи від моменту заснування.

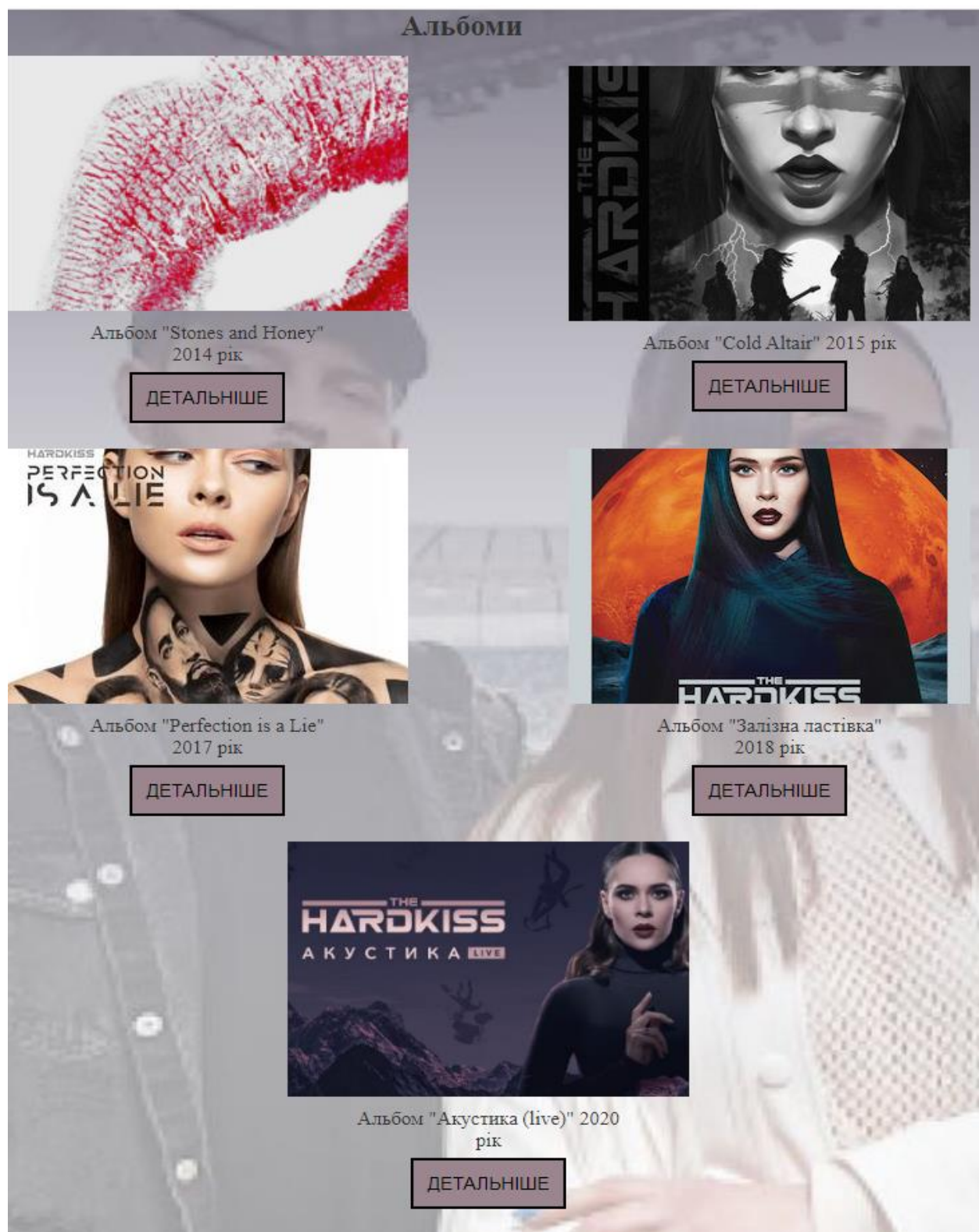


Рисунок 2.10 – Вигляд сторінка «Альбоми»

Наступна сторінка – «Музика», що на рисунку 2.11. На цій сторінці зображено всі можливі платформи на яких представлені пісні даної групи. При натиску у новій вкладці відкривається обрана платформа. Це зручно, оскільки

відкриття нової вкладки дає можливість залишитися на самому сайті без виконання додаткових маніпуляцій. Фоном даної сторінки є фото групи, з використанням ефекту «білої сітки», для кращого відображення основного контенту даної сторінки.

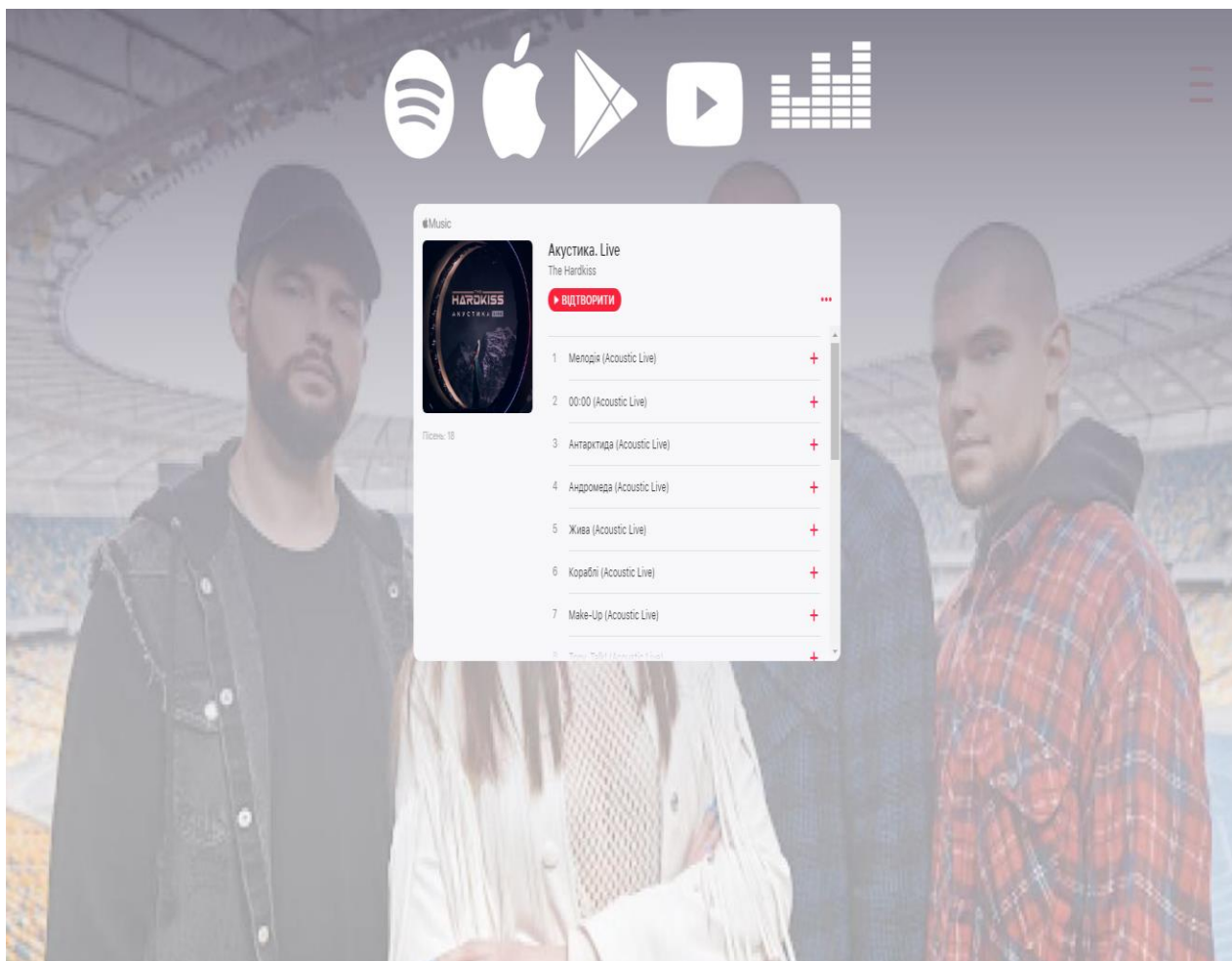


Рисунок 2.11 – Вигляд сторінки «Музика»

Сторінка «Відео», що на рисунку 2.12 - містить вбудований відео фон, що виступає тизером нової відео-роботи гурту. На ньому зображені ще відео роботи, але щоб переглянути більше є кнопка, яка направляє на їх Youtube канал.

Даний відео-фон додає сайту оригінальності та загадковості, тому що сам гурт позиціонується як щось неординарне, нове та загадове.

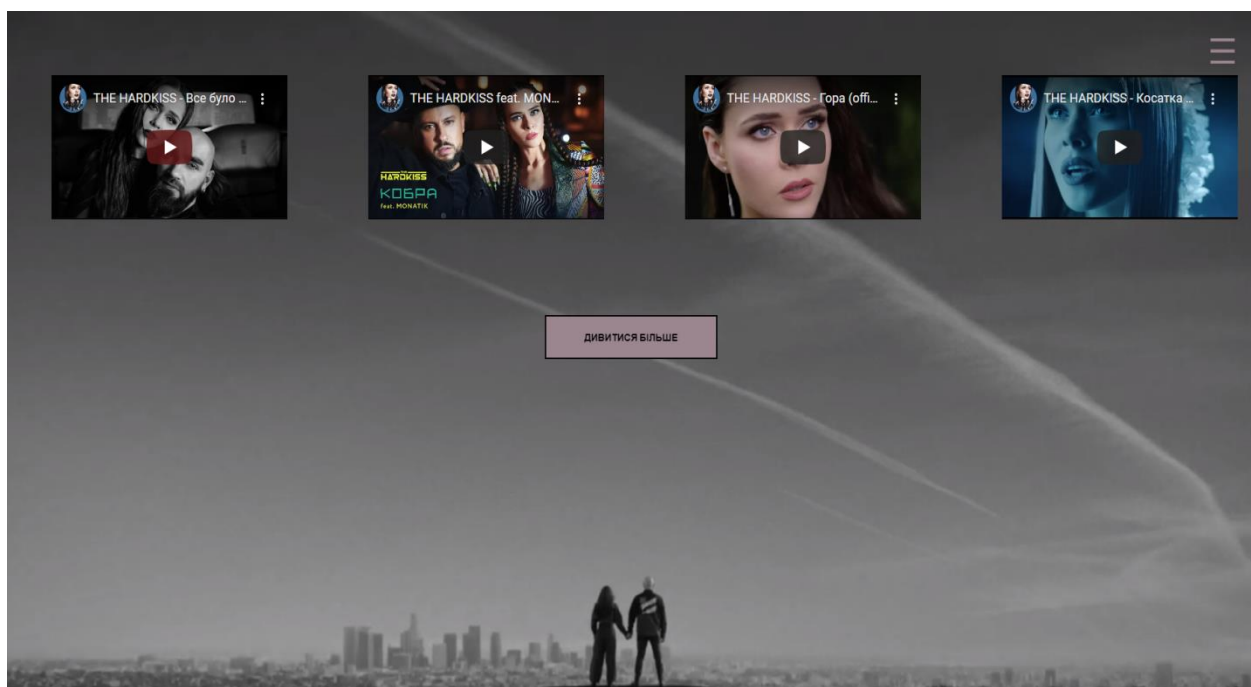


Рисунок 2.12 – Вигляд сторінки «Відео»

Фон спрацьовує автоматично при відкритті сторінки або після перезагрузки, тобто не потрібно виконувати додаткових маніпуляцій для його запуску.

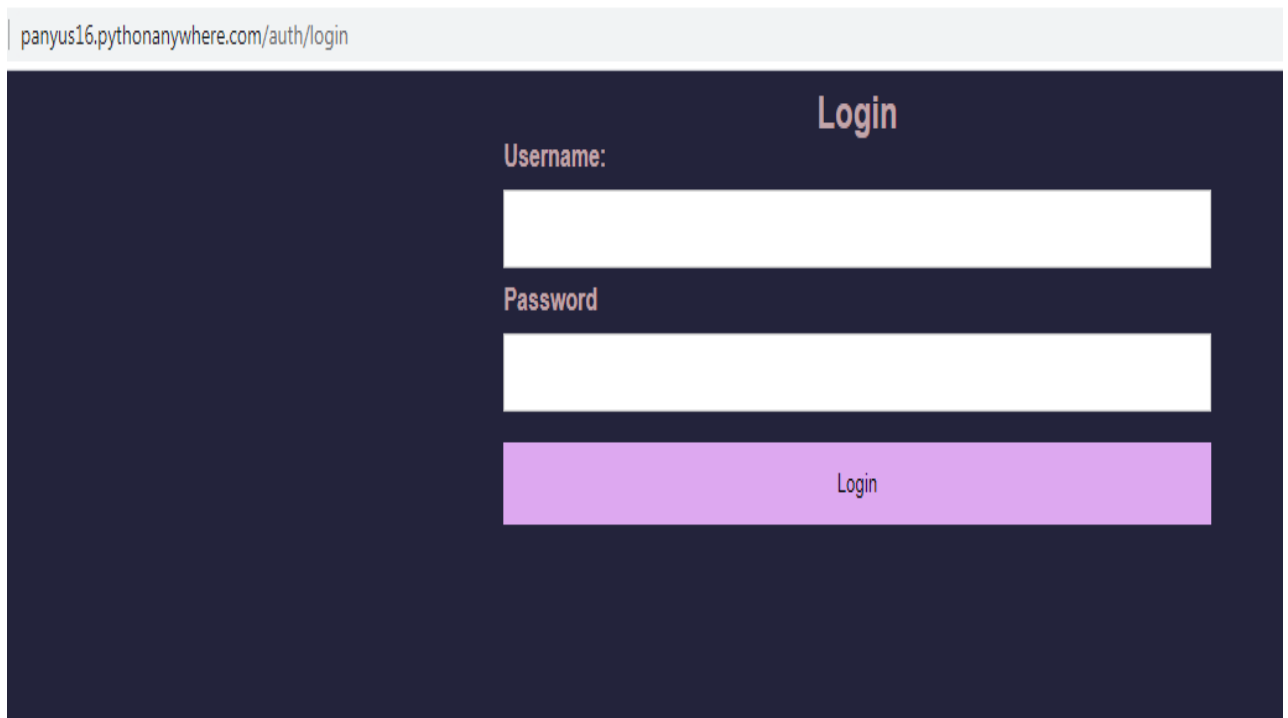
2.3 Створення та обробка форм засобами HTML та Python 3.7

Веб-форма – це певне гіпертекстова область, що дає можливість отримати фіт бек від користувача сайту до адміністратора. Отримання даних від користувача реалізується через використання елементів керування, а саме: кнопки, поля для тексту, блоки перевірки, що розташовуються у тезі <form>.

В загальному, веб-форма це є обмежена область сторінки сайту. Саме у цій обмеженій області користувач може вказати необхідні дані, і також обрати певні дії із запропонованих [5].

Для веб-сайту музичної групи «The Hardkiss» реалізовано 2 форми:

1. Форма входу адміністратора на сайт є зображена на рисунку 2.13. Дана форма складається з двох текстових полів: Ім'я користувача та Пароль, та кнопки, що дає змогу авторизуватися адміністратору.



The screenshot shows a web browser window with the address bar displaying `panyus16.pythonanywhere.com/auth/login`. The main content area has a dark blue background. At the top right, the word "Login" is written in a light blue font. Below it, the label "Username:" is followed by a white rectangular input field. Underneath, the label "Password" is followed by another white rectangular input field. At the bottom, there is a wide, light blue rectangular button with the word "Login" centered on it.

Рисунок 2.13 – Форма входу адміністратора на сайт

На цю форму можна перейти із пункту «Sign up» загального меню, що й показано на рисунку 2.14.

У лістингу 2.1 представлений HTML-код реалізації форми входу адміністратора на сайт.

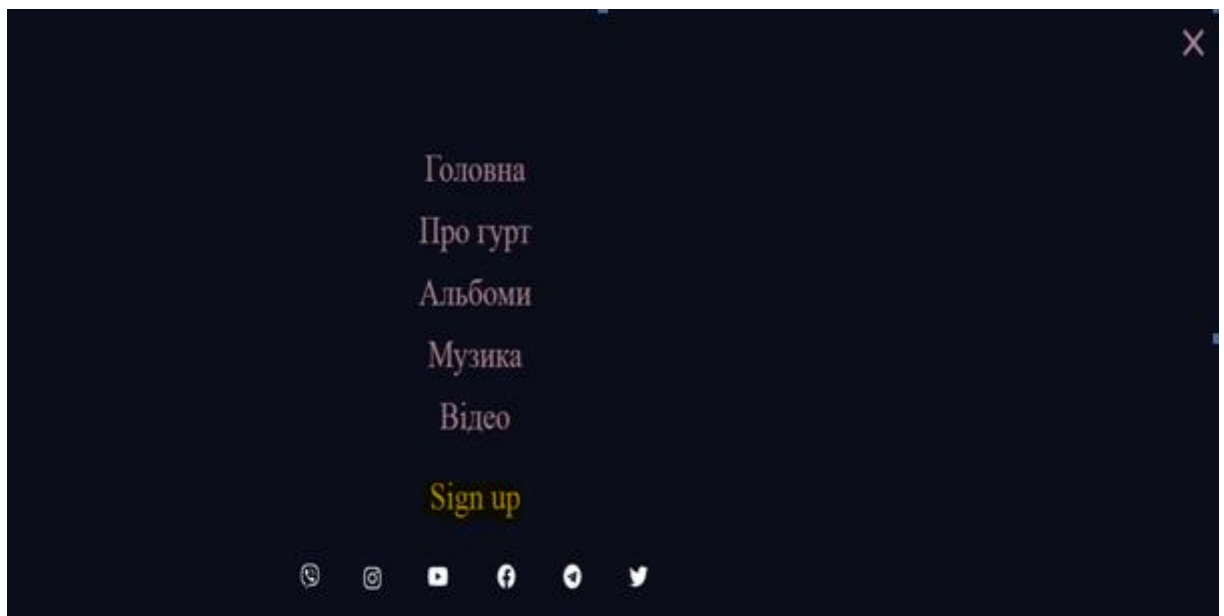


Рисунок 2.14 – Пункту «Sign up» загального меню

Лістинг 2.1 – HTML-код реалізації форми входу адміністратора на сайт

```
<form method="POST">
<h2 style="text-align: center; color:rgba(194, 165, 170,
1)">Login</h2>
<label for="username"><h4 style="color:rgba(194, 165, 170,
1)">Username: </h4></label>
<input type="text" name="username" id="username">
<label for="password"><h4 style="color:rgba(194, 165, 170,
1)">Password</h4></label>
<input type="password" name="password" id="password">
<input type="submit" value="Login">
</form>
```

2. Після авторизації адміністратора, для нього відкривається новий пункт меню (нова сторінка) – «Розсилка», де виводиться Форма для створення анонсу. Форма зображена на рисунку 2.15.

The screenshot shows a web browser window with the address bar displaying 'panyus16.pythonanywhere.com/sender'. The main content area has a dark blue background. At the top, the title 'Mail sender' is displayed in a light blue font. Below the title, there are two labels: 'Subject:' and 'Message body:'. The 'Subject:' label is followed by a single-line text input field. The 'Message body:' label is followed by a multi-line text area. At the bottom of the form, there is a purple button labeled 'Send'.

Рисунок 2.15 – Форма для створення анонсу

Дана форма складається з двох текстових полів, які заповнюються відповідно до анонсу.

У лістингу 2.2 – представлений HTML-код реалізації форми для створення анонсу

Лістинг 2.2 – HTML-код реалізації форми для створення анонсу

```
<form method="POST">
<h2 style="text-align: center; color:rgba(194, 165, 170, 1)">Mail
sender</h2>
<label for="subject"><h4 style="color:rgba(194, 165, 170,
1)">Subject: </h4></label>
<input type="text" name="subject" id="subject">
<label for="msg"><h4 style="color:rgba(194, 165, 170, 1)">Message
body:</h4></label>
<textarea name="msg" id="msg" cols="30" rows="10"></textarea>
<input type="submit" value="Send">
</form>
```

На рисунку 2.16 зображено приклад заповненої форми для створення анонсу.

Рисунок 2.16 – Приклад заповненої форми для створення анонсу

Далі, після натиску на кнопку «Send» текст цього анонсу автоматично надсилається на всі попередньо зареєстровані пошти, тобто до користувачів сайту, які підписалися на розсилку. Підписатися на розсилку дуже легко – це

можна зробити з Головної сторінки сайту The Hardkiss. Приклад на рисунку 2.17.

Дана розсилка є корисною функцією сайту, оскільки, це можливість інформувати фанів групи про можливі події, концерти чи зустрічі.

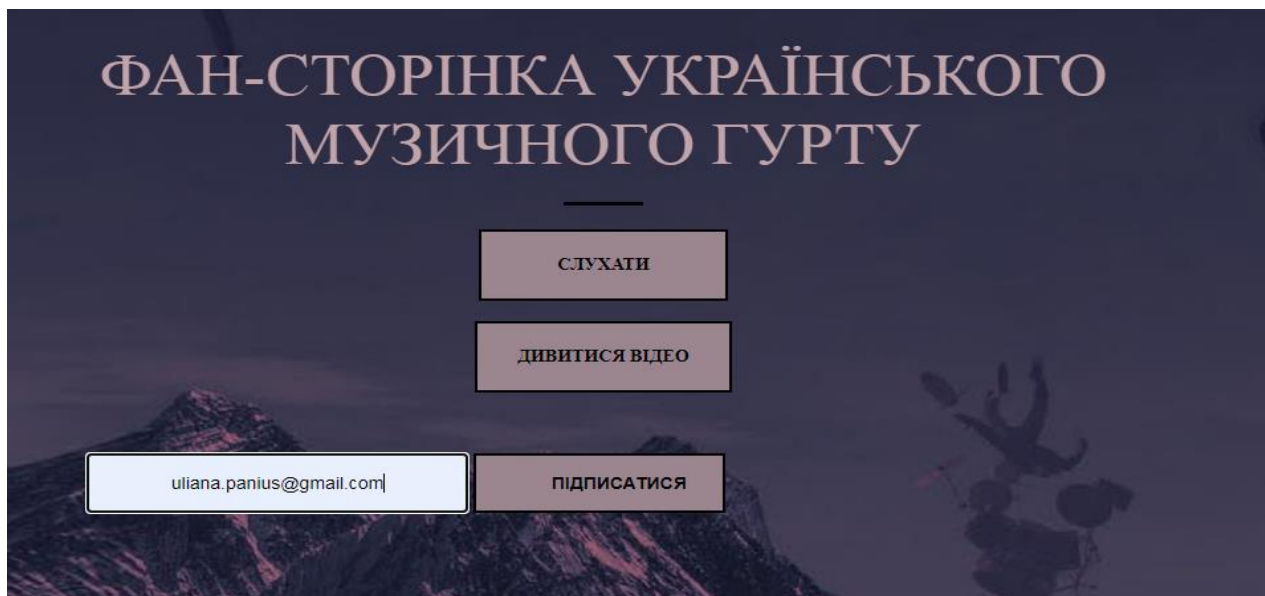


Рисунок 2.17 – Підписка на розсилку

На рисунку 2.18 зображено email повідомлення, що прийшло на пошту після створення анонсу. Дане повідомлення також дублюється і на пошту адміністратора.

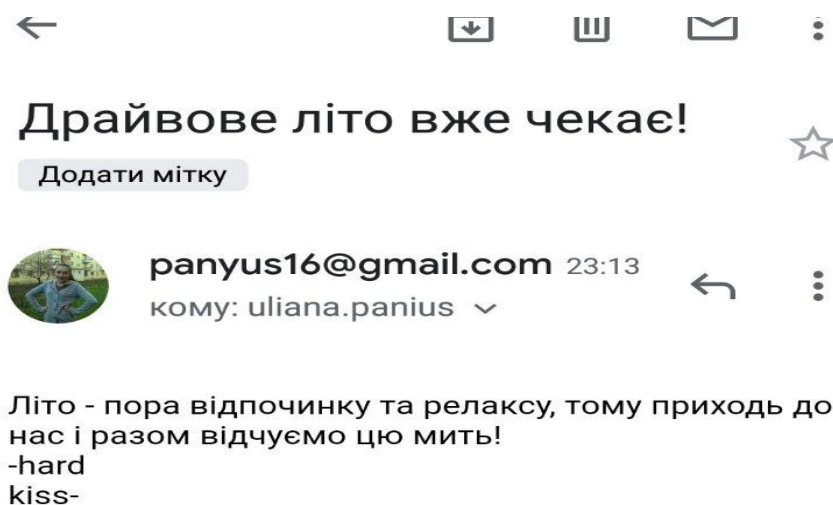


Рисунок 2.18 – Email повідомлення

2.4 Робота з базою даних веб-сайту музичної групи «The Hardkiss»

Для реалізації веб-сайту музичної групи «The Hardkiss» розроблено базу даних MySQL з використанням Python 3.7.

Дана база даних складається з двох таблиць. Реалізована як об'єктно-орієнтована модель. Для ініціалізації і підключення до бази даних запускається спеціальна бібліотека Python – SQLAlchemy, яка і обробляє базу даних (вихідні коди наведено в додатках).

У таблиці 2.1 та 2.2 представлено первинний варіант структури бази для зберігання даних адміністратора «Users» та «Subscribers» (попередньо зареєстрованого) з описом полів та типів даних, також додано опис призначення полів БД.

Таблиця 2.1 – Попередня структура бази даних «Users»

Ім'я поля	Тип даних	Призначення
id	integer	ID адмін сайту The Hardkiss
name	string	Ім'я доступу на сайт The Hardkiss
password	string	Пароль доступу на сайт The Hardkiss
valid	boolean	Відміна входу

Таблиця 2.2 – Попередня структура бази даних «Subscribers»

Ім'я поля	Тип даних	Призначення
id	integer	ID користувача сайту The Hardkiss
mail	string	адреса e-mail
valid	boolean	Відміна надсилання на певний емейл

На рисунку 2.19 представлений код реалізації бази даних.

```

1  from .db import db
2  class User(db.Model):
3      __tablename__ = "users"
4      id = db.Column(db.Integer, primary_key=True)
5      name = db.Column(db.String(80), nullable=False)
6      password = db.Column(db.String(100), nullable=False)
7      valid = db.Column(db.Boolean, nullable=False, default=0)
8      def __repr__(self):
9          return f"<User {self.id} - {self.name}>"
10     def __getitem__(self, item):
11         return getattr(self, item)
12     class Subscribers(db.Model):
13         __tablename__ = "subscribers"
14         id = db.Column(db.Integer, primary_key=True)
15         mail = db.Column(db.String(100), nullable=False)
16         valid = db.Column(db.Boolean, nullable=False, default=0)
17         def __repr__(self):
18             return f"<Subscriber {self.id} - {self.mail}>"
19         def __getitem__(self, item):
20             return getattr(self, item)
21

```

Рисунок 2.19 – Код реалізації бази даних

2.5 Валідація та тестування веб-сайту музичної групи «The Hardkiss»

При створенні веб-сайту важливо не просто переконатися, що дизайн гарний – повинна бути присутня повна впевненість, що сайт працює відмінно, перш ніж його випустять для широкої публіки. Саме для цього на веб-сайті проводиться ряд тестів, спеціально призначених для ознайомлення з усіма функціональними можливостями сайту, щоб спробувати знайти будь-яку проблему, велику чи маленьку, яка може змусити користувачів відчувати будь-які незручності [3].

Веб-тестування – це детальна перевірка сайту, мета якої знайти можливі помилки та недоліки та здійснити комплексне тестування веб-сайтів перед безпосереднім запуском. Веб-система повинна бути повністю перевірена від початку до кінця, перш ніж вона буде запущена для кінцевого користувача.

Також, в ході тестування визначається зручність і привабливість сайту для користувача, швидкість отримання та доступність потрібної інформації, надійність і безпеку використання. І, найголовніше, чи допомагає він досягти тих цілей, які були поставлені на початку проекту [14] [2].

Загальна мета веб-тестування – знайти проблеми, також відомі як помилки, проблеми або дефекти, які можуть мати негативний вплив на веб-сайт або додаток.

Веб-тестування також можна використовувати для виявлення певних областей або аспектів веб-сайту, які можна поліпшити і отримати кращі результати, будь то більше запитів, більше продажів, більше постійних відвідувачів, більш щасливі користувачі або щось ще [14].

Під валідацією сайту розуміється відповідність програмного коду веб-сайту всім встановленим і загальноприйнятим нормам розробки, верстки і веб-дизайну. А також тим умовам, які були висунуті і прописані в ТЗ веб-сайті, що розробляється.

На даний момент існує безліч програмного забезпечення для перевірки валідності, проте завжди можна скористатися онлайн-ресурсами [4].

Наприклад, інструмент W3C, допоможе здійснити оцінку валідності розмітки веб-сайту у таких форматах, як: HTML, XHTML, SMIL та MathML.

Наявні перевірки:

- аналіз всіх стилів та коду написання;
- онлайн перевірка веб-сайту на помилки.

На рисунку 2.20 зображено результати перевірки розробленого часу використовуючи сервіс - W3C markup validation [25].

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <http://panyus16.pythonanywhere.com/>

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by

<http://panyus16.pythonanywhere.com/>

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

1. **Error** Bad value 35% for attribute width on element img: Expected a digit but saw % instead.
From line 182, column 140; to line 182, column 200
`= "blank">=`
2. **Error** An img element must have an alt attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
From line 182, column 140; to line 182, column 200
`= "blank">=`
3. **Error** Bad value 35% for attribute width on element img: Expected a digit but saw % instead.
From line 183, column 74; to line 183, column 138
`= "blank">=`
4. **Error** An img element must have an alt attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
From line 183, column 74; to line 183, column 138
`= "blank">=`
5. **Error** Bad value 35% for attribute width on element img: Expected a digit but saw % instead.
From line 184, column 71; to line 184, column 133
`= "blank">=`

Рисунок 2.20 – Сервіс W3C markup validation

Згідно результатів валідації, найпоширеніші помилки – це встановлення розміру для зображення у відсотка (%), проте це не є глобальної проблемою, тому змінивши % на число можна уникнути цієї зауваги та відсутність атрибуту alt при img. Відсутність цього атрибуту не викликає глобальних помилок, оскільки використовується для підпису зображення та відображається і разі неможливості відкриття зображення.

Ще один сервіс – це CSS перевірка. Даний інструмент дає змогу перевірити CSS-код та HTML-документи, включаючи таблиці. Перевірка починається з аналізу розмітки в тому випадку, якщо стилі CSS інтегровані у HTML-код.

На рисунку 2.21 зображено результати перевірки розробленого часу використовуючи сервіс - CSS validator [13].

сервіс перевірки CSS від W3C

W3C результати перевірки CSS для <http://panyus16.pythonanywhere.com/> (CSS3 + SVG)

Перейти до: [Помилки \(1\)](#) [Попередження \(23\)](#) [Перевірений CSS](#)

W3C результати перевірки CSS для <http://panyus16.pythonanywhere.com/> (CSS3 + SVG)

На жаль, ми знайшли наступні помилки (1)

URI : <http://panyus16.pythonanywhere.com/static/style.css>

94	.section_header	Помилка значення : max-width лише 0 може бути unit. Ви повинні вказати після числа одиниці вимірювання : 950
----	-----------------	--

↑ НАВЕРХ

W3C DEVELOPERS

Interested in understanding what new technologies are coming out of W3C? Follow [@w3cdevs on Twitter](#) to keep track of what the future looks like!

Рисунок 2.21 – Сервіс CSS validator

Згідно результатів валідації, єдина помилка — відсутність одиниць вимірювання для «max-width», проте це не є глобальною проблемою, потрібно лише вказати «px» і заувага зникне.

2.6 Перенесення веб-сайту музичної групи «The Hardkiss» на хостинг

Перенесення сайту на хостинг починаю з реєстрації аканту на платформі Python Anywhere, що на рисунку 2.22.

PythonAnywhere - це інтегроване в Інтернеті середовище розробки та сервіс веб-хостингу на основі мови програмування Python [26].

Host, run, and code Python in the cloud!

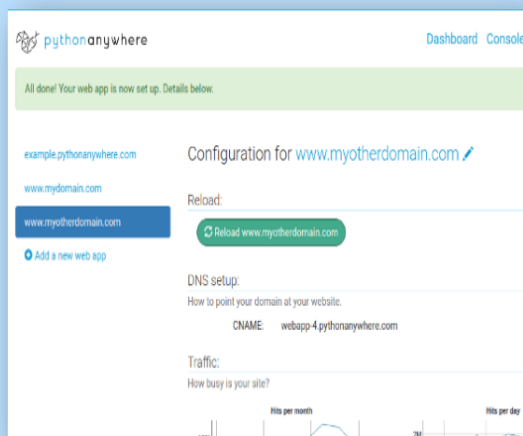
Get started for free. Our basic plan gives you access to machines with a [full Python environment](#) already installed. You can develop and host your website or any other code directly from your browser without having to install software or manage your own server.

Need more power? Upgraded plans start at \$5/month.

[Start running Python online in less than a minute! »](#)

[Watch our one-minute video »](#)

Not convinced? [Read what our users are saying!](#)



Start hosting quickly

Just write your application. No need to configure or maintain a web server – everything is set up and ready to go.

[More »](#)

Develop anywhere

Take your development environment with you! If you have a browser and an Internet connection, you've got everything you need.

[More »](#)

Teach and learn

PythonAnywhere is a fully-fledged Python environment, ready to go, for students and teachers – concentrate on teaching, not on installation hassles.

Amazing support

Need help with PythonAnywhere? If you get in touch, you can talk directly with the development team. Help for developers, from developers.

[More »](#)

Рисунок 2.22 – Платформа Python Anywhere

Починаю зі створення початого проекту – використовую `Fleks` бібліотеку Python 3.7 та встановлюю шлях на хості, де будуть зберігатися всі файли.

Наступний крок – вкладка `Файли`, на якій я завантажую всі файли проекту вручну, а саме – файли з розширенням `«.ру»` – це файли сервера, конфігурації та реалізації бази даних, зображено на рисунку 2.23.

На рисунку 2.24 зображенні файли безпосередньої реалізації сторінок сайту.

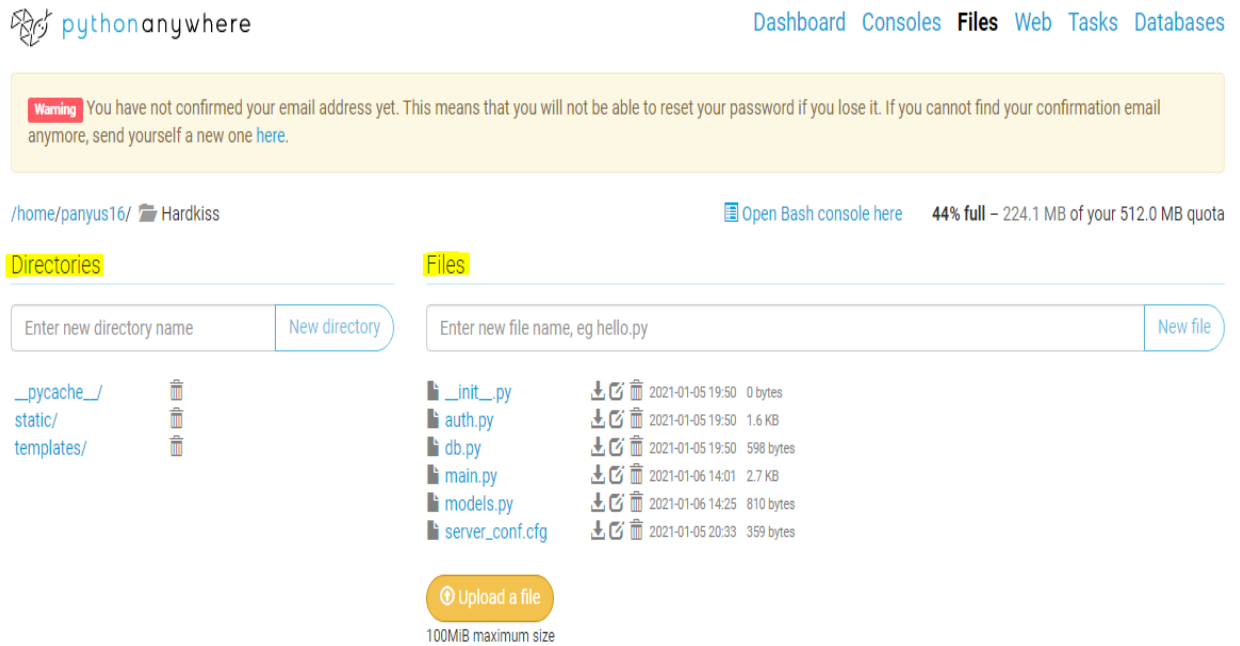


Рисунок 2.23 – Додавання файлів проекту

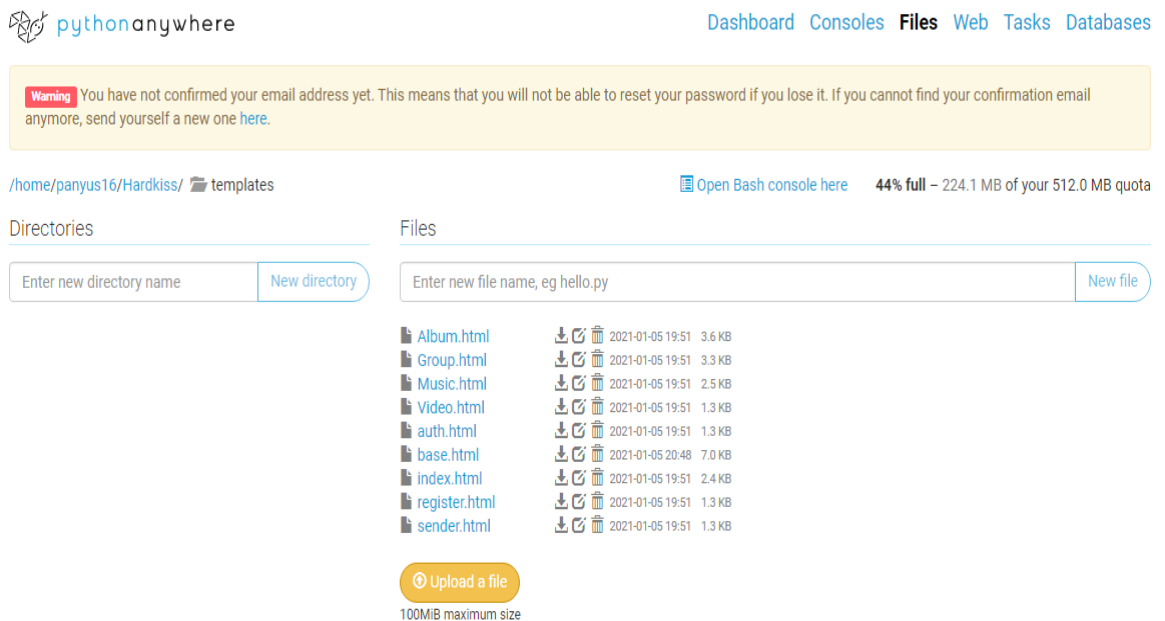


Рисунок 2.24 – Додавання файлів реалізації сторінок сайту

Наступний важливий крок – це конфігурація веб-сайту, в блоці зразу можна побачити адресу захощеного веб-сайту `panyus16.pythonanywhere.com`, що на рисункк 2.25.

Dashboard

Welcome, [panyus16](#)CPU Usage: 62% used – 62.14s of 100s. Resets in 1 hour, 41 minutes [More Info](#)[Upgrade Account](#)

File storage: 44% full – 224.1 MB of your 512.0 MB quota

Recent
Consoles

MySQL: panyus16\$default

Bash console 18467276

New console:

You can have up to 2 consoles. To get more, [upgrade your account](#)!

Recent
Files

/home/panyus16/Hardkiss/models.py

/home/panyus16/Hardkiss/server_conf.cfg

/var/www/panyus16_pythonanywhere_com_wsgi...

/home/panyus16/Hardkiss/main.py

/home/panyus16/Hardkiss/static/style_m.css

[+ Open another file](#)[Browse files](#)Recent
Notebooks

Your account does not support Jupyter Notebooks. [Upgrade your account](#) to get access!

All
Web apps[panyus16.pythonanywhere.com](#)[Open Web tab](#)

Рисунок 2.25 – Конфігурація веб-застосунку

Для цього потрібно додати віртуальне середовище для Python та завантажити необхідні бібліотеки для роботи, що зображено на рисунку 2.26 [12].

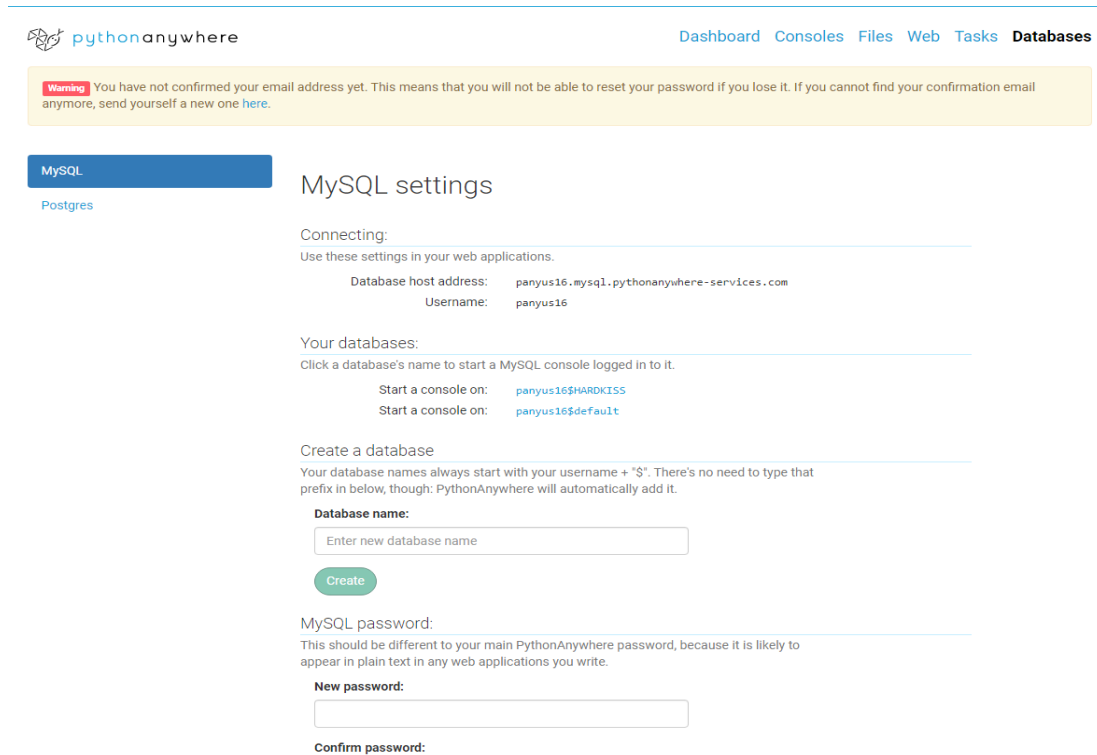
```

Bash console 18467276
(hardkiss) 14:06 ~ $ pip3 install flask_mail
Looking in links: /usr/share/pip-wheels
Collecting Flask_Mail
  Using cached Flask_Mail-0.9.1-py3-none-any.whl
Processing /usr/share/pip-wheels/blinker-1.4-py3-none-any.whl
Collecting Flask
  Using cached Flask-1.1.2-py2.py3-none-any.whl (94 kB)
Collecting click>=5.1
  Using cached click-7.1.2-py2.py3-none-any.whl (82 kB)
Collecting itsdangerous>=0.24
  Using cached itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2>=2.10.1
  Using cached Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
Collecting MarkupSafe>=0.23
  Using cached MarkupSafe-1.1.1-cp37-cp37m-manylinux1_x86_64.whl (27 kB)
Collecting Werkzeug>=0.15
  Using cached Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
Installing collected packages: MarkupSafe, Werkzeug, Jinja2, itsdangerous, click, Flask, blinker, Flask-Mail
Successfully installed Flask-1.1.2 Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.1 blinker-1.4 click-7.1.2 flask-mail-0.9.1 itsdangerous-1.1.0
(hardkiss) 14:06 ~ $ pip3 install flask_sqlalchemy
Looking in links: /usr/share/pip-wheels
Collecting flask_sqlalchemy
  Using cached Flask_SQLAlchemy-2.4.4-py2.py3-none-any.whl (17 kB)
Requirement already satisfied: Flask>=0.10 in ./virtualenvs/hardkiss/lib/python3.7/site-packages (from flask_sqlalchemy) (1.1.2)
Requirement already satisfied: Werkzeug>=0.15 in ./virtualenvs/hardkiss/lib/python3.7/site-packages (from Flask==0.10->flask_sqlalchemy) (1.0.1)
Requirement already satisfied: click>=5.1 in ./virtualenvs/hardkiss/lib/python3.7/site-packages (from Flask==0.10->flask_sqlalchemy) (7.1.2)
Requirement already satisfied: Jinja2>=2.10.1 in ./virtualenvs/hardkiss/lib/python3.7/site-packages (from Flask==0.10->flask_sqlalchemy) (2.11.2)
Requirement already satisfied: itsdangerous>=0.24 in ./virtualenvs/hardkiss/lib/python3.7/site-packages (from Flask==0.10->flask_sqlalchemy) (1.1.0)
Requirement already satisfied: MarkupSafe>=0.23 in ./virtualenvs/hardkiss/lib/python3.7/site-packages (from Jinja2>=2.10.1->Flask==0.10->flask_sqlalchemy) (1.1.1)
Collecting SQLAlchemy>=0.8.0
  Using cached SQLAlchemy-1.3.22-cp37-cp37m-manylinux2010_x86_64.whl (1.3 MB)
Installing collected packages: SQLAlchemy, flask-sqlalchemy
Successfully installed SQLAlchemy-1.3.22 flask-sqlalchemy-2.4.4
(hardkiss) 14:06 ~ $ pip3 install mysql-connector-python
Looking in links: /usr/share/pip-wheels
Collecting mysql-connector-python
  Using cached mysql_connector_python-8.0.22-cp37-cp37m-manylinux1_x86_64.whl (18.0 MB)
Collecting protobuf>=3.0.0
  Using cached protobuf-3.14.0-cp37-cp37m-manylinux1_x86_64.whl (1.0 MB)
Collecting six>=1.9
  Using cached six-1.15.0-py2.py3-none-any.whl (10 kB)
Installing collected packages: six, protobuf, mysql-connector-python

```

Рисунок 2.26 – Встановлення необхідних бібліотек

Додаю базу даних, вибравши вкладку Databases як на рисунку 2.27.



pythonanywhere

Dashboard Consoles Files Web Tasks Databases

Warning You have not confirmed your email address yet. This means that you will not be able to reset your password if you lose it. If you cannot find your confirmation email anymore, send yourself a new one [here](#).

MySQL

Postgres

MySQL settings

Connecting:

Use these settings in your web applications.

Database host address: panyus16.mysql.pythonanywhere-services.com
Username: panyus16

Your databases:

Click a database's name to start a MySQL console logged in to it.

Start a console on: panyus16\$HARDKISS
Start a console on: panyus16\$default

Create a database

Your database names always start with your username + '\$'. There's no need to type that prefix in below, though. PythonAnywhere will automatically add it.

Database name:

Enter new database name

Create

MySQL password:

This should be different to your main PythonAnywhere password, because it is likely to appear in plain text in any web applications you write.

New password:

Confirm password:

Рисунок 2.27 – Вкладка додавання бази даних

На рисунку 2.28 зображено код конфігураційного файлу server_conf.cfg.

```

1 SECRET_KEY="98>87e9e0b6H$dcf9ee6wc0g~"
2 USERNAME="panyus16"
3 PASSWORD="1234qwer"
4 HOST="panyus16.mysql.pythonanywhere-services.com"
5 DB_NAME="panyus16$HARDKISS"
6 SQLALCHEMY_TRACK_MODIFICATIONS=False
7 MAIL_SERVER='smtp.gmail.com'
8 MAIL_PORT=465
9 MAIL_USERNAME='panyus16@gmail.com'
10 MAIL_PASSWORD='cyploguhdyirns'
11 MAIL_USE_TLS=False
12 MAIL_USE_SSL=True
13 INIT_DB=False
14
15

```

Рисунок 2.28 – Код конфігураційного файлу server_conf.cfg

Сайт успішно встановився на хостинг і на рисунку 2.29 зображено Головну сторінку сайту.

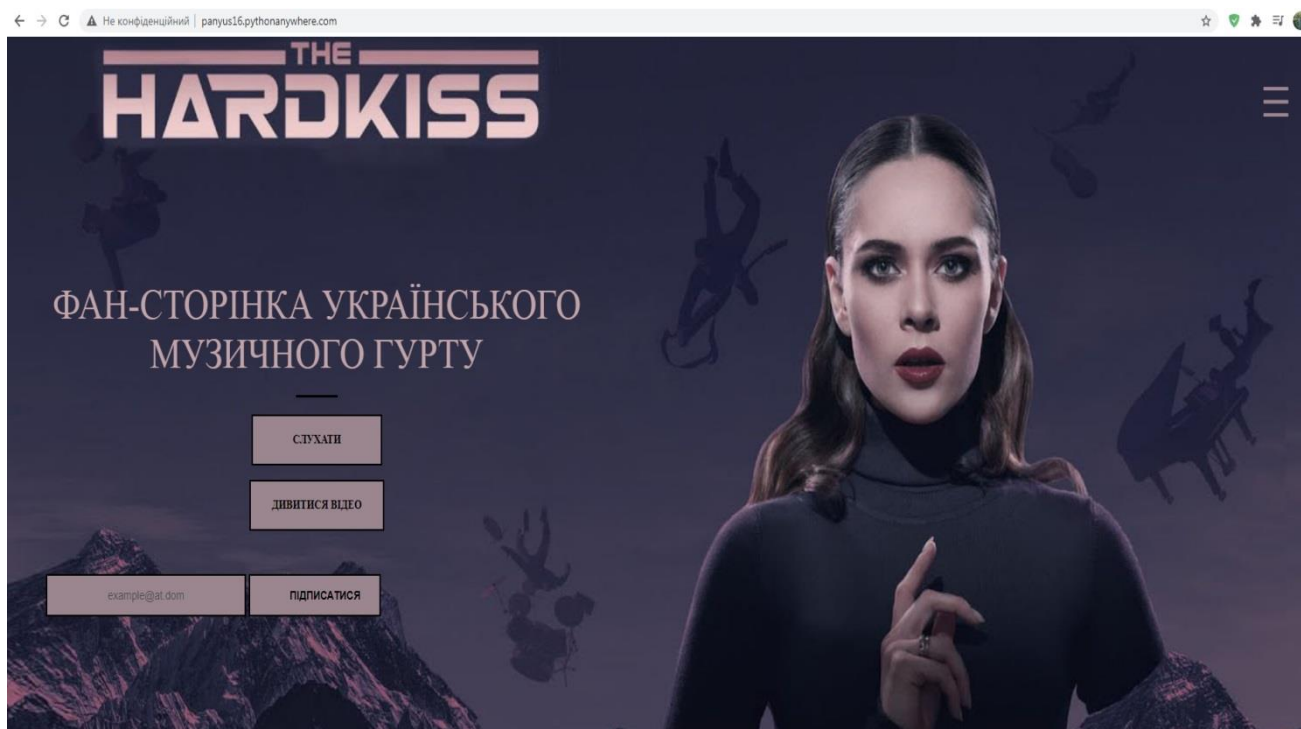


Рисунок 2.29 – Головна сторінка сайту на хості

2.7 Висновок до другого розділу

В ході виконання даного розділу було здійснено проектування архітектури веб-сайту музичної групи «The Hardkiss». Розроблено дизайн сайту за допомогою інструменту Figma та виконано практичну реалізацію архітектури.

Здійснено роботу з базою даних та обробкою форм, для розсилки анонсів подій. Проведено тестування сайту та успішне перенесення на хостинг.

РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

3.1 Контроль за станом охорони праці

Належне управління охороною праці можна здійснювати лише при умові достатньої, повної та своєчасної інформації про стан охорони праці на об'єкті. Виявити порушення, відхилення, дати приписи чи рекомендації з ОП можливо лише на підставі регулярного контролю, контроль стану охорони праці є найбільш відповідальною ланкою управління.

Підприємство підлягає постійному державному нагляду за охороною праці через його територіальні управління та інспекторів.

Разом з тим, здійснюється також зовнішній та внутрішній контроль за станом охорони праці на виробництві.

Зовнішній контроль за станом охорони праці на виробництві поділяється на:

- відомчий, який здійснюють посадові особи, повноважні представники міністерств і відомств даної галузі, головних управлінь, асоціацій, концернів власників підприємств;
- регіональний, який здійснюють посадові особи, повноважні представники місцевих органів виконавчої влади;
- страховий, який здійснюється страховими експертами з охорони праці Фонду соціального страхування від нещасних випадків на виробництві та професійних захворювань;
- громадський, який здійснюється виборними органами профспілок та інших громадських організацій.

Внутрішній контроль за станом охорони праці на виробництві здійснюють:

- повсякденний (оперативний) контроль забезпечують керівники підрозділів та робіт, інші посадові особи на своїх робочих місцях відповідно до посадових обов'язків;

– громадський внутрішній контроль на підприємстві здійснюють громадські інспектори з ОП або уповноважені представники трудового колективу.

Також є триступеневий адміністративно-громадський контроль за охороною праці на виробництві здійснюється за такою схемою:

– перший ступінь - протягом робочої зміни або робочого дня контролюється хоча б один раз кожне робоче місце;

– другий ступінь - здійснюється контроль не рідше одного разу на тиждень кожного структурного підрозділу начальником цього структурного підрозділу (відділу, підрозділу) і громадським інспектором трудового колективу або профспілки підприємства чи структурного підрозділу;

– третій ступінь - здійснюється один раз на місяць (або в інший термін, визначений колективним договором) в обсягу кожного робочого місця всього підприємства керівництвом підприємства (роботодавцем, головним інженером, заступником головного інженера з охорони праці) і відділом охорони праці підприємства.

Служба охорони праці підприємства забезпечує постійний нагляд і контроль виконання вимог з охорони праці у всіх структурах підприємства, забезпечує виконання цих вимог керівництвом.

Охорона праці є дуже важливим питанням в контексті впливу робочого середовища на програміста при розробці веб-сайту. Саме тому контроль за її станом є невід’ємним елементом для виконання посадових обов’язків [7].

3.2 Загальгі вимоги безпеки з охорони праці для користувачів ПК

Техніка безпеки є важливою складовою охорони праці, яка розглядає як технічні так й організаційні методи гарантування безпеки праці. Основною метою заходів з техніки безпеки є профілактика травматизму, тобто запобігання нещасним випадкам на виробництві.

Робота з комп'ютером характеризується значною розумовою напругою і нервово-емоційним навантаженням операторів, високою напруженістю зорової роботи і достатньо великим навантаженням на м'язи рук при роботі з клавіатурою .

У процесі роботи з комп'ютером необхідно дотримувати правильний режим праці та відпочинку. В іншому випадку у персоналу наголошуються значна напруга зорового апарату з появою скарг на незадоволеність роботою, головні болі, дратівливість, порушення сну, втому і хворобливі відчуття в очах, в поясниці, в області ший і руках.

Необхідно дотримуватись певних вимог, а саме:

– система гігієнічних вимог – бо тривала робота з комп'ютером призводить до розладів стану здоров'я. Робота за комп'ютером з грубими порушенням гігієнічних норм і правил, призводить до підвищеного стомлення. Шкідливий вплив комп'ютерної системи на організм людини є комплексним. Обладнання робочого місця впливає на органи опорно-рухової системи;

– вимоги до робочого місця, оскільки, кідливими можуть бути всі частини комп'ютера (не тільки монітор). Процесор генерує промені, що поширюються у просторі у вигляді електромагнітних хвиль, часто несучи дезінформацію електромагнітного поля людини. Через нагрівання материнської плати і корпусу відбувається де іонізація повітря та виділення в навколишнього середовища шкідливих речовин. Саме тому необхідно провітрювати середовище, де встановлене комп'ютерне обладнання. Робочі місця слід розташовувати так, щоб уникнути попадання в очі прямого природного або штучного світла. Джерела освітлення рекомендується розташовувати з обох боків екрану паралельно напрямку погляду. Робочі місця мають бути розташовані між собою на відстані не менше 1,5 м, на відстані не менше 1,5 м від стіни з вікнами, від інших стін на відстані 1м. Відносно вікон робоче місце доцільно розташовувати таким чином, щоб природне світло падало на нього збоку (переважно зліва);

– вимоги до розташування монітору, бо він виділяє небезпечні випромінювання і концентрація їх дії залежить від того, як монітор розташований до користувача.

Передня сторона монітора має захисне покриття, а от задня стінка і бічні поверхні не захищені.

Монітор встановлюється прямо перед користувачем і не вимагає повороту голови або корпусу тіла (на рисунку 3.1 зображено розташування монітора відносно користувача), а також робочий стіл і посадочне місце має висоту, щоб рівень очей користувача знаходився трохи вище центру монітора.

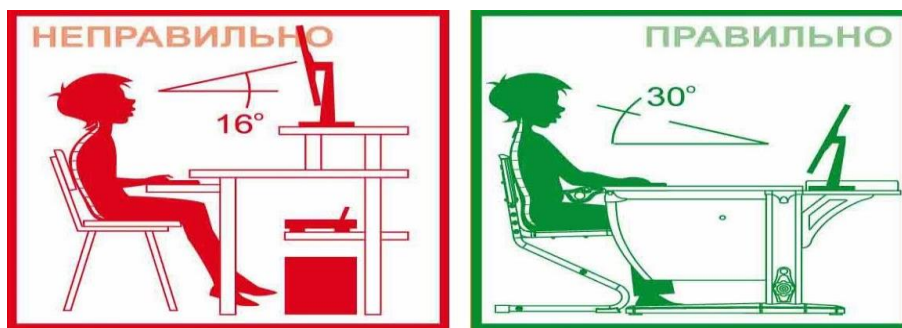


Рисунок 3.1 – Розташування монітора відносно користувача

Клавіатура розташовується на такій висоті, щоб пальці рук розташовувалися на ній вільно, без напруги, а кут між плечем і передпліччям становив 100° - 110° . Для роботи рекомендується використовувати спеціальні комп'ютерні столи, що мають висувні полицьки для клавіатури.

Дотримання техніки безпеки при розробці веб-сайту є важливим аспектом, оскільки правильне положення за робочим столом, освітлення та дотримання правильного режиму та відпочинку є головним фактором, що впливає на стан здоров'я розробника [15].

3.3 Критичні стани людини

До критичних станів відносять непритомність, шок, стенокардію, гіпертонічний криз. В даному розділі розглянути критичні стани, які можуть

виникнути в розробників/користувачів ПК при тривалій і стресовій роботі за комп'ютером.

Критичні стани:

1. Непритомність (зімління) - найбільш поширений і легкий прояв гострої судинної недостатності внаслідок раптового короткочасного малокрів'я головного мозку.

Виникає при крововтратах, травмах, при хвилюванні, духоті, голодуванні, перевтомленні, захворюванні серцево-судинної системи.

Перша допомога:

- горизонтальне положення, доступ свіжого повітря;
- звільнення від тісного одягу (комір, ремінь - послабити), підняти ноги вище тулуба;
- скропити холодною водою, дати понюхати ватку, змочену нашатирним спиртом;
- напоїти гарячим чаєм, кавою, розтерти руки, ноги, дати грілку.

2. Шок – це комплекс грізних симптомів, які супроводжуються різким порушенням нервової регуляції життєво важливих функцій органів і систем. При шоку передусім страждає ЦНС.

Перша допомога:

- усунути джерело патологічної дії на потерпілого,
- покласти, упевнитись у прохідності дихальних шляхів,
- зупинити кровотечу, ввести знеболювальне, іммобілізувати травмовані кінцівки.

3. Стенокардія - гостра ішемія міокарду, зумовлена погіршенням його кровопостачання з наступним швидким відновленням кровообігу в зоні ішемії.

Перша допомога:

- заспокоїти, забезпечити доступ свіжого повітря, сидяче положення;
- валідол або нітрогліцерин під язик, серцеві краплі;
- на груди, ділянку серця - гірчичники, для рук і ніг — гарячу ванну.

4. Гіпертонічний криз. У здорової людини в нормі артеріальний тиск = 120/80 мм.рт.ст. Максимально допустимий рівень АТ=140/90. Підвищення АТ понад максимальну норму називається гіпертонічною хворобою. Загострення гіпертонічної хвороби спричиняється тривалим хвилюванням, нервовим перевантаженням і призводить до піднесення АТ до 170/95 - 200/120, що називається гіпертонічним кризом.

Перша допомога:

- хворого покласти і створити повний фізичний і психічний спокій;
- забезпечити доступ свіжого повітря, масаж шиї і потиличної ділянки;
- гірчичники на шию, потилицю і литкові м'язи;
- холодний компрес до голови;
- транспортування напівсидячи.

Розробка веб-сайту – це досить стресова і виснажлива робота. При постійній напрузі та в схвилюваному стані, можуть виникати критичні стани і надання першої домедичної допомоги може бути життєво необхідним навиком при виникненні такого стану.

ВИСНОВКИ

В процесі роботи над дипломною роботою було розроблено та реалізовано веб-сайт музичної групи «The Hardkiss».

У першому розділі сформовано постановку завдання розроблення веб-сайту музичної групи «The Hardkiss», а саме:

- найменування та область застосування - розробка веб-сайту для фанів музичної групи «The Hardkiss»;
- розроблюваний веб-сайт містить всю інформацію про гурт, його музику, відео роботи та фото галерею, також з можливістю заповнення форми для розсилки анонсів на вказану пошту;
- вибір засобів та обґрунтування їх використання для реалізації веб-сайту.

У другому розділі детально описано проектування веб-сайту музичної групи «The Hardkiss», а саме:

- розробку та реалізацію архітектури веб-сайту музичної групи «The Hardkiss»;
- створення та обробка форм засобами HTML та Python 3.7;
- робота з базою даних веб-сайту музичної групи «The Hardkiss».
- тестування веб-сайту музичної групи «The Hardkiss»;
- здійснено валідацію (перевірка на помилки) за допомогою онлайн ресурсів ;
- короткий опис та усунення помилок веб-сайту музичної групи «The Hardkiss»;
- перенесено веб-сайт музичної групи The Hardkiss на хостинг.

ПЕРЕЛІК ДЖЕРЕЛ

- 1 Архітектура сайту [Електронний ресурс] – Режим доступу до ресурсу: <http://www.plug.org.ua/documentation/about-python> – Дата доступу: 25.02.2021.
- 2 Валідація сайту і її вплив на працездатність ресурсу [Електронний ресурс] – Режим доступу до ресурсу: <https://www.internet-technologies.ru/articles/validaciya-sayta-i-ee-vliyanie-na-rabotosposobnost-resursa.html> – Дата доступу: 15.03.2021.
- 3 Валідація. Тестування валідації [Електронний ресурс] – Режим доступу до ресурсу: <https://training.qatestlab.com/blog/technical-articles/validation-testing/> – Дата доступу: 15.03.2021.
- 4 Валідація і тестування сайтів: 20 онлайн-інструментів [Електронний ресурс] – Режим доступу до ресурсу: <https://www.internet-technologies.ru/articles/validaciya-i-testirovanie-saytov-20-onlayn-instrumentov.html> – Дата доступу: 15.03.2021.
- 5 Ваша перша HTML форма [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/ru/docs/Learn/Forms/Your_first_form - Дата доступу: 15.03.2021.
- 6 Все про веб-сайти [Електронний ресурс] – Режим доступу до ресурсу: <http://repair.lviv.ua/stvorenniya-sajtiv/osnovni-etapi-stvorenniya-sajtu/> – Дата доступу: 20.02.2021.
- 7 Види контролю за станом охорони праці на підприємстві [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/5204052/page:8/> – Дата доступу: 15.05.2021.
- 8 Вимоги до створення сайту [Електронний ресурс] – Режим доступу до ресурсу: <https://recommerce.com.ua/vimogi-do-stvorenniya-saitu>. – Дата доступу: 20.02.2021.
- 9 Довідник по HTML [Електронний ресурс] – Режим доступу до ресурсу: <http://htmlbook.ru/html> – Дата доступу: 01.03.2021.

10 Етапи створення сайту [Електронний ресурс] – Режим доступу до ресурсу: <https://recommerce.com.ua/etapi-stvorenniya-saitu> – Дата доступу: 20.02.2021.

11 Основні етапи створення сайту [Електронний ресурс] – Режим доступу до ресурсу: <http://repair.lviv.ua/stvorenniya-sajtiv/osnovni-etapi-stvorenniya-sajtu/> – Дата доступу: 20.02.2021.

12 Работа с MySQL в Python [Електронний ресурс] – Режим доступу до ресурсу: <https://khashtamov.com/ru/mysql-python/> – Дата доступу: 28.03.2021.

13 Сервіс перевірки CSS від W3C [Електронний ресурс] – Режим доступу до ресурсу: <http://jigsaw.w3.org/css-validator/validator?uri=http%3A%2F%2Fpanyus16.pythonanywhere.com%2F&profile=css3svg&usermedium=all&warning=1&vextwarning=&lang=uk> – Дата доступу: 01.03.2021.

14 Тестування сайту або як не зафакапитися [Електронний ресурс] – Режим доступу до ресурсу: <https://polyarix.com/blog/testirovanie-sajta/> – Дата доступу: 16.04.2021

15 Техніка безпеки [Електронний ресурс] – Режим доступу до ресурсу: <https://sites.google.com/site/slusarpr/vopros-otvet> – Дата доступу: 15.05.2021

16 Топ фреймворки для веб-розробки в 2018 році [Електронний ресурс] – Режим доступу до ресурсу: <https://jetruby.com/ru/blog/top-freimworki-2018/>. – Дата доступу: 01.03.2021.

17 Хочу стати FrontEnd розробником [Електронний ресурс] – Режим доступу до ресурсу: <https://tproger.ru/curriculum/intro-to-frontend-development/>. – Дата доступу: 04.02.2020..

18 Що таке Python? [Електронний ресурс] – Режим доступу до ресурсу: https://ua.kulikoff.com.ua/glossary_view/33/ – Дата доступу: 01.03.2021.

19 Що таке jQuery [Електронний ресурс] – Режим доступу до ресурсу: <https://phoneinfo8.info/sho-take-jquery/> – Дата доступу: 21.03.2021.

20 Figma [Електронний ресурс] – Режим доступу до ресурсу: <https://www.figma.com/> – Дата доступу: 28.02.2021.

21 Flask [Электронный ресурс] – Режим доступа до ресурсу: <https://flask.palletsprojects.com/en/1.1.x/> – Дата доступа: 21.03.2021.

22 JavaScript [Электронный ресурс] – Режим доступа до ресурсу: <https://learn.javascript.ru/intro>. – Дата доступа: 10.03.2021.

23 JavaScript Tutorial [Электронный ресурс] – Режим доступа до ресурсу: <https://www.w3schools.com/js/default.asp> – Дата доступа: 10.03.2021.

24 Angular [Электронный ресурс] – Режим доступа до ресурсу: <https://angular.io/docs> – Дата доступа: 10.03.2021.

25 Markup Validation Service [Электронный ресурс] – Режим доступа до ресурсу: <https://validator.w3.org/> - Дата доступа: 05.12.2020. MySQL [Электронный ресурс] – Режим доступа до ресурсу: <https://en.wikipedia.org/wiki/MySQL> – Дата доступа: 10.05.2021.

26 Pythonanywhere [Электронный ресурс] – Режим доступа до ресурсу: www.pythonanywhere.com – Дата доступа: 21.05.2021.

27 Ssqlalchemy [Электронный ресурс] – Режим доступа до ресурсу: <https://www.sqlalchemy.org> – Дата доступа: 21.03.2021.

28 How to Use Python: Your First Steps [Электронный ресурс] – Режим доступа до ресурсу: <https://realpython.com/python-first-steps/> – Дата доступа: 21.03.2021.

29 The basic stages for building a webpage [Электронный ресурс] – Режим доступа до ресурсу: <https://www.designcontest.com/blog/the-basic-stages-for-building-a-webpage/> – Дата доступа: 21.03.2021.

30 How to Code Your Own Website [Электронный ресурс] – Режим доступа до ресурсу: <https://www.quackit.com/make-your-own-website/> – Дата доступа: 21.03.2021.

ДОДАТКИ

Лістинг файлу index.html

```

{% extends 'base.html' %}
{% block title%} Main page {% endblock %}
{% block head %}<link rel="stylesheet" href="{{ url_for('static',
filename='style.css') }}">
<style>
#mail {
border-width: 2px;
background-color: #9B8690;
padding: 15px 57px;
border: 2px solid #000000;
font-size: 14px;
color: #000000;
text-align: center;
}
#mail::placeholder{
font-size: 14px;
color: #4f4f4f;
text-align: center;
}
</style>
{% endblock %}
{% block content %}
<div class="intro">
<div class="container">
<div class="intro_inner">
<h2 class="intro_title"> Фан-сторінка українського музичного
гурту </h2>
<a class="btn" href="{{ url_for('music') }}"> <strong>Слухати
</strong></a>
<br>
<br>
<a class="btn1"
href="https://www.youtube.com/channel/UCgJxsxqbS_lQ3Y-_gbEq-
lA"><strong> Дивитися відео </strong> </a>
<br><br><br><br>
<form method='post' style='text-align: left; margin: 0 60px;'>
<input type='email' name='mail' id='mail'
placeholder='example@at.dom' required>
<input type='submit' class='btn' style='font-weight:bold;
width:190px; padding: 15px 57px;' value='Підписатися'>
</form>
</div>
</div>
</div>
<section class="section">
<div class="container">
<div class="section_header">

```

```

<div class="section_text">
<p> <center> Вже десять років пісні гурту стають саундтреками
найрізноманітніших моментів життя багатьох людей.
Вони допомагають пережити труднощі, супроводжують радощі та
успіхи, просто надихають і надають сили.
Як же часто ми чуємо такі історії від слухачів. Всі ці саундтреки
прозвучать за рік на НСК «Олімпійський».
Ми об'єднаємо всіх своїх друзів з такими різними історіями. Тих,
хто любить драйвовий Hard і тих, хто обожнює ніжний Kiss.
</center><br><br>- Юлія Саніна </p>
</div>
</div>
</div>
</section>
<div > <video src="{{ url_for('static', filename='Video/THE
HARDKISS1.mp4') }}" style="margin: 0; min-width: 100%; min-
height:100%" controls></video>
</div>
{% endblock %}

```

Лістинг файлу Groupe.html

```

{% extends 'base.html' %}
{% block title%} Group {% endblock %}
{% block head %}
<link rel="stylesheet" href="{{ url_for('static',
filename='gallery/gallery.css') }}">
<script src="{{ url_for('static', filename='gallery/gallery.js')
}}"></script>
<style>
body {
font-family: 'Lato', sans-serif;
background: url("{{ url_for('static', filename='photo/hard2.jpg')
}}") fixed;
background-size: cover;
}
main{
margin: auto;
width: 75%;
background-color: rgba(69, 64, 87, 0.85);
color:white;
min-height: 2000px;
margin-top:1%;
padding: 10px
}
h2 {
margin: 50px;
}
h1 {
margin: 50px;
}

```

```

}
.parent {
width: 100%;
display: flex;
flex-wrap: wrap;
align-items: center;
justify-content: center;
margin: 0 10px 10px 10px;
}
.item {
min-width: calc(100%/5);
padding: 10px;
text-align:center;
}
.item p {
max-width: 200px;
margin: 5px auto;
}
.item img {
width: 300px;
height: 500px;
object-fit: cover;
}
</style>
{% endblock %}
{% block content %}
<main>
<b>
<hr>
</b>
<h1>
<center> The Hardkiss</center>
</h1>
<hr>
<p>
<h2>
<p align="left">"The HARDKISS – український музичний гурт,
створений у 2011 році. Усі пісні гурту написані солісткою
Юлією Саніною та Валерієм Бебком,
який також є креативним продюсером «The Hardkiss».
<h2>
<h1>
<p align="justify"> Склад групи:</h1>
<div class="content" style="margin: auto">
<div class="parent">
<div class="item">
<p>Валерій Бебко</p>
</div>
<div class="item">
<p>Юлія Саніна</p>
</div>

```

```

<div class="item">
<p>Клим Лисюк</p>
</div>
<div class="item">
<p>Кібелев Євген</p>
</div>
</div>
</div>
<h1><p align="justify"> Досягнення та нагороди:</p></h1>
<div class="container p-3 parent">
<div class="fg-gallery parent">
{% for i in range(1, 13) %}

{% endfor %}
</div>
</div>
<script>
var a = new FgGallery('.fg-gallery', {
cols: 4,
style: {
border: '10px solid #fff',
height: '180px',
borderRadius: '5px',
boxShadow: '0 2px 10px -5px #858585',
}
})
var a = new FgGallery('.ns', {
cols: 3,
style: {
border: '0 solid #fff',
height: '240px',
borderRadius: '5px',}})
</script>
</main>
{% endblock %}

```

Лістинг файлу Album.html

```

{% extends 'base.html' %}
{% block title%} Albums {% endblock %}
{% block head %}
<link rel="stylesheet" href="{{ url_for('static',
filename='style_m.css') }}">
<style>
.parent {
width: 100%;
display: flex;

```

```

flex-wrap: wrap;
align-items: center;
justify-content: center;
margin: 0 20px 20px 20px;
}
.item {
min-width: calc(100%/3);
padding: 10px;
text-align:center;
}
.item p {
max-width: 200px;
margin: 5px auto;
}
.item img {
width: 300px;
height: 200px;
object-fit: cover;}
.but{
padding: 10px 10px;
border: 3px solid #000000;
font-size: 14px;
color: #000000;
text-transform: uppercase;
text-decoration: none;
border-width: 2px 2px 2px 2px;
background-color: #9B8690;
}
a, a:hover, a:focus, a:active {
text-decoration: none;
color: #000000 ;
}
</style>
{% endblock %}
{% block content %}
<div class="intro">
<div class="container">
<h1>
<center> Альбоми</center>
</h1>
<div class="content" style="margin: auto">
<div class="parent">
<div class="item">
<p>Альбом "Stones and Honey" 2014 pik</p>
<button class="but"><a target="_blank" rel="noopener noreferrer"

href="https://uk.wikipedia.org/wiki/Stones_and_Honey">
Детальніше</a>
</button>
</div>
<div class="item">

```

```

<p>Альбом "Cold Altair" 2015 рік</p>
<button class="but"><a target="_blank" rel="noopener noreferrer"
href="https://uk.wikipedia.org/wiki/Cold_Altair_(%D0%BC%D1%96%D0%B
D%D1%96%D0%B0%D0%BB%D1%8C%D0%B1%D0%BE%D0%BC)">Детальніше</a></butt
on>
</div>
<div class="item">
<p>Альбом "Perfection is a Lie" 2017 рік</p>
<button class="but"><a target="_blank" rel="noopener noreferrer"
href="https://uk.wikipedia.org/wiki/Perfection_is_a_Lie">
Детальніше</a>
</button>
</div>
<div class="item">
<p>Альбом "Залізна ластівка" 2018 рік</p>
<button class="but"><a target="_blank" rel="noopener noreferrer"
href="https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D0%BB%D1%96%D0%B7
%D0%BD%D0%B0_%D0%BB%D0%B0%D1%81%D1%82%D1%96%D0%B2%D0%BA%D0%B0">Дет
альніше</a>
</button>
</div>
<div class="item">
<p>Альбом "Акустика (live)" 2020 рік</p>
<button class="but"><a target="_blank" rel="noopener noreferrer"
href="https://www.youtube.com/watch?v=TfJAUwYVQPs&t=66s">Детальніш
е</a>
</button>
</div>
</div>
</div>
</div>
</div>
</div>
{% endblock %}

```

Лістинг файлу Music.html

```

{% extends 'base.html' %}
{% block title%} Music {% endblock %}
{% block head %}
<link rel="stylesheet" href="{{ url_for('static',
filename='style_m.css') }}">
{% endblock %}
{% block content %}
<div class="intro" style="flex-direction: row">
<div class="container">
<div style="padding-bottom: 40px; text-align: center;">

```

```

<a
href="https://open.spotify.com/artist/4IJcztNtY5L7G3qK4yulqt?si=rx
cHum0mQnO4c0ubQ7dGQg" target="_blank" title="Spotify"></a>
<a href="https://music.apple.com/ua/artist/the-hardkiss/467797614"
target="_blank" title="Apple Music"></a>
<a
href=https://play.google.com/store/music/artist/The_Hardkiss?id=A3
xdoyatwtei5qljfygcrxzjute target="_blank" title="Google Play
Music"></a>
<a href="https://music.youtube.com/channel/UC-
1FEDJKbjpVfsKEl2cyE5w" target="_blank" title="YouTube Music"></a>
<a href="https://www.deezer.com/artist/1432088" target="_blank"
title="Deezer"></a>
</div>
<div style="font-family: 'Stolzl Med'; letter-spacing: 2px; line-
height: 1.4; color: white; max-width: 660px;
text-align: center;
margin: auto;
padding-bottom: 0px;">
<iframe allow="autoplay *; encrypted-media *;" frameborder="0"
style="height: 450px;
width:100%;maxzwidth:660px;overflow:hidden;background:transparent;
"
sandbox="allow-forms allow-popups allow-same-origin allow-scripts
allow-storage-access-by-user-activation allow-top-navigation-by-
user-activation"
src="https://embed.music.apple.com/ua/album/%D0%B0%D0%BA%D1%83%D1%
81%D1%82%D0%B8%D0%BA%D0%B0-live/1520375589?app=music"></iframe>
</div>
</div>
{% endblock %}

```

Лістинг файлу Video.html

```

{% extends 'base.html' %}
{% block title%} Video {% endblock %}
{% block head %}
<link rel="stylesheet" href="{{ url_for('static',
filename='style_v.css') }}">
{% endblock %}

```



```

{% block content %}
<div class="nac">
<div class="video">
<video class="video_media" src="{{ url_for('static',
filename='Video/THE_HARDKISS.mp4') }}" autoplay muted
loop></video>
</div>
</div>
<div class="content" style="margin: auto; position: fixed; top: 0;
left:0">
<div class="parent">
<div class="item">
<iframe src="https://youtube.com/embed/DI6KM_XiLSE"
frameborder="0"></iframe>
</div>
<div class="item">
<iframe src="https://www.youtube.com/embed/YDqQ9W0_BBM"
frameborder="0"></iframe>
</div>
<div class="item">
<iframe src="https://www.youtube.com/embed/ivroIGMAVig"
frameborder="0"></iframe>
</div>
<div class="item">
<iframe src="https://www.youtube.com/embed/Ol5aHdQADVQ"
frameborder="0"></iframe>
</div>
</div>
<div class="parent" style="margin: 10px">
<div class="item"><a class="btn" href="Music.html">
<strong>Дивитися більше </strong></a></div>
</div>
<br>
<br>
</div>
{% endblock %}

```

Лістинг файлу auth.html

```

{% extends 'base.html' %}
{% block title%} Authorization {% endblock %}
{% block head %}
<style>
body {font-family: Arial, Helvetica, sans-serif;
background-color: rgba(35, 35, 59, 1);}
input[type=text], input[type=password] {
width: 100%;
padding: 12px 20px;
margin: 8px 0;
display: inline-block;

```

```

border: 1px solid #ccc;
box-sizing: border-box;
}
input[type=submit]{
background-color: rgb(221, 168, 240);
padding: 14px 20px;
margin: 8px 0;
border: none;
cursor: pointer;
width: 100%;
}
h2, h4{
margin: 0;
}
</style>
{% endblock %}
{% block content %}
<div style="margin: auto; width: 30%; ">
<form method="POST">
<h2 style="text-align: center; color:rgba(194, 165, 170, 1)">Login</h2>
<label for="username"><h4 style="color:rgba(194, 165, 170, 1)">Username: </h4></label>
<input type="text" name="username" id="username">
<label for="password"><h4 style="color:rgba(194, 165, 170, 1)">Password</h4></label>
<input type="password" name="password" id="password">
<input type="submit" value="Login">
</form>
<br>
</div>
{% endblock %}

```

Лістинг файлу register.html

```

{% extends 'base.html' %}
{% block title%} Registration {% endblock %}
{% block head %}
<style>
body {font-family: Arial, Helvetica, sans-serif;
background-color: rgba(35, 35, 59, 1);}
input[type=text], input[type=password] {
width: 100%;
padding: 12px 20px;
margin: 8px 0;
display: inline-block;
border: 1px solid #ccc;
box-sizing: border-box;
}
input[type=submit]{

```

```

background-color: rgb(221, 168, 240);
padding: 14px 20px;
margin: 8px 0;
border: none;
cursor: pointer;
width: 100%;
}
h2, h4{
margin: 0;
}
</style>
{% endblock %}
{% block content %}
<div style="margin: auto; width: 30%; ">
<form method="POST">
<h2 style="text-align: center;color:rgba(194, 165, 170,
1)">Registration</h2>
<label for="username"><h4 style="color:rgba(194, 165, 170,
1)">Username: </h4></label>
<input type="text" name="username" id="username">
<label for="password"><h4 style="color:rgba(194, 165, 170,
1)">Password</h4></label>
<input type="password" name="password" id="password">
<input type="submit" value="Registration">
</form>
<br>
</div>
{% endblo

```

Лістинг файлу sender.html

```

{% extends 'base.html' %}
{% block title%} Mail sender {% endblock %}
{% block head %}
<style>
body {font-family: Arial, Helvetica, sans-serif;
background-color: rgba(35, 35, 59, 1);}
input[type=text], textarea {
width: 100%;
padding: 12px 20px;
margin: 8px 0;
display: inline-block;
border: 1px solid #ccc;
box-sizing: border-box;
}
input[type=submit]{
background-color: rgb(221, 168, 240);
padding: 14px 20px;
margin: 8px 0;
border: none;

```

```

cursor: pointer;
width: 100%;
}
</style>
{% endblock %}
{% block content %}
<div style="margin: auto; width: 30%; ">
<form method="POST">
<h2 style="text-align: center; color:rgba(194, 165, 170, 1)">Mail
sender</h2>
<label for="subject"><h4 style="color:rgba(194, 165, 170,
1)">Subject: </h4></label>
<input type="text" name="subject" id="subject">
<label for="msg"><h4 style="color:rgba(194, 165, 170, 1)">Message
body:</h4></label>
<textarea name="msg" id="msg" cols="30" rows="10"></textarea>
<input type="submit" value="Send">
</form>
<br>
</div>
{% endblock %}

```

Лістинг файлу base.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <title>{% block title %}{% endblock %} | The Hardkiss</title>

  <script
src="https://kit.fontawesome.com/8f3c004287.js"></script>
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link
href="https://fonts.googleapis.com/css2?family=Lato&display=swap"
rel="stylesheet">
  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link
href="https://fonts.googleapis.com/css2?family=Open+Sans&display=s
wap" rel="stylesheet">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min
.js"></script>
  <style>
    body {

```

```

        font-family: 'Lato', sans-serif;
    }
    .overlay {
        height: 100%;
        width: 0;
        position: fixed;
        z-index: 1;
        top: 0;
        left: 0;
        background-color: rgb(0,0,0);
        background-color: rgba(11, 13, 26, 1);
        overflow-x: hidden;
        transition: 0.5s;
    }
    .overlay-content {
        position: relative;
        top: 10%;
        width: 100%;
        text-align: center;
        margin-top: 30px;
    }
    .overlay a {
        padding: 8px;
        text-decoration: none;
        font-size: 36px;
        color: rgba(170, 129, 147, 1);
        display: block;
        transition: 0.3s;
    }
    .overlay a:hover, .overlay a:focus {
        color: rgba(209, 93, 145, 1);
    }
    .overlay .closebtn {
        position: absolute;
        top: 20px;
        right: 45px;
        font-size: 60px;
    }
    @media screen and (max-height: 450px) {
        .overlay a {font-size: 20px}
        .overlay .closebtn {
            font-size: 40px;
            top: 15px;
            right: 35px;
        }
    }
    nav {
        position: fixed;
        top: 40px;
        right: 50px;
        height: 20px;
    }

```

```

        max-width: 150px;
        z-index: 5;
        color: #9B8690;
        font-size: 30px;
    }
    .back-to-top {
        position: fixed;
        bottom: 25px;
        right: 25px;
        display: none;
        border-radius: 50%;
        background-color: rgba(221, 168, 240, 0.7);
        width: 40px;
        height: 40px;
        text-align: center;
    }
    .back-to-top:hover {
        background-color: rgba(178, 59, 222, 0.6);
        transform: scale(1.2, 1.2);
        transition: transform .3s;
    }
    .back-to-top i {
        margin: 10px;
        text-align: left;
    }
    /* The alert message box */
    .alert {
        padding: 20px;
        background-color: #f44336; /* Red */
        color: white;
        text-align: center;
        font-size: 20px;
    }
    /* The close button */
    .closebtn {
        margin-left: 15px;
        color: white;
        font-weight: bold;
        float: right;
        font-size: 22px;
        line-height: 20px;
        cursor: pointer;
        transition: 0.3s;
    }
</style>
{% block head %}{% endblock %}
</head>
<body>
<script>
$(document).ready(function() {
    $(window).scroll(function () {
        if ($(this).scrollTop() > 50) {
            $('#back-to-top').fadeIn();

```

```

        } else {
            $('#back-to-top').fadeOut();
        }
    });
    // scroll body to 0px on click
    $('#back-to-top').click(function () {
        $('body,html').animate({
            scrollTop: 0
        }, 400);
        return false;
    });
});
</script>
<div id="myNav" class="overlay">
    <a href="javascript:void(0)" class="closebtn"
onclick="closeNav()">&times;</a>
    <div class="overlay-content">
        <a href="{{ url_for('index') }}">Головна</a>
        <a href="{{ url_for('group') }}">Про гурт</a>
        <a href="{{ url_for('album') }}">Альбоми</a>
    <a href="{{ url_for('music') }}">Музика</a>
        <a href="{{ url_for('video') }}">Відео</a>
        <br>

        {% if not g.user %}
        <a href="{{ url_for('auth.login') }}">Sign up</a>
        {% else %}

        <a href="{{ url_for('sender') }}">Розсилка</a>
        <a href="{{ url_for('auth.logout') }}">Sign out</a>
        {% endif %}
        <br>
        <div style="display: flex; justify-content: center;">
            <a
href="https://invite.viber.com/?g2=AQAqJkbtwL22f0q7sX%2FgeuWLx1ggzs90EJKoi%2Bd3gWzy04jsc%2Fbf07PqdOcc1Ejv/" target="_blank"></a>
            <a href="https://instagram.com/the_hardkiss"
target="_blank"></a>
            <a href="https://youtube.com/thehardkiss"
target="_blank"></a>
            <a href="https://facebook.com/thehardkiss"
target="_blank"></a>
            <a href="https://t.me/the_hardkiss"
target="_blank"></a>
            <a href="https://twitter.com/the_hardkiss"
target="_blank"></a>
        </div>

```

```

        </div>
    </div>
    <nav>
        <span id='openMenu' style="font-size:50px;cursor:pointer"
onclick="openNav()" ">&#9776; </span>
    </nav>
    <section>
        <!-- flash messages -->
        {% for error_message in
get_flashed_messages(category_filter=["error"]) %}
            <div class="alert alert-danger alert-dismissible text-center">
                <span class="closebtn"
onclick="this.parentElement.style.display='none';">&times;</span>
                {{ error_message }}
            </div>
        {% endfor %}
        {% for success_message in
get_flashed_messages(category_filter=["success"]) %}
            <div class="alert alert-success alert-dismissible text-
center">
                <span class="closebtn"
onclick="this.parentElement.style.display='none';">&times;</span>
            </div>
        {% endfor %}

        {% for info_message in
get_flashed_messages(category_filter=["info"]) %}
            <div class="alert alert-info alert-dismissible text-center">
                <span class="closebtn"
onclick="this.parentElement.style.display='none';">&times;</span>
                {{ info_message }}
            </div>
        {% endfor %}

        {% block content %}{% endblock %}
    </section>
    <script>
function openNav() {
    document.getElementById("myNav").style.width = "100%";
    document.getElementById("openMenu").style.display = "none";
}
function closeNav() {
    document.getElementById("myNav").style.width = "0%";
    document.getElementById("openMenu").style.display = "block";
}
    </script>
    <a id="back-to-top" href="#" class="back-to-top" role="button"><i
class="fas fa-chevron-up"></i></a>
</body>
</html>

```



```

body{
    margin: 0;
    font-family: 'Montserrat', sans-serif;
    font-size: 15px;
    color: #333;
}
h2 {
    margin: 0;
}
.container {
    width: 100%;
max-width: 1200px;
    margin: 0;
}
.intro {
    width: 100%;
    height: 100vh;
    background:
        url("photo/hard0.jpg") center
        no-repeat;
background-size: cover;
display: flex;
flex-direction: column;
justify-content: center;
}

.intro_inner {
    margin: 0;
    width: 100%;
    max-width: 900px;
    text-align: center;
}

.intro_title {
    color: #C2A5AA;
    font-size: 50px;
    font-weight: 100;
    text-transform: uppercase;
    text-align: center;
}

.intro_title:after{
    content: "";
    display: block;
    width: 60px;
    height: 3px;
    background-color: #000000;
    margin: 20px auto ;
}

.btn {

```

```

display: inline-block;
vertical-align: top;
padding: 20px 57px;
border: 3px solid #000000;
font-size: 14px;
color: #000000;
text-transform: uppercase;
text-decoration: none;
border-width: 2px 2px 2px 2px;
background-color: #9B8690;
}
.btn1 {

display: inline-block;
vertical-align: top;
padding: 20px 30px;
border: 3px solid #000000;
font-size: 14px;
color: #000000;
text-transform: uppercase;
text-decoration: none;
border-width: 2px 2px 2px 2px;
background-color: #9B8690;
}
.section {
padding: 80px 0;
background: url("photo/hard1.jpg") center no-repeat;
width: 100%;
height: 80vh;
margin: 0;
background-size: cover;
display: flex;
flex-direction: column;
justify-content: center;
}
.section_header {
width: 100%;
max-width: 950;
margin: 0 auto;
text-align: center;
}
.section_text {
padding: 100px 575px;
width: 60%;
font-size: 25px;
color: rgba(35, 35, 59, 1);
text-align: justify;
font-family: COMIC SANS MS;
}

.section_text:before{
content: "";

```

```

        display: block;
        width: 100px;
        height: 3px;
        background-color: #000000;
        margin: 20px auto ;
    }

```

Лістинг файлу style_m.css

```

body{
    margin: 0;
    font-family: 'Montserrat', sans-serif;
    font-size: 15px;
    color: #333;
}
h1 {
    margin: 0;
}
*,
*: before,
*: after {
    box-sizing: border-box;
}
.container {
    width: 100%;
    max-width: 1200px;
    margin: 0;
}
.intro {
    width: 100%;
    height: 100vh;
    background:
        url("photo/hard3.jpg") center
        no-repeat;
background-size: cover;
display: flex;
flex-direction: column;
justify-content: center;
}
.intro_inner {
    margin: 0;
    width: 100%;
    max-width: 900px;
    text-align: center;
}
.intro_title {
    color: #C2A5AA;
    font-size: 50px;
    font-weight: 100;
    text-transform: uppercase;
}

```

```

        text-align: center;
    }
.intro_title:after{
    content: "";
    display: block;
    width: 60px;
    height: 3px;
background-color: #000000;
    margin: 20px auto ;
}
.btn {
    display: inline-block;
    vertical-align: top;
    padding: 8px 57px;
    border: 3px solid #000000;
    font-size: 14px;
    color: #000000;
    text-transform: uppercase;
    text-decoration: none;
    border-width: 2px 2px 2px 2px;
    background-color: #9B8690;
}
.btn1 {
    display: inline-block;
    vertical-align: top;
    padding: 8px 30px;
    border: 3px solid #000000;
    font-size: 14px;
    color: #000000;
    text-transform: uppercase;
    text-decoration: none;
    border-width: 2px 2px 2px 2px;
    background-color: #9B8690;
}

.section {
    padding: 80px 0;
    background:
        url("photo/hard1.jpg") center
        no-repeat;
    width: 100%;
    height: 80vh;
    margin: 0;
    background-size: cover;
    display: flex;
    flex-direction: column;
    justify-content: center;
}
.section_header {
    width: 100%;
    max-width: 950;
    margin: 0 auto;
    text-align: center;
}

```

```

        .section_text {
            padding: 100px 575px;
            width: 60%;
            font-size: 25px;
            color: #000000;
            text-align: justify;
font-family: COMIC SANS MS;
        }
        .section_text:before{
            content: "";
            display: block;
            width: 100px;
            height: 3px;
            background-color: #000000;
            margin: 20px auto ;
        }
    }

```

Лістинг файлу style_v.css

```

body{
    margin: 0;
    padding: 0;
}

    .nac {
        max-height: 100vh;
        overflow: hidden;
        display: flex;
    }
    .video {
        padding-bottom: 56.25%;
    }
    .video_media {
        position: fixed;
        right: 0;
        bottom: 0;
        min-width: 100%;
        min-height: 100%;
        top: 0;
        left: 0;
        z-index: -10;
    }
    .btn {
        display: inline-block;
        padding: 20px 57px;
        border: 3px solid #000000;
        font-size: 14px;
        color: #000000;
        text-transform: uppercase;
        text-decoration: none;
        border-width: 2px 2px 2px 2px;
    }

```

```
        background-color: #9B8690;
    }
    .parent {
        width: 100%;
        display: flex;
        flex-wrap: wrap;
        align-items: center;
        justify-content: center;
        margin: 50px 30px;
    }
    .item {
        min-width: calc(100%/5);
        padding: 20px;
        text-align:center;
    }
        .item iframe {
            height: 200px;
            width: 350px;
            padding:10px;
            margin: 30px;
        }
```

Лістинг файлу main.py

```

from flask import Flask, render_template, request, flash, url_for,
redirect
    from Hardkiss import auth
    from Hardkiss.db import db, init_app, init_db
    from Hardkiss.models import Subscribers, User
    from flask_mail import Mail, Message
    import re
    app = Flask(__name__)
    app.register_blueprint(auth.bp)
    app.config.from_pyfile("server_conf.cfg")
    app.config['SQLALCHEMY_DATABASE_URI'] =
f"mysql+pymysql://{app.config['USERNAME']}:{app.config['PASSWORD']}
}" \
f"@{app.config['HOST']}:3306/{app.config['DB_NAME']}"
    init_app(app)
    db.init_app(app)
    with app.app_context():
        if app.config['INIT_DB']:
            init_db()
    app.jinja_env.add_extension('jinja2.ext.loopcontrols')
    mail = Mail(app)
    @app.route('/', methods=['POST', 'GET'])
    def index():
        if request.method == 'POST':
            sub_mail = request.form['mail']
            if re.match(r"(?:[a-z0-9!#$%&'*\+\/=?^_`{|}~--
]+(?:\.(?:[a-z0-9!#$%&'*\+\/=?^_`{|}~--]+)*|\"
r"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]|\\[\x01-
\x09\x0b\x0c\x0e-\x7f])*\"")" r"@(?:((?:[a-z0-9](?:[a-z0-9-]*[a-z0-
9])?)\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?)|"
r"\[(?:((?:2(5[0-5]|[0-4][0-9])|1[0-9][0-9]|1[0-9]?[0-
9]))\.){3}((?:2(5[0-5]|[0-4][0-9])|1[0-9][0-9]|"
r"[1-9]?[0-9])|[a-z0-9-]*[a-z0-9]:(?:[\x01-\x08\x0b\x0c\x0e-
\x1f\x21-\x5a\x53-\x7f]"
r"|\\[\x01-\x09\x0b\x0c\x0e-\x7f])+)\\])", sub_mail):
        new_user = Subscribers(mail=sub_mail, valid=1)
        db.session.add(new_user)
        db.session.commit()
        else:
            flash("Don't support this e-mail address",
'error')
        return render_template('index.html')
    @app.route('/group')
    def group():
        return render_template('Group.html')
    @app.route('/music')

```

```

def music():
    return render_template('Music.html')
@app.route('/video')
def video():
    return render_template('Video.html')
@app.route('/album')
def album():
    return render_template('Album.html')
@auth.login_required
@app.route('/sender', methods=['POST', 'GET'])
def sender():
    if request.method == 'POST':
        users = [sub.mail for sub in
Subscribers.query.filter_by(valid=1).all()]
        with mail.connect() as conn:
            for user in users:
                msg = Message(recipients=[user],
                    body=request.form['msg'],
subject=request.form['subject'],
sender=app.config['MAIL_USERNAME'])
                conn.send(msg)
            return redirect(url_for('index'))
        return render_template('sender.html')
if __name__ == '__main__': app.run()

```

Лістинг файлу db.py

```

import click
from flask_sqlalchemy import SQLAlchemy
from flask.cli import with_appcontext
from werkzeug.security import generate_password_hash
db = SQLAlchemy()
def init_db():
    from .models import User
    db.drop_all()
    db.create_all()
    admin = User(name="admin",
password=generate_password_hash("1234qwer"), valid=1)
    db.session.add(admin)
    db.session.commit()
@click.command("init-db")
@with_appcontext
def init_db_command():
    init_db()
    click.echo("Initialized the database.")
def init_app(app):
    app.cli.add_command(init_db_command)

```


Лістинг файлу models.py

```

from .db import db
class User(db.Model):
    __tablename__ = "users"
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(80), nullable=False)
    password = db.Column(db.String(100), nullable=False)
    valid = db.Column(db.Boolean, nullable=False, default=0)
    def __repr__(self):
        return f"<User {self.id} - {self.name}>"
    def __getitem__(self, item):
        return getattr(self, item)
class Subscribers(db.Model):
    __tablename__ = "subscribers"
    id = db.Column(db.Integer, primary_key=True)
    mail = db.Column(db.String(100), nullable=False)
    valid = db.Column(db.Boolean, nullable=False, default=0)
    def __repr__(self):
        return f"<Subscriber {self.id} - {self.mail}>"
    def __getitem__(self, item):
        return getattr(self, item)

```

Лістинг файлу auth.py

```

import functools
from flask import (
    Blueprint, flash, g, redirect, render_template, request,
    session, url_for
)
from werkzeug.security import check_password_hash,
generate_password_hash
from .db import db
from . import models
bp = Blueprint('auth', __name__, url_prefix='/auth')
@bp.route('/login', methods=('GET', 'POST'))
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        error = None
        user =
models.User.query.filter_by(name=username).first()
        if user is None:
            error = 'Incorrect username.'
        elif not check_password_hash(user.password, password):
            error = 'Incorrect password.'
        elif user.valid == 0:
            error = "Account not valid yet."

```

```

        if error is None:
            session.clear()
            session['user_id'] = user['id']
            return redirect(url_for('index'))
        flash(error, "error")
    return render_template('auth.html')
@bp.route('/logout')
def logout():
    session.clear()
    return redirect(url_for('index'))
def login_required(view):
    @functools.wraps(view)
    def wrapped_view(**kwargs):
        if g.user is None:
            return redirect(url_for("index"))
        return view(**kwargs)
    return wrapped_view
@bp.before_app_request
def load_logged_in_user():
    user_id = session.get("user_id")
    if user_id is None:
        g.user = None
    else:
        g.user = (
            models.User.query.get(user_id)
        )

```