

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет радіофізики, електроніки та комп'ютерних систем

Кафедра комп'ютерної інженерії

Створення веб-сайту з використанням статичної генерації

Кваліфікаційна робота бакалавра
студента 4 року навчання

Спеціальність: 123 «Комп'ютерна
інженерія»

ОНП «Комп'ютерна системи та
мережі»

Богдана ЧОРНОПОЛЬСЬКОГО



Науковий керівник:



Юрій ЮРЧИК _____,

асистент кафедри комп'ютерної
інженерії

Рецензент:

доцент кафедри медичної
радіофізики, к.ф.-м.н., доц.

Олександр СУДАКОВ _____,

До захисту допускаю:

Завідувач кафедрою

Юрій БОЙКО

Ухвалено на засіданні кафедри "16" червня 2022 р., протокол № 18

Київ - 2022

РЕФЕРАТ

Пояснювальна записка: 39 с., 40 рис., 17 джерел, 1 додаток.

Мета роботи – Дослідження методів розробки веб-сайтів з використанням статичної генерації. Розробка веб-сайту Інформаційно-обчислювального центру.

В ході дипломної роботи було досліджено та порівняно різні генератори статичних сайтів. Створено веб-сайт для Інформаційно-обчислювального центру з використанням генератору статичних сайтів Nikola, мови гіпертекстової розмітки HTML, фреймворків таких як Tailwind CSS для стилів та AlpineJS, GSAP для анімацій. Також було оптимізовано створений за допомогою інструменту Google Lighthouse.

Зміст

1. Створення веб-сайтів методами статичної генерації	6
1.1 Веб-сайт, веб-застосунок та різниця між ними.....	6
1.2 Статичні та динамічні веб-сайти	7
1.3 Генератори статичних веб-сайтів	8
1.4 Вибір генератору статичних веб-сайтів.....	9
1.5 Використання Front end фреймворків при створення веб-сайту.....	11
1.5.1 Переваги та недоліки використання Front end фреймворків для створення веб-сайту	11
1.5.2 Вибір Front end фреймворків які будуть використовуватися при створення сайту ІОЦ.....	13
1.6 Обґрунтування потреби створення нового веб-сайту для ІОЦ.....	14
2 Розробка веб-сайту інформаційно-обчислювального центру з використанням Nikola	15
2.1 Завдання і цілі сайту	15
2.2 Загальна структура сайту	16
2.3 Створення та структура проекту Nikola	17
2.4 Створення головної сторінки.....	23
2.5 Створення сторінки новин	25
2.6 Створення інших сторінок	27
Додатку А.....	29
3. Удосконалення та оптимізація створеного сайту	30
3.1 Перевірка веб-сайту на проблеми та помилки.....	30
3.2 SEO оптимізація веб-сайту.....	32
ВИСНОВОК.....	35
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	36
Додаток А.....	38

ПОЗНАЧЕННЯ ТА СКОРОЧЕННЯ

HTML (Hypertext Markup Language) - Мова гіпертекстової розмітки CSS

CSS (Cascading Style Sheets) - Каскадні таблиці стилів

WWW (World Wide Web) - Всесвітня павутина

Вступ

У сучасному світі кількість веб-ресурсів зростає з кожним днем. Вже майже всі мають представництво в Інтернеті, починаючи з персональних блогів до сайтів державних установ, комерційних організацій, шкіл, університетів та їх підрозділів. Зараз щоб ефективно та оперативного надавати необхідну інформацію всім хто її потребує, вищеперелічені організації зіштовхуються з проблемою створення свого власного інтернет ресурсу. Саме це є одним з найкращих способів розповсюдження корисних даних для клієнтів, працівників, користувачів тощо.

У цій роботі було розглянуто що таке веб-сайт, які вони бувають та що потрібно знати та вміти для його створення. Також було досліджено та порівняно різні генератори статичних сайтів.

Метою даної дипломної роботи - є створення веб-сайту для Інформаційно-обчислювального центру та автоматизація генерації його вмісту.

Цей ресурс буде створений з використанням мови розмітки HTML, каскадних таблиць стилів CSS, мов програмування JavaScript та Python, на базі генератора статичних сайтів Nikola.

1. Створення веб-сайтів методами статичної генерації

1.1 Веб-сайт, веб-застосунок та різниця між ними

Сайт - це сукупність веб-сторінок, які пов'язані між собою і містять контент різного формату: текст, картинки, відео, музика та інші. Він може бути наповненим різним контентом, наприклад текстом, картинками та різними медіа файлами та складатися з однієї або декількох сторінок.

Приклади таких веб-сайтів: Wikipedia, Google, Yahoo та інші.

Сайти бувають:

- для знайомств та спілкування з іншими людьми;
- особисті, професійні чи ділові блоги;
- спільноти та співтовариства;
- пошукові системи;
- інтернет магазини;
- сайти-портфоліо та інші.

Для чого створюють сайти:

- для демонстрації та продажу своєї продукції;
- передачі інформації про організацію, заклад, персону тощо;
- реклами, розвитку та просування бізнесу;
- показати свої знання, навички та досягнення;
- для взаємодії з клієнтами.

Веб-додаток – це повноцінна програма, доступ до якої здійснюється через браузер. Іншими словами, це той самий сайт, тільки з інтерактивними елементами та великим функціоналом[3].

Приклади таких веб-застосунків: YouTube, Facebook, Moodle та ін.

Чим відрізняється веб-сайт від веб-застосунку:

- Як відбувається взаємодія з клієнтами - На сайтах користувачі можуть лише переглядати контент та інформацію. А от у веб-додатку, крім споживання, користувачі також можуть змінювати певні дані, наприклад, додавати коментарі до постів тощо.
- Автентифікація - не є обов'язковою для веб-сайтів. А от у веб-додатках без неї не обійтися так як вони мають більший рівень взаємодії з користувачем.
- Рівень складності розв'язуваних задач - крім відображення даних, повинен обробляти дії користувачів, можливо працювати з базою даних та інше.
- Рівень взаємодії з користувачем - на сайтах усі дані статичні та найчастіше відкриті для всіх відвідувачів. Взаємодія веб-програми з користувачами має більш складний характер і неможлива повною мірою без автентифікації.

1.2 Статичні та динамічні веб-сайти

Статичні сайти - складається з HTML-сторінок, документів і медіа, які сервер може прочитати з постійного сховища та надати клієнту (зазвичай веб-браузеру) без подальших налаштувань. Статичний веб-сайт може включати HTML, CSS, мультимедійний вміст і, можливо, клієнтські сценарії/програми, написані на Javascript або інших мовах програмування, які може запускати браузер. Ці сайти зазвичай орієнтовані на вміст (лише читання), але може бути інтерактивним, як гра чи інструмент[1].

Динамічні сайти Веб-сайти, які зберігаються у форматі, який не відображається безпосередньо веб-браузером, називаються динамічними сайтами. Сервер приймає клієнтський запит і генерує відповідь, яку браузер може відтворити після виконання деяких дій між ними. Тим часом операції визначаються мовою сценаріїв або програмування (деякі з найпоширеніших серед них — PHP, Python, Perl, Ruby, Java, C#, JavaScript через Node.js) і часто включають запити до бази

даних. Сайт може бути природно динамічним, забезпечуючи інтерактивну функціональність клієнт-сервер, або може також просто конвертувати статично збережену інформацію на сторінку HTML[1].

1.3 Генератори статичних веб-сайтів

Генератор статичних сайтів — це інструмент, який генерує повний статичний веб-сайт HTML на основі необроблених даних і набору шаблонів. По суті, генератор статичних сайтів автоматизує завдання кодування окремих HTML-сторінок і завчасно готує ці сторінки для показу користувачам. Оскільки ці сторінки HTML попередньо створені, вони можуть дуже швидко завантажуватися у браузерях користувачів.

Генератори статичних сайтів є альтернативою системам керування вмістом (CMS) — іншого типу інструментів для керування веб-контентом, створення веб-сторінок та впровадження шаблонів. (Шаблон — це формат для багаторазового використання веб-контенту; розробники використовують шаблони, щоб уникнути повторного написання одного і того ж форматування.)

Переваги та недоліки використання генератора статичних сайтів:

Переваги:

- **Продуктивність:** оскільки генератори статичних сайтів створюють веб-сторінки заздалегідь, а не на вимогу клієнта, веб-сторінки завантажуються дещо швидше у браузерях користувачів.
- **Налаштування:** розробники можуть створити будь-який шаблон, який забажають. Вони не обмежені полями, наданими CMS, ані вбудованими шаблонами CMS.

- Простота в роботі з сервером: статичні веб-сайти займають малу кількість місця та не потребують стільки коду для роботи на стороні сервера.

Недоліки:

- Немає зручного інтерфейсу: користувачам, які не є розробниками, важче публікувати вміст за допомогою генератора статичних сайтів. Немає інтерфейсу CMS, і робота з необробленими неформатованими даними може налякати користувачів. Крім того, для оновлення веб-сайту часто потрібна підтримка розробників.
- Мало або зовсім немає попередньо створених шаблонів: Недоліком необмеженого налаштування є те, що для початку може знадобитися більше часу. Багато генераторів статичних сайтів не мають шаблонів, і спочатку розробникам доведеться витратити багато часу на їх створення з нуля.

1.4 Вибір генератора статичних веб-сайтів

Таблиця порівняння популярних генераторів статичних сайтів

Назва	Має відкритий код	Мова програмування	Шаблони	Формати	Безкоштовний
NextJS	+	JavaScript	React	Markdown	+
Hugo	+	Go	Go	Markdown, HTML	+
Jekyll	+	Ruby	Liquid	Markdown, HTML	+
Nikola	+	Python	Jinja2,	reStructure	+

			Mako	dText, Markdown, IPython (Jupyter) та HTML	
Pelican	+	Python	jinja2	reStructure dText, Markdown	+
Eleventy	+	JavaScript	Liquid, Nunjucks, Mustache, Haml, Pug, HTML	reStructure dText, Markdown, DocBook, LaTeX	+

Для створення сайту ІОЦ було обрано генератор Nikola.

Переваги Nikola над іншими генераторами статичних веб-сайтів:

- Підтримка більшості форматів вхідних даних, таких як reStructuredText, Markdown, IPython (Jupyter) та HTML, а також він має плагіни для інших форматів.
- Постачається з усім необхідним для створення сучасного веб-сайту: блогами (з коментарями, тегами, категоріями, архівами, RSS/Atom-каналами), багатомовною підтримкою, вбудованими засобами для створення галереї зображень та іншим.
- Швидкість – оновлення контенту відбувається лише на тих сторінках, які потребують перебудови, заощаджуючи час процесора та пропускну здатність завантаження.
- Не потребує великих ресурсів серверу на якому розміщений
- Nikola має дружній користувальницький інтерфейс, який спрощує роботу над сайтом.

- Nikola дуже легко розширюється. Ви можете написати плагін, щоб додати будь-яку функцію використовуючи мову програмування Python, або написати власну тему для свого веб-сайту.

1.5 Використання Front end фреймворків при створення веб-сайту

1.5.1 Переваги та недоліки використання Front end фреймворків для створення веб-сайту

Переваги:

- Більш надійний. Стандартизований, попередньо створений та протестований, фронт-енд фреймворк надає широко використовувану та часто оновлювану кодову базу. Як результат, користувач може бути впевненими, що він буде без помилок і буде широко підтримуватися різними браузерами/пристроями. Також користувачі отримують переваги документації та підтримки від великої спільноти
- Швидка розробка інтерфейсу, швидші прототипи: поряд зі стандартизованим, ефективнішим і надійним рішенням, вони зменшують кількість коду, який потрібно писати, виконують багато важкої роботи та є більш гнучкими, ніж код, написаний для вирішення однієї проблеми для конкретної програми
- Послідовний, стандартизований підхід до інтерфейсу. Фронт-енд фреймворки та бібліотеки заохочують стандартний підхід до проекту, дозволяючи декільком учасникам менше турбуватися про інтерпретацію користувацького коду та більше про дотримання стандартів фреймворку.

- Кращий користувацький досвід: бібліотеки/фреймворки інтерфейсу можуть забезпечити можливість плавної анімації/переходів між змінами сторінок. Крім того, бібліотеки та фреймворки часто оновлюються та вдосконалюються завдяки великій спільноті розробників, які їх використовують, створюючи надзвичайно продуктивний, надійний та стислий код. Це може підвищити продуктивність веб-сайту та забезпечити чистіший код який легко читається.

Недоліки:

- Надто складний для простих проектів. Фронт-енд фреймворки/бібліотеки можуть додати непотрібного розміру на просту цільову сторінку або проекти мікросайтів, які часто не отримують переваги від ефективності, яку вони забезпечують.
- Важко інтегрувати в застарілі інтерфейси. Хоча переваги можуть бути більшими ніж проблема інтеграції фреймворка або бібліотеки в існуючий проект, інтеграція часто може бути складним процесом, особливо для проектів, створених із високоспеціалізованим, застарілим або налаштованим кодом.
- Оновлення можуть спричиняти помилки: оновлення іноді можуть вносити критичні зміни та посилювати потребу в більш ретельному процесі QA, який може перевантажити менші команди без відповідних ресурсів QA.
- Навчання: хоча однією з характерних переваг є стандартизація, вибір фреймворка або бібліотеки вимагає зобов'язань підтримувати та розвивати певний набір навичок. Залежно від ресурсів, використання фреймворку або бібліотеки може створити прогалини в знаннях, технічну заборгованість і збільшити терміни розробки.

1.5.2 Вибір Front end фреймворків які будуть використовуватися при створення сайту ІОЦ

Для створення сайту ІОЦ було обрано такі Front end фреймворки:

- Tailwind CSS
- Alpine JS
- GSAP

Tailwind CSS - це високо налаштований низькорівневий фреймворк CSS, який дає користувачам усі будівельні блоки, необхідні для створення індивідуальних дизайнів без будь-яких наперед визначених стилів, з які потрібно перевизначати[13].

Переваги використання Tailwind CSS:

- Контроль стилів. Tailwind — це унікальний фреймворк CSS, коли справа доходить до створення стилю веб-додатків, що означає, що Tailwind не має теми за замовчуванням, яку потрібно використовувати, як інші фреймворки CSS. Наприклад, розробник може надати кожному проекту різний вигляд, навіть якщо він використовує ті самі елементи (кольорову палітру, розмір тощо).
- Швидкий процес стилізації CSS. Tailwind є дуже швидким фреймворком. За допомогою нього можна легко створювати гарні макети, безпосередньо стилізувати HTML елементи. Це можливо, тому що Tailwind пропонує тисячі вбудованих класів, які не вимагають створювати нічого з нуля. Ці CSS класи і є основною причиною, чому створення та оформлення за допомогою Tailwind відбувається так швидко.
- Адаптивність та безпека. Tailwind має вбудовані класи для адаптивної верстки сайтів. Ці класи дозволяють створювати різні

дизайни для пристроїв різного розміру. Крім того, Tailwind є стабільним фреймворком з моменту його першого випуску.

- Додаткові можливості. Tailwind дозволяє розширювати набір класів що також значно спрощує процес розробки.

Недоліки використання Tailwind CSS:

- Стили та HTML змішані. Для деяких елементів потрібно використовувати багато стилів, тому в них буде багато класів, що робить HTML документ дещо громістким.
- Навчання. Для того щоб розібратися з усіма функціями та можливостями Tailwind потрібен час. Навіть для досвідчених розробників може бути складно навчитися повністю використовувати попередньо створені класи.

Alpine JS - легкий JavaScript фреймворк, який дозволяє писати скрипти прямо в HTML документі. Він дозволяє швидко і ефективно маніпулювати компонентами сайту. Має ті ж самі недоліки що і Tailwind CSS.

GSAP - (GreenSock Animation Platform) - це потужна бібліотека JavaScript, яка дозволяє розробникам і дизайнерам створювати різноманітні анімації на основі шкали часу (тобто анімації які відбуваються протягом визначеного часу).

1.6 Обґрунтування потреби створення нового веб-сайту для ІОЦ

Недоліки старого сайту які призвели до потреби заміни його на новий:

- застарілий дизайн, як видно на Рис. 1.5.1
- технології які використовувалися при створенні старого сайту застаріли і не дозволяють його оновити належним чином

- проблеми в роботі сайту, наприклад більшість сторінок не відкриваються



Рис. 1.5.1

2 Розробка веб-сайту інформаційно-обчислювального центру з використанням Nikola

2.1 Завдання і цілі сайту

Перед розробкою веб-сайту визначимося з завданнями, які він повинен виконувати:

- Надання загальної інформації про ІОЦ: історію створення, основний вид діяльності, її керівників та працівників;
- Публікація новин ІОЦ;
- Інформація про проекти якими займається ІОЦ;
- Контактні дані, для того, щоб зацікавлені користувачі могли зв'язатися з представниками ІОЦ.

2.2 Загальна структура сайту



Рис. 2.1.1 Загальна структура сайту

Головна сторінка містить коротку інформацію про що таке ІОЦ і чим він займається, декілька основних проектів ІОЦ та посилання на інші, також в шапці сайту розміщені посилання на всі інші сторінки проекту.

“Сторінка “Новини” містить список новин, які відбулися на ІОЦ, починаючи з тих, які були додані останніми.

Сторінка “Історія” містить історію створення ІОЦ та те як він змінювався протягом свого існування.

Сторінка “Штат” містить список керівників та працівників ІОЦ.

Сторінка “Ресурси” містить повний список проектів, якими займається ІОЦ.

Сторінка “Контакти” містить контактні дані за допомогою яких користувачі можуть легко зв’язатися з представниками ІОЦ.

2.3 Створення та структура проекту Nikola

Для того щоб створити проект Nikola, спочатку потрібно встановити його у віртуальне середовище мови Python за допомогою менеджера пакетів `pip`. Після чого в консолі можна прописати команду `'nikola init'` для первинного налаштування та ініціалізації нового проекту (Рис. 2.3.1).

```
(venv) Дипломна робота\forExamples nikola init
Creating Nikola Site
=====

This is Nikola v8.1.3. We will now ask you a few easy questions about your new site.
If you do not want to answer and want to go with the defaults instead, simply restart with the '-q' parameter.
--- Questions about the site ---
Destination: Example
Site title [My Nikola Site]: Example site
Site author [Nikola Tesla]: Author
Site author's e-mail [n.tesla@example.com]: author@example.com
Site description [This is a demo site for Nikola.]: This is a example of site using nikola.
Site URL [https://example.com/]: https://example.com/
Enable pretty URLs (/page/ instead of /page.html) that don't need web server configuration? [Y/n] y
--- Questions about languages and locales ---
We will now ask you to provide the list of languages you want to use.
Please list all the desired languages, comma-separated, using ISO 639-1 codes. The first language will be used as the default.
Type '?' (a question mark, sans quotes) to list available languages.
Language(s) to use [en]: uk

Please choose the correct time zone for your blog. Nikola uses the tz database.
You can find your time zone here:
https://en.wikipedia.org/wiki/List_of_tz_database_time_zones

Time zone [Europe/Kiev]: Europe/Kiev
Current time in Europe/Kiev: 12:59:55
Use this time zone? [Y/n] y
--- Questions about comments ---
You can configure comments now. Type '?' (a question mark, sans quotes) to list available comment systems. If you do not want any comments, just leave the field blank.
Comment system:

That's it, Nikola is now configured. Make sure to edit conf.py to your liking.
If you are looking for themes and addons, check out https://themes.getnikola.com/ and https://plugins.getnikola.com/.
Have fun!
[2021-12-07 13:00:10] INFO: init: Created empty site at Example.
```

Рис. 2.3.1

Після первинної ініціалізації та налаштування отримуємо директорію з файлом конфігурації проекту та папками в які ми вже можемо додавати наш контент (Рис. 2.3.2).

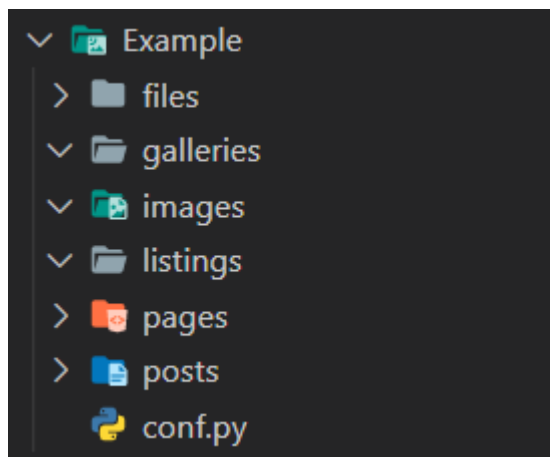


Рис. 2.3.2

Після першого запуску проекту структура файлів набере наступного вигляду (див Рис. 2.3.3).

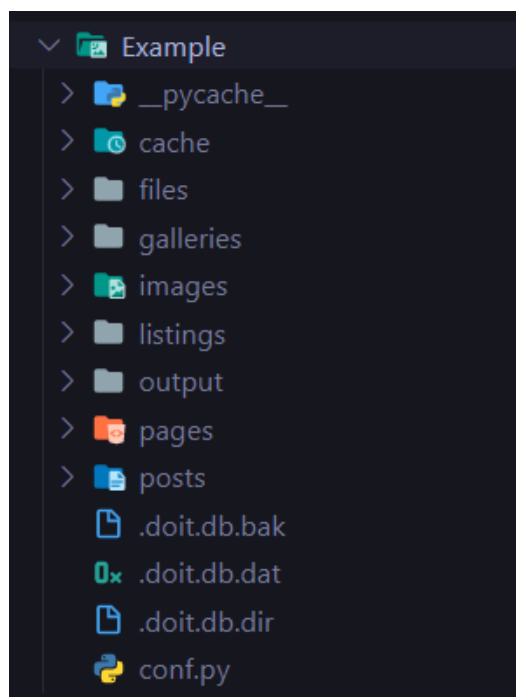


Рис. 2.3.3

З'явилися директорії в яких зберігається кеш, а також директорія “output” в якій зберігаються згенеровані статичні сторінки та файли (див Рис.2.3.4).

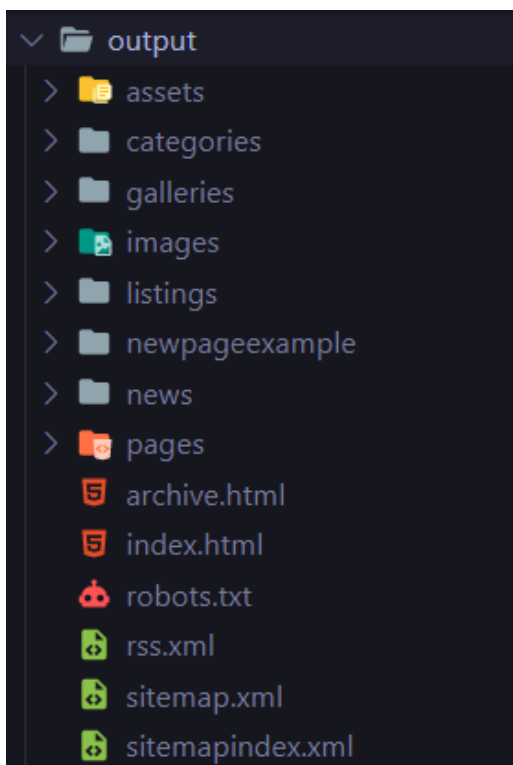


Рис. 2.3.4

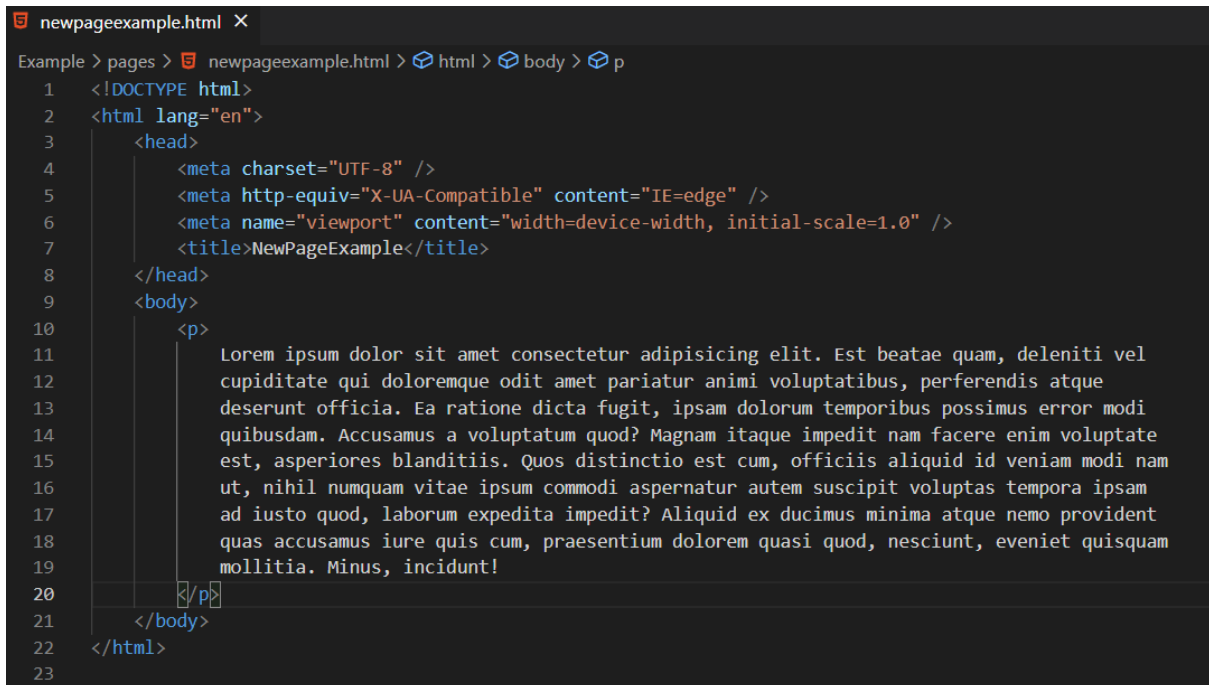
Для того щоб додати нову сторінку до створеного сайту, потрібно в консолі перейти в папку проекту та прописати команду ‘nikola new_page’.

Після виконання команди потрібно вказати Заголовок сторінки і натиснути “Ентер” (Рис. 2.3.5).

```
(venv) D:\Study\4 year\Дипломна робота\forExamples>cd Example
(venv) D:\Study\4 year\Дипломна робота\forExamples\Example>nikola new_page
Creating New Page
-----
Title: NewPageExample
Scanning posts.....done!
[2021-12-07 13:10:32] INFO: new_page: Your page's text is at: pages\newpageexample.rst
```

Рис. 2.3.5

Після чого в папці з назвою “pages”, створюється новий файл (за замовчуванням в форматі .rst). Для зручності формат файлу можна змінити на .html (Рис. 2.3.6).

The image shows a code editor window titled 'newpageexample.html'. The breadcrumb navigation at the top reads 'Example > pages > newpageexample.html > html > body > p'. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <title>NewPageExample</title>
8   </head>
9   <body>
10    <p>
11      Lorem ipsum dolor sit amet consectetur adipisicing elit. Est beatae quam, deleniti vel
12      cupiditate qui doloremque odit amet pariatur animi voluptatibus, perferendis atque
13      deserunt officia. Ea ratione dicta fugit, ipsam dolorum temporibus possimus error modi
14      quibusdam. Accusamus a voluptatum quod? Magnam itaque impedit nam facere enim voluptate
15      est, asperiores blanditiis. Quos distinctio est cum, officiis aliquid id veniam modi nam
16      ut, nihil numquam vitae ipsum commodi aspernatur autem suscipit voluptas tempora ipsam
17      ad iusto quod, laborum expedita impedit? Aliquid ex ducimus minima atque nemo provident
18      quas accusamus iure quis cum, praesentium dolorem quasi quod, nesciunt, eveniet quisquam
19      mollitia. Minus, incidunt!
20    </p>
21  </body>
22 </html>
```

Рис. 2.3.6

Для того щоб згенерувати файли та відкрити проект у браузері потрібно по черзі виконати команди ‘nikola build’ та ‘nikola serve -b’. Як виглядає головна сторінка в щойно згенерованому сайту можна побачити на (Рис. 2.3.7).

Example site

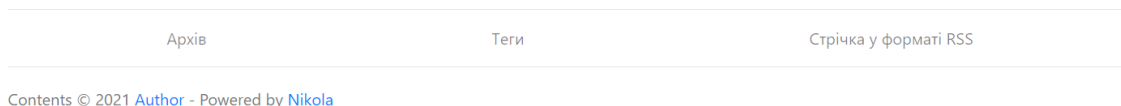


Рис. 2.3.7

Як бачимо на головній сторінці вже є чимало контенту згенерованого за замовчуванням. В адресному рядку добавимо адресу нової сторінки ‘/pages/newpageexample’ (див Рис. 2.3.8).



Рис. 2.3.8

Після того як було підтверджено що проект запускається і працює можна приступати до внесення основних змін. Для початку було внесено зміни в

файл конфігурації - прибрано посилання для навігації, в подальшому туди будуть додані посилання на нові сторінки (Рис. 2.3.9).

```

NAVIGATION_LINKS = {
  DEFAULT_LANG: (
    ("/archive.html", "Архів"),
    ("/categories/", "Теги"),
    ("/rss.xml", "Стрічка у форматі RSS"),
  ),
}

NAVIGATION_LINKS = {
  DEFAULT_LANG: (
    ("/", "Головна"),
  ),
}

```

Рис. 2.3.9

Далі було змінено генерацію нових сторінок. Тепер щоб перейти на сторінку потрібно буде лише вказати її назву без проміжного “/pages/” (Рис. 2.3.10).

```

POSTS = (
  ("posts/*.rst", "posts", "post.tpl"),
  ("posts/*.md", "posts", "post.tpl"),
  ("posts/*.txt", "posts", "post.tpl"),
  ("posts/*.html", "posts", "post.tpl"),
)
PAGES = (
  ("pages/*.rst", "pages", "page.tpl"),
  ("pages/*.md", "pages", "page.tpl"),
  ("pages/*.txt", "pages", "page.tpl"),
  ("pages/*.html", "pages", "page.tpl"),
)

POSTS = (
  ("posts/*.rst", "posts", "post.tpl"),
  ("posts/*.md", "posts", "post.tpl"),
  ("posts/*.txt", "posts", "post.tpl"),
  ("posts/*.html", "posts", "post.tpl"),
)
PAGES = (
  ("pages/*.rst", "", "page.tpl"),
  ("pages/*.md", "", "page.tpl"),
  ("pages/*.txt", "", "page.tpl"),
  ("pages/*.html", "", "page.tpl"),
)

```

Рис. 2.3.10

За замовчуванням створений сайт є блогом, і всі пости з’являються на головній сторінці. Щоб це змінити потрібно в файл конфігурації додати рядок ‘INDEX_PATH = "news"’. Тепер всі пости потраплятимуть на сторінку “/news” (окремо її створювати не треба, вона генерується автоматично). Також було добавлено посилання на сторінку новин в “NAVIGATION_LINKS”. Приклад того як виглядає сайт на даному етапі можна побачити на Рис. 2.3.11.

Example site

Головна

Новини

Contents © 2021 [Author](#) - Powered by [Nikola](#)

Рис. 2.3.11

2.4 Створення головної сторінки

Як було видно в попередньому пункті, головна сторінка вже містить деякий контент. Цей контент надходить з шаблонів, саме з них і потрібно почати редагування. Спочатку їх треба скопіювати в папку проекту (Рис. 2.4.1).

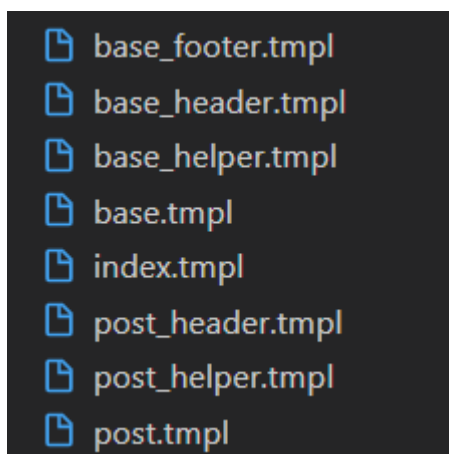


Рис. 2.4.1

В першу чергу було відредаговано “base_helper.tpl”. В цьому файлі задаються елементи, які будуть знаходитись під тегом “head”. В ньому

потрібно додати власні файли CSS та скрипти JavaScript за потребою (Рис. 2.4.2).

```
<link rel="stylesheet" href="/assets/css/custom.css">  
<script defer src="https://unpkg.com/alpinejs@3.x/dist/cdn.min.js"></script>
```

Рис. 2.4.2

Далі було змінено шаблони для “header” та “footer” які знаходяться у файлах “base_header.tpl” та “base_footer.tpl” відповідно. Коли шаблони верхньої і нижньої частини веб-сайту змінені до потрібного вигляду, можна переходити до основної, “main” частини сторінки. Було прийнято рішення прибрати все з цієї частини щоб вона складалася лише з того що міститься в файлі створеної сторінки, таким чином кожна нова сторінка міститиме однакові “header” та “footer”, але різну основну частину. Наступним кроком є додавання інформації та стилів до головної сторінки. Отриману сторінку можна переглянути на Рис. 2.4.3.

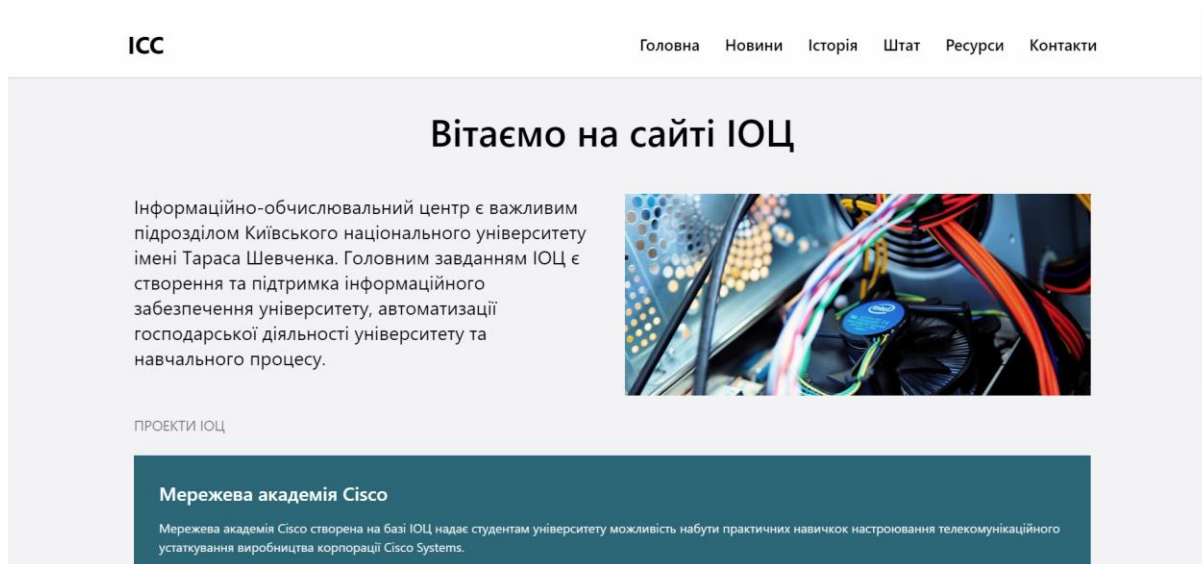


Рис. 2.4.3

2.5 Створення сторінки новин

Сторінка новин як і інші сторінки має “header” та “footer”, але основна її частина не є пустою, а задається за допомогою шаблону з файлу “index.tmpl”. Цей шаблон налаштовується на додавання на сторінку всіх постів з папки “posts”.

Для того щоб створити новий пост/новину потрібно в консоль прописати команду ‘nikola new_post’ та задати заголовок (Рис. 2.5.1).

```
(Nikola) \Дипломна робота\Nikola\ICSS>nikola new_post
Creating New Post
-----
Title: The newest post on the site
Scanning posts.....done!
[2021-12-07 15:03:14] INFO: new_post: Your post's text is at: posts\the-newest-post-on-the-site.rst
```

Рис. 2.5.1

Після чого генерується новий файл в папці “posts”, назвою якого є заданий заголовок (Рис. 2.5.2)

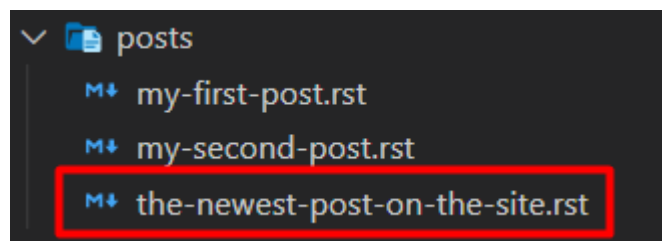


Рис. 2.5.2

Відкриємо файл та додамо в нього випадковий текст (Рис. 2.5.3).

```

the-newest-post-on-the-site.rst X
Nikola > ICC > posts > the-newest-post-on-the-site.rst
1  .. title: The newest post on the site
2  .. slug: the-newest-post-on-the-site
3  .. date: 2021-12-07 15:03:14 UTC+02:00
4  .. tags:
5  .. category:
6  .. link:
7  .. description:
8  .. type: text
9
10 Lorem ipsum dolor sit amet consectetur adipisicing elit. Sed iure nisi voluptates neque ut praesentium numquam
11 laudantium voluptatum impedit aut, quidem qui quos quibusdam quod repellat mollitia. Enim tempora totam sunt
12 asperiores, perspiciatis non modi, saepe molestiae, fugit quisquam ab a vitae voluptatem. Eaque fugit necessitatibus
13 assumenda a beatae at.
14

```

Рис. 2.5.3

Після того як було створено декілька постів можна приступати до зміни шаблону та додавання стилів до цієї сторінки. Після обробки вдалося отримати таку сторінку(див Рис. 2.5.3)

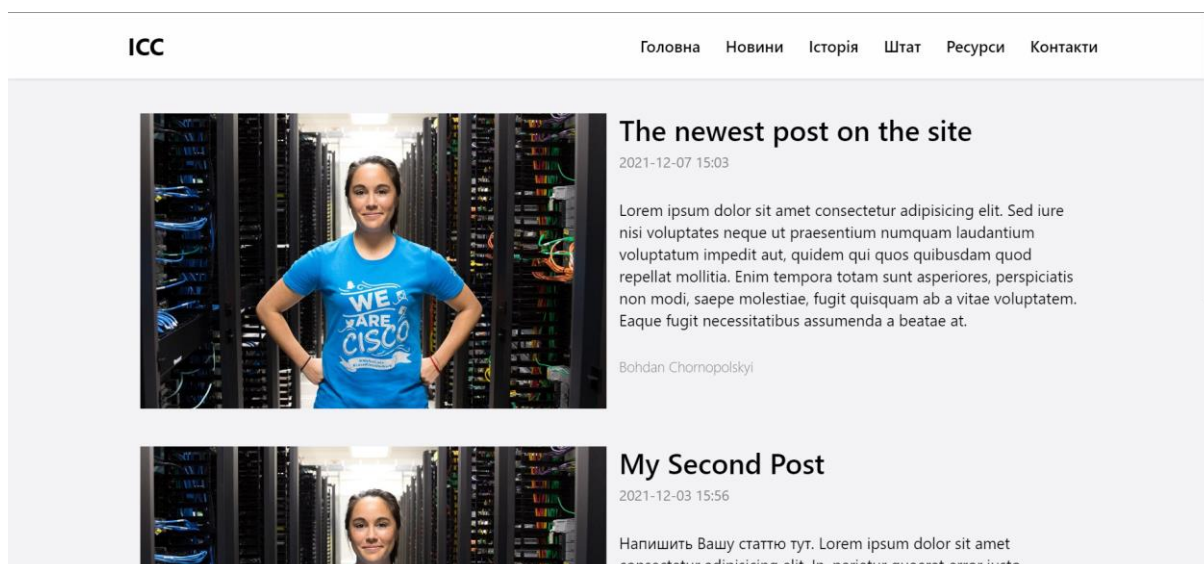


Рис. 2.5.3

Як бачимо кожна новина складається з заголовку, дати створення, основної частини та автора. Якщо натиснути на заголовок, відкриється сторінка, яка містить лише пост на який натиснули (Рис. 2.5.4)

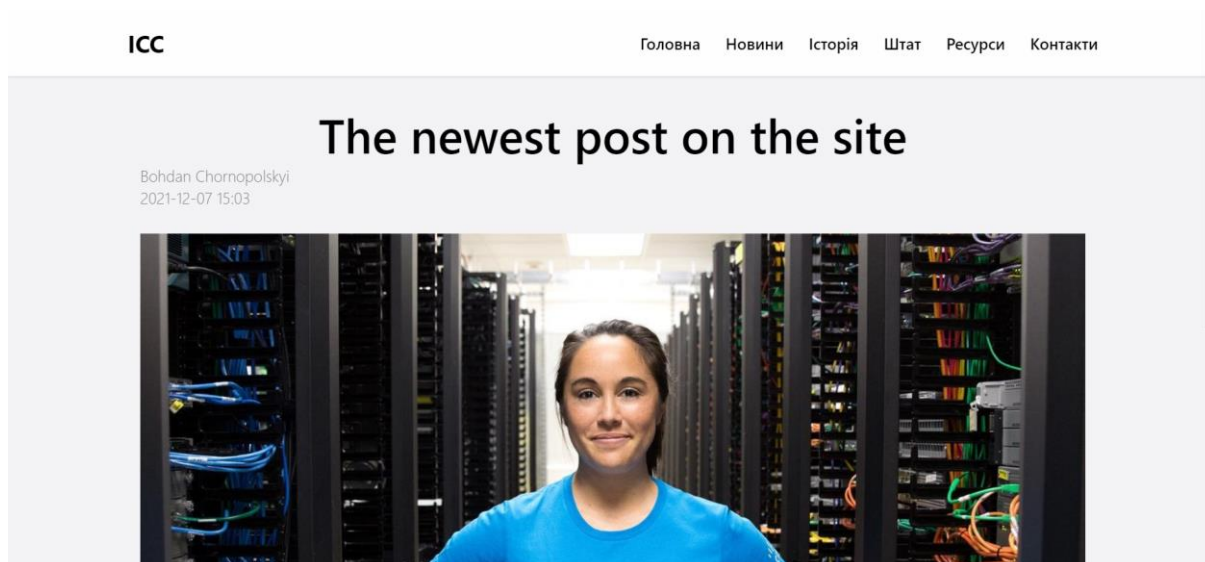


Рис. 2.5.4

2.6 Створення інших сторінок

Сторінка “Історія” - містить історію створення ІОЦ та те як він змінювався протягом свого існування (Рис.2.6.1).

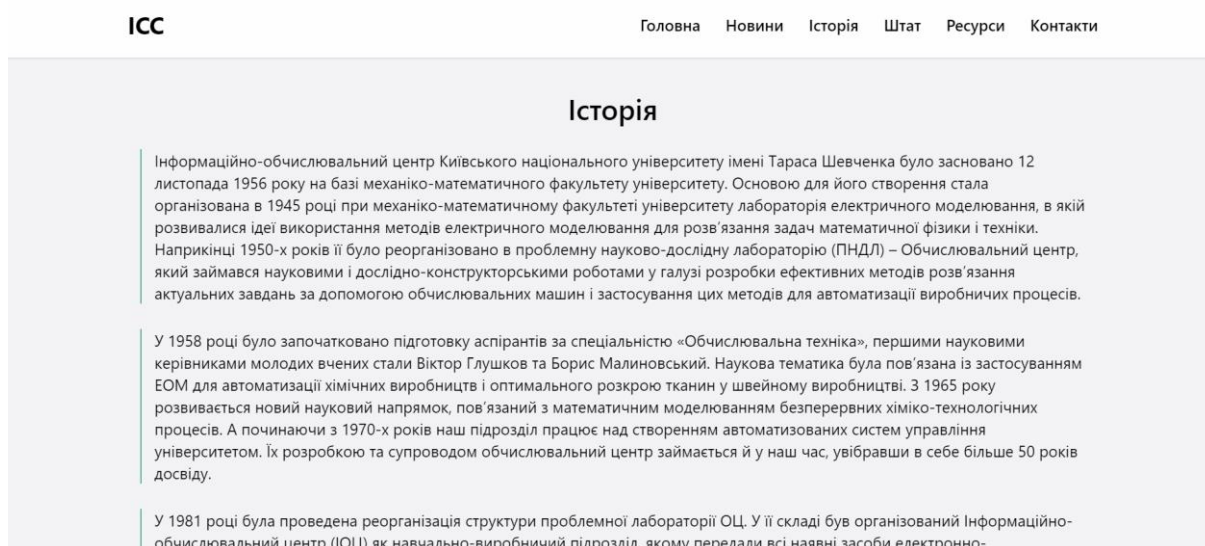


Рис.2.6.1

Також на цій сторінці було додано гарну анімацію появи тексту при завантаженні сторінки.

Сторінка “Штат” містить список керівників та працівників різних підрозділів ІОЦ (Рис. 2.6.2).

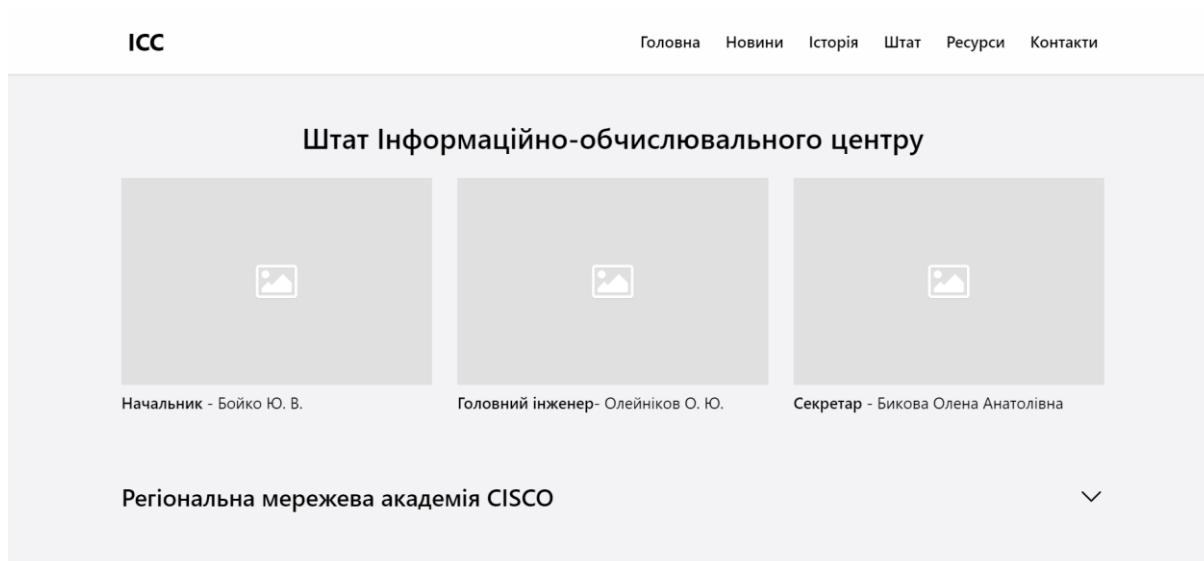


Рис. 2.6.2

Сторінка “Ресурси” містить повний список проектів та ресурсів, які ІОЦ може запропонувати користувачам мережі університету (Рис. 2.6.3).

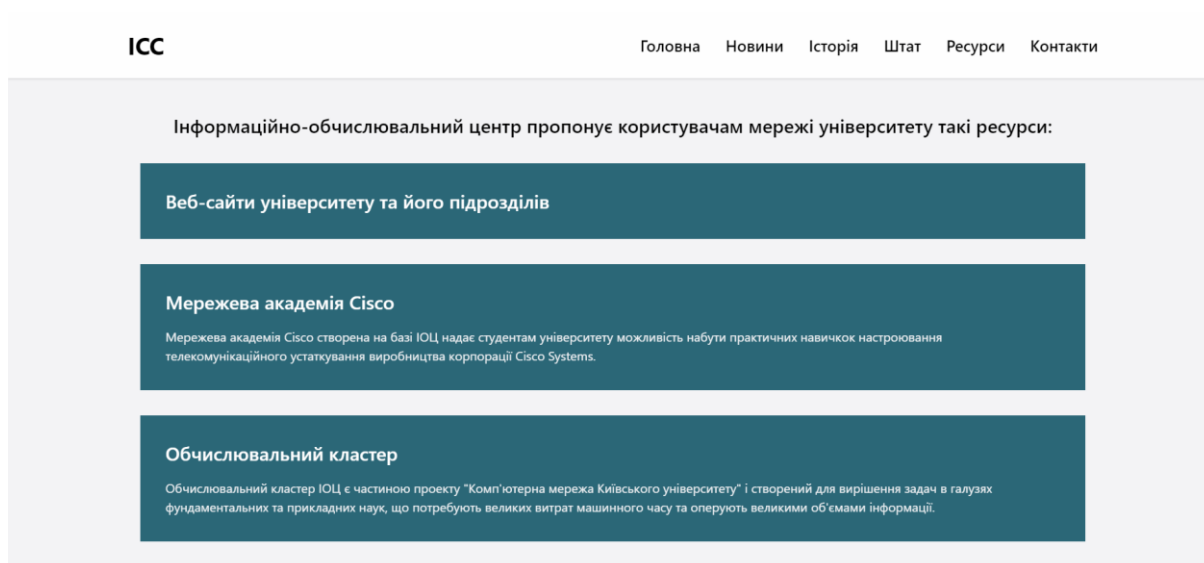


Рис. 2.6.3

На цій сторінці також наявна анімація появи блоків при завантаженні.

Сторінка “Контакти” містить контактні дані за допомогою яких користувачі можуть легко зв’язатися з представниками ІОЦ для отримання якоїсь інформації та переглянути місцезнаходження Інформаційно-обчислювального центру на карті (Рис. 2.6.4).

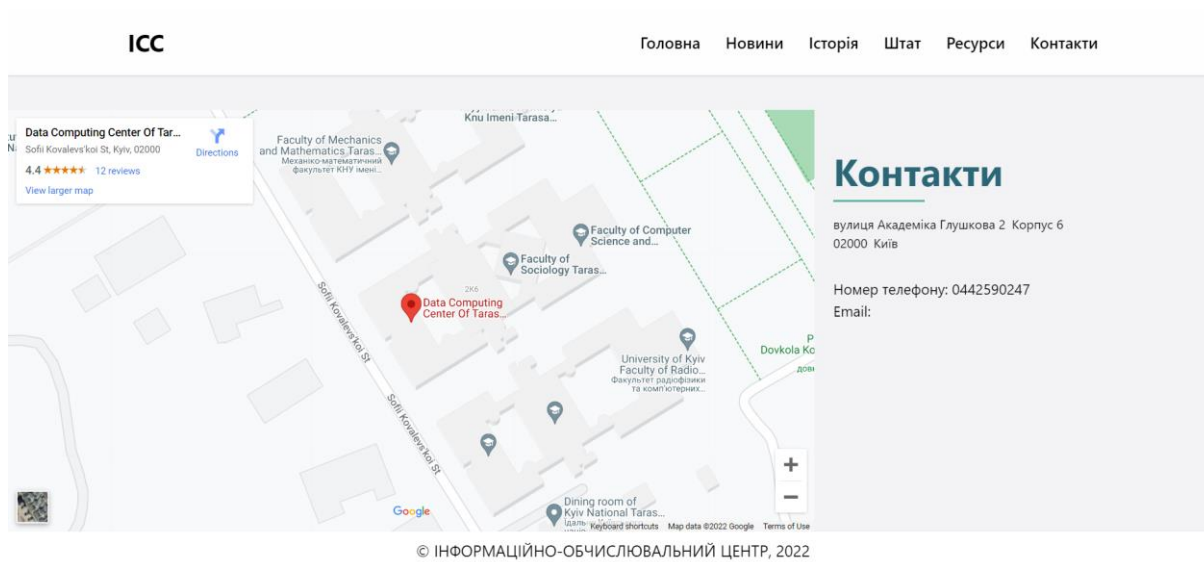


Рис. 2.6.4

Також зважаючи на те що в сучасному світі дуже велика кількість людей відвідують Інтернет ресурси використовуючи телефон, планшет тощо, вищеперелчені сторінки було адаптовано під пристрої різного розміру. Результати відображення сторінок сайту на малому екрані наведені у Додатку А.

3. Удосконалення та оптимізація створеного сайту

3.1 Перевірка веб-сайту на проблеми та помилки

Перед тим як вдосконаливати веб-сайт, спочатку потрібно його перевірити на існуючі проблеми. Для цього використаємо Lighthouse.

Lighthouse — це автоматизований інструмент з відкритим кодом для покращення якості веб-сторінок. Його можна запустити на будь-якій веб-сторінці, загальнодоступній або яка вимагає автентифікації. Він має аудит продуктивності, доступності, SEO тощо.

Lighthouse доступний в Chrome DevTools, з командного рядка або як модуль Node. Потрібно ввести в Lighthouse URL-адресу, далі він проводить серію аудитів сторінки, а потім створює звіт про те, наскільки добре сторінка працювала. В звіті можна переглянути наявні проблеми і шляхи їх вирішення.

Для того щоб перевірити наш сайт в Lighthouse відкриємо DevTools використавши комбінацію клавіш “Ctrl + Shift + I” та перейдемо до вкладки “Lighthouse” (Див Рис. 3.1).

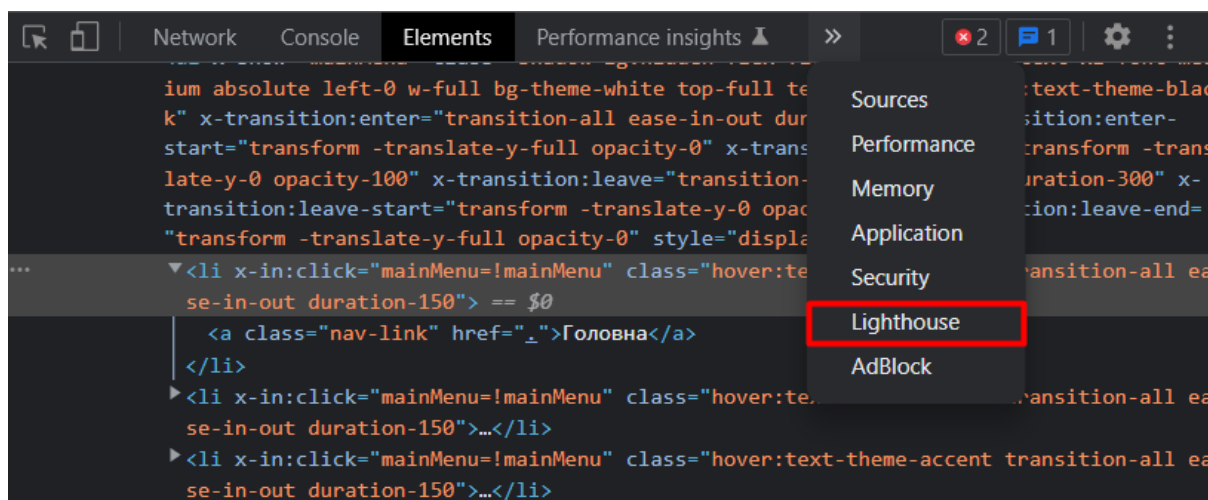


Рис. 3.1

Далі потрібно натиснути на кнопку “Згенерувати звіт” (Див Рис. 3.2)

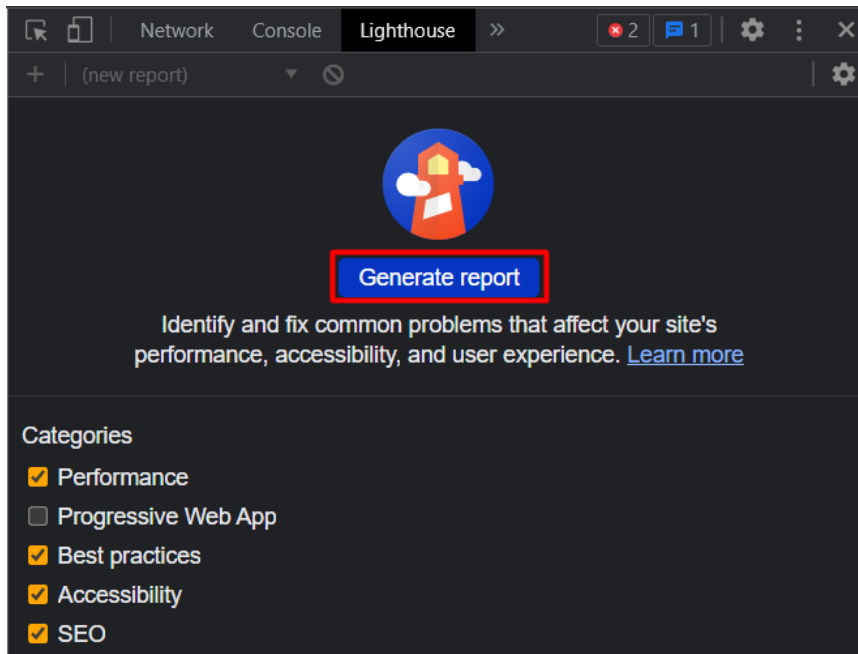


Рис. 3.2

Після того як пройде деякий час на цьому ж місці отримуємо результати нашої перевірки.

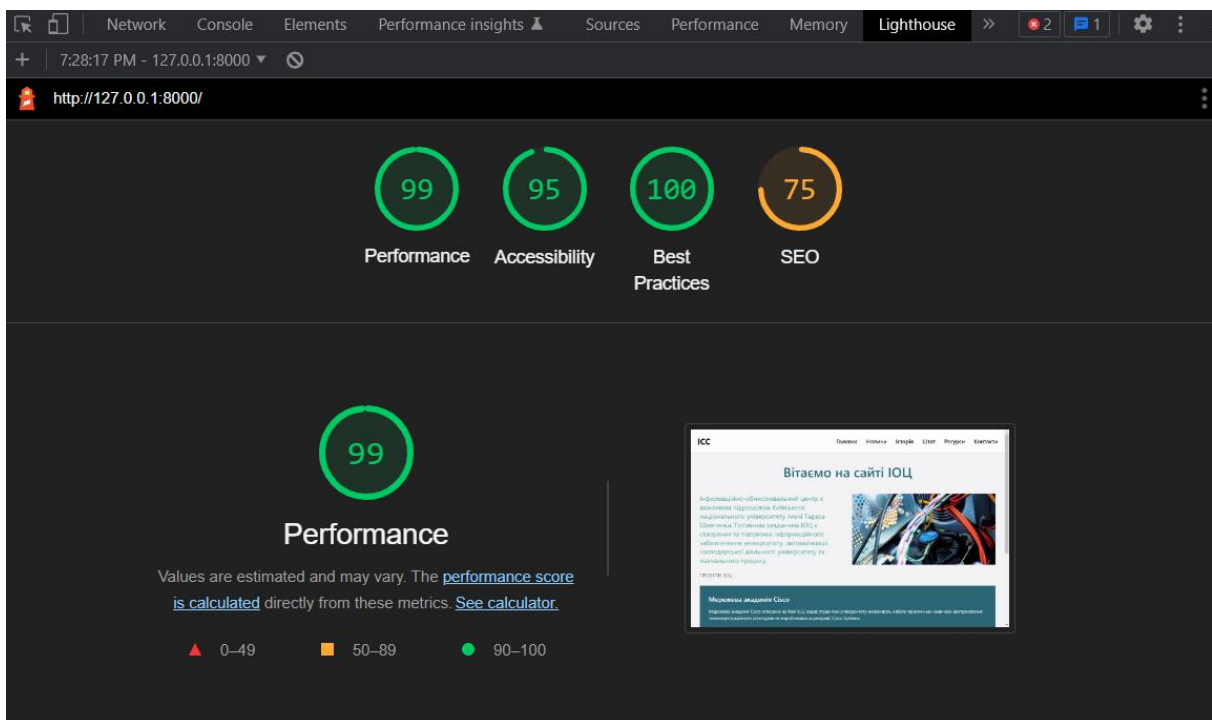


Рис. 3.3

Як видно на Рис. 3.3, на даному етапі вдалося отримати високі показники в продуктивності, доступності та в найкращих практиках, та доволі низький показник SEO оптимізації.

3.2 SEO оптимізація веб-сайту

SEO (з англ. search engine optimization) – це комплекс заходів, які спрямовані на підвищення позицій сайту в пошукових системах за відповідними запитами для загального просування сайту в мережі[14].

За допомогою того ж Lighthouse можна побачити що саме потрібно зробити щоб покращити SEO сайту ІОЦ.

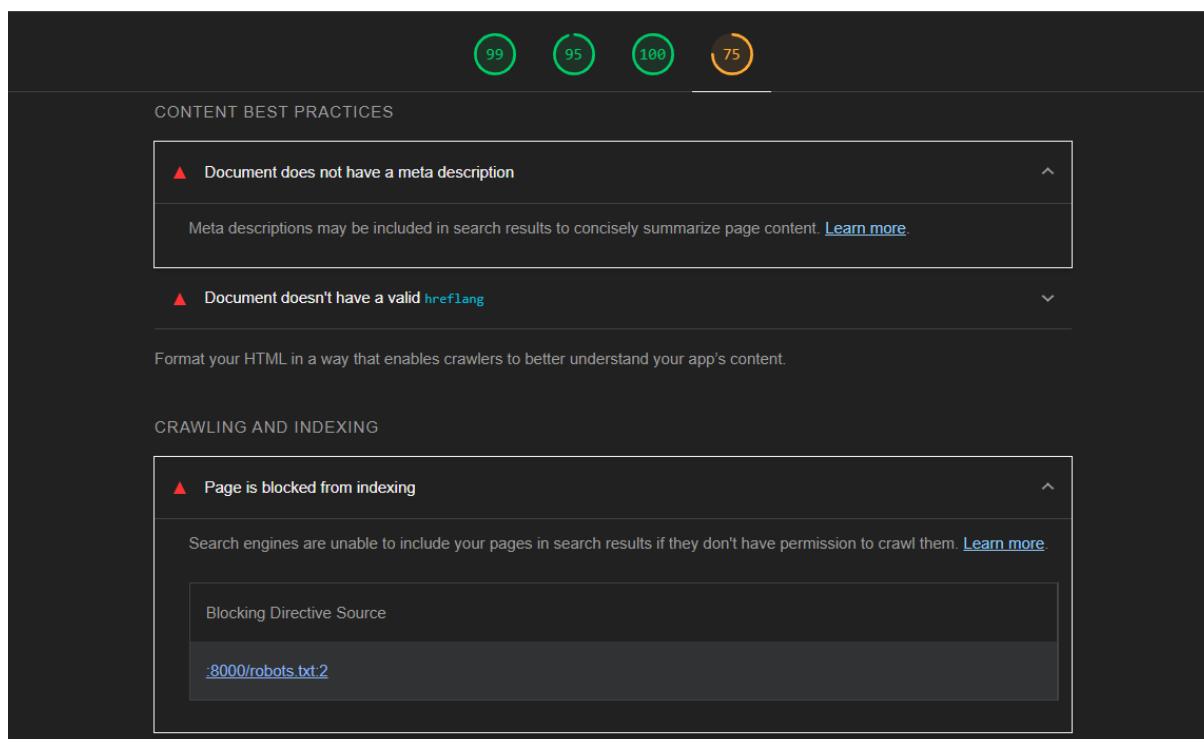


Рис. 3.4

В першу чергу бракує мета тегу “description”. Цей тег надає підсумок вмісту сторінки, який пошукові системи включають в результати пошуку.

Високоякісний унікальний мета-опис робить сторінку більш релевантною та може збільшити пошуковий трафік.

```
%if description:
<meta name="description" content="{description|h}">
%endif
<meta name="description" content="Інформаційно-обчислювальний центр Київського національного університету
```

Рис. 3.5

Було додано два мета теги, перший який береться з поля “description” або “docinfo” файлу reStructuredText та другий який буде однаковий для всіх сторінок сайту. Також Nikola автоматично генерує мета теги для створених постів (див Рис. 3.6).

```
<%def name="open_graph_metadata(post)">
<meta property="og:site_name" content="{blog_title|h}">
<meta property="og:title" content="{post.title()::70|h}">
<meta property="og:url" content="{abs_link(permalink)}">
%if post.description():
  <meta property="og:description" content="{post.description()::200|h}">
%else:
  <meta property="og:description" content="{post.text(strip_html=True)::200|h}">
%endif
%if post.previewimage:
  <meta property="og:image" content="{url_replacer(permalink, post.previewimage, lang, 'absolute')}">
%endif
<meta property="og:type" content="article">
### Will only work with Pinterest and breaks everywhere else who expect a [Facebook] URI.
###   %if post.author():
###     <meta property="article:author" content="{post.author()::h}">
###   %endif
%if post.date.isoformat():
  <meta property="article:published_time" content="{post.formatted_date('webiso')}">
%endif
%if post.tags:
  %for tag in post.tags:
    <meta property="article:tag" content="{tag|h}">
  %endfor
%endif
</%def>
```

Рис. 3.6

Далі було додано мета тег “robots”. Файл robots.txt повідомляє веб-сканерам пошукових систем, до яких URL-адрес вони можуть отримати

доступ на сайті. Це використовується в основному для того, щоб уникнути перевантаження сайту запитами.

```
<meta name="robots" content="noindex">
```

Рис. 3.7

Після повторної перевірки сайту в Lighthouse отримуємо наступний результат (див Рис. 3.8)

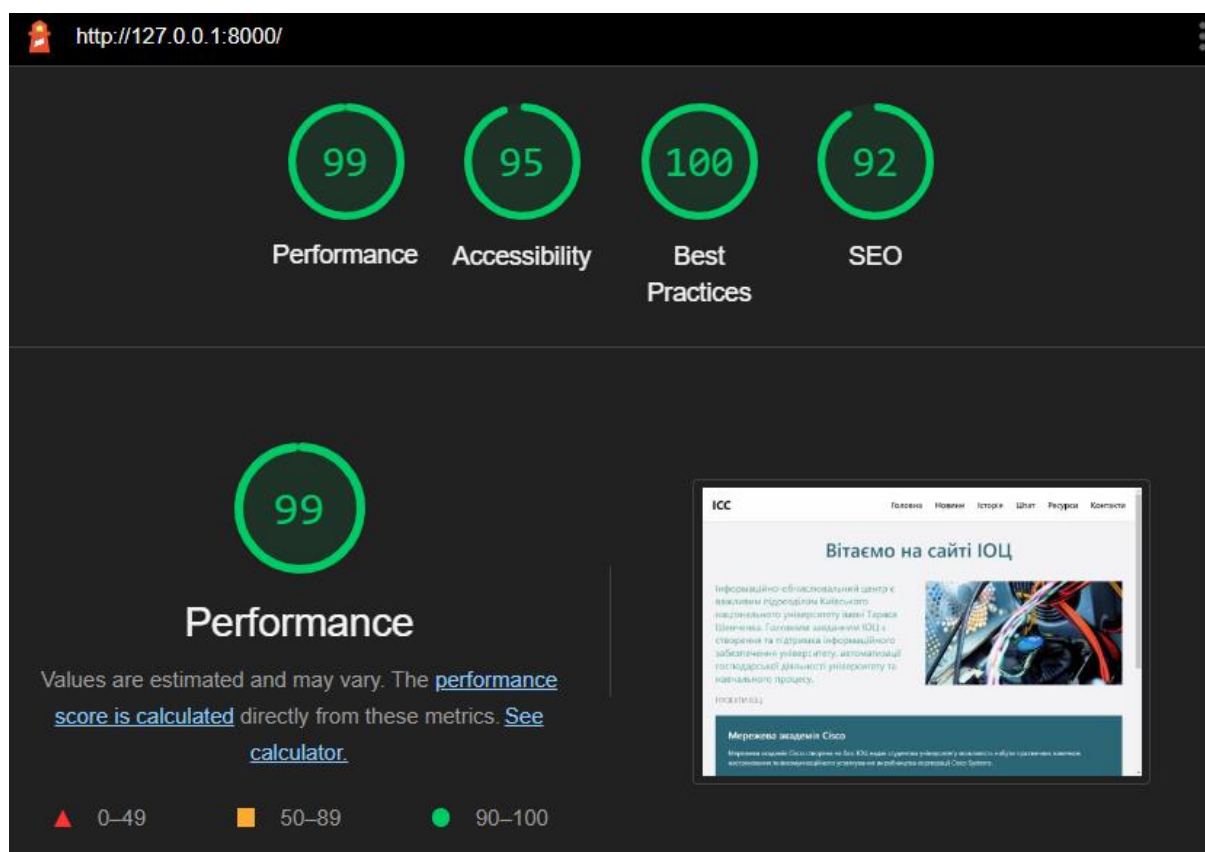


Рис. 3.8

ВИСНОВОК

Генератори статичних сайтів є хорошою альтернативою при створенні власної веб-сторінки. Вони дають можливість генерувати швидкі сайти з мінімальними затратами сил та ресурсів. Так як генерація HTML документів відбувається завчасно, сторінки завантажуються у браузері користувачів набагато швидше. Також вони використовують файли типу reStructuredText, Markdown та інші, що дозволяє будь-кому, без спеціальних знань чи навичок, додавати контент на сторінку. Відкритість та гнучкість генераторів статичних сайтів дозволяють розширювати або змінювати їх функціонал для того щоб досягти потрібних результатів.

У ході виконання даної дипломної роботи було проведено проектування та програмування веб-сайту для Інформаційно-обчислювального центру з використанням генератору статичних веб-сайтів Nikola. Зокрема було реалізовано можливість додавання нових постів, які автоматично будуть з'являтися в стрічці новин. Кінцевий продукт відповідає всім цілям та вимогам, які було зазначено, а саме:

- надання загальної інформації про ІОЦ: історію створення, основний вид діяльності, її керівників і працівників;
- публікація новин ІОЦ;
- надання інформація про проекти якими займається ІОЦ;
- надання контактних даних, для того, щоб зацікавлені користувачі могли зв'язатися з представником ІОЦ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Structuring the web with HTML [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Learn/HTML>
2. CSS: Cascadin Style Sheets [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/CSS>
3. Hillar Petersen From Static and Dynamic Websites to Static Site Generators [Електронний ресурс]. – Режим доступу: <https://core.ac.uk/download/pdf/83597655.pdf>
4. Matthew Finifter, David Wagner Exploring the Relationship Between Web Application Development Tools and Security [Електронний ресурс]. – Режим доступу: https://www.usenix.org/legacy/events/webapps11/tech/final_files/Finifter.pdf
5. What is a static site generator? [Електронний ресурс]. – Режим доступу: <https://www.cloudflare.com/learning/performance/static-site-generator/>
6. getnikola.com [Електронний ресурс]. – Режим доступу: <https://getnikola.com/>
7. nextjs.org [Електронний ресурс]. – Режим доступу: <https://nextjs.org/>
8. blog.getpelican.com [Електронний ресурс]. – Режим доступу: <https://blog.getpelican.com/>
9. gohugo.io [Електронний ресурс]. – Режим доступу: <https://gohugo.io/>
10. jekyllrb.com [Електронний ресурс]. – Режим доступу: <https://jekyllrb.com/>
11. ty.dev [Електронний ресурс]. – Режим доступу: <https://www.11ty.dev/>
12. The pros and cons of using a front-end library of framework [Електронний ресурс]. – Режим доступу: <https://bluemodus.com/articles/pros-and-cons-of-using-a-front-end-library-or-framework>


13. The pros and cons of Tailwind CSS [Електронний ресурс]. – Режим доступу: <https://www.webdesignerdepot.com/2021/09/the-pros-and-cons-of-tailwind-css/>
14. alpinejs.dev [Електронний ресурс]. – Режим доступу: <https://alpinejs.dev/>
15. greensock.com [Електронний ресурс]. – Режим доступу: <https://greensock.com/>
16. Для чого потрібна SEO оптимізація [Електронний ресурс]. – Режим доступу: <https://seoowl.com.ua/blog/dlja-chogo-potribna-seo-optymizatsija/>
17. Google Lighthouse [Електронний ресурс]. – Режим доступу: <https://web.dev/lighthouse-seo/>

Додаток А

ICC ☰

Вітаємо на сайті ІОЦ


Інформаційно-обчислювальний центр є важливим підрозділом Київського національного університету імені Тараса Шевченка. Головним завданням ІОЦ є створення та підтримка інформаційного забезпечення університету, автоматизації господарської діяльності університету та навчального процесу.



ICC ☰


The newest post on the site

Bohdan Chornopolskyi
2021-12-07 15:03



Lorem ipsum dolor sit amet consectetur adipisicing elit. Sed iure nisi voluptates neque ut praesentium numquam laudantium voluptatum impedit aut, quidem qui quos

ICC ☰



The newest post on the site

2021-12-07 15:03

Lorem ipsum dolor sit amet consectetur adipisicing elit. Sed iure nisi voluptates neque ut praesentium numquam laudantium voluptatum impedit aut, quidem qui quos quibusdam quod repellat mollitia.

ICC ☰

Історія створення та розвитку ІОЦ

Інформаційно-обчислювальний центр Київського національного університету імені Тараса Шевченка було засновано 12 листопада 1956 року на базі механіко-математичного факультету університету. Основою для його створення стала організована в 1945 році при механіко-математичному факультеті університету лабораторія електричного моделювання, в якій розвивалися ідеї використання методів електричного моделювання для розв'язання

