

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мережних технологій факультету інформатики

## **ПОБУДОВА МЕРЕЖІ МІКРОСЕРВІСІВ ДЛЯ РОЗМІЩЕННЯ ПУБЛІЧНОЇ ІНФОРМАЦІЇ ЗВО**

**Текстова частина до курсової роботи  
за спеціальністю „Комп’ютерні науки” 122**

Керівник курсової роботи  
асистент  
Кобзар О.О.

\_\_\_\_\_ (підпис)  
“ \_\_\_\_\_ ” \_\_\_\_\_ 2020 р.

Виконала студентка  
Маймескул М.В.  
“ \_\_\_\_\_ ” \_\_\_\_\_ 2020 р.

Київ 2020

## Індивідуальне завдання

ЗАТВЕРДЖУЮ

Зав. кафедри мережних технологій,  
проф., доктор фіз.-мат. наук

\_\_\_\_\_ Г. І. Малашенок  
(підпис)

„\_\_\_\_\_” \_\_\_\_\_ 2019 р.

### ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студентці 3 року навчання БП “Комп’ютерні науки”

Маймескул Марії Володимирівні

на тему:

**Побудова мережі мікросервісів для розміщення  
публічної інформації ЗВО**

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Анотація

Вступ

1. Огляд існуючих рішень для розміщення публічної інформації ЗВО

2. Огляд використаних для розробки мікросервісу інструментів

3. Побудова мережі мікросервісів

Висновки

Список посилань

Додатки

Дата видачі: „\_\_\_\_\_” \_\_\_\_\_ 2019 р.

Керівник: асистент, Кобзар О.О. \_\_\_\_\_  
(підпис)

Завдання отримала \_\_\_\_\_  
(підпис)

## Календарний план

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	11.10.2019	
2.	Огляд технічної літератури за темою роботи.	04.11.2019	
3.	Дослідження існуючого рішення для побудови мережі мікросервісів.	25.12.2019	
4.	Проектування мережі мікросервісів.	27.01.2020	
5.	Розробка мікросервісів згідно з попередньо обговореним планом	01.04.2020	
6.	Написання текстової частини роботи	27.04.2020	
7.	Створення презентації	04.05.2020	
8.	Корегування роботи згідно з результатами перевірки роботи науковим керівником.	09.05.2020	
9.	Остаточне оформлення пояснювальної роботи та презентації	10.05.2020	
10.	Подання роботи на кафедру для перевірки на плагіат	11.05.2020	
11.	Захист курсової роботи	Кінець травня	
12.	Запуск мережі мікросервісів	Кінець травня – початок червня	

Студент \_\_\_\_\_

Керівник \_\_\_\_\_

“        ”  
\_\_\_\_\_

## Зміст

Перелік використаних скорочень .....	5
Анотація .....	6
Вступ.....	7
Розділ 1. Аналіз предметної області. Огляд поточної структури вебсайту НаУКМА .....	9
1.1. Зображення.....	10
1.2. Документи .....	10
1.3. Контакти .....	11
1.4. Інформаційний пакет ЄКТС .....	12
1.5. Висновки.....	14
Розділ 2. Огляд використаних технологій. Аргументація вибору .....	15
2.1. Figma .....	15
2.2. HTML та Less .....	17
2.3. React.js.....	19
2.4. Node.js та Express.js .....	20
2.5. СКБД Microsoft SQL Server .....	21
2.6. Висновки.....	22
Розділ 3. Створення окремого мікросервісу для пакету ЄКТС.....	23
3.1. Проектування структури вебсайту.....	23
3.2. Проектування інтерфейсу користувача.....	25
3.3. Розробка клієнтської частини сайту .....	26
3.4. Розробка серверної частини сайту .....	29
3.5. Ідеї для вдосконалення сайту .....	31
3.6. Висновки.....	32
Висновки .....	33
Список літератури .....	35
Додаток. Результати опитування студентів НаУКМА стосовно поінформованості щодо пакету ЄКТС.....	37

## Перелік використаних скорочень

**CMS** – Content Management System (Система управління контентом);

**ЄКТС** – Європейська кредитно-трансферна система;

**UI** – User Interface;

**UX** – User Experience;

**СКБД** – Система керування базами даних;

**HTML** – Hypertext Markup Language (Мова розмітки гіпертексту);

**CSS** – Cascading Style Sheets;

**DOM** – Document Object Model (Об’єктна модель документу);

**npm** – Node Package Manager;

**API** – Application Programming Interface (Програмний інтерфейс застосунку);

**SQL** – Structured Query Language (Мова структурованих запитів);

**REST** – Representational State Transfer;

**JSON** – JavaScript Object Notation;

**VPN** – Virtual Private Network.

## **Анотація**

У даній роботі розглядається поточна структура офіційного вебсайту Національного університету «Києво-Могилянська академія», описуються її переваги та недоліки, та пропонується спрощення структури за допомогою створення окремих мікросервісів, пов'язаних з основним сайтом. Описується розробка мікросервісів та використані сучасні технології, а саме мова JavaScript, фреймворк React.js, та СКБД Microsoft SQL Server.

## Вступ

В наш час технології стали невід’ємною частиною щоденної рутини більшості людей. Своє почесне місце серед найбільш використовуваних дарів прогресу займає всесвітня мережа Інтернет, яка надає своїм користувачам доступ до необмеженої кількості різноманітних ресурсів та інструментів у будь-який час.

Можливість створити власний вебсайт є одним з таких інструментів. Гарний і сучасний вебсайт може стати чудовим знаряддям для налагодження комунікації з цільовою аудиторією і, як наслідок, для пришвидшення росту популярності теми, яку представляє цей сайт. Він має відповідати усім сучасним веб-стандартам і пропонувати своїм користувачам зручний та дружній інтерфейс, мати просту та зрозумілу структуру (як з боку клієнтської частини, так і з боку розробника).

Києво-Могилянська академія, звичайно, теж має свій вебсайт. Адміністрація університету подбала, щоб там знаходилася більшість потрібної студентам та абітурієнтам інформації: новини, актуальні календарні плани, багато корисних посилань і так далі.

Втім, інтерфейс користувача на сайті академії є дещо застарілим, адже регулярних оновлень, необхідних для підтримки актуальності дизайну, він не отримував. Якщо дослідити дану проблему, виявиться, що сайт знаходиться на доволі старій версії CMS «Joomla!», а саме на версії 2.5.9, що була випущена ще у 2013 році.

Як бачимо, сайт Києво-Могилянської академії, що фактично є візитною карткою нашого університету, потребує міграції на більш сучасну версію CMS. Більше того, внутрішня структура сайту також є доволі складною – наявна велика кількість плагінів та надбудов, що негативно впливають на роботу вебсайту та на швидкість його завантаження в браузері, а також через власноруч дописаний сторонній код унеможлиблюється оновлення програмного забезпечення сайту до актуальних версій.

Однією з вищезгаданих надбудов є інформаційний пакет ЄКТС. Як виявилось при детальному огляді сайту, інформаційний пакет ЄКТС є, так би мовити, сайтом у сайті – він також функціонує на базі CMS «Joomla!», яка, в свою чергу, вбудована в основний сайт. Незважно уявити, наскільки нагромадженою є структура сайту НаУКМА через такий не дуже вдалий варіант реалізації інформаційного пакету ЄКТС. Отже, обрана тема роботи є **актуальною**.

**Об’єкт дослідження** – вебсайт Національного університету «Києво-Могилянська академія».

**Предмет дослідження** – структура вебсайту Національного університету «Києво-Могилянська академія».

**Метою** роботи є осучаснення сайту Національного університету «Києво-Могилянська академія» та спрощення його поточної структури. Цю мету досягнуто шляхом відокремлення деяких складових сайту та їх перетворення у самостійні мікросервіси. Передбачається, що утворені мікросервіси стануть успішним доповненням до сайту Києво-Могилянської академії.

В роботі використано такі **методи дослідження**, як спостереження, порівняння, аналіз та синтез. **Інструменти розробки:** редактор коду Microsoft Visual Studio Code, СКБД MS SQL Server, інтегроване середовище для доступу до баз даних SQL Server Management Studio, мова програмування JavaScript.

Для досягнення мети роботи **поставлено** такі **завдання**:

- Дослідити поточний варіант реалізації вебсайту Києво-Могилянської академії, виявити усі його переваги та недоліки;
- Ознайомитися з програмною технологією Node.js, фронт-енд фреймворком React.js та CSS-препроцесором Less;
- Здійснити відокремлення інформаційного пакету ЄКТС від основного сайту НаУКМА, перетворити його у самостійний мікросервіс.



## **Розділ 1. Аналіз предметної області. Огляд поточної структури вебсайту НаУКМА**

Офіційний вебсайт Києво-Могилянської академії був розроблений на основі системи управління контентом «Joomla!» та наразі функціонує на її версії 2.5.9, що була випущена у 2013 році. Зрозуміло, що версія семирічної давності не може відповідати сучасним стандартам веб-розробки та уявленням про зручний та гарний інтерфейс користувача. Отже, вебсайт НаУКМА потребує міграції на сучаснішу версію використовуваної CMS.

Втім, враховуючи структуру сайту, його оновлення виявилось тривалим процесом, що включає в себе кілька стадій. Однією з цих стадій є спрощення структури сайту шляхом вилучення з нього масивних компонентів. Чому це необхідний етап роботи?

Такий складний багатокомпонентний сайт, яким є офіційний сайт Києво-Могилянської академії, не можна буде перенести на нову версію системи управління контентом виключно за допомогою вбудованих інструментів для оновлення. Автоматична міграція не зможе забезпечити відповідне оновлення усіх використовуваних плагінів та компонентів, тож за такого сценарію частина сайту просто перестане працювати після міграції – це показала спроба міграції, проведена у тестовому режимі на локальному сервері.

Іншим чинником, що ускладнює процес оновлення, є штучно вбудований в структуру сайту інформаційний пакет ЄКТС. У випадку оновлення головного сайту цей компонент лишиться без змін. Таким чином, він також має бути оновлений окремо.

Окрім ручного оновлення плагінів та компонентів сайту НаУКМА, було розглянуто інакшу опцію як один з можливих сценаріїв оновлення сайту – перетворення найбільш масивних компонентів на окремі мікросервіси, що були б пов'язані з основним вебсайтом Києво-Могилянською академії, але водночас функціонували незалежно від нього. Це спростило б роботу як адміністраторам сайту, так і розробникам, що займаються його підтримкою.

Після детального вивчення наявних на вебсайті НаУКМА компонентів, було складено список тих з них, що претендують на роль самостійного мікросервісу:

### **1.1. Зображення**

Усі зображення з фотогалереї, наявної на вебсайті НаУКМА, зберігаються на сервері в загальній директорії для зображень `images/`, що в свою чергу знаходиться в кореневому каталозі сайту. Ці зображення мають досить великий розмір та роблять резервну копію сайту надто об'ємною.

Враховуючи розмір папки з зображеннями, розглядалася можливість винести усі зображення на окремий підсервіс для спрощення роботи з ними. Втім, під час детального аналізу сайту НаУКМА виявилось, що для роботи з зображеннями сайт використовує спеціально створений для цього сторонній компонент – Phoca Gallery [1].

Phoca Gallery – це галерея зображень, розроблена спеціально для CMS «Joomla!». Цей безкоштовний компонент дозволяє у зручному режимі додавати на сайт зображення та відео з відеохостингу YouTube.

Хоча вищезгаданий компонент є безкоштовним, підтримується з боку розробника та регулярно оновлюється (а, отже, є сумісним з більш сучасними версіями CMS «Joomla!»), від його використання вирішено було відмовитись. Поки вивчалось питання щодо фотографій, їх вже було перенесено на зовнішній підсервіс (`media.ukma.edu.ua`). Таким чином, проблема вирішилася сама собою.

### **1.2. Документи**

Оскільки сайт НаУКМА є офіційним сайтом університету, він відповідно до ст. 30 закону «Про освіту» [2] має містити різноманітні офіційні документи: розклади занять та сесії для кожного факультету, різноманітні положення, тощо. За вісім років існування сайту подібних документів накопичилося немало, і всі

вони також зберігаються на сервері, у кореновому каталозі сайту, у папках documents/ та dmdocuments/.

Задля спрощення структури сайту, розглядалася опція переносу документів на сторонній ресурс, створений спеціально для зберігання документів. В тому числі, можливими варіантами виступали Google Drive, спеціалізований сервіс для зберігання файлів, та сайт eKMAIR, що є частиною групи сайтів Києво-Могилянської академії та є репозиторієм робіт студентів, викладачів та працівників НаУКМА.

Щодо першого варіанту, він був відкинутий на користь сховища документів eKMAIR. Оскільки eKMAIR належить Києво-Могилянській академії, було вирішено зупинитися саме на ньому задля уникнення залежності офіційного вебсайту НаУКМА від сторонніх сервісів.

Втім, спілкування з адміністрацією репозиторію eKMAIR виявила, що перенести до цього сервісу всі документи з основного сайту НаУКМА не вийде – eKMAIR має чітко визначені правила щодо того, документи яких категорій можуть там знаходитися: наприклад, наукові роботи студентів та викладачів. Більшість файлів з сайту НаУКМА не відповідають вимогам для того, щоб бути розміщеними у сховищі eKMAIR.

У поточній версії сайту робота з документами здійснюється за допомогою стороннього розширення DOCman [3]. Дане розширення є зручним інструментом для роботи з документами, також воно підтримується для останніх версій CMS «Joomla!». Враховуючи це, DOCman може і надалі використовуватися в якості засобу управління документами.

### **1.3. Контакти**

Контактні дані осіб на сайті НаУКМА розміщуються з використанням власноруч реалізованого інструменту для роботи з контактами, оскільки вбудований у систему управління контентом «Joomla!» менеджер контактів не задовольнив потреби адміністраторів сайту. Власноруч розроблений менеджер

контактів не є якісним рішенням, до того ж, він обмежує можливість оновлення сайту.

Втім, вбудований менеджер контактів [4] є особливо зручним для використання на сучасних версіях використовуваної CMS – він надає зручний інтерфейс для створення, редагування та перегляду контактів. Наявні засоби для фільтрації, сортування та пошуку контактів.

Оскільки менеджер контактів на новій версії CMS «Joomla!» є досить зручним і схожим на інструмент, який наразі використовується на сайті, було прийнято рішення у подальшому не відмовлятися від використання вбудованого менеджера контактів «Joomla!».

#### **1.4. Інформаційний пакет ЄКТС**

Як показало дослідження, проведене в рамках даної роботи, ЄКТС є однією з найбільш суперечливих частин офіційного вебсайту Києво-Могилянської академії.

По-перше, за структурою його коротко можна описати як «сайт у сайті». Тобто у вебсайті НаУКМА пакет ЄКТС присутній саме як вбудований підсайт, а не як набір сторінок. Зрозуміло, що така реалізація значно ускладнила як сам сайт, так і роботу з ним.

По-друге, візуальна складова сайту також не є сучасною. Галузь веб-дизайну давно пішла вперед, з'явилося таке поняття як UI/UX дизайн, що винесло значення інтерфейсу користувача на новий рівень. Ці набуті веб-розробниками знання варто застосувати і для інформаційного пакету ЄКТС.

Розглянемо проблему поточної структури інформаційного пакету ЄКТС детальніше. Вибір подібних архітектурних рішень, що, можливо, могли мати місце вісім років тому, зараз здається необґрунтованим та незрозумілим. За наявності великого вибору різноманітних методів веб-розробки, такий підхід не є рекомендованим.

Інформаційний пакет ЄКТС використовує MySQL як СКБД для бази даних для CMS “Joomla!”, яка в свою чергу фактично містить лише кілька текстових сторінок. До того ж, певні елементи сайту завантажують дані для відображення з СКБД Microsoft SQL Server за допомогою власноруч написаного коду, що порушує звичний процес роботи сайту. Таким чином, інформаційний пакет ЄКТС варто відокремити від сайту НаУКМА та оновити.

Можливим варіантом рішення даної проблеми могло б стати створення на базі системи управління контентом «Joomla!» окремого, вже не вбудованого у основний сайт, веб-застосунку, призначеного для інформаційного пакету ЄКТС. Втім, перевага була надана альтернативному рішення – сайт для ЄКТС буде побудовано на основі клієнт-серверної архітектури за допомогою сучасних інструментів веб-розробки.

Щодо зовнішнього вигляду сайту, на якому зараз знаходиться інформаційний пакет ЄКТС – поточний дизайн також вимагає оновлення з використанням сучасного підходу. Серед зауважень, які є до інтерфейсу користувача, можна виділити наступні:

- Замалий розмір шрифту, яким написаний текст на сайті. Це може спричинити проблеми з розпізнаванням тексту у людей з вадами зору, що є неприпустимим, адже інформація, розміщена на даному сайті, має бути доступна для усіх бажаючих з нею ознайомитися;
- Яскраво-жовтий колір, що присутній в оформленні елементів навігації по сайту, також є не дуже вдалим. За високих яскравості та контрастності екрану користувача, такі фонові кольори елементів можуть сліпити, внаслідок чого виникають проблеми з розпізнаванням присутнього на цих елементах тексту.

Гарний та зручний інтерфейс користувача є справді важливим для студентів. Це доводить опитування, проведене серед студентів різних факультетів (див. Додаток). Згідно з цим опитуванням, 83 відсотки респондентів вважають, що дизайн сайту є важливим, навіть якщо основна ціль ресурсу – донести якусь інформацію до користувача.

Отже, приведення до відповідності сучасним веб-стандартам зовнішнього вигляду сайту для інформаційного пакету ЄКТС також є важливим завданням, що має бути виконане в рамках цієї роботи.

Дослідження інформаційного пакету ЄКТС виявило ще одну проблему – погана поінформованість студентів щодо цього ресурсу. В інформаційному пакеті ЄКТС зібрано багато корисної інформації для студентів, зокрема про навчальний процес, гуртожитки, тощо. Вищезгадане опитування показало, що про практичну користь пакету ЄКТС знають лише 48 студентів, що складає менше половини опитаних.

Можливо, така проблема частково викликана назвою ресурсу, що не є інтуїтивно зрозумілою та не доносить до студента інформацію про зміст сайту. Існує потреба змінити назву «Інформаційний пакет ЄКТС» на користь більш зрозумілого терміну.

## **1.5. Висновки**

У даному розділі було розглянуто декілька компонентів офіційного вебсайту Києво-Могилянської академії, що могли стати окремими сайтами.

Проведене дослідження показало, що суттєвого оновлення потребує сайт, призначений для інформаційного пакету ЄКТС. Через складну структуру та застарілий зовнішній вигляд, цей ресурс виглядає непрезентабельно. Більше того, велика частина студентів є непоінформованою стосовно практичної користі інформаційного пакету ЄКТС.

Отже, в даній роботі буде проведено осучаснення інформаційного пакету ЄКТС – буде змінено його інтерфейс користувача, а також структуру. Після завершення оновлення, сайт пакету ЄКТС буде функціонувати як окремий мікросервіс.

## Розділ 2. Огляд використаних технологій.

### Аргументація вибору

Вибір технологій для розробки проєкту є важливим підготовчим етапом, адже вдало вибрана сукупність інструментів може значно полегшити розробку проєкту та його подальшу підтримку.

Для реалізації сайту для інформаційного пакету ЄКТС було обрано веб-застосунок Figma [5] для побудови макету сайту, мову розмітки HTML [6] для верстки веб-сторінок, мову Less [7] для створення стилів до веб-сторінок, фреймворк React.js [8] для розробки клієнтської частини сайту, програмну платформу Node.js [9] та створений спеціально для неї фреймворк Express.js [10] для розробки серверної частини сайту, та драйвер для технології Node.js, який використовується для доступу до СКБД Microsoft SQL Server [11] з Node.js-серверу.

#### 2.1. Figma

Figma – це векторний графічний редактор для веб-дизайну, що дозволяє створювати, переглядати та редагувати макети та надає можливість колективної роботи в режимі реального часу.

За допомогою Figma можна відтворювати будь-які елементи користувацького інтерфейсу, наприклад кнопки, форми, меню, і так далі. Також існує можливість симуляції сайту за допомогою інтерактивних прототипів, що дозволить перевірити, як працюють елементи інтерфейсу.

Figma досить успішно конкурує з такими відомими аналогами як Adobe Photoshop та Sketch, які, хоч і довго існують на ринку, поступаються редактору Figma у низці властивостей. Основні переваги, що виділяють Figma серед конкурентів, описані нижче.

**Можливість роботи у браузері.** На відміну від Adobe Photoshop та Sketch, які є цілком десктопними застосунками, користуватися Figma можна прямо у

браузері. Єдиним недоліком такого підходу є потреба в постійному доступі до мережі Інтернет. Втім, враховуючи розмір таких програм як Adobe Photoshop, який лише частково використовується для веб-дизайну, перевага веб-програми стає ще очевиднішою, а вищезгаданий недолік можна пробачити.

**Система контролю версій.** Доволі корисна функція, яка може бути знайома розробникам по системі контролю версій Git. Аналогічно, і в Figma можна переглядати список внесених змін та за потреби повертатися до однієї з попередніх версій.

**Відносна безкоштовність.** Якщо говорити про програми, розроблені компанією Adobe, то майже усі вони вимагають немалих фінансових вкладень. Щодо редактора Figma, він має безкоштовну версію, яка передбачає одночасну роботу не більше двох людей та не більше, ніж над трьома проєктами. За безкоштовного користування, історія змін зберігається 30 днів. Такого режиму користування буде досить для більшості невеликих проєктів, яким є і дана робота. За потреби можна перейти на платну підписку, що обійдеться у 12\$ і дозволить мати необмежену кількість проєктів, необмежену історію змін та інтеграцію з корпоративним месенджером Slack. Однак за кожного додаткового редактора потрібно буде доплатити 12\$. Для розробки макету сайту для інформаційного пакету ЄКТС було прийнято рішення зупинитись на безкоштовній версії Figma.

**Спільна робота.** Вже згадана вище функція спільного редагування документів у режимі реального часу є дуже корисною і надає ширші можливості, ніж можна уявити на перший погляд. Окрім власне одночасної роботи над макетом, Figma дозволяє користувачам залишати свої коментарі. Коментарі можуть залишати не лише редактори, а і користувачі, які мають право лише на читання документу (їх кількість необмежена навіть у безкоштовній версії редактора). Цей функціонал значно спрощує зберігання сумісності стилів та процес комунікації між членами команди.

**Прототипування.** Редактор Figma надає можливість тестування користувацького інтерфейсу сайту до того, як його макет буде відтворено у



верстці засобами мови розмітки. Така функція дає змогу виявити недоліки інтерфейсу ще на етапі його проектування, що значно скоротить час розробки. Налаштувати та протестувати можна будь-які елементи: кнопки, спливаючі меню, анімовані вікна, тощо.



**Рисунок 2.1.** Інтерфейс Figma

Враховуючи усі наведені переваги, Figma є вдалим вибором для проектування макету сайту для інформаційного пакету ЄКТС.

## 2.2. HTML та Less

HTML – це стандартна мова розмітки, що використовується для створення веб-сторінок та описує їхню структуру. HTML-сторінка являє собою сукупність представлених тегам елементів, які пояснюють браузеру, як саме варто відображати значущий вміст сторінки. Гнучкість та зрозумілість мови HTML роблять її ідеальним варіантом для верстки веб-сторінок.

Less є розширенням CSS – формальної мови опису зовнішнього вигляду документу. Це означає, що правильний CSS-код може бути проінтерпретований також як і правильний Less-код. Less має синтаксис, аналогічний до синтаксису CSS, тому перехід з CSS на даний препроцесор відбувається швидко й непомітно.

Це розширення має багато функцій, які відсутні або ж не є зручними для використання у CSS. Ось деякі з них: змінні (див. рис. 2.2.), вкладення, оператори та функції.

```
@mycolor: #696969;

#header {
  background-color: @mycolor;
}

/*Скомпільований файл з роширенням
.css міститиме наступні стилі*/

#header {
  background-color: #696969
}
```

**Рисунок 2.2.** Використання змінних у Less

Оскільки CSS є підмножиною Less, але не навпаки, Less-код має бути скомпільований у CSS-код для використання разом з HTML. Це можна зробити як через командний рядок операційної системи (вимагає встановлення додаткового програмного забезпечення), так і прямо в браузері під час завантаження сторінки. Для цього потрібно буде завантажити файл, що виконає компіляцію, та вставити його у HTML-сторінку за допомогою спеціального тегу `<script></script>` (див. рис. 2.3.). Втім, такий підхід не рекомендується, оскільки може сповільнити швидкість обробки веб-сторінки.

```
<!-- Прив'язуємо до сторінки .less файл зі стилями -->
<link rel="stylesheet/less" href="styles.less" />
<!-- Вставляємо компілятор, що створить новий
файл зі стилями з розширенням .css -->
<script src="less.js"></script>
```

**Рисунок 2.3.** Використання файлу less.js для компіляції у браузері

Отже, створений у редакторі Figma макет сайту буде реалізований з використанням засобів HTML та Less. Ця зв'язка є стандартним рішенням у розробці вебсайтів.

### 2.3. React.js

React є JavaScript-бібліотекою з відкритим програмним кодом, що використовується для розробки динамічних користувацьких інтерфейсів. Дана бібліотека підтримується Facebook, компанією-гігантом у сфері інформаційних технологій, та спільнотою окремих розробників і корпорацій. Це означає, що React регулярно отримує оновлення, що покращують його роботу.

Найбільш широко React застосовується для розробки так званих single-page застосунків, отже, він добре підходить для сайту інформаційного пакету ЄКТС. Також даний фреймворк дозволяє створювати доступні для повторного використання елементи – компоненти, – які є цеглинами для побудови більш складного інтерфейсу. Компоненти самі керують своїм станом, що зменшує кількість роботи, яку мусить виконувати розробник клієнтської частини, а завдяки своїй декларативності, React.js робить написаний код зрозумілішим та спрощує процес його налагодження.

Міграція між версіями бібліотеки React.js не є складною, особливо враховуючи той факт, що компанія-розробник Facebook надає засоби для автоматизації більшої частини цього процесу.

Для побудови компонентів використовується JSX – синтаксичне розширення мови JavaScript, що дозволяє використовувати всередині JavaScript коду HTML елементи. Завдяки такому підходу до розробки інтерфейсів користувача забезпечується їх висока гнучкість та динамічність.

Ще однією перевагою бібліотеки React є те, що вона надає інструмент для відслідковування життєвого циклу компонентів. Цими інструментом є набір методів, що викликаються тоді, коли відбувається якась подія у житті компоненту. Це надає змогу на різних етапах його функціонування робити

потрібні дії. Наприклад, метод `componentDidMount()` викликається після того, як компонент вбудовано в DOM. Даний метод рекомендовано для виконання різноманітних запитів (див. рис.2.4.).

```
componentDidMount() {  
    fetch(`http://localhost:3000`)  
    .then(response => response.json())  
    .then(data => console.log(data));  
}
```

**Рисунок 2.4.** Створення запиту до сервера

Щодо роботи з запитами, React є сумісним з такою AJAX-бібліотекою як Axios, що дозволяє зручно та швидко формувати запити до серверу.

Враховуючи вищенаведені факти, вибір React.js в якості основного інструменту розробки клієнтської частини застосунку можна вважати вдалим.

## **2.4. Node.js та Express.js**

Дані технології отримали особливо широке застосування в якості основних інструментів розробки веб-серверів. Нарівні з такими відомими та популярними мовами для backend-розробки як PHP, .NET та Python, Node.js активно використовується у сфері веб-програмування.

Node.js за своєю сутністю є JavaScript-оточенням, що значно розширює можливості мови, надаючи засоби для підключення різноманітних бібліотек та уможлиблюючи виконання JavaScript-коду за межами браузеру. Таким чином, JavaScript із мови вузького призначення перетворюється у доволі багатофункціональний інструмент для розробки масштабованих застосунків.

Додатковою перевагою Node.js є наявність власного пакетного менеджера – npm, що надає дуже зручний інтерфейс для завантаження, оновлення та управління модулями, що використовуються в процесі розробки проекту. Варто зазначити, що за допомогою npm можна завантажувати як бібліотеки для серверної частини застосунку, так і для клієнтської (в тому числі і React.js).

Завдяки цьому Node.js може використовуватися і за умови створення застосунків з реалізованою лише клієнтською частиною.

Express.js є безкоштовним Node.js-фреймворком з відкритим кодом, що завдяки наявним HTTP-методам та проміжним обробникам дозволяє легко прописувати систему маршрутів та створювати надійні API – саме те, що потрібно для створення вебсайту для інформаційного пакету ЄКТС (див. рис.2.5.).

```
// підключаємо Express
const express = require("express");

// створюємо об'єкт застосунку
const app = express();

// визначаємо обробку маршруту "/"
app.get("/", function(request, response){

    // відправляємо відповідь
    response.send("<h2>Hello world!</h2>");

});

//Відкриваємо порт для сервера
app.listen(3000);
```

**Рисунок 2.5.** Створення простого серверу засобами Node та Express

Таким чином, зв'язка React.js+Node.js є реалізацією парадигми «JavaScript усюди», що об'єднує розробку усіх компонентів веб-застосунку навколо однієї мови без потреби звертатися до додаткових сторонніх технологій. Даний набір технологій задовольняє усі потреби, що можуть виникнути під час реалізації вебсайту, тому було вирішено зупинитися саме на ньому.

## 2.5. СКБД Microsoft SQL Server

Microsoft SQL Server – це система керування базами даних, що була розроблена та підтримується корпорацією Microsoft. Для запитів

використовується мова Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO щодо структурованої мови запитів SQL із розширеннями. Microsoft SQL Server вже багато років є сильним конкурентом для таких систем керування базами даних як MySQL, PostgreSQL, тощо.

Вибір Microsoft SQL Server в якості системи керування базами даних продиктований тим, що поточний сайт для інформаційного пакету ЄКТС використовує дані з університетської бази даних «Оптіма», яка, в свою чергу, функціонує на основі саме цієї СКБД. Відповідно, аби уникнути складного процесу міграції даних з однієї бази даних до іншої, було вирішено і надалі продовжувати використовувати Microsoft SQL Server.

Для перегляду вмісту бази даних буде використовуватися спеціально створена для цього утиліта SQL Server Management Studio, що дозволяє зручно переглядати та редагувати дані у базі.

З'єднання серверу з базою даних «Оптіма» буде забезпечено шляхом встановлення та налаштування драйвера для СКБД Microsoft SQL Server, розробленого спеціально для технології Node.js.

## **2.6. Висновки**

До вибору інструментів розробки слід підходити відповідально, оскільки від цього залежить зручність, швидкість та якість реалізації вебсайту. Засоби для реалізації вебсайту, присвяченого інформаційному пакету ЄКТС, було обрано з огляду на їх доцільність у виконанні поставлених задач та їх сумісність. Також особливий акцент було зроблено на тому, як саме обрані інструменти можуть полегшити процес розробки.

## **Розділ 3. Створення окремого мікросервісу для пакету ЄКТС**

Вебсайт для інформаційного пакету ЄКТС являтиме собою клієнт-серверний застосунок та буде розроблений з використанням засобів веб-розробки, описаних у розділі 2.

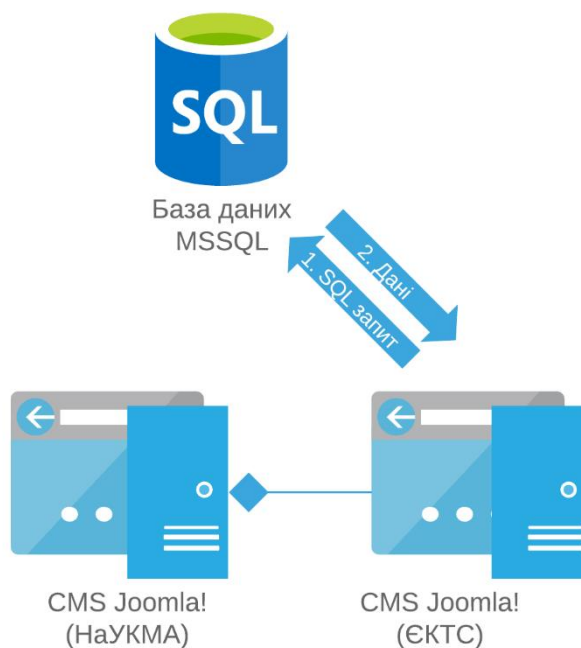
Процес розробки буде складатися з етапу проєктування системи, а також з етапу імплементації, що включатиме в себе реалізацію клієнтської та серверної частин.

### **3.1. Проєктування структури вебсайту**

Грамотне планування є важливим етапом у розробці програмного забезпечення. Протягом цього етапу майже повністю визначається наступний процес розробки та формується уявлення про те, яким має бути її кінцевий продукт. Під час проєктування важливо продумати усі нюанси розробки та експлуатації програмного забезпечення, що розробляється. Це допоможе зменшити кількість потенційних помилок, що можуть виникнути на етапі поствиробництва.

Стосовно поточної структури сайту для інформаційного пакету ЄКТС (див. рис. 3.1.), вона є досить складною та нелогічною. Основна проблема в тому, що вебсайт є дивно вбудованою складовою офіційного вебсайту Києво-Могилянської академії. Їхня взаємодія утворює структуру «сайт у сайті», створюючи непотрібні залежності та додаткові складності під час підтримки та оновлення обох вебсайтів.

На сьогодні, вибір такої несучасної та непрактичної архітектури не є виправданим, тому було прийнято рішення повністю її переробити.



**Рисунок 3.1.** Поточна структура вебсайту ЄКТС

Оскільки було вирішено відмовитися від створення сайту на основі CMS «Joomla!», натомість надавши перевагу вебсайту з простою клієнт-серверною архітектурою, структура сайту для інформаційного пакету ЄКТС тепер є набагато простішою та зрозумілішою (див. рис. 3.2.).



**Рисунок 3.2.** Оновлена структура вебсайту ЄКТС



Оновлена структура вебсайту для пакету ЄКТС дозволить значно спростити процеси оновлення та підтримки вебсайту.

Для створення діаграм, що відображають структуру сайту для інформаційного пакету ЄКТС, було використано засоби веб-застосунку для побудови схем Lucidchart ([lucidchart.com](https://lucidchart.com)).

### **3.2. Проектування інтерфейсу користувача**

Інтерфейс користувача було спроектовано з використанням графічного онлайн-редактора Figma, основний функціонал якого був описаний у другому розділі.

Під час створення макету сайту було вирішено в цілому зберегти поточне розміщення елементів управління, але загалом переосмислити дизайн інтерфейсу користувача та подати його у новому, сучасному вигляді.

Таким чином, елементи управління для вибору основних трьох категорій були переміщені у шапку сайту. Зони тексту та меню підкатегорій було вирішено залишити так, як є зараз на поточному сайті для інформаційного пакету ЄКТС, оскільки таке розміщення даних елементів є більш звичним для користувачів та адміністрації сайту.

Щодо текстового елементу з описом ЄКТС, було прийнято рішення виділити окрему категорію «Про ЄКТС» у шапці сайту, та видавати відповідну інформацію у випадаючому вікні. Таким чином, користувач у будь-який момент зможе звернутися до цієї категорії та побачити пояснення стосовно того, що таке ЄКТС.

Оскільки Києво-Могилянська академія має офіційний бренд-бук [12], що містить правила використання символіки НаУКМА, дизайн сайту було вирішено узгодити з цими правилами. Так, акцентним кольором було обрано Pantone 288C – глибокий синій колір. В цілому ж новий дизайн сайту можна описати як мінімалістичний (див. рис. 3.3.).

## Загальний опис закладу

### Загальний опис закладу

Назва і адреса

Академічний календар

Адміністрація

Загальні вимоги до зарахування

Національний університет «Києво-Могилянська академія» (НаУКМА) - спадкоємець одного з найстаріших вищих навчальних закладів України і Центрально-Східної Європи, першого офіційно визнаного в східнослов'янському світі - Києво-Могилянської академії. Академія заснована 15 жовтня 1619 року як школа для дітей усіх станів на кошти, відписані знатною князю Галицькою Гулевичівною. 1632 року школа набула статусу колегіуму. 1658 року колегіум офіційно набув юридичних прав вищої школи й титулу «академія» за Гадяцькою угодою. 1817 року рішенням Священного синоду Києво-Могилянську академію закрито, а на її території відкрито Київську Духовну семінарію. 1819 року її було перетворено на Київську духовну академію на території та в приміщеннях Києво-Могилянської академії, яка проіснувала до 1918 року.

Офіційне відродження Києво-Могилянської академії як сучасного університету почалося із здобуттям Україною незалежності. 19 вересня 1991 року розпорядженням Голови Верховної Ради України Києво-Могилянська академія відроджена на своїй історичній території як незалежний вищий навчальний заклад України - Університет «Києво-Могилянська академія». Перший набір студентів відбувся у вересні 1992 року. Навчальні програми та плани УКМА були схвалені на спеціальній нараді експертів ЮНЕСКО (22-23 травня 1993 року). Статусу Національного Університету набув 1994 року.

Державний Національний університет «Києво-Могилянська академія» має IV рівень акредитації.

2010 року згідно з рейтингом кращих ВНЗ України, складеному з урахуванням оцінок кращих роботодавців (журнал «Корреспондент» от 21 мая 2010 года), НаУКМА посів 2-е місце.

У НаУКМА надають сучасну освіту шляхом вільного творчого навчання та впровадження наукової діяльності відповідно до суспільних потреб, зумовлених розвитком української держави, науки, економіки та культури, а також глобальними процесами розвитку людської цивілізації.

НаУКМА вперше в Україні запровадив: бакалаврські програми (1992); вільну траєкторію навчання - формування студентом власного навчального плану (Liberal Arts Education - 1992); 100-бальну кредитно-рейтингову систему оцінювання знань (1992); триместрову систему організації навчального процесу (1992); власний університетський диплом; можливість вступу на дві спеціальності; магістерські дворічні програми; докторські програми західного зразка (станом на 2009 р. це перші п'ять PhD-програм у межах першої в Україні чотирирічної Докторської школи, без державного фінансування); спеціальності та унікальні школи в співпраці з провідними західними університетами: соціальна робота, екологія, охорона громадського здоров'я, бізнес школа, школа журналістики. Єдині в Україні Центр мембранних технологій, Лабораторія генетики та клітинної біології.

## Рисунок 3.3. Макет вебсайту ЄКТС

### 3.3. Розробка клієнтської частини сайту

Розробка клієнтської частини сайту відбувалась з використанням мови сценаріїв JavaScript та фреймворку React.js, для якого створена велика кількість бібліотек, що дозволяють легко отримати потрібний функціонал. В процесі розробки було зроблено особливий акцент на створенні гнучких компонентів, що легко можуть бути повторно використанні, та які не є надто складними для розуміння.

Так, за допомогою бібліотеки React Router [13] було створено наступні основні шляхи (routes) для навігації по сайту (див. рис. 3.4.):

- general-info/ – для перегляду категорії «Загальна інформація», що містить опис закладу та загальну інформацію про нього: присвоєння кредитів ЄКТС, консультаційну допомогу, і так далі;
- for-students/ – для перегляду категорії «Інформація для студентів», що містить таку інформацію для студентів, як інформацію про гуртожитки, харчування, тощо;

- about-programs/ – для перегляду категорії «Інформація по програмах», що містить перелік усіх факультетів, освітніх програм, та актуальну інформацію щодо дисциплін, які вивчаються на кожній з цих програм.

```

<Router>
  <div className="App">
    <Switch>
      <Route path="/" exact>{<Redirect to="/general-info" />}</Route>
      <Route path='/general-info' component={GeneralInfo}/>
      <Route path='/for-students' component={StudentsInfo}/>
      <Route path='/about-programs' component={ProgramsInfo}/>
    </Switch>
  </div>
</Router>

```

**Рисунок 3.4.** Приклад створення шляхів для навігації по сайту

Наразі інформація з усіх трьох категорій є статичною, оскільки на поточному вебсайті для інформаційного пакету ЄКТС вона також подана статично. Поточний сайт для пакету ЄКТС зберігає статті у базі даних MySQL, адже саме її було підключено при початковому налаштуванні CMS «Joomla!». Такі статті можуть редагуватися через панель адміністратора, але оскільки питання можливості редагування інформації з сайту для пакету ЄКТС наразі не є важливим і така задача не була поставлена, в оновленому додатку для інформаційного пакету ЄКТС дані статті зберігаються статично у JavaScript-об'єкті у вигляді HTML-коду, який React.js автоматично завантажує як тільки користувач обере категорію.

Список актуальних дисциплін, що пропонуються до вивчення на конкретній освітній програмі, знаходиться в університетській базі даних «Оптіма», і саме звідти поточний сайт для інформаційного пакету ЄКТС отримує цей список. Оновлений сайт отримує цей список від сервера, що, в свою чергу, завантажує його з бази даних «Оптіма».

Для відправлення запитів на сервер використовується бібліотека Axios [14], що дозволяє легко виконувати запити до серверу та має вбудований захист від деяких видів кібератак.

Запит до серверу слід виконувати після того, як компонент було вбудовано в DOM, отже метод `componentDidMount()`, що належить до методів життєвого циклу компонента та викликається відразу після того, як компонент вбудовується в DOM, цілком підходить для виконання запитів до серверу (див. рис. 3.5.).

```
componentDidMount() {
  axios.get('http://localhost:3000/courses-info')
    .then(response => {
      this.setState({courses: response.data});
    })
    .catch(error => {
      console.log(error);
    });
}
```

**Рисунок 3.5.** Виконання запиту до серверу з використанням бібліотеки Axios

Оскільки було неможливо займатися одночасно розробкою клієнтської частини та розробкою серверу, в певний момент часу знадобилося тестування створеного за допомогою React.js застосунку за відсутності серверу. В цьому випадку став у нагоді npm-пакунок `json-server` [15]. Даний пакунок дозволяє швидко та з мінімумом зусиль створювати простий штучний REST API, маючи лише файл з даними, на яких буде тестуватися клієнт, у форматі JSON. Запустити даний тестовий сервер можна з командного рядку, ввівши команду «`json-server`» та передавши їй вищезгаданий файл з даними у якості параметру (див. рис. 3.6.).

```
C:\Users\Marie\Desktop\coursach\client>json-server test-data.json

\{^_^}/ hi!

Loading test-data.json
Done

Resources
http://localhost:3000/general-info

Home
http://localhost:3000
```

**Рисунок 3.6.** Запуск тестового серверу з тестовими даними

Таким чином, клієнтська частина оновленого сайту є сучасною, досить простою та гнучкою у використанні, що забезпечується засобами фреймворку React.js. Оновлення та технологічна підтримка такого застосунку буде проходити набагато легше, ніж оновлення та підтримка наразі використовуваної CMS «Joomla!».

### **3.4. Розробка серверної частини сайту**

Сервер для отримання з університетської бази даних «Оптіма» актуального списку дисциплін, що вивчаються на різних освітніх програмах, було реалізовано за допомогою технології Node.js та фреймворку для створення серверів Express.js. Для перегляду та доступу до бази даних «Оптіма» використовувалася утиліта SQL Server Management Studio, створена компанією Microsoft.

За замовчуванням доступ до серверу університетської бази даних «Оптіма» відкритий лише з локальної мережі НаУКМА, тому для зручності роботи та можливості віддаленого доступу до бази даних «Оптіма» було налагоджене VPN-з'єднання з використанням засобів операційної системи Windows. Втім, в такого підходу є один недолік, що полягає у відсутності можливості доступу до мережі Інтернет. Щоб уникнути цієї незручності, було застосовано концепцію роздільного тунелювання (split tunneling) [16], що дозволяє одночасно мати доступ як до мережі Інтернет, так і віртуальної приватної мережі. Таким чином, під час перегляду змісту бази даних «Оптіма» можна було також доступатися до ресурсів, наданих мережею Інтернет.

Для того, щоб сервер міг під'єднатися до бази даних та надсилати їй SQL-запити на отримання даних, необхідно встановити та налаштувати драйвер для відповідної СКБД. В даному випадку, виникла необхідність у використанні драйвера для Microsoft SQL Server.

Оскільки за використання технології Node.js процесом встановлення драйверів (які є також npm-пакунками) керує менеджер пакетів npm,

встановлення драйвера відбувається доволі швидко. Після того, як прт покаже повідомлення про успішне встановлення драйвера, драйвер готовий до використання та просто включається у потрібний модуль застосунку за допомогою службової функції `require()`, що служить для імпорту потрібних бібліотек та фреймворків (див. рис. 3.7.).

```
const sql = require("mssql/msnodesqlv8");
```

**Рисунок 3.7.** Імпорт драйверу для Microsoft SQL Server

Сервер формує запит до бази даних тоді, коли надходить відповідний запит до самого сервера, наприклад, запит з клієнтської частини застосунку. Запити до бази даних сервер надсилає мовою SQL після того, як драйвер здійснить успішне з'єднання з базою даних (див. рис. 3.8.).

```
app.get("/courses", function(req , res){
    sql.connect(config, function (err) {
        if (err) console.log(err);

        var request = new sql.Request();
        var querySQL = 'select * from View_naukma_site_opis_ects';

        request.query(querySQL, function (err, recordset) {
            if (err) console.log(err)
            res.send(recordset);
        });
    });
});
```

**Рисунок 3.8.** Встановлення зв'язку з базою даних та відправка SQL-запиту до неї

Таким чином, створений сервер для оновленого вебсайту для інформаційного пакету ЄКТС виконує роль посередника між клієнтською частиною, розробленою за допомогою фреймворку React.js, та університетською базою даних «Оптіма», що функціонує на основі СКБД Microsoft SQL Server.

### 3.5. Ідеї для вдосконалення сайту

Хоча новий вебсайт для інформаційного пакету ЄКТС і є кращою версією поточного сайту, має простішу структуру та сучасніший інтерфейс користувача, простір для його вдосконалення все ж існує.

Інтерфейс користувача на новому сайті наразі є переосмисленим варіантом інтерфейсу користувача на поточному вебсайті для пакету ЄКТС. Це не є погано, адже таке розміщення елементів управління є інтуїтивно зрозумілим, до того ж, звичним для користувачів сайту та адміністрації, проте його зовнішній вигляд може бути покращений. Шляхом дослідження сучасних тенденцій веб-дизайну та існуючих рішень з оформлення подібних сайтів, можна розробити оновлений інтерфейс користувача, що буде більше відповідати сучасності та буде ще більш дружнім.

Аналогічно, оновлену версію інтерфейсу можна розробити і для мобільних пристроїв. Такий підхід вимагатиме створення окремого макету інтерфейсу користувача, адже мобільна версія застосунку має дійсно відрізнитися від основної. Наприклад, список підкатегорій варто прибрати у бокове меню, аби він не заважав користувачеві переглядати текстовий зміст сайту. Подібні рішення вимагають додаткових зусиль від програміста, але роблять значно комфортнішим процес використання застосунку незалежно від вибраного пристрою.

Наразі статті, що містяться на новому сайті для інформаційного пакету ЄКТС, можуть бути відредаговані тільки у вихідному коді клієнтського застосунку, тобто панель адміністратора та інтерактивна система редагування розділів цілком відсутня. Завдання створення панелі адміністратора не входило у дану роботу, втім, залежно від відповідного завдання ІОЦ, може виникнути необхідність додати її у майбутньому, аби адміністрація сайту НаУКМА могла за потреби оновлювати статті, що знаходяться на сайті для пакету ЄКТС. Поява на сайті подібного функціоналу зробить його гідною альтернативою поточному рішенню, створеному на основі CMS «Joomla!».

### **3.6. Висновки**

У даному розділ був описаний процес проєктування та створення нового вебсайту для інформаційного пакету ЄКТС. Проєктування структури та інтерфейсу користувача було здійснене за допомогою веб-застосунків Figma та Lucidchart. Клієнтська частина застосунку була розроблена за допомогою JavaScript фреймворку React, в той час як у розробці серверу були задіяні технології Node.js та Express.



## Висновки

У ході виконання курсової роботи «Побудова мережі мікросервісів для розміщення публічної інформації ЗВО» було виконано наступні *завдання*:

- проаналізовано поточну структуру офіційного вебсайту Національного університету «Києво-Могилянська академія», визначено переваги та недоліки даної структури;
- вивчено такі технології як фреймворк React.js, що дозволяє створення гнучких компонентів, готових до повторного використання, Node.js та фреймворк Express.js, що дозволяє легко створювати веб-сервери;
- розроблено веб-застосунок для розміщення публічної інформації у вигляді інформаційного пакету ЄКТС, що замінить собою поточне застаріле рішення, розроблене за допомогою CMS «Joomla!».

Вирішені наступні *цілі*:

- створено оновлену версію сайту для інформаційного пакету ЄКТС, що відповідає усім сучасним стандартам веб-розробки.

Серед основних недоліків офіційного вебсайту НаУКМА можна назвати власноруч написані плагіни, що порушують нормальний цикл життя сайту, та складну структуру, що значно ускладнює процес його оновлення. Одним з елементів, що формують таку структуру, є вбудований вебсайт для інформаційного пакету ЄКТС.

Для вирішення цієї проблеми було вирішено відокремити даний сайт та зробити його самостійним мікросервісом. Для досягнення цієї мети було застосовано такі сучасні технології як React.js (з боку клієнта) та Express.js (з боку сервера).

Хоча новий сайт і виконаний з урахуванням усіх недоліків свого попередника, він також має певний простір для вдосконалень. Серед можливих

майбутніх оновлень можна назвати створення мобільної версії застосунку та панелі адміністратора для редагування текстового змісту сайту.

Очікується, що створений для інформаційного пакету ЄКТС вебсайт стане достойною заміною рішення, що використовується зараз.

## Список літератури

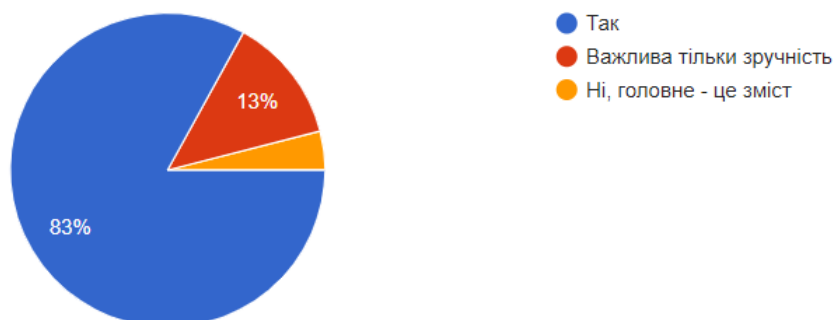
1. Phoca Gallery – розширення для CMS «Joomla!» [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.phoca.cz/phocagallery>
2. Закон України «Про освіту» від 5 вересня 2017 року № 2145-VIII. - Ст. 3. [Електронний ресурс]. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2145-19>
3. DOCman – розширення для CMS «Joomla!» [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.joomlatools.com/extensions/docman/>
4. Стандартний компонент CMS «Joomla!» для керування контактами [Електронний ресурс]. – Режим доступу до ресурсу: [https://docs.joomla.org/Help39:Components\\_Contact\\_Manager\\_Options](https://docs.joomla.org/Help39:Components_Contact_Manager_Options)
5. Векторний графічний онлайн-редактор Figma [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.figma.com/>
6. Мова розмітки HTML [Електронний ресурс]. – Режим доступу до ресурсу: [https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)
7. Less – a backwards-compatible language extension for CSS [Електронний ресурс]. – Режим доступу до ресурсу: <http://lesscss.org/>
8. React – a JavaScript library for building user interfaces [Електронний ресурс]. – Режим доступу до ресурсу: <https://reactjs.org/>
9. Технологія Node.js [Електронний ресурс]. – Режим доступу до ресурсу: <https://nodejs.org/en/>
10. Фреймворк Express.js [Електронний ресурс]. – Режим доступу до ресурсу: <https://expressjs.com/>
11. sql-server-2017 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2017>.

- 12.Правила використання символіки НаУКМА (Brandbook) [Електронний ресурс]. – Режим доступу до ресурсу: [http://ekmair.ukma.edu.ua/bitstream/handle/123456789/10634/NAUKMA\\_Brand\\_Book.pdf?sequence=3&isAllowed=y](http://ekmair.ukma.edu.ua/bitstream/handle/123456789/10634/NAUKMA_Brand_Book.pdf?sequence=3&isAllowed=y)
- 13.Бібліотека React Router [Електронний ресурс]. – Режим доступу до ресурсу: <https://reacttraining.com/react-router/web/guides/quick-start>
- 14.Бібліотека Axios [Електронний ресурс]. – Режим доступу до ресурсу: <https://alligator.io/react/axios-react/>
- 15.json-server [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.journaldev.com/10660/json-server>
- 16.Split tunneling [Електронний ресурс]. – Режим доступу до ресурсу: [https://www.webopedia.com/TERM/S/split\\_tunneling.html](https://www.webopedia.com/TERM/S/split_tunneling.html)

## Додаток. Результати опитування студентів НаУКМА стосовно поінформованості щодо пакету ЄКТС

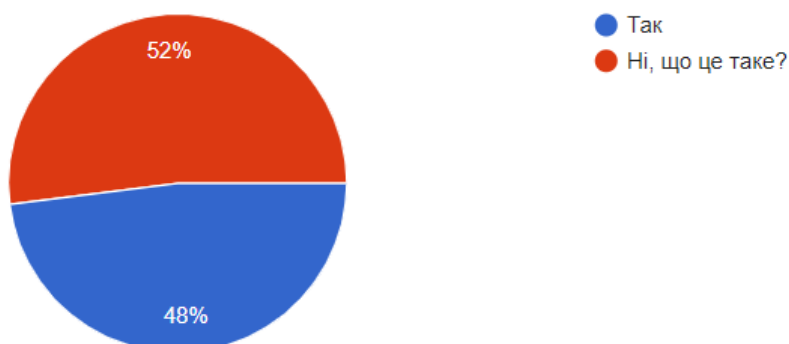
Якщо ви використовуєте сайт (будь-який) як джерело інформації, чи вам важливо, щоб його інтерфейс був красивим та зручним?

100 ответов



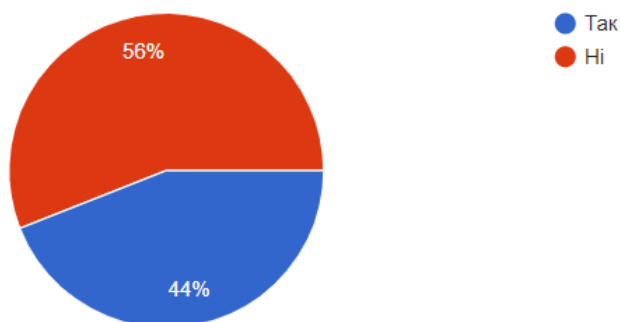
Чи ви знаєте, що таке інформаційний пакет ЄКТС?

100 ответов



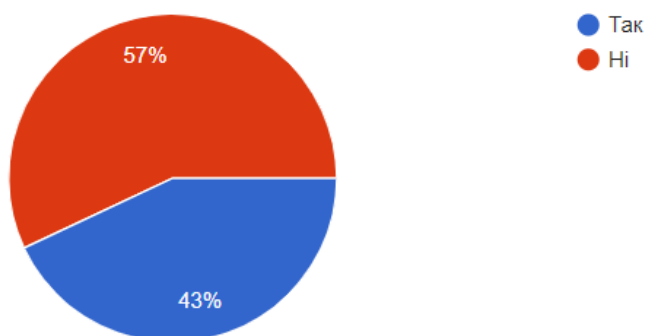
Чи назва "Інформаційний пакет ЄКТС" говорить вам про щось та є для вас зрозумілою?

100 ответов



Чи ви знаєте, де можна знайти інформаційний пакет ЄКТС НаУКМА?

100 ответов



Якщо ви знаєте що таке інформаційний пакет ЄКТС, то чи користуєтеся ви ним?

43 ответа

