

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

## **Розробка користувацького інтерфейсу веб- застосунку системи “SmartUkma”**

**Текстова частина до курсової роботи за спеціальністю «Інженерія  
програмного забезпечення» 121**

Керівник курсової роботи  
Асистент,  
Корнійчук М. А.

Виконав студент  
Ксьондзик М.Ю.

Київ – 2022

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ О. П. Жежерун  
(підпис)

«\_\_» \_\_\_\_\_ 2022р.

## ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту Ксьондзик М.Ю. факультету інформатики 4-го курсу

ТЕМА: Розробка користувацького інтерфейсу веб-застосунку системи  
«SmartUkma»

Зміст ТЧ до курсової роботи:

- Індивідуальне завдання
- Календарний план
- Вступ
  - Аналіз предметної області. Постановка завдання курсової роботи
  - Теоретичні відомості
  - Опис реалізації програмного продукту
- Висновки
- Перелік використаних джерел
- Додатки

Дата видачі «\_\_» \_\_\_\_\_ 2022р. Керівник \_\_\_\_\_

(підпис)

Завдання отримав \_\_\_\_\_ (підпис)

# Календарний план виконання курсової роботи

Тема: Веб-сайт для продажу квартири та нежитлових приміщень у новобудові

## Календарний план виконання роботи:

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1	Отримання завдання на курсову роботу.	16.09.2021	
2	Аналіз сучасного стану питання	21.09.2022	
3	Початок написання текстової частини	01.01.2022	
4	Огляд систем аналогів	05.01.2022	
5	Дослідження технологій для веб-розробки	05.02.2022	
6	Складання плану роботи та формулювання технічних вимог до системи	15.02.2022	
7	Початок написання практичної частини	20.02.2022	
8	Надання проміжної версії текстової частини	09.05.2022	
9	Надання проміжної версії практичної частини	12.05.2022	
10	Тестування практичної частини	14.05.2022	
11	Остаточне завершення написання практичної частини	15.05.2022	
12	Підготування презентації	16.05.2022	
13	Остаточне завершення написання текстової частини	17.05.2022	

Студент Ксьондзик М. Ю.

Керівник Корнійчук М. А.

« \_\_\_\_ » \_\_\_\_\_

# Зміст

.....	2
<b>Календарний план виконання курсової роботи .....</b>	<b>3</b>
<b>Зміст .....</b>	<b>4</b>
<b>Перелік термінів та умовних позначень .....</b>	<b>5</b>
<b>Вступ .....</b>	<b>6</b>
<b>РОЗДІЛ 1 Аналіз предметної області. Постановка завдання курсової роботи.....</b>	<b>8</b>
1.1 Аналіз сучасного стану питання та обґрунтування теми .....	8
1.3 Постановка завдання .....	9
<b>РОЗДІЛ 2 Теоретичні відомості.....</b>	<b>10</b>
<b>РОЗДІЛ 3 Опис реалізації програмного продукту .....</b>	<b>13</b>
3.1 Аналіз технічного завдання .....	13
3.2 Опис процесу роботи над застосунком .....	13
3.3 Створення об'єктів і розробка головної програми .....	15
3.4 Опис інтерфейсу програми .....	16
3.5 Тестування програми і результати її виконання .....	22
<b>Висновки .....</b>	<b>25</b>
<b>Список використаних джерел .....</b>	<b>26</b>
<b>Додатки.....</b>	<b>27</b>
Додаток А – список ілюстрацій.....	27
Додаток Б – програмний код.....	28

# Перелік термінів та умовних позначень

- стек технологій – список, набір технологій, що використовуються в сукупності для розробки.
- IDE – Integrated Development Environment (середовище розробки).
- Футер – компонент, що розташовується у самому низу веб-сторінки.
- Фреймворк – програмний каркас, інфраструктура програмних рішень, що полегшує розробку складних систем
- Бекенд – частина застосунку, що недоступна користувачу, серверна частина.
- Спрінт – короткий період, у який команда працює над задачею
- Ревью – перегляд

# Вступ

Розробка зручного інтерфейсу для сервісу запису на курсові – надважлива задача, адже це перше що бачить студент, коли розпочинається його досвід із написання курсової. Необхідно зробити сайт інтуїтивно зрозумілим, простим, приємним на вигляд та адаптованим для мобільних пристроїв, адже за статистикою мобільний трафік склав близько 55% від загального станом на кінець 2021 року [1]. Проте інтерфейс не існує в окремому абстрактному світі, це всього лише частина готового продукту, а у випадку системи запису на курсові, яка має витримувати великий трафік у пікові моменти навантажень, зберігати велику кількість інформації та тримати її надійно захищеною, роль інтерфейсу часто применшується. Проект SmartUkma вже був у розробці до початку роботи над ним у рамках цієї курсової. На момент запису на курсові роботи, система вже була задіяна для виконання даної задачі. Студенти мали записатися на курсові через наявний на той момент сайт. У процесі запису, викрилося багато проблем і студенти висловлювали своє невдоволення системою. Переважна більшість негативу стосувалася саме інтерфейсу. З мобільного телефону здійснити запис було надзвичайно складно. Щоби отримати доступ до потрібних елементів інтерфейсу було необхідно користуватися сайтом у горизонтальній орієнтації девайсу, а якщо екран пристрою був меншим за 5 дюймів, то запис ставав просто неможливим, адже кнопка логіну не показувалася і усе що бачив студент, було біле полотно.

Задачею курсової роботи є розробка користувацького інтерфейсу веб-застосунку системи “SmartUkma”.

Метою є автоматизація та централізація процесу запису на курсові, а також спрощення запису для студентів та покращення користувацького досвіду.

Об'єктом дослідження є процес розробки користувацького інтерфейсу сервісу запису на курсові в умовах наявної системи, досвіду користувачів та працюючи командою у форматі щотижневих спринтів.

Предметом дослідження є використання наявного стеку технологій веб-програмування із фреймворком Angular в основі для розробки сайту запису на курсові.

Процес розробки також включав роботу багатьох членів команди для дослідження потреб користувачів, тестування макетів дизайну для визначення найкращої комбінації розташування елементів, їхнього розміру, форми, ступені виділення з загального інтерфейсу. Розроблену систему можна використовувати для запису на курсові та як зразок успішної імплементації системи автоматизації в організаційну структуру університету та навчального процесу загалом.

Для розробки було використано наступне програмне забезпечення:

- Система контролю версій GitLab;
- Середовище розробки IntelliJ Idea
- Брокер повідомлень ActiveMQ
- Система комунікації Microsoft Teams
- Редактор векторної графіки Figma

Загальна структура роботи складається зі вступу, трьох розділів: аналіз предметної області та постановка завдання курсової роботи, теоретичні відомості, опис реалізації програмного продукту. А також висновків, списку використаних джерел і додатків.

# **РОЗДІЛ 1 Аналіз предметної області.**

## **Постановка завдання курсової роботи**

### **1.1 Аналіз сучасного стану питання та обґрунтування теми**

Щороку відбувається запис на курсові і студенти змушені переглядати незручну Excel-табличку у пошуках тем, а потім писати індивідуально кожному викладачу щоби дізнатися чи ця тема актуальна, чи ніхто її ще не обрав та які вимоги до написання. Цей процес є сповненим стресу та стану непевності, у тому числі внаслідок довготривалого очікування відповіді від викладачів. Відповідно, дана система є вкрай актуальною та покликана позбути студентів від багатьох незручностей у процесі запису на курсові. На жаль, після початкового запуску, не зважаючи на, загалом, функціонуючу систему, студенти не були задоволені такою заміною традиційного способу запису, не зважаючи на усі його недоліки. Тобто дана система, що мала покращити досвід користувачів, викликала зворотній ефект, хоча абсолютна більшість студентів підтримала загальну ідею продукту. Переважна більшість скарг були якраз щодо незручності, недоступності та незрозумілості інтерфейсу системи. На жаль, даний аспект продукту часто нехтується, якщо в команді розробки відсутні дизайнери та фронтенд-розробники, адже тоді головне щоби основна функціональність просто працювала з точки зору поставлених задач, а не щоби потреби користувачів були задоволені.



### 1.3 Постановка завдання

В першу чергу необхідно проаналізувати наявну систему, як вона влаштована та написати документацію по ній. Також запустити проект на локальній машині та задокументувати процес і проблеми, які виникали. Паралельно із цим необхідно дослідити предметну область, у цілях розуміння, яку функціональність додати, яку видалити, а яку залишити та, можливо, покращити. Також необхідно зібрати та проаналізувати відгуки користувачів щодо наявного інтерфейсу задля розуміння слабких та сильних сторін існуючого продукту. Потрібно залучити методистів для отримання важливої інформації щодо процесу запису на курсові, зокрема, яка функціональність важлива для викладачів. Також необхідно підключити дизайнера до процесу розробки. Після отримання інформації від методистів, викладачів та студентів, можна сформуванати початковий набір елементів, що будуть розташовані на сторінці. Дану інформацію потрібно передати дизайнеру, щоби він сформував початковий макет. Надалі відбуватиметься процес обговорення та аналізу макету командою. Через декілька ітерацій матимемо макет, який задовольняє усіх. Тоді можна приступити до розробки інтерактивного макету та передачі його групі людей, що представляють цільову аудиторію продукту, а саме – студентам. Далі отримуємо відгуки від користувачів, обговорюємо, дизайнер дороблює макет і можна приступати до розробки. Розробку варто проводити за принципом mobile-first щоби забезпечити адаптивне відображення на мобільних пристроях.

## РОЗДІЛ 2 Теоретичні відомості.

Існує безліч методів та підходів для розв'язання даної задачі. На момент початку написання вже були використані певні технології. Зокрема, бекенд був написаний на мові Java, для фронтенду була використана мова JavaScript та фреймворк Angular.

Фреймворк – платформа для розробки програм. Фреймворки зазвичай містять заздалегідь визначені класи та функції. Оскільки розробляють їх зазвичай кваліфіковані фахівці, фреймворки є надійною та ефективною основою для написання програм, економлячи час та зусилля. При їх застосуванні, можна зосередитися виключно на високорівневій розробці, а зекономлений час зосередити на інших аспектах проекту. Найбільш розповсюдженими фреймворками на сьогодні є Angular, React та Vue. Вибір часто зводиться до персональних вподобань, адже кожен із них майже однаково добре впорається із поставленою задачею [2,3,4].

Для back-end частини часто використовуються мови програмування Java, JavaScript, Python, Ruby та інші. Вибір конкретної залежить від проекту. Часто це є JavaScript, адже на ній пишуть і front-end частину. Також для цієї мови створено багато різноманітних бібліотек та фреймворків, що значно спрощують розробку. Мова Python також має велику кількість бібліотек та фреймворків, але мінусом є те, що вона сильно залежить від них. Java буде хорошим вибором в умовах розробки великих проектів. Вона потребує доволі багато ресурсів для роботи і тому краще працює на дорогих системах. Ruby має плюси у вигляді мета-програмування та швидкості розробки, проте сама по собі є не дуже швидкою і має невелику кількість бібліотек у порівнянні з іншими мовами. [5]

Існує багато різних середовищ розробки. Вибір конкретного залежить від обраної мови, стеку технологій та власних вподобань. Вибір IDE може здатися

неважливим, проте це програма, в якій потрібно буде працювати над проектом і цей процес має бути максимально комфортним та приємним, відповідно варто приділити особливу увагу тому, чи подобається IDE та чи підходить вона особисто Вам. Серед найпопулярніших IDE на сьогодні є такі варіанти:

Visual Studio Code – одна з найкращих IDE для мови JavaScript із вбудованою підтримкою Node.js, JavaScript, TypeScript. Також є розширення для інших мов, таких як Python, PHP, Ruby та C#. Visual Studio Code це досить проста IDE, яка добре підходить для людей із невеликим досвідом розробки, оскільки пояснює теги HTML, обробку помилок, синтаксис та інші аспекти. Зважаючи на велику кількість переваг, це все ж open-source проект, тож дана IDE безкоштовна.

Sublime Text 3 – гнучка та безкоштовна IDE, що підтримує багато різних мов, зокрема C, Python, JavaScript. Також вона підтримує мову розмітки HTML на стилі CSS. Має швидкий та простий інтерфейс. Підтримує множинні виділення, редагування у режимі розділеного екрану та автозавершення коду.

PyCharm – IDE із підтримкою різних мов, зокрема Python, JavaScript, середовище виконання Node.js, стилі CSS та мову розмітки HTML. Хоча програма безкоштовна, існує платна версія, що більш надійна та стабільна. Безкоштовна версія PyCharm має багато багів, особливо це стосується автозавершення коду.

IntelliJ IDEA – java-центрична IDE, що є однією з найкращих у своєму сегменті. Має можливість автоматично додавати інструменти та бібліотеки залежно від контексту. Окрім Java, це середовище розробки також підтримує JavaScript, Ruby, PHP, Python, HTML, CSS, та багато інших. Загалом є безкоштовною, проте існує платна версія з покращеним функціоналом та більшою кількістю інструментів для розробки.

WebStorm – IDE, що була розроблена спеціально для створення веб-застосунків. Підтримує JavaScript, CSS, HTML та інші. Також має підтримку різноманітних бібліотек та фреймворків, наприклад React та Angular.

PHPStorm – чудово підходить для розробки із PHP-фреймворками, такими як WordPress, Drupal, Magento та іншими. Підтримує HTML, CSS, JavaScript та інші.

Загалом усі середовища розробки, створені IntelliJ, а до таких відносяться: IntelliJ IDEA, WebStorm, PyCharm та PHPStorm, мають ряд спільних переваг, таких як автоматичний рефакторинг, інтеграція Unit-тестів, інтеграція із системою керування версіями Git, розумне автозавершення коду, зручна навігація та приємний дизайн [6].

# РОЗДІЛ 3 Опис реалізації програмного продукту

## 3.1 Аналіз технічного завдання

Для розробки сайту було обрано середовище розробки IntelliJ Idea, адже бекенд написаний на Java, середовище підтримує CSS, HTML та Javascript, має зручну функціональність, інтеграцію із системою контролю версій Git. Також дана IDE є безкоштовною завдяки корпоративному акаунту НаУКМА та часто мною використовувалася протягом навчання.

Для забезпечення гарного відображення використаємо стилі CSS, а для кращої та простішої інтеграції складної логіки на сторінку – фреймворк Angular, який вже використовувався у цьому проєкті раніше.

## 3.2 Опис процесу роботи над застосунком

Робота велася у форматі спрінтів зі щотижневими ревью, проміжними зустрічами для обговорення результатів та командними зустрічами для постановки цілей та задач на тиждень. Напочатку, основним завданням було задокументувати увесь проєкт, розписати структуру та за що відповідає кожен із компонентів. Надалі задачею було запуснути сервіси, важливі для розробки, зокрема, для роботи ui-server, в рамках якого проходила розробка для даної курсової, необхідно було запуснути ще 3 додаткові сервіси: user-info, account-server та course-work. Запуск сервісів був однією з найважчих частин, на яку було витрачено декілька тижнів. Необхідно було встановити додаткові залежності, оновити існуючі, підібрати компоненти, що працюватимуть на конкретному комп'ютері, на якому проводиться розробка. Відмінність початкового середовища розробки від інших, що різнилися для кожного розробника, спричинили велику затримку, бо після кожного оновлення

локальної версії проекту чи вивантаження змін на GitLab, виникали конфлікти та нові помилки, які необхідно було вирішити. В умовах коли на роботу над проектом не можна було приділити увесь час, а цілі ставилися кожного тижня, довелося відмовитися від постійного оновлення локальної та основної версії застосунку задля економії часу, що зробило проект фрагментованим, а конфлікти важчими для вирішення. Часто після годинних спроб виправити помилки було простіше вручну скопіювати змінені файли та вставити у чистий застосунок.

Паралельно із запуском сервісів та написанням документації, проводилися аналіз та дослідження потреб користувачів, у результаті чого було отримано інформацію, на основі якої сформовано основну функціональність. Поступово йшла робота по виправленню вже наявних проблем із дизайном та впроваджувався принцип mobile first. З отриманих даних був розроблений макет сайту, після чого розпочалася верстка. Проблема полягала у тому, що мобільної версії макету не було, а сам дизайн доволі сильно відрізнявся від початкового, внаслідок цього довелося відкинути мобільно центричність застосунку та зосередитися на повному переробленні наявних сторінок. Розробка рухалася доволі повільно та важко. Пришвидшити процес вдалося відмовою працювати із наявними елементами та бібліотеками, замінивши їх на базові, які було легше налаштувати під шаблон та поєднати із логікою на Ангулярі.

### 3.3 Створення об'єктів і розробка головної програми

Файлова структура проекту зображена на рисунку 1. У кореневій папці проекту знаходяться папка `src`, у якій знаходиться сервер для роботи даного сервісу та папка `web-src`, де розташований фронтенд сервісу, у якій і проводилася розробка в рамках даної курсової.

Папка `src` у директорії `web-src` містить файл `index.html`, що є основною сторінкою застосунку, на яку виводяться усі компоненти. У папці `app` (див. Рисунок 2) знаходяться файли `app.component` із розширеннями:

`.html` – розмітка структури сторінки із панеллю меню, хедером та футером;

`.scss` – стилі для розмітки

`.ts` – логіка сторінки

Також у папці знаходиться `app.module.ts`, у якому визначаються компоненти сайту та імпортуються модулі, файл `app-routing.module.ts`, у якому прописані шляхи до різних компонентів програми.

У директорії `api` знаходиться функціональність, відповідальна за зв'язок із базою даних, а у папці `components`, власне, розташовані усі компоненти інтерфейсу із відповідними файлами розмітки,

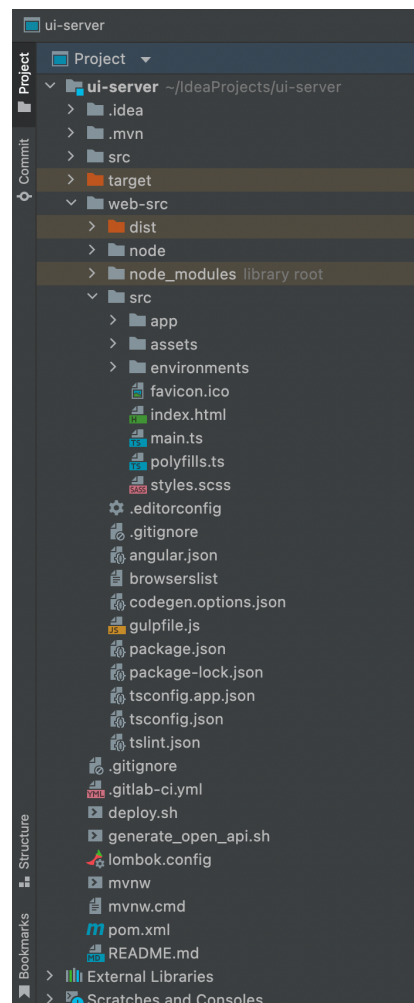


Рисунок 1 - файлова структура проекту

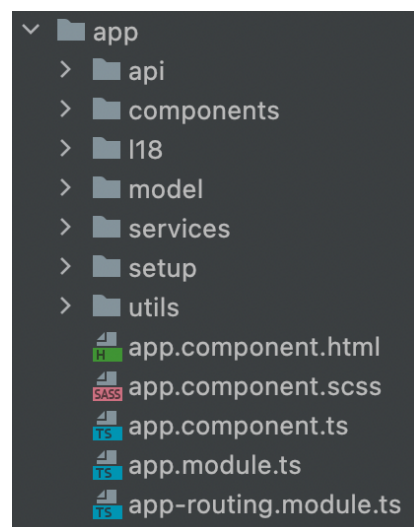


Рисунок 2 - вміст папки `app`

стилями та typescript файлами, відповідальними за логіку кожного індивідуального компоненту.

### 3.4 Опис інтерфейсу програми

Інтерфейс програми складається із трьох основних компонентів: хедер – інформаційна панель, основна частина, де розташовані основна функціональність, інформація та бокова панель навігації футер сторінки, на якому розташовані контакти та назва університету.

На навігаційній панелі (див. Рисунок 3) присутній блок курсових (при подальшому розвитку системи додаватимуться й інші блоки). Блок курсових можна згорнути, натиснувши на назву блоку (див. Рисунок 4)

Усю навігаційну панель можна зменшити у ширині, щоби залишилися тільки іконки блоків, натиснувши кнопку «Згорнути меню». Так само можна потім розгорнути меню, клікнувши на стрілочку (див. Рисунок 5)



Рисунок 3 - навігаційна панель



Рисунок 5 - зменшена навігаційна панель

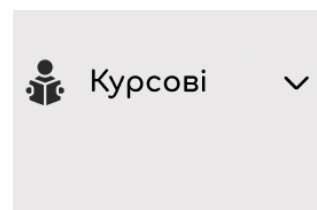


Рисунок 4 - зменшений блок курсових



На хедері сторінки присутній логотип, при натисканні на який, відбувається перехід на головну сторінку. Також назва сторінки, на якій знаходиться користувач (Головна, Запис на курсові...), ім'я та прізвище користувача та їхнє фото.



Рисунок 6 - хедер

На футері розташовується назва університету та посилання на соціальні мережі.

Національний університет "Кієво-Могилянська Академія"



Рисунок 7 - футер

Загалом, на головній сторінці розташоване повідомлення із дедлайном на задачу курсової роботи та блок із записом на курсові із посиланнями на загальний список усіх курсових робіт та на заявки на написання курсових робіт для поточного користувача.

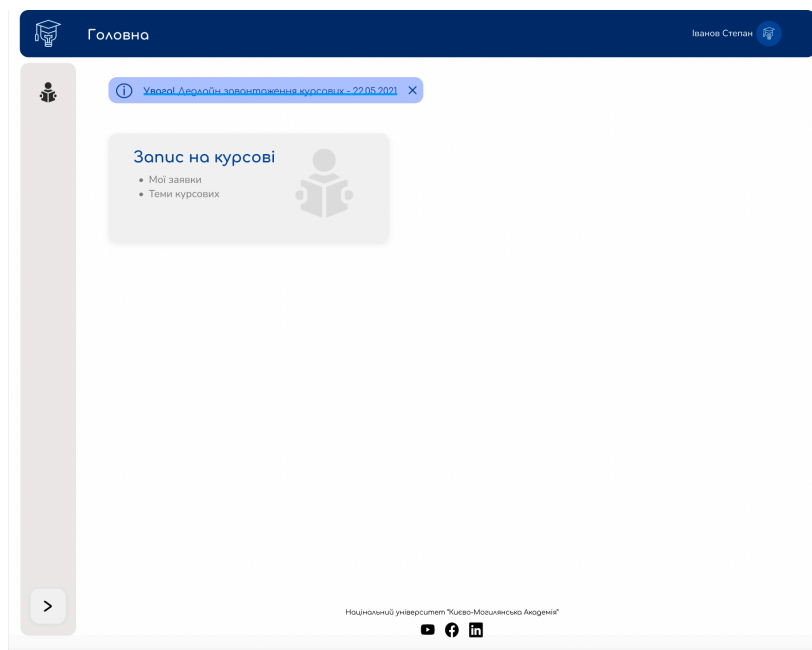


Рисунок 8 - головна сторінка

Сторінка логіну складається з тієї самої основи, проте компоненти користувача, та бокової панелі приховані внаслідок непройденої аутентифікації і видно тільки компоненту входу на сторінку за допомогою корпоративного акаунту Microsoft. Компонент складається з назви, кнопки входу та адреси підтримки, клікнувши на яку, користувачу автоматично відкривається поштовий клієнт, встановлений за замовчуванням у системі.

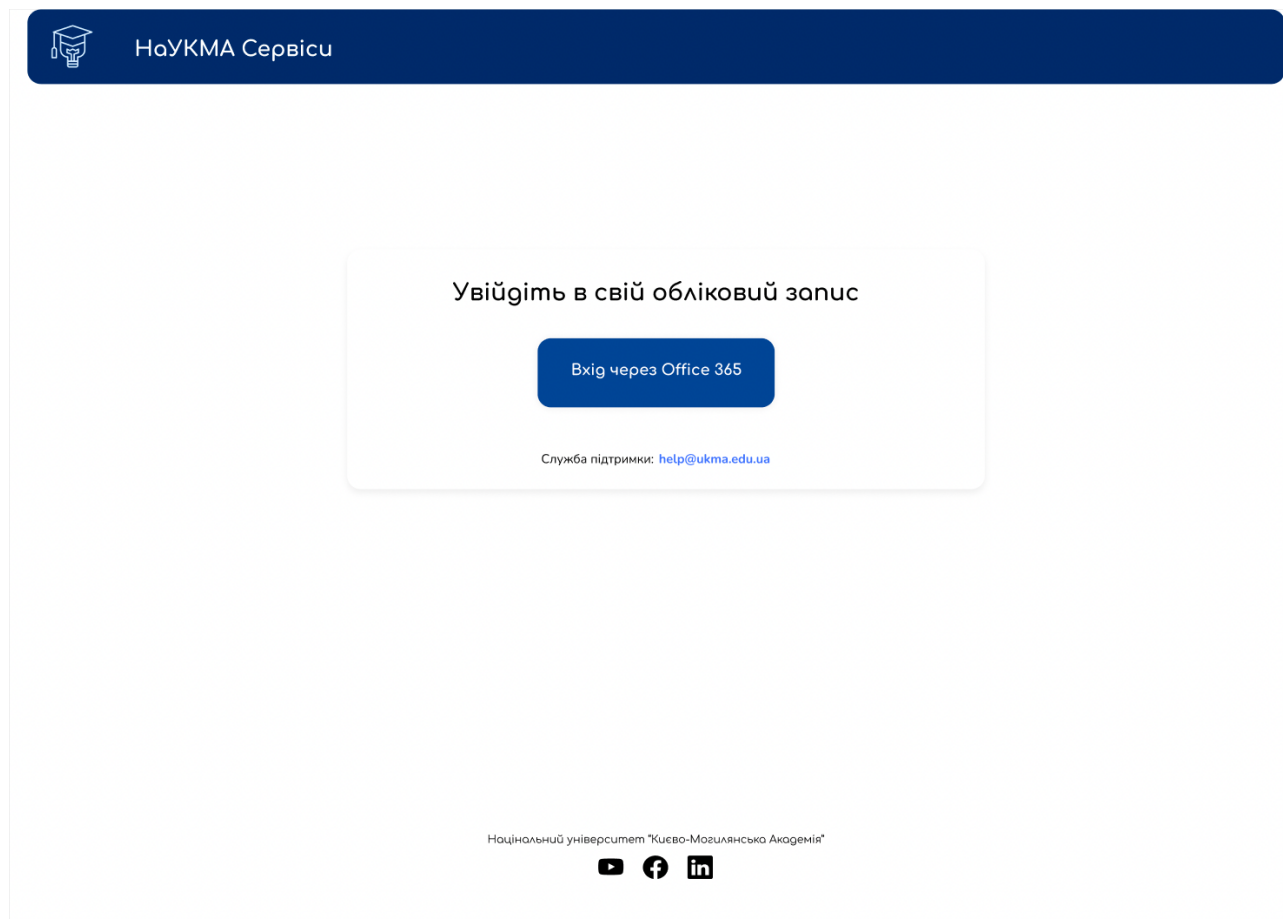


Рисунок 9 - сторінка логіну

Сторінка із заявками на написання курсових поточного користувача містить таблицю із колонкою тема, натиснувши на яку, можна відкрити сторінку курсової, викладач, Кафедра та статус, у якій указаний конкретний статус кожної із заявок (Заявка подана, недоступно, підтверджено):



Курсові ^  
 Мої заявки  
 Теми курсових

Тема	Викладач	Кафедра	Статус
Дослідження впливу мета-модерну в особливо великих кількостях на організм людини.	Подольський Є. С.	Кафедра мета-модерну та гарного настрою	Заявка подана
Дослідження впливу мета-модерну в особливо великих кількостях на організм людини.	Подольський Є. С.	Кафедра мета-модерну та гарного настрою	Недоступно
Дослідження впливу мета-модерну в особливо великих кількостях на організм людини.	Подольський Є. С.	Кафедра мета-модерну та гарного настрою	Підтверджено

< Згорнути меню

Національний університет "Кієво-Могилянська Академія"



Рисунок 10 - сторінка заявок на курсові

Сторінка із усіма темами на курсові містить таблицю із колонкою тема, натиснувши на яку, можна відкрити сторінку курсової, викладач, Кафедра та статус, у якій указаний конкретний статус кожної із заявок (Заявка подана, недоступно, підтверджено, подати заявку). Також на сторінці присутній блок із фільтрами. Серед колонок, за якими можлива фільтрація, присутні «Викладач», «Кафедра», «Навчальний рік», «Освітній рівень» та «Статус». Кожен із фільтрів, при застосуванні, додається до верхнього рядку із наявними фільтрами, а при видаленні – зникає. (див Рисунок)

**Запис на курсові**
Іванов Степан

**Курсові** ^

Мої заявки

Теми курсових

Подольський Є.С. × Кафедра мета-модерну та гарного настрою × 2020-2021 × БП-3 ×

Всього: 4   Вільно: 2   Зайнято: 2

Тема	Викладач	Кафедра	Статус
Дослідження впливу мета-модерну в особливо великих кількостях на організм людини.	Подольський Є. С.	Кафедра мета-модерну та гарного настрою	Заявка погодно
Дослідження впливу мета-модерну в особливо великих кількостях на організм людини.	Подольський Є. С.	Кафедра мета-модерну та гарного настрою	Негоступно
Дослідження впливу мета-модерну в особливо великих кількостях на організм людини.	Подольський Є. С.	Кафедра мета-модерну та гарного настрою	Підтверджено
Дослідження впливу мета-модерну в особливо великих кількостях на організм людини.	Подольський Є. С.	Кафедра мета-модерну та гарного настрою	Подати заявку

**Фільтри**

Викладач v

Кафедра v

Навчальний рік v

Освітній рівень v

Статус v

Виділити всі

← Згорнути меню

Національний університет "Києво-Могилянська Академія"

Рисунок 11 - сторінка тем курсових

Сторінка курсової містить назву та опис курсової роботи, додаткову інформацію по ній, а також поле, де студент пише свою мотивацію для написання та кнопку «Скасувати заявку» чи «Подати заявку»



## Дослідження впливу мета-модерну в великих кількостях на організм людини

### Очікує підтвердження

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in tortor ligula. Cras ullamcorper sem id vehicula gravida. In vel condimentum augue. Curabitur accumsan, ante at tempus porttitor, massa diam commodo nisi, quis semper enim massa at ante. Donec sollicitudin magna in ultrices mollis. Duis laoreet interdum ultricies. Cras semper lacinia leo. In mattis ipsum lacinia massa elementum, a gravida ipsum rhoncus.

Vivamus lacinia lorem in lorem semper, sed pellentesque mi iaculis. Morbi lobortis augue eget justo dignissim elefend. Cras mollis condimentum dui in feugiat. Praesent tempus ornare mauris et pretium. Phasellus eros neque, auctor eget gravida et, cursus sed urna. Duis vulputate ante sit amet condimentum scelerisque. Morbi eu ante cursus dui scelerisque vulputate non in nulla.

Подольський Егор Сергійович  
([igor.podolskyi@ukma.edu.ua](mailto:igor.podolskyi@ukma.edu.ua))  
Кафедра мета-модерну та гарного настрою  
2020-2021  
БІТ-3  
Остання зміна: 1:07 PM, 11/16/2021

### Мотивація

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in tortor ligula. Cras ullamcorper sem id vehicula gravida. In vel condimentum augue. Curabitur accumsan, ante at tempus porttitor, massa diam commodo nisi, quis semper enim massa at ante. Donec sollicitudin magna in ultrices mollis. Duis laoreet interdum ultricies. Cras semper lacinia leo. In mattis ipsum lacinia massa elementum, a gravida ipsum rhoncus.

Vivamus lacinia lorem in lorem semper, sed pellentesque mi iaculis. Morbi lobortis augue eget justo dignissim elefend. Cras mollis condimentum dui in feugiat. Praesent tempus ornare mauris et pretium. Phasellus eros neque, auctor eget gravida et, cursus sed urna. Duis vulputate ante sit amet condimentum scelerisque. Morbi eu ante cursus dui scelerisque vulputate non in nulla.

Скасувати заявку



Національний університет "Києво-Могилянська Академія"



## Дослідження впливу мета-модерну в великих кількостях на організм людини

### Можна подати заявку

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in tortor ligula. Cras ullamcorper sem id vehicula gravida. In vel condimentum augue. Curabitur accumsan, ante at tempus porttitor, massa diam commodo nisi, quis semper enim massa at ante. Donec sollicitudin magna in ultrices mollis. Duis laoreet interdum ultricies. Cras semper lacinia leo. In mattis ipsum lacinia massa elementum, a gravida ipsum rhoncus.

Vivamus lacinia lorem in lorem semper, sed pellentesque mi iaculis. Morbi lobortis augue eget justo dignissim elefend. Cras mollis condimentum dui in feugiat. Praesent tempus ornare mauris et pretium. Phasellus eros neque, auctor eget gravida et, cursus sed urna. Duis vulputate ante sit amet condimentum scelerisque. Morbi eu ante cursus dui scelerisque vulputate non in nulla.

Подольський Егор Сергійович  
([igor.podolskyi@ukma.edu.ua](mailto:igor.podolskyi@ukma.edu.ua))  
Кафедра мета-модерну та гарного настрою  
2020-2021  
БІТ-3  
Остання зміна: 1:07 PM, 11/16/2021

### Мотивація

Тезово опишіть вашу мотивацію

Подати заявку



Національний університет "Києво-Могилянська Академія"



Рисунок 12 - сторінка курсової

### 3.5 Тестування програми і результати її виконання

Інструкція типового сценарію використання.

Заходимо на сайт (див. **Error! Reference source not found.**).

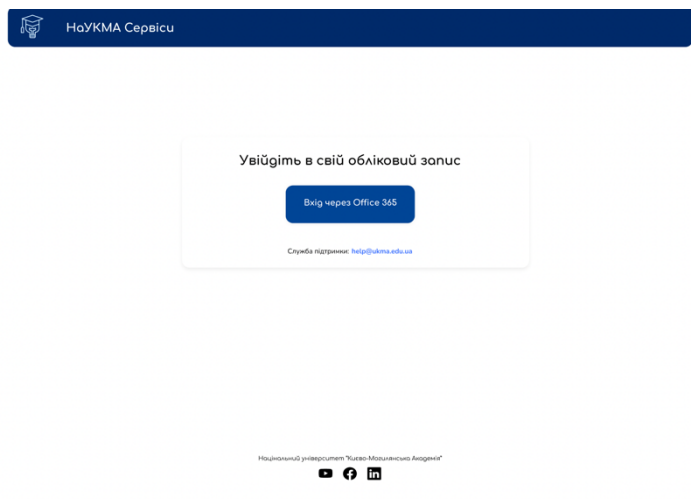


Рисунок 13 - сторінка логіну (інструкція)

Натискаємо вхід через Office 365 та входимо до корпоративного облікового запису. Потрапляємо на головну сторінку застосунку (див. **Error! Reference source not found.**).

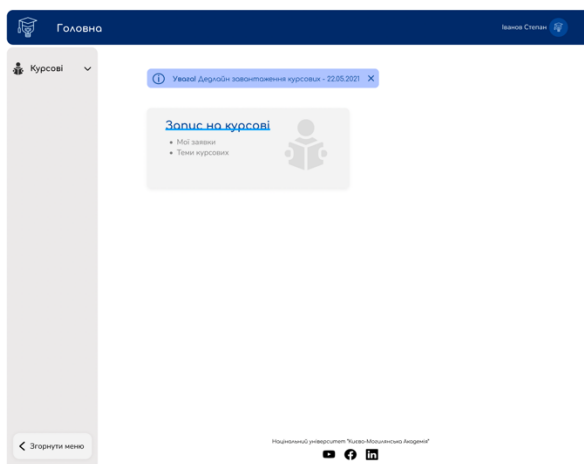


Рисунок 14 - головна сторінка (інструкція)

Натискаємо на «Теми курсових» та потрапляємо на сторінку тем курсових робіт.

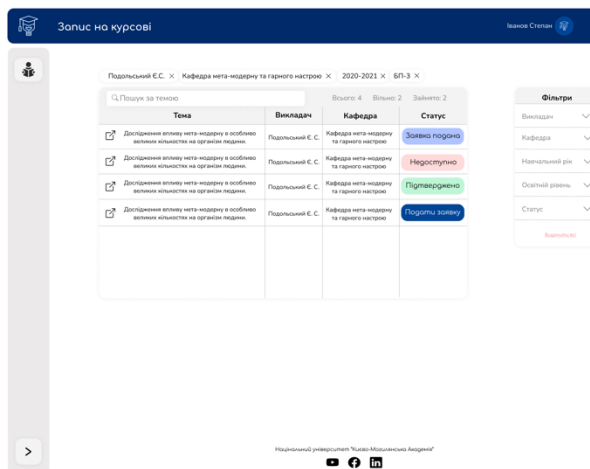


Рисунок 15 - сторінка тем курсових (інструкція)

Натискаємо на кнопку «Подати заявку» в останньому рядку таблиці та потрапляємо на сторінку курсової роботи.

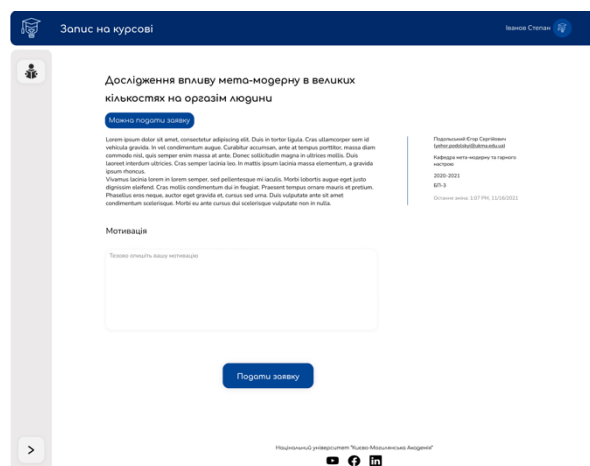


Рисунок 16 - сторінка подачі заявки на курсову (інструкція)

Пишемо мотивацію для написання курсової роботи та натискаємо «подати заявку» після чого сторінка змінюється та заявка подана.

Запис на курсові Іванна Світлана

---

### Дослідження впливу мета-модерну в великих кількостях на оразім людини

**Очікує підтвердження**

Latent ipsum dolor et amet, consectetur adipiscing elit. Duis in turpis ligula. Suspendisse potenti. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, elit ipsum, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, elit ipsum, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, elit ipsum, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.

**Мотивація**

Latent ipsum dolor et amet, consectetur adipiscing elit. Duis in turpis ligula. Suspendisse potenti. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, elit ipsum, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, elit ipsum, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, elit ipsum, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.

Розроблений Євг Світланом  
 latents@latents.com.ua  
 Користувач мета-модерну та гравця  
 вільно

2020-2021  
 07.11

Остання зміна: 1:07 PM 11/10/2021

**Скопуювати заявку**

Наданий код управління: "Університет "Київський національний університет"

Рисунок 17 - сторінка поданої заявки за курсову (інструкція)



# Висновки

Результатом роботи над проектом вийшов сайт, що є візуально приємним на вигляд, інтуїтивно зрозумілим, налічує усю необхідну функціональність та покращує загальний досвід користування. Проте так і не вдалося впровадити спеціально виділену мобільну версію, хоча тепер сайт можна використовувати на мобільних пристроях.

Під час розробки програми, було отримано важливий досвід командної розробки продукту у форматі щотижневих спринтів. Було виявлено багато недоліків та можливостей покращити процес розробки. Зокрема, виділю важливість мати готовий, остаточно узгоджений макет перед початком верстки сторінок застосунку. Також неймовірно важливо забезпечити стабільну роботу усіх сервісів на кожному пристрої де ведеться розробка та простий запуск їх, особливо в умовах, коли над проектом працює команда людей, деякі з яких працюватимуть лише над певними сервісами. Можна, наприклад, винести деякі сервіси на виділений сервер, до яких можна буде підключатися з локально розгорнутих сервісів, над якими ведеться робота. При доробленні вже існуючого продукту, не потрібно чіплятися за наявні бібліотеки і компоненти та не боятися позбуватися їх у випадку якщо це призведе до отримання кращого результату швидше.

## Список використаних джерел

- [1] [Percentage of mobile device website traffic worldwide from 1st quarter 2015 to 4th quarter 2021](#)
- [2] [Web development stacks in 2021 – software trends | TSH.io](#)
- [3] [What is Frameworks? \[Definition\] Types of Frameworks \(hackr.io\)](#)
- [4] [Framework Definition \(techterms.com\)](#)
- [5] [The Best Ten Backend Languages To Code | Back4App Blog](#)
- [6] [15 Best Web Development IDE in 2021\[Updated\] \(hackr.io\)](#)

# Додатки

## Додаток А – список ілюстрацій

Рисунок 1 - файлова структура проекту .....	15
Рисунок 2 - вміст папки app .....	15
Рисунок 3 - навігаційна панель.....	16
Рисунок 4 - зменшений блок курсових.....	16
Рисунок 5 - зменшена навігаційна панель.....	16
Рисунок 6 - хедер .....	17
Рисунок 7 - футер.....	17
Рисунок 8 - головна сторінка.....	17
Рисунок 9 - сторінка логіну .....	18
Рисунок 10 - сторінка заявок на курсові.....	19
Рисунок 11 - сторінка тем курсових.....	20
Рисунок 12 - сторінка курсової.....	21
Рисунок 13 - сторінка логіну (інструкція).....	22
Рисунок 14 - головна сторінка (інструкція) .....	22
Рисунок 15 - сторінка тем курсових (інструкція) .....	23
Рисунок 16 - сторінка подачі заявки на курсову (інструкція) .....	23
Рисунок 17 - сторінка поданої заявки за курсову (інструкція).....	24

## Додаток Б – програмний код

app.component.html

```
<div class="outer-container" id="wrapper">

<!-- Header -->

  <div class="header" >
    <a routerLink="/dashboard"></a>
    <span class="header-title">НаУКМА Сервіси</span>
    <ul class="navbar-nav ml-auto mt-2 mt-lg-0"
*ngIf="loginService.isAuthenticated()">
      {{loginService.getUser().username}}
    </ul>
  </div>

<!-- Body -->

  <div class="body">
    <div *ngIf="loginService.isAuthenticated()" class="{{classSidenav}}
sidenav">
      <div class="inner-sidenav">
        <div class="coursework-block"
*ngIf="loginService.hasAccessToServer(servers.COURSEWORK)">
          <a routerLinkActive="active-sidebar" routerLink="/dashboard"
class="course-item {{classCourseMenuSize}}">
            
            </a>
            <mat-expansion-panel class="{{classCourseMenu}}">
              <mat-expansion-panel-header (click) = "toggleCourseMenu()">
                <mat-panel-title class="sidebar-heading">
                  
                  Курсові
                </mat-panel-title>
              </mat-expansion-panel-header>
              <div class="mat-content">
                <a routerLinkActive="active-sidebar" routerLink="/course-
work/course-work" class="course-item">Теми курсових</a>
                <a routerLinkActive="active-sidebar"
*ngIf="loginService.hasAnyPermission(servers.COURSEWORK,
[courseWorkServerPermissions.VIEW_APPLICATION,
courseWorkServerPermissions.VIEW_APPLICATION_ANY,
courseWorkServerPermissions.VIEW_FACULTY_APPLICATION])" routerLink="/course-
work/applications" class="course-item">
                  Мої заявки
                </a>
              </div>
            </mat-expansion-panel>
          </div>
          <button class="btn toggle-sidenav-button" (click)="toggleSidenav()">
            
          </button>
        </div>
      </div>
    </div>
  </div>
```

```

    <div class="mat-sidenav-content-div">
      <div class="page-content-wrapper">
        <router-outlet class="container"></router-outlet>
      </div>
    </div>
  </div>
</div>

<!-- Footer -->

  <div class="footer">
    <label>Національний університет "Кієво-Могилянська Академія"</label>
    <div class="icons">
      
      
      
    </div>
  </div>

</div>

<!-- Links to fonts -->

<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Comfortaa&family=Nunito:wght@400;700&display=swap" rel="stylesheet">

```

## App.component.scss

```

.outer-container{
  display: flex;
  flex-direction: column;
  align-items: center;
  text-align: center;
}

// Header

.header{
  width: 97%;
  height: 66px;
  line-height: 66px;
  top: 0;
  background: #042966;
  border-radius: 15px;
  display: flex;
  flex-direction: row;
}

.header-logo {
  width: 50px;
  height: 60px;
  margin-left: 10px;
  //margin-top: -10px;
  color: white;
  mix-blend-mode: lighten;
}

```

```

}
.header-title{
  font-family: 'Comfortaa', cursive;
  font-style: normal;
  font-weight: bold;
  font-size: 24px;
  color: #FFFFFFF;
  vertical-align: middle;
  margin-left: 5%;
}
.navbar-nav{
  color: white;
  margin-left: 5px;
}

//Body

.body{
  width: 97%;
  display: flex;
}

.sidenav{
  transition-duration: 0.4s;
}

//classSidenav
.sidenav-big{
  width: 200px;
  border-right: none;
  background-color: #EBE9E9;
  text-align: center;
  border-radius: 15px;
  margin-top: 1%;
  margin-bottom: 1%;
}

.sidenav-small{
  width: 70px;
  border-right: none;
  background-color: #EBE9E9;
  text-align: center;
  border-radius: 15px;
  margin-top: 1%;
  margin-bottom: 1%;
}

.inner-sidenav{
  width: 100%;
  height: 100%;
  display: flex;
  flex-direction: column;
}

.active-sidebar {
  font-family: Nunito, serif;
  font-style: normal;
  font-weight: normal;
  font-size: 14px;
  line-height: 19px;

  color: #000000;

```

```

}
.course-item {
  color: black;
}

//classCourseMenuSize
.sidenav-course-small{
  display: block;
  margin: 20px;
}
.sidenav-course-big{
  display: none;
}

//classCourseMenu
.courseMenu-big{
  background: #F3F3F3;
  box-shadow: 0px 1px 20px rgba(0, 0, 0, 0.1);
  border-radius: 15px;
  margin: 10px;
  transition-duration: 0.4s;
}
.courseMenu-small{
  background-color: #EBE9E9;
  border-radius: 15px;
  box-shadow: none;
  margin: 10px;
  transition-duration: 0.4s;
}
.courseMenu-none {
  display: none;
  transition-duration: 0.4s;
}
.sidebar-heading{
  font-family: 'Comfortaa', cursive;
  font-style: normal;
  font-weight: bold;
  font-size: 18px;
  color: black;
}
.mat-content{
  display: flex;
  flex-direction: column;
}
.toggle-sidenav-button{
  height: 50px;
  background: #F3F3F3;
  box-shadow: 0px 1px 10px rgba(0, 0, 0, 0.07);
  border-radius: 15px;
  margin: auto 10px 10px;
}
.button-image{
  height: 20px;
  transition-duration: 0.4s;
  //"Згорнути меню"
}

//toggleSidenavArrow
.sidenav-arrow-right{

```

```

transform: rotate(180deg);
-ms-transform: rotate(180deg);
-webkit-transform: rotate(180deg);
}

.mat-sidenav-content-div{
margin: 0;
width:100%
}

.page-content-wrapper {
display: flex;
flex-direction: column;
//align-items: center;
text-align: center;
}

// Footer

.footer {
font-family: 'Comfortaa', cursive;
font-weight: 200;
text-align: center;
display: flex;
flex-direction: column;
margin-bottom: 10px;
background-color: white;
color: black;
width: 100%;
}

.icons {
margin-bottom: 15px;
margin-top: 10px;
display: flex;
column-gap: 15px;
justify-content: center;
}

.youtube {
margin-top: 3px
}

.media-icon {
width: 25px;
height: 25px;
}

```

#### app.component.ts

```

import {AfterViewInit, Component, ElementRef, OnInit, ViewChild} from
'@angular/core';
import { LoginSessionService } from "../services/login-session.service";
import { AuthControllerService, BuildInfoControllerService, BuildInfoDto,
ErrorDto } from "../api";
import { Router } from "@angular/router";
import { handleError } from "../utils/toasts";
import { BaseComponent } from "../components/base-component";
import {MatSidenav} from "@angular/material/sidenav";
import {BreakpointObserver} from "@angular/cdk/layout";
import {MatExpansionModule} from '@angular/material/expansion';

@Component({

```



```

selector: 'app-root',
templateUrl: './app.component.html',
styleUrls: ['./app.component.scss']
})
export class AppComponent extends BaseComponent implements OnInit, AfterViewInit
{
    @ViewChild(MatSidenav)
    sidenav!: MatSidenav;

    buildInfo: BuildInfoDto;

    constructor(
        public loginService: LoginSessionService,
        private authService: AuthControllerService,
        private buildInfoService: BuildInfoControllerService,
        private router: Router,
        private observer: BreakpointObserver
    ) {
        super();
    }

    public classSidenav: String = "sidenav-big"
    public classCourseMenu: String = "courseMenu-small"
    public classCourseMenuSize: String = "sidenav-course-big"
    public toggleSidenavArrow: String = 'sidenav-arrow-left'
    toggleSidenav() {
        if(this.classSidenav == 'sidenav-big'){
            this.classSidenav = 'sidenav-small'
            this.toggleSidenavArrow = 'sidenav-arrow-right'
            this.classCourseMenu = 'courseMenu-none'
            this.classCourseMenuSize = 'sidenav-course-small'
        } else {
            this.classSidenav = 'sidenav-big'
            this.toggleSidenavArrow = 'sidenav-arrow-left'
            this.classCourseMenuSize = 'sidenav-course-big'
            this.classCourseMenu = 'courseMenu-small'
        }
    }
    toggleCourseMenu() {
        if(this.classCourseMenu == 'courseMenu-big'){
            this.classCourseMenu = 'courseMenu-small'
        } else {
            this.classCourseMenu = 'courseMenu-big'
        }
    }
    logout() {
        this.authService.logout()
            .subscribe(
                response => {
                    this.loginService.logout();
                    this.router.navigateByUrl('logged-out');
                },
                error => handleError(error.error as ErrorDto)
            );
    }

    ngOnInit(): void {
        this.buildInfoService.getBuildInfo()
            .subscribe(response => this.buildInfo = response);
    }
}

```

```

        this.sidenav.open();
    }

    ngAfterViewInit() {
        this.observer.observe(['(max-width: 800px)']).subscribe((res) => {
            if (res.matches) {
                this.sidenav.mode = 'over';
                this.sidenav.close();
            } else {
                this.sidenav.mode = 'side';
                this.sidenav.open();
            }
        });
    }
}
}

```

### app.module.ts

```

import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppRoutingModule } from './app-routing.module';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { CommonModule } from '@angular/common';

import { AppComponent } from './app.component';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { EnvironmentApiConfigDelegate } from './setup/environment-api-setup';
import { ApiModule } from './api';
import { HttpClientModule } from '@angular/common/http';

import { MatIconModule } from "@angular/material/icon";
import { MatTableModule } from "@angular/material/table";
import { MatPaginatorModule } from "@angular/material/paginator";
import { FlatpickrModule } from "angularx-flatpickr";
import { MatSortModule } from "@angular/material/sort";
import { MatCardModule } from "@angular/material/card";
import { MatMenuModule } from "@angular/material/menu";
import { MatProgressSpinnerModule } from "@angular/material/progress-spinner";
import { AngularDatatableAdvancedModule } from "angular-datatable-advanced";
import { NgSelectModule } from "@ng-select/ng-select";
import { GetUserComponent } from './components/get-user/get-user.component';
import { FailedAuthComponent } from './components/failed-auth/failed-auth.component';
import { DashboardComponent } from './components/dashboard/dashboard.component';
import { IndexComponent } from './components/index/index.component';
import { FacultyCreateComponent } from './components/info-server/faculties/faculty-create/faculty-create.component';
import { FacultyEditComponent } from './components/info-server/faculties/faculty-edit/faculty-edit.component';
import { FacultyListComponent } from './components/info-server/faculties/faculty-list/faculty-list.component';
import { DepartmentCreateComponent } from './components/info-server/departments/department-create/department-create.component';
import { DepartmentEditComponent } from './components/info-server/departments/department-edit/department-edit.component';
import { DepartmentListComponent } from './components/info-server/departments/department-list/department-list.component';

```

```

import { TeacherCreateComponent } from './components/info-server/teachers/teacher-create/teacher-create.component';
import { TeacherEditComponent } from './components/info-server/teachers/teacher-edit/teacher-edit.component';
import { TeacherListComponent } from './components/info-server/teachers/teacher-list/teacher-list.component';
import { YearOfStudyingCreateComponent } from './components/info-server/year-of-studying/year-of-studying-create/year-of-studying-create.component';
import { YearOfStudyingListComponent } from './components/info-server/year-of-studying/year-of-studying-list/year-of-studying-list.component';
import { TeacherCourseThemesNumberCreateComponent } from './components/course-work/teacher-course-themes-number/teacher-course-themes-number-create/teacher-course-themes-number-create.component';
import { TeacherCourseThemesNumberUpdateComponent } from './components/course-work/teacher-course-themes-number/teacher-course-themes-number-update/teacher-course-themes-number-update.component';
import { TeacherCourseThemesNumberListComponent } from './components/course-work/teacher-course-themes-number/teacher-course-themes-number-list/teacher-course-themes-number-list.component';
import { CourseWorkCreateComponent } from './components/course-work/course-work/course-work-create/course-work-create.component';
import { CourseWorkEditComponent } from './components/course-work/course-work/course-work-edit/course-work-edit.component';
import { CourseWorkListComponent } from './components/course-work/course-work/course-work-list/course-work-list.component';
import { ApplicationCreateComponent } from './components/course-work/application/application-create/application-create.component';
import { ApplicationEditComponent } from './components/course-work/application/application-edit/application-edit.component';
import { ApplicationListComponent } from './components/course-work/application/application-list/application-list.component';
import { FacultyStaffCreateComponent } from './components/info-server/faculty-staff/faculty-staff-create/faculty-staff-create.component';
import { FacultyStaffEditComponent } from './components/info-server/faculty-staff/faculty-staff-edit/faculty-staff-edit.component';
import { FacultyStaffListComponent } from './components/info-server/faculty-staff/faculty-staff-list/faculty-staff-list.component';
import { StudentCreateComponent } from './components/info-server/student/student-create/student-create.component';
import { StudentEditComponent } from './components/info-server/student/student-edit/student-edit.component';
import { StudentListComponent } from './components/info-server/student/student-list/student-list.component';
import { LoginAsComponent } from './components/login-as/login-as.component';
import { MarkingConfigUpdateComponent } from './components/course-work/marking-config/marking-config-update/marking-config-update.component';
import { MarkingConfigListComponent } from './components/course-work/marking-config/marking-config-list/marking-config-list.component';
import { CourseWorkViewComponent } from './components/course-work/course-work/course-work-view/course-work-view.component';
import { NgxSpinnerModule } from "ngx-spinner";
import { MatSidenavModule } from "@angular/material/sidenav";
import { MatDividerModule } from "@angular/material/divider";
import { MatExpansionModule } from "@angular/material/expansion";

export function apiConfig() {
  return new EnvironmentApiConfigDelegate();
}

```

```

@NgModule ({
  declarations: [
    AppComponent,
    GetUserComponent,
    FailedAuthComponent,
    DashboardComponent,
    IndexComponent,
    FacultyCreateComponent,
    FacultyEditComponent,
    FacultyListComponent,
    DepartmentCreateComponent,
    DepartmentEditComponent,
    DepartmentListComponent,
    TeacherCreateComponent,
    TeacherEditComponent,
    TeacherListComponent,
    YearOfStudyingCreateComponent,
    YearOfStudyingListComponent,
    TeacherCourseThemesNumberCreateComponent,
    TeacherCourseThemesNumberUpdateComponent,
    TeacherCourseThemesNumberListComponent,
    CourseWorkCreateComponent,
    CourseWorkEditComponent,
    CourseWorkListComponent,
    ApplicationCreateComponent,
    ApplicationEditComponent,
    ApplicationListComponent,
    FacultyStaffCreateComponent,
    FacultyStaffEditComponent,
    FacultyStaffListComponent,
    StudentCreateComponent,
    StudentEditComponent,
    StudentListComponent,
    LoginAsComponent,
    MarkingConfigUpdateComponent,
    MarkingConfigListComponent,
    CourseWorkViewComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserModuleAnimationsModule,
    FormsModule,
    CommonModule,
    ReactiveFormsModule,
    HttpClientModule,
    ApiModule.forRoot(apiConfig),
    FlatpickrModule.forRoot(),
    MatIconModule,
    MatTableModule,
    MatPaginatorModule,
    MatSortModule,
    MatCardModule,
    MatMenuModule,
    MatProgressSpinnerModule,
    AngularDatatableAdvancedModule.forRoot({
      filterTranslate: {
        contains: 'Містить',
        startsWith: 'Починається з',

```

```

        endsWith: 'Закінчується з',
        equals: 'Рівне',
        notEquals: 'Нерівне',
        range: 'Проміжок',
        in: 'Входить в',
        nin: 'Невходить в',
        greaterThan: 'Більше ніж',
        greaterThanOrEquals: 'Більше ніж або рівне',
        lessThan: 'Менше ніж',
        lessThanOrEquals: 'Менше ніж або рівне',
        applyFilter: 'Застосувати',
        clearFilter: 'Очистити',
        filterOutOptions: 'Фільтрувати',
        filterRangeFrom: 'З',
        filterRangeTo: 'До',
        filterPlaceholder: 'Фільтр'
    }
 )),
  NgSelectModule,
  NgxSpinnerModule,
  MatSidenavModule,
  MatDividerModule,
  MatExpansionModule
],
providers: [],
bootstrap: [AppComponent]
})
export class AppModule {
}

```

### app-routing.module.ts

```

import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { LoginSessionService } from '../services/login-session.service';
import { GetUserComponent } from '../components/get-user/get-user.component';
import { FailedAuthComponent } from '../components/failed-auth/failed-auth.component';
import { DashboardComponent } from '../components/dashboard/dashboard.component';
import { IndexComponent } from '../components/index/index.component';
import { FacultyCreateComponent } from '../components/info-server/faculties/faculty-create/faculty-create.component';
import { ServerNameDto } from '../api';
import { CourseWorkPermission, InfoServerPermission } from '../model/permissions';
import { FacultyEditComponent } from '../components/info-server/faculties/faculty-edit/faculty-edit.component';
import { FacultyListComponent } from '../components/info-server/faculties/faculty-list/faculty-list.component';
import { DepartmentCreateComponent } from '../components/info-server/departments/department-create/department-create.component';
import { DepartmentEditComponent } from '../components/info-server/departments/department-edit/department-edit.component';
import { DepartmentListComponent } from '../components/info-server/departments/department-list/department-list.component';
import { TeacherCreateComponent } from '../components/info-server/teachers/teacher-create/teacher-create.component';
import { TeacherEditComponent } from '../components/info-server/teachers/teacher-edit/teacher-edit.component';

```

```

import { TeacherListComponent } from "../components/info-server/teachers/teacher-list/teacher-list.component";
import { YearOfStudyingCreateComponent } from "../components/info-server/year-of-studying/year-of-studying-create/year-of-studying-create.component";
import { YearOfStudyingListComponent } from "../components/info-server/year-of-studying/year-of-studying-list/year-of-studying-list.component";
import { TeacherCourseThemesNumberCreateComponent } from "../components/course-work/teacher-course-themes-number/teacher-course-themes-number-create/teacher-course-themes-number-create.component";
import { TeacherCourseThemesNumberListComponent } from "../components/course-work/teacher-course-themes-number/teacher-course-themes-number-list/teacher-course-themes-number-list.component";
import { TeacherCourseThemesNumberUpdateComponent } from "../components/course-work/teacher-course-themes-number/teacher-course-themes-number-update/teacher-course-themes-number-update.component";
import { CourseWorkListComponent } from "../components/course-work/course-work/course-work-list/course-work-list.component";
import { CourseWorkCreateComponent } from "../components/course-work/course-work/course-work-create/course-work-create.component";
import { CourseWorkEditComponent } from "../components/course-work/course-work/course-work-edit/course-work-edit.component";
import { ApplicationListComponent } from "../components/course-work/application/application-list/application-list.component";
import { ApplicationCreateComponent } from "../components/course-work/application/application-create/application-create.component";
import { ApplicationEditComponent } from "../components/course-work/application/application-edit/application-edit.component";
import { FacultyStaffListComponent } from "../components/info-server/faculty-staff/faculty-staff-list/faculty-staff-list.component";
import { FacultyStaffCreateComponent } from "../components/info-server/faculty-staff/faculty-staff-create/faculty-staff-create.component";
import { FacultyStaffEditComponent } from "../components/info-server/faculty-staff/faculty-staff-edit/faculty-staff-edit.component";
import { StudentListComponent } from "../components/info-server/student/student-list/student-list.component";
import { StudentCreateComponent } from "../components/info-server/student/student-create/student-create.component";
import { StudentEditComponent } from "../components/info-server/student/student-edit/student-edit.component";
import { LoginAsComponent } from "../components/login-as/login-as.component";
import { MarkingConfigUpdateComponent } from "../components/course-work/marking-config/marking-config-update/marking-config-update.component";
import { MarkingConfigListComponent } from "../components/course-work/marking-config/marking-config-list/marking-config-list.component";
import { CourseWorkViewComponent } from "../components/course-work/course-work/course-work-view/course-work-view.component";

const routes: Routes = [
  {
    path: '',
    component: IndexComponent
  },
  {
    path: 'start-login',
    redirectTo: '/get-user'
  },
  {
    path: 'get-user',
    component: GetUserComponent,

```

```

    canActivate: [ LoginSessionService ]
  },
  {
    path: 'failed-auth',
    component: FailedAuthComponent,
    canActivate: [ LoginSessionService ],
    data: { anonymous: true }
  },
  {
    path: 'dashboard',
    component: DashboardComponent,
    canActivate: [ LoginSessionService ],
    data: { authenticated: true }
  },

  {
    path: 'info/faculties',
    component: FacultyListComponent,
    canActivate: [ LoginSessionService ],
    data: {
      serverName: ServerNameDto.INFOSERVER
    }
  },
  {
    path: 'info/faculties/create',
    component: FacultyCreateComponent,
    canActivate: [ LoginSessionService ],
    data: {
      serverName: ServerNameDto.INFOSERVER,
      permission: InfoServerPermission.CREATE_FACULTY
    }
  },
  {
    path: 'info/faculties/:id/edit',
    component: FacultyEditComponent,
    canActivate: [ LoginSessionService ],
    data: {
      serverName: ServerNameDto.INFOSERVER,
      permission: InfoServerPermission.EDIT_FACULTY
    }
  },

  {
    path: 'info/departments',
    component: DepartmentListComponent,
    canActivate: [ LoginSessionService ],
    data: {
      serverName: ServerNameDto.INFOSERVER
    }
  },
  {
    path: 'info/departments/create',
    component: DepartmentCreateComponent,
    canActivate: [ LoginSessionService ],
    data: {
      serverName: ServerNameDto.INFOSERVER,
      permission: InfoServerPermission.CREATE_DEPARTMENT
    }
  }
}

```

```

},
{
  path: 'info/departments/:id/edit',
  component: DepartmentEditComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.INFOSERVER,
    permission: InfoServerPermission.EDIT_DEPARTMENT
  }
},

{
  path: 'info/teachers',
  component: TeacherListComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.INFOSERVER
  }
},

{
  path: 'info/teachers/create',
  component: TeacherCreateComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.INFOSERVER,
    anyPermission: [ InfoServerPermission.CREATE_ANY_TEACHER ]
  }
},

{
  path: 'info/teachers/:email/edit',
  component: TeacherEditComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.INFOSERVER,
    anyPermission: [ InfoServerPermission.EDIT_TEACHER_INFO,
InfoServerPermission.EDIT_ANY_USER_INFO ]
  }
},

{
  path: 'info/faculty-staff',
  component: FacultyStaffListComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.INFOSERVER
  }
},

{
  path: 'info/faculty-staff/create',
  component: FacultyStaffCreateComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.INFOSERVER,
    anyPermission: [ InfoServerPermission.EDIT_FACULTY_STAFF_INFO,
InfoServerPermission.EDIT_ANY_USER_INFO ]
  }
},

{

```



```

path: 'info/faculty-staff/:email/edit',
component: FacultyStaffEditComponent,
canActivate: [ LoginSessionService ],
data: {
  serverName: ServerNameDto.INFOSERVER,
  anyPermission: [ InfoServerPermission.EDIT_FACULTY_STAFF_INFO,
InfoServerPermission.EDIT_ANY_USER_INFO ]
}
},
{
  path: 'info/students',
  component: StudentListComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.INFOSERVER
  }
},
{
  path: 'info/students/create',
  component: StudentCreateComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.INFOSERVER,
    anyPermission: [ InfoServerPermission.EDIT_STUDENT_INFO,
InfoServerPermission.EDIT_ANY_USER_INFO ]
  }
},
{
  path: 'info/students/:email/edit',
  component: StudentEditComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.INFOSERVER,
    anyPermission: [ InfoServerPermission.EDIT_STUDENT_INFO,
InfoServerPermission.EDIT_ANY_USER_INFO ]
  }
},
{
  path: 'info/year-of-studying/create',
  component: YearOfStudyingCreateComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.INFOSERVER,
    permission: InfoServerPermission.CREATE_YEAR_OF_STUDYING
  }
},
{
  path: 'info/year-of-studying',
  component: YearOfStudyingListComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.INFOSERVER
  }
},
{
  path: 'course-work/teacher-course-themes-number',

```

```

component: TeacherCourseThemesNumberListComponent,
canActivate: [ LoginSessionService ],
data: {
  serverName: ServerNameDto.COURSEWORK
}
},
{
  path: 'course-work/teacher-course-themes-number/create',
  component: TeacherCourseThemesNumberCreateComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.COURSEWORK,
    anyPermission: [ CourseWorkPermission.CREATE_NUMBER_OF_CW,
CourseWorkPermission.CREATE_NUMBER_OF_CW_ANY ]
  }
},
{
  path: 'course-work/teacher-course-themes-number/:id/edit',
  component: TeacherCourseThemesNumberUpdateComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.COURSEWORK,
    anyPermission: [ CourseWorkPermission.EDIT_NUMBER_OF_CW,
CourseWorkPermission.EDIT_NUMBER_OF_CW_ANY ]
  }
},
{
  path: 'course-work/course-work',
  component: CourseWorkListComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.COURSEWORK
  }
},
{
  path: 'course-work/course-work/create',
  component: CourseWorkCreateComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.COURSEWORK,
    anyPermission: [ CourseWorkPermission.CREATE_COURSE_WORK,
CourseWorkPermission.CREATE_COURSE_WORK_ANY,
CourseWorkPermission.CREATE_COURSE_WORK_DEPARTMENT ]
  }
},
{
  path: 'course-work/course-work/:id/edit',
  component: CourseWorkEditComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.COURSEWORK,
    anyPermission: [ CourseWorkPermission.EDIT_COURSE_WORK_ANY,
CourseWorkPermission.EDIT_COURSE_WORK,
CourseWorkPermission.EDIT_COURSE_WORK_DEPARTMENT ]
  }
},
{

```

```

path: 'course-work/course-work/:id/view',
component: CourseWorkViewComponent,
canActivate: [ LoginSessionService ],
data: {
  serverName: ServerNameDto.COURSEWORK
}
},
{
  path: 'course-work/applications',
  component: ApplicationListComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.COURSEWORK,
    anyPermission: [ CourseWorkPermission.VIEW_APPLICATION,
CourseWorkPermission.VIEW_APPLICATION_ANY,
CourseWorkPermission.VIEW_FACULTY_APPLICATION ]
  }
},
{
  path: 'course-work/applications/create',
  component: ApplicationCreateComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.COURSEWORK,
    anyPermission: [ CourseWorkPermission.CREATE_APPLICATION,
CourseWorkPermission.CREATE_APPLICATION_ANY,
CourseWorkPermission.CREATE_FACULTY_APPLICATION ]
  }
},
{
  path: 'course-work/applications/:id/edit',
  component: ApplicationEditComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.COURSEWORK,
    anyPermission: [ CourseWorkPermission.EDIT_APPLICATION,
CourseWorkPermission.EDIT_APPLICATION_ANY,
CourseWorkPermission.EDIT_FACULTY_APPLICATION ]
  }
},
{
  path: 'course-work/marking-config',
  component: MarkingConfigListComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.COURSEWORK,
    anyPermission: [ CourseWorkPermission.VIEW_MARKING_CONFIG,
CourseWorkPermission.VIEW_MARKING_CONFIG_ANY ]
  }
},
{
  path: 'course-work/marking-config/create',
  component: MarkingConfigUpdateComponent,
  canActivate: [ LoginSessionService ],
  data: {
    serverName: ServerNameDto.COURSEWORK,
    anyPermission: [ CourseWorkPermission.CREATE_MARKING_CONFIG,
CourseWorkPermission.CREATE_MARKING_CONFIG_ANY ]
  }
}

```

```

    }
  },
  {
    path: 'course-work/marking-config/:id/edit',
    component: MarkingConfigUpdateComponent,
    canActivate: [ LoginSessionService ],
    data: {
      serverName: ServerNameDto.COURSEWORK,
      anyPermission: [ CourseWorkPermission.EDIT_MARKING_CONFIG,
CourseWorkPermission.EDIT_MARKING_CONFIG_ANY ]
    }
  },
  {
    path: 'login-as',
    component: LoginAsComponent,
    canActivate: [ LoginSessionService ],
    data: {
      serverName: ServerNameDto.UISERVER,
      permission: 'admin'
    }
  }
]
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule {
}

```

### index.component.html

```

<div class="outer-container">
  <div class="login-div">
    <div class="title">
      <label class="title-text">Увійдіть у свій обліковий запис</label>
    </div>
    <div class="login-btn">
      <a href="/start-login" class="btn btn-success">Вхід через Office 365</a>
    </div>
    <div class="support">
      <label class="bottom-text">Служба підтримки: </label>
      <a class="mail" href="mailto:help@ukma.edu.ua">help@ukma.edu.ua</a>
    </div>
  </div>
</div>

<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Comfortaa&family=Nunito:wght@400;
700&display=swap" rel="stylesheet">

```

### Index.component.scss

```

.login-div{
  position: relative;
  width: 60%;
}

```

```

max-width: 954px;
max-height: 320px;
height: 50%;
margin-top: 12em;
text-align: center;

background: rgba(255, 255, 255, 0.7);
box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.07);
border-radius: 15px;

display: flex;
flex-direction: column;
justify-content: space-evenly;
// margin: 0 auto;
}
.body{
background-color: rgb(229,229,229);
}

.btn{
// position: absolute;
max-width: 290px;
width: 70%;
height: 60px;
//margin-top: 70px;
background: #164491;
border-radius: 15px;
font-size: 15px;
padding-top: 16px;

font-family: 'Nunito', sans-serif;
font-style: normal;
font-weight: bold;
line-height: 23px;

color: #FFFFFF;
}

.outer-container{
height: calc(100vh - 147px);
background-color: rgb(255,255,255);
text-align: center;
display: flex;
justify-content: center;
}
.title {
//margin-top: 10px;
}
.title-text{

font-family: 'Comfortaa', cursive;
font-style: normal;
font-weight: bold;
font-size: 26px;
line-height: 29px;
text-align: center;

color: #000000;
}

```

```

.support{
  bottom: 0;
  //margin-top: 100px;
  display: flex;
  column-gap: 9px;
  justify-content: center;

  //position: absolute;
  //width: 100px;
}

.bottom-text {
  font-family: 'Nunito', sans-serif;
  font-style: normal;
  font-weight: normal;
  font-size: 14px;
  line-height: 19px;

  color: #000000;
}

.mail {
  font-family: 'Nunito', sans-serif;
  font-style: normal;
  font-weight: bold;
  font-size: 14px;
  line-height: 19px;

  color: #4870FF;
}

.login-btn {
  justify-content: center;
}

```

### Index.component.ts

```

import { Component, OnInit } from '@angular/core';
import { LoginSessionService } from "../../services/login-session.service";

@Component({
  selector: 'app-index',
  templateUrl: './index.component.html',
  styleUrls: ['./index.component.scss']
})
export class IndexComponent implements OnInit {

  constructor(
    public loginService: LoginSessionService
  ) { }

  ngOnInit(): void {
  }

}

```

## dashboard.component.html

```
<div class="body-div">
<div class="outer-div">
  <div class="course-menu">
    <label>Запис на курсові</label>
    <ul>
      <li><a routerLink="/course-work/applications">Мої заявки</a></li>
      <li><a routerLink="/course-work/course-work">Теми курсових</a></li>
    </ul>
  </div>
</div>
</div>
```

## dashboard.component.scss

```
.outer-div {
  display: flex;
  padding: 3em;
  margin-top: -40px;
}
li {
  color: black;
}
.course-menu {
  margin-top: 5%;
  width: 483px;
  height: 191px;
  left: 0px;
  top: 0px;

  background: #F3F3F3;
  box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.07);
  border-radius: 10px;

  text-align: left;
  padding: 2%;
}
label {
  font-family: Comfortaa, serif;
  font-style: normal;
  font-weight: bold;
  font-size: 27px;
  /* main_blue */

  color: #164491;
}
.body-div {
  min-height: 85vh;
  display: flex;
  flex-direction: column;
  width: 100%;
}
a {
  font-family: Nunito, serif;
  font-style: normal;
  font-weight: normal;
  font-size: 18px;
}
```

```

    /* muted */

    color: #7C7C7C;
}

//table

```

#### dashboard.component.ts

```

import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-dashboard',
  templateUrl: './dashboard.component.html',
  styleUrls: ['./dashboard.component.scss']
})
export class DashboardComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}

```

#### index.html

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>UKMA services</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <link
href="https://fonts.googleapis.com/css?family=Roboto:300,400,500&display=swap"
rel="stylesheet">
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
</head>
<body style="width:100%; height:100%; margin:0px; padding:0px">
  <app-root></app-root>
</body>
</html>

```

#### main.ts

```

import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

if (environment.production) {
  enableProdMode();
}

```



```
platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.error(err));
```

## course-work-list.component.html

```
<div class="body-div">
  <div class="outer-div">
    <div class="course-menu">
      <div class="table">
<!--      <ada-table-->
<!--        [pageSizeOptions]="[10, 20]"-->
<!--        [pageSize]="10"-->
<!--        [columns]="columns"-->
<!--        [rowsLoader]="rowsLoader"-->
<!--        [cogMenu]="menu"-->
<!--      ></ada-table-->
      <table>
        <input type="text" placeholder="Пошук за темою">
        <tr class="table-header">
          <th>Тема</th>
          <th>Викладач</th>
          <th>Кафедра</th>
          <th>Статус</th>
        </tr>
        <tr>
          <td>Дослідження впливу мета-модерну в особливо великих кількостях на
організм людини.</td>
          <td>Подольський Є. С.</td>
          <td>Кафедра мета-модерну
та гарного настрою</td>
          <td>Заявка подана</td>
        </tr>
        <tr>
          <td>Дослідження впливу мета-модерну в особливо великих кількостях на
організм людини.</td>
          <td>Подольський Є. С.</td>
          <td>Кафедра мета-модерну
та гарного настрою</td>
          <td><a routerLink="/course-work/course-work/create" class="btn btn-
success" *ngIf="loginService.hasAnyPermission(servers.COURSEWORK,
[courseWorkServerPermissions.CREATE_COURSE_WORK,
courseWorkServerPermissions.CREATE_COURSE_WORK_ANY])">Додати</a></td>
        </tr>
      </table>
    </div>
    <div class="filters-box">
      <div class="coursework-block"
*ngIf="loginService.hasAccessToServer(servers.COURSEWORK)">
        <mat-expansion-panel>
          <mat-expansion-panel-header>
            <mat-panel-title class="sidebar-heading">
              Фільтри
            </mat-panel-title>
          </mat-expansion-panel-header>
          <div class="mat-content">
            <label>Викладач</label>
            <label>Кафедра</label>
          </div>
        </mat-expansion-panel>
      </div>
    </div>
  </div>
</div>
```

```
        </mat-expansion-panel>
      </div>
    </div>
  </div>
</div>
```

### Course-work-list.component.scss

```
.title {
  //margin-top: 10px;
}
.title-text {
  font-size: 30px;
  color: #164491;
}
.outer-div {
  display: flex;
}

.course-menu {
  margin-top: 1%;
  display: flex;
  flex-direction: row;
}

.body-div {
  min-height: 85vh;
  display: flex;
  flex-direction: column;
}

table{
  background: #F3F3F3;
  box-shadow: 0px 1px 10px rgba(0, 0, 0, 0.07);
  border-radius: 15px;
  margin-left: 5%;
  margin-right: 5%;
}

.table-header {
  background: #F3F3F3;
}

th {
  border-top: none;
}

th:nth-child(1){
  border-top-left-radius: 15px;
}

th:nth-child(4) {
  border-top-right-radius: 15px;
}

td {
  font-family: 'Nunito', sans-serif;
  font-style: normal;
  font-weight: normal;
}
```

```

    font-size: 14px;
    line-height: 19px;
    background: #FFFFFF;
  }
mat-expansion-panel{
  filter: drop-shadow(0px 1px 10px rgba(0, 0, 0, 0.07));
  border-radius: 15px;
}
mat-expansion-panel-header {
  border-radius: 0;
}
.mat-content{
  display: flex;
  flex-direction: column;
}
input {
  -webkit-appearance: none;
  background: #FFFFFF;
  box-shadow: 0px 1px 20px rgba(0, 0, 0, 0.05);
  border-radius: 5px;
  width: 100%;
  margin: 1%;
}
}

```

#### Course-work-l1ist.component.ts

```

import { Component } from '@angular/core';
import { CogMenu, Column, RowsLoader } from "angular-datatable-advanced";
import { CourseWorkControllerService, CourseWorkDto, ServerNameDto } from
"../../../../../../api";
import { YearOfStudyEnumDto_118 } from "../../../../../../118/year-of-study";
import { CourseWorkPermission } from "../../../../../../model/permissions";
import { map } from "rxjs/operators";
import { LoginSessionService } from "../../../../../../services/login-
session.service";
import { FilterListService } from "../../../../../../utils/filter-list-service";
import { Router } from "@angular/router";
import { FilterIn } from "angular-datatable-advanced/lib/model/filter-in";
import { BaseComponent } from "../../../../../../base-component";

@Component({
  selector: 'app-course-work-list',
  templateUrl: './course-work-list.component.html',
  styleUrls: ['./course-work-list.component.scss']
})
export class CourseWorkListComponent extends BaseComponent {
  columns: Column[] = [
    {
      columnKey: 'topic',
      columnName: 'Тема',
      filterable: true
    }, {
      columnKey: 'description',
      columnName: 'Опис',
      filterable: true,
      cellRender: item => `

```
${item.description}</pre>`
    }, {

```


```

```

columnKey: 'emails',
columnName: 'Викладач',
filterable: true,
filterInPromise: this.filterListService.loadTeachers(),
cellRender: item => {
  const teacher = (item as CourseWorkDto).teacher.teacher;
  if (teacher.firstName) {
    return `${teacher.firstName} ${teacher.lastName}
(${teacher.departmentName})`;
  } else {
    return teacher.email;
  }
}, {
  columnKey: 'teacher.department.name',
  columnName: 'Кафедра',
  filterable: true,
  filterInPromise: this.filterListService.loadDepartments()
}, {
  columnKey: 'yearPeriod',
  columnName: 'Навчальний рік',
  filterable: true,
  filterInPromise: this.filterListService.loadYearOfStudying(),
  cellRender: item => {
    const yearPeriod = (item as CourseWorkDto).teacher.yearPeriod;
    return yearPeriod.start + ' - ' + yearPeriod.end
  }
}, {
  columnKey: 'yearOfStudy',
  columnName: 'Рік навчання студента',
  filterable: true,
  filterIn: this.yearOfStudyIn(),
  cellRender: item => {
    const yearOfStudy = (item as CourseWorkDto).teacher.yearOfStudy;
    return YearOfStudyEnumDto_118[yearOfStudy];
  }
}, {
  columnKey: 'application',
  columnName: 'Заявка',
  cellRender: item => {
    const courseWork = (item as CourseWorkDto);
    return courseWork.approvedApplicationId
      ? ``
      : '-';
  }
}, {
  columnKey: 'timeOfCreation',
  columnName: 'Час створення',
  sortable: true,
  cellRender: item => this.formatDate\(\(item as
CourseWorkDto\).createdDateTime, 'dd/MM/yyyy HH:mm:ss'\)
}
\];

menu: CogMenu\[\] = \[
  {
    name: 'Переглянути',

```

```

        renderCheck: item =>
this.loginService.hasAccessToServer(ServerNameDto.COURSEWORK),
        onClick: item => this.router.navigateByUrl(`course-work/course-
work/${item.id}/view`)
      }, {
        name: 'Редагувати',
        renderCheck: item =>
this.loginService.hasAnyPermission(ServerNameDto.COURSEWORK,
[CourseWorkPermission.EDIT_COURSE_WORK,
CourseWorkPermission.EDIT_COURSE_WORK_ANY]),
        onClick: item => this.router.navigateByUrl(`course-work/course-
work/${item.id}/edit`)
      }, {
        name: 'Створити заявку',
        renderCheck: item =>
this.loginService.hasAnyPermission(ServerNameDto.COURSEWORK,
[CourseWorkPermission.CREATE_APPLICATION,
CourseWorkPermission.CREATE_APPLICATION_ANY,
CourseWorkPermission.CREATE_FACULTY_APPLICATION]),
        onClick: item => this.router.navigateByUrl(`course-
work/applications/create?courseWorkId=${item.id}`)
      }
    ]

    rowsLoader: RowsLoader = parameters => {
      let department_ids = null;
      let emails = null;
      let year_periods = null;
      let year_of_studys = null;
      let topic = null;
      let description = null;

      parameters.filters.forEach(filter => {
        if (filter.column.columnKey === 'teacher.department.name') {
          department_ids = filter.filter.value1;
        }
        if (filter.column.columnKey === 'emails') {
          emails = filter.filter.value1;
        }
        if (filter.column.columnKey === 'yearPeriod') {
          year_periods = filter.filter.value1;
        }
        if (filter.column.columnKey === 'yearOfStudy') {
          year_of_studys = filter.filter.value1;
        }
        if (filter.column.columnKey === 'topic') {
          topic = filter.filter.value1;
        }
        if (filter.column.columnKey === 'description') {
          description = filter.filter.value1;
        }
      });
    };

    const restrict = {
      offset: parameters.size * parameters.page,
      limit: parameters.size,
      topic,
      description,
      department_ids,

```

```

        emails,
        year_periods,
        year_of_studys,
        order_by: parameters.sort ? parameters.sort.column.columnKey : 'id',
        order_direction: parameters.sort ? parameters.sort.direction.toUpperCase()
: null
    };

    return this.courseWorkService.getList(JSON.stringify(restrict))
        .pipe(map(response => {
            return {
                rows: response.items,
                totalItems: response.total
            }
        }));
    });

constructor(
    public loginService: LoginSessionService,
    private courseWorkService: CourseWorkControllerService,
    private filterListService: FilterListService,
    private router: Router,
) {
    super();
}

private yearOfStudyIn(): Array<FilterIn> {
    return Object.keys(YearOfStudyEnumDto_118).map(item => {
        return {
            value: item,
            label: YearOfStudyEnumDto_118[item]
        };
    });
}
}
}

```