

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут телекомунікаційних систем

(повне найменування інституту, факультету)

Кафедра телекомунікацій

(повна назва кафедри)

До захисту допущено

В.о. завідувача кафедри

_____ **Валерій ЯВІСЯ**

(підпис)

(Ім'я, прізвище)

“ 04 ” _____ червня _____ 2020р.

Дипломна робота

на здобуття освітнього ступеня “бакалавр”

(назва ОС)

Спеціальність 172 Телекомунікації та радіотехніка,

(код і назва)

на тему: Кафедральний та інститутський веб-сайти як засоби інфокомунікацій

Виконав: студент IV курсу, групи ТЗ-61

(шифр групи)

_____ Придибайло Артем Миколайович

(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник _____ доцент кафедри ТК Цурін О.П.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Консультант _____

(назва розділу)

_____ (посада, вчене звання, науковий ступінь, прізвище, ініціали)

_____ (підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент _____

(підпис)

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут телекомунікаційних систем
(повна назва)

Кафедра телекомунікацій
(повна назва)

Освітній ступінь бакалавр

Спеціальність 172 Телекомунікації та радіотехніка
(код і назва)

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Валерій ЯВІСЯ
(підпис) (ім'я, прізвище)

“ 22 ” січня 2020р.

З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Придибайло Артем Миколайович
(прізвище, ім'я, по батькові)

1. Тема роботи: Кафедральний та інститутський веб-сайти як засоби інфокомунікацій

Керівник роботи доцент кафедри ТК Цурін О.П.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 30 березня 2020 р. № 924-с

2. Термін подання студентом роботи 04.06.2020

3. Вихідні дані до роботи: CMS Drupal 8, phpMyAdmin, Open Server, FTP-клієнт FileZilla, редактор VS Code.

4.Зміст роботи:

1) Аналіз ринку веб-ресурсів

2) Розгляд сайту як джерело інформації і засіб надання результатів користувачу при різних режимах роботи

3) Технічна реалізація веб сайту на основі CMS

5. Орієнтований перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

Слайд №1 Актуальність, мета та практична цінність роботи.

Слайд №2 Чому саме Drupal?

Слайд №3 Засоби управління контентом.

Слайд №4 Теми у CMS.

Слайд №5 Користувачі, ролі та права.

Слайд №6 Сайт ІТС.

Слайд №10 Висновки до роботи, подальший розвиток ресурсу.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 07.11.2019

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Розробка, оформлення, узгодження та затвердження технічного завдання на дипломну роботу	10.11.2019 – 21.11. 2019	
2	Вивчення та аналіз ринку веб-сайтів та веб додатків	01.11.2019 – 24.11.2019	
3	Аналіз доступних методів розробки веб-сайтів	25.11.2019 – 22.12.2019	
4	Вивчення CMS Drupal	13.01.2020 – 01.03.2020	
5	Розробка сайту, налаштування прав, наповнення контентом	02.03.2020 – 10.05.2020	
6	Розгортання веб-сайту в мережі	11.05.2020 – 24.05.2020	
7	Підведення підсумків та оформлення пояснювальної записки	25.05.2020 – 04.06.2020	

Студент

(підпис)

Придибайло А.М.

(прізвище та ініціали)

Керівник роботи

(підпис)

Цурін О.П.

(прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської дипломної роботи містить: 55 сторінок, 23 рисунки, 4 додатки та 15 джерел.

Метою роботи є розробка веб-сайту ІТС та розгляд його як джерела та приймача інформації, побудова та організація взаємодії користувачів сайту: відвідувачів, контент менеджерів, веб дизайнерів та адмінів, їх роль та організація роботи.

В сучасному світі інтернет займає значну частину життя кожної людини, тому нехтування просуванням своїх послуг чи товарів в мережі стає абсурдним. В даній роботі розробляється та аналізується сайт інституту як джерело інформації і засіб надання потрібних результатів користувачу при різних режимах роботи.

Ключові слова: CMS, Drupal, FTP, система управління контентом, джерело інформації, потік інформації.

ABSTRACT

The text part of the bachelor's thesis contains: 55 pages, 23 figures, 4 appendices and 15 sources.

The purpose of the work is to develop the ITS website and consider it as a source and receiver of information, building and organizing the interaction of site users: visitors, content managers, web designers and administrators, their role and organization.

In today's world, the Internet occupies a significant part of everyone's life, so neglecting to promote their services or products online becomes absurd. This paper develops and analyzes the site of the institute as a source of information and a means of providing the desired results to the user in different modes of operation.

Keywords: CMS, Drupal, FTP, content management system, information source, information flow.

ЗМІСТ

1. Аналіз веб-сайтів, методи та засоби їх побудови	7
1.1 Класифікація сайтів	7
1.2 Основні методи для розробки веб-сайтів	13
1.3 Інформаційна архітектура.....	18
1.4 Можливості CMS	19
1.5 Вибір CMS	23
2. Джерела та потоки інформації при встановленні та супроводженні сайту	27
2.1 Джерела інформації	27
2.2 Потоки.....	27
2.3 Опис джерел інформації	30
3. Практична реалізація	33
3.1 Встановлення CMS	33
3.2 Базові налаштування	35
3.3 Налаштування засобів для управління контентом	39
3.4 Розробка теми.....	49
3.5 Надання певних можливостей для супроводження сайту.....	52
3.6 Розгортання сайту в мережі.....	54
ДЖЕРЕЛА	56
ДОДАТОК А.....	57
ДОДАТОК Б.....	59
ДОДАТОК В	60
ДОДАТОК Г	61

РОЗДІЛ 1

1. Аналіз веб-сайтів, методи та засоби їх побудови

1.1 Класифікація сайтів

На даний момент в Інтернеті налічується близько 2 млрд сайтів. Саме стільки приблизно використовується доменних імен і їх кількість з кожним днем зростає[2]. При такій кількості сайтів будь-яка класифікація стає умовною. Хоча по деяким параметрам це зробити можливо. В нашому випадку ми будемо класифікувати сайти за такими параметрами, що можуть використовуватися у сайтах ВНЗ та різних освітніх закладах. Таким параметрами є призначення та технології, що використовуються при побудові сайтів.

Класифікація веб-сайтів за призначенням

Персональна сторінка. Персональною сторінкою раніше називалися сайти, які дійсно складалися тільки з однієї веб-сторінки. Сьогодні до цього типу вже належать всі багатосторінкові ресурси, зроблені однією або декількома особами, що хочуть чимось поділитися з читачами, про щось їм розказати. Сюди можна віднести численні блоги, викладені на платних хостингах.

У освітній сфері це можуть бути будь-які блоги викладачів та студентів, що діляться власними знаннями, пишучи освітні статті, які викладають на особистих веб-ресурсах.

Інформаційний сайт. Це досить великий віртуальний масив інформації. Інформація може бути по одній тематиці чи по різних, може бути багато тематичних розділів, чи навіть декілька самостійних проектів. Дані веб-сайти створюються таким чином, щоб для користувача це було основним джерелом інформації і нагадувало енциклопедію чи спеціалізований журнал по тематиці сайту. Суттєвою особливістю даного виду сайтів є наявність гіперпосилань на

веб-ресурси. Це створює зручну роботу користувача з інформаційним сайтом[1].

Прикладом такого ресурсу може бути бібліотека якогось ВНЗ, де знаходиться усі матеріали для студентів та викладачів у електронному вигляді, а також архіви з дипломними роботами чи дисертаціями.

Сайт-візитка. Це сайт окремої людини або невеликої компанії. Ресурс такого типу, як правило, містить від декількох сторінок, наприклад, сайти підприємців малого бізнесу на яких вказуються діяльність, ціни, контакти і т.д., до великих розгалужених сайтів із засобами онлайн зв'язку, з форумами та гостьовими книгами[3]. Через це, все призначення сайту зводиться до реклами власних послуг.

До даної групи можна віднести сайти викладачів, на яких вказані біографічні дані, навчальна та наукова діяльність, виховна діяльність, методичний матеріал.

Корпоративно-інформаційний сайт. Він необхідний фірмам, компаніям, організаціям усіх форм власності для забезпечення іміджевої присутності їх у мережі Інтернет. Сайт призначено для надання деталізованої інформації про фірму, організацію, компанію та інше. Зміст сайту пов'язано з професійною діяльністю.

Так для сайту ВНЗ це інформаційне обслуговування абітурієнтів, студентів, викладачів, співробітників, випускників і просто відвідувачів цього сайту. На такому ресурсі може бути інформація про історію інституту, його педагогічний склад, події, навчання, студентське життя з багатьма підрозділами, науку, видання, вихід до бібліотеки та інше. З активних компонентів використовують гостьові книги, форми, спеціалізовані форуми, чати та інше. З графічних засобів широко використовується текст з графічними ілюстраціями, фотогалереї, мультимедійні засоби та інше. Звичайно активна робота ведеться з абітурієнтами – надаються презентації кафедр, коментуються дні відкритих дверей, ведеться супроводження приймальної комісії та інше.

Корпоративно-інформаційний сайт є сучасним засобом організації роботи різних структур. Дизайн такого Веб-сайту звичайно досить простий і відповідає фірмовому стилю. В більшості випадків такий сайт має функції CMS і дозволяє певному рангу співробітників вводити та змінювати інформацію своїх розділів. Також такий сайт повинен мати засоби обмеження доступу до інформації.

Сайти, а точніше програмні Веб-системи для дистанційного навчання.

Одним з перших таких ресурсів був WebCT, але найбільшу популярність отримав MOODLE. Ці системи мають три типи користувачів: гість, студент (учень), викладач. Кожний з них має свої права. Як і у всіх CMS є реєстрація нових відвідувачів з підтвердженням на особистий email, викладач може використовувати ресурс для введення своїх лекцій, вставляти різні види тестів, на кожний курс лекцій уставлено форум і багато інших можливостей[1].

Класифікація сайтів по технологіях, що використовуються

Всі сайти в мережі можна розділити на дві великі групи: статичні і динамічні. З точки зору відвідувача сайту часто не важливо, на якій сторінці він знаходиться, статичній чи динамічній, іноді навіть важко точно це визначити. Розглянемо цей поділ з точки зору розробника сайту.

Статичний сайт.

Статичний сайт – сайт, що складається і статичних сторінок, написаних мовами розмітки (HTML, HTM, XHTML)[4]. Ці веб-сторінки цілком зберігається на сервері і показується відвідувачу у своєму незмінному вигляді (слід врахувати, що статична сторінка може містити деякі змінні елементи, наприклад банери, проте вона все одно залишається статичною).

Сторінки можуть містити в собі текст, зображення, мультимедійний вміст у вигляді HTML розмітки. Всі зміни на сайт вносяться в вихідний код документів (сторінок) сайту, для чого необхідно мати доступ до файлів на веб сервері.

Тобто дані сайти – це просто набір різних сторінок, що знаходяться на сервері.

Переваги статичних сайтів:

- Швидко завантажується, порівняно з динамічними веб-сайтами[6].
- Легко змінити зовнішній вигляд будь-якої конкретної сторінки, не вплинувши на вигляд інших сторінок.
- Нескладно додати на сайт нову сторінку, скопіювавши і виправивши файл з існуючої сторінки.
- Мінімальні вимоги до веб-сервера і мінімальне навантаження на нього[4]. Тому сайт буде працювати на будь-якому сервері хостингу, навіть з дуже обмеженими можливостями.
- Сайт можна переглядати локально, не використовуючи додаткове програмне забезпечення. Веб-майстру зручно робити сторінки, не виходячи в Інтернет.
- Простота переносу на інший сервер або локальний комп'ютер.
- Забезпечення покращеної безпеки над динамічними веб-сайтами (динамічні веб-сайти загрожують атакам веб-оболонки, якщо є вразливість)
- Менше або взагалі не залежить від таких систем, як бази даних або інші серверних додатків
- Економія витрат від використання хмарного сховища[5]

Недоліки статичних сайтів:

- Складно вносити зміни в структуру і зовнішній вигляд сайту. Наприклад, якщо треба змінити меню сайту, логотип або структуру сторінок, то для цього необхідно змінити вміст усіх сторінок, оскільки HTML-код цих елементів продубльований на кожній сторінці.
- Неможливість динамічної генерації вмісту[4].
- Система не гарантує єдиний стиль сторінок сайту - кожна сторінка фактично створюється заново і будь-яка помилка при зміні файлів, що містять сторінки, призводить до того, що різні сторінки сайту виглядають по-різному.

- Треба розробляти чи знаходити програмні модулі для реалізації активних компонентів (коментарі та відгуки користувачів, голосування, форум, чат та інше).
- Досить складно забезпечити розділення прав доступу до вмісту сайту.
- Для наповнення сайту інформацією необхідно отримувати доступ до файлів-сторінок за допомогою FTP, або стороннім веб-скриптів, що дозволяє редагувати сторінки[4].

Динамічний сайт.

Динамічний сайт - це сайт, який складається зі змінюваних веб-сторінок.

Вихідний код таких веб-сторінок зазвичай генерується під час обробки HTML файлу інтерпретатором якої-небудь мови програмування. Мови програмування на стороні сервера використовуються різні, найбільш часто зустрічаються: Java, PHP, Ruby[6].

Розглянемо процес генерації динамічної сторінки докладніше.

Якщо статична сторінка не зазнає практично ніяких змін і потрапляє на екран до відвідувача у своєму "первозданному" вигляді, то динамічні сторінки, у тому вигляді, що показується користувачу, на сервері не існує. Вона збирається кількома різними способами з даних, що зберігаються на сервері, і лише після цього показується відвідувачеві.

Методи збирання динамічних сторінок.

Першим варіантом може бути об'єднання декількох (двох і більше) окремих частин в одну сторінку - це найпростіший спосіб генерації.

Другим варіантом є заповнення шаблонної сторінки будь-якою інформацією, що зберігається окремо.

І третій, що використовується найчастіше – комбінований метод.

Динамічний сайт - сайт, в якому більшість або всі сторінки якого створюються динамічно.

Переваги динамічних сайтів:

З появою мов програмування, що виконуються на стороні сервера, з'явилася можливість вносити зміни в дані, що відправляються відвідувачеві. І це дало наступне:

- Розділення інформації та дизайну.
- Спрощення модифікації та оновлення сторінок.

Можливість змінювати контент, реагуючи на дії відвідувача. Використання динамічних сторінок дозволяє зберігати якийсь шаблон дизайну, в який, в залежності від сторінки, на яку зайшов відвідувач, розміщується необхідне наповнення. Такий варіант дуже зручний, адже для всього сайту створюється один або декілька шаблонів, і всі зміни дизайну, які потрібно зробити на сайті, виконуються тільки з ними.

З точки зору організації роботи і розподілу праці, варіант створення сайту на основі шаблонів практично ідеальний. Людина, яка відповідальна за дизайн сайту не зможе втрутитися в процес наповнення сторінок контентом, і за всі недоліки в дизайні буде нести відповідальність тільки він. І навпаки, той, хто займається інформаційним наповненням сайту, не зможе порушити дизайн сайту. Кожен займається своєю справою, не заважаючи іншим. Це прискорює роботу і знижує витрати.

Розділення інформації та дизайну на сайті дозволяє прискорити процес оновлення і наповнення сайту, тому що не вимагає від людей, що виконують ці операції, знань в областях html-розмітки, графіки і т.д. Тобто можна навіть підготувати дизайн сайту (замовити у фірми, яка надає такі послуги), а потім робити наповнення сайту власними силами, не витрачаючи кошти на постійне залучення сторонніх або включення у свій штат фахівців з дизайну, що треба було б робити при статичній організації сайту. Та й часові витрати на оновлення значно скорочуються.

Недоліки динамічних сайтів.

Першим недоліком є необхідність використання додаткових програмних засобів для побудови динамічного сайту. На статичному сайті всі сторінки вже готові, серверу залишається лише показати їх відвідувачу, а на динамічному сайті необхідно вносити до них якісь зміни, для цього потрібно відповідні програмні рішення.

Другим недоліком є підвищення вимог до апаратних потужностей серверних систем. Цей недолік безпосередньо впливає з попереднього, тому що тепер сервера потрібно ще виконати якусь програму для модифікації сторінки сайту, а тільки потім видати її відвідувачеві.

Особливо помітною ця проблема стає на сайтах з великою відвідуваністю. Часто в таких випадках доводиться робити додаткові оптимізації коду для нормальної роботи сайту. Отже, вартість послуг, з утримання такого сайту, набагато вища, ніж статичного.

Третім недоліком, що також впливає з першого, є складність великих структурних змін сайту пов'язаних зі зміною програмного забезпечення.

І четвертий недолік - вразливість таких сайтів значно більша за рахунок програмного забезпечення, яке використовується.

Висновок. Роблячи підсумок, для сайту кафедри чи інституту найбільш точно підходить структура корпоративно-інформаційних сайтів з динамічним наповненням. Дані сайти будуть створенні на основі однієї з популярних систем керування змістом, яку ми оберемо виходячи із наших вхідних даних.

1.2 Основні методи для розробки веб-сайтів

Можна виділити 5 основних методів розробки

1. За допомогою HTML/CSS/JavaScript (ручна)
2. За допомогою програмних засобів розробки сайтів

3. За допомогою інструментальних систем, таких як CMS
4. З використанням популярних на цей час фреймворків
5. На SaaS-платформах у CLOUD

За допомогою HTML/CSS/JavaScript (ручна)

З початком розвитку мережі Інтернет це був найпоширеніший метод. Він і досі користується популярністю для односторінкових сайтів. Але цей метод має суттєві недоліки при створенні багатосторінкових сайтів. Одним з них є те, що сайти побудовані даним методом, без використання фреймворків, в своїй більшості статичні і складно масштабовані.

За допомогою програмних засобів розробки сайтів

На даний момент існує безліч готових рішень для більш швидкої і зручної розробки сайтів. Вони надають можливість генерувати HTML код, розробляти сайт у візуальному режимі і мають багато інших можливостей.

Виділимо декілька інструментальних систем для розробки HTML:

- **Design-based editor** - програми, що мають у своєму складі візуальні редактор, які автоматично формують необхідний HTML-код, дозволяючи розробляти веб-сторінки в режимі WYSIWYG;
- **Code-based editors** - програми, які надають редактор і допоміжні засоби для автоматизації написання коду.

Розглянемо найбільш популярні design-based редактори:

Microsoft FrontPage - є класичним WYSIWYG-редактором, в якому, однак, є можливість ручного редагування коду, простий в засвоєнні і зручний веб-редактор для проектування, підготовки і публікації веб-сайтів. Інтерфейс програми багато в чому нагадує Microsoft Word, що дозволяє швидко засвоїти роботу навіть початківцям, які знайомі з основами роботи в MS Word[7]. Одним з основних переваг програми є велика кількість наявних шаблонів, що дозволяють користувачеві вибрати необхідний шаблон, після чого можна приступати безпосередньо до наповнення сторінки контентом. Процес створення HTML-сторінки нічим не відрізняється від створення звичайного текстового документа в Microsoft Word.

Adobe DreamWeaver – можливо найкращий візуальний редактор. Він надає спрощений процес створення й програмування динамічних веб-сайтів, а також

керування ними завдяки інтелектуальній технології програмування. Також за допомогою редактора можна створювати інтерактивні веб-сайти, а потім адаптувати їх відповідно до потрібного розміру екрану[8]. DreamWeaver на кілька кроків випереджає інші редактори, що використовують технологію WYSIWYG, у першу чергу завдяки тому, що генерує «чистий» HTML-код.

Популярні code-based редактори:

Visual Studio Code - редактор вихідного коду, розроблений Microsoft для Windows, Linux і macOS. Позиціонується як «легкий» редактор коду для кроссплатформенної розробки веб-додатків. Включає в себе відладчик, інструменти для роботи з Git, підсвічування синтаксису, IntelliSense і засоби для рефакторинга. Розповсюджується безкоштовно[9].

Sublime Text - підтримує плагіни на мові програмування Python. Розробник дозволяє безкоштовно і без обмежень ознайомитися з продуктом, проте програма повідомляє про необхідність придбання ліцензії[10].

PhpStorm - інтелектуальний редактор для PHP, HTML і JavaScript з можливостями аналізу коду на льоту, запобігання помилок в коді і автоматизованими засобами рефакторинга для PHP і JavaScript. Є повноцінний SQL-редактор з можливістю редагування отриманих результатів запитів[11].

Atom - безкоштовний текстовий редактор з відкритим вихідним кодом для Linux, macOS, Windows з підтримкою плагінів, написаних на Node.js, і вбудованих під керуванням Git[12].

Brackets - вільний текстовий редактор для веб-розробників. Brackets орієнтований на роботу з HTML, CSS і JavaScript. Ці ж технології лежать в основі самого редактора, що забезпечує його кроссплатформенність. Сумісний з операційними системами Mac, Windows і Linux. Brackets створений і розвивається Adobe Systems під ліцензією MIT License та підтримується на GitHub[13].

Саме на редакторах, що направлені на розробку через написання коду і проходить більшість сучасної складної розробки. Важливо зазначити, що code-based редактори не є методом розробки, а лише інструмент для виконання цієї розробки і наведені вони тільки як прикладу і загального розуміння.

За допомогою інструментальних систем, таких як CMS

Щоб створити динамічний сайт, можна скористатися вже існуючими системами, які і називаються системами керування вмістом (CMS). Великою перевагою даного методу є зменшення часу розробки та коштів на цю розробку

порівняно з самописними системами. Але гнучкість при власноруч написаній системі значно краща, не містить в собі нічого зайвого і залежить тільки від потреб замовника.

Цей спосіб на даний момент є основним і найпоширенішим для створення складних багатосторінкових веб-ресурсів

CMS - це програмне забезпечення, яке може використовуватися для управління створенням і модифікацією цифрового контенту[14].

Будь-яка система керування вмістом працює у парі з якоюсь СУБД. І тому для роботи з CMS потрібен веб-сервер. Це насамперед ускладнює розробку і додає значні часові затрати.

Проте людський ресурс при розробці бюджетних сайтів не дуже великий. Через відділення дизайну від контенту, вам потрібно лише замовити роботу спеціаліста по встановленні та налаштування системи, а наповнювати сайт можна і своїми силами.

Дуже позитивним проявом даного методу є також можливість «з коробки», тобто відразу, без лишніх затрат, зареєструватися користувачу та залишати коментарі на сайті. CMS вже «вміє» редагувати сторінки і управляти їх структурою, управляти користувачами системи і їх рівнем доступу, сортувати і фільтрувати різні об'єкти і багато іншого.

Майже всі CMS модульні, а модулі багатьох з них зібрані в комплекти (або редакції), призначені для тих чи інших видів сайтів. Є коробкові CMS для простих сайтів, для каталогів, для інтернет-магазинів, для блогів, для новинних порталів і для інших видів сайтів.

З використанням популярних на цей час фреймворків

Фреймворк - це програмний продукт, який служить основою для сайту, але зазвичай не містить в собі готових програмних модулів для реалізації конкретних бізнес-процесів. Висловлюючись технічною мовою, фреймворк - це більш низькорівневе рішення, ніж CMS. Розробники, при створенні сайту на фреймворку, створюють не тільки публічну частину сайту, але і проектують базу даних, розробляють алгоритми для модулів системи, а також створюють адміністративний інтерфейс для управління проектом. Необхідність серйозних витрат на програмування робить розробку більш дорогим, але і результат виходить більш індивідуальним.

Спілкування з фреймворком відбувається завдяки його API (інтерфейс програмування), що складається зі спеціальних функцій та інших специфічних конструкцій, описаних в офіційній документації.

Цей вид платформ використовують майже всі великі веб-проекти (на CMS побудована лише дуже мала частина дійсно серйозних проектів), а також переважна більшість веб-додатків і веб-сервісів (коробкових рішень для унікальних бізнес-процесів просто не існує, а використання не дуже відповідних CMS в якості основи для кастомізації дуже ускладнює розробку).

На сьогоднішній день, фреймворків безліч. Для одного тільки мови PHP їх існує кілька десятків (всім відомі Yii, Laravel, Symphony, CakePHP і ін.). Також існують фреймворки і для інших мов, які застосовуються в вебі: CSS (Twitter Bootstrap, Zubr Foundation), JavaScript (AngularJS, Vue.js), Python (Django), Ruby (Ruby-on-Rails).

Великим плюсом фреймворків є повна свобода творчості. Тобто створення веб-ресурсу за допомогою фреймворка - це написання його коду з нуля, а не користування «тим, що дають», як у випадку з CMS. Тож розробка тепер полягає саме в написанні нового функціоналу, а не в урізанні наявного, як у випадку з CMS.

Варто зазначити, що зробити ресурс, розроблений на базі фреймворка, здатним зберігати високу швидкість роботи при великій відвідуваності буде досить просто, на відміну від інших розглянутих систем.

На противагу цьому, слід зазначити, що розробка такого продукту забирає багато часу, тому що більшість функціоналу доведеться створювати самостійно з нуля. Навіть банальні функції на кшталт кошика товарів і адмінки, які доступні в CMS «з коробки», на сайті, що базується на фреймворці доведеться прописувати вручну. Та й складність розробки збільшується в рази.

На SaaS-платформах у CLOUD

Найпростіший варіант – SaaS.

SaaS (software as a service - програмне забезпечення як послуга) - це спосіб поширення ПО на орендній основі. Тобто не треба купувати CMS і замовляти індивідуальну розробку, досить тільки конфігурувати виданий постачальником сайт під свої завдання і щомісяця (або щорічно) вносити абонентську плату.

SaaS-платформи ще часто називають «конструкторами сайтів». З прикладів - Тільда і Wix для простих сайтів, Shopify і inSales для електронної комерції. Основна перевага цього варіанту - досить хороший рівень якості за дуже невеликі гроші.

Але SaaS рішення недостатньо гнучкі: вони допускають тільки мінімальну настройку бізнес-логіки і мають обмежений функціонал по візуального оформлення. Якщо вимоги до оформлення або до функціональності проекту

досить високі, то «конструктор сайтів» за характеристиками вже може не підійти, так як реалізація деяких речей на цих платформах просто неможлива.

Висновок. Виходячи з вищезазначеного, для наших потреб, а точніше для сайтів інституту та кафедри найбільше підходить розробка на CMS.

Ручна розробка не підходить, бо в нас як мінімум має бути адміністративна панель для легкого доступу до редагування та написання контенту менеджерам.

Design-based редактори – це та ж сама ручна розробка, але у набагато спрощеному виді.

SaaS-платформи насамперед зобов'язують оплати і недостатньо гнучкі та масштабовані. А наші ресурси будуть достатньо громіздкі.

І залишається, здавалося б найкращий варіант – взяти популярний фреймворк і написати всю логіку з нуля. Але в цьому, як було сказано дуже величезний мінус і в рамках дипломного проекту у мене не вистачить часу для написання повноцінного сайту для нашого інституту.

Тож CMS – кращий вибір для нас. Багато потрібного функціоналу вже готово, залишається його допрацювати та оформити.

1.3 Інформаційна архітектура

Інформаційна архітектура — це перш за все ефективне управління вмістом, чітко визначені процедури та інформаційна політика сайту.

Інформаційна архітектура займається принципами систематизації інформації та навігації по ній з метою допомогти людям більш успішно знаходити і обробляти потрібні їм дані.

Існує багато визначень інформаційної архітектури:

1. Поєднання схем організації, предметизації і навігації, реалізованих в інформаційній системі.
2. Структурний проектування інформаційного простору, що сприяє виконанню завдань і інтуїтивному доступу до вмісту.
3. Мистецтво та наука структурування та класифікації веб сайтів та інтразв'язків з метою полегшення користувачам пошуку інформації та управління нею.
4. Розвивається дисципліна і спільнота практиків, що ставить своїм завданням поширення принципів проектування та архітектури на цифрових просторах[15].

Можливість знайти інформацію — вирішальний фактор для зручності сайту в цілому. Якщо користувачі виявляються не в змозі знайти потрібні їм відомості, переглядаючи веб-сторінки, виконуючи пошук по сайту і задаючи питання, сайт треба вважати провальним. Але недостатньо лише орієнтуватися на кінцевого користувача при проектуванні. Необхідно врахувати завдання організацій і тих, хто управляє інформацією. Інформаційна архітектура повинна приводити в рівновагу потреби користувачів і завдання бізнесу. Велике значення мають ефективне управління вмістом, чітко визначені процедури та інформаційна політика сайту.

Висновок. Добре спланована інформаційна архітектура дуже корисна як споживачам, так і виробникам. Вперше звернувшись до сайту, споживачі можуть швидко зрозуміти його. Вони можуть швидко знайти потрібну інформацію, тим самим скорочуючи витрачений час на пошук інформації.

У ході дипломної роботи ми побудуємо та втілимо в життя правильну та зручну архітектуру для наших користувачів, аби пошук інформації не став перешкодою для викладачів, студентів чи просто користувачів сайту.

1.4 Можливості CMS

В основі CMS лежить трьохланцюгова архітектура клієнт-сервер. Така архітектура розбиває процес обробки даних між:

- клієнтом;
- сервером додатків;
- сховищем даних (база даних).

На відміну від традиційної двохланцюгової архітектури тут присутній сервер додатків як проміжна ланка між клієнтом і сховищем даних[1].

Ця ланка, в першу чергу, продиктована вимогами додатків для підприємств, що взаємодіють з клієнтами (через Інтернет), партнерами (через Extranet) і власними працівниками (через Intranet). У відсутності сервера додатків більшість додатків виконується прямо на клієнтському комп'ютері, з якого клієнт посилає запити. При цьому для доступу до необхідних даних клієнт повинен знати, як саме вони організовані і де зберігаються. До того ж комп'ютери клієнтів повинні бути достатньо потужними для обробки даних зі сховища (така архітектура зазвичай називається "товстий клієнт"). На противагу їй трьохланцюгова архітектура, її ще називають "товстий сервер", а частіше - "тонкий клієнт", дозволяє уникнути посилення комп'ютерів клієнтів і завантаженості мережі через переміщення даних. Отримуючи запит, сервер додатків обробляє його, зв'язуючись з сховищем даних, в якому б місці

необхідні дані не знаходилися. Клієнт лише отримує результат у вигляді HTML-файлу.

Таким чином, сервер додатків є стандартизованою платформою для динамічної доставки контенту та побудови основних додатків. Треба зауважити, що самих серверів додатків може бути багато, а зв'язок з ними відбувається через веб-сервер.

Для створення і обслуговування динамічних сайтів можна скористатися спеціальними системами управління веб-контентом. Вони є програмним забезпеченням, що встановлюються на веб-сервері. Їх основним завданням є контроль контенту, що надходить на сайт, для забезпечення достовірності та своєчасності.

Якщо подивитися тепер на системи управління контентом в узагальненому вигляді, то можна відзначити, що їх складовими можуть бути компоненти управління контентом, сховища даних, серверів додатків. З них лише управління контентом є обов'язковою складовою, хоча різні виробники і тут пропонують різні можливості.

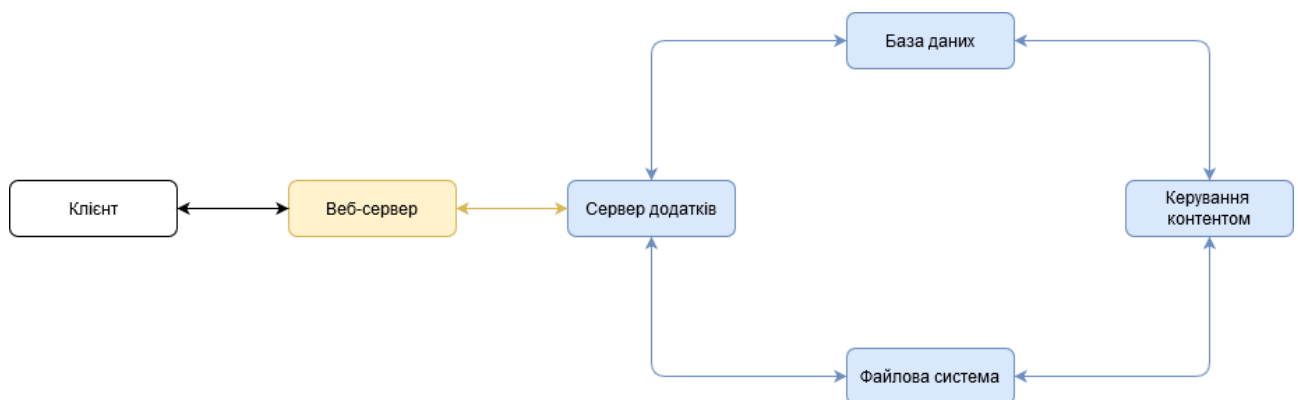


Рисунок 1.4.1 Типова архітектура CMS.

Основні функції CMS

Так, як систем управління веб-контентом дуже багато, вони можуть суттєво відрізнятися по направленню та структурі. Але все-таки, основні параметри можна визначити:

Управління сайтом

На цьому рівні відбувається розробка самого сайту, попередній перегляд і публікація підготовленого контенту. Тут розробляється зовнішній вигляд, готуються шаблони, розподіляються ролі користувачів і класифікація необхідної бізнес-інформації (наприклад, товари, ціни). Важливими

компонентами цього рівня є служби що підтримують своєчасність надходження необхідного контенту[1].

Розробка контенту

Контент — це ключовий компонент будь-якої CMS.

На етапі розробки контенту і починається життєвий цикл будь-якого матеріалу, що публікується на сайті. Тут відбувається створення, редагування та затвердження контенту, а роль системи полягає в автоматизації цих процесів. Завдання підтримки спільної роботи авторів, редакторів, програмістів та менеджерів повністю покладається на систему. Це завдання здійснюється завдяки розподілу контенту і дизайну.

Всі компоненти сайту, включаючи шаблони і наповнення, зберігаються у БД, пов'язаної з CMS. Система ж автоматично «спілкується» з БД, дозволяючи безлічі користувачів, які навіть не є технічними фахівцями, працювати над підготовкою контенту до публікації, включаючи перевірку його достовірності.

Доставка контенту

Після того, як контент повністю готовий до публікації, завдання системи — зрозуміти, що робити з контентом на сайті в залежності від прав користувача. У зв'язку з цим, одним з важливих компонентів даного етапу є персоналізація або розподіл профілів, щоб кожен користувач отримував тільки ту інформацію, яка відповідає його ролі.

Так, анонімний користувач зможе лише переглядати контент, але не створювати його. Авторизований користувач крім перегляду може і генерувати та відправляти контент на перегляд адміну. І в решті-решт, публікувати контент може лише адміністратор сайту.

Слід зазначити, що незважаючи на різноманітність систем управління контентом, всі вони дуже схожі тим, що зосереджуються на управлінні контентом, а не на його публікацію.

Оформлення Веб-сайту

Дуже швидко CMS перетворилися із просто систем управління контентом в повнофункціональні системи управління сайтом. При цьому популярні і широко розповсюдженні скрипти, які окремо використовувалися на сайтах, були об'єднані загальною програмою під єдиним інтерфейсом. В результаті отримали на 100% готовий до застосування повнофункціональний сайт — портал[1].

З розвитком технологій, інсталиувати CMS на будь-який підготовлений для цього сервер стало дуже просто. Навіть не підготовлений фахівець будь-якої професії зможе розгорнути свій веб-сайт для просування його в інтернеті.

Проте обрати модулі, пов'язати їх, розробити шаблони для сторінок і т.п. все одно доведеться програмісту. Наповнювати ж сайт, як і сказано раніше, може будь-яка людина без спеціальних знань.

Часто програмні можливості CMS дозволяють в повністю автоматичному режимі змінювати розташування виведення інформації на Веб-сторінці, вибирати схеми кольорів, дизайн із шаблонів, що є в наявності.

Будь-яка CMS надає так зване «ядро» і модулі для сайту. Ядро CMS — це програма, яка підпорядкована певному алгоритму введення і виведення інформації. Цей алгоритм у кожній CMS різний, свій, в відповідності з ідеєю і ціллю розробника. По-простому, це стандартний набір файлів системи та модулів. Саме модулі роблять систему придатною для користування. Вони додають новий функціонал, якого раніше не було. А для того, щоб сайт не виглядав як набір тексту з посиланнями є теми.

Це дає можливість користувачу вибирати CMS, що найбільш підходить. В зв'язку з цим, зовнішній вигляд інформації що виводиться (наприклад, новин), в кожній CMS різний, індивідуальний і змінити його досить складно (або неможливо, без кастомізації самого ядра). Тому звичайним користувачам, що хочуть зробити свій веб-сайт слід враховувати призначення системи для тієї чи іншої задачі, а не займатися її модернізацією для досягнення поставленої мети.

Універсальність і, багато в чому популярність CMS, залежить від кількості наявних модулів, можливості и простоти створення нових. Наявність, кількість, оновлення модулів є одним із критеріїв оцінки CMS. Модульна структура дозволяє розробникам охопити велике коло користувачів з різними вимогами до функціональності порталу, програмістам — створювати свої модулі під конкретні задачі чи замовлення, а користувачам — отримати зручність і можливість мобільно реалізовувати різні свої ідеї[1].

Більшість CMS, крім модулів, мають шаблону структуру, тобто тему, візуальну складову сайту. За допомогою шаблонів можна швидко змінити зовнішній вигляд всього ресурсу або окремих модулів. При цьому можна створювати і власні теми та поширювати їх у відповідних спільнотах.

Розробка нового свого шаблону досить трудомісткий процес. Використання стандартних шаблонів значно полегшує створення сайтів, але зменшує їх індивідуальність.

Значно полегшують роботу користувачів з CMS при введенні та редагуванні інформації вбудовані в них редактори (їх часто називають WYSIWYG).

CMS з одного боку дає широкі можливості відвідувачам-користувачам, а з іншого - адміністраторам, які керують ресурсами, адмініструють інформацію, керують загальним виглядом сайту, спілкуються з відвідувачами і клієнтами.

Більшість CMS забезпечені системою авторизації, і адміністратор може дозволяти, забороняти, обмежувати доступ до інформації, або частини інформації на своєму ресурсі окремим відвідувачам[1].

Висновок. Як видно з вищесказаного – CMS все таки найкращий вибір для нас. Через велику кількість готових рішень ми зможемо обрати найбільш відповідну нашим вимогам систему, а також збережемо дуже багато часу.

1.5 Вибір CMS

Так як ми визначилися з типом нашого сайту і методом його розробки, то будемо відштовхуватися від цього.

На даний момент, наш корпоративно-інформаційний сайт має такі основні вимоги:

- Безкоштовна
- Можливість публікації новин та інших матеріалів для студентів
- Можливість створення бази файлів і документів
- Форум для внутрішнього спілкування
- Швидка розробка

CMS бувають безкоштовні, платні та самописні.

Якщо при розробці корпоративно-інформаційного сайту стає ясно, що потрібна гнучка або унікальна функціональність, написання нової, власної системи управління контентом - найкраще рішення. При цьому безпека найвища, але висока вартість й дуже тривалі терміни.

Тож цей вид систем управління контентом ми відкидаємо за відсутністю часу та фінансів.

У більшості випадків найкращим рішенням буде покупка комерційної CMS. Вони не настільки поширені, як безкоштовні версії, тому їх уразливості невідомі хакерам. Крім того, вони надають широку функціональність, можливість інтеграції з корпоративними CRM і іншими системами. Також вони чудово масштабуються і мають гарну підтримку від розробників.

Це звичайно найкращий варіант, але ж знову фінансова частина питання не дає нам можливості реалізувати веб-сайт на платних CMS.

Але ж звичайно, існує безліч CMS, якими може скористатися кожен бажаючий. Навколо них розростаються цілі співтовариства відданих фанатів і

шанувальників. Для доопрацювання функціональності існує маса плагінів, які створюють користувачі. Вони прості і через це розробка не займе дуже багато часу. Проте є і недоліки - широка популярність таких систем привертає зловмисників, які використовують відомі уразливості для отримання несанкціонованого доступу до закритої інформації і перехоплення контролю над сайтом.

Виходячи з вищесказаного, ми будемо шукати безкоштовну систему для корпоративно-інформаційних сайтів, але з гарним захистом від уразливості. Другим фактором є те, що я буду надавати перевагу тим системам, якими я користувався, тобто з власного досвіду.

Якщо подивитися рейтинг CMS для корпоративних рішень на просторах СНД, то ми отримуємо таку статистику

ОТРАСЛИ ▼		КОРПОРАТИВНЫЙ САЙТ ▼		
#	Название	Балл ?	Тренд ?	Проекты ?
1	WordPress	37.84	—	8055
2	Joomla!	22.18	—	5014
3	MODX	17.50	+1	3194
4	Drupal	17.33	-1	2309

WordPress є найпопулярнішою майже по всіх «фронтах». Її поважають професіонали і дуже люблять новачки, так як це проста, ефективна і недорога система. І хоча WordPress більше підходить для блогів і новинних порталів, він відмінно справляється з веденням корпоративно-інформаційних сайтів. Систему WordPress вибирають за можливість інтеграції з іншими сервісами, велика кількість шаблонів і легкий візуальний редактор, який легко освоїти новачкові.

Я дуже люблю користуватися WordPress у своїх проектах, але задля справедливості скажу, що ця система підходить в першу чергу для невеликих сайтів-блогів, тому при роботі з великими корпоративними інформаційними порталами вона може бути недостатньо потужною. Тож нажаль, вона нам не підходить.

Joomla займає друге місце в рейтингу CMS для корпоративно-інформаційних сайтів. Її позиція в цьому топі абсолютно виправдана, так як вона має безліч переваг.

Перше - функціональність. Інструментарій можна збільшувати за допомогою різних розширень (плагінів, додаткових компонентів).

Друге - хороша безпека. У Joomla є розширена система багаторівневої авторизації користувачів і модераторів, тому зламати сайт і отримати конфіденційну інформацію дуже складно.

Третя перевага полягає в широкому виборі різних шаблонів, щоб змінювати дизайн сайту, окрім цього існує великий функціонал для створення власних макетів порталів. Взагалі Joomla відома тим, що дуже багато моментів можна підлаштувати під себе, відредагувати існуючі розширення.

Веб-майстрам також безумовно сподобається регулярний вихід оновлень. Не можна не згадати про те, Joomla взагалі створювалася, як CMS для корпоративного інформаційного порталу, тому його можна сміливо використовувати.

З цією CMS я також мав справу, хоча і декілька разів. І тут я можу сказати точно: «При всіх перевагах забувати про те, що ця система вимагає значних фінансових вкладень, на жаль, не можна. Тому Joomla теж нам не підходить.

MODX - входить в розряд CMF(**Content Management Framework**), тому відрізняється гнучкістю і надійністю. Система працює швидко і відрізняється зручною адмін-панеллю. Варто зауважити: хоча MODX поширюється безкоштовно, але розширення все-таки потрібно купувати.

Мені довелося користуватися даною системою пару разів. Через те, що це CMF, тобто каркас для проектування систем управління вмістом, розробка на цій системі займає дуже багато часу. І на мій розсуд, вона дуже важка для освоєння. І до того ж багато корисних плагінів є платними.

Не дивлячись на це, дану CMF, я можу рекомендувати і долучити до вибору нашої системи.

Drupal - open-source система управління контентом. Також Drupal використовується як каркас для веб-додатків (CMF), тож вона теж складна для розробки. Але це більш захищена і гнучка система, порівняно з іншими.

Також вона радує веб-майстрів багатьма елементами. По-перше, це багатофункціональна система управління, де можна розширювати можливості завдяки додаванню нових модулів. Багато корисних модулів є безкоштовними.

По-друге, на базі цієї системи налаштовується будь-який контент, тому дана CMS підходить для сайтів різної тематики, в тому числі і корпоративних та інформаційних.

По-третє, для кращої безпеки можна розділити адміністрування на різних модераторів. І звичайно, не можна не згадати про гарні SEO-налаштування.. Звикати до цієї системи доведеться досить довго.

Також Drupal може похвалитися потужною адмін-панеллю.

З Drupal, я знайомий давно. Так, ця система складна для освоєння і розробки як і MODX. Проте звичайному користувачу дуже складно розібратися в адмін-панелі MODX. Drupal же на порядок легше, тому тут ми надаємо перевагу йому.

Висновок. Дивлячись на наші вимоги, а саме: публікація новин, створення бази файлів і форуму всі розглянуті CMS підходять.

Проте WordPress нам не підходить через те, що дана система дуже популярна, а тому дуже схильна до взломів. Також дана система більше підходить для блогів, ніж для великих корпоративно-інформаційних сайтів.

Здавалосьь, Joomla – наш варіант, але вона потребує значних вкладень, на які у нас не має можливостей.

Обираючи між MODX та Drupal я надаю перевагу Drupal через просту адмін-панель, на відміну від першої.

Отже, наша CMS – Drupal!

РОЗДІЛ 2

2. Джерела та потоки інформації при встановленні та супроводженні сайту

2.1 Джерела інформації

K₁ веб-адмін (На початку: Обробка попередньо підготовленої інформації (*P*) і адаптація цієї інформації під можливості CMS

K₂ Системний адмін (КПІ), отримання хоста та IP, узгодження з вимогами CMS

K₃ Адмін сайту. Узгодження змісту (*P*), супроводження новин, зв'язок з (*A₁ – A_n*)

M₁ – M_n Контент менеджери сайту. Основні джерела інформації при супроводженні сайту

A₁ – A_n Адмін керівники напрямками діяльності підрозділу

B Підрозділ внутрішньої оцінки сайтів КПІ.

C Зовнішня оцінка доступу, попередження при втручанні (Google)

P Початкова інформація для сайту

U Користувачі сайту

2.2 Потоки

$\{F(K_1, K_2)\}$ - заявка на хостинг та IP, узгодження версії PHP та потрібного обсягу пам'яті для розміщення, підготовка інструкцій по супроводженню.

$\{F(K_2, K_1)\}$ - попередження про обмеження та втручання, отримання даних для аналізу сайтів.

$\{F(K_1, K_3)\}$ - передача доступів Адміну сайту (K_3).

$\{F(K_3, K_1)\}$ - інформація про встановлену CMS, замовлення про додавання нового функціоналу, перевірка можливостей замовлень IP та хосту і при необхідності зміна PHP.

$\{F(K_1, A_n)\}$ - надання прав контент менеджерам.

$\{F(K_3, M_n)\}$ - уточнення та корекція прав контент менеджерів.

$\{F(K_1, M_n)\}$ - передача прав контент менеджерам, надання інструкцій по роботі інформацією особливо графічної (підготовка інструкцій по роботі з "Views" та "Paragraph").

$\{F(M_n, K_1)\}$ – консультація з веб-адміном по роботі з сайтом, вимоги до додаткових можливостей CMS, замовлення на додаткові можливості CMS, наприклад встановлення YouTube чи встановлення у розділу Слайд шоу.

$\{F(K_1, P)\}$ - введення первинного варіанту сайту з перетворенням у формати даних CMS.

$\{F(K_1, D)\}$ – перетворення D у формат вибраної CMS.

$\{F(K_1, C)\}$ – дані по необхідним параметрам для служб Університету.

$\{F(C, K_1)\}$ - зв'язок з google.com.ua/webmaster, оперативний аналіз відвідувань сайту, боротьба з втручанням, відбивання вірусних атак та інше (Webmasterces - Resources to Track website@Search Performance).

$\{F(C, A_n)\}$ – дані по параметрам для оцінки наукового рівня співробітників кафедри.

$\{F(B, A_n)\}$ - попередження про стан сайту та параметри зовнішньої оцінки, передача керівництву кафедри даних про оцінку Університету сайту кафедри.

$\{F(B, K_1)\}$ – отримання даних для аналізу наукової роботи кафедри.

$\{F(U)\}$ – відвідування та запити користувачів сайту.

Потоки визначаються як:

1. рідкі ймовірностні і їх більшість
2. одноразові
3. події з певною частотою наприклад, магістерські та бакалаврські роботи
4. події з підвищеною частотою (новини)

Веб параметри по яким оцінюється розроблений сайт залежать від значення $\{F(U)\}$ і основною задачею при розробці є бажання зробити більшим $\max\{F(U)\}$.

Особливо це є важливим у період коли сайт виходить на ринок тобто у період вступної компанії. У цьому випадку дещо змінюються задачі що стоять перед сайтом і підходи до його побудови.

Актуальність: Під час вступної компанії і державні навчальні заклади мають риси конкурентної боротьби (боротьба за абітурієнта) і засоби маркетингу для їх існування є важливими. Значення Інтернету для молоді, що вибирає певні заклади освіти, досить високе і тому практично всі поважаючи себе заклади освіти мають свої сайти. Ці сайти у більшості випадків є інформаційними корпоративними і часто мають такі вади: надлишок чи недостатньо повне інформаційне наповнення, не чітко підкреслена особливості навчального закладу, не якісне відображення на мобільних пристроях.

Тому пропонується використати один із сучасних методів, що існує у маркетингу () і має назву Landing Page. Звичайно Landing Page це одно сторінковий рекламний сайт орієнтований на абітурієнтів з наданням інформації у ясній і доступній для абітурієнтів формі з можливістю швидкого доступу на мобільних пристроях.

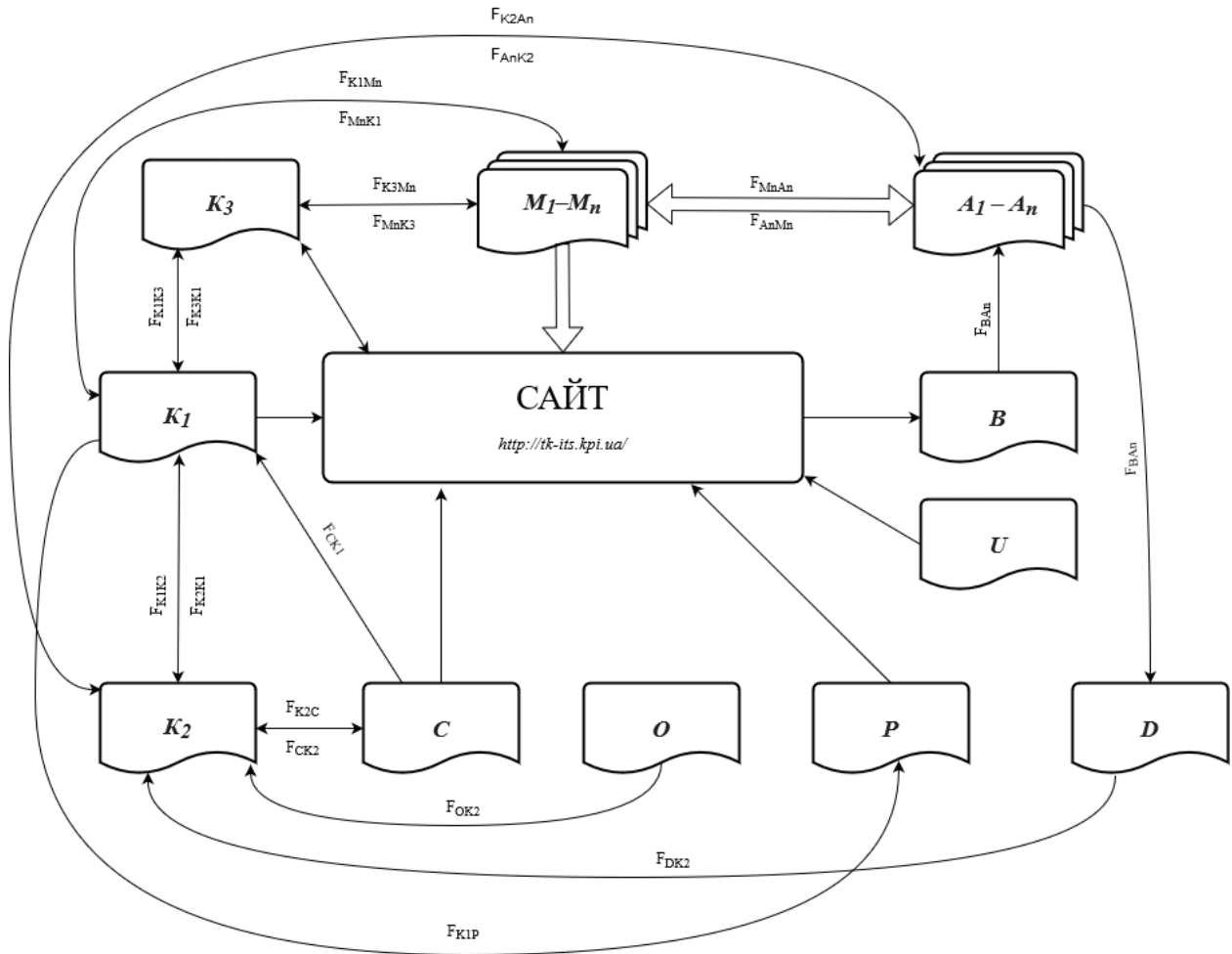


Рисунок 2.2.1 Потoki інформації на сайті інституту.

2.3 Опис джерел інформації

K1 – веб-адміністратор

Функції: надання вимог Адмінам Університетської мережі (K_3) в залежності від вибраної до використання CMS, по версії PHP та варіанта бази даних встановлення CMS, аналіз попереднього варіанту сайту D і узгодження з адміністрацією (A) розділів сайту, встановлення розділів на новий сайт та перезапис, перезапис з корекцією чи перекодування відповідної інформації з D на новий сайт; корекція графічного оформлення, підготовка інструкцій для супроводження, корекція сайту при потребі, надання адміністративних прав контент менеджерам, робота із сайтом при втручання на сайт;

Професійна підготовка: вміння роботи з FTP та MySQL, знання можливостей та досвід роботи з CMS (бажано виконати 2 лабораторну роботу курсу «Веб проектування» (можна в дистанційному режимі за адресою <http://info.kpi.ua> в розділі «Дистанційні навчальні курси ІТС»);

K2 - Адміністратор сайту

Функції: актуалізація інформації на сайті, при необхідності поточні зміни, представлення сайту на рівні Університету, отримання зауважень стосовно сайту від відповідних служб Університету, реєстрація в Google як адміністратор сайту та включення таких функцій як Google search.

Професійна підготовка: знання можливостей та досвід роботи з Drupal 8, володіння засобами супроводження Веб сайтів;

K3 - Адмін Університетської мережі

Функції: по замовленню підрозділу, яке передає K1, Адмін Університетської мережі надає хостинг (можливі додаткові вимоги до версії PHP та та версії бази даних) та IP адресу сайту;

M1, M2...Mn - Контент менеджери сайту

Функції Вони є основним джерелом інформації при створенні та поточній роботі сайту і мають права що дозволяють вводити та редагувати інформацію у виділені розділи сайту. Контент менеджери призначаються керівництвом каФедри;

Професійна підготовка: знання можливостей та досвід роботи з Drupal 8.

D – підготовлений інформативний матеріал для сайту чи попередній варіант сайту, який переноситься на новий хостинг та з новим CMS у нашому випадку на Drupal 8;

O - служби Університету , що аналізують стан сайтів , вони проводять періодичні навчання та надають результати свого аналізу адміністрації кафедри;

C- зовнішні засоби аналізу сайту так, наприклад , Google Seach Console аналізує ефективність сайту і висилає результати по Мейлу. При цьому надає кількість відвідувань, найбільш популярні сторінки, сторінки з найвищою ефективністю, як знаходять сторінки і звідки користувачі, звідки запити з найбільшою ефективністю, пристрої які використовують користувачі, найактивніші країни. Адміністратор сайту (K2) повинен зареєструвати сайт і свій Мейл Google administrator.

РОЗДІЛ 3

3. Практична реалізація

3.1 Встановлення CMS

Після всіх підготовчих етапів можемо приступити до практичної реалізації задуманого.

Для початку роботи з сайтом потрібно встановити CMS Drupal. Розробка всіх сайтів зазвичай ведеться локально, тобто на комп'ютері розробника і тільки після закінчення всіх робіт переноситься у мережу. Якщо ж після цього знадобиться вносити корективи у вигляд сайту, то треба зробити бекап на комп'ютер і проводити всі маніпуляції з сайтом локально.

Підготовка оточення

Що ж, отримати актуальну версію обраної CMS можна на офіційному сайті Drupal (<https://www.drupal.org>). На даний момент актуальна версія Drupal 8.8.6. Завантажуємо її на комп'ютер у будь-яку директорію.

Так як CMS використовує PHP у своєму ядрі, то нам потрібен сервер. Підійде будь-який, але я використовуватиму Open Server. В ньому апріорі встановлено весь потрібний конфіг для початку роботи.

Тепер треба перенести архів з CMS та видобути весь вміст у папку із назвою нашого майбутнього сайту та перенести її у потрібну папку нашого сервера. Після цього – запустити сервер.

Встановлення Drupal 8

В більшості випадків для початку роботи з файлами на локальному сервері потрібно в будь-якому браузері у адресному рядку вказати http://назва_сайту/.

Якщо все зроблено правильно, нас буде вітати початковий екран Drupal.

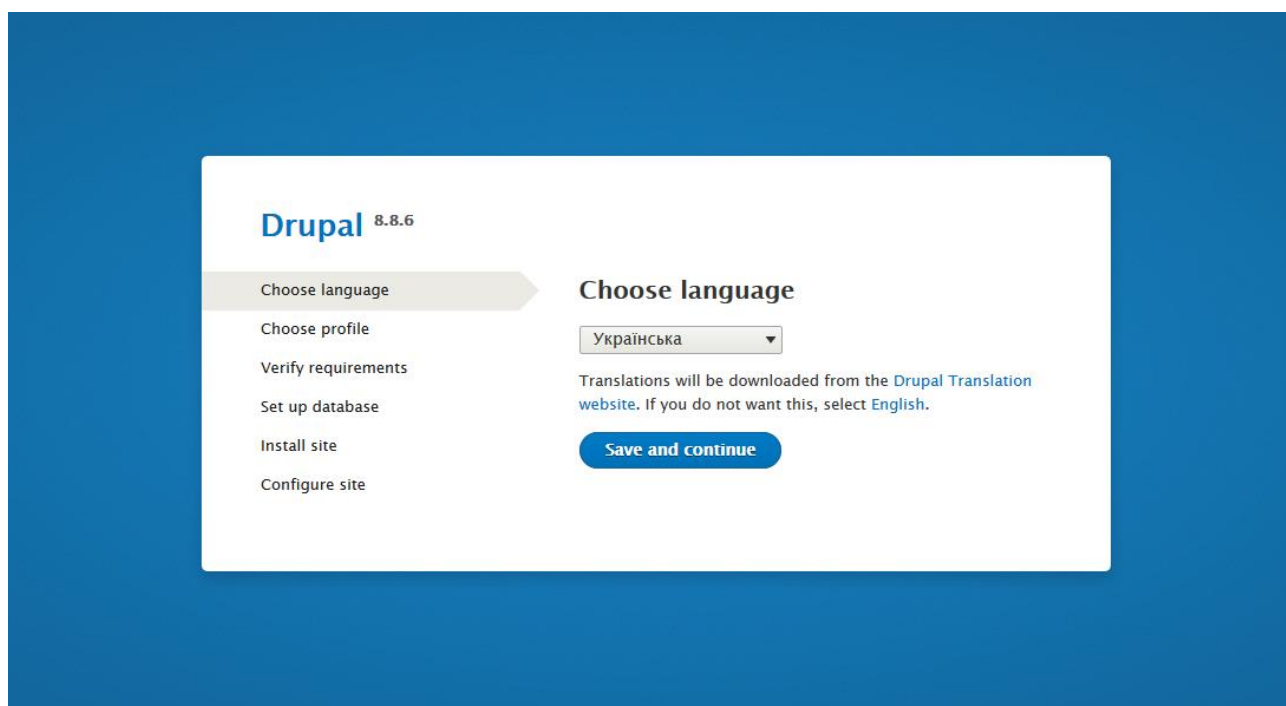


Рисунок 3.1.1 Початковий екран CMS Drupal.

Далі йде звичайний процес установки системи управління контентом. Обираємо зручну мову для сайту, стандартний набір інструментів та опцій, перевіряємо правильність конфігу.

Наш конфіг:

<i>Веб-сервер</i>	Apache
<i>PHP</i>	7.1.32
<i>Розширення PHP</i>	Увімкнено
<i>Генерування випадкового числа</i>	Прийняте
<i>Підтримка баз даних</i>	Увімкнено
<i>Обмеження пам'яті PHP</i>	1536M
<i>Файлова система</i>	Запис дозволено
<i>Бібліотека Unicode</i>	Розширення PHP Mbstring
<i>Файл налаштувань</i>	<code>./sites/default/settings.php</code> існує.

Так як у нас новий проект, то і бази даних немає. Тож тепер потрібно створити її. Базу даних обираємо MySQL, а для її адміністрування будемо використовувати phpMyAdmin.

Цей веб-додаток уже наявний у Open Server, тому переходимо в нього та створюємо базу. Зіставлення кодування з'єднання обираємо **utf8mb4_general_ci**.

Тепер продовжуємо встановлювати Drupal. Прописуємо назву бази, ім'я користувача та пароль.

На останньому етапі потрібно вказати назву сайту, пошту власника сайту та його логін і пароль.

Після закінчення встановлення системи, ми нарешті побачимо наш сайт.

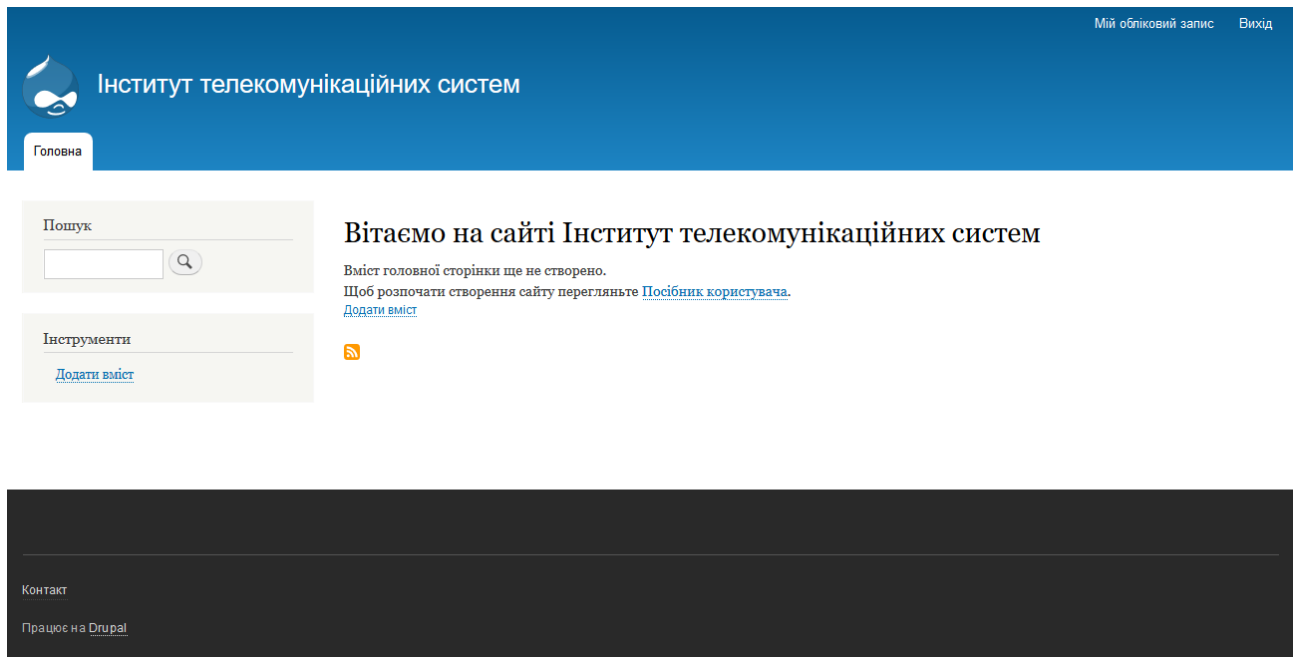


Рисунок 3.1.2 Початковий сайт ІТС.

Висновок. Сучасні CMS надають дуже зручний інтерейс для встановлення та початку використання своїх систем. Через це будь-яка людина без досвіду у розробці, після невеликої підготовки, зможе почати користуватися системою. А для розробників такий простий процес встановлення системи надає можливість скоротити час на саму розробку.

3.2 Базові налаштування

Для початкового налаштування нам потрібно вказати всі потрібні дані для уніфікації сайту, а також встановити необхідні модулі, доповнення, мови, бібліотеки. В кінці - оновити їх до останньої версії та позбутися конфліктів.

Всі модулі завантажуються з офіційного сайту CMS та додаються до конфігурації сайту шляхом видобування архіву з модулем у задану директиву сайту – *modules*. Після додавання розширення до структури сайту – вони не активні. Для активації потрібно увімкнути модулі через адмін-панель нашого сайту.

Якщо модуль використовує сторонні бібліотеки, то їх потрібно додавати у директорію *sites/all/libraries*.

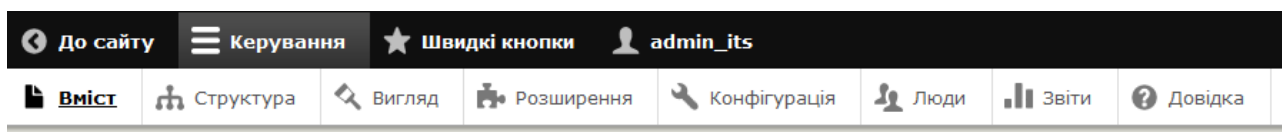


Рисунок 3.2.1 Адмін-панель сайту.

Першим розширенням, що ми встановимо буде Admin Toolbar. Даний модуль нам потрібний для зручного та швидкого користування адмін панеллю.

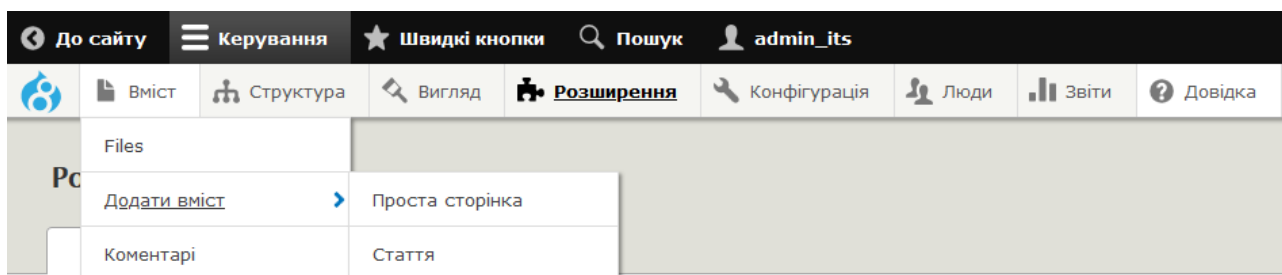


Рисунок 3.2.2 Адмін-панель сайту із встановленим модулем Admin Toolbar.

Так як сайт у нас український, нам потрібно зробити його багатомовним для іноземних студентів. Тож наш сайт буде розроблятися і перекладатися відразу на три мови – українська, англійська, російська.

Основною мовою CMS Drupal є англійська, тому нам потрібно додати дві інші. Це можна зробити в конфігурації сайту. А для того, щоб все почало працювати, потрібно в розширеннях увімкнути стандартні модулі для багатомовності: Configuration Translation, Content Translation, Interface Translation, Language. I

після цього «Конфігурація > Мова та регіональні стандарти > Мова і переклад матеріалу» увімкнути всі блоки, які будуть потребувати переклад: Вміст, Власний блок, Посилання власного меню, Термін таксономії, Користувач.

Тепер ми можемо перекладати додані нами матеріали, блоки чи навіть ядро сайту. Проте користувач досі не може змінити мову. Для того, щоб додати таку функціональність, встановлюємо модулі Language Switcher Dropdown та Language Icons. Перший додає функціонал перемикавання мови, а інший додає гарні іконки для обраної мови.

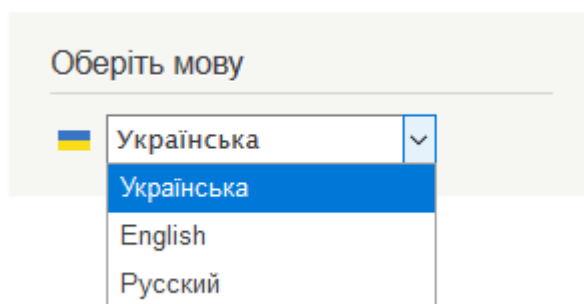


Рисунок 3.2.3 Віджет з мовами.

Модуль Superfish потрібен для випадаючого меню. Має гнучкі налаштування та можливість редагування вигляду меню. Цей модуль потребує сторонніх бібліотек, тож їх теж потрібно встановити. Це власна бібліотека superfish та бібліотека для анімацій easing.

Video Embed Field – модуль для швидкого підключення відео з YouTube на сайт.

Для зручного додавання та зберігання доданих файлів на сайт нам знадобиться модуль IMCE. Це завантажувач зображень/файлів, який підтримує особисті каталоги та квоти.

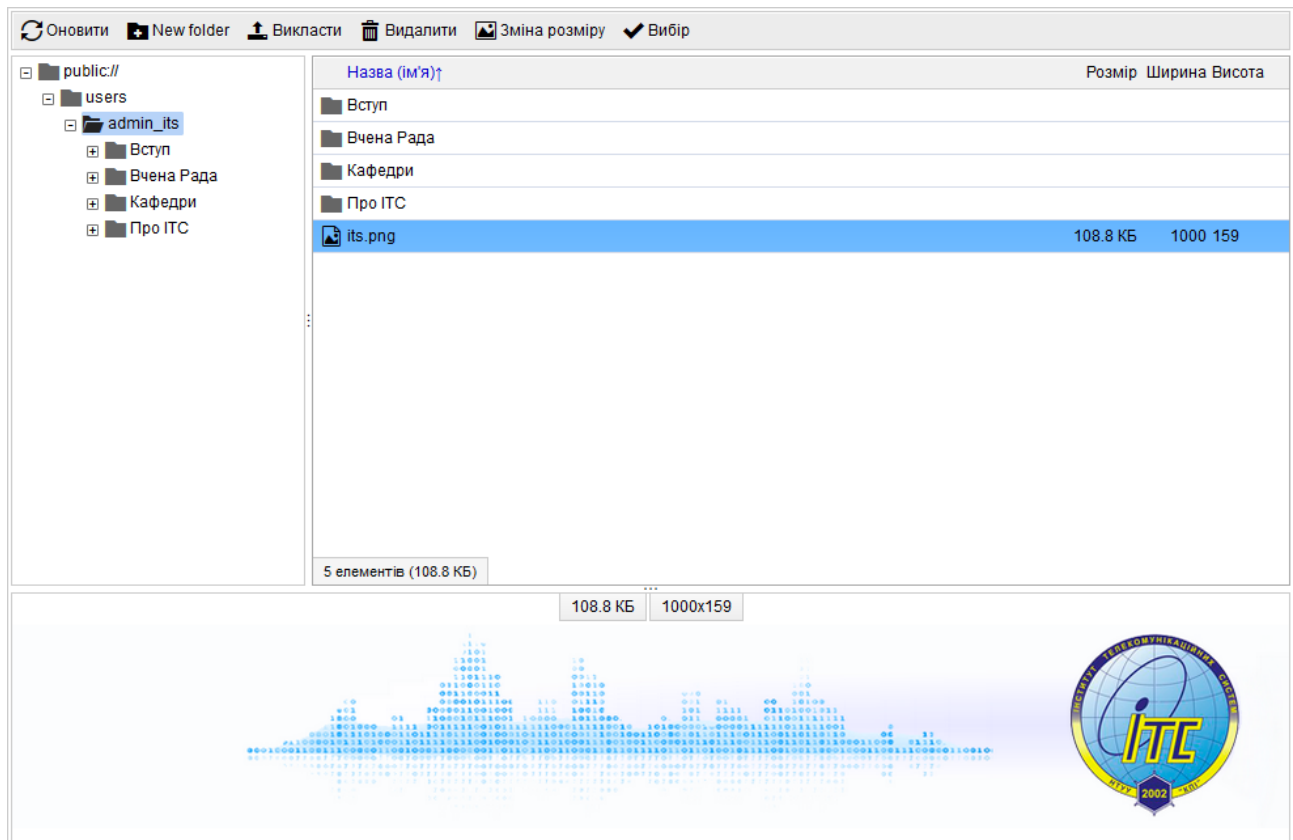


Рисунок 3.2.4 Вікно модуля ІМСЕ.

Paragraphs – найбільш корисний модуль. Він дозволяє створювати шаблони для розміщення контенту. Замість того, щоб розміщувати весь вміст в одному корпусі WYSIWYG, включаючи зображення та відео, кінцеві користувачі тепер зможуть вибирати між попередньо визначеними типами абзаців, незалежними один від одного. Типами абзаців можуть бути все, що завгодно, від простого текстового блоку чи зображення до складного слайд-шоу.

В подальшому завдяки цим плагінам буде розроблено гнучка та зручна у використанні система для користувачів сайту.

Для уніфікації сайту можна зробити брендування – дати правильну назву, встановити логотип та фавіконку сайту, а також задати короткий опис сайту.

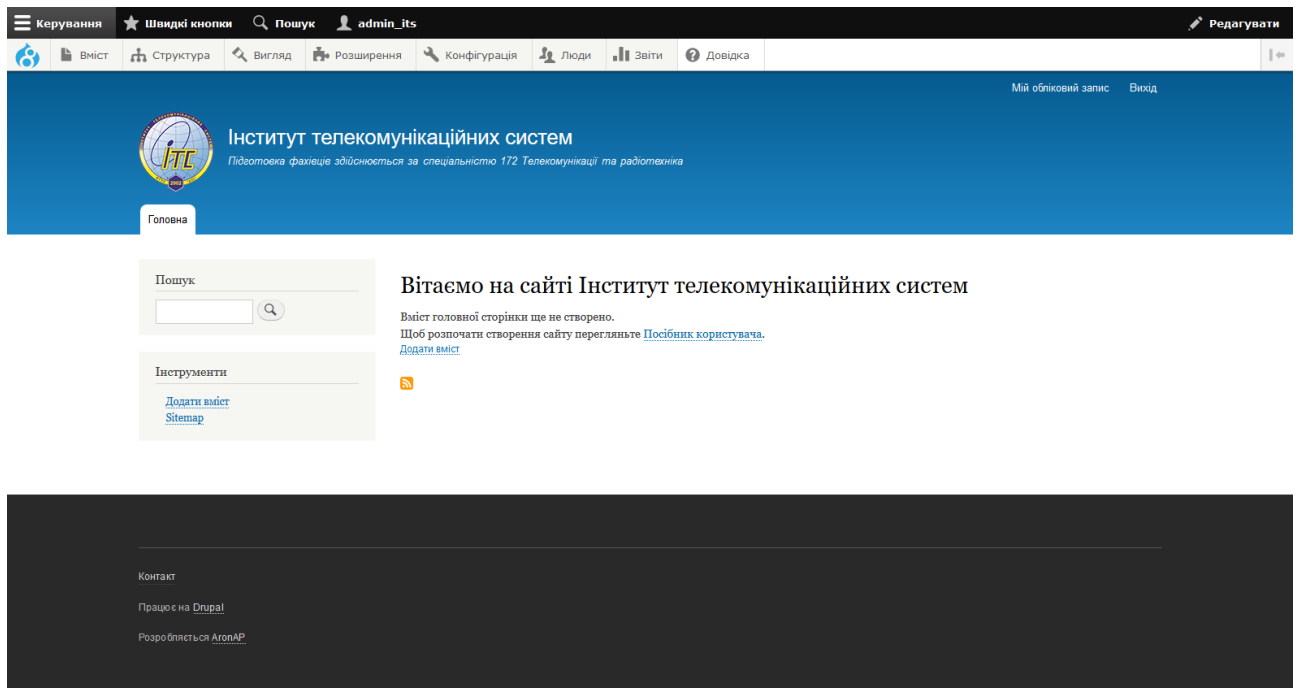


Рисунок 3.2.5 Вигляд сайту після брендування.

Висновок. Завдяки великій кількості вже існуючих модулів нам не потрібно створювати весь потрібний функціонал самостійно. Все, що потрібно – вивчити найбільш корисні модулі. Завдяки цьому розробка сайтів стає ще швидшою.

3.3 Налаштування засобів для управління контентом

Перш за все, нам потрібно зробити сайт зручним для користувачів, тобто додати новий функціонал для контент менеджерів.

Було створено та налаштовано дев'ять типів матеріалу, які можна розділити на чотири основні види:

1. Звичайні сторінки
2. Статті
3. Складні сторінки
4. Конструктор

НАЗВА (ІМ'Я)	ОПИС
Конструктор	Конструктор для швидкого додавання матеріалу
Проста сторінка	Використовуйте <i>прості сторінки</i> для статичного вмісту, наприклад, "Про нас".
Рішення Вченої Ради	Матеріали прийняті рішенням Вченої Ради
Стаття	<i>Статті</i> використовуються для чутливих до дати публікації матеріалів на зразок новин, прес-релізів чи дописів до блогу.
Новини вступникам	Сторінка для розміщення новин вступникам
Новини міжнародної діяльності	Сторінка для новин міжнародної діяльності
Новини навчального процесу	Сторінка для новин навчального процесу
Новини науки	Сторінка із колекцією новинами науки
Новини студжиття	Сторінка із новинами студжиття

Рисунок 3.3.1 Список всіх типів матеріалу.

Звичайні сторінки

Сюди відноситься тільки тип матеріалу *Проста сторінка*.

Нічого особливого. Це базова сторінка з двома полями — заголовок та контент.

Рисунок 3.3.2 Поля Простої сторінки.

Тип можна використовувати для звичайного тексту з картинками, але весь контент буде йти в одному потоці (один за одним)

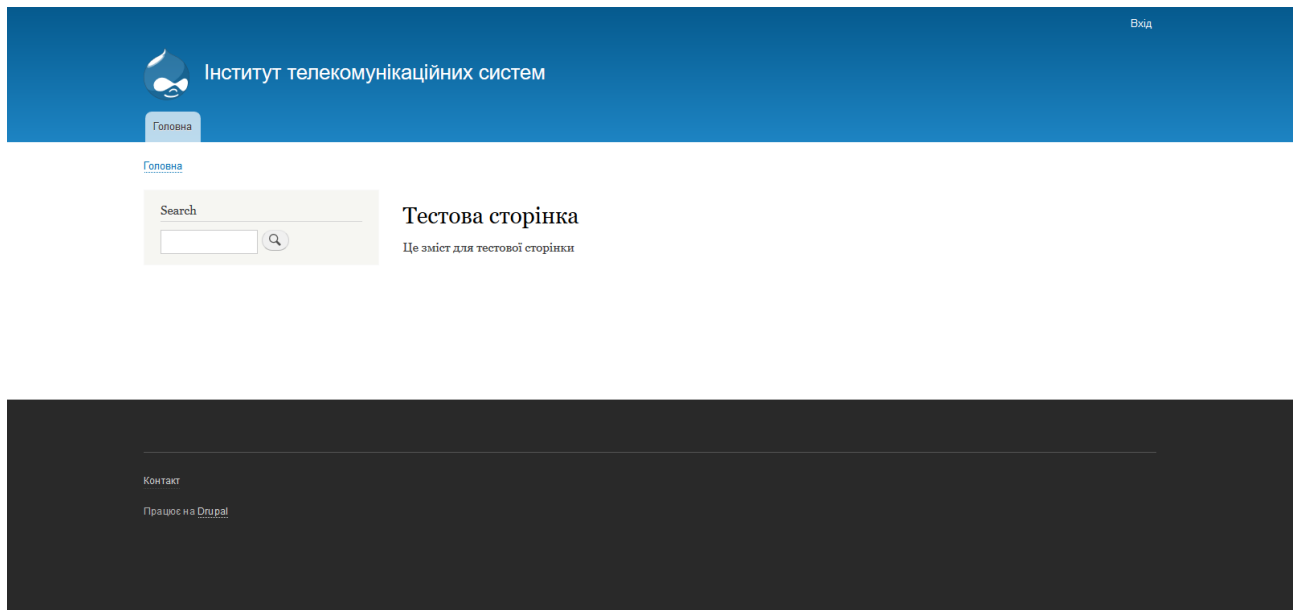


Рисунок 3.3.3 Вигляд Простої сторінки.

Статті

Тип матеріалу – Стаття.

Відрізняється від сторінки зображенням для попереднього перегляду в анонсах та можливістю залишати коментарі авторизованим користувачам.

Також у полях матеріалу є спеціальне поле для додавання зображення, що буде в анонсі і тегів для пошуку схожих статей.

[illegible]

Рисунок 3.3.4 Поля Статті.

Рисунок 3.3.5 Вигляд Статті.

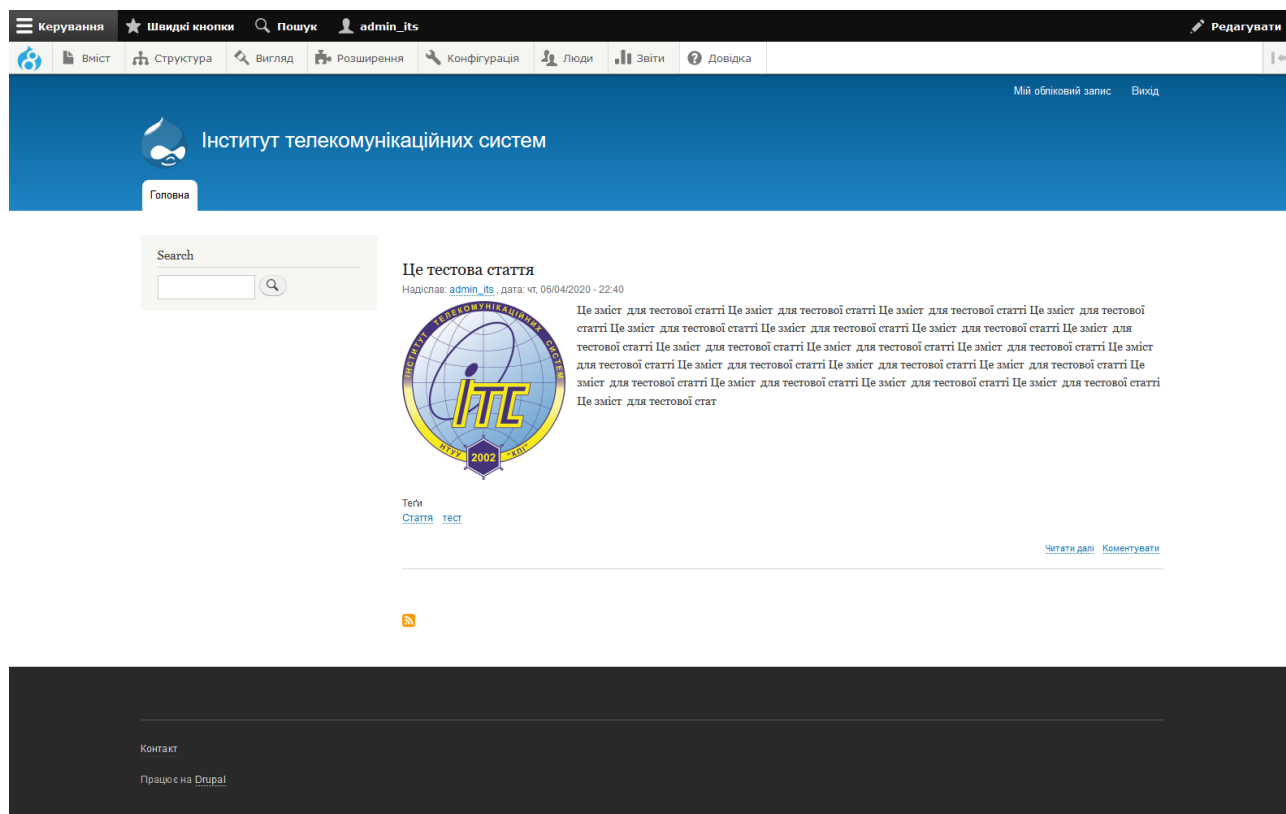


Рисунок 3.3.6 Анонс Статті.

Складні сторінки

До цієї категорії відноситься більшість типів: Рішення Вченої Ради, Новини вступникам, Новини міжнародної діяльності, Новини навчального процесу, Новини науки, Новини студжиття.

Ці сторінки схожі на звичайні, проте у них багато додаткових полів, що дають змогу зручно заповнювати інформацію і дотримуватися одного виду.

[illegible]

Рисунок 3.3.7 Поля Рішення Вченої Ради.

Інститут телекомунікаційних систем
Підготовка фахівців здійснюється за спеціальністю 172 Телекомунікації та радіотехніка

Головна

Пошук

Мій обліковий запис Вихід

ЗАСІДАННЯ ВЧЕНОЇ РАДИ ІНСТИТУТУ ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ КПІ ІМ. ІГОРЯ СІКОРСЬКОГО від 26.02.18 р.

Надіслав: [admin_its](#), дата: пт, 06/12/2019 - 00:50

На Вченій раді присутні члени Ради згідно „Листа присутності”, що додається до протоколу.

Порядок денний додається до протоколу.

1. СЛУХАЛИ: Доповідь молодого вченого Новоградської Р.Л. про стан науки в області телекомунікацій за лютий 2018р. [2018.pptx](#)

ВИСТУПИЛИ:

3.4. СЛУХАЛИ: Затвердження звітів аспірантів ІТС (1 та 2 рік підготовки в аспірантурі) за перший семестр 2017/2018 навчального року. Членам вченої ради було представлено інформацію завідувачів кафедр (зав. кафедрою ІТМ д.т.н., проф. глоба Л.С., заст. зав. кафедрою ТЗ, к.т.н., доц. Явіс В.С.) про виконання індивідуального плану роботи аспірантів ІТС першого року навчання за період 01.10.2017 р.– 28.02.2018 р. Звітну інформацію наведено у таблиці:

Назва кафедри/код спеціальності	Прізвище, ініціали наукового керівника	Прізвище, ініціали аспіранта	Відмітки про виконання індивідуального плану роботи аспіранта за 2017–2018 навч.рок	
ІТМ/172	Глоба Л.С.	Гасвий В.В.	Освітня складова: перелік складених дисциплін з позначенням кількості кредитів Інноваційні напрямки розвитку телекомунікацій та радіотехніки–1 Сучасні технології надання сервісів та обробки великих даних – 1 Математичні методи наукових досліджень в телекомунікаціях та радіотехніці–1 Іноземна мова для наукової діяльності–1 Філософська гносеологія та	наукова складова: виконана робота по дисертації, публікації, конференції, тощо Одна публікація в Український антарктичний журнал, Аналіз и усовершенствование инфраструктуры единой информационной среды НАНЦ, Гаевой В. В., Глоба Л. С., Мазанка Р. М., Омельченко А. И.

підготовки магістра.

ВИСТУПИЛИ:

- доповідач: перший заст. директора ІТС, к.т.н., доц. Правило В.В.

УХВАЛИЛИ:

1. Затвердити програму комплексного фахового випробування для вступу на освітньо-професійну програму підготовки магістра.

Голова Вченої ради ІТС М.Ю. Гільченко
Вчений секретар Р.Л. Новоградська

Дата прийняття: 26.02.2018
Номер протоколу: 2

Контакт

Працює на Drupal!

Розробляється Drupal

Рисунок 3.3.8 Поля Рішення Вченої Ради.

Конструктор

Конструктор – це найголовніший тип матеріалів. Він дає змогу конструювати унікальні сторінки, які захоче контент менеджер.

Цей тип зроблений за допомогою модуля Paragraphs. В ньому було розроблено вісім унікальних полів.

Зручність цього підходу дає можливість використовувати поля у різних комбінаціях. Наприклад, можна зробити сторінку, що починається з тексту,

після якого контент розділений на дві колонки. В цих колонках може бути будь-яке поле з відео чи зображенням і так далі.

ІКОНКА	МІТКА
	Відео
	Дві колонки (33/67)
	Дві колонки (50/50)
	Дві колонки (67/33)
	Зображення
	Зображення + Текст (Обтікає)
	Текст
	Текст (Обтікає) + Зображення

Рисунок 3.3.9 Поля створені у модулі Paragraphs.

В полі «Відео» можна вставити посилання на відео з YouTube.

Поле «Зображенням» дає можливість вставити зображення на сторінку.

Поля «Дві колонки» ділять сторінку на дві частини. В кожному з них можна помістити будь-яке поле із зазначених.

Поля «Текст (Обтікає) + Зображення» та «Зображення + Текст (Обтікає)» дають змогу обтікати тексту зображення зліва чи справа.

Поле «Текст» дає можливість вивести на сторінку звичайний контент з таблицями, зображеннями, посиланнями тощо.

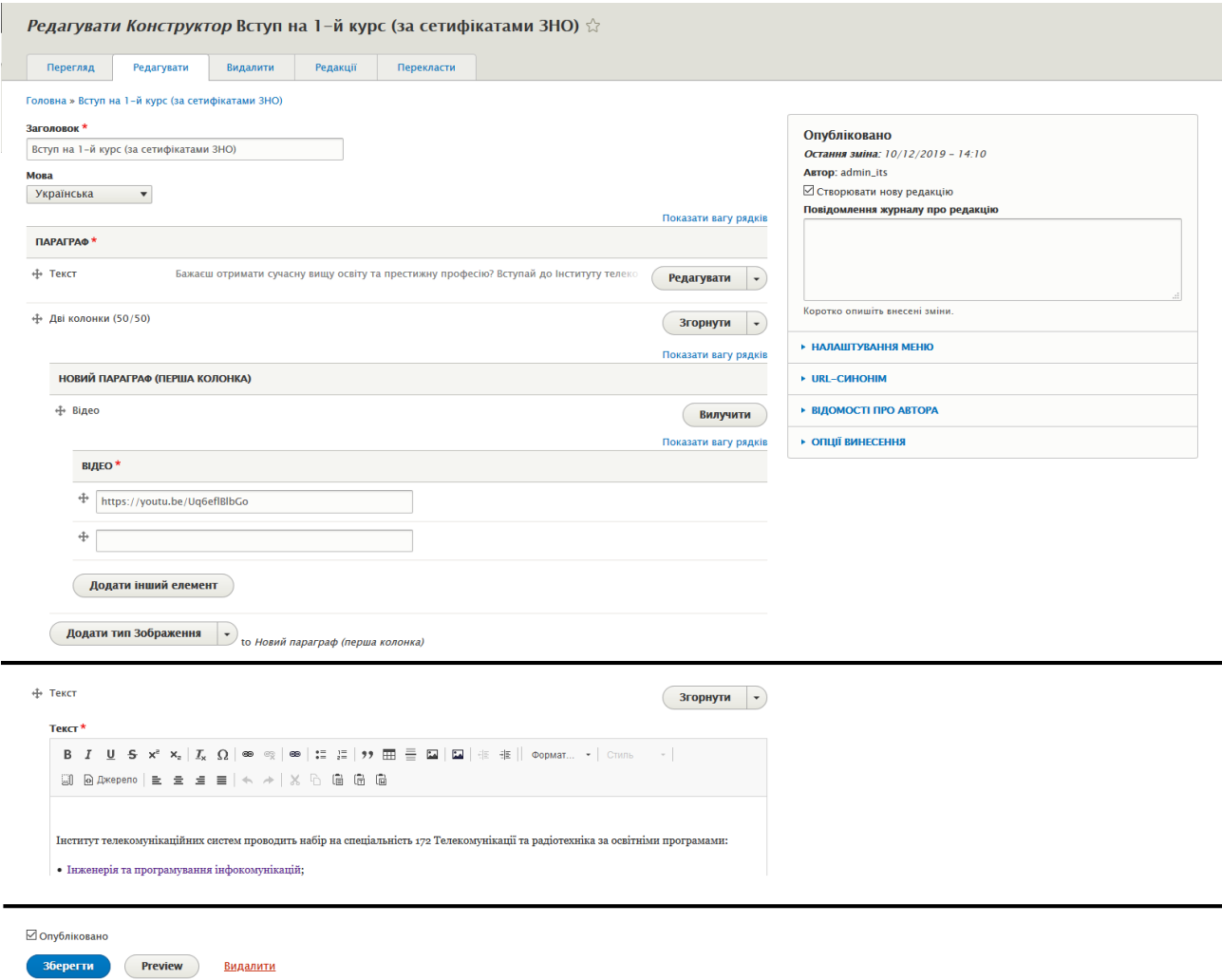


Рисунок 3.3.10 Поля Конструктора.

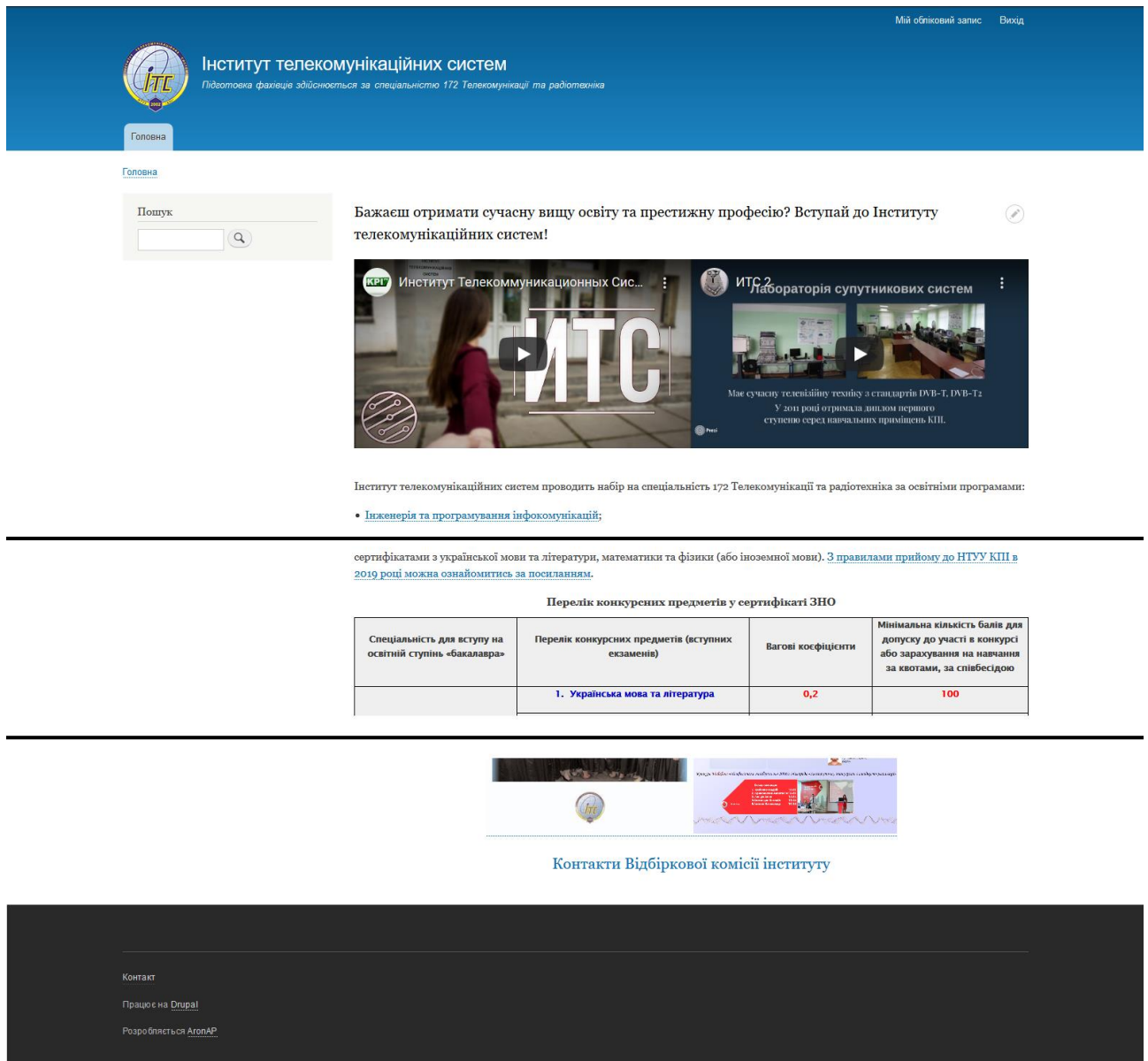


Рисунок 3.3.11 Вигляд Конструктора.

Висновок. Велика кількість типів матеріалу дає можливість зробити сайт більш гнучким та унікальним. Завдяки модулю Paragraphs можна надавати контент менеджерам можливість самим створювати потрібні шаблони, не звертаючись за поміччю до розробників.

3.4 Розробка теми

Розробка теми для сайту – це найбільш трудомістка та затратна частина всього процесу створення будь-якого веб-ресурсу. Вона потребує залучення не тільки розробника, а й дизайнера та контент менеджера.

Після обговорення всіх тонкостей та аналізу майбутнього матеріалу, було зіставлений базовий функціонал та уявлення про вигляд сайту. На основі цих даних можна приступити до розробки теми.

Новий функціонал та зміна структури сайту

Drupal дає можливість створювати не тільки сторінки, а й блоки для них. Таким чином, можна зробити власний блок і поставити його де завгодно.

Так, наприклад, був зроблений блок із контактами.

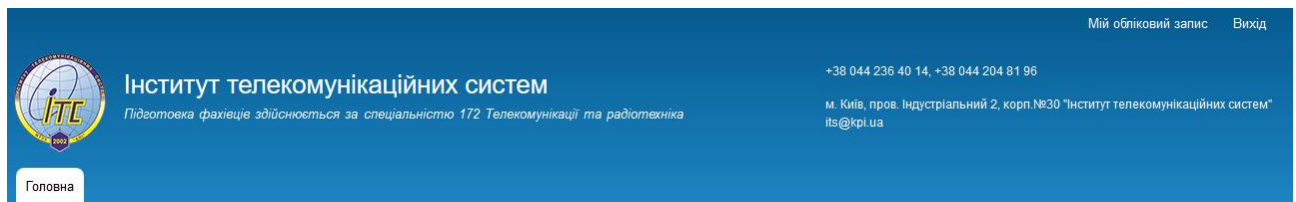


Рисунок 3.4.1 Шапка сайту із новим блоком.

Зробивши блок через плагін Superfish, можна зробити меню з випадаючими списками.

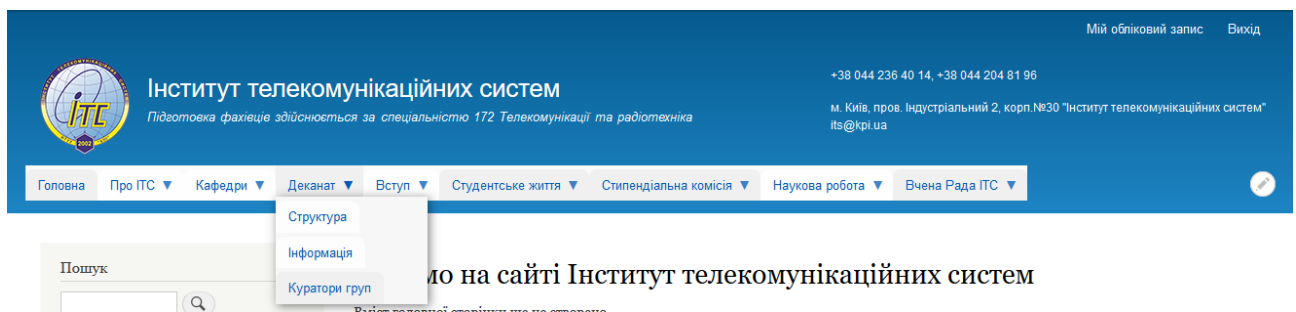


Рисунок 3.4.2 Шапка сайту із випадаючим меню.

Таким же чином можна створити блок для головної сторінки. Використовуючи модуль Paragraphs блок також можна робити гнучким, як і сторінку чи матеріал. Це дає змогу в швидкі строки отримати бажаний результат.

Зробивши гнучкий блок, ми можемо створювати цілі landing page, наприклад для абітурієнтів.

Для динамічного виводу контенту існує вбудований модуль Views. З його допомогою можна зробити вивід анонсів з пагінацією чи блок з останніми доданими публікаціями.

Також, Drupal дає можливість створювати власні області та шаблони. Для цього CMS використовує шаблонізатор PHP – Twig. За допомогою даного шаблонізатора можна створювати гнучкі шаблони. Наприклад, шаблон для

базової сторінки [Додаток А]. Тобто ми створюємо блоки не через веб інтерфейс, як описано вище, а у відповідних файлах. І вже після цього вставляємо їх у потрібне місце коду.

Це базові можливості для зміни структури та функціоналу сайту.

У Додатку Б показана головна сторінка сайту на даному етапі

Вигляд сайту

Структуру сайту ми змінили, але веб-сайт виглядає дуже погано. Для цього CMS Drupal, надає можливості для зміни відображення сайту – теми.

Що таке тема

Тема оформлення - це набір файлів, які визначають зовнішній вигляд вашого сайту. Саме вона відповідає, якими HTML та CSS буде «покритий ваш сайт». Так як візуальна складова сайту є важливою частиною першого враження користувача разом із UX та UI, то було б абсурдно нею нехтувати.

Базове ядро Друпал надає декілька тем для оформлення сайту. Одну з них ми і будемо використовувати та змінювати.

Варто зауважити, що змінювати базову, а також будь-яку іншу тему так як і ядро сайту не рекомендується! Саме тому ми будемо створювати підтему.

Що таке підтема

Підтема (sub-theme) - це тема, яка успадковує функціонал іншої теми. Під функціоналом розуміється - шаблони (файли *tpl.php*), css/js файли, налаштування, функції визначені в *template.php*.

Навіщо потрібні підтеми

Головний девіз Drupal – «**Never hack core!**», що означає – «Не чіпай ядро!». Саме для цього і потрібні підтеми - щоб можна було змінити якусь частину базової теми (що входить в ядро Друпал або завантажену з офіційного сайту) не чіпаючи її файли.

Наприклад в підтеми можна підключити свій css файл або змінити один з шаблонів, при цьому батьківську тему можна буде спокійно оновлювати, не боячись затерти зроблені зміни. Якщо ж робити зміни базових або сторонніх темах, то при оновленні всі зміни зникнуть.

Створення та активування теми

Ми будемо використовувати базову тему Bartik. Для того, щоб створити нашу підтему нам потрібно взяти головну тему за основу.

Для цього нам потрібно створити папку з підтемою та підключити її до головної теми, вказавши конфігурацію сайту [Додаток 3]. І після цього активувати її на сайті у розділі Вигляд.

Якщо називати файли таким чином, як в батьківській темі, то вони будуть доповнювати один одного. Це дає нам можливість не писати вже написаний за нас код.

Після закінчення розробки першого варіанту сайту він став виглядати набагато привабливіше для користувачів [Додаток Г].

3.5 Надання певних можливостей для супроводження сайту

Розділення контенту є основним принципом, на якому будуються всі сучасні системи управління контентом. Саме це дає нам дуже зручну ієрархію користувачів зі своїми правилами.

CMS Drupal дає можливість створювати Ролі та надавати їм певні права доступу. А вже ролі видавати окремим користувачам.

Користувач

Всі хто відвідують наш сайт, є Користувачами. Вони діляться на три групи:

- Користувачі, які не увійшли в систему, або анонімний користувач
- Користувачі, які увійшли в систему, або авторизований користувач
- Обліковий запис адміністратора, яка була автоматично створена при установці вашого сайту, або Користувач 1.

Права доступу

Можливість здійснювати дії на сайті (включаючи перегляд контенту, додавання контенту, і зміна конфігурації) регулюється Правами доступу. Кожне право доступу має ім'я (таке, як Перегляд опублікованого контенту) і охоплює одну дію або невелике підмножина дій. Користувач повинен отримати дозвіл для того, щоб зробити відповідну дію на сайті.

Права доступу визначаються модулями, які забезпечують дії.

Роль

Замість того, щоб призначати індивідуальні дозволи безпосередньо кожному користувачеві, права доступу згруповані в ролі. Ви можете визначити одну або кілька ролей на сайті, і потім надати права доступу кожної ролі.

Види користувачів на сайті інституту

На даний момент на ресурсі налічується п'ять видів користувачів, тобто п'ять ролей:

- Веб-адмін
- Адміністратор
- Контент менеджер
- Авторизований користувач
- Анонімний користувач

Можливості веб-адміна

Ця роль належить розробнику сайту, тобто мені. Я є першим користувачем і мені надані всі можливі права на сайт. Тільки я можу створювати всі ролі та надавати їх користувачам. Також тільки Веб-адміну доступні налаштування конфігурації сайту. Веб-адмін не управляє сайтом, а тільки підтримує його подальше існування.

Можливості адміністраторів

Всі права на управління сайтом, крім вищесказаних надаються адміністраторам. Сюди входять і звіти по статистикам сайту, і управління контентом, і зміни оформлення сайтом тощо. Адміністратор може створювати та надавати всі ролі, окрім Веб-адміна та Адміністратора.

Також даній ролі надане право видаляти авторизованих користувачів з сайту та попереджати їх про це листом на особисту пошту, вказану при реєстрації.

Можливості контент менеджерів

Роль Контент менеджера дуже проста. Він має право створювати та публікувати нові сторінки та статті, змінювати меню та редакції статей.

Кожен контент менеджер може видаляти та редагувати лише свій матеріал, а інший тільки переглядати.

Також він може переглядати та публікувати статті та сторінки будь-якого авторизованого користувача.

Можливості авторизованих користувачів

Кожен обліковий запис користувача на сайті автоматично отримує роль Авторизований користувач, і при бажанні, йому може бути призначена одна або декілька додаткових ролей. Коли ви призначите роль облікового запису користувача, користувач матиме всі права доступу при вході в систему.

Всі авторизовані користувачі можуть залишати коментарі під статтями, створювати нові сторінки та статті на сайті, але не можуть їх публікувати.

Можливості анонімних користувачів

Анонімний користувач – кожен відвідувач сайту, поки він не увійде у свій обліковий запис або не авторизується. У нього найменше прав з усіх наведених. Йому може бути обмежений доступ до деяких сторінок чи блоків сайту. Наприклад, він не може залишати коментарі під статтями.

Кожен Анонімний користувач може зареєструватися на сайті і отримати роль Авторизованого користувача.

3.6 Розгортання сайту в мережі

Коли весь сайт вже зроблений на локальній машині, настав час перенести його у мережу Інтернет. Для цього, в першу чергу, нам мають виділити місце на хостингу КПП Телеком для розміщення там нашого сайту. Після отримання хосту можна починати трансфер сайту.

Спочатку необхідно зробити свіжий дамп бази даних, що знаходиться у нас на комп'ютері. Це можна зробити через phpMyAdmin. Особливо важливо уточнити у хостинга можливий розмір завантаженої бази та заархівувати її у разі необхідності.

Далі, для швидкого завантаження через FTP-клієнта, краще заархівувати папку з нашим сайтом і завантажити на хостинг одним файлом. Якщо ж передавати сайт колекцією файлів, то це займе дуже багато часу. Коли завантаження завершиться потрібно просто розархівувати сайт у потрібну директорію.

Після переносу необхідно створити нову базу даних на виділеному хостингу та завантажити туди локальну базу. Якщо пароль та ім'я бази нові, то потрібно буде змінити їх в конфігурації сайту на хостингу.

Якщо все зроблено правильно то наш сайт завантажиться за виданим йому доменним ім'ям.

ДЖЕРЕЛА

1. Курс лекцій з дисципліни «Веб-проектування»/ Уклад.: О.П. Цурін. – К.: НТУУ «КПІ», 2011 р. ст.7-19, 34-47
2. Кількість сайтів в мережі // електрон. текст. дані URL: <https://psm7.com/special-projects/infographics/skolko-veb-sajtov-v-mire-infografika.html> (дата звернення 19.11.2019)
3. Классификация сайтов. Сайт-визитка. // електрон. текст. дані URL: http://wiki.mvtom.ru/index.php/Классификация_веб-сайтов (дата звернення 20.11.2019)
4. Статический сайт. Его преимущества и недостатки. // електрон. текст. дані URL: https://ru.wikipedia.org/wiki/Статический_сайт (дата звернення 21.11.2019)
5. Static site. Its advantages and disadvantages. // електрон. текст. дані URL: https://en.wikipedia.org/wiki/Static_web_page (дата звернення 21.11.2019)
6. Static vs Dynamic Websites: Pros and Cons. // електрон. текст. дані URL: <https://wsvincent.com/static-vs-dynamic-websites-pros-and-cons> (дата звернення 23.11.2019)
7. Программные средства для создания сайта // електрон. текст. дані URL: https://studbooks.net/2082456/informatika/programmnye_sredstva_sozdaniya_sayta (дата звернення 28.11.2019)
8. Design-based editor Adobe DreamWeaver // електрон. текст. дані URL: <https://www.adobe.com/ua/products/dreamweaver.html> (дата звернення 29.11.2019)
9. Code-based editor VS Code // електрон. текст. дані URL: https://ru.wikipedia.org/wiki/Visual_Studio_Code (дата звернення 05.12.2019)
10. Code-based editor Sublime Text // електрон. текст. дані URL: https://ru.wikipedia.org/wiki/Sublime_Text (дата звернення 05.12.2019)
11. Code-based editor Atom // електрон. текст. дані URL: [https://ru.wikipedia.org/wiki/Atom_\(текстовый_редактор\)](https://ru.wikipedia.org/wiki/Atom_(текстовый_редактор)) (дата звернення 05.12.2019)
12. Code-based editor PhpStorm // електрон. текст. дані URL: <https://ru.wikipedia.org/wiki/PhpStorm> (дата звернення 05.12.2019)
13. Code-based editor Brackets // електрон. текст. дані URL: <https://ru.wikipedia.org/wiki/Brackets> (дата звернення 05.12.2019)

14. Content management system // електрон. текст. дані URL:

https://en.wikipedia.org/wiki/Content_management_system (дата звернення 09.12.2019)

15. Information Architecture for the World Wide Web: Designing Large-Scale Web Sites, 3rd Edition p. 3-5, 392-426 (дата звернення 10.12.2019)

ДОДАТОК А

Програмний код шаблону сторінки

```
<div id="page-wrapper">
  <div id="page">
    <header id="header" class="header" role="banner" aria-
label="{{ 'Site header'|t }}">
      <menu id="main-menu" class="main-menu">
        <div class="section layout-container clearfix">
          {{ page.primary_menu }}
        </div>
      </menu>
      <div class="section layout-container clearfix">
        {{ page.secondary_menu }}
        {{ page.header }}
      </div>
    </header>
    {% if page.highlighted %}
      <div class="highlighted">
        <aside class="layout-
container section clearfix" role="complementary">
          {{ page.highlighted }}
        </aside>
      </div>
    {% endif %}
    {% if page.featured_top %}
      <div class="featured-top">
        <aside class="featured-top__inner section layout-
container clearfix" role="complementary">
          {{ page.featured_top }}
        </aside>
      </div>
    {% endif %}
    <div id="main-wrapper" class="layout-main-wrapper layout-
container clearfix">
      <div id="main" class="layout-main clearfix">
        {{ page.breadcrumb }}
        <main id="content" class="column main-
content" role="main">
          <section class="section">
            <a id="main-content" tabindex="-1"></a>
```

```

        {{ page.content }}
    </section>
</main>
{% if page.sidebar_first %}
    <div id="sidebar-first" class="column sidebar">
        <aside class="section" role="complementary">
            {{ page.sidebar_first }}
        </aside>
    </div>
{% endif %}
{% if page.sidebar_second %}
    <div id="sidebar-second" class="column sidebar">
        <aside class="section" role="complementary">
            {{ page.sidebar_second }}
        </aside>
    </div>
{% endif %}
</div>
</div>
{% if page.featured_bottom_first or page.featured_bottom_
second or page.featured_bottom_third %}
    <div class="featured-bottom">
        <aside class="layout-
container clearfix" role="complementary">
            {{ page.featured_bottom_first }}
            {{ page.featured_bottom_second }}
            {{ page.featured_bottom_third }}
        </aside>
    </div>
{% endif %}
<footer class="site-footer">
    <div class="layout-container">
        {% if page.footer_first or page.footer_second or page
.footer_third or page.footer_fourth %}
            <div class="site-footer__top clearfix">
                {{ page.footer_first }}
                {{ page.footer_second }}
                {{ page.footer_third }}
                {{ page.footer_fourth }}
            </div>
        {% endif %}
        {% if page.footer_fifth %}
            <div class="site-footer__bottom">
                {{ page.footer_fifth }}
            </div>
        {% endif %}
    </div>
</footer>
</div>

```

</div>

ДОДАТОК Б

Головна сторінка сайту без теми.

Пошук

Оберіть мову


 Українська

ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ



Інститут телекомунікаційних систем проводить підготовку фахівців за трьома рівнями вищої освіти:

- перший (бакалаврський);
- другий (магістерський);
- третій (освітньо-науковий).

Підготовка фахівців здійснюється за спеціальністю **172 Телекомунікації та радіотехніка**.

В інституті функціонують три кафедри, які готують випускників за трьома освітніми програмами.

Кафедра Телекомунікацій:

- [Інженерія та програмування інфокомунікацій](#)

Кафедра Телекомунікаційних систем:

- [Телекомунікаційні системи та мережі](#)

Кафедра Інформаційно-телекомунікаційних мереж:

- [Інформаційно-комунікаційні технології \(ІКТ\)](#)

Відео про ІТС:



Останні новини:

Це тестова стаття

Надіслав: admin_its, дата: чт, 06/04/2020 - 22:40



Теги

[Стаття](#) [тест](#)

Це зміст для тестової статті Це зміст для тестової статті Це зміст для тестової статті Це зміст для тестової статті Це зміст для тестової статті Це зміст для тестової статті Це зміст для тестової статті Це зміст для тестової статті Це зміст для тестової статті Це зміст для тестової статті Це зміст для тестової статті Це зміст для тестової статті Це зміст для тестової статті Це зміст для тестової статті Це зміст для тестової статті Це зміст для тестової статті Це зміст для тестової статті Це зміст для тестової статті Це зміст для тестової статті Це зміст для тестової статті

[Читати далі](#) [Коментувати](#)

[Контакт](#)

[Працює на Drupal](#)

[Розробляється Drupal](#)

ДОДАТОК В

Конфігурація підтеми сайту

```
name: ITS Theme
type: theme
```

```

base theme: bartik
description: 'This theme developed for ITS site'
package: Core
# version: VERSION
# core: 8.x
libraries:
  - bartik_custom/global-styling
  - bartik_custom/custom-styling
  - bartik_custom/custom-js
ckeditor_stylesheets:
  - css/base/elements.css
  - css/components/captions.css
  - css/components/table.css
  - css/components/text-formatted.css
regions:
  header: Header
  primary_menu: 'Primary menu'
  secondary_menu: 'Secondary menu'
  page_top: 'Page top'
  page_bottom: 'Page bottom'
  highlighted: Highlighted
  featured_top: 'Featured top'
  breadcrumb: Breadcrumb
  content: Content
  sidebar_first: 'Sidebar first'
  sidebar_second: 'Sidebar second'
  featured_bottom_first: 'Featured bottom first'
  featured_bottom_second: 'Featured bottom second'
  featured_bottom_third: 'Featured bottom third'
  footer_first: 'Footer first'
  footer_second: 'Footer second'
  footer_third: 'Footer third'
  footer_fourth: 'Footer fourth'
  footer_fifth: 'Footer fifth'

version: '1.0'
core: '8.x'
project: 'drupal'

```

ДОДАТОК Г
Головна сторінка сайту з новою темою

Головна
Про ІТС
Кафедри
Деканат
Вступ
Студентське життя
Стипендіальна комісія
Наукова робота
Вчена Рада ІТС

Мій обліковий запис
Вихід



Інститут телекомунікаційних систем

Підготовка фахівців здійснюється за спеціальністю 172 Телекомунікації та радіотехніка

+38 044 236 40 14, +38 044 204 81 96

м. Київ, пров. Індустріальний 2, корп. №30 "Інститут телекомунікаційних систем"
its@kpi.ua




Інститут телекомунікаційних систем проводить підготовку фахівців за трьома рівнями вищої освіти:

- перший (бакалаврський);
- другий (магістерський);
- третій (освітньо-науковий).

Підготовка фахівців здійснюється за спеціальністю **172 Телекомунікації та радіотехніка**.

В інституті функціонують три кафедри, які готують випускників за трьома освітніми програмами.

Кафедра Телекомунікацій:

- Інженерія та програмування інфокомунікацій

Кафедра Телекомунікаційних систем:

- Телекомунікаційні системи та мережі

Кафедра Інформаційно-телекомунікаційних мереж:

- Інформаційно-комунікаційні технології (ІКТ)

Пошук

Оберіть мову


Українська

Останні новини

ПРОТОКОЛ № 5
нд, 08/12/2019 - 19:48

ПРОТОКОЛ № 4
нд, 08/12/2019 - 19:43

ПРОТОКОЛ №3
нд, 08/12/2019 - 19:34

ПРОТОКОЛ № 2
нд, 08/12/2019 - 19:25

ПРОТОКОЛ № 1
нд, 08/12/2019 - 19:12

ПРОТОКОЛ № 10 ЗАСІДАННЯ...
нд, 08/12/2019 - 16:58

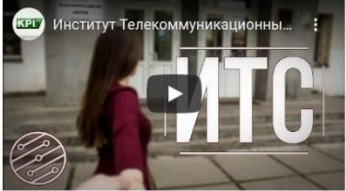
ПРОТОКОЛ № 9 ЗАСІДАННЯ...
нд, 08/12/2019 - 16:41

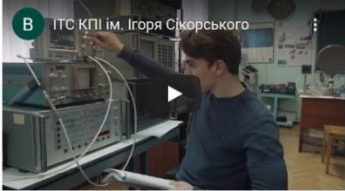
ПРОТОКОЛ № 8 ЗАСІДАННЯ...
нд, 08/12/2019 - 16:33

ЗАСІДАННЯ ВЧЕНОЇ РАДИ...
вт, 10/12/2019 - 14:18

ІТС КПІ ім. ІГОРЯ...
сб, 07/12/2019 - 18:51

Відео про ІТС:





Останні новини:

ПРОТОКОЛ № 5
Надіслав: [admin_its](#), дата: нд, 08/12/2019 - 19:44

ПОРЯДОК ДЕННИЙ: Доловіль молодого вченого про стан науки в області телекомунікацій за травень 2019 р.; Удосконалення взаємодії між адміністрацією ІТС та студентством ІТС; Підготовка до вступної компанії; Про хід взаємодії кафедр ІТС із компаніями-партнерами ІТС; Поточні справи.

[Читати далі](#)

1
2
3
4
[Наступна >](#)
[Остання >](#)

Контакт

Працює на Drupal

Розробляється Drupal