

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних тех-
нологій

Кваліфікаційна робота бакалавра

на тему: Реалізація веб-порталу для Наукового товариства
університету

Виконав студент групи К-21і
спеціальності 122 Комп'ютерні науки
Дідур Іван Олексійович

Керівник Доктор філософії, доцент
Бучинська Ірина Вікторівна

Консультант _____

Рецензент к.ф.-м.н., професор
Ковальчук Володимир
Володимирович

ЗМІСТ

Терміни, скорочення та умовні позначки	5
Вступ.....	7
1 Аналіз основних етапів розробки інформаційної системи	9
1.1 Опис етапу розробки «Дизайн ІС»	10
1.2 Опис етапу розробки «Верстка ІС»	11
2 Характеристика архітектури та структури ІС у вигляді веб-порталу	16
2.1 Аналіз клієнтських мов програмування.....	21
2.2 Аналіз серверних мов програмування.....	26
3 Опис етапів розробки веб-порталу для наукового товариства університету	30
Висновки	48
Перелік джерел посилання	50

ТЕРМІНИ, СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

Апаратне забезпечення – це фізичний компонент технології.

Веб-дизайн (англ. web design) – галузь веброзробки, що охоплює цілий ряд напрямів і дисциплін із створення та супроводу сайтів або вебзастосунків, таких як графічний вебдизайн, проєктування інтерфейсів, авторинг (у тому числі стандартизований код і власницьке програмне забезпечення), використання та оптимізація для пошукових систем.

Веб-додаток – це програмне забезпечення або програма, яку можна відкрити за допомогою будь-якого браузера.

Верстання вебсторінки – це процес створення вебсторінки із попередньо створеного макету дизайну сайту, заздалегідь намальованого за допомогою графічних редакторів.

Домен – це ім'я інформаційної системи, яке вказується в адресному рядку.

Заголовки HTTP (англ. HTTP Headers) – це рядки в HTTP-повідомленні, що містять розділене двокрапкою пару ім'я-значення.

Інформаційна система (IS) – це поєднання програмного забезпечення, апаратного забезпечення та телекомунікаційних мереж для збору корисних даних, особливо в організації.

Семантичне ядро – це база пошукових слів, фраз чи морфологічних форм, що характеризує вид діяльності, послуги чи товари, що пропонуються інформаційною системою.

Системне програмне забезпечення – це операційна система, яка керує обладнанням, програмними файлами та іншими ресурсами, пропонуючи користувачеві керувати ПК за допомогою графічного інтерфейсу користувача.

CGI (від англ. Common Gateway Interface – «загальний інтерфейс шлюзу») – стандарт інтерфейсу, який використовується для організації взаємодії програми вебсервера із зовнішньою програмою. Програму, яка працює за таким інтерфейсом спільно з вебсервером, прийнято називати шлюзом.

HTTP – протокол передачі даних, що використовується в комп'ютерних мережах. Назва скорочена від HyperText Transfer Protocol, протокол передачі гіпертекстових документів.

Internet Server Application Programming Interface (ISAPI) – це API для Internet Information Services, колекції мережних служб Microsoft Windows.

URL – це адреса, який має будь-який мережевий ресурс і який можна використовувати для роботи в Інтернеті

User experience (UX / Дизайн користувальницького досвіду) – це процес, який команди проектувальників використовують для створення продуктів, які забезпечують значущі та відповідні враження для користувачів.

WEB – система доступу до пов'язаних між собою документів на різних комп'ютерах, підключених до Інтернету.

IS – інформаційна система.

ШІ – штучний інтелект.

CGI – Common Gateway Interface

HTTP – HyperText Transfer Protocol.

ISAPI – Internet Server Application Programming Interface.

URL – Uniform Resource Locator.

UX – User experience.

ВСТУП

Інформаційні системи з роками набули величезної популярності. Майбутнє інформаційних систем та їх важливість залежить від автоматизації та впровадження технології ШІ.

Інформаційні технології можуть використовуватися в спеціалізованих і узагальнених цілях. Узагальнена інформаційна система надає загальні послуги, такі як система керування базою даних, де програмне забезпечення допомагає організувати загальну форму даних. Наприклад, різні набори даних отримують за допомогою формули, яка дає змогу зрозуміти певні тенденції за конкретний проміжок часу. Навпаки, спеціалізована інформаційна система будується для виконання певної функції для бізнесу. Наприклад, експертна система, яка вирішує складні задачі. Ці проблеми зосереджені на певній галузі дослідження, основна мета – запропонувати швидші та точніші послуги.

Мета кваліфікаційної роботи бакалавра полягає в покращенні доступу до інформації, створенні сприятливого середовища для спілкування та обміну думками, підтримці співпраці та взаємодії між науковцями, спрощенні організаційних процесів та пропаганді науки.

Для надання цікавого та цінного контенту, який принесе користь користувачам, розробляються та впроваджуються інформаційні портали. Це може бути інформаційна система з практичними порадами чи рекомендаціями, блог. Кожен вид інформаційної системи, так чи інакше, націлений на залучення користувачів, тільки відбуваються це за різною специфікою. Спілкування з клієнтами / користувачами за допомогою корпоративного сайту, сайту-візитки та сайту-каталогу виконується через зворотний зв'язок, від якості якого залежить чи згоден користувач співпрацювати далі. Інші види інформаційних систем допоможуть купити або замовити товар тут і зараз, натиснувши одну кнопку. Інформаційний портал відноситься до некомерційної діяльності, але з його допомогою можна почати отримувати прибуток, розміщуючи рекламні посилання на інформаційній системі, обмінюючись банера-

ми, роблячи e-mail розсилку та ін. Домен – це ім'я інформаційної системи, яке вказується в адресному рядку. Воно має бути простим, що запам'ятовується, легко писатися на трансліті, щоб людина при ручному введенні змогла швидко впоратися із завданням.

Одним з найважливіших етапів створення інформаційної системи є збір семантичного ядра. Семантичне ядро – це база пошукових слів, фраз чи морфологічних форм, що характеризує вид діяльності, послуги чи товари, що пропонуються інформаційною системою. Семантика підбирається з метою подальшого просування інформаційної системи в Інтернеті та його розвитку. Семантичне ядро може збиратися для загальної інформаційної системи з урахуванням слів, які характеризують діяльність загалом, і під кожен сторінку, що найчастіше рекомендується SEO-фахівцями, оскільки це ефективніше. Ключові фрази (ключи, ключові входження) повинні відповідати тематиці інформаційної системи, не мати граматичних помилок і підбиратися за частотою. Щоб зробити інформаційну систему зручним для відвідувача, важливо проаналізувати всі запити, які мають попит.

Інформаційна система повинна бути незабутнім, зручним у використанні та грамотно структурованим. Це дуже важливо, тому що від цього залежить не лише подальша розробка дизайну інформаційної системи та його наповнення контентом, а й комфорт відвідувачів. Потрапляючи на сайт, кожен відвідувач повинен розуміти як знайти на інформаційній системі товар/послугу, що цікавить, прочитати його характеристики. Усі блоки мають бути збудовані логічно[1].

Дана кваліфікаційна робота бакалавра, складається з 50 сторінок, 16 рисунків та 6 джерел посилання.

1 АНАЛІЗ ОСНОВНИХ ЕТАПІВ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Значення терміну інформаційна система, найчастіше асоціюється з програмним забезпеченням, яке засноване на зберіганні інформації або щось подібне. Однак інформаційна система набагато більше, ніж це.

Інформаційна система (ІС) – це поєднання програмного забезпечення, апаратного забезпечення та телекомунікаційних мереж для збору корисних даних, особливо в організації. Багато підприємств використовують інформаційні технології для завершення та керування своїми операціями, взаємодії зі своїми споживачами та випередження конкурентів. Деякі компанії сьогодні повністю побудовані на інформаційних технологіях, як-от eBay, Amazon, Alibaba та Google.

До типових компонентів інформаційної системи, можна віднести наступні чотири компоненти – апаратне забезпечення, програмне забезпечення, дані та телекомунікації.

Апаратне забезпечення – це фізичний компонент технології. Сюди входять комп'ютери, жорсткі диски, клавіатури, iPad тощо. Вартість апаратного забезпечення швидко знизилася, а його швидкість і ємність пам'яті значно зросли. Проте сьогодні величезне занепокоєння викликає вплив використання обладнання на навколишнє середовище. Сьогодні послуги зберігання пропонуються з хмари, до якої можна отримати доступ із телекомунікаційних мереж.

Програмне забезпечення може бути двох типів: системне та прикладне. Системне програмне забезпечення – це операційна система, яка керує обладнанням, програмними файлами та іншими ресурсами, пропонуючи користувачеві керувати ПК за допомогою графічного інтерфейсу користувача. Прикладне програмне забезпечення призначене для керування певними завданнями користувачів. Коротше кажучи, системне програмне забезпечення робить апаратне забезпечення придатним для використання, тоді як прикладне про-

грамне забезпечення виконує конкретні завдання. Прикладом системного програмного забезпечення є Microsoft Windows, а прикладного програмного забезпечення – Microsoft Excel. Великі компанії можуть використовувати ліцензовані програми, які розроблені та керовані компаніями-розробниками програмного забезпечення для задоволення своїх конкретних потреб. Програмне забезпечення може бути пропрієтарним і відкритим, доступним у мережі для безкоштовного використання.

Дані являють собою набір фактів і марні самі по собі, але коли їх зібрати й упорядкувати разом, вони можуть бути дуже потужними для бізнес-операцій. Компанії збирають усі дані та використовують їх для прийняття рішень, які можна проаналізувати на предмет ефективності бізнес-операцій[2].

Телекомунікації використовуються для підключення до комп'ютерної системи або інших пристроїв для поширення інформації. Мережу можна встановити за допомогою дротового або бездротового режимів. Дротові технології включають оптоволокно та коаксіальний кабель, тоді як бездротові технології включають радіохвилі та мікрохвилі.

1.1 Опис етапу розробки «Дизайн ІС»

Дизайн ІС – це його обличчя та перше, що впадає у вічі при відвідуванні ресурсу. Логічно, що він має бути виконаний професійно. Дизайн ресурсу слід продумати заздалегідь, приділивши цьому часу. Він повинен надихати відвідувача якомога довше провести час на сайті, очам має бути інтуїтивно зрозуміло та зручно читати інформацію.

Задля розробки гарного, сучасного та ефективного дизайну необхідно у роботі використовувати наступні 4 принципи.

По-перше підбирати нейтральне тло і правильно поєднуйте кольори. Кольори мають сильний психологічний фактор, тому слід підбирати оформлення в 90% нейтральних кольорів та 10% яскравих та помітних. Червоний та

інші «отруйні» кольори можуть бути акцентом на будь-яку пропозицію, не варто їх брати за основу, тому що їм властиво викликати неоднозначні, частіше негативні, емоції у людини. Фон має бути нейтральним і не надто привертати до себе увагу. Використовувати необхідно правильне поєднання кольорів для ІС. Це може бути градація відтінків, що належать до однієї групи, або кольори, які добре поєднуються один з одним. Поєднання червоного та зеленого кольору у фоні відштовхнуть користувачів та залишать неприємний осад після відвідин ІС.

По-друге обирати шрифти оптимального кольору та розміру. Занадто великий шрифт виглядає безглуздо, надто дрібний нерозбірливо. Колір шрифту має максимально виділятися на фоні. Оптимальний колір шрифту – чорний. Можна вибрати білий, якщо фон сторінки планується виконати у темних тонах. Шрифт має бути розбірливим та читабельним. Скористатися класичними типами шрифту, не слід вигадувати, якщо варто розробити серйозне ставлення до ІС.

По-третє ґрунтуватися на факторах UX. UX (user experience) – термін, що характеризує рівень задоволеності відвідувача від відвідування сайту. Якщо людині зручно та цікаво користуватися та перебувати на ІС, то вона повернеться, якщо ні, користувач не повернеться. Після розробки з врахування UX факторів, слід об'єктивно оцінити зручність перебування на сайті з обраним дизайном.

По-четверте, відповідати обраному фірмовому стилю. Кожна організація, що поважає себе, має фірмовий стиль, який надійно осідає у людини в підсвідомості. Слід продумати фірмовий стиль і дотримуйтесь його від початку до кінця.

1.2 Опис етапу розробки «Верстка ІС»

Верстка – це створення макета у графічних редакторах та перетворення на HTML та CSS-код для перегляду сайту у браузерях. Без верстки ІС зали-

шиться нефункціональною «картинкою». Ключовою проблемою неправильної верстки може стати неякісне або незручне відображення ІС у різних браузерах. Згорнувши інформаційну систему для Opera, вона може по-іншому відображатися в Firefox і повністю спотворитися в Internet Explorer. Кожен браузер розробляється з урахуванням власних технологій та правил, тому дуже важливо кросбраузерну ІС, щоб вона правильно відображалася у будь-якому браузері.

Не варто забувати про мобільні та планшетні пристрої. Кількість користувачів Інтернету за допомогою електронних гаджетів щороку збільшується, тому важливо зробити ІС адаптивною, щоб вона коректно відображалася на будь-яких типах мобільних та планшетних пристроїв з будь-яким розміром діагоналі. Так вдасться отримати частину мобільних користувачів як клієнтів, що позитивно вплине на відвідування та зацікавленість даною ІС. Після верстки потрібно обов'язково перевірити роботу сайту, його чутливість, роботу кнопок, форми зворотного зв'язку тощо.

Часто верстці не приділяють належної уваги, тому навіть ексклюзивний сайт зі складним функціоналом стає практично марним.

Існує декілька варіантів розробки верстки ІС, кожен з цих варіантів має свої переваги та недоліки.

По-перше, якщо звернутися до фахівців, перевагами буде:

- фахівці, приступлять до справи без зволікань;
- досить чітко сформульовані завдання;
- ІС розроблені професіоналами зазвичай працюють ефективно, з їх допомогою вдається розширити базу клієнтів, оскільки за справу береться людина або колектив з досвідом;
- на фінальному етапі сайт тестують, виявляють та виправляють помилки у розмітці та контенті, якщо такі є.

До недоліків у свою чергу слід віднести:

- необхідний час на пошук відповідного виконавця;

– у разі необхідності, потрібно буде підготувати докладне технічне завдання або замовити його за окрему плату у фахівців.

По-друге, розробити ІС самостійно, у цьому випадку теж є як переваги так і недоліки.

До переваг слід віднести:

– мінімальні витрати включають лише оплату домену та хостингу або конструктора сайтів;

– необхідність аналізувати та обробляти всю інформацію самостійно, але нові знання – корисний вантаж, який надалі стане в нагоді.

До недоліків слід віднести:

– витрати великого проміжку часу на вивчення HTML, адміністративної панелі та нюансів роботи CMS;

– у разі, якщо ІС яка розробляється це перший проєкт, велика ймовірність, що не вдасться втілити у життя всі ідеї;

– слабкий рівень реалізації проєкту гальмуватиме просування сайту та «відлякуватиме» деяких вибагливих відвідувачів.

Займатися розробкою ІС самостійно є сенс, якщо є бажання вникати у всі нюанси, час на навчання і терміни не горять, величезне бажання заощадити.

У разі, якщо все ж таки було вирішено залучити до розробки ІС фахівців, постає питання щодо вибору або ж фрілансера, або ж веб-студії. Якщо було вирішено обрати для розробки веб-студію, то слід вказати наступні переваги, а саме:

– можливість комплексного замовлення послуг: від розробки, вибору домену та хостингу до просування;

– веб-студія виділить менеджера з управління проєктом (з яким можна вирішувати всі питання, такі фахівці вміють спілкуватися з клієнтами зрозумілою для них мовою);

- командна робота хороша тим, що у проєктах беруть участь вузькі спеціалісти (дизайнери, веб-програмісти, копірайтери), кожен з яких профі у своїй справі;

- веб-студії хороші можливості закінчити розробку сайту в обумовлений термін (у разі, якщо фахівець захворіє, то йому знайдуть заміну);

- веб-студія здає проєкт повністю після попереднього тестування;

До недоліків, можна віднести фінансову складову, а саме висока вартість послуг.

На сьогоднішній день, є можливість замовити виконання проєкту у фрілансерів. Перевагами такої співпраці є [3]:

- послуги окремих фахівців у сукупності коштують дешевше, ніж розробка ІС веб-студії;

- є можливість спілкуватися з кожним спеціалістом та контролювати їх безпосередньо (це плюс для людей, які хоч трохи знаються на сайтобудуванні і мають достатньо часу).

До недоліків, в свою чергу варто віднести:

- доведеться стати менеджером проєкту, здійснювати контроль на кожному етапі;

- іноді фрілансери хворіють, не справляються з іншими несподіваними перешкодами, що ускладнюють виконання роботи вчасно.

1.3 Опис етапу розробки «Наповнення ІС контентом»

Відвідувач заходить на ІС насамперед за контентом. Його цікавить конкретний інформація, а зручний функціонал, структура та гарний дизайн лише доповнюють його. Тому контент має бути не лише унікальним для пошукових систем, а й цікавим для користувача ІС. Важливі моменти, які потрібно враховувати під час створення ІС. Також незначну роль контент грає у SEO-просуванні. Вирішивши розкрити ІС цим способом, контент має бути

оптимізований, а статті написані за структурою, з використанням семантичного ядра.

Контент – це будь-яка інформація, яка представляє хоч якусь цінність для відвідувачів. Говорячи про ІС, найчастіше мається на увазі текстовий контент: статті, новини, огляди тощо. Є ще медіаконтент. До нього можна віднести аудіо, відео, картинки. Також буває ігровий контент. До цієї категорії належать всілякі ігри.

Тобто заповнення ІС контентом – це заповнення його інформацією. Абсолютно будь-який: від опису сторінки до повноцінної статті у якомусь розділі. Будь-яку інформацію, яка публікується на ІС, можна називати контентом. Але тут є певні винятки. Наприклад, товари в інтернет-магазині це не контент, а саме товари.

Тут обов'язково потрібно бачити різницю, адже контент – це саме такий пізнавально-розважальний вид інформації, який допомагає користувачам дізнатися про щось нове. Елементи контенту є в тих же описах товарів, але в digital-середовищі при слові контент найчастіше маються на увазі статті, новини, інформаційні тексти, картинки і т.д.

2 ХАРАКТЕРИСТИКА АРХІТЕКТУРИ ТА СТРУКТУРИ ІС У ВИГЛЯДІ ВЕБ-ПОРТАЛУ

Зазвичай веб-програми розробляються як застосунки в архітектурі «клієнт-сервер», проте серверна сторона може мати різні між собою архітектурні рішення [4].

Схематично архітектура «клієнт-сервер» представлена на рис. 1.

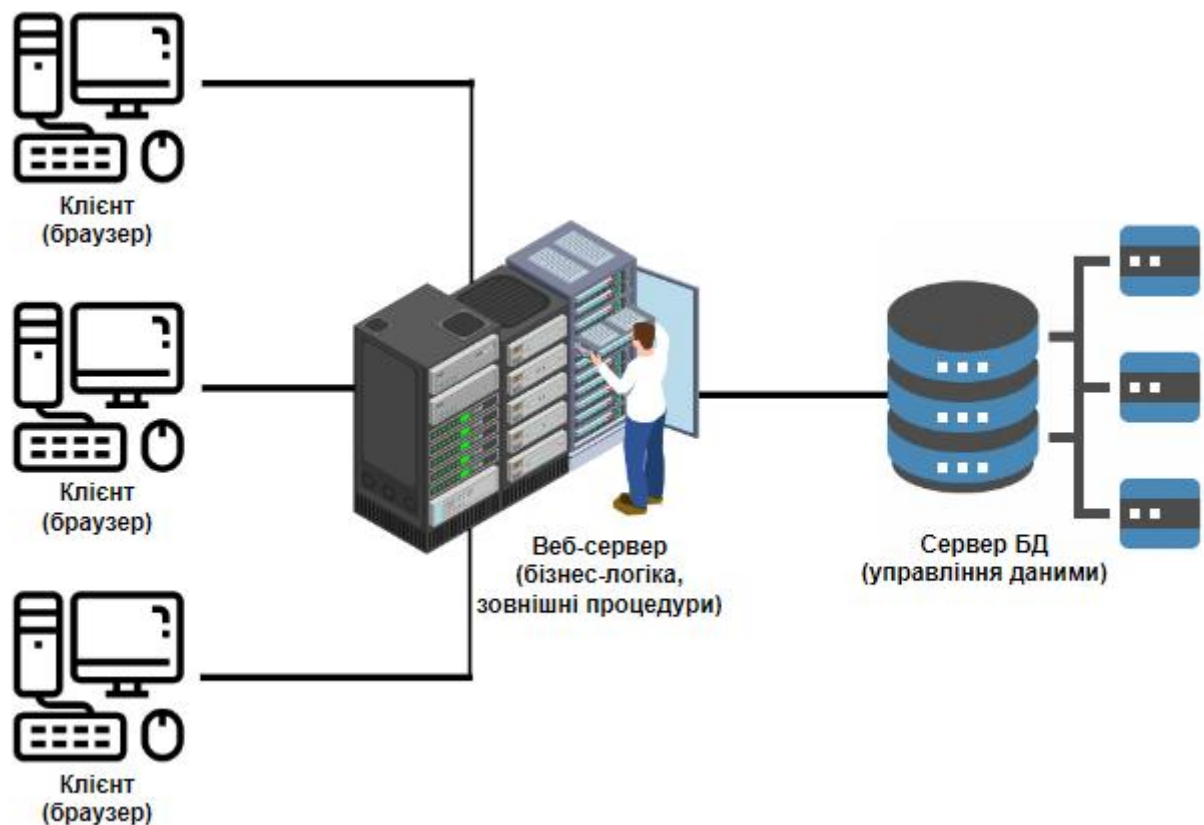


Рисунок 1 – Архітектура веб-застосунків

Спочатку «World Wide Web» представлялася її розробникам як «система для обміну інформацією, в якій люди та комп'ютери будуть спілкуватися один з одним». Через це перші веб-застосунки ґрунтувалися на простій файл-серверній архітектурі, тобто відправляли на запит користувача статичну html-сторінку. Таким чином, Web зароджувалося як звичайна документна система.

Другий етап розвитку Web пов'язані з появою нових застосунків, які розроблялися таких інтерфейсах, як «CGI», а надалі – на «ISAPI». Common Gateway Interface – є стандартним інтерфейсом, що працює з серверами і дозволяє виконувати різні серверні програми, що викликаються через URL. Вхідною інформацією даних застосунків служить вміст HTTP-заголовків. CGI-програми генерують html-розмітку, яка повертається браузеру. Але головною проблемою CGI-програм стало те, що при кожному запиті від браузера сервер знову виконує CGI-програму, записуючи її в окремий адресний простір сервера.

Поява Internet Server API (ISAPI) дозволило як вирішити проблеми продуктивності, пов'язані з використанням CGI-застосунків, а й надати у розпорядження розробників розширений програмний інтерфейс. ISAPI DLL могли асоціюватися з розширеннями імен файлів через спеціальну мета-базу. Саме ці два механізми (CGI і ISAPI) і послужили основою створення першого типу веб-застосунків, у яких, залежно від будь-яких клієнтських дій, виконувався серверний код. Це дало можливість динамічно генерувати вміст веб-сторінок, завдяки чому веб-контент перестав бути чисто статичним.

Інтерфейс ISAPI є особливістю Microsoft Internet Information Server. ISAPI-застосунки це динамічні бібліотеки, що завантажуються (DLL), які виконуються в адресному просторі Веб-сервера. Через деякий час на Веб-серверах інших розробників також було реалізовано можливість виконання програм за допомогою динамічних бібліотек. Для веб-серверів Netscape цей програмний інтерфейс називався NSAPI (Netscape Server API). Найпопулярніший веб-сервер Apache також має можливість виконувати веб-застосунки за допомогою механізму Apache DSO (Dynamic Shared Objects).

Оскільки при використанні як CGI-, так і ISAPI-застосунків розробники в основному вирішували одні і ті ж завдання, то наступним природним кроком стала поява нового високорівневого інтерфейсу, завдяки якому спростилася генерація HTML-коду, з'явилася можливість звернення до компонентів і

баз даних. Таким інтерфейсом стала об'єктна модель Active Server Pages (ASP), побудована з урахуванням ISAPI-фільтра.

Основною ідеєю ASP з погляду створення інтерфейсу програми у тому, що у веб-сторінці присутні фрагменти коду, інтерпретовані веб-сервером, а користувач отримує вже кінцевий результат виконання цих фрагментів коду.

Після ASP створили й інші технології, реалізують ідею розміщення всередині веб-сторінки коду, виконуваного веб-сервером. Найбільш відомою з них сьогодні є технологія JSP (Java Server Pages). Основна ідея даної технології полягає в одноразовій компіляції Java-коду (т.зв. сервлета) при першому зверненні до нього, виконання методів цього сервлету та поміщені результатів виконання цих методів у набір даних, що надсилаються до браузера.

Найновіша версія технології Active Server Pages – це ASP.NET, що є ключовою в архітектурі Microsoft .NET Framework. За допомогою ASP.NET можна створювати як веб-застосунки, так і веб-сервіси, які дозволяють реалізувати не тільки динамічну генерацію HTML-сторінок, а й дають можливість інтеграції з серверними компонентами і можуть використовуватися для вирішення широкого кола бізнес-завдань, що постають перед розробниками сучасних веб-застосунків.

В наш час клієнтом веб-сервера може бути не лише персональний комп'ютер, оснащений звичайним веб-браузером. Одночасно з широким розповсюдженням мобільних пристроїв постало завдання надання веб-серверами даних, які можуть бути інтерпретовані цими пристроями. Оскільки характеристики мобільних пристроїв скромніші у порівнянні з характеристиками ПК (обмежений розмір екрану, менший об'єм пам'яті, для старих моделей – можливість роботи тільки в текстовому режимі), то для них існують інші протоколи передачі даних (WAP – Wireless Access Protocol) та відповідні мови розмітки (WML – Wireless Markup Language, XHTML – Compact HTML тощо). При цьому передача даних на мобільний пристрій здійснюється або у відповідному форматі (шляхом створення спеціальних версій сайтів), або, що зручніше, відбувається впізнання типу пристрою в момент його звернення до

сервера та перетворення вихідного документа (наприклад, у форматі XML) у формат, потрібний мобільний пристрій (наприклад, за допомогою XSLT-перетворення).

Іншим способом підтримки різних типів клієнтів є створення «розумних» серверних компонентів, здатних генерувати різний код залежно від типу клієнта (як це реалізовано в Microsoft ASP.NET).

Іншим напрямком розвитку клієнтських елементів веб-застосунків стало розміщення деякої частини логіки застосунка [5] (такий як перевірка коректності даних, що вводяться) в самому веб-браузері (рис.2). Зокрема, сучасні веб-браузери здатні інтерпретувати скриптові мови (VBScript, JavaScript), код якими, як і ASP-код, впроваджується у веб-сторінку, але інтерпретується не веб-сервером, а браузером і відповідно виконується на клієнтському пристрої. Крім того, сучасні браузери здатні відображати та виконувати Java-аплети – спеціальні Java-застосунки, які користувач отримує у складі Веб-сторінки, а деякі з браузерів можуть також служити контейнерами для елементів керування ActiveX, які виконуються в адресному просторі браузера спеціальних COM-серверів, також одержуваних у складі веб-сторінки. За допомогою Java-аплетів та елементів керування ActiveX можна реалізувати практично будь-яку функціональність.

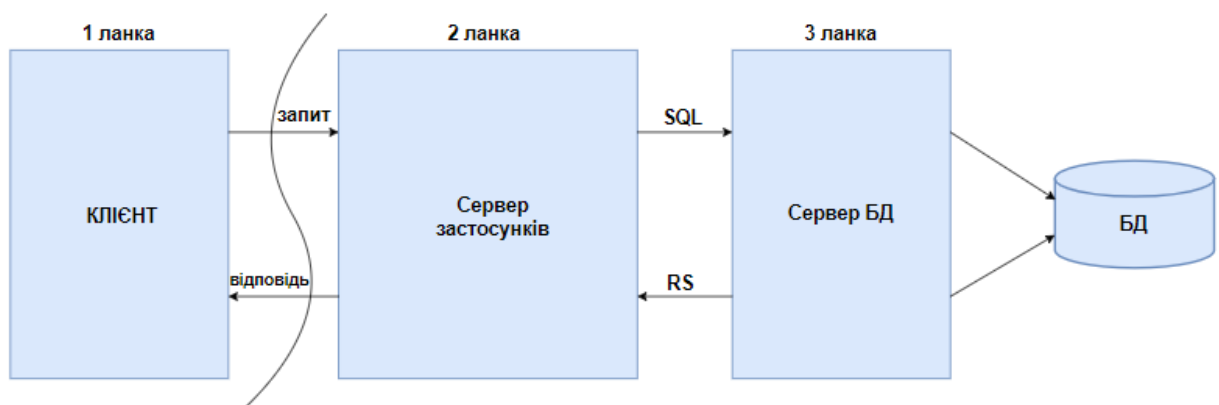


Рисунок 2 – Триланкова архітектура веб-застосунків

Варто відзначити, що зі зростанням обсягу використовуваних даних та числа відвідувачів веб-сайтів зростають і вимоги до надійності, продуктивності та масштабованості веб-застосунків. Наступним етапом еволюції подібних застосунків стало відділення бізнес-логіки, реалізованої у веб-застосунку, а нерідко і сервісів обробки даних та реалізації транзакцій від його інтерфейсу. У цьому випадку в самому веб-застосунку зазвичай залишається так звана презентаційна частина, а бізнес-логіка, обробка даних та реалізація транзакцій переносяться на сервер застосунків у вигляді бізнес-об'єктів. Залежно від типу сервера застосунків такі бізнес-об'єкти можуть бути самостійно виконуються COM-серверами, CORBA-серверами, а також об'єктами COM+, що виконуються за допомогою служб компонентів MS Windows, або об'єктами EJB (Enterprise Java Beans), що виконуються сервером застосунків, що підтримують специфікацію J2EE (Java 2 Enterprise Edition). Як механізм доступу до даних подібні об'єкти зазвичай використовують OLE DB, ODBC, JDBC (залежно від архітектурної реалізації самого бізнес-об'єкта). Слід сказати, що будучи складовою такого рішення, веб-сервер повинен вміти як виконувати програми та взаємодіяти з сервером застосунків, але й використовувати сервіси інтеграції, сервіси управління застосунками і даними, і навіть сервіси для розробників.

Наступним кроком еволюції веб-застосунків, окрім доступу до корпоративних даних та даних партнерів, стало отримання доступу до корпоративних програм. Для вирішення цього завдання інтеграції веб-застосунків із внутрішніми інформаційними системами підприємств та з додатками, що забезпечують взаємодію з клієнтами та партнерами, використовуються спеціальні рішення, які називаються корпоративними порталами.

Узагальнюючи вищесказане, можна виділити основні особливості веб-архітектури:

- відсутність необхідності використовувати додаткове програмне забезпечення на стороні клієнта, що дозволяє реалізувати крос-платформну клієнтську частину;

- можливість підключення практично необмеженої кількості клієнтів;
- обсяг збережених та оброблюваних даних обмежується лише пропускною спроможністю каналів зв'язку та потужністю веб-серверів;
- завдяки єдиному місцю зберігання даних та наявності системи управління базами даних забезпечуються простота адміністрування;
- доступність залежить від працездатності сервера та каналів передачі даних;
- нижча швидкість відгуку веб-сервера і каналів передачі (порівняно з локальною мережею).

Таким чином, веб-програмування (веб-розробка) – найбільший розділ, що бурхливо розвивається в даний час, програмування, орієнтований на створення динамічних Інтернет-застосунків.

Мови, які використовуються для веб-програмування, можна розділити на дві групи: клієнтські та серверні.

2.1 Аналіз клієнтських мов програмування

Веб-програми, реалізовані з використанням даної технології, виконуються браузером на стороні клієнта (користувача).

Одна з переваг використання клієнтської мови полягає в тому, що процес обробки скриптів проходить до відправлення даних на сервер, оскільки перевіряється коректність введених даних, а при виникненні необхідності виводяться повідомлення про помилки. Як наслідок, впливає обмеження, що за допомогою клієнтської мови не можна записати дані на сервер.

Більше того, обробка та виконання клієнтського скрипту повністю залежить від інтернет-браузера, що використовується користувачем та його налаштувань. Браузери застарілих версій не зможуть підтримувати більшість тих чи інших можливостей, закладених у нових версіях мов, з пощою яких розробник писав скрипт. Також користувач через браузер може повністю відключити підтримку виконання скриптів.

HyperText Markup Language («мова розмітки гіпертексту») – стандартна мова розмітки документів у Всесвітньому павутинні. Більшість веб-сторінок створюються за допомогою HTML. Хоча HTML інтерпретується браузером і відображається у вигляді документа, у зручній для людини формі, він не є мовою програмування у стандартному розумінні цього слова. HTML (як і його продовження XHTML/DHTML, XML) є додатком SGML (стандартної узагальненої мови розмітки) та відповідає міжнародному стандарту ISO 8879.

HTML створювався як мова для обміну науковою та технічною документацією, придатний для використання людьми, які не є фахівцями в галузі верстки. HTML успішно справлявся із проблемою складності SGML шляхом визначення невеликого набору структурних та семантичних елементів – дескрипторів. Дескриптори також називають «тегами». За допомогою HTML можна просто створити відносно простий, але красиво оформлений документ. Крім спрощення структури документа, HTML внесена підтримка гіпертексту. Пізніше було додано мультимедійні можливості. Спочатку мова HTML була задумана і створена як засіб структурування та форматування документів без їх прив'язки до засобів відтворення (відображення).

Текстові документи, що містять HTML-код, обробляються спеціальними додатками, які відображають документ у вже форматованому вигляді. Такі програми, які називаються «браузерами» або «Інтернет-браузерами», зазвичай надають користувачеві зручний інтерфейс для запиту веб-сторінок, їх перегляду (і виведення на інші зовнішні пристрої) і, при необхідності, відправки, введених користувачем даних на сервер.

Cascading Style Sheets (CSS, каскадні таблиці стилів) – також є мовою програмування, а представляють технологію опису зовнішнього вигляду документа, написаного мовою розмітки. Переважно використовується як засіб оформлення веб-сторінок у форматі HTML та XHTML, але може застосовуватись з будь-якими видами документів у форматі XML, включаючи SVG та XUL.

Extensible Hypertext Markup Language (XHTML, розширювана мова розмітки гіпертексту) – мова розмітки веб-сторінок, за можливостями можна порівняти з HTML, але створена на базі XML. Як і HTML, XHTML відповідає специфікації SGML, оскільки XML є її підмножиною.

Стандарт XHTML побудований не як самодостатній опис мови, а як перелік відмінностей між HTML 4.01 та XHTML.

eXtensible Markup Language (XML, «розширювана мова розмітки») – рекомендована Консорціумом Всесвітньої павутини мова розмітки, що фактично є зведенням загальних синтаксичних правил. XML – текстовий формат, призначений для зберігання структурованих даних (замість існуючих файлів баз даних), для обміну інформацією між програмами, а також для створення на основі більш спеціалізованих мов розмітки (наприклад, XHTML), іноді званих словниками. XML є спрощеним підмножиною мови SGML. Роком народження XML можна вважати 1996, наприкінці якого з'явився чорновий варіант специфікації мови, або 1998, коли ця специфікація була затверджена.

Метою створення XML було забезпечення сумісності при передачі структурованих даних між різними системами обробки інформації, особливо під час передачі таких даних через Інтернет. Словники, що базуються на XML (наприклад, RDF, RSS, MathML, XHTML, SVG), самі по собі формально описані, що дозволяє програмно змінювати та перевіряти документи на основі цих словників, не знаючи їх семантики, тобто не знаючи смислового значення елементів. Важливою особливістю XML також є застосування про просторів імен (namespace).

Найбільш поширені три способи перетворення XML-документа у вид, що відображається користувачеві:

- застосування стилів CSS;
- застосування перетворення XSLT;
- написання будь-якою мовою програмування обробника XML-документа.

Без використання CSS або XSL XML-документа відображається як простий текст у більшості Веб-браузерів. Деякі браузери, такі як Internet Explorer, Mozilla та Mozilla Firefox, відображають структуру документа у вигляді дерева, дозволяючи згортати та розгортати вузли за допомогою натискання клавіші миші.

Об'єктно-орієнтована скриптова мова програмування JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів застосунків. Найширше застосування знаходить у браузерах як мову сценаріїв для надання інтерактивності веб-сторінок.

Основні архітектурні риси: динамічна типізація, слабка типізація, автоматичне керування пам'яттю, прототипне програмування, функції як об'єкти першого класу.

JavaScript має ряд властивостей об'єктно-орієнтованої мови, але реалізоване в мові прототипування зумовлює відмінності у роботі з об'єктами порівняно з традиційними об'єктно-орієнтованими мовами. Крім того, JavaScript має ряд властивостей, властивих функціональним мовам: функції як об'єкти першого класу, об'єкти як списки, каринг, анонімні функції, замикання, що надає додаткову гнучкість мові.

Структурно JavaScript можна у вигляді об'єднання трьох чітко помітних друг від друга частин:

- ядро (ECMAScript);
- об'єктна модель браузера (Browser Object Model чи BOM);
- об'єктна модель документа (Document Object Model чи DOM).

Якщо розглядати JavaScript в оточенні от браузера, то об'єктна модель браузера і об'єктна модель документа можуть не підтримуватися. Об'єктну модель документа іноді розглядають як окрему від JavaScript сутність, що узгоджується з визначенням DOM як незалежного від мови інтерфейсу документа.

Visual Basic Scripting Edition (VB Script) – скриптова мова програмування, що інтерпретується компонентом Windows Script Host. Є об'єктно-

орієнтованою мовою програмування, схожою на JavaScript. На відміну від JavaScript, підтримує і функції, і процедури (підпрограми), не чутливий до регістру символів в описі змінних.

Скрипти на мові VBScript найчастіше використовуються в таких областях, що використовують продукцію Microsoft:

- автоматизація адміністрування систем Windows;
- серверний програмний код у сторінках ASP;
- клієнтські скрипти у браузері Internet Explorer.

Adobe Flash – це мультимедійна платформа (раніше відома як Macromedia Flash), яка використовується для створення векторної анімації та інтерактивних програм, а також для інтеграції відео у веб-сторінки.

В основі Flash лежить векторний морфінг, тобто плавне «перетікання» одного ключового кадру в інший. Це дозволяє робити складні мультиплікаційні сцени, задаючи лише кілька ключових кадрів кожному за персонажа.

Flash використовує мову програмування ActionScript, що базується на ECMAScript. Основний недолік Flash-застосунків – надмірна вимогливість до ресурсів процесора. Недостатня потужність комп'ютера може вплинути на продуктивність операційної системи в цілому або призвести до спотворення результатів роботи Flash-програми, пов'язаних з відображенням анімації або підрахунком часу.

Інший важливий недолік полягає в тому, що не завжди є можливість запустити Flash-застосунок або вона пов'язана з деякими труднощами (наприклад, необхідно встановити плагін або оновити його до останньої версії). Також використання Flash для розміщення текстової інформації запобігає її індексуванню пошуковими системами. Однак існує безліч способів вирішити цю проблему.

Microsoft Silverlight – це мультимедійна технологія, схожа на розв'язувані завдання з Adobe Flash. Також Microsoft Silverlight – це плагін для браузера, який дозволяє запускати застосунки, що містять анімацію, векторну

графіку та аудіо-відео ролики, що є характерним для RIA (Rich Internet application).

Silverlight поєднує мультимедіа, графіку, анімацію та інтерактивність в одній програмній платформі. Він був розроблений для роботи з XAML і з мовами Microsoft .NET. XAML використовується для розмітки сторінок, які використовують векторну графіку та анімацію. Текст, який міститься в Silverlight застосунках, доступний для пошукових систем, оскільки він не компілюється, а доступний у вигляді XAML.

Silverlight дозволяє динамічно завантажувати XML та використовувати DOM для взаємодії з ним. Silverlight містить об'єкт Downloader, завдяки якому можна завантажувати скрипти, медіа файли і т. д., якщо це потрібно програмі. Починаючи з версії 2.0, логіка програми може бути описана в будь-якій мові.).

2.2 Аналіз серверних мов програмування

Можливості, пов'язані з виконанням скриптів у веб-браузерах, мають суттєві технологічні обмеження, а також внаслідок застосування адміністративних та/або налаштувань користувача. Для подолання даних обмежень з'явилися технології, пов'язані з виконанням коду на веб-сервері. Тобто. коли користувач робить запит через свій браузер, то викликана сторінка спочатку обробляється на сервері (виконуються програми, пов'язані з цією сторінкою) і лише після цього повертається до відвідувача, що запросив її у вигляді HTML-файлу.

У цьому випадку коректність роботи повністю залежить від роботи самого веб-сервера та його програмного забезпечення.

Важливим компонентом роботи серверних мов є системи управління базами даних. За своєю суттю це також сервер (фізичний або програмний), на якому в упорядкованому вигляді зберігається різна інформація, яку можуть

запросити користувачі. Найбільш поширеними СУБД для веб-застосунків на даний час є:

- MySQL;
- PostgreSQL;
- MS SQL Server;
- SQLite;
- Oracle;
- IBM DB2.

Server Side Includes (SSI, «Увімкнення на стороні сервера») – нескладна мова для динамічної «складання» веб-сторінок на сервері з окремих складових частин та видачі клієнту отриманого HTML-документа. Реалізовано у веб-сервері Apache. Синтаксис SSI дозволяє включати інші SSI-сторінки, викликати зовнішні CGI-скрипти, реалізовувати умовні операції (if/else), працювати зі змінними тощо. Завдяки крайній простоті мови, складання SSI-сторінок відбувається дуже швидко, проте багато можливостей повноцінних мов програмування, наприклад, робота з файлами, у SSI відсутні. Окремі елементи синтаксису SSI використовуються і в інших мовах скриптів, наприклад, в ASP.

PHP: Hypertext Preprocessor («PHP: препроцесор гіпертексту»), Personal Home Page Tools – («Інструменти для створення персональних веб-сторінок») – скриптова мова програмування загального призначення, що інтенсивно використовується для розробки веб-застосунків. В даний час підтримується переважна більшість хостинг-провайдерів і є одним з лідерів серед мов програмування, що застосовуються для створення динамічних веб-сайтів.

В області веб-програмування PHP – одна з найпопулярніших скриптових мов (поряд з JSP, Perl та мовами, що використовуються в ASP.NET) завдяки своїй простоті, швидкості виконання, багатій функціональності, кросплатформенності та поширенню вихідних кодів на основі ліцензії PHP. Популярність у сфері побудови веб-сайтів визначається наявністю великого набору вбудованих засобів для розробки веб-застосунків.

Perl – це високорівнева інтерпретована динамічна мова програмування загального призначення, створена в 1987 р. Ларрі Уоллом, лінгвістом за освітою. Назва мови є аббревіатурою, яка розшифровується як Practical Extraction and Report Language – «практична мова для вилучення даних та складання звітів». Основною особливістю мови вважаються багаті можливості для роботи з текстом, у тому числі реалізовані за допомогою регулярних виразів. Перл успадкував багато властивостей мов C, shell script, awk. Perl також відомий величезною колекцією додаткових модулів CPAN.

ASP.NET – це технологія створення веб-застосунків та веб-сервісів від компанії Microsoft. Вона є складовою платформи Microsoft .NET та розвитком більш старої технології Microsoft ASP.

Хоча ASP.NET бере свою назву від старої технології Microsoft ASP, вона значно від неї відрізняється. Microsoft повністю перебудувала ASP.NET, ґрунтуючись на Common Language Runtime (CLR), який є основою всіх програм Microsoft .NET. Розробники можуть писати код для ASP.NET, використовуючи практично будь-які мови програмування, в тому числі і входять до комплекту .NET Framework (C#, Visual Basic.NET, і JScript .NET). ASP.NET має перевагу в швидкості в порівнянні зі скриптовими технологіями, так як при першому зверненні код компілюється і поміщається в спеціальний кеш, і згодом тільки виконується, не вимагаючи часу на парсинг, оптимізацію, і т.д.

Переваги ASP.NET перед ASP:

- код що компілюється виконується швидше, більшість помилок відловлюється ще стадії розробки;
- значно покращена обробка помилок часу виконання з використанням блоків try..catch;
- елементи управління (controls) дозволяють виділяти часто використовувані шаблони, такі як меню сайту;
- використання метафор, які вже застосовуються у Windows-застосунках, наприклад, таких як елементи керування та події;

- розширюваний набір елементів управління та бібліотек класів дозволяє швидше розробляти програми;
- ASP.NET спирається на багатомовні можливості .NET, що дозволяє писати код сторінок на VB.NET, Delphi.NET, Visual C#, J# і т.д.;
- можливість кешування всієї сторінки чи її частини збільшення продуктивності;
- можливість кешування даних, що використовуються на сторінці;
- можливість поділу візуальної частини та бізнес-логіки за різними файлами («code behind»);
- модель моделі обробки запитів, що розширюється;
- розширена подієва модель;
- модель, що розширюється серверних елементів управління;
- наявність master-сторінок для завдання шаблонів оформлення сторінок;
- підтримка CRUD операцій під час роботи з таблицями через GridView;
- вбудована підтримка AJAX.

3 ОПИС ЕТАПІВ РОЗРОБКИ ВЕБ-ПОРТАЛУ ДЛЯ НАУКОВОГО ТОВАРИСТВА УНІВЕРСИТЕТУ

Перший етап розробки веб-порталу для Наукового товариства університету, розпочинається зі створення макетів та розробки унікального / індивідуального дизайну. Під час розробки дизайну, було обрано використовувати чотири основні кольори (див.рис.3), а саме:

- темно синій #062850;
- синій #1866ce;
- світло синій #3876cac4;
- жовтий #edbd2f.

Ці чотири кольори, використовуються задля оформлення основних блоків ІС: кнопок, заднього фону, header, footer тощо.

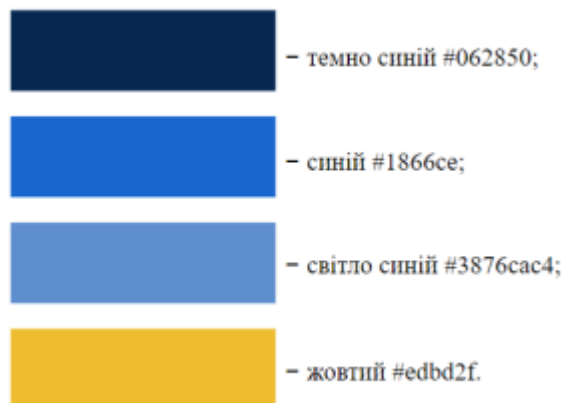


Рисунок 3 – Опис основних, головних чотирьох кольорів

Для тексту, що використовується в header, заголовків або ж прописаний на синьому фоні застосовується білий колір (#ffffff). У блоці footer, використовується сірий колір (#aca9a9). Посилання оформлюються жовтим кольором (#edbd2f). А як основний колір для тексту, застосовується чорний колір (#000000).

На рис. 4 зображено скріншот головної сторінки веб-порталу.



Рисунок 4 – Скріншот головної сторінки веб-порталу

Як можна побачити, header складається з меню та кнопки перемикання мови. На головній сторінці зображено також Назву , блок Вітальне слово, блок Новини, блок Проєкти. У блоці footer, відтворена інформація щодо контактних даних Наукового товариства, а саме: назва товариства, адреса, телефон, електронна пошта та посилання на сайт.

Блоки header та footer, повторюються на всіх існуючих сторінках ІС.

Нижче описаний фрагмент коду, який після запиту до БД, до відповідної таблиці у відповідь прийме та підставиться необхідну інформація.

```
<?php
    if ($db->num_rows($resultCont)) {
        while ($data = $db->fetch_row($resultCont)) {
            $mm[] = $data;
        }
    }
    ?>
<div class="footer-contacts__inner-block">
    <div class="footer-contacts__structure-address">
    
    <p> <?php echo $data[6] ?></p>
    </div>
    <div class="footer-contacts__structure-tel">
    
    <a href="tel:<?php echo $data[12] ?>">
    <?php echo $data[12] ?>
    </a> </div>
    <div class="footer-contacts__structure-mail">
    
    <a href="mailto:<?php echo $data[9] ?>" >
    <?php echo $data[9] ?>
    </a> </div>
    <div class="footer-contacts__structure-site">
    
    <a target="_blank" href="<?php echo $data[11] ?>">
    <?php echo $data[11] ?></a></div>
</div>
<?php
    }
}
?>
```

На рис. 5 представлено сторінку Новини на якій відображаються всі новини, що мають відмітку «Відобразити на сайті». Дані (дата створення новини, заголовок, короткий та повний контент, галерея) вносяться через Адміністративну панель. Новини у даному блоці генеруються автоматично.

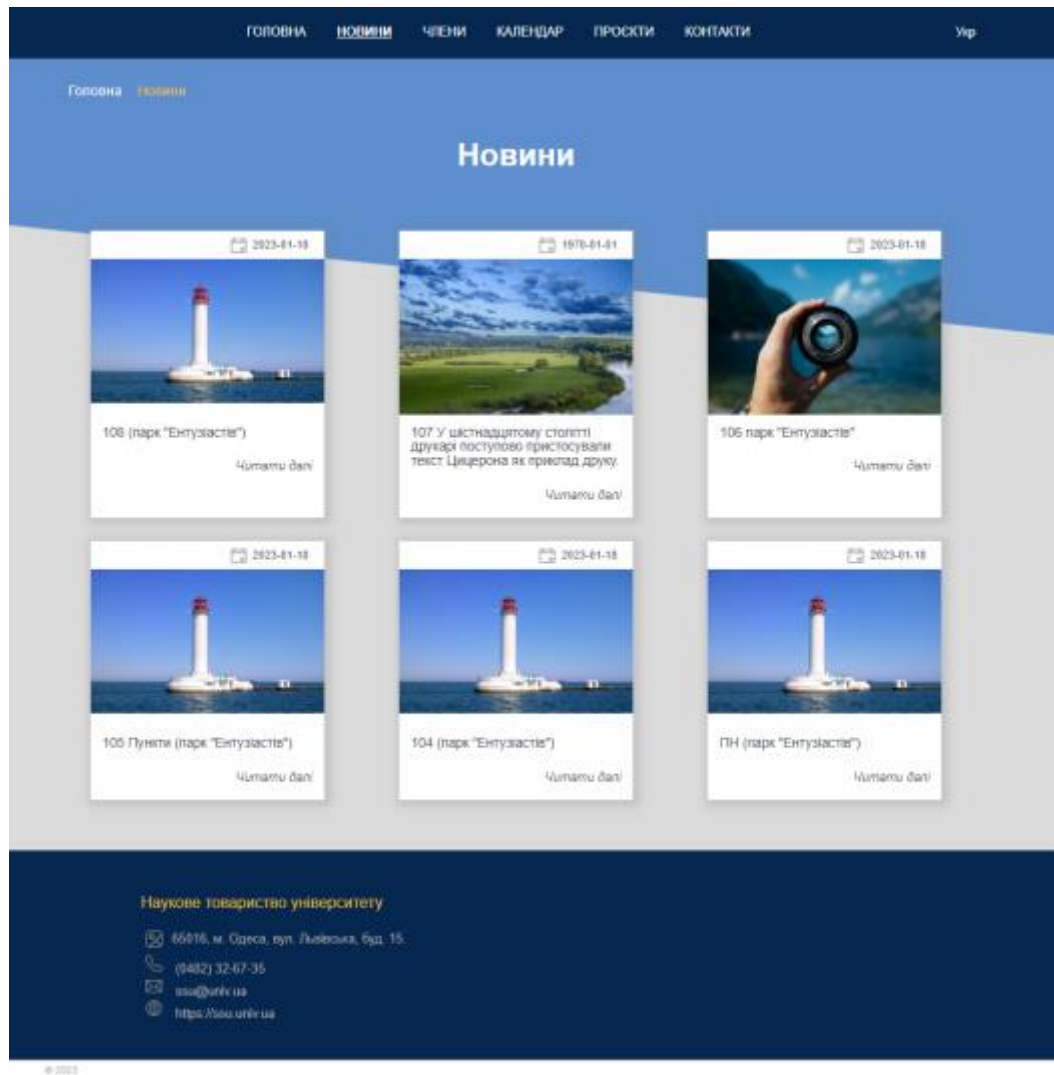


Рисунок 5 – Скріншот сторінки Новини

Наступну сторінку, яку можна розглянути, це сторінка Календар (рис.6) на якій відображається інформація щодо подій які відносяться до наукового товариства. На дану сторінку інтегровано, підключено на налаштовано карту Leaflet OpenStreetMap. Leaflet [6] – це провідна бібліотека JavaScript з відкритим кодом для інтерактивних карт, зручних для мобільних пристроїв. Він важить лише близько 42 КБ JS і має всі функції відображення, які коли-небудь

знадобляться більшості розробників. Leaflet розроблена з урахуванням простоти, продуктивності та зручності використання. Ця бібліотека ефективно працює на всіх основних настільних і мобільних платформах, може бути розширена за допомогою багатьох плагінів, має гарний, простий у використанні та добре задокументований API, а також простий, читабельний вихідний код.

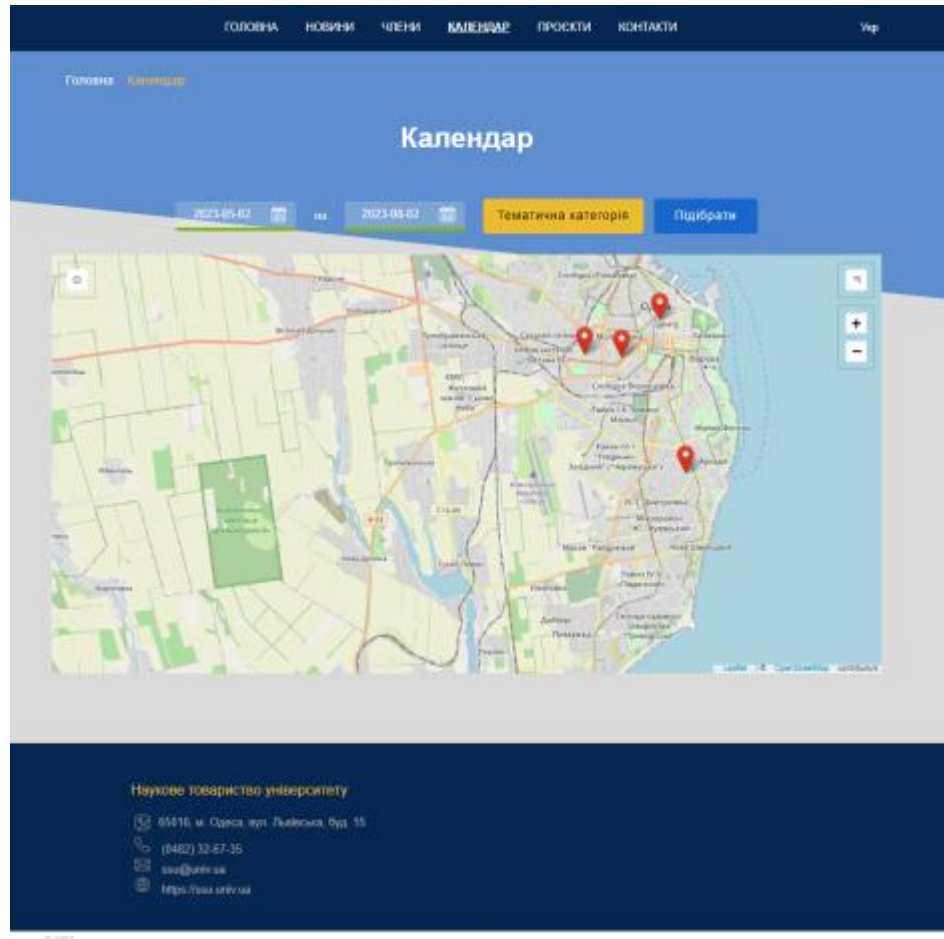


Рисунок 6 – Скріншот сторінки Календар

Для того щоб скористатися користувачеві всіма можливостями даної сторінки, необхідно по-перше встановити діапазон дати з – по, далі слід обрати тематичну категорію за якою будуть відбиратися події для подальшого відображення. Для запуску скрипта, та подальшого виведення інформації на поточній сторінці – необхідно натиснути на кнопку Підібрати.

Після роботи скрипта, на карті будуть відображені мітки з відповідними подіями які відповідають параметрам, що напередодні вказав користувач.

Що стосується міток, якщо натиснути на мітку яка відображена на карті, можна переглянути певну інформацію (перелік полів, які будуть відображатися можна налаштувати через внесення змін у відповідний скрипт). У даному випадку, відображається інформація щодо id (відповідає id в БД), назва та дати проведення даної події (див.рис. 7)

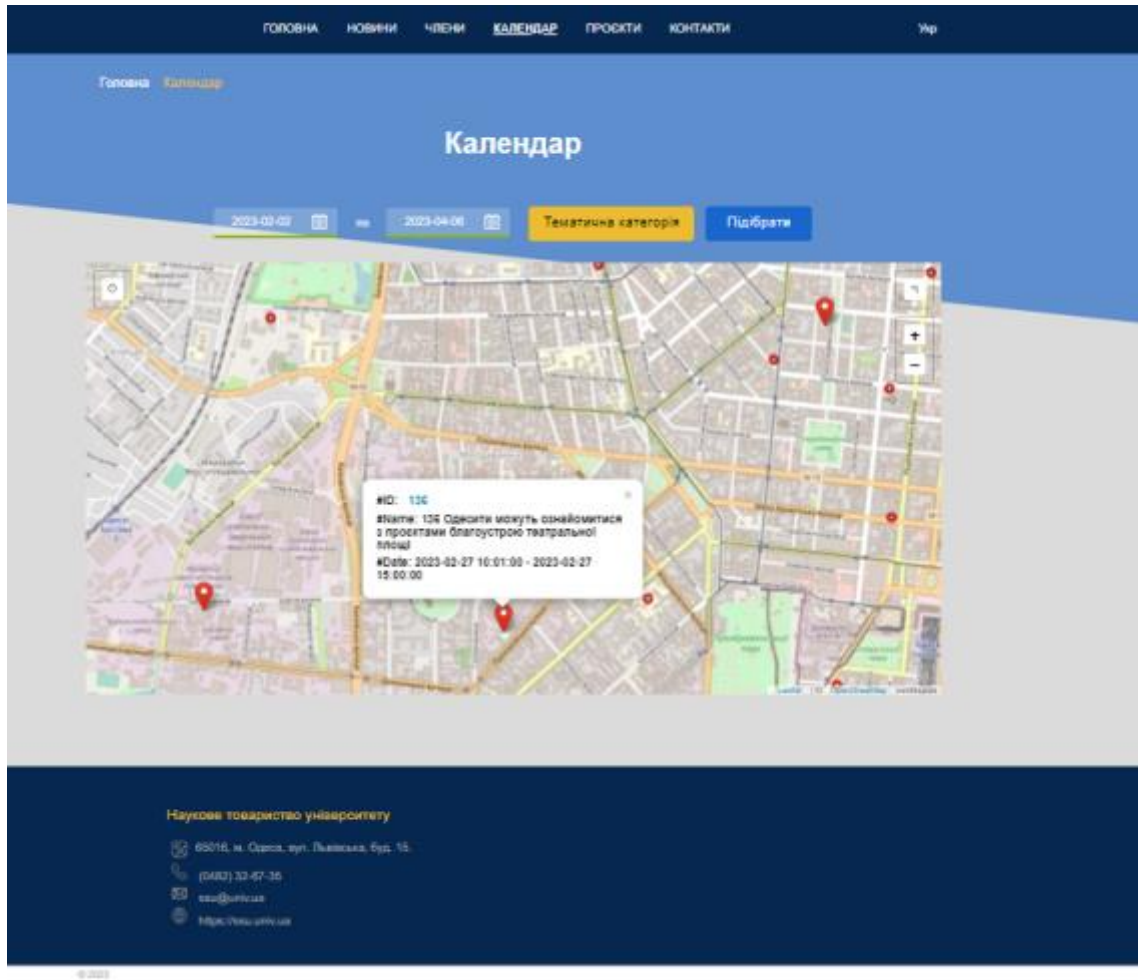


Рисунок 7 – Скріншот сторінки Календар

На сторінці Проєкти відображаються блоки проєктів, які було розроблено, розробляються в даний час або ж плануються розроблятися Науковим товариством університету. Блок який містить інформацію загальну про проєкт, складається з: дати коли була висвітлена інформація щодо проєкту, заголовок даного проєкту та короткий контент, всі ці дані вносяться через Адміністративну панель, та автоматично генеруються в залежності від вказаних

даних вибраних для відображення блоку на відповідній сторінці. Задля переходу та ознайомлення з більш детальною інформацією слід натиснути на кнопку “Читати далі” (рис.8).

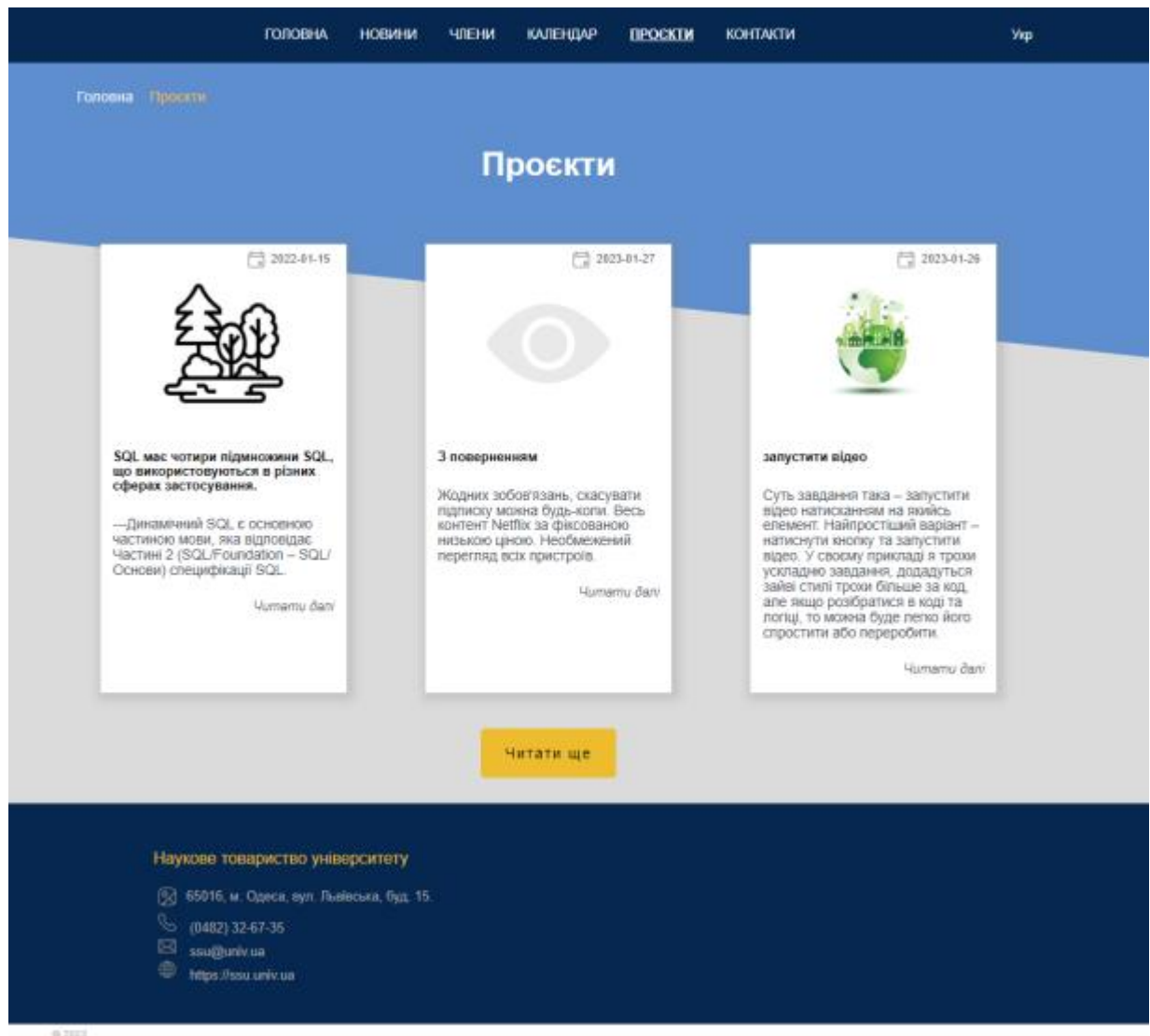


Рисунок 8 – Скріншот сторінки Проекти

На рис. 9 зображено внутрішню сторінку parent page Проекти, після натискання на кнопку “Читати далі”. На згенерованій сторінці відображається наступна інформація:

- breadcrumb (навігаційна стежка, «хлібні крихти») допоміжний механізм навігації, що використовується в інтерфейсах користувача;
- заголовок проекту;

- коротка інформація/контент про проєкт;
- основна інформація про проєкт;
- за наявності, відображення галереї.

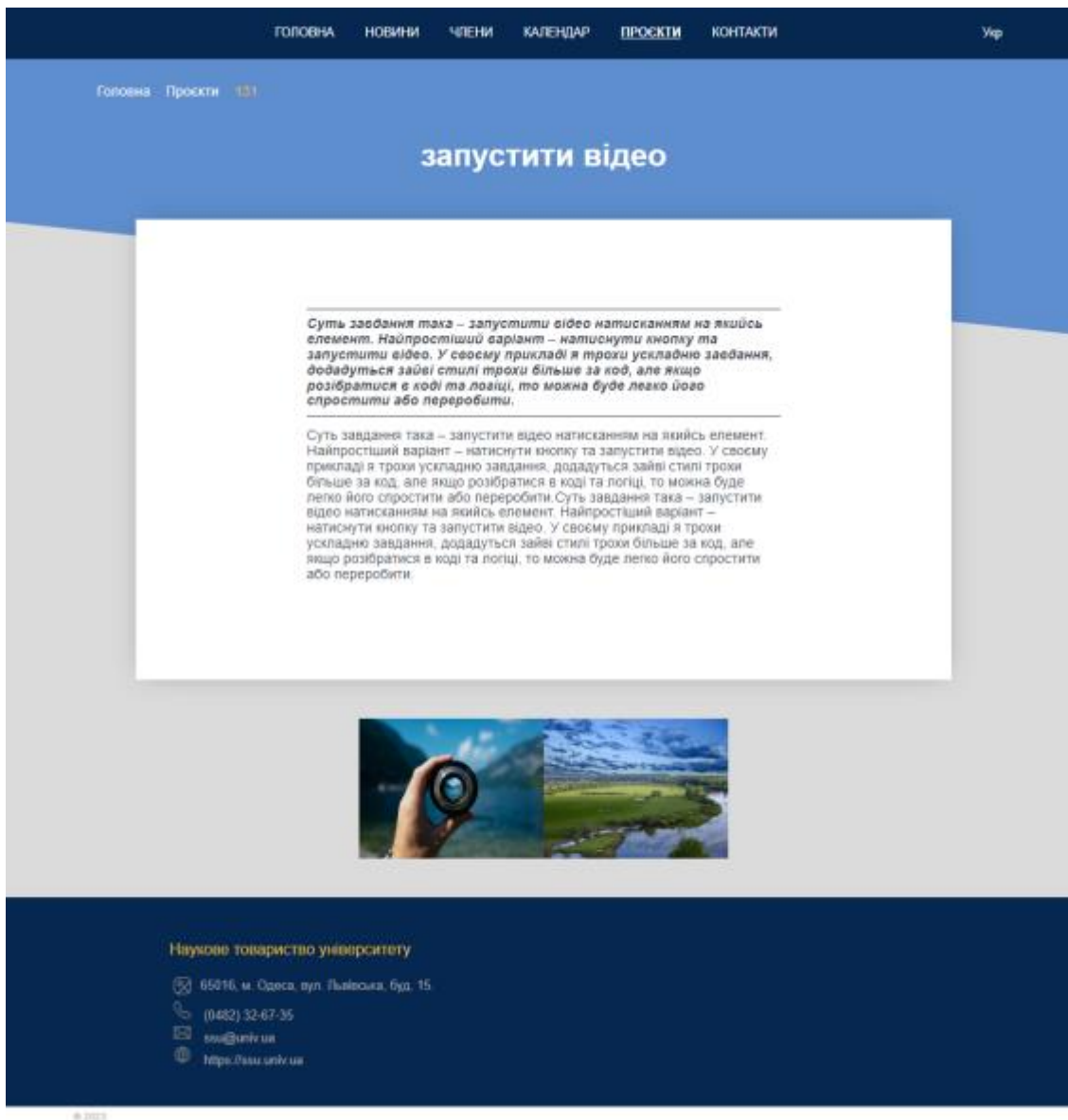


Рисунок 9 – Скріншот внутрішньої сторінки Проєкти

Далі, також слід звернути увагу на побудову сторінки Контакти (рис.10). На даній сторінці відображається контактна інформація (дублюється в блок footer), а також форма зворотного зв'язку, яка повинна містити заповнені наступні поля Ім'я, Пошта, Телефон, Тема повідомлення, Повідомлення.

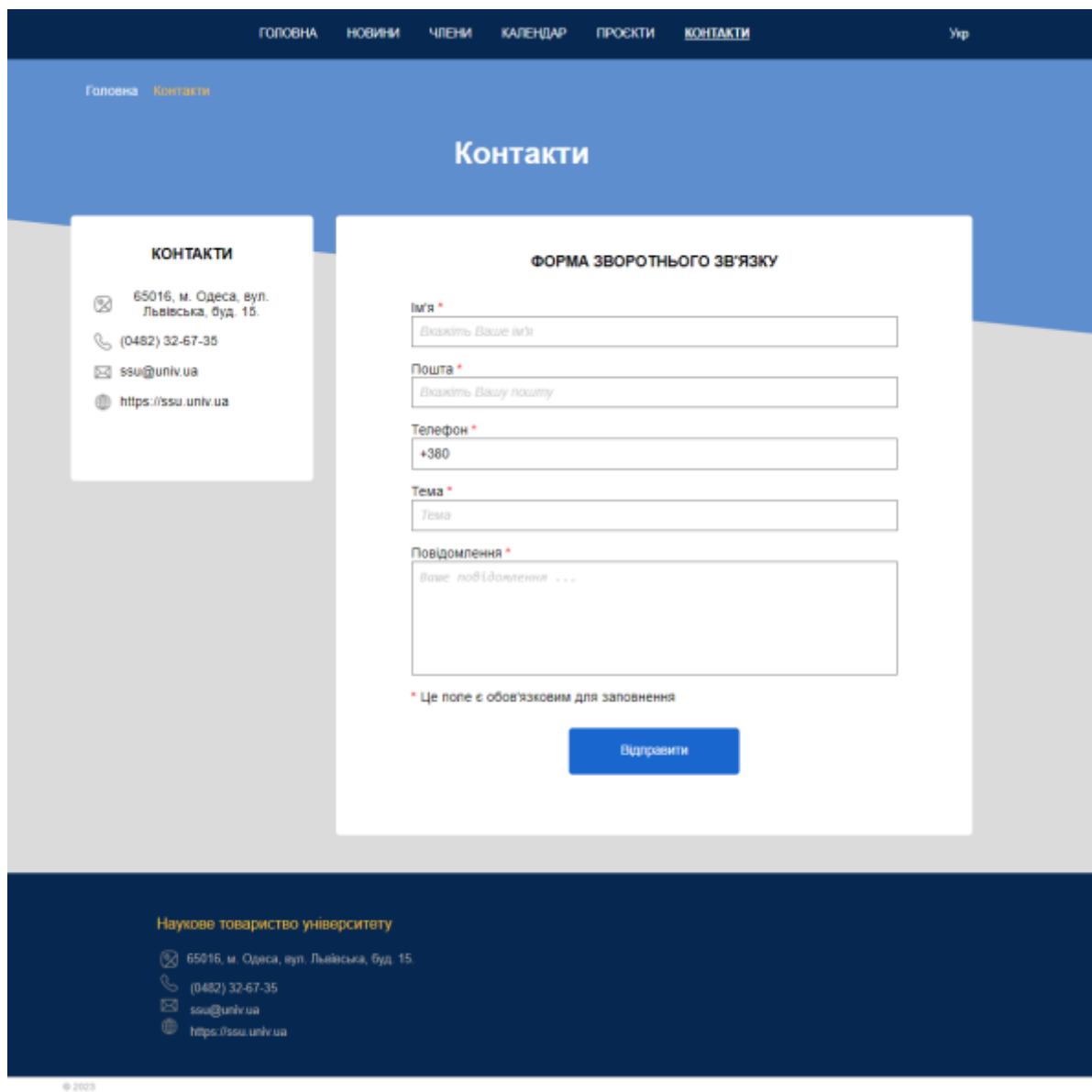


Рисунок 10 – Скріншот сторінки Контакти

```

<?php
    $contact_text = array (
        0 => ($lang == 'ru' ? 'Форма обратной связи' : ($lang
== 'en' ? 'Feedback Form' : 'Форма зворотного зв\''язку')),
        1 => ($lang == 'ru' ? 'Имя' : ($lang == 'en' ? 'Name' :
'Им\''я')),
        2 => ($lang == 'ru' ? 'Почта' : ($lang == 'en' ?
'Email' : 'Пошта')),
        3 => ($lang == 'ru' ? 'Телефон' : ($lang == 'en' ?
'Phone' : 'Телефон')),
        4 => ($lang == 'ru' ? 'Тема' : ($lang == 'en' ?
'Subject' : 'Тема')),
        5 => ($lang == 'ru' ? 'Сообщение' : ($lang == 'en' ?
'Message' : 'Повідомлення')),

```

```

        6 => ($lang == 'ru' ? 'Отправить' : ($lang == 'en' ?
'Submit' : 'Відправити')),
        7 => ($lang == 'ru' ? 'Это поле обязательно для запол-
нения' : ($lang == 'en' ? 'This field is required' : 'Це поле є
обов\язковим для заповнення')),
        8 => ($lang == 'ru' ? 'Обязательное поле.' : ($lang ==
'en' ? 'Required field.' : 'Обов\язкове поле.')),
        9 => ($lang == 'ru' ? 'Неверный адрес электронной поч-
ты.' : ($lang == 'en' ? 'Invalid Email Address.' : 'Невірна ад-
реса електронної пошти.')),
        10 => ($lang == 'ru' ? 'Неправильный номер телефона.' :
($lang == 'en' ? 'Invalid Phone Number.' : 'Недійсний номер те-
лефону.'))
    );
?>

```

Вищевказаний фрагмент коду написаний на мові програмування PHP. Він, відповідає за створення масиву з відповідною інформацією/текстом, яка в подальшому буде змінюватися в залежності від мови, на якій відображається веб-портал.

Далі слід розглянути фрагмент коду, який допоможе сформувати безпосередньо форму зворотного зв'язку.

```

<div class="input-row">
  <label style="padding-top: 20px;">
<?php echo $contact_text[1] ?>
<span class="required">*</span>
  </label>
  <span id="userName-info" class="info"></span><br />
  <input type="text" placeholder="<?php echo ($lang == 'ru' ? 'У-
кажите Ваше имя' : ($lang == 'en' ? 'Enter your name' : 'Вкажіть
Ваше ім\я')) ?>"
class="input-field" name="userName" id="userName" />
</div>

<p class='<?php echo $type; ?> Message'><?php echo
$message; ?></p>

```

```

    $contact_mail_text = array (
        0 => ($lang == 'ru' ? 'Ваша контактная информация ус-
пешно сохранена.' : ($lang == 'en' ? 'Your contact information
is saved successfully.' : 'Вашу контактну інформацію успішно
збережено.')),

```

```

1 => ($lang == 'ru' ? 'Сообщение электронной почты от-
правлено.' : ($lang == 'en' ? 'Email message sent.' : 'Електрон-
не повідомлення надіслано.')),
2 => ($lang == 'ru' ? 'Извините, что-то пошло не
так.' : ($lang == 'en' ? 'Sorry, something went wrong.' : 'Виба-
че, щось пішло не так.'))
);

```

Наступний фрагмент коду, дасть можливість безпосередньо на сторінці, без використання додаткового JS скрипта перевірити форму зворотного зв'язку перед відправкою форми. Нижче описана частина коду, яка перевіряє форму перед відправкою на відповідність полів userEmail та userPhone та виконує перевірку на відповідність регулярному виразу.

```

if (!empty($_POST["send"])) {
    $message = '';

    if (isset($_POST['userEmail']) && preg_match('/^[\\w\\d\\.-
]+@[\\w\\d\\.\\-]+\\. [a-z]{2,4}$/i', $_POST['userEmail'])) {
        $email = $_POST["userEmail"];
    }
    else {
        $message .= "WOW.ERROR - in the block 'email'";
    }

    if (isset($_POST['userPhone']) && preg_match('/^[\\d\\-
\\(\\)\\+\\s]{7,24}$/i', $_POST['userPhone'])) {
        $phone = $_POST["userPhone"];
    }
    else {
        $message .= "WOW.ERROR - in the block 'phone'";
    }
}

```

На рис. 11 представлено скріншот модального вікна, на якому зображено кнопки для перемикання та вибору мови на якій буде відображатися веб-портал. Обрана користувачем мова, зберігається в Куки, що в свою чергу покращить взаємодію користувача з веб-порталом. Зручність у тому, що браузер запам'ятовує вказані користувачем параметри налаштування (в даному випадку мову інтерфейсу) та при повторному запуску сторінки, відобразить її взявши мову з Куки. HTTP-cookie або «Куки» – у комп'ютерній те-

рмінології поняття, яке використовується для опису інформації у вигляді текстових або бінарних даних, отриманих від веб-сайту на веб-сервері, яка зберігається у клієнта, тобто браузера, а потім відправлена на той самий сайт, якщо його буде повторно відвідано.

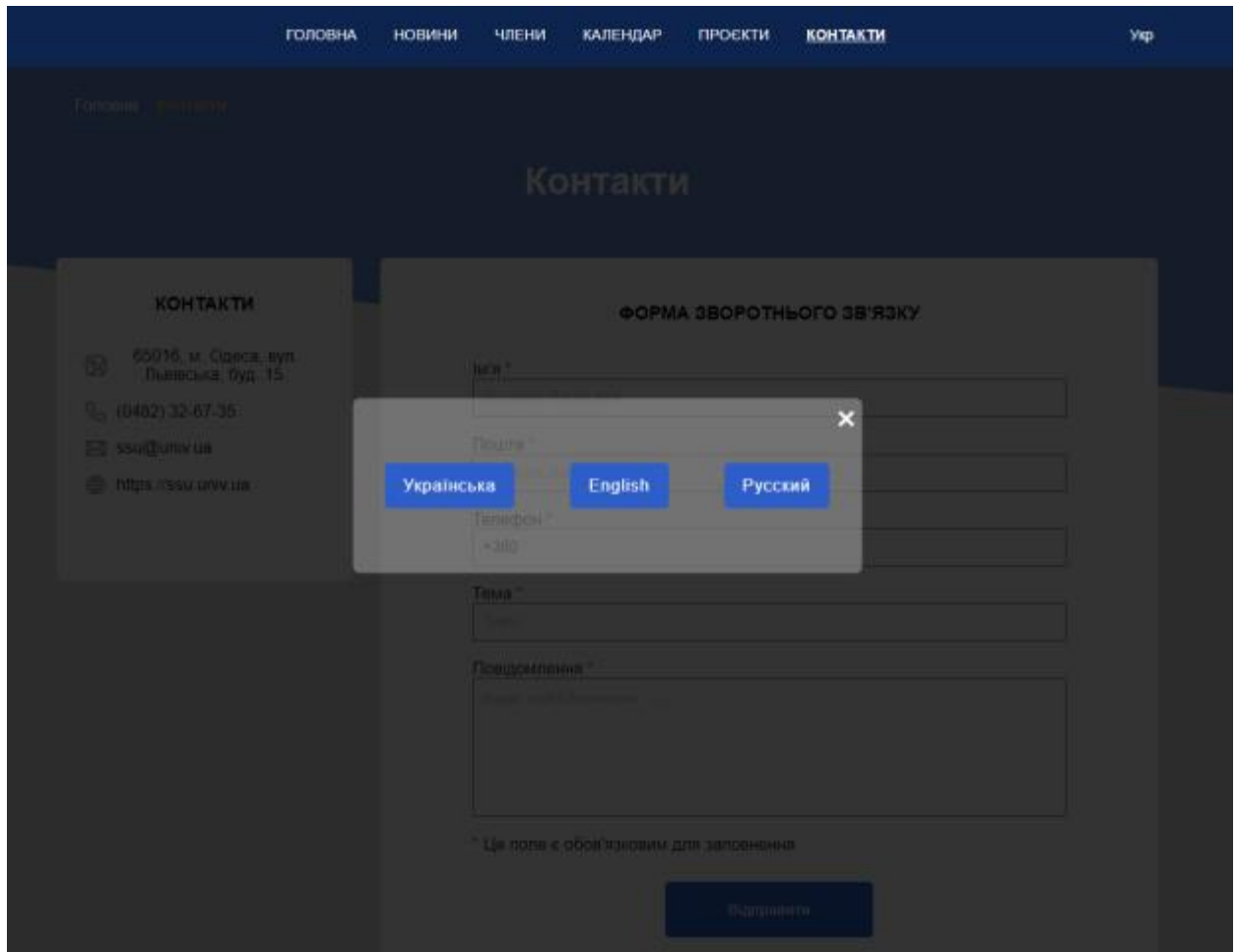


Рисунок 11 – Скріншот меню вибору мови

Нижче наведено фрагмент коду, в якому створюється змінна типу масив з ім'ям \$langs.

```
$langs = array(
    'ua' => array('Українська', 'Укр', 'uk-UA'),
    'en' => array('English', 'Eng', 'en-US'),
    'ru' => array('Русский', 'Рус', 'ru-UA')
);
```

Фрагмент коду, який відповідає за кнопку в header, а саме за текст який буде відображено. Орієнтується на змінну \$langs, значення якої записується в Куки.

```
<div id="language" class="header__nav-button-lang" >
  <a class="opener">
<?php
  switch ($lang) {
    case 'ua':
echo 'Укр';
break;

    case 'en':
echo 'Eng';
break;

    case 'ru':
echo 'Рус';
break;

    default:
echo 'Укр';
break;
  }
?>
</a>
</div>
```

Наприклад, якщо в модальному вікні вибрати мову якою буде відображатися інформація на даному веб-порталі – Англійська. Можна отримати відповідний вигляд інтерфейсу (див.рис. 12).

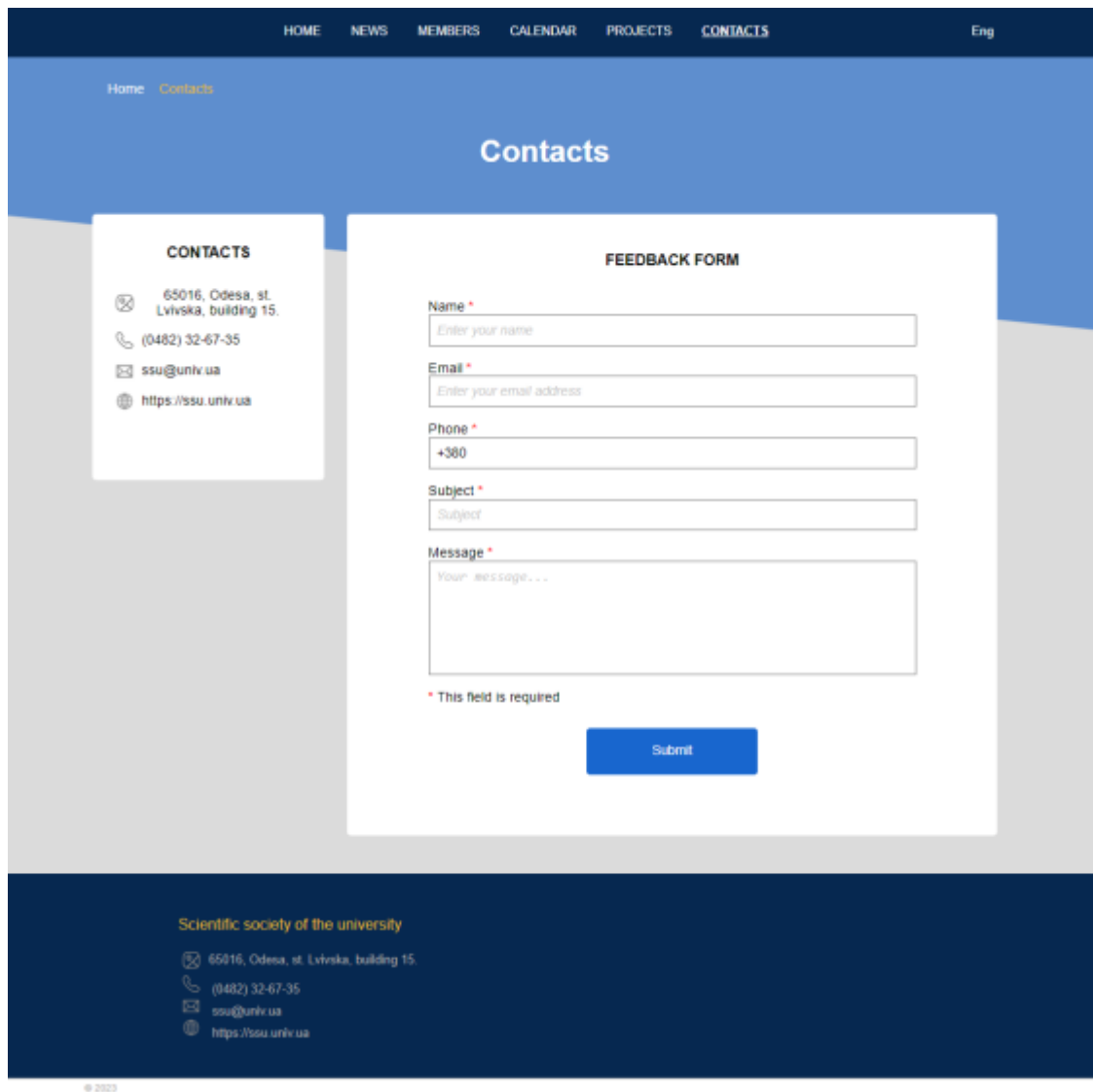


Рисунок 12 – Скріншот сторінки Контакти при виборі мови сторінки “Англійська”

Далі, слід розглянути побудову та структуру БД. Під час роботи над даним проектом, було обрано використовувати середовище для розробки БД phpMyAdmin.

phpMyAdmin [7] – це безкоштовний програмний інструмент, написаний на PHP, призначений для адміністрування MySQL через Інтернет. phpMyAdmin підтримує широкий спектр операцій з MySQL і MariaDB. Операції, які часто використовуються (керування базами даних, таблицями, стовпцями, відношеннями, індексами, користувачами, дозволами тощо) можна

виконувати через інтерфейс користувача, у той час як у вас залишається можливість безпосередньо виконувати будь-який оператор SQL.

phpMyAdmin – це зрілий проєкт зі стабільною та гнучкою кодовою базою.

Проєкт phpMyAdmin є членом Software Freedom Conservancy. SFC – це некомерційна організація, яка допомагає просувати, покращувати, розвивати та захищати проєкти безкоштовного програмного забезпечення та програмного забезпечення з відкритим кодом (FLOSS).

Схема бази даних ssu складається з 5 таблиць, а саме: pages, galleries, feedback, settings, users (рис. 13).

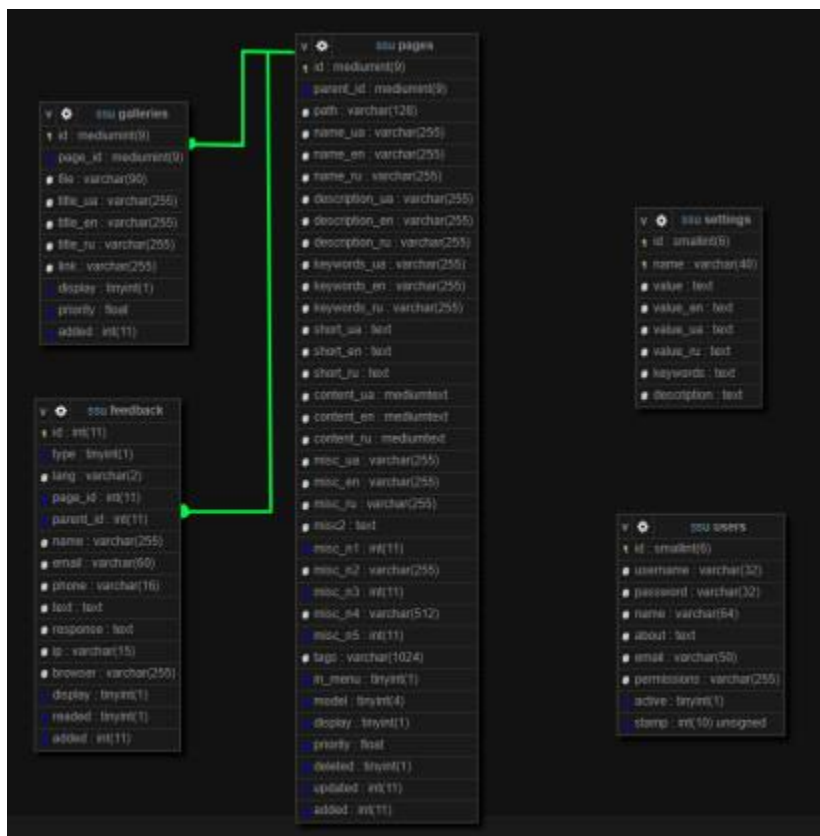


Рисунок 13 – Схема бази даних ssu

Таблиці pages, galleries, feedback пов'язані між собою. PRIMARY KEY поле id таблиці pages, пов'язане з полями page_id в galleries та feedback. Таким чином, відбувається прив'язка відповідних даних, які внесені в таблиці до id батьківського запису.

В свою чергу, таблиці settings та users розглядаються окремо. Оскільки, перша таблиця відповідає за внесення та подальше використання даних задля заміни відповідно до мови яку обрав для відображення інтерфейсу користувач. А друга таблиця, users за відповідає створення нових та внесення змін до вже існуючих користувачів, які мають доступ до Адміністративної панелі. На рис. 14 зображено структуру БД ssu, в якій вказано всі існуючі таблиці та їх розмір / наповненість.

Table	Action	Rows	Type	Collation	Size	Overhead
feedback		0	InnoDB	utf8_general_ci	32.0 K18	-
galleries		18	InnoDB	utf8_general_ci	32.0 K18	-
pages		26	InnoDB	utf8_general_ci	128.0 K18	-
settings		28	InnoDB	utf8_general_ci	32.0 K18	-
users		3	InnoDB	utf8_general_ci	16.0 K18	-
5 tables	Sum	75	InnoDB	utf8_general_ci	240.0 K18	0 B

Рисунок 14 – Структура бази даних ssu

Далі, слід перейти та акцентувати увагу, на розробці Адміністративної панелі. Адміністративна панель – це важлива складова розробки ІС. Дана панель, дасть можливість спростити роботу адміністратор, яка буде полягати в наповненні веб-порталу контентом, безпосередньо через взаємодію адміністратору та БД через розроблений інтерфейс. В даному випадку для організації її роботи, не використовуються існуючі CMS. Було обрано, розробити самостійну, унікальну CMS на мові PHP яка буде містити лише необхідний функціонал та до якої будуть підключені лише необхідні плагіни в залежності від поставлених задач. Для прикладу, на рис. 15 зображено інтерфейс зі сторони адміністратора, для розробки / редагування внутрішньої сторінки блоку Новини. Як можна побачити, за допомогою інтерфейсу можна вказати та занес-

ти до БД наступні дані назва, зображення (яке буде використовуватись як головне), зображення для формування галереї, короткий опис, основний контент, чекбокси для відображення на сайті, пріоритет та дата створення даної новини. Слід зауважити, що відповідно до занесених даних в подальшому будуть генеруватися внутрішні сторінки блоку Новини відповідно до шаблону. По такому ж шаблону, створюються і внутрішні сторінки блоків Проекти, Календар (події).

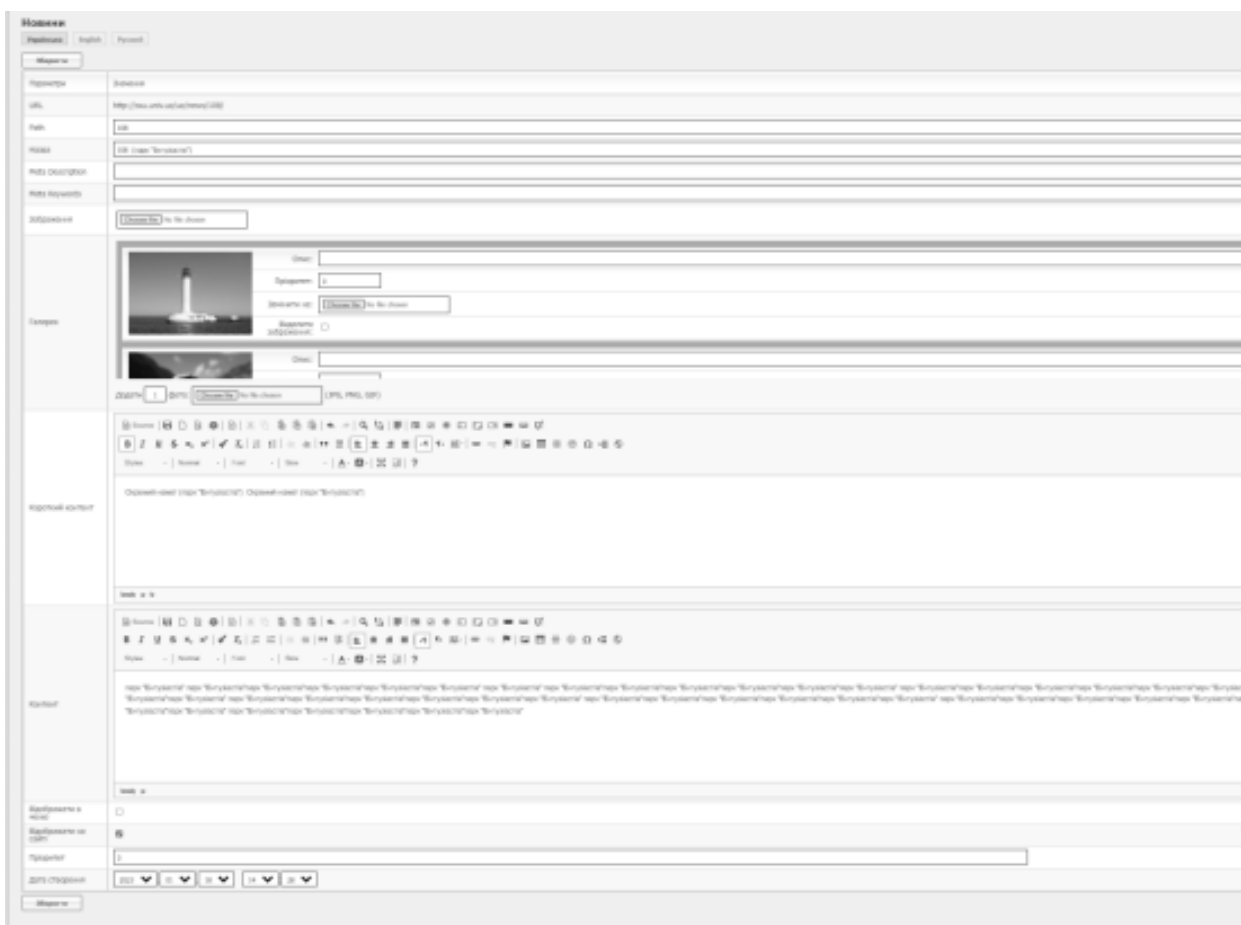


Рисунок 15 – Інтерфейс Адміністративної панелі для створення внутрішньої сторінки блоку Новини

На рис. 16 розглядається інтерфейс вкладки / блоку Контакти. Як можна побачити з даного скріншоту, через Адміністративну панель вносяться та мають можливість змінюватись наступні дані: назва вкладки (Контакти), координати місцезнаходження (за необхідності додається карта на вкладці

Контакти), адреса місцезнаходження (яка буде відображатися на сайті), актуальний номер телефону, електронна пошта, URL офіційного веб-сайту.



Рисунок 16 – Інтерфейс Адміністративної панелі для редагування блоку Контакти

ВИСНОВКИ

Продукти інформаційних технологій є частиною повсякденного життя. У кожній організації є комп'ютерні операції, які мають вирішальне значення для виконання роботи. У організації може виникнути потреба в комп'ютерному програмному забезпеченні, реалізації мережевої архітектури для досягнення цілей організації або розробки програм, ІС (веб-сайтів) чи ігор. Отже, будь-яка організація, яка прагне забезпечити своє майбутнє, потребує інтеграції добре розробленої інформаційної системи.

Така система також корисна для зберігання оперативних даних, документів, записів спілкування та історії. Оскільки ручні дані можуть коштувати багато часу, інформаційні системи можуть бути в цьому дуже корисні. Інформаційна система зберігає дані складним чином, що значно полегшує процес пошуку даних.

Інформаційна система допомагає організації в процесі прийняття рішень. Завдяки ІС надання всієї важливої інформації легше для прийняття кращих рішень. Крім того, ІС дозволяє співробітникам ефективно спілкуватися. Оскільки документи зберігаються в папках, співробітникам легше ділитися ними та отримувати до них доступ.

В результаті виконання кваліфікаційної роботи бакалавра було розроблено веб-порталу для Наукового товариства університету. Якісна веб-розробка, реалізація та впровадження розробленого порталу, дозволяє розширювати межі функціонування ІС, дає можливість збагатити та урізноманітнити ІС, додаючи до його структури вищезгадані дії, а саме в подальшому можливе голосування, форуми, рекламні ролики та багато іншого. Основними етапами такі заходи, як веб-дизайн, верстка сторінок сайту, веб-програмування на стороні сервера та клієнта, а також роботи зі конфігурування веб-сервера.

Під час розробки, були виконані наступні задачі. Розроблено унікальний дизайн веб-порталу – це створення дизайну сторінок, а й робота над елементами бренду.

Сучасний сайт має бути однаково зручним і з комп'ютера, і з телефону, адже більшість клієнтів використовує мобільний інтернет. Важливо, щоб відвідувач міг замовити продукт або відправити заявку з будь-якого девайсу. Даний етап відноситься до розробки адаптивної верстки.

Розроблено Адміністративну панель, яка дозволяє налаштовувати систему управління, щоб вона була простою, зручною та надійною у використанні.

Ексклюзивний сайт має бути наповнений унікальним та цікавим контентом, щоб відвідувач затримався та став клієнтом. Створення такого наповнення – важка та довга робота, яка потребує заглиблення у деталі організації.

Також було проведено тестування проєкту щоб запустити справді якісний та ексклюзивний веб-портал.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. З чого розпочати створення сайту: 11 цінних порад. URL: <https://impulse-design.com.ua/s-chego-nachat-sozdanie-sajta.html> (дата звернення: 02.05.2023)
2. What is Information System? Definition, Examples, & Facts. URL: <https://emeritus.org/in/learn/information-system/> (дата звернення: 02.04.2023)
3. 10 кроків, щоб створити сайт з нуля. URL: <https://contextonline.ru/blog/10-shagov-chtoby-sozdat-sayt-s-nulya/> (дата звернення: 12.04.2023)
4. Архітектура веб-додатків та технології їх створення. URL: <https://studfile.net/preview/6196529/page:2/> (дата звернення: 27.04.2023)
5. Концепція: Web-архітектури шаблонів. Проектування та розробка корпоративних web додатків. URL: <https://sammeteo.ru/uk/konserciya-shablony-web-arhitektury-proektirovanie-i-razrabotka-korporativnyh-web/> (дата звернення: 01.05.2023)
6. Leaflet. URL: <https://leafletjs.com/> (дата звернення: 06.05.2023)
7. Bringing MySQL to the web. phpMyAdmin. URL: <https://www.phpmyadmin.net/> (дата звернення: 12.05.2023)