

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

«До захисту допущено»

Завідувач кафедри

_____ С. Г. Стіренко

«__» _____ 20__ р.

Дипломний проект

на здобуття ступеня бакалавра

з напрямку підготовки 6. 050102 – «Комп'ютерна інженерія»

**на тему: «Система електронного забезпечення навчально-методичної діяльності вищого
навчального закладу»**

Виконав:

Студент ІV курсу, групи ІО-64

Новосаєнко Владислав Іванович _____

Керівник:

доц. каф. ОТ

к.т.н., доц. Ткаченко В.В. _____

Консультант з нормоконтролю:

проф. каф. ОТ, д.т.н., проф.

Сімоненко В.П. _____

Рецензент: _____

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2020 року

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

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалаврський) Спеціальність –
123 «Комп'ютерна інженерія» Освітньо-професійна програма
«Комп'ютерні системи та мережі»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Сергій СТИПЕНКО

« ___ » _____ 2020 р.

ЗАВДАННЯ

на дипломний проект студенту

Новосаденку Владиславу Івановичу

1. Тема проекту *«Система електронного забезпечення навчально-методичної діяльності вищого навчального закладу»*, керівник проекту Ткаченко Валентина Василівна, старший викладач, затверджені наказом по університету від «07» травня 2020 р. № 1081-с
2. Термін подання студентом проекту 26 травня 2020р.
3. Вихідні дані до проекту див. технічне завдання
4. Зміст пояснювальної записки Аналіз і характеристика об'єкта проектування, обґрунтування оптимального варіанта реалізації мети цієї роботи, розробка додатку: вибір технологій та їх обґрунтування, основні рішення з реалізації додатку. Висновки
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) принципова схема, функціональна схема, структурна схема.
6. Консультанти розділів проекту.

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Сімоненко В. П., професор, д.т.н.		

7. Дата видачі завдання 01.09.2019

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Затвердження теми роботи	01.09.2019	
2.	Вивчення та аналіз завдання	15.12.2019-15.03.2020	
3.	Розробка архітектури додатку	15.03.2020-25.03.2020	
4.	Написання програмної частини	25.03.2020-05.04.2020	
5.	Тестування та виправлення помилок	05.04.2020-15.04.2020	
6.	Оформлення пояснювальної записки	15.04.2020-20.05.2020	
7.	Захист програмного продукту	25.04.2020	
8.	Передзахист	26.05.2020	
9.	Захист	16.06.2020	

Студент

Владислав Новосащенко

Керівник

Валентина ТКАЧЕНКО

АНОТАЦІЯ

В даній дипломній роботі було створено систему електронного забезпечення навчально-методичної діяльності вищого навчального закладу. У ході роботи були досліджені можливості платформи у порівнянні з аналогами та була проаналізована можливість оптимізації обробки користувацьких запитів в контексті платформи. У якості серверної частини була розроблена адміністративна панель за допомогою Symfony. А для написання фронтенд частини було обрано бібліотеку React.

АННОТАЦИЯ

В данной дипломной работе была создана система электронного обеспечения учебно-методической деятельности высшего учебного заведения. В ходе работы были исследованы возможности платформы по сравнению с аналогами и была проанализирована возможность оптимизации обработки пользовательских запросов в контексте платформы. В качестве серверной части была разработана административная панель с помощью Symfony. А для написания фронтенд части была выбрана библиотека React.

ABSTRACT

In this diploma was created a system of electronic support for educational and methodological activities of higher education. Implementing a project, the possibilities of the platform in comparison with analogues were investigated and the possibility of optimization of processing of user requests in the context of the platform was analyzed. An administrative panel was developed as a server part using Symfony. And the React library was chosen to write frontend part.

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4		Завдання на дипломний проект	2	
2	A4	6417.02.000 ТЗ	Технічне завдання	3	
3	A4	6417.03.000 ПЗ	Пояснювальна записка	62	
4	A4	6417.04.000 Д1	Структурна схема	1	
5	A4	6417.05.000 Д2	Функціональна схема	1	
6	A4	6417.06.000 ДЗ	Принципова схема	1	

					ДП.6417.01.000 ВП			
Зм.	Арк.	№ докум.	Підпис	Дата	<i>Система електронного забезпечення навчально-методичної діяльності вищого навчального закладу</i> Відомість проекту	Літ.	Аркуш	Аркушів
Розробив		Новосаденко В.І.					1	1
Перевірив		Ткаченко В.В.				НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ		
Реценз.						Група ІО-64		
Н. Контр.		Сімоненко В. П.						
Затвердив								

Технічне завдання
до дипломного проекту

На тему «Система електронного забезпечення навчально-методичної
діяльності вищого навчального закладу»

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ	2
2. ПІДСТАВИ ДЛЯ РОЗРОБКИ	2
3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ.....	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ	2
6. ЕТАПИ РОЗРОБКИ	3

					ДП.6417.01.000 ТЗ			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розробив	Новосаденко В.І.				<i>Система електронного забезпечення навчально- методичної діяльності вищого навчального закладу Технічне завдання</i>	Літ.	Аркуш	Аркушів
Перевірив	Ткаченко В.В.						1	1
Реценз.						НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ Група ІО-64		
Н. Контр.	Сімоненко В. П.							
Затвердив								

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Найменування: «Система електронного забезпечення навчально-методичної діяльності вищого навчального закладу» Область застосування: програма може використовуватися в вищих навчальних закладах.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання бакалаврського дипломного проекту, затверджене кафедрою обчислювальної техніки Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського».

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою розробки є створення системи управління підрозділами університету і навчальними планами.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки є науково-технічна література, публікації в спеціалізованих періодичних виданнях, довідники по платформах дистанційного навчання, публікації в мережі Інтернет по даній темі.

5. ТЕХНІЧНІ ВИМОГИ

Додаток, що розробляється повинен бути зручним у використанні. Повинен мати зрозумілий інтерфейс, із швидкою навігацією по системі. можливість створювати, редагувати та виконувати інші дії з навчальними планами. Потрібно, щоб була висока швидкість відгуку на запити користувачів.

					ДП.6417.02.000 ТЗ	Арк.
						2
Зм.	Арк.	№ докум.	Підпис	Дата		

6. ЕТАПИ РОЗРОБКИ

	Дата
Вивчення необхідної літератури	19.02.2020
Складання і узгодження технічного завдання	06.03.2020
Написання вступної частини та огляд рішень	19.03.2020
Розробка архітектури додатку	03.04.2020
Написання програмної частини	10.04.2020
Тестування та виправлення помилок	01.05.2020
Оформлення документації дипломного проекту	15.05.2020
Попередній захист та проходження нормативного контролю	29.05.2020
Захист дипломного проекту	18.06.2020

					ДП.6417.02.000 ТЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

Пояснювальна записка

до дипломного проекту

На тему «Система електронного забезпечення навчально-методичної діяльності вищого навчального закладу»

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1	4
ОГЛЯД РІШЕНЬ СИСТЕМ ЕЛЕКТРОННОГО ЗАБЕЗПЕЧЕННЯ НАВЧАЛЬНО-МЕТОДИЧНОЇ ДІЯЛЬНОСТІ ВНЗ	4
1.1. Актуальність систем електронного забезпечення	4
1.2. Електронна підтримка	5
1.3. Адміністрування в системах електронного забезпечення	6
1.4. Підтримка студентів	7
1.5. Безпека системи.....	8
1.6. Огляд існуючих систем електронного забезпечення	9
Висновки до розділу 1	13
РОЗДІЛ 2	14
ПРОЕКТУВАННЯ СИСТЕМИ ЕЛЕКТРОННОГО ЗАБЕЗПЕЧЕННЯ НАВЧАЛЬНО-МЕТОДИЧНОЇ ДІЯЛЬНОСТІ ВНЗ	14
2.1. Складнощі проектування і реалізації системи електронного забезпечення навчально-методичної діяльності ВНЗ	14
2.2. Вибір веб-фреймворку	16
2.2.1. Огляд існуючих фреймворків	16
2.2.1.1. Огляд веб-фреймворків мови програмування Python.....	18
2.2.1.2. Огляд веб-фреймворків мови програмування PHP	22
2.2.1.2. Огляд веб-фреймворків мови програмування Ruby.....	27
2.2.1.2. Огляд веб-фреймворків мови програмування C#.....	29
2.2.2. Критерії вибору фреймворку для розробки системи електронного забезпечення	31
2.2.3. Аналіз розглянутих рішень	33
Висновки до розділу 2	34
РОЗДІЛ 3	35
РЕАЛІЗАЦІЯ СИСТЕМИ ЕЛЕКТРОННОГО ЗАБЕЗПЕЧЕННЯ	35

					ДП.6417.03.000 ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата	<i>Система електронного забезпечення навчально- методичної діяльності вищого навчального закладу Пояснювальна записка</i>	Літ.	Аркуш	Аркушів
Розробив	Новосаденко В.І.						1	62
Перевірив	Ткаченко В.В.							
Реценз.								
Н. Контр.	Сімоненко В. П.					НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ Група ІО-64		
Затвердив								

3.1. Архітектура системи.....	35
3.2. Структура університету.....	36
3.3. Застосовані технології	39
3.3.1. Опис елементів системи.....	39
3.4. Опис програмної логіки роботи системи.....	40
Висновки до розділу 3	43
РОЗДІЛ 4	44
ТЕСТУВАННЯ РОБОТИ СИСТЕМИ.....	44
4.1. Опис налаштування і розгортки проекту	44
4.2. Опис інтерфейсу адміністративної панелі системи.....	45
4.2.1. Опис інтерфейсу структури університету.....	46
4.2.2. Опис інтерфейсу процесу взаємодії з навчальними планами.....	52
Висновки до розділу 4	58
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	61

ВСТУП

Освіта - це складний процес або система, яку люди використовують для досягнення усвідомлення себе та середовища, в якому вони існують - як локального, так і глобального; реальні та віртуальні. Таким чином, освіта є багатовимірним явищем, яке дозволяє людям здобувати необхідні навички, знання та розуміння для того, щоб впоратися з проблемами та проблемами повсякденного життя - будь то в побутовій, соціальній чи професійній обстановці. Вимоги, що ставляться до людей (і те, що їм потрібно знати і робити), зазвичай змінюються з часом - іноді дуже швидко. Таким чином, освітній системі, можливо, доведеться динамічно та швидко реагувати на зміни вимог. Реактивність освітньої системи часто сильно впливає на швидкість, з якою її цільове співтовариство може підтримувати її конкурентоспроможність у межах ринку вмінь та знань. У таких ситуаціях вкрай важливо, щоб відповідна (електронна) технологія була вбудована в основні системи, щоб забезпечити адекватний рівень роботи. Безсумнівно, один з найбільш популярних тенденцій в освіті стало використання програмного забезпечення для управління і адміністрування освітніх ресурсів організації.

І хоча чіткою місією університету є освіта, це також дуже велике підприємство, що тягне за собою величезну адміністративну діяльність. З точки зору менеджменту, університетський кампус - це досить велика і складна організація. Він повинен надавати багато різноманітних послуг великій кількості людей, які займаються величезною різноманітністю діяльностей. Служби повинні працювати безперебійно для задоволення своїх користувачів; вони повинні допомагати, а не порушувати встановлену діяльність користувача.

					ДП.6417.03.000 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1

ОГЛЯД РІШЕНЬ СИСТЕМ ЕЛЕКТРОННОГО ЗАБЕЗПЕЧЕННЯ НАВЧАЛЬНО-МЕТОДИЧНОЇ ДІЯЛЬНОСТІ ВНЗ

1.1. Актуальність систем електронного забезпечення

Для покращення рівня та якості освіти велике значення має проблема технологічного забезпечення методичної роботи. Зараз при забезпеченні навчально-методичної діяльності вищого навчального закладу значно важливим є реалізація саме електронного варіанту системи, що може забезпечити автоматизацію багатьох процесів взаємодії з даними; зберігання і оновлення інформації про студентів та працівників ВНЗ; розробка навчальних планів та управління ними. Особливості системи - це інструменти для підтримки академічної, технічної, адміністративної та професійної співпраці. Продуктивність роботи викладачів та студентів при очному або дистанційному навчанні залежить від рівня та різних форм підтримки інформаційно-комунікаційних технологій. Електронне забезпечення навчально-методичної діяльності ВНЗ відіграє важливу центральну роль для пришвидшення і спрощення процесів і роботи з управління інформацією. Управління всіма вищевикладеними процесами може бути досить трудомістким без централізованої електронної системи, так як доводиться вручну обробляти величезну кількість інформації. Також система надає високий рівень наочності, що так само збільшує швидкість і якість роботи з усіма необхідними даними і покращує сприйняття за допомогою зручного інтерфейсу. Тому на поточний момент для ВНЗ ключовим і актуальним завданням є розробка такого технологічного інформаційно-комунікаційного засобу, що втілить у собі реалізацію вирішення наведених проблем для всіх учасників освітнього процесу: викладачів, студентів, аспірантів та ін.

					ДП.6417.03.000 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2. Електронна підтримка

Більшість університетів надають якусь комп'ютерну підтримку, чи то вона забезпечує програми та навчання для певних додатків, чи обслуговує та підтримує існуючі системи. Однак більшість із цих систем забезпечують незначну частину підтримки в неробочий час. Зазвичай у них є довідкова служба або офіс, який надає ці послуги протягом встановленого часу, як правило, в робочий час. Університети дистанційного викладання мають подібні обмежувальні практики та можуть надавати телефонну службу підтримки лише трохи збільшених робочих годин, незважаючи на те, що це не часи, коли студенти часто навчаються (вечорами та вихідними). Ми зрозуміли, що це обмежує студентів і може вимагати від них чекати до наступного дня, щоб звернутися за допомогою. Досліди з використання електронної підтримки, а також телефонної підтримки показали, що використання обох підходів у тандемі може допомогти при різних видах проблем, а також запропонувати тип підтримки в неробочий час в електронному вигляді. Це особливо важливо для студентів, які навчаються в різних часових поясах, коли звернення до університету по телефону є дорогим та тимчасовим обмеженням. Одним із результатів цього досвіду є те, що загальні проблеми можуть бути вирішені електронною поштою, щоб попередити їх про вирішення певної проблеми або розмістити поширені запитання у веб-сховищі. Електронна пошта також спонукає студентів задуматися про те, в чому полягає їхня проблема, та детально розкрити її рішення. Частина цього процесу спонукає студентів проаналізувати проблему та повідомити її у письмовій формі - процес, який зазвичай не виконується. Це особливо корисно для студентів, навчання яких пов'язано з комп'ютерними науками, для заохочення їх навичок вирішення проблем.

Хоча зрозуміло, що нам потрібно покращити рівень комп'ютерної підтримки в неробочий час, все ж можливо надати частину цієї послуги в електронному вигляді. Насправді, для проблем, які не потребують допомоги

					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

для вирішення у вигляді зв'язку через Інтернет, більшість студентів віддають перевагу саме такому способу. Відповідь, яку вони отримують, зберігається, і до неї можна звертатись коли завгодно. Для установи електронний аспект цього типу підтримки забезпечує механізм аудиту, а також багаторазові матеріали, що складають основу сховища проблем та рішень. Надання комп'ютерної підтримки через Інтернет, безумовно, можливо і корисно в таких умовах. Однак існує потреба у наданні телефонної підтримки для проблем, характерних для Інтернет-комунікацій, наприклад під час вимкненого сервера і т. д. Тим не менш, заклад набуває корисної особливості у вигляді електронного забезпечення діяльності університету.

Як видно з цих прикладів, надавати такі види послуг підтримки студентів за допомогою електронного середовища не тільки можливо, але й вигідно. Ці заходи є загальними для всіх установ і можуть бути корисними для надання в електронному вигляді незалежно від конкретної реалізації чи суб'єктів, які беруть участь у будь-якому з видів діяльності. Сьогодні ми спостерігаємо досить детальний підхід, коли установи здійснюють лише частину процесу в електронному вигляді. Спостерігалось, що коли частина процесу стає доступною в електронному вигляді, студенти та викладачі починають запитувати, чому весь процес не передбачений аналогічно.

1.3. Адміністрування в системах електронного забезпечення

Адміністрування є дуже важливою функцією в будь-якому навчальному закладі, забезпечуючи, вхідні двері, через які студенти вперше контактують із закладом. Це обличчя, яке надає студентам початкову інформацію, яка їм потрібна, наприклад, книги, реквізити курсу, вартість вступу, вимоги до вступу, реєстрація та зарахування на курс. На пізніх етапах він розглядає реєстрацію іспитів, посвідчення на іспити та загальну кореспонденцію щодо правил та правил університету. Також потрібно розібратися з діями з особистих проблем, які можуть вплинути на студентів, які закінчують курс, і

					ДП.6417.03.000 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

він надає рішення щодо цих справ. У деяких університетах цією діяльністю займається одна центральна служба, і дуже конкретні запити щодо курсів передаються до навчальної одиниці, яка веде курс. Незалежно від впровадження, на початку та в кінці курсу часто існує висока концентрація активності, яка зазвичай супроводжується об'ємною кількістю паперової документації.

Очевидно, що існує потреба в обміні даними між центральною базою даних університету та місцевими системами викладачів. Основне правило полягає у тому, що у всій університетській системі має бути лише одне джерело інформації. Наприклад, ідентифікаційний номер студента повинен бути виданий у центральній системі та скопійований у локальні системи. З іншого боку, бали, що отримують студенти за результатами навчання, збираються в підрозділах факультету, а до центральної бази даних надсилаються вже сукупні позначки.

Чітко визначені протоколи та процедури обміну даними між підрозділами та центральною базою даних є важливими для безперебійної та ефективної роботи всієї системи управління інформацією університету. Дані повинні обмінюватися в електронному вигляді, копіювання інформації вручну з однієї системи в іншу очевидно застаріло.

1.4. Підтримка студентів

Підтримка студентів охоплює низку заходів, що надаються університетом, і хоча реалізація цих процесів та суб'єктів, що здійснюють ці процеси, можуть відрізнятися від закладу до установи, основні заходи залишаються однаковими. По суті, реалізується спроба забезпечення адміністративних процесів, консультування та комп'ютерної підтримки (особливо в навчанні CS, computer science) для студентів.

Основою послуг електронного забезпечення є система баз даних або мережа локальних баз даних, безперебійно пов'язаних між собою. Більшість

					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

збереженої інформації включає індивідуальні дані користувача, такі як особиста інформація або знаки, здобуті студентами. Така конфіденційна індивідуальна інформація не повинна розкриватися стороннім особам. Таким чином, користувачі передають інформацію в базу даних і читають її з бази даних через захищені канали зв'язку. Щоб зробити канал захищеним, користувача потрібно ідентифікувати за допомогою унікального паролю, який використовується, наприклад, для шифрування інформації та/або приєднання електронних підписів до фрагментів даних, завантаження даних для перевірки.

1.5. Безпека системи

В останні роки, при частому виникненні інцидентів із безпекою, підприємства та організації тепер усвідомили важливість безпеки інформаційної системи. Сьогодні інформаційні системи значною мірою покладаються на технології веб та баз даних, тому ризики та загрози, з якими стикаються ці технології, також вплинуть на безпеку інформаційних систем. Технології безпеки веб та баз даних можуть забезпечити конфіденційність, цілісність та зручність використання даних в інформаційній системі.

Існує маса факторів, які входять у безпеку та захист веб-сторінок. Будь-який захищений веб-сайт або додаток, безумовно, підтримується різними видами контрольних методів для його безпеки. Існує цілий ряд стандартів безпеки, яких необхідно дотримуватися постійно, і ці стандарти впроваджуються та висвітлюються OWASP (Відкритий проект з безпеки веб-застосунків). Основними кроками щодо захисту веб-додатків від атак є застосування сучасного шифрування, встановлення належної автентифікації, постійне виправлення виявлених уразливостей, уникнення крадіжки даних.

Засоби захисту даних, вбудовані в систему управління інформацією, включають:

- обмежений доступ до даних (потрібна авторизація користувача; декілька типів користувачів, визначених обмеженими правами доступу),

					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

- захист вмісту даних від несанкціонованого читання (шифрування),
- заходи перевірки цілісності та достовірності даних.

Засоби захисту системи дозволяють реалізувати різні механізми захисту доступу до збережених даних. Право на виконання будь-якої операції над вмістом файлів - читання, запис, модифікація - може бути обмежено.

1.6. Огляд існуючих систем електронного забезпечення

Усе більше університетів у всьому світі переходять до систем електронного забезпечення навчально-методичної діяльності своїх закладів, що допомагають в адмініструванні, підтримці студентів, управлінні навчальними та науковими статтями, дисертаційними та дипломними роботами тощо. Приклади реалізацій таких систем:

1. У Технічному університеті в Лодзі (Польща) виникла ідея реалізації і застосування системи електронного забезпечення у вигляді електронних карток для зберігання студентських відміток та допомоги в управлінні навчально-методичною діяльністю. Необхідність впровадження електронної системи стала добре визнана. Чотириразове збільшення кількості студентів за кілька років спричинило багато організаційних проблем академічному та адміністративному персоналу. Ці проблеми загрожували якості надання послуг студентам та потребували швидкого вирішення. Спочатку студент повинен був отримати пластикову картку зі своєю фотографією для візуальної ідентифікації та цифровою пам'яттю, достатньою для зберігання всіх даних протягом усього навчання. Картки, оснащені мікропроцесором та операційною системою, називаються «розумними картами» і зчитуються за допомогою кардридерів. На відміну від карт магнітної смуги, смарт-карти несуть всю необхідну інформацію та функції обробки. Тому вони не потребують доступу до віддалених баз даних на момент здійснення транзакції. Вартість однієї неперсоніфікованої

					ДП.6417.03.000 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

картки може становити від 0,2 євроценти для магнітної смуги, від 2,50 євро для картки з пам'яттю і до 10 євро для смарт-карти. Смарт-карта не є рішенням від усіх проблем із автентифікацією користувачів. Оскільки карту можна вкрасти разом з PIN-кодом до неї, то в додатках підвищеної безпеки є потреба в інших, додаткових методах ідентифікації людей. Більшість цих методів засновані на вимірюванні особистих характеристик тіла або поведінки користувача, таких як відбитки пальців, структури судин сітківки або голосові параметри. Ці види біометричної інформації можуть бути використані для покращення (збільшення) або заміни інформації, що зберігається на картці. Розробка приладів біометричної автентифікації/ідентифікації, інтегрованих із системами карток електронного забезпечення, вимагає нових датчиків для надійного вимірювання особистої інформації та нових методів обробки сигналів, аналізу зображення та розпізнавання шаблонів. Основним інструментом захисту інформації є мікропроцесорна карта. Її внутрішні джерела використовуються для зберігання:

- ідентифікаційних даних користувача,
- криптографічних ключів,
- електронних файлів гаранцій,
- інформації про контроль доступу.

Обмеження доступу можуть застосовуватися до створення нових елементарних файлів та папок. Операційна система картки включає два тести, які необхідно пройти для отримання прав на виконання необхідних операцій. Один покладається на перевірку паролів, що зберігаються в елементарних файлах секретного коду. Інша заснована на верифікації криптографічного ключа за допомогою методу відповідь-виклику. В останньому випадку мікропроцесор картки генерує псевдовипадковий номер і надсилає його на комп'ютерну програму, яка вимагає створення нової папки. Додаток шифрує отримане число за

					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

допомогою алгоритмів 3DES на основі асиметричного ключа, що надається спільною карткою, і надсилає зашифроване повідомлення на картку. Картка порівнює отримане повідомлення з тим, яке генерується локально його мікропроцесором. Якщо повідомлення однакові, картка виконує запитувану операцію; в іншому випадку в програму надсилається повідомлення про помилку.

Очевидно, що існує потреба в обміні даними з загальною базою даних університету, тому існують засоби синхронізації карток і баз даних – центральної і дочірніх підрозділів.

З огляду даної системи видно головні її недоліки: впровадження спеціальних карток є невиправданим ускладненням, адже за допомогою сучасних технологій існує можливість створення веб та мобільних додатків, що будуть підтримувати всі вищенаведені функції та можливості карток. Також реалізація електронної системи у вигляді додатку набагато спростить процес розробки, забезпечення безпеки та синхронізації даних тощо. А так як у сучасному світі у всіх є доступ до електронних ресурсів за допомогою різних засобів, тому і в наданні спеціальних карток немає потреби і їх придбання буде марною тратою коштів університету.

2. Система Moodle - надзвичайно налаштована система, що має багато стандартних функцій, яка дозволяє створити власний приватний веб-додаток, наповнений динамічними ресурсами, що загалом являють собою електронну систему управління навчанням. Система розроблена таким чином, щоб бути гнучкою і відкритою, а завдяки зручному інтерфейсу навігація по платформі інтуїтивно зрозуміла на різних видах пристроїв з різними розширеннями екранів. Персоналізована інформаційна панель Moodle відображає поточну інформацію про навчання, адміністративну панель та ін. Також присутні інструменти та засоби для спільної роботи; зручне управління файлами, що дозволяє

					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

завантажувати файли з хмарних служб зберігання даних, включаючи Google Drive, MS OneDrive, Dropbox тощо; та багато інших функцій для онлайн навчання.

Moodle має систему сторонніх плагінів (додаткових бібліотек), що створюються спільнотою. Прикладом такого плагіну є Моніторинг навчальних планів. Основна мета цього плагіна - полегшити роботу керівників навчальних планів. Він надає огляд плану навчання користувачів, не залишаючи сторінки, щоб отримати інформацію, пов'язану з цим планом (наприклад, оцінювання, свідоцтво користувача). Він також пропонує статистику за навчальними планами. Для плагіну з дуже великою кількістю навчальних планів, він є найкращим рішенням, з його вдосконаленим фільтром, що дозволяє фільтрувати плани навчання за кількома критеріями. Також існують і інші подібні плагіни, що пропонують бази даних, плани для всіх навчальних заходів в організації для структурованої програми навчання, включаючи звіти керівництва тощо. Moodle разом із системою плагінів створюють серйозну розширювану платформу, що дозволяє покращити різні електронні особливості, такі як адміністрування, аналітика і статистика, перевірка на плагіат, бази даних та сховища.

Основними недоліками Moodle з боку системи електронного забезпечення навчально-методичної діяльності – ця система більше створена для онлайн навчання, а все інше досягається за допомогою сторонніх розширень. Але потрібних плагінів може не бути або вони не сильно налаштовуванні, тому доведеться чекати поки розробники не нададуть потрібний функціонал.

					ДП.6417.03.000 ПЗ	Арк.
						12
Зм.	Арк.	№ докум.	Підпис	Дата		

Висновки до розділу 1

Розвиток інформаційних технологій сьогодні відіграє значну роль у житті людей та організацій, у тому числі освітніх установ, таких як школи від початкового до вищого рівня. Система електронного забезпечення у вищих навчальних закладах стала абсолютною необхідністю, якою необхідно володіти та використовувати для покращення якості процесу управління та адміністрування. Призначення такої системи полягає в тому, що будь-яке систематичне зусилля щодо зміни моделі фізичної активності стає справді цифровою або віртуальною подією, яка дозволяє користувачам (зацікавленим сторонам) взаємодіяти та здійснювати транзакції в електронному вигляді. Можна відчувати, що до бібліотеки можна звертатися вночі безпосередньо з дому, а також замінити більшість фізичної адміністративної діяльності на електронну.

Після огляду існуючих систем виявлено їх основні переваги та недоліки, що можна використати під час розробки системи електронного забезпечення. Головними проблемами розглянутих системи є надлишковість функцій та можливостей і відсутність інших - потрібних. Також виявлено недоліки у проектуванні систем.

У цій роботі представлена інформація про архітектуру, функції та питання впровадження системи електронного забезпечення навчально-методичної діяльності ВНЗ. Система є прикладом важливого, широко розповсюдженого та все ще зростаючого застосування захищених мережевих баз даних, призначених для підтримки управління дуже складним підприємством - університетом.

					ДП.6417.03.000 ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2

ПРОЕКТУВАННЯ СИСТЕМИ ЕЛЕКТРОННОГО ЗАБЕЗПЕЧЕННЯ НАВЧАЛЬНО-МЕТОДИЧНОЇ ДІЯЛЬНОСТІ ВНЗ

2.1. Складнощі проектування і реалізації системи електронного забезпечення навчально-методичної діяльності ВНЗ

Проектування і розробка електронних систем прискорюється агресивно. Гнучкіші і зручніші інтерфейси користуються попитом. Що стосується розробки хорошої системи, то існує ряд факторів, що визначають цей успіх.

Основні проблеми при розробці електронних систем:

- інтерфейс користувача і досвід користувача (UI – user interface, UX – user experience),
- масштабованість,
- продуктивність,
- вибір правильних технологій і рівень їх знань,
- безпека.

Зараз очікується найбільш простіша і орієнтована на користувачів система. Іноді значний вплив роблять маленькі елементи інтерфейсу користувача. Ще одна частина, якою розробники часто нехтують - навігація. Інтуїтивна навігація створює кращий рівень UX для відвідувача системи і спрямовує аудиторію до інформації, яку вони шукають.

Масштабованість - врівноваження навантаження між серверами, отже, коли навантаження збільшується (тобто більше трафіку на сторінці), для його збалансування можуть бути додані додаткові сервери. Потрібно спроектувати програмне забезпечення таким чином, щоб воно могло працювати на кластері серверів. Сервісно-орієнтована архітектура (SOA) може допомогти покращити масштабованість, коли додається все більше серверів. SOA є дуже гнучкою і дає можливість легко змінюватись. Сервісно-орієнтована архітектура - це

					ДП.6417.03.000 ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

дизайн, де компоненти програми надають послуги іншим компонентам через протокол зв'язку, в основному по мережі.

Як правило, прийнято, що продуктивність системи має передове значення для успішної системи. Потрібно думати про ефективність, перш ніж розробляти веб-додаток. Деякі проблеми продуктивності - це погано написаний код, неоптимізовані бази даних, некерований ріст даних, стрибки трафіку, поганий розподіл навантаження, конфігурація за замовчуванням, клопітні послуги сторонніх служб тощо.

Фреймворки є основним засобом для прискорення розробки: вони підвищують продуктивність, пропонують бібліотеки та розширюють можливості, тому розробникам не потрібно розробляти з нуля. Фреймворки пропонують такі функції, як моделі, API, фрагменти коду та інші елементи для розробки динамічних платформ. Деякі фреймворки мають жорсткий підхід до розробки, а деякі є гнучкими, даючи більше можливостей для конфігурації. Найпоширенішими прикладами веб-фреймворків є PHP (Symfony, Laravel тощо), ASP.Net, Ruby on Rails, Python Django. Нова додаткова функціональність може бути додана через зовнішній API за допомогою можливостей, пропонуємих фреймворками.

Безпекою часто нехтують, але за безпекою потрібно слідкувати протягом усього життєвого циклу розробки програмного забезпечення, особливо коли програма має справу з важливою інформацією, такою як контактна інформація та конфіденційні дані. Існує багато речей, які слід враховувати, коли мова йде про безпеку веб-додатків, такі як відмова в сервісних атаках, безпека даних користувачів, несправності в базі даних, несанкціонований доступ до обмежених частин веб-сайту тощо. Систему слід ретельно проектувати і розробляти, щоб бути безпечним від цих проблем безпеки. Вибір фреймворку є також важливим і з точки зору безпеки, адже багато з них пропонують реалізацію необхідного функціоналу з вирішення деяких розповсюджених вразливостей.

					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

2.2. Вибір веб-фреймворку

Серверні веб-фреймворки - це програмні фреймворки, які спрощують розробку, підтримку та масштабування системи. Вони надають інструменти та бібліотеки, які спрощують вирішення поширених завдань веб-розробки, включаючи маршрутизацію URL-адрес до відповідних обробників, взаємодію з базами даних, підтримку сесій та авторизацію користувачів, форматування вихідних даних (наприклад, HTML, JSON, XML) та покращення безпеки від веб-атак.

2.2.1. Огляд існуючих фреймворків

Функції, які часто надаються веб-фреймворками (але не кожний фреймворк буде обов'язково їх надавати):

- робота безпосередньо з HTTP-запитами та HTTP-відповідями,
- направлення запитів до відповідного обробника,
- спрощення доступ до даних у запиті,
- абстрагування та спрощення доступу до бази даних,
- система шаблонів для відображення даних.

Веб-сервери та браузері спілкуються через протокол HTTP - сервери чекають HTTP-запитів від браузера, а потім повертають інформацію у відповідях HTTP. Веб-фреймворки дозволяють писати спрощений синтаксис, який буде генерувати код на стороні сервера для роботи з цими запитами та відповідями. Це значно полегшує взаємодію, додаючи абстракції для роботи з шарами вищого рівня, а не мережевими примітивами нижчого рівня.

Більшість сайтів надають ряд різних ресурсів, доступних через різні URL-адреси. Працювати з усіма ними в одній функції буде важко, тому веб-фреймворки забезпечують прості механізми для відображення шаблонів URL-адрес на конкретні функції-обробники. Цей підхід також має переваги щодо технічного обслуговування, оскільки можна змінити URL-адресу, не змінюючи базовий код певної функції. Різні фреймворки використовують різні механізми для відображення, такі як реєстрації URL-адреси через спеціальні

					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

маршрутизатори; використання декораторів; використання строк документації функції-обробника; використання списку/таблиці відображень між шаблонами адрес та функціями.

Дані можуть бути закодовані в HTTP-запиті різними способами. HTTP GET-запит на отримання файлів або даних з сервера може кодувати дані, необхідні в параметрах URL-адреси або в структурі URL-адреси. Запит HTTP POST на оновлення ресурсу на сервері замість цього буде містити інформацію про оновлення як «дані POST» в тілі запиту. Запит HTTP також може містити інформацію про поточний сеанс або користувача в файлі cookie на стороні клієнта. Веб-фреймворки забезпечують відповідні для мови програмування механізми доступу до цієї інформації.

Платформи використовують бази даних для зберігання інформації як для обміну з користувачами, так і про користувачів. Веб-фреймворки часто надають шар бази даних, який абстрагує операції з читання, запису, запиту та видалення бази даних. Цей шар абстракції називається об'єктно-реляційним відображенням (ORM – Object-Relational Mapping). Використання ORM має дві переваги:

- можна замінити базу даних, не обов'язково змінюючи код, який її використовує. Це дозволяє розробникам оптимізувати характеристики різних баз даних на основі їх використання,
- базова перевірка даних може бути реалізована в фреймворках. Це полегшує та робить безпечнішим перевірку, наприклад чи дані, що зберігаються, правильного типу, має правильний формат (наприклад, електронна адреса) та не є шкідливими.

З використанням ORM створюються моделі бази даних, з визначенням відповідних полів. Визначення поля визначає максимальний розмір, значення за замовчуванням, параметри списку вибору, текст довідки для документації, текст мітки для форм тощо.

					ДП.6417.03.000 ПЗ	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

2.2.1.1. Огляд веб-фреймворків мови програмування Python

1. Django - це веб-фреймворк Python високого рівня, яка заохочує швидкий розвиток та чистий, прагматичний дизайн. Побудований досвідченими розробниками, він піклується про великі труднощі веб-розробки, тому можна зосередитись на написанні програми, не потребуючи винаходити колесо. Django безкоштовний та має відкритий код. Оскільки все включено за замовчуванням, це все працює разом, дотримується послідовних принципів дизайну та має велику та сучасну документацію. Він також швидкий, надійний і дуже масштабований. Базуючись на Python, код Django легко читати та підтримувати.

Django має власну систему імен для всіх функцій та компонентів. Він також має панель адміністратора, з якою вважається, що працювати легше, ніж у Laravel, та інші технічні характеристики, зокрема:

- простий синтаксис,
- свій власний веб-сервер (для розробки і тестування),
- основна архітектура - MVC (Model-View-Controller),
- поставляється з усім необхідним для вирішення загальних випадків,
- ORM,
- HTTP-бібліотеки,
- підтримка функцій проміжної обробки (middleware).

Крім того, Django забезпечує динамічний інтерфейс CRUD (створення, читання, оновлення та видалення), налаштований з моделями адміністратора та генерується за допомогою самоаналізу. CRUD використовується для опису основних команд бази даних, що означає, що інтерфейс полегшує перегляд, зміну та пошук інформації.

Однією з головних цілей Django є спрощення роботи для розробників. Це забезпечує DRY (Don't Repeat Yourself) філософія - це означає, що розробники можуть повторно використовувати існуючий код і зосередитись на написанні унікального.

					ДП.6417.03.000 ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

Безпека також є головним пріоритетом для Django. У ньому є одна з найкращих вбудованих систем безпеки, і це допомагає розробникам уникати поширених проблем із безпекою, зокрема міжсайтовий скриптинг (cross-site scripting), SQL ін'єкції тощо. Django оперативно випускає нові виправлення безпеки. Розробники фреймворку зазвичай перші, хто реагує на вразливості та попереджає інші.

За допомогою Django можна вирішувати проекти будь-якого розміру та обсягу, чи то простий веб-сайт або веб-додаток із великим навантаженням. Він повністю завантажений додатковими пакетами та масштабованими, так що можна робити додатки, які обробляють великий трафік та великий обсяг інформації. Django кросплатформний, що означає, що проект може базуватися на Mac, Linux або ПК. Він працює з більшістю основних баз даних і дозволяє використовувати базу даних, яка більше підходить для конкретного проекту, або навіть декілька баз даних одночасно. У Django є велика спільнота, яка підтримує доступ, через численні форуми, канали та спеціальні веб-сайти. Легко знайти допомогу, коли в коді є проблемна функція, і знайти розробників. Фреймворк має одну з найкращих документацій, і вона підтримується на високому рівні, оновлюється разом з новими функціями та виправленнями, щоб можна було легко адаптуватися до змін.

Незважаючи на те, що Джанго є дивовижною основою, є кілька мінусів, які можуть бути проблемою. По-перше, вказівка URL-адреси за допомогою регулярних виразів не є легким завданням, принаймні для початківців. Крім того, він відчуває себе трохи роздутим для малих проектів, і деякі люди вважають його досить насиченим великими проектами, оскільки, наприклад, моделі, включені в один файл. Можна витратити багато часу, намагаючись з'ясувати, що з програмою не так, або ще гірше, можна навіть не знати, що у вашій програмі є проблема через помилки, які виникають у шаблонах.

Недоліки Django:

- швидкість,

					ДП.6417.03.000 ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

- не завжди правильний вибір,
- Django є монолітним.

Погано розроблена архітектура в поєднанні з Python, яка не є найшвидшою мовою навколо, може призвести до повільних веб-сайтів. Тому потрібно переконатися, що систему оптимізовано належним чином. Django пропонує власні орієнтири, щоб перевірити швидкість роботи внутрішніх одиниць та виявити всі вузькі місця. Можна застосувати кешування та купу різних оптимізацій. Забезпечення оптимізованої та масштабованої архітектури з самого початку має вирішити багато проблем зі швидкістю в майбутньому. Проблеми зі швидкістю не є проблемою самого Django, а питанням правильної конфігурації та архітектурного дизайну.

Django може бути неправильним вибором для дійсно невеликих сайтів, таких як статичні односторінкові проекти або мікросервіси - саме в цьому випадку можна скористатися іншими фреймворками/мікрофреймворками (FastAPI, Flask, Laravel, Symfony).

Фреймворк Django має певний спосіб визначення та виконання завдань, логічну структуру файлів, яку легко вивчити. Але це також виникає умова, що ви не можна використовувати власну структуру файлів. Якщо не дотримуватись правил побудови структури, буде складно проект розгорнути побудований за допомогою Django, адже фреймворк і сервер шукає інформацію в цих файлах.

2. Flask – Python мікрофреймворк. Хоча він мінімалістичний, за допомогою Flask можна створювати серйозні проекти. Він містить сервер розробки та відладчик і включає підтримку шаблону Jinja2, захищені файли cookie, тестування блоків та відправлення RESTful запиту. Має гарну документацію та активну спільноту. Flask став надзвичайно популярним, особливо для розробників, яким потрібно надавати веб-сервіси в невеликих системах, обмежених ресурсами. Flask надає набагато більше контролю на етапі розробки проекту. Він дотримується принципів мінімалізму, і дає

					ДП.6417.03.000 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

свободу вирішення, як будувати програму. Переваги використання Flask фреймворку:

- Flask має легку та модульну конструкцію, тому її легко перетворити на потрібну або змінити під інший веб-фреймворк за допомогою декількох розширень без сильних проблем,
- дає можливість підключити улюблену ORM бібліотеку, наприклад SQLAlchemy, mongoengine тощо,
- документація є вичерпною, наповнена прикладами та добре структурована,
- розгорнути Flask систему дуже просто,
- конфігурація навіть гнучкіша, ніж у Django, що дає багато рішень для кожної потреби в розробці.

Flask - це один з найбільш відполірованих та найбагатших на функції мікрофреймворк. Ще молодий, Flask має процвітаючу спільноту, розширення першого класу та елегантний API. Flask має переваги швидких шаблонів, сильні функції WSGI, ретельну перевірку одиниць на рівні веб-додатків та бібліотеки, обширну документацію.

Недоліки Flask:

- не підходить для великих додатків,
- спільнота не настільки велика,
- немає адміністративної системи за замовчуванням,
- немає вбудованої системи розробки аутентифікації,
- складне управління міграціями.

При розробці великого проекту, використання Flask може зайняти багато часу. Він підходить лише для невеликих масштабів застосування.

Спільнота Flask не така велика, як Django. тож знайти проблему важко, оскільки не буде багато активних користувачів, які б відповіли на питання.

На сайті адміністрування можна побачити всі моделі, вставити нові записи, оновити та видалити старі записи. Тоді як у Flask цього зробити не

					ДП.6417.03.000 ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

можна, адже немає сайту адміністратора. Але ця проблема вирішується встановленням сторонніх бібліотек.

2.2.1.2. Огляд веб-фреймворків мови програмування PHP

1. Symfony – це фреймворк PHP, який спрямований на прискорення створення та обслуговування систем. Для його встановлення необхідні декілька передумов: Linux, FreeBSD, Mac OS або Microsoft Windows та веб-сервер з встановленим PHP. Symfony сумісний майже з усіма RDBMS (система управління реляційними базами даних) і має низькі накладні витрати. Фреймворк простий у використанні завдяки чистому дизайну та читабельності коду. Symfony пропонує помічники, плагіни Ajax та інтерфейси адміністратора, що робить програмування великих систем справді простими. Розробники можуть зосередитись на логіці розробки, не витрачаючи часу на запис нескінченних файлів конфігурації XML. Symfony можна використовувати для побудови надійних додатків у корпоративному контексті, оскільки це допомагає розробникам тестувати, налагоджувати та документувати проекти, надаючи їм повний контроль над конфігурацією та налаштуванням - від структури каталогу до іноземних бібліотек. Symfony використовує модель дизайну Model-View-Controller, яка відокремлює бізнесу-логіку від презентаційного шару.

Переваги використання Symfony:

- високий рівень гнучкості,
- компанія, що стоїть за технологією,
- доведена надійність,
- легке тестування,
- довгострокова підтримка,
- велика спільнота розробників,
- хороша документація.

					ДП.6417.03.000 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

Symfony - один з найбільш насичених функціями фреймворк PHP. Дві найвидатніші технологічні переваги Symfony, це пакети (bundles) та компоненти (components). Пакет майже те саме, що і плагін. Можна розглядати їх як пакет файлів (файли PHP, таблиці стилів, JavaScript-файли, зображення) для реалізації будь-якої функції (наприклад, блог, кошик тощо). Основна перевага пакетів полягає в тому, що вони відокремлюються. Можна переналаштувати та повторно використовувати їх для багатьох проектів, щоб зменшити загальну вартість розробки. Компоненти - це загальні функції, що зменшують звичайні завдання та дозволяють розробникам зосередитися на конкретних особливостях розробки. Є 30 корисних компонентів Symfony, які полегшують процес розробки. Можна використовувати компоненти незалежно та додавати власні модулі без жодних архітектурних проблем. Компоненти Symfony можна також використовувати окремо в інших структурах (наприклад, Laravel) або в звичайних PHP-рішеннях. Пакети та компоненти допомагають усунути суворі залежності в архітектурі. Чим менше у залежностей, тим простіше буде вносити зміни, не ризикуючи порушити інші частини системи. Таким чином, можна адаптувати рішення під будь-які вимоги та сценарії користувачів, щоб створити надзвичайно гнучку програму.

Symfony - один з небагатьох фреймворків, комерційно підтримуваних. SensioLabs, компанія-творець та спонсор, активно сприяє його репутації. Вони надають офіційні підручники та сертифікацію, що показує масштаб та серйозність їхніх намірів та переконань. Symfony з часом довів свою надійність, тоді як багато інших фреймворків не вдалися. Багато відомих платформ, використовують його компоненти.

Кожний рядок коду, написаний користувачем фреймворку, повинен бути протестований, щоб гарантувати стабільну роботу програми. Повторне використання пакетів, відсутність суворих залежностей та можливість створення моделей дизайну сприяють кращому обслуговуванню та тестуванню в Symfony. Тестування одиниць дуже легко завдяки використанню

					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

незалежної бібліотеки PHPUnit. Функціональне тестування також автоматизоване для зменшення витрат часу розробника.

Symfony - це стабільна і добре перевірена структура з регулярними оновленнями. Найновіші версії залишаються довгостроково підтримуваними та сумісні з новими випусками: для деяких версій - до 3 років.

Одним із впливових аспектів для життєздатності системи є хороша спільнота. Не лише основні розробники працюють над вдосконаленнями. Symfony є відкритим кодом. Це означає, що фахівці PHP та ентузіасти з усього світу беруть участь у вдосконаленні коду. У спільноті люди співпрацюють один з одним. Вони створюють нові компоненти, намагаються вирішити виниклі проблеми або допомагають іншим порадами.

Неповна або застаріла документація є проблемою багатьох технологій. Документація Symfony вважається однією з найкращих, порівняно з документацією інших фреймворків PHP і не тільки. Вона чітко написана, добре структурований, постачається із прикладами та оновлюється від версії до версії. Можна знайти пояснення кожного компонента та процесу розробки в цілому.

Недоліки Symfony:

- продуктивність,
- швидкість запуску,
- не дуже простий в освоєнні.

Основним недоліком, на який часто скаржаться, є продуктивність Symfony. Продуктивність важлива для додатків, які працюють у режимі реального часу з високим навантаженням (коли багато користувачів взаємодіють із додатком у той самий момент, як у соціальних мережах). Але Symfony росте, накопичуючи найкращі речі з інших технологій. Одним із способів підвищення продуктивності є перенесення програми на найновішу версію PHP, що може зробити систему майже вдвічі швидшою.

					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

Загальна практика Symfony - використовувати шаблони проектування, тобто застосовувати один раз створений код для інших частин системи. Це полегшує тестування та подальшу підтримку, але може знизити швидкість розробки та затримати дату запуску.

Для навчання Symfony потрібно набагато більше зусиль, ніж, наприклад, Laravel, які зрозуміліші інтуїтивно. Але це один з найбільш стабільних фреймворків, який дозволяє приймати складні рішення з багатьма змінами вперед.

2. Laravel - це веб-фреймворк на PHP для створення веб-додатків високого класу, використовуючи його значні та витончені синтаксиси. Він постачається з надійною колекцією інструментів і пропонує архітектуру проектів. Більше того, він включає різні характеристики таких технологій, як ASP.NET MVC, CodeIgniter, Ruby on Rails та багатьох інших. Laravel – фреймворк з відкритим кодом. Це полегшує роботу розробників, заощаджуючи величезний час і допомагає зменшити розмірковування та планування розробки всієї системи з нуля. Поряд із цим, Laravel також дбає про безпеку програми. Отже, всі його функції можуть примножити темп розробки веб-сторінок для вас.

Основні функції і переваги Laravel:

- модульна система упаковки з управлінням залежностями. Це означає, що можна легко додавати функції у додаток Laravel, не розробляючи їх з нуля. Можна або створити власні пакети для коду, який регулярно використовується, або встановити готові до використання пакети через Composer,
- повна система аутентифікації,
- об'єктно-реляційне відображення. Виразний ORM, що входить до складу Laravel, відображає таблиці баз даних як класи для полегшення доступу до даних та маніпулювання ними.

					ДП.6417.03.000 ПЗ	Арк.
						25
Зм.	Арк.	№ докум.	Підпис	Дата		

- інтерфейс командного рядка (CLI), який постачається з десятками вбудованих команд (Artisan),
- автоматичне тестування. Автоматизовані тести представлені як невід'ємна частина Laravel,
- портативне, віртуальне середовище розробки, що надає розробникам усі інструменти, необхідні для розробки Laravel прямо з коробки.

Laravel надає користувачеві гнучкий підхід до визначення маршрутів у веб-програмі. Маршрутизація допомагає краще масштабувати додаток та підвищує його ефективність.

Веб-додаток, розроблений в Laravel, буде працювати в різних середовищах, а це означає, що буде постійно змінюватися його конфігурація. Laravel забезпечує послідовний підхід до ефективної обробки конфігурації.

Laravel використовує систему шаблонів Blade, легку мову розмітки, що використовується для проектування ієрархічних блоків та макетів із заздалегідь визначеними блоками, що містять динамічний вміст. Також Laravel включає поштовий клас, який допомагає надсилати пошту з багатим вмістом та вкладеннями з веб-програми.

Недоліки Laravel:

- відсутність вбудованої підтримки,
- проблематичний з певними оновленнями,
- може здатися складним на перший погляд,
- розробка не така швидка порівняно з іншими рамками,
- багато методів, що входять до процесу зворотної маршрутизації, є складними.

У порівнянні з іншими фреймворками, такими як Ruby on Rails та Django, Laravel має обмежену вбудовану підтримку, оскільки вона легка. Проте цю проблему можна вирішити за допомогою сторонніх інструментів, і можна повернутися до розробки дуже швидко.

					ДП.6417.03.000 ПЗ	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

До PHP платформ є кілька питань щодо версій з довгостроковою підтримкою, і Laravel іноді критикують через це. Це правда, що оновлення можуть спричинити незначні проблеми, однак при належній увазі розробники швидко налагоджують процес.

Laravel передбачає важку документацію, тобто, доведеться мати певний досвід, щоб розпочати роботу зі створення програм. Однак завдяки відданій підтримці Laravel та величезній спільноті розробників в Інтернеті ці початкові кроки можуть бути безболісними.

2.2.1.2. Огляд веб-фреймворків мови програмування Ruby

Ruby on Rails - це програмний код, який додається до мови програмування Ruby. Технічно це бібліотека пакетів, яка встановлюється за допомогою інтерфейсу командного рядка операційної системи. Rails встановлює конвенції для більш легкої співпраці та обслуговування. Ці конвенції систематизовані як API Rails (інтерфейс програмування додатків або директиви, що управляють кодом). Rails поєднує мову програмування Ruby з HTML, CSS та JavaScript, щоб створити веб-додаток, який працює на веб-сервері.

Одним із ключових принципів розробки Ruby on Rails є домовленість щодо конфігурації. Програмісту не потрібно витрачати багато часу на налаштування файлів; щоб отримати налаштування, Rails поставляється з набором конвенцій, які допомагають прискорити розробку.

Особливості Ruby on Rails:

- процес програмування відбувається набагато швидше, ніж з іншими фреймворками та мовами, частково через об'єктно-орієнтований характер Ruby та велику колекцію відкритого коду, доступного в спільноті Rails,

					ДП.6417.03.000 ПЗ	Арк.
						27
Зм.	Арк.	№ докум.	Підпис	Дата		

- конвенції Rails також дозволяють розробникам легко переходити між різними проектами Rails, оскільки кожен проект буде дотримуватися однакової структури та практики кодування,
- Rails добре підходить для швидкої розробки додатків, оскільки фреймворк полегшує пристосування до змін,
- код Ruby дуже читабельний і переважно самодокументований. Це збільшує продуктивність, оскільки менше необхідності вписувати окрему документацію, що полегшує іншим розробникам залучення до існуючих проектів,
- Rails приділяє сильну увагу тестуванню та має хороші фреймворки для тестування,
- Rails та більшість його бібліотек є відкритим кодом, тому на відміну від інших комерційних систем розробки, жодних витрат на ліцензування не потребує.

Там, де інші фреймворки використовують створення коду з нуля, Rails використовує методи метапрограмування для написання програм. Ruby - одна з найкращих мов для метапрограмування, і Rails добре використовує цю можливість. Rails також користується генерацією коду, але значно більше покладається на метапрограмування.

Rails представляє структуру Active Record, яка зберігає об'єкти в базі даних. Версія Rails Active Record виявляє стовпці в схемі бази даних і автоматично приєднує їх до об'єктів домену за допомогою метапрограмування.

Rails надає три середовища за замовчуванням: розробка, тестування та виробництво. Кожен поводить ся дещо інакше, полегшуючи весь цикл розробки програмного забезпечення. Наприклад, Rails створює нову копію тестової бази даних для кожного тестового запуску.

Недоліки Ruby on Rails:

- низька продуктивність,

					ДП.6417.03.000 ПЗ	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

- проблеми масштабованості,
- не всі хости веб-сайтів можуть підтримувати Rails,
- інші мови та фреймворки використовуються ширше і мають більше число зацікавлених розробників.

Хоча Rails має більш низькі показники швидкості обробки запитів, ніж, наприклад, Node.js або Golang, це стає помітним лише для величезних продуктів із великим трафіком. І якщо це не величезна програма з великою кількістю користувачів і запитів, Rails не обов'язково повинен бути винуватцем низьких швидкостей - також слід враховувати архітектуру сервера або бази даних. Завдяки продуманій архітектурі та інфраструктурі (необхідність у всіх масштабних проектах, незалежно від мови програмування), навіть величезні програми або їх частини, написані на Rails, можуть бути швидкими. Оскільки Rails робить стільки за розробника, недосвідчені розробники, як правило, приймають неправильні рішення при написанні коду. При поганому коді простежується суттєве падіння продуктивності.

Хоча це правда, що не всі веб-хости підтримують Rails, але це передусім тому, що Rails проекти можуть бути більш ресурсомісткими, ніж PHP, що стримує хостинг-провайдерів низького класу. Однак, існують деякі провайдери і PaaS (Platform as a Service) рішення, що дають змогу розгорнути Rails проект, наприклад, Heroku. Крім того, можна розмістити свою Rails систему на віртуальному приватному сервері (VPS) за допомогою Amazon, Google Cloud Platform, Microsoft Azure та інших. Вони забезпечують повний контроль над сервером і можливість виділити достатню кількість ресурсів для програми.

2.2.1.2. Огляд веб-фреймворків мови програмування C#

ASP.NET - це веб-фреймворк з відкритим кодом, розроблена Microsoft для створення сучасних веб-додатків та служб. За допомогою ASP.NET ви

					ДП.6417.03.000 ПЗ	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дата		

можете швидко створювати веб-сайти на основі HTML, CSS та JavaScript, масштабувати їх для використання мільйонами користувачів та легко додавати більш складні можливості, такі як веб-API, форми даних та комунікації в режимі реального часу.

Існує маса вагомих причин використовувати ASP.NET при розробці веб-сайту чи програми. Висока швидкість, низька вартість та широка підтримка мови є одними з найбільш значущих переваг. ASP.NET вбудований у звичне середовище сервера Windows, вимагає меншої настройки та конфігурації, ніж інші платформи веб-розробки, які повинні бути встановлені та налаштовані окремо. Популярність ASP.NET дозволяє легко знайти Інтернет-ресурси та кваліфікованих розробників. Веб-сайти та програми, створені за допомогою ASP.NET, можуть бути швидшими та ефективнішими, ніж, наприклад, створення веб-сайтів за допомогою PHP. Програми ASP.NET компілюються, що означає, що код переводиться в об'єктний код, який потім виконується. Цей процес компіляції займає невелику кількість часу, але відбувається лише один раз. Після компіляції код може бути виконаний знову і знову платформою .Net дуже швидко. Інтерпретований код не виконується машиною безпосередньо, але його потрібно читати та інтерпретувати кожен раз перед його виконанням. Скомпільований код, як правило, швидший і більш масштабований, ніж інтерпретований код, і він може робити все, що може інтерпретувати код. Приклади інтерпретованих мов включають PHP, JavaScript та Ruby. Процес компіляції також забезпечує перевірку відповідності всього коду.

Вартість програмного забезпечення є важливим фактором при розробці веб-сайту. Раніше розробникам доводилося купувати дороге програмне забезпечення (IDE - Інтегровані середовища розробки) для створення додатків. У наші дні більшість коду ASP.NET можна написати, використовуючи безкоштовні інструменти.

Важливе рішення при розробці веб-додатку - який тип бази даних використовувати. Програми ASP.NET підтримують всі популярні бази даних,

					ДП.6417.03.000 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

включаючи Microsoft SQL Server, MySQL, MariaDB, Postgres, MongoDB та CouchDB. Хоча Microsoft SQL Server може вимагати дорогих ліцензій на програмне забезпечення для складних розгортань, безкоштовна версія Express підходить для переважної більшості веб-сайтів. Усі інші бази даних, згадані тут, є безкоштовними та відкритими.

Недоліки ASP.NET:

- не визначена архітектура прикладних програм із відсутністю поділу відповідальностей (SoC - Separation of Concerns),
- складні веб-сторінки з проблемами продуктивності,
- відсутність абстракцій для контролю над HTML,
- обмежена підтримка тестування та SEO (оптимізація для пошукових систем),
- недостатня можливість повторного використання коду та паралельної розробки,
- труднощі вивчення.

2.2.2. Критерії вибору фреймворку для розробки системи електронного забезпечення

Існує чимало веб-фреймворків майже для кожної мови програмування. Завдяки такому численному вибору, може бути важко вибрати найбільш відповідний, який є найкращою відправною точкою для розробки нової системи.

Деякі критерії, які можуть вплинути на рішення:

- зусилля для навчання,
- продуктивність розробки,
- мета, направленість фреймворку,
- підтримка кешування,
- продуктивність фреймворку,
- веб-безпека.

					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

Кількість зусиль для вивчення веб-фреймворку залежить від ступеню ознайомлення з базовою мовою програмування, узгодженістю її API, якістю її документації та розміром та активністю спільноти.

Продуктивність розробки - це міра того, як швидко можна створити нові функції, після ознайомлення з фреймворком, і включає в себе як зусилля для написання, так і підтримання коду. Більш продуктивними є орієнтовні фреймворки, що, як правило, є більш продуктивними при вирішенні загальних проблем, проте вони менш гнучкі.

Деякі веб-фреймворки спочатку були створені для вирішення певних типів проблем і залишаються кращими у створенні платформ із подібними обмеженнями.

Кешування - це оптимізація, де ви зберігаєте всю або частину веб-відповіді, щоб її не потрібно було створювати заново при наступних запитах. Повернення кешованої відповіді набагато швидше, ніж обчислення спочатку. Кешування може бути реалізовано на внутрішньому або зовнішньому сервері.

Зазвичай продуктивність фреймворку або мови програмування, що його реалізує, не є найбільшим фактором у виборі, тому що навіть відносно повільні періоди виконання є більш ніж достатніми для систем середнього розміру, що працюють на помірному обладнанні.

Деякі веб-рамки надають кращу підтримку для захисту від поширених веб-атак. Наприклад, валідує весь вхід користувача з шаблонів HTML, щоб неможливо було запустити JavaScript, введений користувачем. Фреймворки забезпечують подібний захист, але він не завжди включений за замовчуванням.

Існує багато інших можливих факторів, включаючи ліцензування, незалежно від того, чи активно розробляється фреймворк тощо.

					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

2.2.3. Аналіз розглянутих рішень

На основі розглянутих технологій можна створити порівняльну характеристику за основними критеріями, що наведено у таблиці 2.1.

Таблиця 2.4. Порівняльна характеристика розглянутих фреймворків

Назва фреймворку	Мови програмування	Продуктивність розробки	Рівень безпеки	Складність навчання	Продуктивність роботи
Django	Python	висока	високий	легкий	низька
Flask	Python	середня	середній	легкий	низька
Symfony	PHP	висока	високий	середній	середня
Laravel	PHP	середня	середній	високий	низька
Ruby on Rails	Ruby	середня	середній	середній	низька
ASP.NET	C#	середня	середній	високий	висока

З таблиці порівняння видно, що ефективність фреймворків варіюється за різними критеріями. З огляду зрозуміло, що всі вищенаведені фреймворки можуть підійти для розробки необхідної системи, але був обраний Symfony через високу гнучкість, різноманітність та зручність налаштування, простоту у застосуванні. До того ж вся платформа автоматизації навчального процесу розробляється з допомогою цього фреймворку, а отже складність інтегрування системи забезпечення навчально-методичної діяльності стає набагато меншою.

Висновки до розділу 2

В результаті аналізу поставленої задачі розробки системи забезпечення навчально-методичної діяльності для платформи автоматизації навчального процесу було розглянуто основні технології, необхідні для розробки і інтегрування цієї системи.

Крім того, було розглянуто порівняльну характеристику між найбільш прийнятними веб-фреймворками. Було встановлено, що Symfony - потужний і міцний PHP фреймворк, спрямований на спрощення та пришвидшення процесу розробки. Проекти, створені на цій платформі, є дуже розширюваними через модульну архітектуру. Використання пакетів та компонентів робить його блискучим рішенням для веб-сайтів та додатків будь-якого розміру та складності. Хоча навчитися цьому нелегко, завжди можна знайти гарну документацію, офіційні курси та підтримку спільноти розробників. Отже, Symfony найкраще підходить для вирішення поставленої задачі.

					ДП.6417.03.000 ПЗ	Арк.
						34
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3

РЕАЛІЗАЦІЯ СИСТЕМИ ЕЛЕКТРОННОГО ЗАБЕЗПЕЧЕННЯ

3.1. Архітектура системи

Програмна реалізація системи електронного забезпечення навчально-методичної діяльності університету являє собою реалізацію функціоналу управління структурою університету і його підрозділів та навчальними планами і взаємодії з ним через гнучкий і зрозумілий користувацький інтерфейс. Такий функціонал має бути інтегрований в існуючу платформу автоматизації навчального процесу і повинен вміти взаємодіяти з існуючими сутностями, не шкодячи і не ламаючи суміжних систем. На рис. 3.1 зображено структурну схему системи автоматизації навчального процесу з головними компонентами цієї системи.

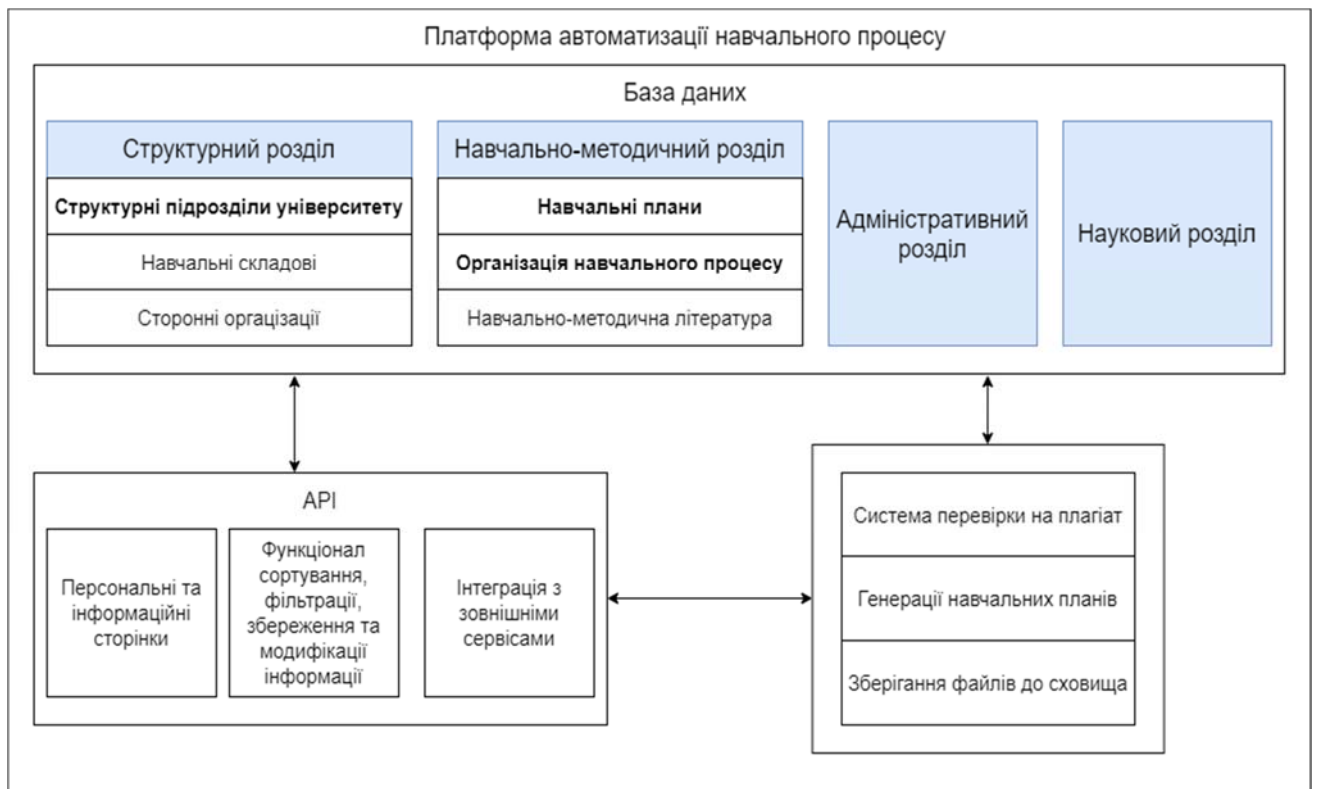


Рис. 3.1. Схема системи автоматизації навчального процесу

Як видно зі схеми, система являє собою величезну платформу з великою кількістю процесів взаємодії компонентів. До основних компонентів

платформи можна віднести базу даних, що також поділяється на деякі логічні розділи, прикладний програмний інтерфейс (API, Application Programming Interface), що містить ендпоінти – точки прийому і обробки запитів та повернення даних – для роботи з персональними сторінками студентів і викладачів, інформаційними сторінками; для взаємодії з базою даних: читання, обробка і повернення інформації. Також платформа має системи перевірки на плагіат та нормаконтроль, генерації навчальних планів, взаємодії із сховищем. Головними складовими системи забезпечення навчально-методичної діяльності є виділені на схемі, а саме структурні підрозділи університету, навчальні плани, організація навчального процесу, а також робота з API у вигляді створення ендпоінтів для зв'язку з інтерфейсом.

Для реалізації структури університету визначено головні складові:

- кафедра,
- спеціальність,
- спеціалізація,
- напрямок навчання,
- рівень вищої освіти,
- форма навчання,
- підрозділ.

3.2. Структура університету

З попереднього підрозділу зрозуміло, що реалізація управлінням структурою університету є однією із першочергових завдань.

Вся структура університету організована ієрархічно (у вигляді дерева), що зображено на діаграмі на рис. 3.2. На вершині ієрархічної структури знаходиться сам заклад - університет, а далі йдуть так звані структурні підрозділи. За університетом розташовується факультет - структурний підрозділ ВНЗ, що об'єднує кілька кафедр, які є належать певному факультету (два факультети не можуть ділити одну кафедру). Після факультету йде

					ДП.6417.03.000 ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підпис	Дата		

кафедра - основний, центровий функціональний підрозділ, що навчає студентів за спеціальностями і визначає подальше розгалуження структури. Далі знаходиться спеціальність - структурний підрозділ, що визначає вид професійної підготовки. В університеті декілька кафедр можуть весту одну і ту саму спеціальність, а кожна спеціальність може поділятися на кілька спеціалізацій, але спеціалізації усі унікальні. Наступним підрозділом є спеціалізація - особлива унікальна сутність, що не може повторюватись двічі в університеті, тобто не може бути двох однакових спеціалізацій. Коли формуються освітні програми, всі студенти поступають на визначену спеціалізацію. І нарешті для кожної спеціалізації створюється навчальний план - сутність за допомогою якої створюється організація навчального процесу. Кожний навчальний план є індивідуальним для спеціалізації.

Також у системі структурних підрозділів університету можна вирізняти і такі додаткові одиниці:

- деканат,
- бібліотека,
- аспірантура,
- студентська рада,
- профком,
- вчена рада,
- приймальна комісія,
- ректорат,
- адміністрація.

					ДП.6417.03.000 ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

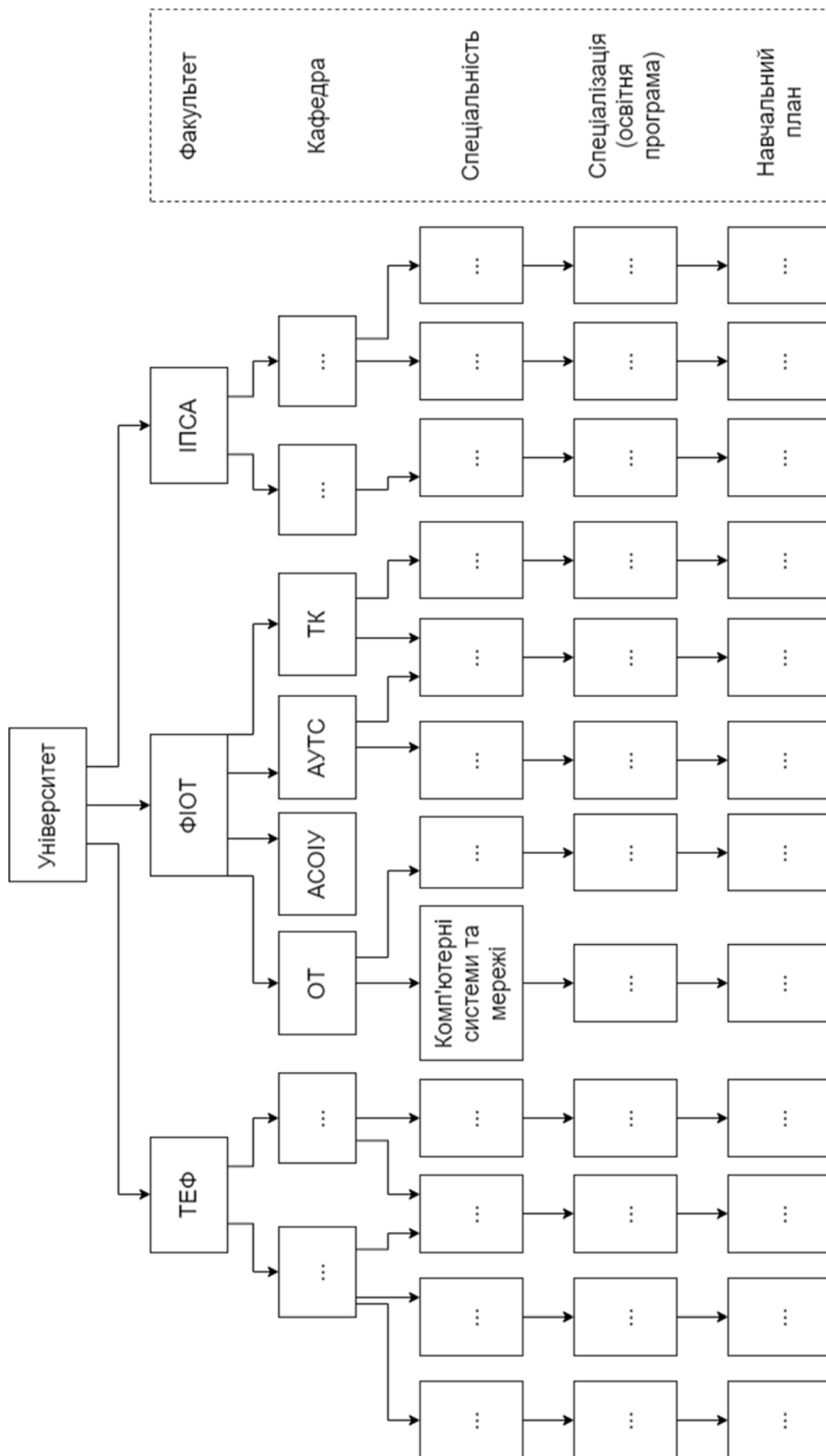


Рис. 3.2. Ієрархічне зображення структури університету

Зм.	Арк.	№ докум.	Підпис	Дата

3.3. Застосовані технології

Платформа використовує широкий спектр технологій, який охоплює всі напрямки розробки, такі як розробка клієнтської частини (фронтенд), розробка серверної частини (бекенд), системне адміністрування, бази даних. У наступних підпунктах розглядаються технології, застосовані при розробці.

3.3.1. Опис елементів системи

Для побудови архітектури використовується Docker-контейнеризація. Docker полегшує створення, розгортання та запуск програм за допомогою контейнерів, а контейнери дозволяють розробнику пакувати додаток з усіма необхідними йому частинами, такими як бібліотеки та інші залежності, і доставляти все це як один пакет. Роблячи це, розробник може бути впевнений, що додаток буде працювати на будь-якій іншій подібній за системними вимогами машині незалежно від будь-яких налаштованих параметрів, які можуть відрізнятися.

Зв'язує роботу клієнтської та серверної частини, обробляючи запити, веб-сервер nginx – веб-сервер з відкритим кодом, який також є і балансувальником навантаження. Цей сервер був обраний через одну з найкращих продуктивностей, хорошу масштабованість на машинах з не дуже хорошими технічними можливостями, простоту налаштувань і конфігурацій зв'язку. Продуктивність nginx досягається завдяки легковажним воркерам, що працюють в одному процесі конкуруючи один з одним за процесорний час у моменти блокування, пов'язані з вводом-виводом. Використання воркерів є кращим за розпаралелювання системи з допомогою потоків або процесів, адже останні є набагато важчими і витрачають багато часу на розмноження і переключення контекстів.

У якості бекенд фреймворку було обрано Symfony на основі порівняльної характеристики, щ обула проведена у другому розділі. За допомогою Symfony виконується розробка API ендпоінтів з якими взаємодія фронтенд платформа.

					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

Також Symfony управляє додатковими пакетами, такими як Doctrine-ORM – бібліотека відображення об’єктів мови PHP у SQL-сутності; Sonata Admin – пакет, що спрощує побудову панелі адміністрування, абстрагуючи розробника від різноманітної рутинної роботи.

Для зберігання інформації використовується система управління базами даних PostgreSQL – сучасна СУБД, що підтримує усі необхідні типи даних, легко масштабується, є дуже міцною і надійною, потребує мінімум витрат на технічне обслуговування, а також експлуатаційних витрат і процес відновлення, підтримує наслідування таблиць. Екземпляр PostgreSQL запускається в окремому Docker-контейнері. Доступ до бази даних відбувається за допомогою Symfony і Doctrine-ORM.

Для побудови інтерфейсу користувача використовується бібліотека ReactJS – сучасна бібліотека, що має компонентний підхід до розробки інтерфейсів. Для управління залежностями використовується пакетний менеджер npm і Webpack для збірки і тестування фронтенду.

Сховище даних – останній Docker-контейнер. В якості системи зберігання файлів вибрана система Minio – високоефективне сховище об’єктів, що сумісне з сервісами хмарного зберігання Amazon S3.

Описана архітектура контейнерів Docker і інших компонентів представлена у вигляді структурної схеми, що зображена у Додатку 1.

3.4. Опис програмної логіки роботи системи

У роботі платформи автоматизації управління університету бере участь велика кількість підсистем з певними класами, що виконують визначені дії і описують логіку роботи підсистем, обробляючи дані через моделі і сутності. Логіка роботи полягає у тому, що користувач системи повинен мати змогу взаємодіяти і управляти структурою університету і навчальними планами через певні розділи панелі адміністрування.

					ДП.6417.03.000 ПЗ	Арк.
						40
Зм.	Арк.	№ докум.	Підпис	Дата		

Опираючись на MVC-шаблон проектування, сутності представляються у вигляді моделей, що забезпечується Doctrine-ORM. У випадку кафедри, сутність має відповідні інтерфейси *DepartmentInterface* і *DivisionInterface*, що реалізуються класами моделей *Department* і *Division*. Дані класи забезпечують доступ до даних відображених таблиць в базі даних, а також їх модифікацію. Складові класу *Division*: *name* – ім'я (аббревіатура) підрозділу, *fullName* – повне ім'я підрозділу, *webSite* – веб-сайт, *priority* – пріоритет, *parent* – батьківський підрозділ, *children* – посилання на дочірні підрозділи, *type* – тип підрозділу. Також *Division* є складовою і інших підрозділів. Клас *Department* має такі складові елементи: *division* – OneToOne посилання на відповідний підрозділ, *specialities* – перелік спеціальностей (реалізований зв'язок ManyToMany).

Сутність підрозділу спеціальності реалізовує інтерфейс *SpecialityInterface*, який реалізовується класом *Speciality*. Складові класу *Speciality*: *identifier* – ідентифікатор спеціальності, *name* – ім'я, *studyDirection* – напрямок навчання, *specializations* – перелік відповідних спеціалізацій. Всі інші елементи підрозділ отримує із класу *Division*.

Сутність підрозділу спеціалізації реалізовує інтерфейс *SpecializationInterface*, який реалізовується класом *Specialization*. Складові класу *Specialization*: *identifier* – ідентифікатор спеціальності, *name* – ім'я, *speciality* – відповідна спеціальність, *qualification* – кваліфікації, *departments* - кафедри.

Логіка взаємодії з сутностями реалізовується за допомогою інструментів-обробників пакету SonataAdmin, що дозволяють швидко створювати інтерфейси відображення інформації в панелі адміністрування, а також створювати форми для модифікації або видалення певних сутностей, аналізуючи створені моделі. Тож за логіку моделей *Division*, *Department*, *Speciality*, *Specialization* турбуються класи панелі адміністрування

					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

DivisionAdmin, *DepartmentAdmin*, *SpecialityAdmin*, *SpecializationAdmin* відповідно.

Сутність навчального плану *AcademicPlan* реалізовує інтерфейс *AcademicPlanInterface*. *AcademicPlan* має такі складові елементи: *creationDate* – дата створення, *enrollmentYear* – рік зарахування, *speciality* – спеціальність, *specialization* – спеціалізація, *qualification* – кваліфікація, *studyLevel* – рівень навчання, *studyForm* – форма навчання, *studyDuration* – тривалість навчання, *department* – кафедра, *groups* – групи, *dean* – декан, *headOfDepartment* – завідуючий кафедри, *protocolApprovedLocally* - затверджено проректором, *protocolApprovedGlobally* - ухвалено на засіданні Вченої ради факультету.

Логіка управління навчальними планами реалізується класом панелі адміністрування *AcademicPlanAdmin*.

					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

Висновки до розділу 3

Було визначено і описано загальну архітектуру платформи автоматизації навчального процесу. На основі поставлених задач, було визначено загальну схему складових проекту і побудовано схему системи автоматизації навчального процесу.

Так як реалізація управління структурою університету є одним із основних завдань, було визначено чітку структуру і її ієрархічну природу, а також встановлено чіткі зв'язки між підрозділами. На основі визначених даних було побудовано ієрархічне зображення структури університету.

Після визначення структури університету було описано застосовані технології. Серед них основними компонентами є система контейнеризації Docker, веб-сервер nginx, СУБД – PostgreSQL, сховище Minio; в якості веб-фреймворку був обраний Symfony, а фронтенд-фреймворку ReactJS.

Було виконано опис програмної логіки роботи системи і розглянуто взаємодію між сутностями системи. У наступному розділі буде описано розгортання, тестування і користування системою.

					ДП.6417.03.000 ПЗ	Арк.
						43
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 4

ТЕСТУВАННЯ РОБОТИ СИСТЕМИ

4.1. Опис налаштування і розгортки проекту

Проект має деякі технічні обмеження і складнощі, що потрібно вирішити перед локальним запуском на комп'ютері, а саме його апаратні та системні характеристики. Можна визначити мінімальні технічні характеристики обчислювальної машини:

- процесор, бажано багатоядерний, з частотою 1.7 ГГц,
- об'єм ОЗУ 4 гігабайт для плавної роботи системи,
- HDD або SSD з вільною пам'яттю близько 50 гігабайт.

Характеристики і різновиди програмного забезпечення також мають вплив на розгортання проекту. Найкраще система буде працювати на UNIX-подібних операційних системах, а саме macOS і системах з ядром Linux. Бажані версії macOS – 10.9 і вищі. В залежності від встановленого дистрибутиву Linux потреби щодо встановлення пакетів можуть відрізнятись, але є рекомендоване програмне забезпечення загальне для всіх:

- GTK+ 3.6 +,
- Dbus 1.3 +,
- GLib 2.37 +,
- GNOME 2.22 +,
- Pango 1.48 +.

Запуску проекту на ОС Windows може бути дуже складним для реалізації через технічні особливості розробки проекту. Адже багато компонентів системи знаходяться в Docker-контейнерах, а їх локальне налаштування може бути дуже трудомістким і затратним у часі.

Також для запуску необхідне певне програмне забезпечення, із частин якого проект і складається, а саме:

- PHP 7.2 +,

					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

- Docker останньої версії,
- Docker-compose останньої версії,
- npm 6 +,
- СУБД PostgreSQL 10 +,
- Webpack CLI.

Встановивши усе необхідне, потрібно завантажити репозиторій з проектом з системи управління системою контролю версій gitlab, після отримання прав розробника:

```
git clone git@gitlab.com:kpi-connect/kpi-connect.git
```

Етапи налаштування проекту у його директорії:

1. *composer install* – встановити всі додаткові пакети, потрібні для запуску Docker-контейнерів.
2. *npm i* і *gulp dev* – для встановлення необхідних конфігурацій інтерфейсу панелі адміністрування.
3. *cd dev/docker* і *docker-compose up* – для запуску контейнерів.
4. *./build.sh* – запустити конфігураційні сценарії проекту.
5. *php bin/console server:start *:80* (macOS) або *sudo php7.2 bin/console server:start *:80* (Linux) – запуск локального серверу

Проект розгорнеться і буде доступний за адресою *localhost:80*.

Далі виконано розгляд роботи із системою і опис інтерфейсів користувача і форм заповнення даних, пов'язаних із структурою університету і навчальними планами.

4.2. Опис інтерфейсу адміністративної панелі системи

Після авторизації у системі під профілем адміністратора, буде переадресовано на головну сторінку адміністративної панелі, що виглядає як зображено на рис. 4.1:

					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

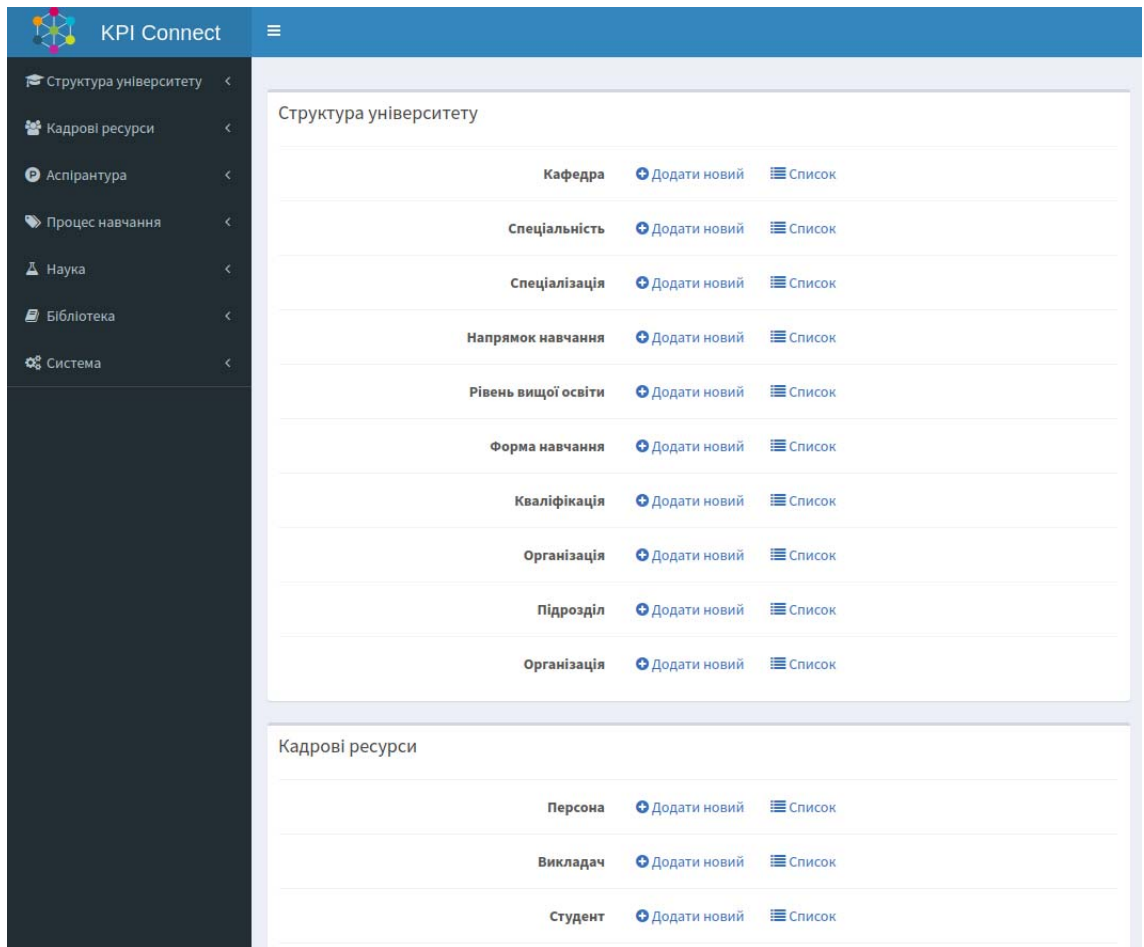


Рис. 4.1. Головна сторінка адміністративної панелі

З головної сторінки можна перейти до всіх розділів платформи автоматизації навчальних процесів, у тому числі до структури університету і навчальних планів. Бічна панель зліва є навігаційним меню по всій системі.

4.2.1. Опис інтерфейсу структури університету

Як зазначалося у третьому розділі, КПІ ім. Ігоря Сікорського має досить складну ієрархічну структуру. Структура університету в системі складається з розділів, зображених на рис. 4.2:

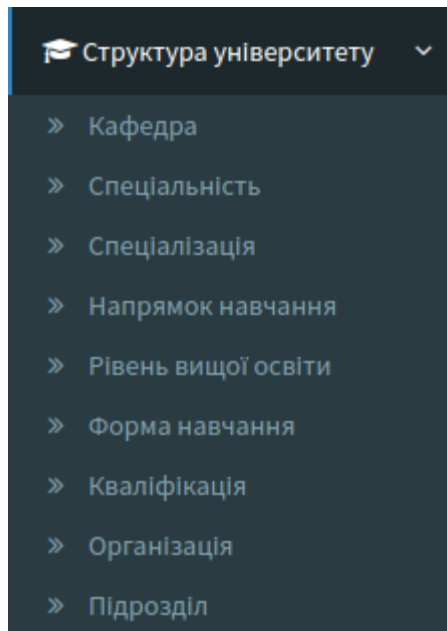


Рис. 4.2. Перелік розділів блоку «Структура університету»

Орієнтуючись на це меню, можна переходити до відповідних сторінок структур і підрозділів університету.

Кафедра

<input type="checkbox"/>	Назва	Повна назва	Веб-сайт	Факультет	Дія
<input type="checkbox"/>	ОТ	Обчислювальної техніки	http://comsys.kpi.ua/	ФІОТ	<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>
<input type="checkbox"/>	АСОІУ	Автоматизованих систем обробки інформації і управління	http://asu.kpi.ua/	ФІОТ	<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>
<input type="checkbox"/>	АУТС	Автоматики і управління в технічних системах	http://acts.kpi.ua/	ФІОТ	<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>
<input type="checkbox"/>	ТК	Технічної кібернетики	http://tc.kpi.ua/uk/	ФІОТ	<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>
<input type="checkbox"/>	АПЕПС	Автоматизації проектування енергетичних процесів та систем	http://apeps.kpi.ua/	ТЕФ	<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>
<input type="checkbox"/>	АТЕП	Автоматизації теплоенергетичних процесів	http://atep.kpi.ua/	ТЕФ	<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>

Рис. 4.3. Вікно розділу «Кафедра»

Розділ Кафедра являє собою довідкову сторінку з переліком усіх кафедр у вигляді списку. Кожна кафедра містить аббревіатуру і повну назву кафедри, веб-сайти і назву факультету, до якого кафедра відноситься з прямими посиланнями на них. Усі кафедри зі списку якимось пов'язані з Факультетом

інформатики та обчислювальної техніки, тому якщо у списку є кафедри з інших факультетів - це означає, що ця кафедра викладає предмети із навчальних планів, які відносяться до ФІОТ. Також є поля з діями Показати, Редагувати, завдяки яким можна відвідати сторінку кафедри та внести якісь зміни, якщо це необхідно. При відвідуванні сторінки кафедри, вона виглядає як на рис. 4.4:

Кафедра	
Повна назва	Обчислювальної техніки
Факультет	ФІОТ
Назва	ОТ
Пріоритет	1
Веб-сайт	http://comsys.kpi.ua/
Спеціальності	
Дата оновлення	12.11.2018 15:23:52
Дата створення	25.05.2018 14:29:22

Рис. 4.4. Сторінка з детальною інформацією про кафедру

Як видно з рисунка, з'являються додаткові поля, такі як пріоритет, перелік спеціальностей, дати створення і оновлення. Вікно редагування містить форми зміни основних полів та форму додавання відповідних кафедрі спеціальностей і виглядає подібним чином:

Загальний

Підрозділ *

Обчислювальної техніки

Спеціальності

або

Рис. 4.5. Вікно редагування кафедри

Після внесених змін можна зберегти інформацію і повернутись назад до списку кафедр, а також можна видалити кафедру за необхідності.

Факультет

При переході по посиланню на факультет відкривається відповідна сторінка з інформацією про нього, як на рис. 4.6. Підрозділ «Факультет» містить такі поля: тип (назва підрозділу), назва, батьківський підрозділ (для факультетів завжди КПП ім. Ігоря Сікорського), пріоритет, веб-сайт, дати створення та оновлення. Поле дочірні підрозділи є переліком усіх підрозділів, пов'язаних з визначеним факультетом. Наприклад, для ФІОТу такими підрозділами є його кафедри (ТК, АУТС, АСОІУ, ОТ), науково-дослідний інститут інформаційних процесів, дві вчені ради, деканат.

Підрозділ	
Тип	Факультет
Повна назва	Факультет інформатики та обчислювальної техніки
Батьківський підрозділ	КПП ім. Ігоря Сікорського
Назва	ФІОТ
Пріоритет	3
Веб-сайт	http://fiot.kpi.ua
Дочірні підрозділи	<ul style="list-style-type: none"> • Технічної кібернетики • Автоматики і управління в технічних системах • Автоматизованих систем обробки інформації і управління • Обчислювальної техніки • Науково дослідний інститут інформаційних процесів • Спеціалізована вчена рада університету Д 26.002.02 з 04.04.2018 до 31.12.2020 р., Луцький Г.М., Тарасенко В.П., Орлова М.М. • High Performance Computing • Спеціалізована вчена рада університету Д 26.002.29 з 11.07.2017 до 31.12.2019 р., Новіков О.М., Павлов О.А., Теленик С.Ф. • Деканат факультету інформатики та обчислювальної техніки
Дата оновлення	21.12.2018 15:51:13
Дата створення	12.11.2018 15:23:52

Рис. 4.6. Детальна інформація про певний факультет

Також, після переходу до сторінки факультету, вкладка у меню «Структура Університету» зліва буде змінена на «Підрозділ», перейшовши до якої, можна побачити сторінку, зображену на рис. 4.7:

					ДП.6417.03.000 ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

Всі підрозділи						
Факультет		Кафедра	Деканат	Бібліотека	Аспірантура	Студентська рада
Профком		Наукова команда	Вчена Рада	Підготовчі курси		
Приймальна комісія		Ректорат	Адміністрація	Спеціалізована вчена рада	Інститут	Конструкторське бюро
Оргкомітет		Конференція				
<input type="checkbox"/>	Назва	Повна назва	Тип	Веб-сайт	Батьківський підрозділ	Дія
<input type="checkbox"/>	ТЕФ	Теплоенергетичний факультет	Факультет	http://tef.kpi.ua	КПІ ім. Ігоря Сікорського	<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>
<input type="checkbox"/>	АТЕП	Автоматизації теплоенергетичних процесів	Кафедра	http://atep.kpi.ua/	ТЕФ	<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>
<input type="checkbox"/>	АПЕПС	Автоматизації проектування енергетичних процесів та систем	Кафедра	http://apeps.kpi.ua/	ТЕФ	<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>
<input type="checkbox"/>	ФМФ	Фізико-математичний факультет	Факультет	http://fmf.kpi.ua/	КПІ ім. Ігоря Сікорського	<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>

Рис. 4.7. Сторінка вкладки «Підрозділ»

Дана сторінка містить таблицю переліку усіх підрозділів, а їх перерахування зверху допомагає фільтрувати таблицю за необхідними підрозділами. Перерахування підрозділів насправді має певні ієрархічні зв'язки, які можна частково простежити при додаванні нового підрозділу, форма чого зображена на рис. 4.8:

Загальний

Назва *

Повна назва *

Тип *

Пріоритет *

Веб-сайт

Батьківський підрозділ

Рис. 4.8. Додавання нового підрозділу

При додаванні нового підрозділу обов'язковими полями є назва, повна назва, тип і пріоритет. За бажанням можна додати веб-сайт і вказати батьківський підрозділ. При вказуванні типу необхідно вибрати один із запропонованих підрозділів зі списку, як на рис. 4.9:

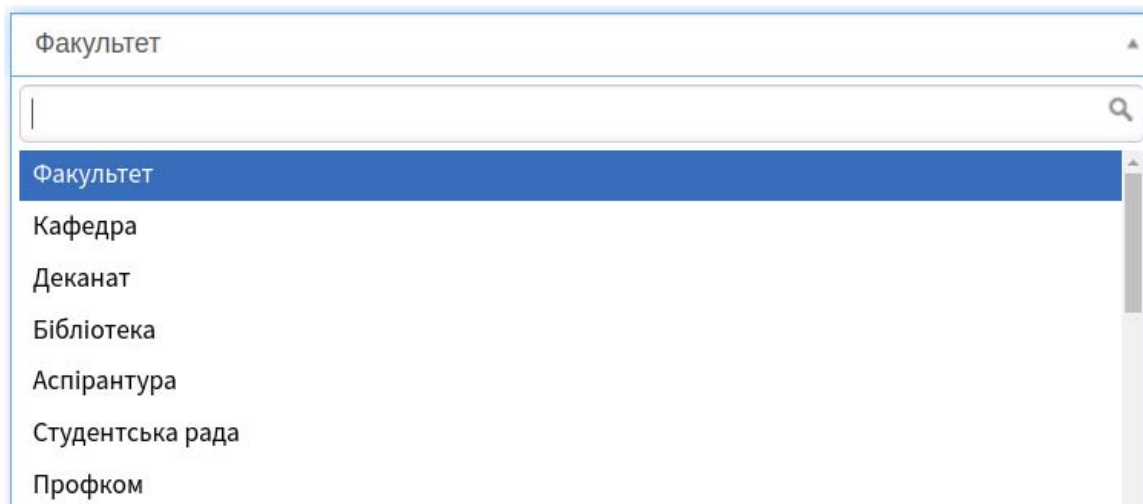


Рис. 4.10. Вибір типу підрозділу

При виборі батьківського підрозділу відкривається зручне модальне вікно, що зображене на рис. 4.11, з можливістю швидкого знаходження і вибору необхідного варіанту. Наприклад, якщо створюється кафедра, то її батьківським підрозділом буде факультет, тому у вікні вибору потрібно перейти до вкладки Факультет і із відсортованого списку обрати потрібний.

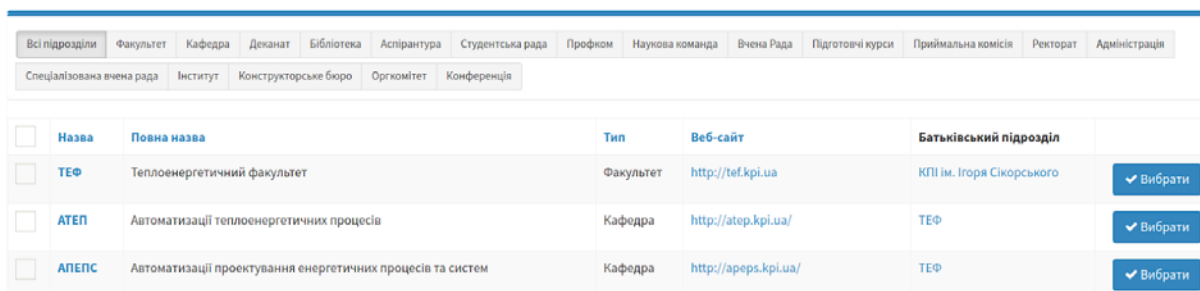


Рис. 4.11. Вікно вибору батьківського підрозділу

За умови, якщо потрібний підрозділ не був знайдений, можна повернутися назад до форми створення нового підрозділу і додати його за допомогою відповідної кнопки і після цього оголосити його батьківським підрозділом. Ці можливості дуже зручні, адже не треба витратити багато часу на переходи у інші розділи.

Спеціальність та спеціалізація

Кожна кафедра навчає студентів за спеціальностями. Перейшовши по вкладці «Спеціальність», буде відкрито сторінку-довідник з переліком спеціальностей, кожна з яких має назву, ідентифікатор та напрямок навчання, рис. 4.12. Кожна кафедра має одну або декілька спеціалізацій (освітніх програм). Перейшовши по вкладці «Спеціалізація», можна побачити таблицю унікальних спеціалізацій і відповідних їм спеціальностей, рис. 4.13.

<input type="checkbox"/>	Назва	Ідентифікатор	Кафедри	Напрямок навчання	Дія
<input type="checkbox"/>	Інженерія програмного забезпечення	121		Інформаційні технології	<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>
<input type="checkbox"/>	Комп'ютерні науки	122		Інформаційні технології	<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>
<input type="checkbox"/>	Комп'ютерна інженерія	123		Інформаційні технології	<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>
<input type="checkbox"/>	Інформаційні системи та технології	126		Інформаційні технології	<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>

Рис. 4.12. Таблиця спеціальностей

<input type="checkbox"/>	Назва	Ідентифікатор	Спеціальність	Кафедри	Дія
<input type="checkbox"/>	Комп'ютерні системи та мережі (ФІОТ)	12320401	Комп'ютерна інженерія		<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>
<input type="checkbox"/>	Технології програмування для комп'ютерних систем та мереж (ФІОТ)	12320402	Комп'ютерна інженерія		<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>
<input type="checkbox"/>	Програмне забезпечення високопродуктивних комп'ютерних систем та мереж (ФІОТ)	12120401	Інженерія програмного забезпечення		<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>
<input type="checkbox"/>	Програмне забезпечення інформаційних управляючих систем та технологій (ФІОТ)	12120402	Інженерія програмного забезпечення		<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>

Рис. 4.13. Таблиця спеціалізацій

4.2.2. Опис інтерфейсу процесу взаємодії з навчальними планами

Для кожної спеціалізації створюється і затверджується навчальний план - документ, що нормує формування навчального процесу. Для кожної форми навчання плани складаються індивідуально в залежності від освітньої програми. На рис. 4.14 зображено інтерфейс форми додавання навчального плану:

					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

Загальний

Кафедра *

Нічого не вибрано

Спеціальність *

Нічого не вибрано

Спеціалізація *

Нічого не вибрано

Кваліфікація *

Нічого не вибрано

Рік зарахування *

Форма навчання *

Нічого не вибрано

Освітній рівень *

Нічого не вибрано

Декан *

Нічого не вибрано

Завідуючий кафедри *

Нічого не вибрано

Ухвалено на засіданні Вченої ради факультету *

Затверджено проректором *

1 січ. 2015

Дата створення *

1 січ. 2015

Групи

Рис. 4.14. Форма додавання навчального плану

Інструкція додавання навчального плану

1. У полі «Кафедра» зі списку існуючих кафедр вибрати потрібну, рис. 4.15. Якщо потрібної кафедри не знайдено можна створити нову, не покидаючи сторінки з формою, натиснувши кнопку «Додати новий», яка відкриє модальне вікно з формою створення підрозділу, рис. 4.16. Якщо випадково було обрано не ту кафедру, її можна видалити за допомогою однойменної кнопки. У вікні вибору кафедри є можливість фільтрування за батьківським підрозділом і за назвою кафедри, рис. 4.15.

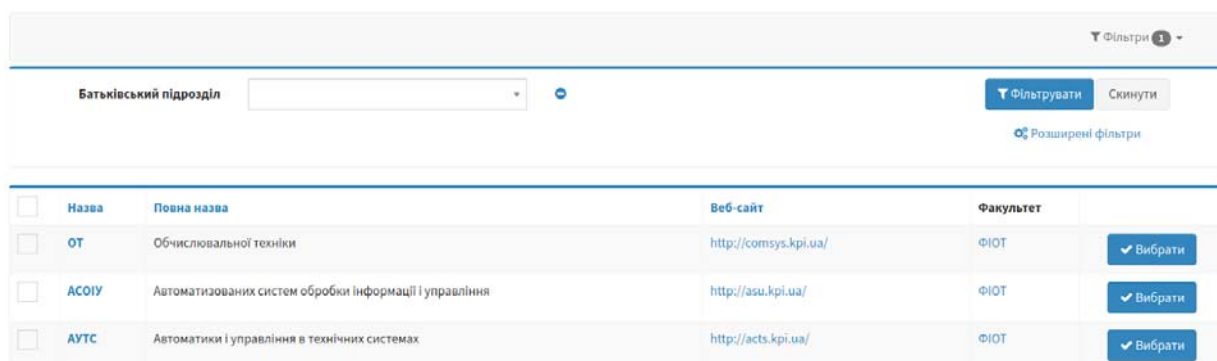


Рис. 4.15. Модальне вікно вибору кафедри

Кафедра

Загальний

Підрозділ *

Нічого не вибрано

Спеціальності

Рис. 4.16. Модальне вікно створення кафедри

2. У полі «Спеціальність» обрати потрібну спеціальність так само, як у попередньому пункті обиралась кафедра. При виборі спеціальності з'являються нові фільтри: за назвою напрямку навчання і за назвою спеціальності, рис. 4.17.

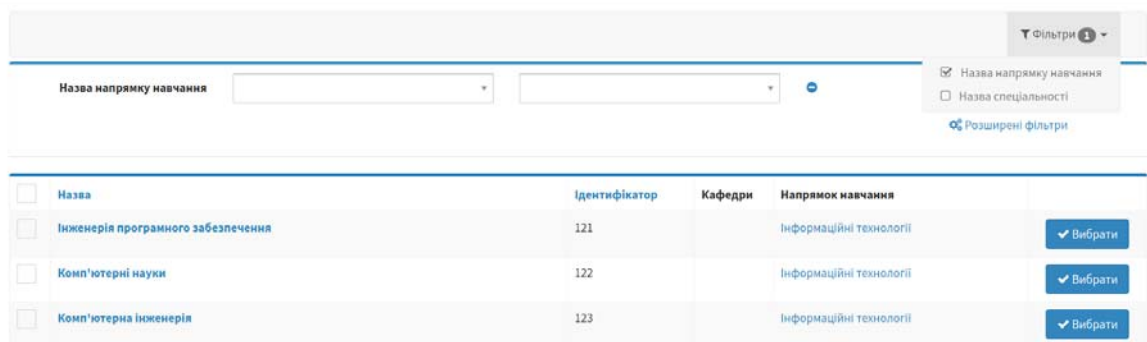


Рис. 4.17. Вікно вибору спеціальності

3. Аналогічно з попередніми кроками вибрати потрібну спеціалізацію. Можливі фільтри: за назвою спеціальності, за назвою спеціалізації, рис. 4.18.

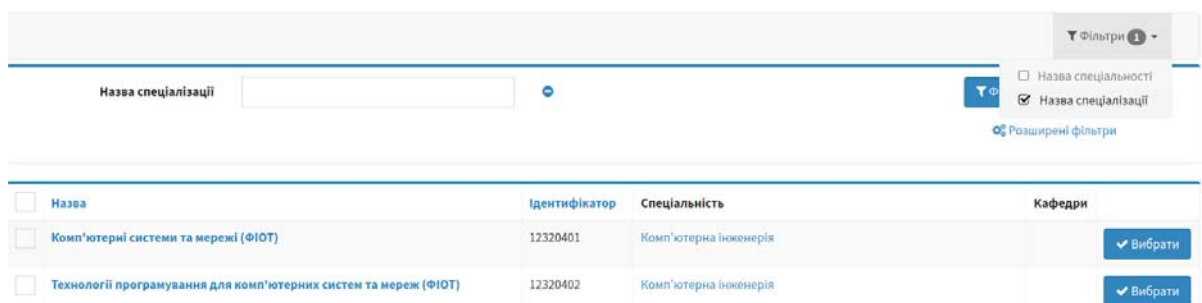


Рис. 4.18. Вікно вибору спеціалізації

4. Наступним кроком йде вибір кваліфікації. Кваліфікація обирається так само - зі списку із доступними фільтрами, а саме: назва спеціалізації, назва освітнього рівня, назва кваліфікації, рис. 4.19.

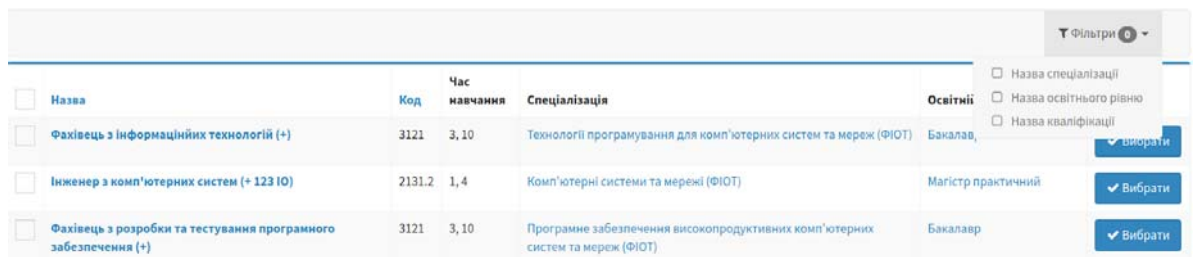


Рис. 4.19. Вікно вибору кваліфікації

5. Вказати рік зарахування. За роком зарахування буде визначатися курс.
6. Вибрати форму навчання зі списку.
7. Вибрати освітній рівень.

8. Вибрати декана. Для цього поля існує широкий вибір фільтрів: ім'я, e-mail, чи є персона студентом, викладачем або аспірантом, показувати чи ні сторонніх персон, рис. 4.20.

<input type="checkbox"/>	ПІБ	ПІБ англійською	Стать	
<input type="checkbox"/>	Клименко Ірина Анатоліївна	Iryna Klumenko	Жіноча	<input type="button" value="✓ Вибрати"/>
<input type="checkbox"/>	Буцький Юрій Петрович	Yurii Butskiy	Чоловіча	<input type="button" value="✓ Вибрати"/>

Рис. 4.20. Вікно вибору декана

9. Вибрати завідуючого кафедри. Спектр фільтрів такий самий як і при виборі декана.

10. Зазначити чи ухвалено на засіданні Вченої ради факультету.

11. Вказати дати затвердження проректором та створення.

12. Обрати визначені для навчального плану групи. Якщо потрібної групи не знайшлося, можна створити групу, натиснувши «Додати новий», рис. 4.21.

Рис. 4.22. Вікно створення групи

Після створення навчального плану є можливість його переглянути, як зображено на рис. 4.23:

Загальний	
Кафедра	Обчислювальної техніки
Спеціальність	Комп'ютерна інженерія
Спеціалізація	Комп'ютерні системи та мережі (ФІОТ)
Кваліфікація	Фахівець з інформаційних технологій (+)
Рік зарахування	2018
Курс	2
Форма навчання	Денна
Освітній рівень	Бакалавр
Тривалість навчання	3,10
Декан	Павлов Олександр Анатолійович
Завідуючий кафедри	Стіренко Сергій Григорович
Дата створення запису	29.01.2018
Ухвалено на засіданні Вченої ради факультету	РНП-1
Затверджено проректором	01.01.2014
Групи	<ul style="list-style-type: none"> • IO-81 • IO-82 • IO-83 • IO-84мн

Рис. 4.23. Сторінка з інформацією про певний навчальний план
 Переглянути список навчальних планів можна у вкладці *Процес навчання* > *Навчальний план*. Вигляд таблиці навчальних планів:

<input type="checkbox"/>	Повна назва	Кафедра	Курс	Форма навчання	Дія
<input type="checkbox"/>	Комп'ютерні системи та мережі (ФІОТ), 2018	ОТ	2	Денна	<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>
<input type="checkbox"/>	Технології програмування для комп'ютерних систем та мереж (ФІОТ), 2018	ОТ	2	Денна	<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>
<input type="checkbox"/>	Програмне забезпечення високопродуктивних комп'ютерних систем та мереж (ФІОТ), 2018	ОТ	2	Денна	<input type="button" value="Показувати"/> <input type="button" value="Редагувати"/>

Рис. 4.24. Таблиця навчальних планів

Висновки до розділу 4

Під час огляду програмної реалізації, була встановлена чітка інструкція щодо розгортання проекту на локальній машині і розглянуто основні проблеми, що можуть виникнути при запуску. Були описані кроки налаштування і конфігурації проекту після завантаження його з системи Gitlab.

Було виконане тестування і показано основні можливості взаємодії з інтерфейсом користувача. Повністю описано як працювати зі структурою університету, його підрозділами, а саме огляд і пошук інформації, створення, модифікація і інші види взаємодії з кафедрами, факультетами, спеціальностями, спеціалізаціями.

Проведено огляд форми додавання навчальних планів. Сформульована чітка інструкція додавання навчального плану з правильним заповненням полів форми. Було детально розглянуто способи заповнення, пошук і фільтрація таких полів: Кафедра, Спеціальність, Спеціалізація, Кваліфікація, Форма навчання, Освітній рівень, Декан, Групи. Розглянуто і описано інтерфейс перегляду навчальних планів.

На основі попередніх висновків і результатів тестування, можна сказати, що система працює і розроблена правильно. Можливість легко і швидко взаємодіяти з структурою університету і навчальними планами підтверджують ефективність роботи системи електронного забезпечення.

					ДП.6417.03.000 ПЗ	Арк.
						58
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В розглянутому дипломному проекті було виконано розробку системи електронного забезпечення навчально-методичної діяльності вищого начального закладу для платформи автоматизації навчального процесу, що складається з бази даних, додатку для обробки, зберігання і модифікації інформації студентів.

На етапі розробки проекту було виконано ряд задач, що дозволило виконати реалізацію роботи системи у необхідному форматі, з правильною роботою розділів структури університетів і навчальних планів. Спочатку було проведено дослідження і визначено актуальність тематики роботи. Потім було визначено і розглянуто основні компоненти платформи автоматизації, такі як адміністрування, підтримка студентів, безпека системи. Після проведення огляду існуючих рішень було визначено їх основні переваги та недоліки. Виходячи з загального положення і опираючись на сукупність всіх перерахованих вище фактів проведеного аналізу було поставлено основні задачі розробки системи.

Наступним кроком було виконання аналізу проектування системи та проблем і складнощів, що можуть виникнути під час реалізації системи. Виконано огляд технологій розробки, за допомогою яких буде виконуватись проект. Далі було визначено основні критерії за якими буде обиратися єдина технологія. В ході проведеного дослідження була побудована порівняльна таблиця наступних фреймворків: Django, Flask, Symfony, Laravel, Ruby on Rails, ASP.NET. На основі зазначених факторів було обрано фреймворк Symfony як головний для розробки.

Під час реалізації системи електронного забезпечення було продумано і описано архітектуру платформи, яку зображено на структурній схемі системи автоматизації навчального процесу Додатку 1. В ході проведеного дослідження в попередніх розділах було з'ясовано, що структура університету є важливою частиною навчально-методичної діяльності ВНЗ, а тому було

					ДП.6417.03.000 ПЗ	Арк.
						59
Зм.	Арк.	№ докум.	Підпис	Дата		

розроблено ієрархічну схему зв'язків підрозділів і виявлено способи їх взаємодії один з одним.

Серед обраних технологій для реалізації проекту наявні інструмент для управління ізольованими контейнерами Docker, швидкісний веб-сервер nginx, фреймворк PHP Symfony, система управління БД PostgreSQL і ReactJS для побудови сучасних веб-інтерфейсів.

Логіка роботи системи описана за прикладом взаємодії з сутностями підрозділів університету і навчальних планів. Детально розглянуто поля моделей і функціональних класів, що з ними взаємодіють. Функціональну схему взаємодії моделей і класів зображено у Додатку 2.

Підтвердженням правильності виконання завдань є перевірка системи у вигляді тестування інтерфейсу та функціоналу. Тестування полягає у розгляді взаємодії користувача з моделями підрозділів закладу, а також створення нових навчальних планів з детальною покроковою інструкцією заповнення даних. Результати представлені у вигляді рисунків підтверджують правильність виконаних дій. Принципова схема створення навчального плану зображена у Додатку 3.

Результатом роботи є розроблений програмний продукт у вигляді взаємодії системи електронного забезпечення навчально-методичної діяльності з платформою автоматизації навчального процесу. Отримані результати підтверджують, що система була добре інтегрована. Такі програмні продукти є перспективним напрямом у сфері освіти, адже у сучасному світі все більше діяльностей переходить у цифровий формат, особливо підрозділи, пов'язані з інформаційними технологіями. Отже, додавання нового функціоналу у вигляді інтеграції окремих систем є хорошою практикою для платформи в цілому.

					ДП.6417.03.000 ПЗ	Арк.
						60
Зм.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Greening T. Computer Science Education in the 21st Century / Tony Greening., 2012. – 266 с.
2. Ewaryst T. Internet - Technical Development and Applications / T. Ewaryst, K. Adrian. – Berlin: Springer Science & Business Media, 2009. – 284 с.
3. Philip B. Electronic Performance Support: Using Digital Technology to Enhance Human Ability / B. Philip, V. Paul., 2012. – 326 с.
4. Reza H. The Digital University - Building a Learning Community / H. Reza, H. Stephen. – Berlin: Springer Science & Business Media, 2008. – 252 с.
5. Why We Use Django Framework & What Is Django Used For [Електронний ресурс] // Django Stars EDITORIAL TEAM – Режим доступу до ресурсу: <https://djangostars.com/blog/why-we-use-django-framework/>.
6. Youssef N. What is Django? Advantages and Disadvantages [Електронний ресурс] / Nader Youssef. – 2020. – Режим доступу до ресурсу: <https://hackr.io/blog/what-is-django-advantages-and-disadvantages-of-using-django>.
7. FLASK PYTHON FRAMEWORK [Електронний ресурс] // Quintagroup – Режим доступу до ресурсу: <https://quintagroup.com/cms/python/flask>.
8. Stevonn H. 7 Good Reasons to Use Symfony Framework for Your Project [Електронний ресурс] / Hansen Stevonn. – 2017. – Режим доступу до ресурсу: <https://hackernoon.com/7-good-reasons-to-use-symfony-framework-for-your-project-265f96dcf759>.
9. Symfony [Електронний ресурс] // ntc – Режим доступу до ресурсу: <https://www.ntchosting.com/encyclopedia/frameworks/symfony/>.

					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

- 10.Laravel - Overview [Электронный ресурс] // tutorialspoint – Режим доступа до ресурсу:
https://www.tutorialspoint.com/laravel/laravel_overview.htm.
- 11.Matt W. what is asp.net and why should i use it? [Электронный ресурс] / Wiseley Matt // Wakefly. – 2018. – Режим доступа до ресурсу:
<https://www.wakefly.com/blog/what-is-asp-net-and-why-should-i-use-it/>.
- 12.Web and Database Security [Электронный ресурс] / X.Jiping, X. Lifeng, Z. Jian, H. Tao. – 2012. – Режим доступа до ресурсу:
<https://www.intechopen.com/books/security-enhanced-applications-for-information-systems/web-and-database-security>.
- 13.Django Documentation [Электронный ресурс] // Django Software Foundation. – 2020. – Режим доступа до ресурсу:
<https://buildmedia.readthedocs.org/media/pdf/django/3.0.x/django.pdf>.
- 14.Symfony Documentation [Электронный ресурс] // Symfony. – 2020. – Режим доступа до ресурсу:
<https://symfony.com/doc/current/index.html>.
- 15.Ruby on Rails Guides [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://guides.rubyonrails.org/>

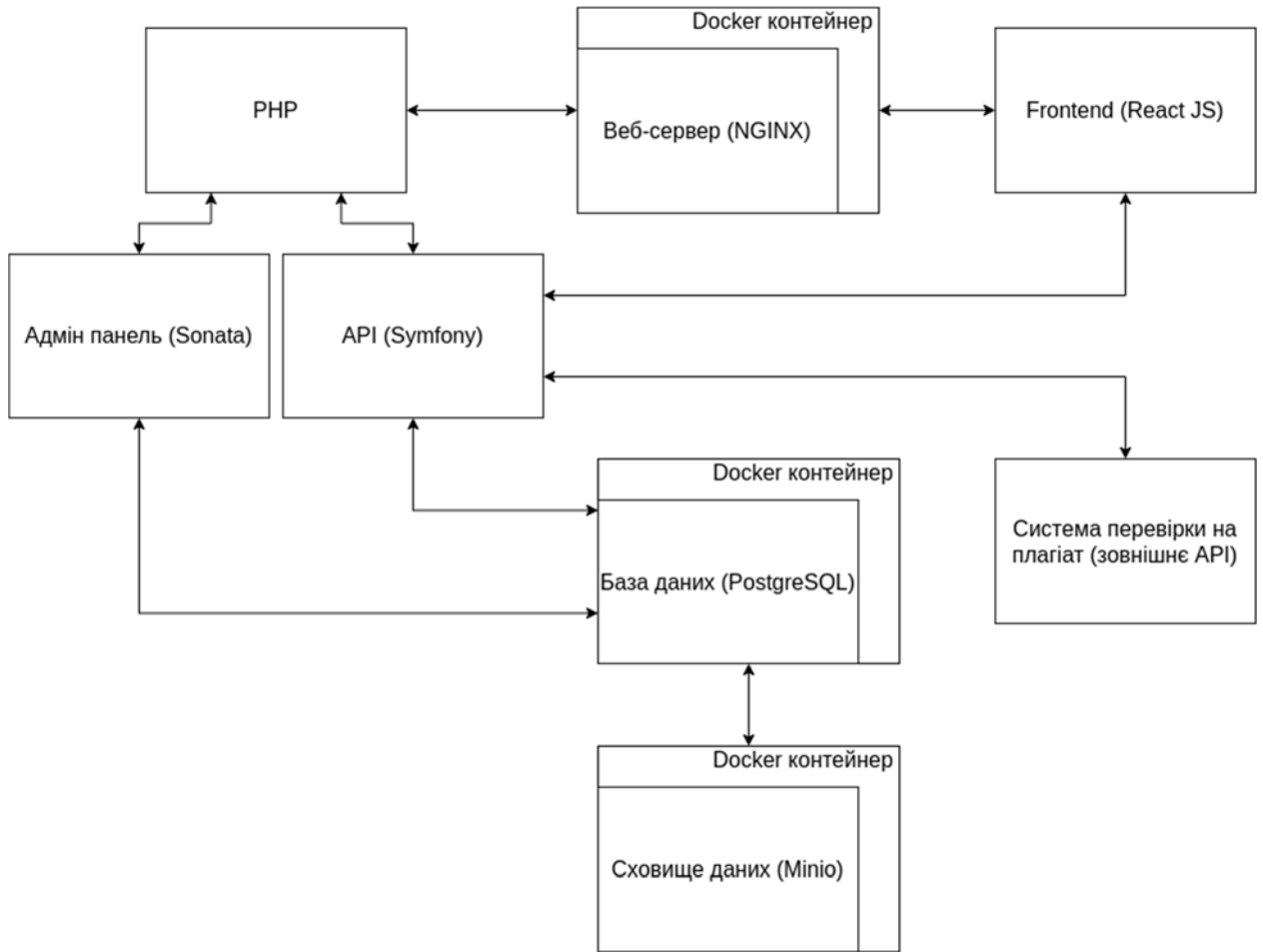
					ДП.6417.03.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

СТРУКТУРНА СХЕМА

до дипломного проекту

На тему «Система електронного забезпечення навчально-методичної діяльності вищого навчального закладу»

СТРУКТУРНА СХЕМА



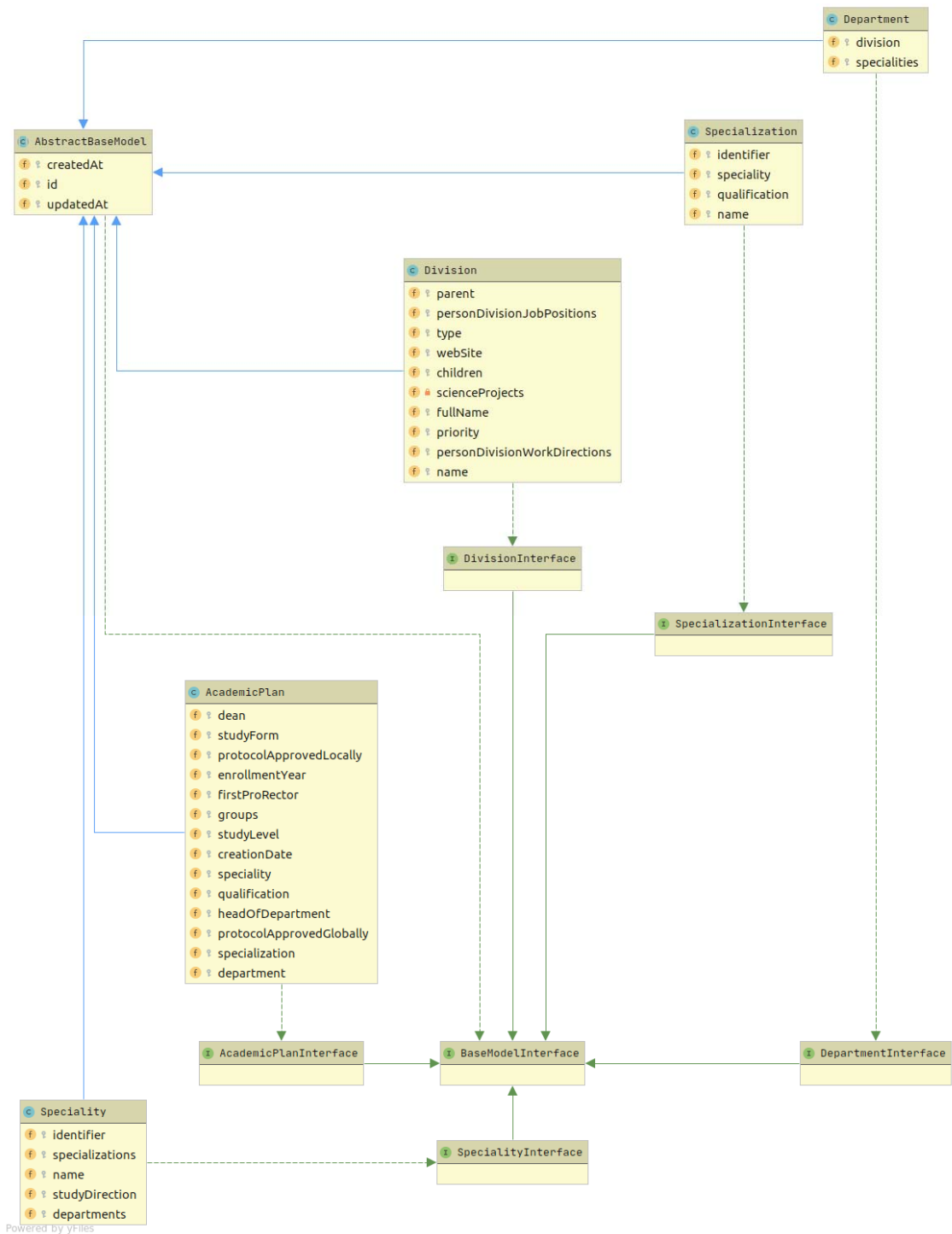
					ДП.6417.04.000 Д1			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розробив	Новосаденко В.І.				<i>Система електронного забезпечення навчально- методичної діяльності вищого навчального закладу</i> Додаток 1	Літ.	Аркуш	Аркушів
Перевірив	Ткаченко В.В.						1	1
Реценз.						НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ Група ІО-64		
Н. Контр.	Сімоненко В. П.							
Затвердив								

ФУНКЦІОНАЛЬНА СХЕМА

до дипломного проекту

На тему «Система електронного забезпечення навчально-методичної діяльності вищого навчального закладу»

ФУНКЦІОНАЛЬНА СХЕМА



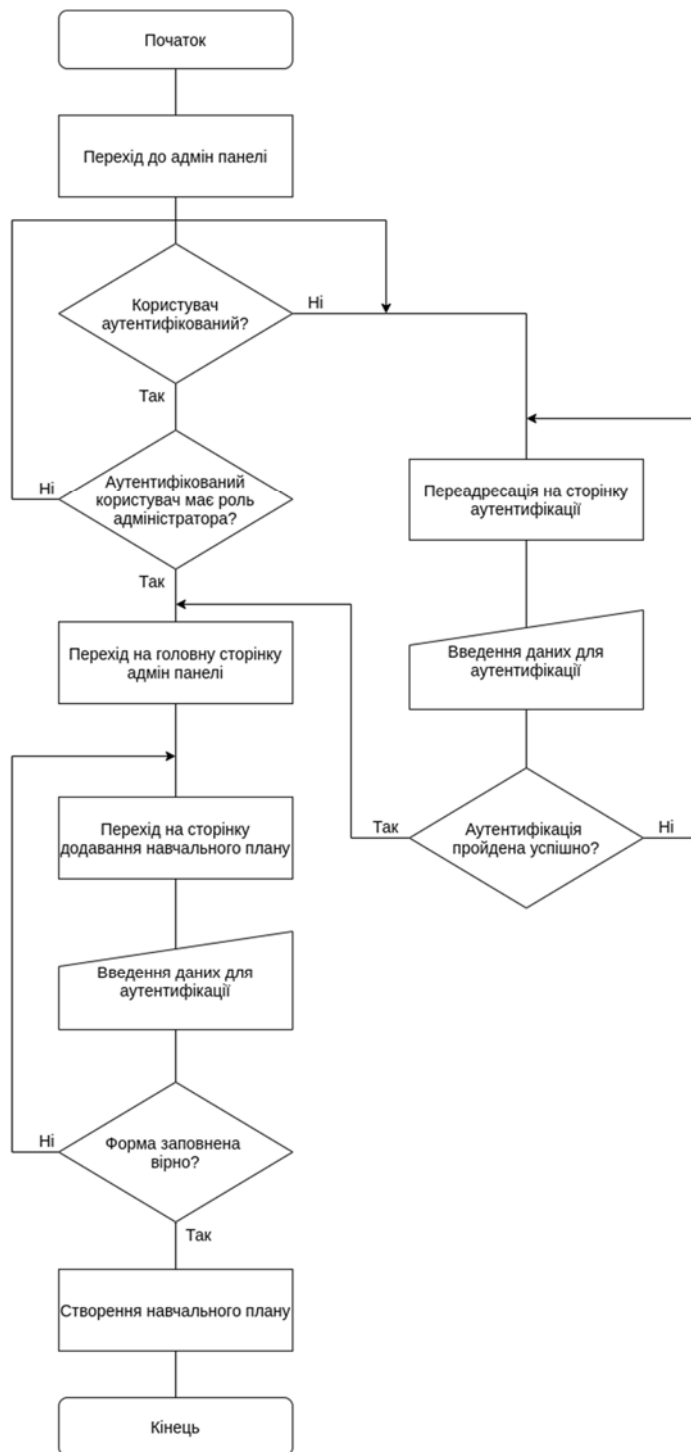
					ДП.6417.05.000 Д2						
Зм.	Арк.	№ докум.	Підпис	Дата	Система електронного забезпечення навчально-методичної діяльності вищого навчального закладу Додаток 2						
Розробив	Новосаденко В.І.								Літ.	Аркуш	Аркушів
Перевірив	Ткаченко В.В.									1	1
Реценз.									НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ Група ІО-64		
Н. Контр.	Сімоненко В. П.										
Затвердив											

ПРИНЦИПОВА СХЕМА

до дипломного проєкту

На тему «Система електронного забезпечення навчально-методичної діяльності вищого навчального закладу»

ПРИНЦИПОВА СХЕМА



					ДП.6417.06.000 ДЗ						
					<i>Система електронного забезпечення навчально-методичної діяльності вищого навчального закладу</i> Додаток 3						
Зм.	Арк.	№ докум.	Підпис	Дата							
Розробив	Новосаденко В.І.								Літ.	Аркуш	Аркушів
Перевірив	Ткаченко В.В.									1	1
Реценз.									НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ Група ІО-64		
Н. Контр.	Сімоненко В. П.										
Затвердив											