

Київський національний торговельно-економічний університет

Кафедра програмної інженерії та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка інформаційно-аналітичної моделі діяльності деканату (модуль складання звітів та створення інтерфейсів користувачів)»

Студента 2м курсу, 5 групи,
спеціальності 121«Інженерія
програмного забезпечення»,
спеціалізації «Інженерія
програмного забезпечення»

підпис студента

Прокоп'єва Андрія
Костянтиновича

Науковий керівник
кандидат технічних наук,
доцент

підпис керівника

Харченко Олександр
Анатолійович

Гарант освітньої програми
доктор технічних наук,
професор

підпис гаранта

Криворучко Олена
Володимирівна

КИЇВ – 2019

Київський національний торговельно-економічний університет

Факультет ФОАІС Кафедра «Програмної інженерії та кібербезпеки»

Спеціальність 121 «Інженерія програмного забезпечення»

Спеціалізація «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри

_____ Криворучко О.В.

«__» _____ 2019р.

Завдання

на випускн у кваліфікаційну роботу студенту

Прокоп'єву Андрію Костянтиновичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи «Розробка інформаційно-аналітичної моделі діяльності деканату (модуль складання звітів та створення інтерфейсів користувачів)»

Затверджена наказом ректора від «07» листопада 2018 р. №4183

2. Строк здачі студентом закінченої роботи _____

3. Цільова установка та вихідні дані до роботи

Мета роботи – розробка модулю складання звітів та інтерфейсу інформаційно-аналітичної моделі діяльності деканату.

Об'єкт дослідження – освітня діяльність деканату університету.

Предмет дослідження – методи та алгоритми створення програмного забезпечення інформаційно-аналітичної моделі діяльності деканату.

4. Перелік графічного матеріалу 20 рисунків.

5. Консультанти по роботі із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

6. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ОГЛЯД ОРГАНІЗАЦІЇ ОСВІТНЬОГО ПРОЦЕСУ В КНТЕУ ТА СИСТЕМИ УПРАВЛІННЯ У ДІЯЛЬНОСТІ ДЕКАНАТУ

- 1.1. Організація освітнього процесу в КНТЕУ
- 1.2. Дослідження організаційної структури КНТЕУ
- 1.3. Аналіз наявних програмних засобів управління в діяльності деканату
- 1.4. Висновки та пропозиції до розділу 1

РОЗДІЛ 2. ПРОЕКТУВАННЯ МОДУЛЯ СКЛАДАННЯ ЗВІТІВ ТА КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ

- 2.1. Постановка задачі по створенню звітів
- 2.2. Проектування модуля складання звітів
- 2.3. Аналіз груп користувачів та виділення ролей в інтерфейсі
- 2.4. Методи створення інтерфейсу
- 2.5. Висновки та пропозиції до розділу 2

РОЗДІЛ 3. РОЗРОБКА МОДУЛЯ СКЛАДАННЯ ЗВІТІВ ТА ІНТЕРФЕЙСУ КОРИСТУВАЧІВ

- 3.1. Визначення основних термінів framework bootstrap
- 3.2. Можливості bootstrap
- 3.3. Інструменти bootstrap
- 3.4. Контейнери framework bootstrap

3.5. Ряди та блоки framework bootstrap

3.6. Початкова розмітка сторінки

3.7. Розробка структури інтерфейсу

3.8. Розробка стилю інтерфейсу системи

3.9. Виділення модулів для розподілу за ролями користувачів

3.10. Розробка модуля складання звітів

3.11. Висновки та пропозиції до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускної кваліфікаційної роботи</i>	25.09.2018	
2.	<i>Розробка та затвердження завдання на випускну кваліфікаційну роботу (Наказ №4183 від 07.11.18)</i>	07.11.2018	
3.	<i>Вступ та перелік літературних джерел</i>	28.02.2019	
4.	<i>Розробка технічного завдання</i>	22.03.2019	
5.	<i>Розділ 1. Огляд організації освітнього процесу в КНТЕУ та системи управління у діяльності деканату</i>	18.04.2019	
6.	<i>Розділ 2. Проектування модуля складання звітів та користувацького інтерфейсу</i>	24.05.2019	
7.	<i>Розділ 3. Розробка модуля складання звітів та інтерфейсу користувачів</i>	20.09.2019	
8.	<i>Розробка програми та методики тестування</i>	18.10.2019	
9.	<i>Написання наукової статті</i>	27.09.2019	
10.	<i>Керівництво користувача</i>	23.10.2019	
11.	<i>Висновки та пропозиції</i>	01.11.2019	
12.	<i>Здача випускної кваліфікаційної роботи на кафедру (перша перевірка)</i>	13.11.2019	
13.	<i>Підготовка автореферату та презентації доповіді</i>	14.11.2019	
14.	<i>Попередній захист випускної кваліфікаційної роботи</i>	14.11.2019 – 15.11.2019	
15.	<i>Здача зброшурованої випускної кваліфікаційної роботи</i>	25.11.2019	
16.	<i>Зовнішнє рецензування випускної кваліфікаційної роботи</i>	26.11.2019	
17.	<i>Підготовка до публічного захисту випускної кваліфікаційної роботи</i>	02.12.2019– 04.12.2019	

7. Дата видачі завдання **«26» вересня 2018 р.**

8. Науковий керівник випускної кваліфікаційної роботи

(прізвище, ініціали, підпис)

9. Гарант освітньої програми

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент

(прізвище, ініціали, підпис)

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

$(nidnuc, \partial am a)$

(ПИБ, підпис, дата)

(прізвище, ініціали)

(прізвище, ініціали, підпис)

(підпис, прізвище, ініціали)

Анотація

Актуальність даної роботи полягає в тому, що в даний час деканату доводиться оперувати великою кількістю інформації, яку досить складно обробити вже наявними офісними пакетами, до того ж це буває досить незручно. Як наслідок, гостро постає питання автоматизації роботи, що надалі спричинить вивільнення певної кількості трудових ресурсів. Створення веб-додатків з кожним роком набуває все більшої популярності, адже спрямоване на полегшення роботи користувача та надає постійний доступ до потрібних даних, та значно економить час на саму реалізацію.

Метою роботи є створення інтерфейсу користувача та модуля звітів, які дають змогу використовувати функціонал програмного забезпечення та отримувати звітні матеріали в зручному вигляді.

Annotation

The urgency of this work is that in a given moment the dean has to prove a large number of informations, which he suffices to try already with the existing official officials. As a result, the question of the automation of the beat begins to emerge, which will subsequently release some of the number of rigid bins. The stepping-up of the dot-dot with the dot-pockets is getting a higher populapnocte, aga the tap-popping of the poppet of the poppet of the poppa

The purpose of the work is to create a user interface and a reporting module that allow you to use the functionality of the software and receive reporting materials in a convenient form.

Київський національний торговельно-економічний університет

**Інформаційно-аналітична модель діяльності
деканату**

ТЕХНІЧНЕ ЗАВДАННЯ

на 6 аркушах

1. Загальна інформація

1.1. Дане технічне завдання (надалі іменоване ТЗ) було розроблено відповідно до вимог ДСТУ 34.602-89 і є основним документом, що визначає вимоги та порядок проектування і розробки інформаційно-аналітичної моделі діяльності декана Київського національного торговельно-економічного університету.

1.2. Повна назва: Інформаційно-аналітична модель діяльності деканату.

1.3. Крайній термін: 07.11.2018 - 25.11.2019.

1.4. Допускається внесення змін в дане ТЗ за взаємною згодою сторін відповідно до ДСТУ 34.602-89.

2. Мета і призначення створення системи

Інформаційно-аналітична модель діяльності деканату – це інформаційна система, яка являє собою веб-додаток для автоматизованої обробки, зберігання та передачі інформації, отриманої за результатами діяльності деканату.

Метою системи є мінімізація документообігу в процесі діяльності деканату, скорочення часу виконання кожної операції шляхом надання всім учасникам навчального процесу оперативної, правильної та достовірної інформації.

3. Характеристика об'єкта автоматизації

Деканат університету.

4. Системні вимоги

4.1. Вимоги до системи в цілому

Система повинна передбачати обмін інформацією між учасниками освітнього процесу в режимі реального часу (при роботі безпосередньо в системі).

Клієнтами системи бути:

- співробітники деканату;
- співробітники кафедр;

4.2. Вимоги до функцій (завдань), що виконуються системою

4.2.1. Загальна функціональність

Система повинна дозволяти користувачам виконувати такі операції:

- ідентифікація, авторизація та аутентифікація в системі;

- отримання інформації з системи, що надається різними учасниками освітнього процесу, відповідно до прав доступу і ролей;
- надання інформації в систему у вигляді електронних документів;
- функціональність електронного архіву (термін зберігання документів – не менше 75 років).

За структурою додатків система повинна складатися з:

- клієнтської частини
- серверної частини

4.2.2. Вимоги до клієнтської частини

Клієнтська частина повинна реалізовувати наступні функціональні можливості:

- забезпечити доступ в реальному часі до системи користувача;
- забезпечити завдання формування призначеного для користувача інтерфейсу в системі;
- забезпечити виконання завдань CSIR (інтегрований захист інформації), застосовуючи різні групи доступу до інформації для різних користувачів, підключаючись до серверної частини через безпечні канали зв'язку.

4.2.3. Вимоги до серверної частини

На стороні сервера повинні бути реалізовані наступні функції:

- впровадження бізнес-логіки відповідних підсистем;
- забезпечення виконання завдання CSIS (комплексний захист інформації) шляхом застосування різних груп доступу до інформації для різних користувачів, ідентифікації, авторизації та аутентифікації користувачів системи, управління сесіями (сесіями) користувачів;
- забезпечення формування повідомлень клієнтської частини і прийом повідомлень від клієнтської частини відповідно до формату та порядку електронного документообігу в інформаційно-аналітичній системі.

4.3. Вимоги до кількості користувачів і обслуговуючого персоналу

Наступні можуть бути підключені До системи можуть бути підключені:

- співробітники деканатів – до 50 підключень;
- адміністрація університету – до 20 підключень;

- викладачі – до 500 підключень;
- здобувачі вищої освіти – до 5000 підключень.

Максимальна кількість транзакцій в системі (в день) – до 20 000.

Робота системи повинна забезпечуватися черговим персоналом цілодобово.

4.4. Вимоги до доступності

Система повинна гарантувати 99,95% стабільності (час простою 4,38 години на рік).

Якщо навантаження на систему перевищене (в залежності від кількості з'єднань), робота серверу буде призупинена. У цьому випадку після зниження навантаження в допустимих межах доступність послуги повинна бути відновлена.

Час на відновлення (TTR) не повинен перевищувати 2 години.

4.5. Вимоги до інформаційної безпеки

Система повинна містити комплексну інформаційну безпеку (CSIS).

Спеціалізоване програмне забезпечення системи повинно отримати експертний висновок ДСТЗІ на відповідність вимогам щодо захисту інформації, яка не містить державної таємниці.

CSIS повинна вирішувати такі завдання:

- безперервність інформації;
- доступ до інформації;
- захист інформації.

4.6. Вимоги до стандартизації та уніфікації

В процесі розробки системи необхідно керуватися наступним:

- Державні стандарти України (ДСТУ);
- нормативно-правові документи законодавчих і виконавчих органів;
- відповідні відомчі нормативно-методичні документи;
- міжнародні стандарти і правила.

4.7. Ліцензійні вимоги

Система повинна використовувати тільки ліцензійне програмне забезпечення.

Спеціалізоване програмне забезпечення, розроблене за рахунок власних ресурсів підрядника, має використовувати ліцензійні компоненти (бібліотеки) і засоби розробки.

4.8. Вимоги до програмного забезпечення

Програмне забезпечення (в подальшому іменується ПЗ) має включати загальне (або системне) програмне забезпечення і спеціалізоване програмне забезпечення.

Загальні вимоги до програмного забезпечення:

- всі сервери, що працюють в системі, працюють на операційних системах сімейства MS Windows або Linux, якщо інше не обумовлено в ТЗ;
- всі сервери та інше комп'ютерне обладнання системи інтегровані в єдину централізовану систему управління і моніторингу;
- СУБД PostgreSQL повинна використовуватися в якості бази даних;
- клієнтська частина системи повинна бути встановлена на персональному комп'ютері (ПК) під керуванням операційної системи MS Windows 32 або 64 bit або Linux.

4.9. Вимоги до обладнання.

Устаткування системи повинно мати сертифікати відповідності УкрСЕПРО.

Апаратний склад повинен відповідати вимогам до доступності та безпеки інформації.

Сервер повинен бути забезпечений двома вимикачами живлення з автоматичним перемиканням за допомогою AVR.

Серверні шафи повинні бути підключені до джерела живлення 220В через окремі автоматичні вимикачі.

Серверні шафи повинні бути оснащені джерелом безперебійного живлення.

Термін служби батареї ДБЖ повинен складати не менше 30 хвилин.

4.10. Додаткові вимоги

Окремі документи можуть бути узгоджені в процесі створення веб-додатків.

5. Склад і зміст робіт зі створення системи

5.1. Визначити системні групи користувачів

5.1.1. співробітник деканату;

5.1.2. викладач;

5.1.3. здобувач вищої освіти;

5.1.4. неавторизований користувач.

5.2. Перелік документів згідно ДСТУ 34.201, представлених після завершення відповідного етапу і етапів робіт

5.2.1. Системні інструкції користувача

6. Порядок контролю та приймання системи

6.1. Типи, склад, обсяги і методи випробувань системи та її компонентів

При розробці системи виконуються наступні типи тестів:

– тестування елементів системи;

– тестування системи в цілому;

Крім того, з впровадженням комплексної системи захисту інформації, додаткові тести виконуються відповідно до інструкцій DSTZI.

6.2. Загальні вимоги до приймання робіт поетапно

Після завершення відповідного етапу робіт формується комплект документації, наданої відповідно до пункту 5.2.

7. Вимоги до складу та змісту робіт з підготовки об'єкта автоматизації до розгортання системи

7.1. Умови експлуатації

Серверна кімната повинна бути обладнана двома незалежними блоками живлення від різних ТР і подана на AVR.

Підключення до каналів загального доступу має бути зарезервовано і реалізовано окремо. Кількість точок каналів зв'язку, що перекриваються, повинно бути мінімізовано.

Серверна кімната повинна мати систему кондиціонування, оснащена пожежною сигналізацією та системою газового пожежогасіння відповідно до ДБН В.2.2-20: 9.22.4.

Вхідні двері серверної повинні бути обладнані системою контролю доступу.

7.2. Організаційна структура.

Для роботи системи потрібна організація 24-годинної змінної Service Desk.

7.3. Експериментальна експлуатація та навчання

Навчання користувачів системи може проводитися як віддалено (онлайн, відеоінформація, друкована документація), так і на місці.

8. Вимоги до документації

Системна документація повинна бути виконана у відповідності до вимог ДСТУ-34 201.

Спеціалізовані елементи системи, що поставляються підрядними організаціями, повинні супроводжуватися відповідним комплектом документації, а також (коли це можливо) сертифікатами відповідності.

ЗМІСТ

ВСТУП	1
РОЗДІЛ 1. ОГЛЯД ОРГАНІЗАЦІЇ ОСВІТНЬОГО ПРОЦЕСУ В КНТЕУ ТА СИСТЕМИ УПРАВЛІННЯ У ДІЯЛЬНОСТІ ДЕКАНАТУ	2
1.1. Організація освітнього процесу в КНТЕУ	2
1.2. Дослідження організаційної структури КНТЕУ	3
1.3. Аналіз наявних програмних засобів управління в діяльності деканату	4
1.4. Висновки та пропозиції до розділу 1	5
РОЗДІЛ 2. ПРОЕКТУВАННЯ МОДУЛЯ СКЛАДАННЯ ЗВІТІВ ТА КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ	6
2.1. Постановка задачі по створенню звітів	6
2.2. Проектування модуля складання звітів	7
2.3. Аналіз груп користувачів та виділення ролей в інтерфейсі	12
2.4. Методи створення інтерфейсу	13
2.5. Висновки та пропозиції до розділу 2	17
РОЗДІЛ 3. РОЗРОБКА МОДУЛЯ СКЛАДАННЯ ЗВІТІВ ТА ІНТЕРФЕЙСУ КОРИСТУВАЧІВ	18
3.1. Визначення основних термінів framework bootstrap	18
3.2. Можливості bootstrap	19
3.3. Інструменти bootstrap	20
3.4. Контейнери framework bootstrap	20
3.5. Ряди та блоки framework bootstrap	21

					КНТЕУ 121– 05–26.МР			
					Розробка інформаційно-аналітичної моделі діяльності деканату (модуль складання звітів та створення інтерфейсів користувачів)	Стадія	Аркуш	Аркушів
						3	1	2
Зм	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.						
Керівник		Харченко О.А.						
Гарант		Криворучко О.В.			ЗМІСТ	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		
Розроб		Прокоп'єв А.К.						

3.6. Початкова розмітка сторінки	24
3.7. Розробка структури інтерфейсу	26
3.8. Розробка стилю інтерфейсу системи	36
3.9. Виділення модулів для розподілу за ролями користувачів	37
3.10. Розробка модуля складання звітів	38
3.11. Висновки та пропозиції до розділу 3	39
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	42
ДОДАТКИ	44

ВСТУП

Розробка інтерфейсної частини веб-додатку моделі діяльності деканату.-
Прокоп'єв А. 2 к 5м гр.

Актуальність: в даний час деканату доводиться оперувати великою кількістю інформації, яку досить складно обробити вже наявними офісними пакетами, до того ж це буває досить не зручно. Як наслідок, гостро постає питання автоматизації роботи, що надалі спричинить вивільнення певної кількості трудових ресурсів.

Створення веб-додатків з кожним роком набуває все більшої популярності, адже спрямоване на полегшення роботи користувача та надає постійний доступ до потрібних даних, та суттєво економить час на саму реалізацію. Тому використання веб-додатку для роботи як деканату, так і в інших сферах людської діяльності на даний момент являється актуальною темою.

Об'єкт дослідження: освітня діяльність деканату університету.

Предмет дослідження: розробка пропозицій щодо створення бази даних для веб-додатку моделі діяльності деканату.

Мета роботи: розробити модуль складання звітів та інтерфейс інформаційно-аналітичної моделі діяльності деканату.

Задачі дослідження: розробити інтерфейс користувача та модуль звітів, які дають змогу користувачеві використовувати функціонал програмного забезпечення та отримувати звітні матеріали в зручному вигляді.

Методи і технології розробки: аналіз алгоритмів пошуку, веб-програмування, технологія front-end розробки, візуальне і об'єктно-орієнтоване програмування, порівняння, спостереження, побудова програмного продукту на основі проаналізованих даних.

Результат роботи: Створення інтерфейсу та модуля звітів веб-додатку моделі діяльності деканату.

					КНТЕУ 121- 05-26.МР			
					Розробка інформаційно-аналітичної моделі діяльності деканату (модуль складання звітів та створення інтерфейсів користувачів)	Стадія	Аркуш	Аркушів
Зм	Аркуш	№ докум.	Підпис	Дата		Р1	4	43
Зав. каф.	Криворучко О.В.							
Керівник	Харченко О.А.							
Гарант	Криворучко О.В.							
Розроб	Прокоп'єв А.К.				Розділ 1. Огляд організації освітнього процесу в кнтеу та системи управління у діяльності деканату	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		

РОЗДІЛ 1. ОГЛЯД ОРГАНІЗАЦІЇ ОСВІТНЬОГО ПРОЦЕСУ В КНТЕУ ТА СИСТЕМИ УПРАВЛІННЯ У ДІЯЛЬНОСТІ ДЕКАНАТУ

1.1. Організація освітнього процесу в КНТЕУ

Відповідно до Положення про організацію освітнього процесу студентів метою освітнього процесу у КНТЕУ є підготовка конкурентоспроможних фахівців, забезпечення та набуття ними необхідних загальних та фахових компетентностей. Освітній процес у КНТЕУ базується на принципах активізації мобільності студентів і науково-педагогічних кадрів.

Навчання здійснюється за такими формами: очна (денна та вечірня); заочна (дистанційна).

Освітній процес у КНТЕУ здійснюється за такими формами: навчальні заняття; самостійна робота; практична підготовка; контрольні заходи.

Основною формою проведення навчальних занять у КНТЕУ є лекція. Проведення лекцій з навчальних дисциплін здійснюють науково-педагогічні працівники відповідної спеціальності. Іншою формою навчального заняття є лабораторне заняття. Під час лабораторного заняття, студент під керівництвом викладача особисто проводить натурні або імітаційні експерименти чи досліди. Лабораторне заняття проводиться зі студентами, кількість яких не перевищує, зазвичай, половини академічної групи. Практичне заняття – форма навчального заняття, під час якого викладач організовує детальний розгляд студентами окремих теоретичних положень навчальної дисципліни та формує вміння і навички їх практичного застосування шляхом виконання студентом поставлених завдань.

Семінарське заняття – форма навчального заняття, під час якого викладач організовує дискусію щодо попередньо визначених тем, до яких студенти готують тези виступів на підставі індивідуально виконаних завдань. Семінарські заняття проводять у формі бесіди, диспутів, брифінгів тощо.

					КНТЕУ 121– 05–26.МР			
					Розробка інформаційно-аналітичної моделі діяльності деканату (модуль складання звітів та створення інтерфейсів користувачів)	Стадія	Аркуш	Аркушів
Зм	Аркуш	№ докум.	Підпис	Дата		Р1	2	43
Зав. каф.	Криворучко О.В.							
Керівник	Харченко О.А.							
Гарант	Криворучко О.В.							
Розроб	Прокоп'єв А.К.				Розділ 1. Огляд організації освітнього процесу в кнтеу та системи управління у діяльності деканату	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		

Оцінювання результатів навчання – це сукупність організаційно-методичних у контрольних заходів щодо перевірки та оцінювання знань, умінь і навичок студентів, набуття ними фахових компетентностей з метою раціональної організації освітнього процесу та забезпечення якості освітньої діяльності у КНТЕУ відповідно до встановлених вимог. Оцінювання результатів навчання студентів передбачає проведення таких контрольних заходів: вхідний, поточний, модульний та підсумковий контролю, атестація. Результати навчання студентів оцінюються за 100-бальною шкалою.

1.2. Дослідження організаційної структури КНТЕУ

КНТЕУ – лідер у реформуванні вищої освіти України, першим у країні здійснив суттєве оновлення змісту освіти, впровадження новітніх навчальних технологій на базі програмних продуктів та глобальних інформаційних мереж. В університеті навчається більше 40 тис. студентів. Функціонують 6 факультетів: міжнародної торгівлі та права; економіки, менеджменту та психології; фінансів та банківської справи; обліку, аудиту та інформаційних систем; ресторанно-готельного та туристичного бізнесу; торгівлі та маркетингу. Здійснюється професійна підготовка фахівців за 42 бакалаврськими і 47 магістерськими програмами. Схематично організаційна структура університету зображена на рисунку 1.1.



Рисунок 1.1. Організаційна структура КНТЕУ

1.3. Аналіз наявних програмних засобів управління в діяльності деканату

На даний момент в університеті існує автоматизована система управління навчальним процесом. Наявна система має такі особливості:

- а. Інформація створюється у системі невеликою кількістю «активних» операторів. Користуватись же цією інформацією має широке коло її споживачів (керівний склад, працівники деканатів та кафедр, викладачі та студенти – кожен в межах своїх прав);
- б. Узгоджено форми документів для обліку та звітності внутрішньокафедральних та внутрішньофакультетських видів діяльності (навчально-методична, наукова, організаційно-виховна, профорієнтаційна та ін.);
- в. Програмний продукт для автоматизації управління навчальним процесом сумісний з існуючим апаратним і програмним забезпеченням з можливістю використання програми, як на окремому взятому комп'ютері, так і в локальній загальноуніверситетській мережі;
- г. Передбачено заходи захисту інформації від несанкціонованого доступу та від непередбачених втрат інформації: пошкоджень програми та інформаційної бази вірусами й іншими негативними факторами;

Існуючий програмний продукт має наступний функціонал:

- а. Пошук та створення академічних груп;
- б. Створення, коригування та видалення навчальних планів;
- в. Введення даних поточного та підсумкового модульного контролю;
- г. Робота з особовими картками студентів;
- д. Робота з картками порушень;
- е. Робота з наказами;
- ж. Формування структури університету;
- з. Робота з особовими картками викладачів;
- и. Робота з довідниками науково-педагогічної діяльності;
- к. Викладачі по кафедрам;

- л. Звіти викладачів та кафедр;
- м. Розподіл обов'язків викладачів;
- н. Попередній перегляд та друк документів.

Не зважаючи на те, що система дозволяє автоматизувати найоб'ємнішу частину роботи деканату, вона має недоліки. До недоліків наявного програмного продукту можна віднести:

- а. Застаріла технологія розробки;
- б. Повільна робота системи;
- в. Висока ймовірність відмови;
- г. Застарілий інтерфейс;
- д. Невелика кількість користувачів, які одночасно можуть працювати;

1.4. Висновки та пропозиції до розділу 1

В першому розділі було розглянуто організацію освітнього процесу в КНТЕУ, визначено за якими формами здійснюється навчання (очна, заочна). Освітній процес у КНТЕУ здійснюється за такими формами: навчальні заняття, самостійна робота, практична підготовка, контрольні заходи, що передбачає різні види його контролю.

Досліджено організаційну структуру університету та визначено, що в КНТЕУ навчається більше 40 тис. студентів. Функціонують 6 факультетів: міжнародної торгівлі та права; економіки, менеджменту та психології; фінансів та банківської справи; обліку, аудиту та інформаційних систем; ресторанно-готельного та туристичного бізнесу; торгівлі та маркетингу.

Проведено аналіз наявних програмних засобів управління в діяльності деканату та визначено, що хоч і існує програмний продукт, який автоматизує управління навчальним процесом, але він має ряд недоліків. Тому було прийнято рішення розробити програмне забезпечення, яке б не мало існуючих недоліків та дозволяло оптимізувати процес управління навчальним процесом.

РОЗДІЛ 2. ПРОЕКТУВАННЯ МОДУЛЯ СКЛАДАННЯ ЗВІТІВ ТА КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ

2.1. Постановка задачі по створенню звітів

Для створення звітів викладачів та звітів кафедр необхідно розробити форму з універсальним звітом. Під час відкриття форми на екрані повинно бути відображено форму «Звіти по викладачам та кафедрам». Під час вибору із списку назви факультету та кафедри у окремому полі буде відображено список викладачів відповідної кафедри, а також статистичні дані кадрового складу відповідної кафедри: кількість та відсоток докторів, професорів, кандидатів наук, доцентів тощо. Також на формі необхідно відображати форму вибору періоду, за який будується звіт з навчальної, методичної, наукової та організаційної роботи за будь-який проміжок часу. Повинно бути передбачено створення загального звіту з усіх видів науково-педагогічної діяльності викладацького складу. Для формування загального звіту необхідно вибрати один із режимів (по факультету, по кафедрі, по викладачу) і натиснути кнопку «Загальний звіт».

2.2. Технології аналізу інформації та побудови звітів

Для побудови звітів було обрано технологію JasperReports. JasperReports – це бібліотека звітів з відкритим кодом, яка дозволяє користувачам створювати ідеальні для користувачів звіти, які можна друкувати або експортувати у багатьох форматах, включаючи PDF, HTML та XLS. Аналіз даних за допомогою цієї бібліотеки виконується на основі даних БД та вхідних параметрів, які визначає користувач.

Побудова звітів відбувається за такими етапами життєвого циклу:

- а. Розробка і створення шаблону. Шаблон це XML-файл, складений за особливими правилами бібліотеки «JasperReports». За замовчуванням має

					КНТЕУ 121– 05–26.МР			
					Розробка інформаційно-аналітичної моделі діяльності деканату (модуль складання звітів та створення інтерфейсів користувачів)	Стадія	Аркуш	Аркушів
Зм	Аркуш	№ докум.	Підпис	Дата		P2	6	43
Зав. каф.		Криворучко О.В.						
Керівник		Харченко О.А.						
Гарант		Криворучко О.В.						
Розроб		Прокоп'єв А.К.			Розділ 2. Проектування модуля складання звітів та користувацького інтерфейсу	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		

розширення «jrxml». Може бути створений вручну в текстовому редакторі (як і будь-який інший XML-файл), а може бути створений за допомогою редактора шаблонів «iReport Designer»;

- б. Збірка / компіляція шаблону. Перевірка XML-файла шаблону, приведення його до типу «Java serialization data» і збереження результату у вигляді файлу з розширенням «jasper»;
- в. Заповнення звіту даними. Використовується зібраний на попередньому етапі шаблон і вибірка з джерела даних. Як джерело можуть виступати: JDBC, CALS Table Models, XML, XSL, CSV, JavaBeans, EJBQL, Hibernate. Заповнений шаблон зберігається в файлі з розширенням «jrprint»;
- г. Експорт звіту. Приведення його до необхідного читабельного формату (PDF, HTML, XLS, RTF, ODT, CSV, XML, XHTML, DOCX, XLSX).

Збереження проміжних результатів у вигляді файлів «jasper» і «jrprint» дозволяє при необхідності створювати звіт з цих точок, що заощадить час в процесі промислового використання системи. Наприклад, якщо вже є затверджений і зібраний шаблон звіту, то процес можна починати з етапу заповнення його даними, а якщо звіт з введеними даними потрібно зберегти в будь-якому іншому форматі, то відразу можна переходити до експорту.

2.2. Проектування модуля складання звітів

Оскільки програмний продукт базується на основі Spring, тому для розробки звітів було обрано бібліотеку звітів JasperReports. Перед початком проектування модуля звітів необхідно додати залежність від jasperreports до pom.xml:

```
<dependency>  
  <groupId>net.sf.jasperreports</groupId>  
  <artifactId>jasperreports</artifactId>  
  <version>6.4.0</version>  
</dependency>
```

Конструкції звітів визначені у файлах JRXML. Це звичайні XML-файли з певною структурою, які двигун JasperReports може інтерпретувати. Файли JRXML

потрібно зібрати, щоб механізм звітів міг заповнити їх даними. За це відповідає клас `JasperCompilerManager` :

```
InputStream employeeReportStream
```

```
= getClass().getResourceAsStream("/employeeReport.jrxml");
```

```
JasperReport jasperReport
```

```
= JasperCompileManager.compileReport(employeeReportStream);
```

Щоб при подальшому використанні не компілювати його кожного разу, необхідно зберегти його у файл:

```
JRSaver.saveObject(jasperReport, "employeeReport.jasper");
```

Найпоширеніший спосіб заповнення складених звітів - це записи із бази даних. Це вимагає, щоб у звіті містився запит SQL, який двигун буде виконувати для отримання даних. Необхідно змінити звіт, щоб додати запит SQL:

```
<jasperReport ... >
```

```
<queryString>
```

```
<![CDATA[SELECT * FROM EMPLOYEE]]>
```

```
</queryString>
```

```
...
```

```
</jasperReport>
```

На цьому етапі проектування необхідно чітко визначити які параметри повинні передаватися в запит, та розробити SQL-запит таким чином, щоб він був максимально гнучким та відповідав усім вимогам технічного завдання.

Після чого необхідно створити просте джерело даних:

```
@Bean
```

```
public DataSource dataSource() {
```

```
    return new EmbeddedDatabaseBuilder()
```

```
        .setType(EmbeddedDatabaseType.HSQL)
```

```
        .addScript("classpath:employee-schema.sql")
```

```
        .build();
```

```
}
```


Тепер можна заповнити звіт:

```
JasperPrint jasperPrint = JasperFillManager.fillReport(  
    jasperReport, null, dataSource.getConnection());
```

Оскільки другому аргументу передається нуль, звіт ще не отримує жодних параметрів.

Параметри корисні для передачі даних у механізм звітів, які він не може знайти у своєму джерелі даних або коли дані змінюються залежно від різних умов виконання. В процесі заповнення звіту може змінитися частина або весь запит SQL.

Спочатку необхідно змінити звіт, щоб отримати три параметри:

```
<jasperReport ... >  
    <parameter name="title" class="java.lang.String" />  
    <parameter name="minSalary" class="java.lang.Double" />  
    <parameter name="condition" class="java.lang.String">  
        <defaultValueExpression>  
            <![CDATA["1 = 1"]]></defaultValueExpression>  
        </parameter>  
    // ...  
</jasperreport>
```

Після чого необхідно додати розділ заголовка, щоб показати параметр заголовка:

```
<jasperreport ... >  
    // ...  
    <title>  
        <band height="20" splitType="Stretch">  
            <textField>  
                <reportElement x="238" y="0" width="100" height="20"/>  
                <textElement/>  
                <textFieldExpression class="java.lang.String">  
                    <![CDATA[$P{title}]]></textFieldExpression>
```

```

        </textField>
    </band>
</title>
...
</jasperreport/>

```

Далі необхідно змінити запит, щоб використовувати параметри minCourse та умови :

```

SELECT * FROM STUDENT
WHERE COURSE >= $P{minCourse} AND $P!{condition}

```

Після цього можна підготувати параметри та заповнити звіт:

```

Map<String, Object> parameters = new HashMap<>();
parameters.put("title", "Employee Report");
parameters.put("minSalary", 15000.0);
parameters.put("condition", " LAST_NAME ='Smith' ORDER BY FIRST_NAME");
JasperPrint jasperPrint
= JasperFillManager.fillReport(..., parameters, ...);

```

Ключі-параметри відповідають назвам параметрів у звіті. Якщо двигун bootstrap виявить, що параметр відсутній, він отримає значення з параметра defaultValueExpression (параметр, який передається за замовчуванням), якщо такий є.

Після того як розроблено безпосередньо формування звіту, необхідно його експортувати в формат, з яким зручно працювати користувачеві. Щоб експортувати звіт, спочатку необхідно спроектувати об'єкт класу-експортера, який відповідає потрібному файловому формату. Потім встановити попередній заповнений звіт як вхідний і визначити, куди виводити отриманий файл. Нижче наведені спроектовані приклади класів-експортерів:

а. Об'єкт класу-експортера PDF формату.

```

JRPdfExporter exporter = new JRPdfExporter();
exporter.setExporterInput(new SimpleExporterInput(jasperPrint));

```

```

exporter.setExporterOutput(
    new SimpleOutputStreamExporterOutput("employeeReport.pdf"));
SimplePdfReportConfiguration reportConfig
    = new SimplePdfReportConfiguration();
reportConfig.setSizePageToContent(true);
reportConfig.setForceLineBreakPolicy(false);
SimplePdfExporterConfiguration exportConfig
    = new SimplePdfExporterConfiguration();
exportConfig.setMetadataAuthor("baeldung");
exportConfig.setEncrypted(true);
exportConfig.setAllowedPermissionsHint("PRINTING");
exporter.setConfiguration(reportConfig);
exporter.setConfiguration(exportConfig);
exporter.exportReport();

```

б. Об'єкт класу-експортера XLS формату.

```

JRXlsxExporter exporter = new JRXlsxExporter();
// Set input and output ...
SimpleXlsxReportConfiguration reportConfig
    = new SimpleXlsxReportConfiguration();
reportConfig.setSheetNames(new String[] { "Employee Data" });
exporter.setConfiguration(reportConfig);
exporter.exportReport();

```

в. Об'єкт класу-експортера CSV формату.

```

JRCsvExporter exporter = new JRCsvExporter();
// Set input ...
exporter.setExporterOutput(
    new SimpleWriterExporterOutput("employeeReport.csv"));
exporter.exportReport();

```


Оскільки PDF формат кардинально відрізняється своєю структурою від реляційної форми звіту, його об'єкт класу-експортеру має достатньо складну будову. На відміну від PDF формату XLS та CSV формати зберігають в собі інформацію в табличному вигляді, так само як і в БД, тому їх об'єкти класів-експортерів мають значно менше програмного коду.

2.3. Аналіз груп користувачів та виділення ролей в інтерфейсі

Групи користувачів описують характеристики користувачів продукту, тобто людей, які використовують продукт. Для програмного продукту для університету на базі веб-додатку характеристики, визначені для груп користувачів, містять детальну інформацію про:

- а. Посадові обов'язки та завдання;
- б. Обладнання (персональний комп'ютер, ноутбук, планшет, смартфон);
- в. Частота використання (щодня, декілька разів на тиждень, щомісяця, під час сесії);
- г. Програмне забезпечення (операційна система, версія браузера);
- д. Демографія.

Джерела інформації при розробці груп користувачів для нового продукту включають загальні дослідження діяльності деканату та університету в цілому. Перше завдання при розробці груп користувачів – це визначення груп користувачів.

Аналізуючи користувачів програмного продукту в інтерфейсі можна виділити такі ролі:

- а. Неавторизований користувач;
- б. Студент;
- в. Викладач;
- г. Співробітник деканату;
- д. Декан;
- е. Проректор;
- ж. Адміністратор системи.

2.4. Методи створення інтерфейсу

Інтерфейс користувача – це простір, де відбувається взаємодія людина-машина. Це визначає, як користувач вводить інформацію та як система представляє результат. Існують різні способи взаємодії з комп'ютеризованими системами прийняття рішень.

Однак, який стиль користувацького інтерфейсу слід використовувати, потрібно вирішити до початку процесу проектування.

Розробник інтерфейсу може надати кілька послідовностей управління для управління або запуску програмного забезпечення, залежно від потреб користувача. Є шість домінуючих стилів користувацького інтерфейсу, які можна використовувати:

1. **Інтерфейс командного рядка.** Як випливає з назви, інтерфейс командного рядка передбачає використання команд для встановлення взаємодії користувача з програмною системою. Користувач вводить команду і система виконає її. Цей інтерфейс вимагає від користувача ввести команду, щоб сказати машині, що робити. Хоча такі інтерфейси були потужними, користувачеві насправді довелося б навчитися командам, щоб змусити систему працювати для них, що, безумовно, було обмежуючим.
2. **Інтерфейс меню.** Інтерфейс меню пропонує користувачам список функцій у вигляді варіантів. Випадаюче меню полегшує їм вибір функції, яку потрібно виконати. Перевага такого інтерфейсу полягає в тому, що користувачі не повинні вивчати команди, щоб використовувати програмну систему. Однак меню підходять для простих систем. Зі збільшенням складності користувачі вимагають вибрати кілька наборів меню. Навіть пункти в меню можуть мати підменю. Але коли справа стосується побудови DSS за допомогою інтерфейсу меню, для розробки інтерфейсу програмного забезпечення в довгостроковій перспективі потрібні величезні час та ресурси.
3. **Графічний інтерфейс** – це система інтерфейсів, яка дозволяє користувачам давати команди через видимі об'єкти. Користувач може вказувати або

торкатися зображень, значків чи символів, щоб виконати дію. Графічний користувацький інтерфейс орієнтований більше на мультимедіа, а не на текст.

4. **Інтерфейс запитання і відповіді.** Такий тип інтерфейсу дозволяє машині задавати питання, а користувачеві вводити відповіді на питання. Це перетворюється на діалог, коли користувач продовжує відповідати на запитання системи. Цей тип інтерфейсу є спробою спонукати взаємодію людини до людини через систему. Однак головна проблема виникає, коли відповіді, подані користувачем, неструктуровані, оскільки комп'ютер не розуміє неструктурованих відповідей.

5. **Голосовий інтерфейс користувача.** Як випливає з назви, це робить можливим взаємодію людини та машини через мовлення. Для управління машиною або змушення її виконувати дію потрібен людський голос. Зараз голосові користувацькі інтерфейси стали звичними.

6. **Сенсорний інтерфейс користувача** – це найпопулярніший і останній тип користувацького інтерфейсу. Він спирається на відчуття дотику і спрямовує систему виконувати обрану дію, коли користувач торкається певної візуалізації. У цифровому середовищі він використовується поряд із голосовими та графічними інтерфейсами користувачів.

Процес створення інтерфейсу можна розділити на три основні методи: створення схеми організації, створення організаційної структури та створення каркасів.

Створення схеми організації залежать від класифікації вмісту і способів створення взаємозв'язків між кожним елементом структури. Більшість вмісту можна класифікувати кількома способами. Схеми можна розділити на точні та відносні. Залежно від вмісту, інтерфейс може поєднувати схеми організації.

Точні схеми організації

Точні організаційні схеми об'єктивно розділяють інформацію на взаємовиключні розділи. Ці системи порівняно прості для архітекторів інформації,

для створення та категоризації вмісту всередині. Однак вони можуть бути проблемою для користувачів. Приклади точних організаційних структур включають:

- а. Алфавітні схеми використовують наш алфавіт для організації їх вмісту. Для успішного використання цього типу схем важливо, щоб маркування вмісту відповідало словам, які шукають користувачі. Іноді алфавітні схеми у вигляді індексу АЯ служать вторинними навігаційними компонентами для доповнення можливостей вмісту, організованих інакше;
- б. Хронологічні схеми впорядковують вміст за датою. Щоб ці схеми були успішними, має бути узгоджено питання про те, коли відбулася тема змісту;
- в. Географічні схеми впорядковують контент на основі місця. Якщо не існує прикордонних спорів, такий тип схеми досить простий у розробці та використанні.

Часто ці типи схем служать гарним додатковим способом навігації по сайту, який інакше організований. Наприклад, ви можете надати карту для відображення інформації або індекс АЯ, щоб перейти до тем, згрупованих головним чином за однією з суб'єктивних схем.

Відносні схеми організації

Відносні схеми організації класифікують інформацію таким чином, який може бути специфічним для організації або сфери. Хоча їх складно спроектувати, вони часто є більш корисними, ніж точні схеми організації. Коли архітектори інформації потребують часу, щоб розглянути ментальні моделі користувача та групувати вміст змістовно, ці схеми можуть бути досить ефективними у виробництві конверсій. Цей тип категоризації також може сприяти полегшенню навчання, допомагаючи користувачам зрозуміти та встановити зв'язки між фрагментами вмісту. Приклади суб'єктивних схем включають наступне:

- а. Тематичні схеми впорядковують контент на основі конкретної тематики;
- б. Схеми завдань організовують контент, враховуючи потреби, дії, питання або процеси, які користувачі доводять до конкретного контенту;

- в. Схеми аудиторії організовують контент для окремих сегментів користувачів. Схеми аудиторії можуть бути закритими або відкритими, тобто користувачі можуть переходити від однієї аудиторії до іншої. Цей тип схеми не викликає труднощів, якщо контент не піддається користувачам дуже легко визначити, до якої аудиторії вони належать і, можливо, не підходить для кількох профілів аудиторії;
- г. Метафорові схеми допомагають користувачам, пов'язуючи вміст із знайомими поняттями. Це використовується в дизайні інтерфейсів (папки, сміття тощо), але може спричинити проблеми при використанні в якості основної схеми організації сайту.

Проблеми створення гібридів

Незалежно від впровадження схем є свої переваги, оскільки вони просто корисні для користувача. Вони можуть визначити категоризацію та сформувані ментальну модель, яку можна швидко зрозуміти. Змішування схем шляхом створення гібридів може викликати плутанину у користувачів. Це часто пропонується як рішення, коли проектні команди не можуть домовитись про єдину схему для категоризації вмісту.

Іншим методом створення інтерфейсу є створення організаційної структури. Організаційна структура – це те, як визначаються взаємозв'язки між частинами змісту. Успішні структури дозволяють користувачам передбачати, де вони знайдуть інформацію на сайті. Важливо враховувати очікування користувачів та застосовувати послідовні методи організації та відображення інформації, щоб користувачі могли поширити свої знання зі знайомих сторінок на незнайомі. Три основні організаційні структури – це ієрархічна, послідовна та матрична.

У Ієрархічних структурах, які іноді називаються структурами дерев, між частинами інформації існує підхід зверху вниз або відносини батько / дитина. Користувачі починають з більш широких категорій інформації (батьківська), а потім детальніше вивчають структуру, щоб знайти більш вузьку та детальну інформацію (дочірня).

Інтерфейс з послідовними структурами вимагають від користувачів крок за кроком, слідуючи певному шляху через вміст.

Матрична структура дозволяє користувачам визначати свій власний шлях, оскільки вміст пов'язаний численними способами. Цей тип структури повністю використовує принципи гіпертексту або HTML.

Останній метод створення інтерфейсу це створення каркасів. Каркас є двовимірною ілюстрацією інтерфейсу сторінки, який спеціально фокусується на розподілі простору та визначенні пріоритетності вмісту, доступних функцій та передбачуваної поведінки. З цієї причини, каркаси, як правило, не включають стилів, кольорів чи графіки. Каркас також допомагає встановити зв'язки між різними шаблонами веб-сайту.

Каркаси слугують для різних цілей, допомагаючи:

- а. Підключення інформаційної архітектури до її візуального дизайну, показавши шляхи між сторінками;
- б. Уточнення послідовності способів відображення певних типів інформації в інтерфейсі користувача;
- в. Визначення функціоналу в інтерфейсі;
- г. Визначення пріоритету вмісту за допомогою визначення кількості місця для виділення певного елемента та місця його розташування.

2.5. Висновки та пропозиції до розділу 2

У розділі 2 було визначено постановку задачі, обрано технологію JasperReports, як основну технологію побудови звітів. Також було спроектовано універсальний звіт для користувача. Проаналізовано групи користувачів програмного продукту та виділено такі групи: неавторизований користувач, студент, викладач, співробітник деканату, декан, проректор, адміністратор системи. Детально розглянуто методи створення інтерфейсів.

РОЗДІЛ 3. РОЗРОБКА МОДУЛЯ СКЛАДАННЯ ЗВІТІВ ТА ІНТЕРФЕЙСУ КОРИСТУВАЧІВ

3.1. Визначення основних термінів framework bootstrap

Bootstrap (також відомий як Twitter Bootstrap) – вільний набір інструментів для створення сайтів і веб-додатків. Включає в себе HTML і CSS шаблони оформлення типографіки, веб-форм, кнопок, міток, блоків навігації та інших компонентів інтерфейсу, включаючи JavaScript-розширення. Bootstrap використовує найсучасніші напрацювання в області CSS і HTML, необхідно бути уважним при підтримці старих браузерів.

Основні переваги Bootstrap:

- а. Економія часу – Bootstrap дозволяє заощадити час і зусилля, використовуючи шаблони дизайну і класи;
- б. Висока швидкість – динамічні макети Bootstrap масштабуються на різні розширення екрану без будь-яких змін в розмітці;
- в. Гармонійний дизайн – всі компоненти платформи Bootstrap використовують стиль і шаблони за допомогою центральної бібліотеки;
- г. Простота у використанні – платформа проста у використанні, користувач із знаннями HTML і CSS може почати розробку з Bootstrap;
- д. Сумісність з браузерами – Bootstrap сумісний з Mozilla Firefox, Google Chrome, Safari, Internet Explorer і Opera;
- е. Відкрите програмне забезпечення – Bootstrap це фреймворк з відкритим вихідним кодом, який передбачає зручність використання, за допомогою відкритих вихідних кодів і безкоштовного завантаження;
- ж. Він також підтримує кастомізацію на веб-основі.

Bootstrap включає в себе:

					КНТЕУ 121– 05–26.МР			
					Розробка інформаційно-аналітичної моделі діяльності деканату (модуль складання звітів та створення інтерфейсів користувачів)	Стадія	Аркуш	Аркушів
Зм	Аркуш	№ докум.	Підпис	Дата		РЗ	18	43
Зав. каф.		Кривоучко О.В.						
Керівник		Харченко О.А.						
Гарант		Кривоучко О.В.						
Розроб		Прокоп'єв А.К.			Розділ 2. Проектування модуля складання звітів та користувацького інтерфейсу	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		

- а. Scaffolding – Bootstrap надає базову структуру з табличній системою Grid System стилями посилань, фоном;
- б. CSS – Bootstrap поставляється з функцією глобальних CSS налаштувань, основними HTML елементами, що мають стилі і поліпшення за допомогою розширюваних класів, і передову табличну систему;
- в. Components – Bootstrap містить більше десятка компонентів багаторазового використання, побудованих для забезпечення іконографії, випадаючих елементів навігації, сповіщень, спливаючих панелей popovers, і багато іншого;
- г. JavaScript Plugins – Bootstrap містить більше десяти користувацьких jQuery модулів, які можна підключити;
- д. Customize – дозволяє налаштувати Bootstrap компоненти, LESS змінні та jQuery плагіни, щоб отримати власну версію Bootstrap.

3.2. Можливості bootstrap

Bootstrap має безліч особливостей. Ці функції не тільки роблять його виділеним, але й роблять його більш популярним навіть серед тих веб-дизайнерів, які люблять сприймати речі дуже звичайним чином. Незважаючи на те, що Bootstrap розроблений у чутливих сітках, макетах та компонентах з 12 стовпцями, його легко налаштувати. Незалежно від того, чи потрібна вам фіксована сітка або чуйна мережа, це можна зробити, внісши кілька змін. Зміщення та вкладення стовпців також легко зробити як у сітках браузера на основі процесора, так і на мобільних пристроях.

Ще одна помітна особливість Bootstrap – це чутливі класи корисних програм. Використовуючи чутливі класи утиліти, певний вміст можна відобразити так, щоб він відображався або ховався на пристроях, в залежності від розміру екрана, який використовується. Ця функція надзвичайно корисна для розробників, які хочуть зробити мобільну та планшетну версію своїх веб-сайтів чи веб-додатків.

Деякі компоненти, що поставляються заздалегідь в Bootstrap:

- а. Випадання;

- б. Кнопка;
- в. Навігація;
- г. Повідомлення про значки;
- д. Індикатор виконання.

Випадаюче меню компонентів - функція адаптивного реагування веб-сайту. Для включення його в веб-додаток існує безліч різних плагінів, розроблених на основі Java. Але через Bootstrap та його легкі налаштування, це легко зробити за лічені хвилини.

3.3. Інструменти bootstrap

До основних інструментів Bootstrap можна віднести наступні:

- а. Сітки – заздалегідь задані розміри колонок, які можна відразу ж використовувати;
- б. Шаблони – фіксований або гумовий шаблон документа.
- в. Типографіка – опис шрифтів, визначення деяких класів для шрифтів, таких як цитати і т.п.;
- г. Медіа - представляє деякий спосіб упорядкування зображень та відео;
- д. Таблиці – засоби для оформлення таблиць, включаючи додавання функціоналу сортування;
- е. Форми - класи для оформлення форм і деяких подій, що відбуваються з ними;
- ж. Навігація – класи оформлення для табів, вкладень, сторінковості, меню і тулбара.

3.4. Контейнери framework bootstrap

Контейнер – це елемент сітки Bootstrap, з якого зазвичай починається створення макета сторінки або її частини. Тобто, це базовий елемент, в якому необхідно розміщувати всі інші елементи сітки (ряди і адаптивні блоки). Його основна мета – це встановити шаблоном ширину і вирівняти його по центру сторінки. Для фреймворка bootstrap існує два види контейнерів: адаптивно-фіксований та гумовий.

Адаптивно-фіксований контейнер призначений для створення контейнера з постійною шириною, яка буде залишатися постійною тільки в межах дії певної контрольної точки. Тобто, наприклад, на контрольній точці sm до дії контрольної точки md, він буде мати одну фіксовану ширину, а на md до дії lg- іншу фіксовану ширину. Єдина контрольна точка, на якій даний контейнер не матиме фіксовану ширину це breakpoint без позначення. Тут контейнер буде займати 100% ширину. Приклад оголошення контейнера: «<div class = "container" > ... </ div> ».

У горизонтальному напрямку контейнер розташовується по центру. Здійснюється це за допомогою установки йому CSS властивостей margin-left: auto і margin-right: auto в файлі «bootstrap.css».

Другий вид контейнера – це гумовий. Він застосовується тоді, коли вам необхідно створити повністю гнучкий макет цілої сторінки або її частини. Даний контейнер має на будь-яких контрольних точках 100% ширину. Приклад оголошення контейнера: «<div class = "container-fluid" > ... </ div> ». Крім цього, контейнери (.containerі .container-fluid) мають ще внутрішні відступи зліва і справа по 15px. Установка внутрішніх відступів контейнерів задається в «bootstrap.css» за допомогою CSS-властивостей padding-left: 15pxі padding-right: 15px.

3.5. Ряди та блоки framework bootstrap

Ряд – це елемент bootstrap сітки, який виступає в ролі безпосереднього контейнера для адаптивних блоків (інших елементів сітки. Це означає що ряд – це просто батьківський елемент для адаптивних блоків. Тобто адаптивні блоки завжди повинні бути розміщені безпосередньо в ньому. HTML-розмітка ряду: «<div class = "row" > ... </ div> ». Крім цього, адаптивні блоки логічно не пов'язані між собою не обов'язково поміщати в один ряд. Найбільш коректно їх розбити на окремі логічні групи і помістити кожну з них в окремий ряд (.row).

Адаптивні блоки – це основні будівельні елементи сітки. Саме від них буде залежати, як макет веб-сторінки буде відображатися на різних контрольних точках. Створення адаптивного блоку здійснюється за допомогою додавання одного або декількох класів col-?-? до необхідного HTML елементу, розташованого

безпосередньо в ряду. У класі col-?-? замість першого знака питання вказується назва контрольної точки (без позначення, sm, md, lg або xl). Замість другого знака питання вказується ширина адаптивного блоку, яку він повинен мати на зазначеній контрольній точці. Ширина адаптивного блоку задається у відносній формі за замовчуванням за допомогою числа від 1 до 12 (колонок Bootstrap). Дане число визначає, яку частину ширини буде займати адаптивний блок починаючи з зазначеної контрольної точки від ширини батьківського блоку, тобто ряду. При цьому ширина ряду в числовому виразі (колонках Bootstrap) за замовчуванням дорівнює 12.

Адаптивні блоки як обгортковий контейнери мають внутрішні відступи зліва і справа по 15px. Дані відступи адаптивним блокам в фреймворку Bootstrap встановлюються за допомогою CSS властивостей padding-left: 15px і padding-right: 15px. Розміщувати адаптивні блоки необхідно безпосередньо в ряд. Тобто у будь-якого адаптивного блоку як батько повинен бути обов'язково елемент .row.

У сітці Bootstrap є спеціальні класи col, col-sm, col-md, col-lg, col-xl, col-auto, col-sm-auto, col-md-auto, col-lg-auto і col-xl-auto. Перша група класів (col, col-sm, col-md, col-lg, col-xl) призначена для створення адаптивних блоків, ширина яких буде залежати від вільного простору в лінії. Розподіл незайнятої ширини (вільного простору) лінії між усіма такими блоками здійснюється рівномірно. Крім цього, дані адаптивні блоки перед розподілом вільного простору лінії (за замовчуванням) мають нульову ширину. Друга група класів (col-auto, col-sm-auto, col-md-auto, col-lg-auto і col-xl-auto) призначена для створення адаптивних блоків, ширина яких буде визначатися їх вмістом.

Адаптивні блоки в ряду за замовчуванням розташовуються горизонтальними лініями. В межах горизонтальної лінії адаптивні блоки шикуються послідовно зліва направо. В одну горизонтальну лінію можуть поміститися адаптивні блоки з сумарним числом колонок не більше 12. Адаптивні блоки, які не поміщаються в поточну лінію переходять на наступну.

Вирівнювання адаптивних блоків в горизонтальному і вертикальному напрямку здійснюється в Bootstrap за допомогою службових флекс-класів. Вирівнювання адаптивних блоків в межах лінії ряду по вертикалі здійснюється за допомогою одного з наступних класів, який необхідно додатково додати до row:

- а. align-items-start (Щодо початку лінії);
- б. align-items-center (по центру);
- в. align-items-end (Щодо кінця).

Для вирівнювання адаптивних блоків в горизонтальному напрямку призначені наступні класи:

- а. justify-content-start (Щодо початку лінії ряду - за замовчуванням);
- б. justify-content-center (По центру);
- в. justify-content-end (Щодо кінця лінії);
- г. justify-content-around (Рівномірно, з урахуванням простору перед першим і останнім адаптивним блоком);
- д. justify-content-between (Рівномірно, з однаковим простором між адаптивними блоками).

У Bootstrap 4 зміщення адаптивних блоків можна виконати за допомогою:

- а. класів offset (на певну кількість колонок);
- б. службових (утілітних) margin класів .

Класи offset призначені для зміщення адаптивних блоків вправо на певну кількість колонок. Дані класи мають наступний синтаксис: «proffset - { 1 } або offset - { breakpoint } - { 1 }». {breakpoint} – контрольна точка, починаючи з якої до даного блоку буде застосовано зсув (якщо не вказано, то зміщення буде застосовано, починаючи з самих крихітних пристроїв), {2} – величина зміщення, що вказується за допомогою кількості колонок Bootstrap.

У четвертій версії Bootstrap встановлювати зміщення адаптивним блокам також можна за допомогою margin відступів (margin-left: auto і (або) margin-right: auto). Цей варіант зміщення з'явився завдяки тому, що сітка в новій версії Bootstrap ґрунтується на CSS Flexbox. Даний варіант зміщення (за допомогою margin класів) використовують, коли блоки необхідно змістити один відносно одного на якусь

змінну величину. Для більш зручного і адаптивного завдання блокам `margin` відступів (`margin-left: auto` (або) `margin-right: auto`) можна використовувати класи `ml-auto`, `mr-auto`, `ml-{breakpoint}-auto` `mr-{breakpoint}-auto`.

3.6. Початкова розмітка сторінки

Розмітка сторінки розпочинається зі створення основного HTML файлу. Нижче приведено основну структуру HTML, яка буде використовуватися для проекту.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Bootstrap V4 template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Bootstrap -->
    <link href="bootstrap-3.0.0/dist/css/bootstrap.min.css"
rel="stylesheet" media="screen">

    <!-- HTML5 shim and Respond.js IE8 support of HTML5 elements and media
queries -->
    <!--[if lt IE 9]>
      <script src="bootstrap-3.0.0/assets/js/html5shiv.js"></script>
      <script src="bootstrap-3.0.0/assets/js/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="//code.jquery.com/jquery.js"></script>
    <!-- Include all compiled plugins (below), or include individual files
as needed -->
    <script src="bootstrap-3.0.0/dist/js/bootstrap.min.js"></script>
  </body>
</html>
```

Для підтримки IE8 в цей шаблон додані html5shiv.js та respo.min.js. Цей файл необхідно помістити в папку bootstrap-3.0.0 в папку twitter-bootstrap, яка розміщена в корені веб-сервера.

Для створення навігації необхідно додати наступний код у HTML-файл, відразу після відкриття тегу body.

```
<nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
  <ul class="nav navbar-nav">
    <li><a href="new.html" class="navbar-brand">
</a></li>
    <li class="active"><a href="#">Home</a></li>
    <li><a href="price.html">Price</a></li>
    <li><a href="contact.html">Contact</a></li>
    <li class="dropdown">
      <a href="#" class="dropdown-toggle" data-toggle="dropdown">Social<b
class="caret"></b></a>
      <ul class="dropdown-menu">
        <li class="socials"><g:plusone annotation="inline"
width="150"></g:plusone></li>
        <li class="socials"><div class="fb-like" data-
href="https://developers.facebook.com/docs/plugins/" data-width="The pixel
width of the plugin" data-height="The pixel height of the plugin" data-
colorscheme="light" data-layout="standard" data-action="like" data-show-
faces="true" data-send="false"></div></li>
        <li class="socials"><a href="https://twitter.com/share"
class="twitter-share-button">Tweet</a>
<script>!function(d,s,id){var
js,fjs=d.getElementsByTagName(s)[0];if(!d.getElementById){js=d.createEl
ement(s);js.id=id;js.src="//platform.twitter.com/widgets.js";fjs.parentNode
.insertBefore(js,fjs);}}(document,"script","twitter-wjs");</script></li>
      </ul>
    </li>
  </ul>
</nav>
```

Для навігації Bootstrap використовує клас "navbar" на рівні контейнера. Отже, він призначається елементу nav, який вміщує всю навігацію. Клас 'navbar-fix-top' гарантує, що navbar залишається фіксованим у верхньому положенні, коли ми прокручуємо нашу HTML-сторінку. Використання role = "navigation" є новим у Bootstrap V4 під час створення навігації. Bootstrap рекомендує використовувати це для навігації для доступу. У контейнері nav, знаходиться не упорядкований список з класами 'nav' та 'navbar-nav'. У цьому не упорядкованому списку, кожен елемент списку містить посилання в навігації.

Для лівого посилання додано спадне меню. Для цього клас 'dropdown' додається до асоційованого li, одразу після цього додається якір із двома класами 'dropdown-toggle' та 'caret'. Цей якір насправді містить текст якоря у проєкті. Потім у цьому li зберігається не упорядкований список у цьому списку, і знову кожен елемент списку цього вкладеного списку містить посилання, подане у спадному меню.

3.7. Розробка структури інтерфейсу

При вході на сайт неавторизованому користувачу доступна лише основна інформація, що включає в себе інформацію про факультети, кафедри та спеціальності. При відкритті головного вікна буде видно стартову сторінку з логотипом КНТЕУ (рис 3.1).

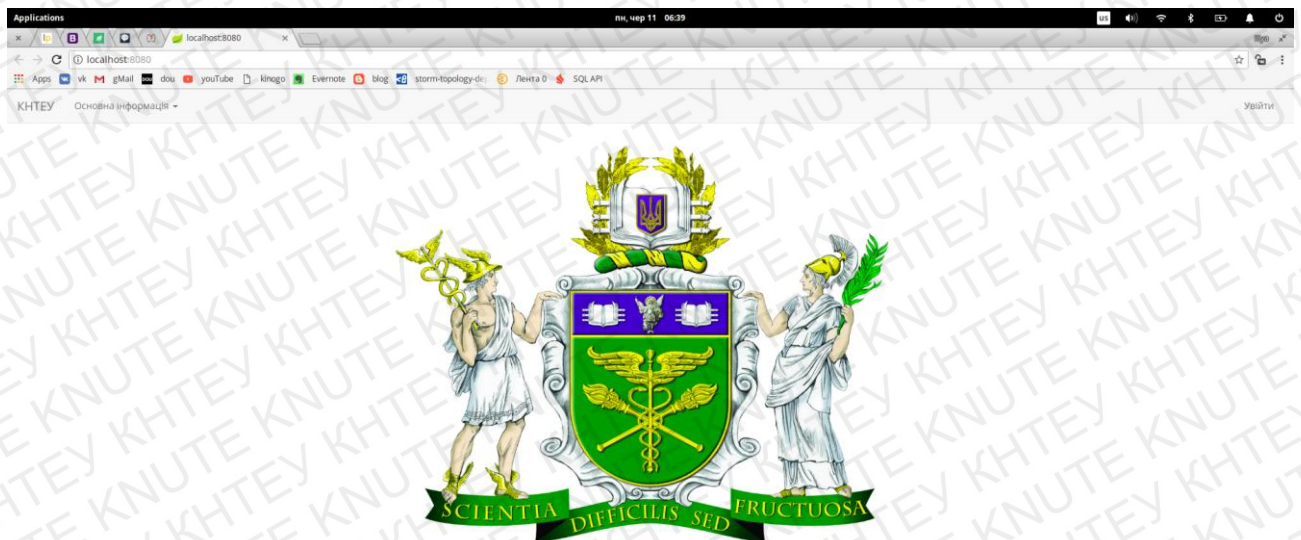


Рисунок 3.1. Стартова сторінка

Будь-який користувач має змогу подивитися основну інформацію, для цього у правому верхньому куті потрібно натиснути кнопку «Основна інформація», при натисканні на яку, відкриється ще кілька опцій (рис. 3.2), а саме:

- Спеціальності
- Факультети
- Кафедри

При натисканні на відповідну кнопку відкриється вікно зі спеціальностями (рис. 3.3), факультетами (рис. 3.4), або кафедрами (рис. 3.5). Слід зазначити, що спеціальності покажуться не лише ті, що є в КТНЕУ, а всі, які є в Україні. Також сторінка спеціальності створена у режимі пагінації, там перехід між іншими частинами здійснюється за допомогою нижніх кнопок сторінок.

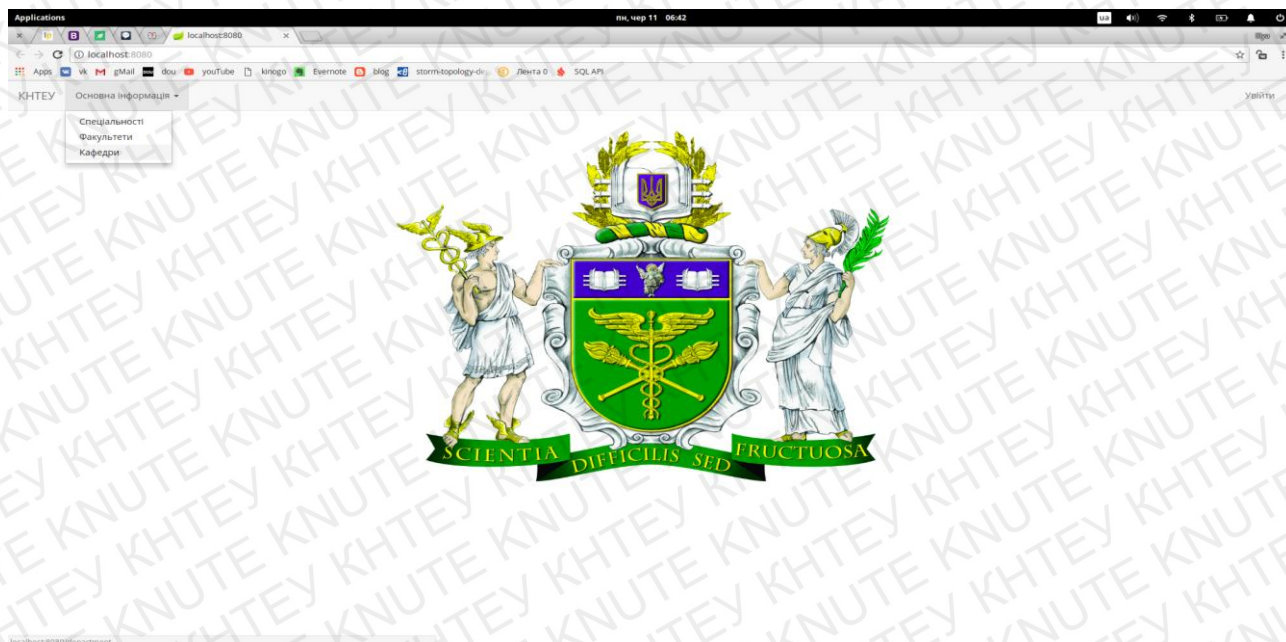


Рисунок 3.2. Меню «Основна інформація»

Шифр	Спеціальність	Галузь науки
35	Філологія	Гуманітарні науки
41	Богослов'я	Богослов'я
51	Економіка	Соціальні та поведінкові науки
52	Політологія	Соціальні та поведінкові науки
53	Психологія	Соціальні та поведінкові науки
54	Соціологія	Соціальні та поведінкові науки
55	Міжнародні відносини, суспільні комунікації та регіональні студії	Соціальні та поведінкові науки
56	Міжнародні економічні відносини	Соціальні та поведінкові науки
61	Журналістика	Журналістика
71	Облік і оподаткування	Управління та адміністрування
72	Фінанси, банківська справа та страхування	Управління та адміністрування
73	Менеджмент	Управління та адміністрування
74	Публічне управління та адміністрування	Управління та адміністрування
75	Маркетинг	Управління та адміністрування
76	Підприємництво, торгівля та біржова діяльність	Управління та адміністрування
81	Право	Право
82	Міжнародне право	Право
91	Біологія	Біологія
101	Екологія	Природничі науки
102	Хімія	Природничі науки

Рисунок 3.3. Вікно «Спеціальності»

Назва	Абревіатура	Код
Факультет Економіки, Менеджменту та Психології	ФЕМП	124
Факультет Ресторанно-Готельного та Туристичного Бізнесу	ФРГТБ	125
Факультет Торгівлі та Маркетингу	ФТМ	126
Факультет Фінансів та Банківської Справи	ФФБС	127
Факультет Міжнародної Торгівлі та Права	ФМТП	128
Факультет Обліку, Аудиту та Інформаційних Систем	ФОАІС	123

Рисунок 3.4. Вікно «Факультети»

Код кафедри	Назва	Факультет
16	Кафедра програмної інженерії та інформаційних систем	Факультет Обліку, Аудиту та Інформаційних Систем
19	Кафедра психології	Факультет Економіки, Менеджменту та Психології
17	Кафедра економічної кібернетики	Факультет Обліку, Аудиту та Інформаційних Систем

Рисунок 3.5. Вікно «Кафедри»

Після входу в систему, в залежності від ролі користувача, з'являється відповідний функціонал.

Студент окрім основної інформації має можливість дивитися ще свої оцінки та інформацію про свою групу, для цього йому потрібно авторизуватися під своїм логіном та паролем. Для авторизації потрібно перейти на вкладку авторизації. Для

цього потрібно натиснути на кнопку увійти у правому верхньому куті, після чого відкриється сторінка авторизації (рис. 3.6).

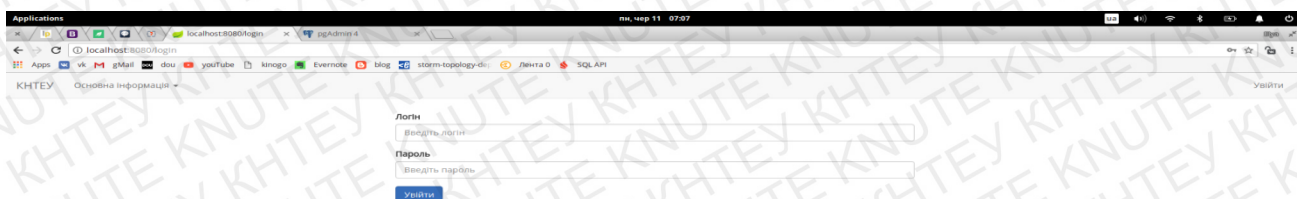


Рисунок 3.6. Вікно «Сторінка авторизації»

Після введення паролю потрібно натиснути кнопку увійти. У випадку невірною паролю система сповістить про це, як показано на рисунку 3.7.



Рисунок 3.7. Попередження про некоретне введення паролю

Після успішного вводу логіна та пароллю користувача перекине на головну сторінку, але з'являться додаткові кнопки на верхній панелі, що доступні студенту:

Мої оцінки

Моя группа

					КНТЕУ 121- 05-26.МР	Архив
						29
Зм.	Архив	№ докум.	Підпис	Дата		

При натисканні на кнопку мої оцінки відкриється сторінка з оцінками студента, що зайшов у систему за всі роки навчання в університеті, впорядковані по спаданню за семестрами навчання (рис 3.8).

Предмет	Тип оценивания	Модульная оценка	Оценка	Семестр	Начальный роки
Основы программирования	Экзамен	90	3	1	2014 - 2016
Программирование интернет	Экзамен	89	88	2	2014 - 2015
Математический анализ	Экзамен	87	90	1	2014 - 2015
Психология	Залік	75	78	1	2014 - 2015

Рисунок 3.8. Вікно з оцінками студента

При натисканні на кнопку моя група студент має змогу побачити інформацію про свою групу та всіх своїх одногрупників, упорядкованих у алфавітному порядку за прізвищем (рис 3.9).

Рис. 3.9. Інформація про академічну групу

Викладач, за допомогою даної системи може дивитися свої предмети та виставляти оцінки студентам.

При авторизації викладача перекине на основне меню з такою кнопкою у верхній панелі, як мої предмети, при натисканні на яку відкриється меню з предметами викладача (рис 3.10).

Назва	Кредити	Тип	Лекційні заняття(години)	Лабораторні заняття(години)	Практичні заняття(години)	Група	Факультет	Семестр	Роки	Виставити оцінки
Математичний аналіз	90	Екзамен	10	20	20	7	ФФАІС	1	2014 - 2015	Виставити
Психологія	12	Залік	20	20	0	7	ФФАІС	1	2014 - 2015	Виставити
Аналітична геометрія	90	Екзамен	20	20	20	12	ФФАІС	2	2014 - 2015	Виставити
Програмування інтернет	10	Екзамен	20	20	20	7	ФФАІС	2	2014 - 2015	Виставити

Назва	Кредити	Тип	Лекційні заняття(години)	Лабораторні заняття(години)	Практичні заняття(години)	Група	Факультет	Семестр	Роки	Виставити оцінки
Математичний аналіз	90	Екзамен	10	20	20	7	ФФАІС	1	2014 - 2015	Виставити
Психологія	12	Залік	20	20	0	7	ФФАІС	1	2014 - 2015	Виставити
Аналітична геометрія	90	Екзамен	20	20	20	12	ФФАІС	2	2014 - 2015	Виставити
Програмування інтернет	10	Екзамен	20	20	20	7	ФФАІС	2	2014 - 2015	Виставити
Основи програмування	120	Екзамен	20	24	24	7	ФФАІС	3	2015 - 2016	Виставити

Рисунок 3.10. Предмети

Викладач може побачити на яких предметах він є лектором, на яких асистентом, та виставити оцінки по цим предметам. Для того щоб виставити оцінку по предмету потрібно знайти потрібний предмет за курсом, групою та факультетом та натиснути кнопку виставити, після цього відкриється вікно зі студентами, щоб виставити їм оцінки (рис 3.11).

Прізвище	Ім'я	По-батькові	Предмет	Тип оцінювання	Модульна оцінка	Оцінка	Семестр	Навчальні роки
Прокоп'єв	Андрій	Костянтинович	Математичний аналіз	Екзамен	90	Змінити	94	Змінити
Слобожанін	Ілля	Іванович	Математичний аналіз	Екзамен	87	Змінити	90	Змінити

Рисунок 3.11. Вікно для виставлення оцінок

Для того щоб виставити оцінку потрібно у полі вводу модульної оцінки, або оцінки за залік або екзамен виставити відповідну оцінку та натиснути кнопку змінити. За замовчуванням, якщо оцінка не стоїть, то буде стояти 0. При натисканні кнопки змінити сторінка оновиться з новими даними про оцінку.

Найбільше привілеїв в системі має, звичайно, адміністратор, це може бути співробітник деканату, декан, та ін. Люди, які можуть змінювати дані про викладачів, студентів та ін.

При авторизації з роллю адміністратора, людина побачить у контекстному меню зверху основні функції, що може робити адміністратор (рис 3.12):

- а. Кафедри;
- б. Групи;
- в. Студенти;
- г. Викладачі;
- д. Предмети;
- е. Результати сесії.

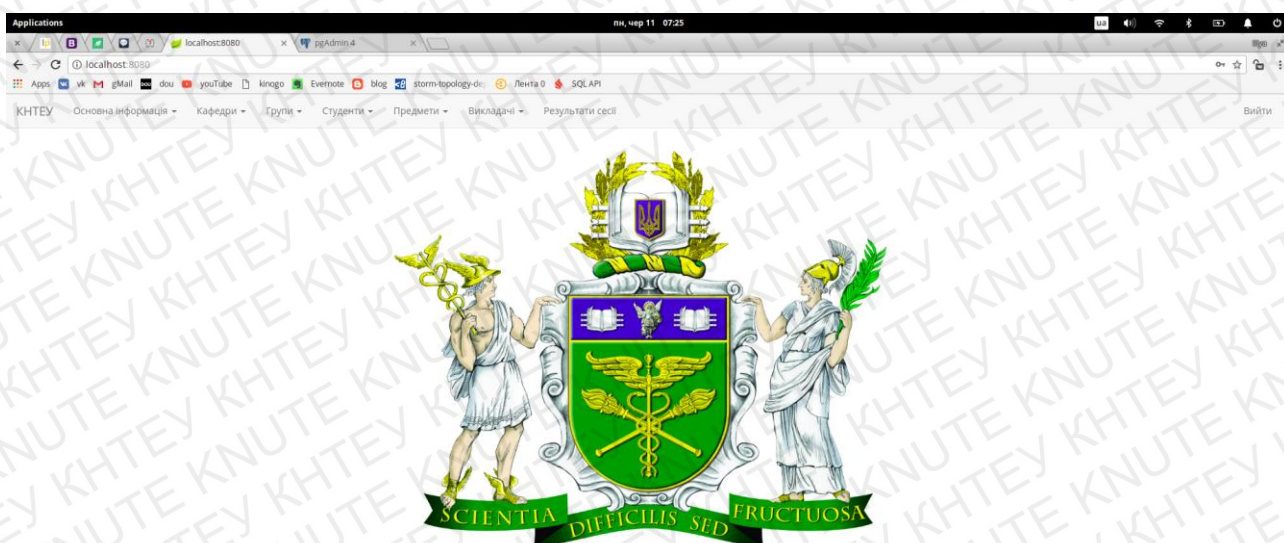


Рисунок 3.12. Можливості співробітника деканату

Співробітник деканату може додавати, редагувати та видаляти всі дані з бази даних. На рисунку нижче зображено вікно для додавання академічної групи до БД (рис.3.13).

The screenshot shows a web browser window with the URL `localhost:8080/group/add`. The page has a navigation bar with links: КНТЕУ, Соціальна інформація, Групи, Студенти, Предмети, Викладачі, Результати тестів, and Вайтс. The main form contains the following fields:

- Номер групи**: Введіть номер групи (text input)
- Рік набору**: Вибір року набору (dropdown menu)
- Спеціальність**: Наука про освіту (dropdown menu)
- Кваліфікація**: Бакалавр (dropdown menu)
- Форма навчання**: Дізнатися (dropdown menu)
- Кафедра**: Кафедра програмної інженерії та інформаційних систем (dropdown menu)
- Додати**: Blue button to submit the form.

Рис. 3.13. Інтерфейс додавання академічної групи

Також співробітнику деканату необхідно створювати особисті картки кожного студента, які містять інформацію про нього. Для цього реалізовано інтерфейс додавання студента в базу даних (рис.3.14).

Applications

localhost:3000/studentedit/3

КНТЕУ | Основна інформація | Групи | Студенти | Предмети | Викладачі | Результати сесії

Персональні дані студента

Прізвище:

Ім'я:

По-батькові:

Стать:

Група:

Дата народження:

Сім'йний статус:

Закордонка:

Телефон:

Адреса:

Місто:

Тип вулиці:

Рис. 3.14. Інтерфейс додавання студента

При необхідності співробітник деканату також може додати новий навчальний предмет, та призначити викладача, який буде проводити заняття. Інтерфейс додавання предмету зображено на рисунку 3.15.

Applications

localhost:3000/subjectadd

КНТЕУ | Основна інформація | Групи | Студенти | Предмети | Викладачі | Результати сесії

Додавання навчального предмету

Назва предмету:

Кредити:

Тип оцінювання:

Група:

Семестр:

Лекційні години:

Лабораторні години:

Практичні години:

Лектор:

Асистент:

Рисунок 3.15. Інтерфейс додавання навчального предмету

При необхідності додати до бази даних нового викладача є можливість зробити це у вікні додавання викладача (рис.3.16).

Персональні дані викладача

Прізвище:

Ім'я:

По-батькові:

Стать:

Телефон:

Дата народження:

Дата вступу в університет:

Кафедра:

Вчене звання:

Вчена ступінь:

Посада:

Рисунок 3.16. Інтерфейс додавання викладача

Також реалізовано можливість змінювати чи видаляти дані про студента. Під час видалення студента з'являється вікно для підтвердження видалення, для запобігання випадкового видалення (рис.3.17).

Прізвище	Ім'я	По-батькові	Стать	Номер групи	Факультет	Рік набору	Спеціальність	Форма навчання	Кваліфікація	Змінити	Видалити
Слобожанін	Ілля	Іванович	Чоловіча	7	ФІАІС	Інженерія програмного забезпечення	2014	Заочна	Бакалавр	Змінити	Видалити
Трохим'єв	Андрій	Костянтинович	Чоловіча	7	ФІАІС	Інженерія програмного забезпечення	2014	Заочна	Бакалавр	Змінити	Видалити
Ноздренко	Дмитро	Іванович	Чоловіча	12	ФІАІС	Науки про освіту	2014	Довча	Бакалавр	Змінити	Видалити

Рисунок 3.17. інтерфейс зміни та видалення даних про студента

3.8. Розробка стилю інтерфейсу системи

Елементи керування формою автоматично отримують деякі глобальні стилі за допомогою Bootstrap. Всі текстові `<input>`, `<textarea>` і `<select>` елементи з класом `.form-control` мають ширину 100%. Bootstrap пропонує три типи макетів форм:

1. Вертикальна форма (за замовчуванням);
2. Горизонтальна форма;
3. Вбудована форма.

Для початка необхідно створити вертикальну форму з двома полями введення, одним прапорцем і кнопкою подання.

```
<form action="/action_page.php">
  <div class="form-group">
    <label for="email">Email address:</label>
    <input type="email" class="form-control" id="email">
  </div>
  <div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" class="form-control" id="pwd">
  </div>
  <div class="checkbox">
    <label><input type="checkbox"> Remember me</label>
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

У вбудованій формі всі елементи вбудовані, розташовані ліворуч, а мітки розташовані поряд. Це стосується лише форм у вікнах перегляду, які мають ширину не менше 768 пікселів.

Після цього необхідно додати клас `.form-horizontal` до `<form>` елемента та клас `.control-label` до всіх `<label>` елементів для того, щоб розташувати елементи горизонтально.

```
<form class="form-horizontal" action="/action_page.php">
  <div class="form-group">
    <label class="control-label col-sm-2" for="email">Email:</label>
    <div class="col-sm-10">
      <input type="email" class="form-control" id="email" placeholder="Enter email">
    </div>
  </div>
```

```

</div>
</div>
<div class="form-group">
  <label class="control-label col-sm-2" for="pwd">Password:</label>
  <div class="col-sm-10">
    <input type="password" class="form-control" id="pwd" placeholder="Enter
password">
  </div>
</div>
<div class="form-group">
  <div class="col-sm-offset-2 col-sm-10">
    <div class="checkbox">
      <label><input type="checkbox"> Remember me</label>
    </div>
  </div>
</div>
<div class="form-group">
  <div class="col-sm-offset-2 col-sm-10">
    <button type="submit" class="btn btn-default">Submit</button>
  </div>
</div>
</form>

```

Після того, як елементи на формі розташовані правильно, необхідно задати кольорове наповнення форми: колір текстів, фону, рамок.

3.9. Виділення модулів для розподілу за ролями користувачів

Контроль доступу користувачів – одна з основних задач безпеки будь-якого програмного продукту. Тому, невід’ємною частиною роботи з будь-якою інформаційною системою є початкова реєстрація та подальша авторизація користувача. Процес авторизації включає в себе ідентифікацію та аутентифікацію. Ідентифікація – надання при вході в систему свого імені або реєстраційного номера, що одержується користувачем при попередній реєстрації в системі.

Аутентифікація – підтвердження права на доступ, відповідності наданої ідентифікаційної інформації користувача.

Процедура аутентифікації є обов'язковим етапом функціонування будь-якого програмного продукту та основною умовою доступу користувача до бази даних.

У веб-додатку виділено чотири групи користувачів: неавторизований користувач, студент, викладач, методист.

Група неавторизований користувач має доступ до переліку спеціальностей, факультетів та кафедр університету.

Група студент має такий самий доступ як і група неавторизований користувач, а також може переглядати свої оцінки та інформацію про свою групу.

Група викладач може дивитися свої предмети та виставляти оцінки студентам. Переглядати інформацію про те, на яких предметах він є лектор, а на яких асистентом.

Група методист має найбільше привілегій. Ця група має можливість вносити та редагувати інформацію про студентів та викладачів. Також ця група користувачів має управління над кафедрами, академічними групами, предметами, результатами сесії.

3.10. Розробка модуля складання звітів

У веб-додатку реалізовано можливість виконання звітів, які можна переглянути у відповідному вікні (рис.3.18) або ж зберегти у вигляді файлу з розширенням «.xls» для подальшої зручної роботи зі звітом поза межами програмного комплексу (рис.3.19).

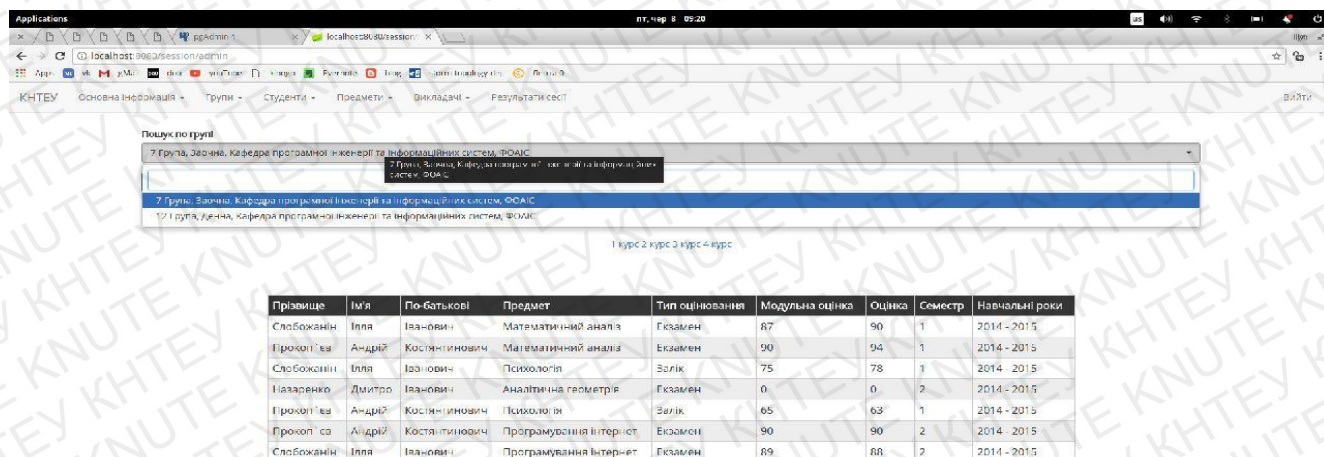


Рис. 3.18. Перегляд звіту у вікні веб-додатку

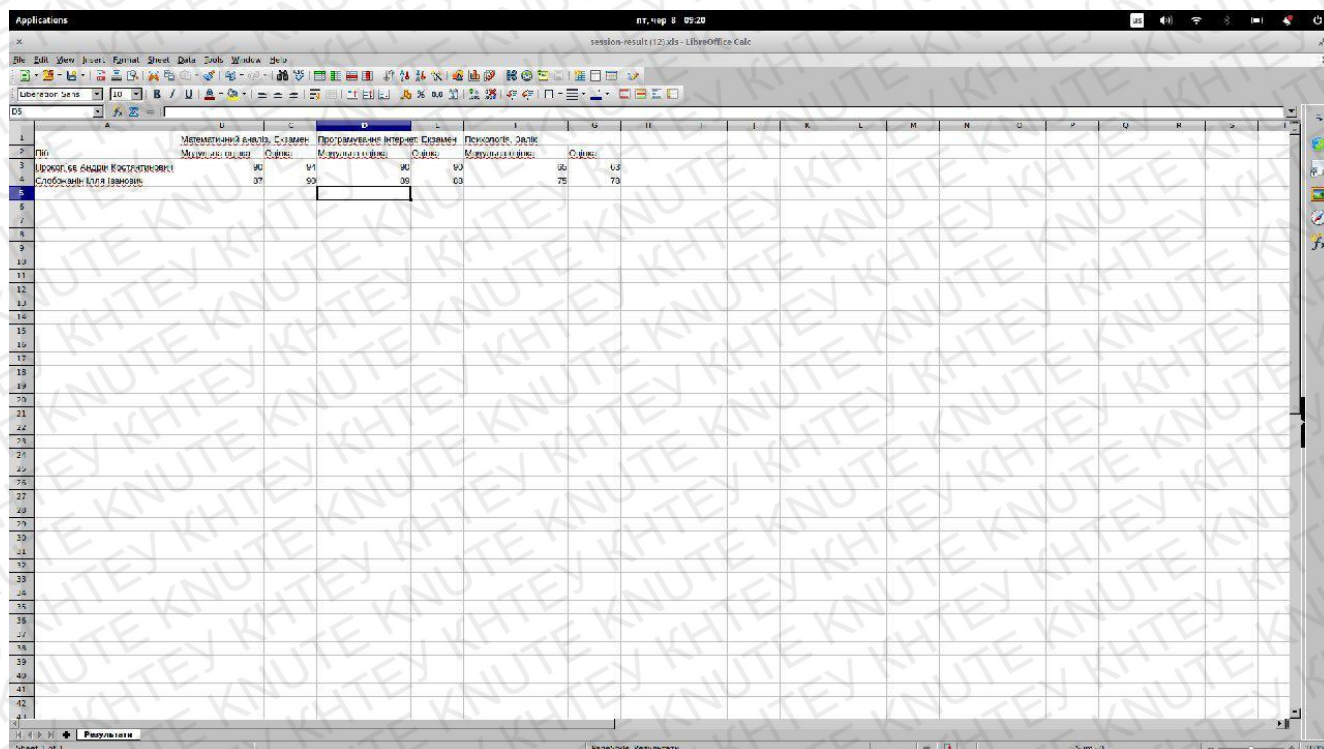


Рис. 3.19. Звіт, збережений у форматі «.xls»

3.11. Висновки та пропозиції до розділу 3

Отже, Bootstrap 4 – це відмінний інструмент для створення інтерфейсів веб-додатків. Він дає розробнику високоякісні компоненти, які легко настроюються і відмінно поєднуються один з одним. Крім того, Bootstrap дозволяє створювати гнучкі макети, які добре виглядають на пристроях з різними розмірами екранів.

Також у третьому розділі було виконано початкову розмітку сторінки, розроблено структуру та стиль інтерфейсу системи. Виділено чотири групи користувачів та розподілено їх в модулі. Також було розроблено та детально описано модуль складання звітів та інтерфейс всіх груп користувачів.

Виходячи з того, що інтерфейс є сучасним та зручним пропонується використання його для веб-додатку інформаційно-аналітичної моделі діяльності деканату.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Однією з найважливіших задач, які стоять перед ЗВО на сучасному етапі розвитку, постає ефективна організація процесів управління. Широкі можливості для вирішення цієї задачі надають сучасні інформаційні технології. Під час виконання випускної кваліфікаційної роботи, було розроблено інтерфейс веб-додатку управління діяльністю деканату, створений за допомогою фреймворку bootstrap. Даний фреймворк було обрано тому що він дозволяє створювати адаптивні інтерфейси, він є кросбраузерним, легким у використанні.

Описано можливості, інструменти, контейнери, ряди та блоки bootstrap. Описано технології аналізу інформації. Спроековано та розроблено модуль складання звітів. Проаналізовано групи користувачів та виділено відповідні їм ролі в інтерфейсі. Виконано повний цикл розробки користувацького інтерфейсу. Головними перевагами розробленого інтерфейсу є інтуїтивність, модульність та кросбраузерність.

					КНТЕУ 121– 05–26.МР			
					Розробка інформаційно-аналітичної моделі діяльності деканату (модуль складання звітів та створення інтерфейсів користувачів)	Стадія	Аркуш	Аркушів
Зм	Аркуш	№ докум.	Підпис	Дата		В	41	43
Зав. каф.		Криворучко О.В.						
Керівник		Харченко О.А.						
Гарант		Криворучко О.В.						
Розроб		Прокоп'єв А.К.			Висновки та пропозиції	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Організаційна структура КНТЕУ – Режим доступу: <https://knute.edu.ua/blog/read/?pid=5&uk/>.
2. Положення про організацію освітнього процесу студентів в КНТЕУ <https://knute.edu.ua/file/MjExMzA=/0ce40f69d25d44d4a2619ad857a2edd7.pdf>.
3. Van de Akker J.M., Hoogeveen J.A., van de Velde S.L. Parallel machine scheduling by column generation // Oper. Res.– 1999.– V. 47, N 6.– P. 862 – 872.
4. Van de Akker J.M., Hurkens C.A.J., Savelsbergh M.W.P. Timeindexed formulations for single-machine scheduling problems: column generation // INFORMS J. on Computing.– 2000.– V. 12, N 2.– P. 111 – 124.
5. Baptiste Ph., Le Pape C., Nuijten W. Constraint-based scheduling: applying constraint programming to scheduling problems // Kluwer Academic Publishers, 2001.– 198 p.
6. Brooks G.N., White C.R. An algorithm for finding optimal or near – optimal solutions to the production scheduling problem // J. Ind. Eng.– 1965.– V. 16, N 1.– P. 34 – 40.
7. Brucker P. Scheduling Algorithms. Germany: Springer-Verlag 2001. 365 p.
8. SiteMarker.Ru Інтернет технології. – Режим доступу: <http://sitemaker.ru/technologies/database/mysqlvspostgresql/>.
9. Bootstrap documentation – Режим доступу: <https://getbootstrap.com/docs/4.3/getting-started/introduction/>.
10. Bootstrap 4 Tutorial - Режим доступу: <https://www.w3schools.com/bootstrap4/>.
11. Benjamin J. Bootstrap 4 - Responsive Web Design / J. Benjamin.
12. JasperReports – Режим доступу: <https://uk.wikipedia.org/wiki/JasperReports>.
13. JasperReports® Librar – Режим доступу: <https://community.jaspersoft.com/project/jasperreports-library/>.
14. Короткий посібник зі створення звітів в JasperReports – Режим доступу: <http://ts-soft.ru/blog/jasperreports-quickstart/>.
15. JasperReports® - Free Java Reporting Library – Режим доступу: <https://github.com/TIBCOSoftware/jasperreports/>.

16. 50+ кращих доповнень до Bootstrap – Режим доступу:
<https://habr.com/ru/company/dataart/blog/258301/>.
17. Створення призначених для користувача компонент для Bootstrap 4 – Режим доступу: <https://habr.com/ru/post/416235/>.
18. Опенсорс-рішення для автоматизації звітності – Режим доступу:
<https://habr.com/ru/company/tinkoff/blog/461719/>.
19. Положення про особливості організації освітнього процесу у вищих навчальних закладах – Режим доступу:
<https://zakon.rada.gov.ua/laws/show/z1126-15/print>.

ДОДАТКИ

Додаток А

Програма та методики тестування

UI тестування – процес перевірки графічного інтерфейсу користувача на предмет його відповідності специфікаціям, загальним принципам і вимогам конкретного проекту.

Тестування користувацького інтерфейсу проводилося вручну, оскільки цей вид тестування має такі переваги:

- а. UI тестування покриває більшу частину дій користувача і дозволяє зрозуміти якість взаємодії потенційної аудиторії з веб-додатком;
- б. Це дає можливість на практиці перевірити взаємодію компонентів і сервісів між собою;
- в. Збільшується надійність програми, а знайдені недоліки можна перевірити ще до релізу.

Цей процес передбачає імітацію дій користувача – кліки на кнопки, переходи за посиланнями, та інші дії подібного плану. Таким чином перевіряється коректність роботи, взаємодія компонентів один з одним та зручність інтерфейсу в цілому.

До мінусів такого способу тестування можна віднести лише великі затрати часу безпосередньо на тестування.

Тестування інтерфейсу інформаційно-аналітичної моделі діяльності деканату проводилося за наступним планом:

1. Відповідність інтерфейсу прототипам.

На цьому етапі перевірялась візуальна відповідність положення елементів та відстаней між ними відповідно до макету проекту.

2. Перевірка типографіки.

Перевірено текст на читабельність на основному фоні та інших елементах на різних браузерах.

3. Відповідність стилю.

Перевірено чи всі елементи відповідають одній кольоровій гаммі, шрифти та інші стандартизовані елементи.

4. Адаптивність.

Перевірено як елементи інтерфейсу відображаються на екранах різних розмірів та на різних орієнтаціях.

5. Відповідність стандартам.

Перевірено інтерфейс на відповідність вимогам Human Interface Guidelines. Також перевірено чи інтерфейс правильно відображається в середовищах різних операційних систем.

6. Використання функціоналу.

Перевірено, що інтерактивні елементи інтерфейсу поведуться належним чином: кнопки реагують на натискання. Також перевірено чи всі елементи достатнього розміру, щоб користувач міг без проблем їх використовувати.

7. Перевірка полів та стандартних елементів.

Проведено тестування на те, як поля будуть поводитися при введенні некоретних даних, якщо вводиться довга назва, при виділенні даних. Також перевірено вигляд, положення і реакція чек-боксів, кнопок, полів.

8. Перевірка орфографії.

Перевірено текст на орфографічні помилки, оскільки через помилки в тексті чи назвах елементів користувачі можуть неправильно зрозуміти призначення того чи іншого елементу.

9. Елементи інформування.

Перевірено вигляд і розміщення всіх повідомлень про помилки, сповіщення тощо.

При проектуванні інтерфейсу веб-додатку головними цілями повинні бути користь та зручність для кінцевого користувача. Адже від того, наскільки хорошим він є, залежить чи те, буде людина користуватися додатком далі чи ні.

Керівництво користувача

При вході на сайт неавторизованому користувачу доступна лише основна інформація, що включає в себе інформацію про факультети, кафедри та спеціальності. При відкритті головного вікна буде видно стартову сторінку з гербом КНТЕУ (рис Б.1).

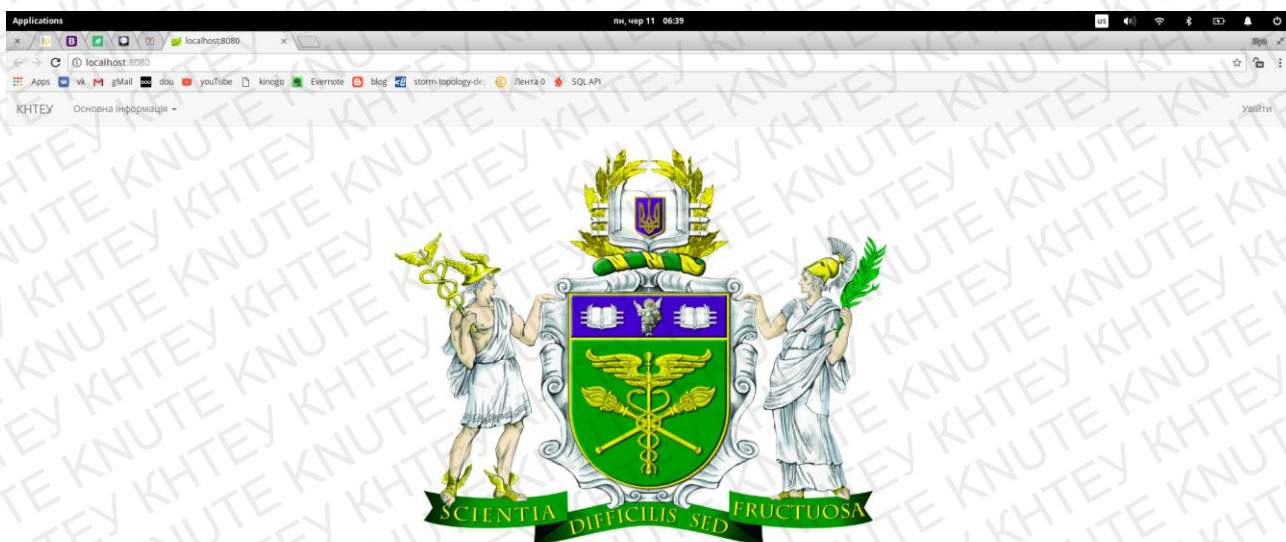


Рисунок Б.1. Стартова сторінка

У верхній правій частині екрану знаходиться кнопка Увійти. Під час натискання на неї відкривається форма, де необхідно ввести логін та пароль користувача для авторизації (рис. Б.2).

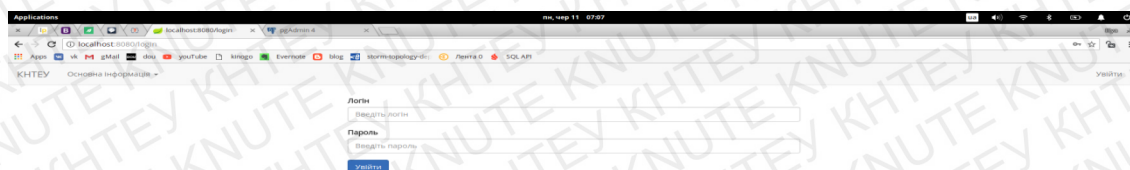


Рисунок Б.2. Вікно «Сторінка авторизації»

Після введення паролю потрібно натиснути кнопку Увійти. У випадку невірного паролю система сповістить про це, як показано на рис. Б.3.



Рисунок Б.3. Попередження про некоретне введення паролю

Після успішного вводу логіна та паролю користувача перекине на головну сторінку, але з'являться додаткові кнопки на верхній панелі.

Самостійно користувач не має змоги зареєструватися. Реєструвати нових користувачів може лише користувач з правами адміністратора (методист, декан або інша відповідальна особа).

Після успішного вводу логіна та паролю користувача перекине на головну сторінку, але з'являться додаткові кнопки на верхній панелі.

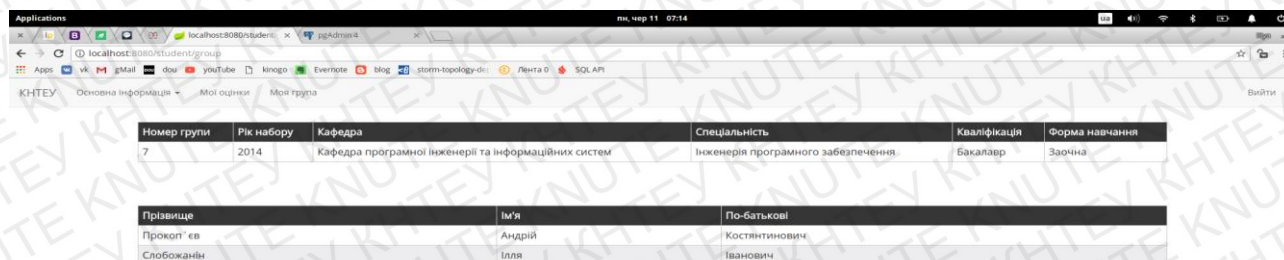
Якщо вхід здійснено з правами доступу студента, то з'являються дві додаткові кнопки. Для перегляду оцінок користувачеві необхідно натиснути на кнопку Мої оцінки. Після чого буде відображена сторінка з оцінками студента, що зайшов у систему за всі роки навчання в університеті, впорядковані по спаданню за семестрами навчання (рис Б.4).

A screenshot of a web browser window displaying a table of student grades. The table has six columns: 'Предмет' (Subject), 'Тип оцінювання' (Type of evaluation), 'Модульна оцінка' (Module grade), 'Оцінка' (Grade), 'Семестр' (Semester), and 'Навчальні роки' (Academic years). The table contains four rows of data for different subjects: 'Основні програмування', 'Програмування Інтернет', 'Математичний аналіз', and 'Психологія'.

Предмет	Тип оцінювання	Модульна оцінка	Оцінка	Семестр	Навчальні роки
Основні програмування	Екзамен	90	89	3	2015 - 2016
Програмування Інтернет	Екзамен	89	88	2	2014 - 2015
Математичний аналіз	Екзамен	87	90	1	2014 - 2015
Психологія	Залік	75	78	1	2014 - 2015

Рисунок Б.4. Вікно з оцінками студента

Для перегляду студентів академічної групи, в якій навчається студент передбачена кнопка Моя група. При натисканні на кнопку моя група студент має змогу побачити інформацію про свою групу та всіх своїх одногрупників, упорядкованих у алфавітному порядку за прізвищем (рис Б.5).

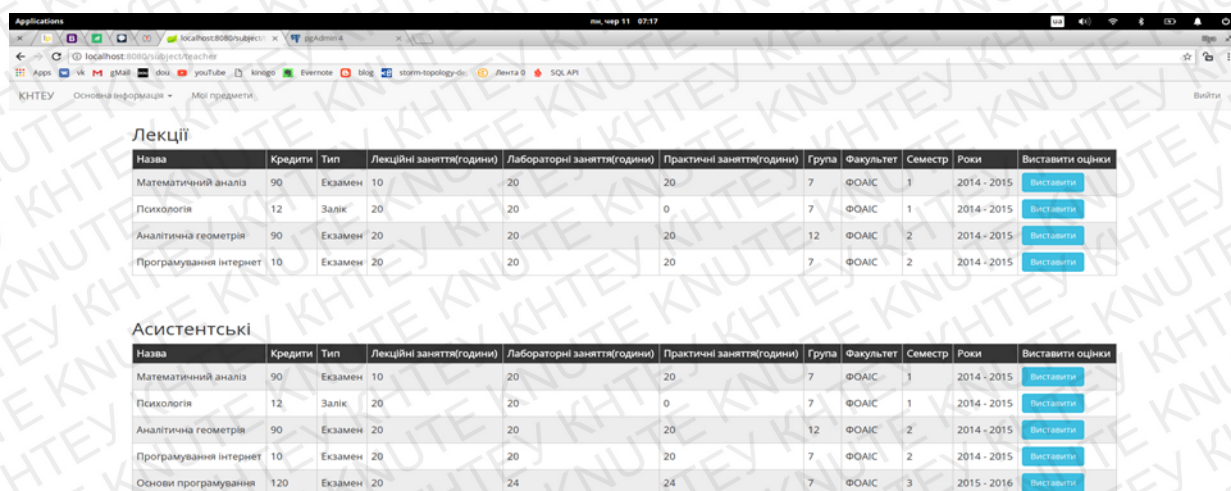


Номер групи	Рік набору	Кафедра	Спеціальність	Кваліфікація	Форма навчання
7	2014	Кафедра програмної інженерії та інформаційних систем	Інженерія програмного забезпечення	Бакалавр	Заочна

Прізвище	Ім'я	По-батькові
Прокоп'єв	Андрій	Костянтинович
Слобожанін	Ілля	Іванович

Рисунок Б.5. Інформація про академічну групу

Якщо вхід в систему виконав користувач з правами доступу викладач у верхній панелі буде відображена кнопка «Мої предмети», при натисканні на яку відкриється меню з предметами викладача (рис Б.6).



Лекції

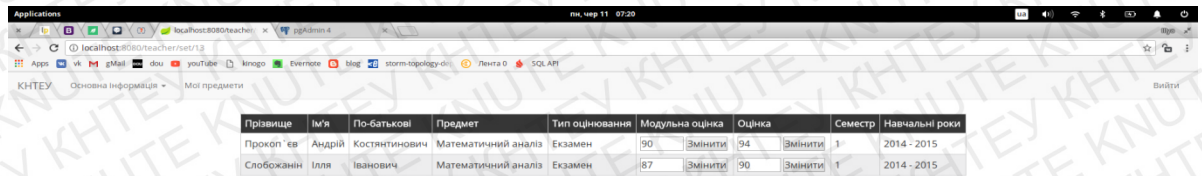
Назва	Кредити	Тип	Лекційні заняття(години)	Лабораторні заняття(години)	Практичні заняття(години)	Група	Факультет	Семестр	Роки	Виставити оцінки
Математичний аналіз	90	Екзамен	10	20	20	7	ФОАІС	1	2014 - 2015	Виставити
Психологія	12	Залік	20	20	0	7	ФОАІС	1	2014 - 2015	Виставити
Аналітична геометрія	90	Екзамен	20	20	20	12	ФОАІС	2	2014 - 2015	Виставити
Програмування Інтернет	10	Екзамен	20	20	20	7	ФОАІС	2	2014 - 2015	Виставити

Асистентські

Назва	Кредити	Тип	Лекційні заняття(години)	Лабораторні заняття(години)	Практичні заняття(години)	Група	Факультет	Семестр	Роки	Виставити оцінки
Математичний аналіз	90	Екзамен	10	20	20	7	ФОАІС	1	2014 - 2015	Виставити
Психологія	12	Залік	20	20	0	7	ФОАІС	1	2014 - 2015	Виставити
Аналітична геометрія	90	Екзамен	20	20	20	12	ФОАІС	2	2014 - 2015	Виставити
Програмування Інтернет	10	Екзамен	20	20	20	7	ФОАІС	2	2014 - 2015	Виставити
Основи програмування	120	Екзамен	20	24	24	7	ФОАІС	3	2015 - 2016	Виставити

Рисунок Б.6. Предмети

Викладач може побачити на яких предметах він є лектором, на яких асистентом, та виставити оцінки по цим предметам. Для того щоб виставити оцінку по предмету потрібно знайти потрібний предмет за курсом, групою та факультетом та натиснути кнопку виставити, після цього відкриється вікно зі студентами, щоб виставити їм оцінки (рис. Б.7).



Прізвище	Ім'я	По-батькові	Предмет	Тип оцінювання	Модульна оцінка	Оцінка	Семестр	Навчальні роки
Прокоп'єв	Андрій	Костянтинович	Математичний аналіз	Екзамен	90	94	3	2014-2015
Слобожанін	Ілля	Іванович	Математичний аналіз	Екзамен	87	90	1	2014-2015

Рисунок Б.7. Вікно для виставлення оцінок

Для того щоб виставити оцінку потрібно у полі вводу модульної оцінки, або оцінки за залік або екзамен виставити відповідну оцінку та натиснути кнопку змінити. За замовчуванням, якщо оцінка не стоїть, то буде стояти 0. При натисканні кнопки змінити сторінка оновиться з новими даними про оцінку.

Якщо вхід виконано користувачем з правами доступу співробітник деканату, то у верхній частині екрану з'являється меню зі всіма функціями веб-додатку (рис. Б.8).

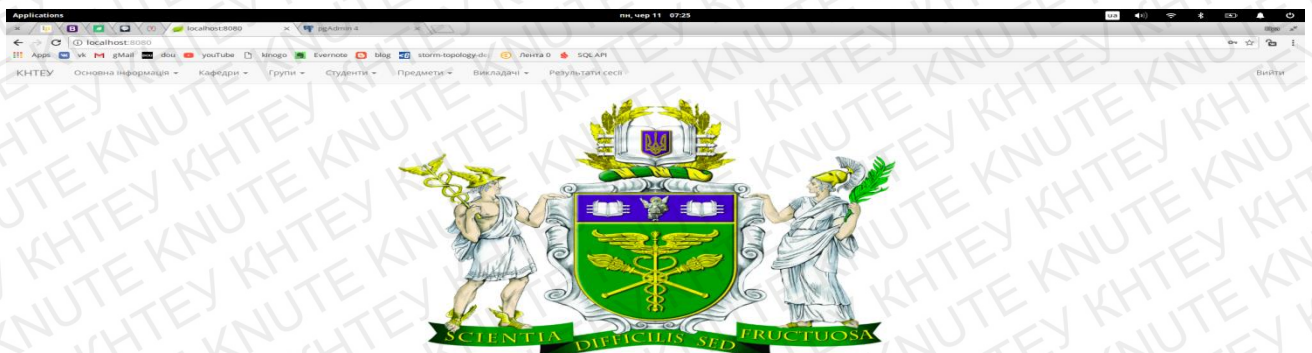


Рисунок Б.8. Можливості співробітника деканату

Методист може додавати, редагувати та видаляти всі дані з бази даних. Для додавання академічної групи необхідно натиснути на кнопку Групи, після чого відкриється вікно для додавання академічної групи до БД (рис. Б.9).

Applications

localhost:8080/group/add

КНТЕУ | Соціальна інформація | Групи | Студенти | Предмети | Викладачі | Результати сесії

Вхід

Номер групи
Введіть номер групи

Рік набору
Введіть рік набору

Спеціальність
Виберіть спеціальність

Кваліфікація
Виберіть кваліфікацію

Форма навчання
Виберіть форму навчання

Кафедра
Виберіть кафедру

Додати

Рисунок Б.9. Інтерфейс додавання академічної групи

Також методисту необхідно створювати особисті картки кожного студента, які містять інформацію про нього. Для цього реалізовано інтерфейс додавання студента в базу даних (рис. Б.10), який відкривається при натисканні на кнопку «Студенти».

Applications

localhost:8080/student/add

КНТЕУ | Соціальна інформація | Групи | Студенти | Предмети | Викладачі | Результати сесії

Вхід

Персональні дані студента

Прізвище
Введіть прізвище

Ім'я
Введіть ім'я

Поб-батьком
Введіть поб-батьком

Стать
Виберіть стать

Група
Виберіть групу

Дата народження
Введіть дату народження

Сімейний статус
Виберіть сімейний статус

Залікова
Введіть залікову

Телефон
Введіть телефон

Адреса

Місто
Введіть місто

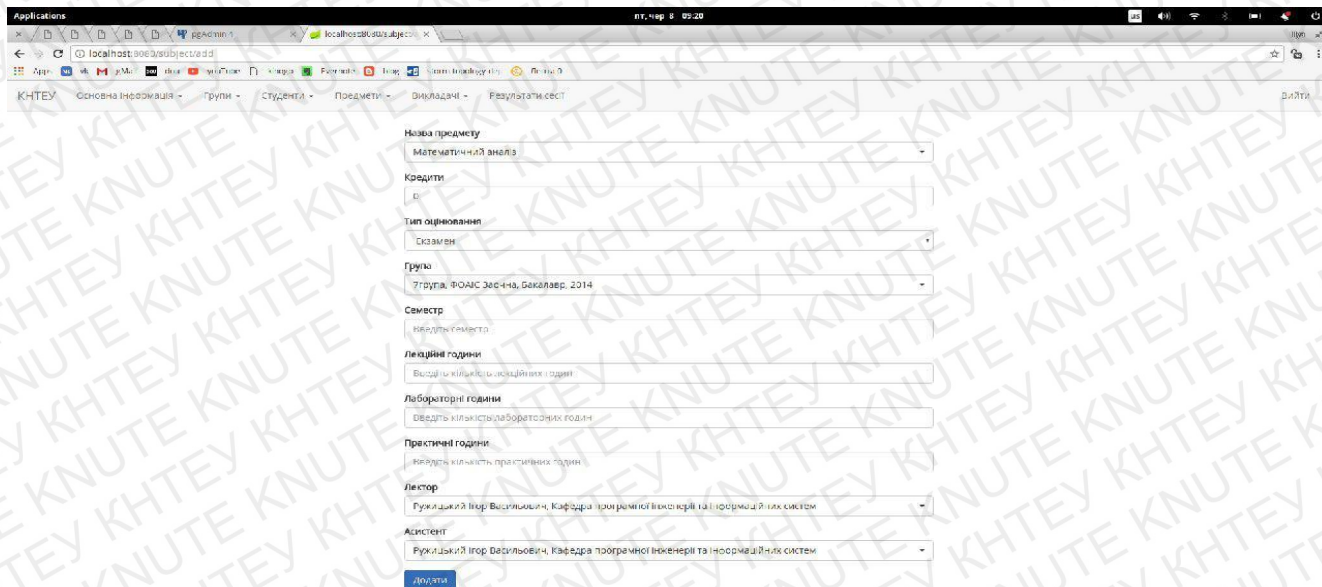
Вулиця
Введіть вулицю

Тип вулиці
Виберіть тип вулиці

Додати

Рисунок Б.10. Інтерфейс додавання студента

При необхідності співробітник деканату також може додати новий навчальний предмет, та призначити викладача, який буде проводити заняття. Інтерфейс додавання предмету зображено на рис. Б.11. Для переходу в інтерфейс на верхній панелі передбачена кнопка «Предмети».



Applications | 17. чер. 8 09:20

localhost:8080/subject/add

КНТЕУ | Основна інформація | Групи | Студенти | Предмети | Викладачі | Результати сесії

Назва предмету: Математичний аналіз

Кредити: 0

Тип оцінювання: Сказати

Група: 7 група, ФОНС Засічна, Бакалавр, 2014

Семестр: Кредитні години

Лекційні години: Введіть кількість лекційних годин

Лабораторні години: Введіть кількість лабораторних годин

Практичні години: Введіть кількість практичних годин

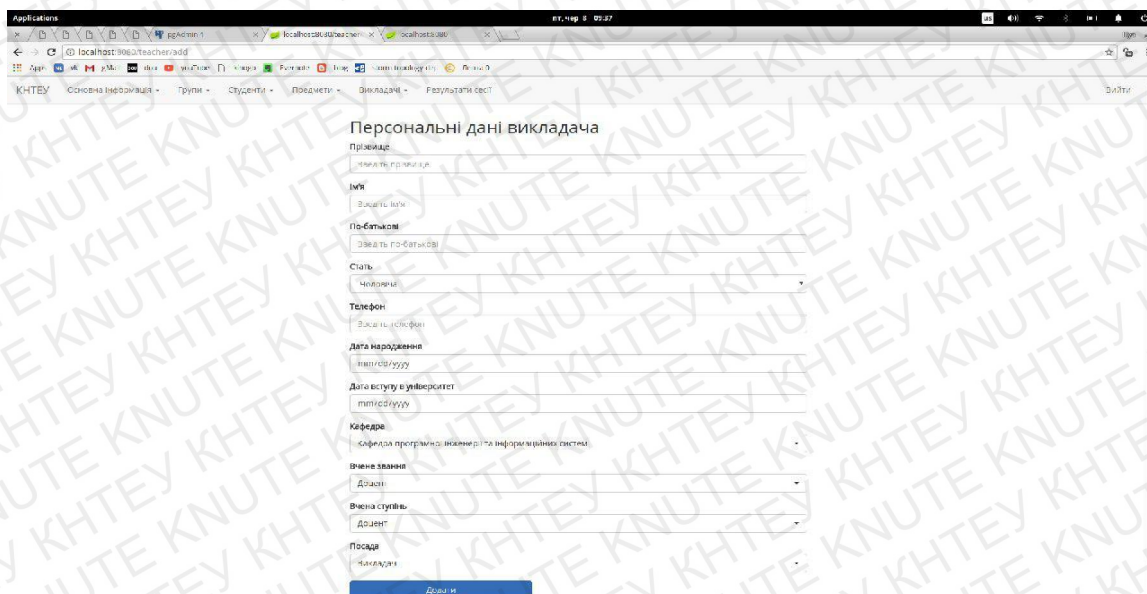
Лектор: Рукляк Ігор Васильович, Кафедра програмної інженерії та інформаційних систем

Асистент: Рукляк Ігор Васильович, Кафедра програмної інженерії та інформаційних систем

Додати

Рисунок Б.11. Інтерфейс додавання навчального предмету

При необхідності додати до бази даних нового викладача є можливість зробити це у вікні додавання викладача (рис. Б.12), яке відкривається при натисканні на кнопку «Викладачі».



Applications | 17. чер. 8 09:27

localhost:8080/teacher/add

КНТЕУ | Основна інформація | Групи | Студенти | Предмети | Викладачі | Результати сесії

Персональні дані викладача

Прізвище: Введіть прізвище

Ім'я: Введіть ім'я

По-батькові: Введіть по-батькові

Стать: Чоловічий

Телефон: Введіть телефон

Дата народження: Введіть дату

Дата вступу в університет: Введіть дату

Кафедра: Кафедра програмної інженерії та інформаційних систем

Вчене звання: Доктор

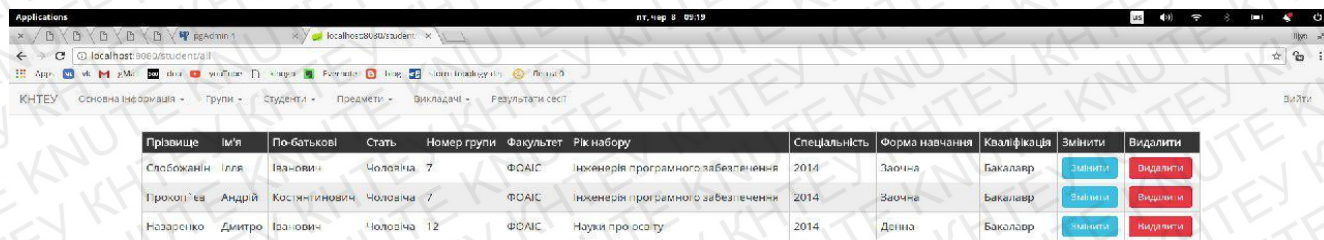
Вчене ступінь: Доктор

Посада: Викладач

Додати

Рисунок Б.12. Інтерфейс додавання викладача

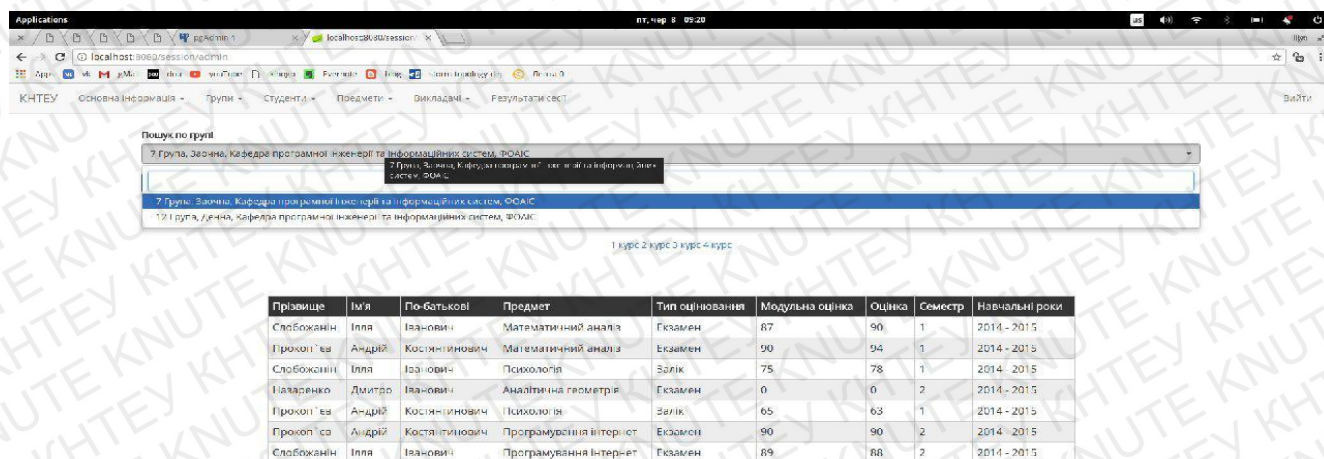
Також реалізовано можливість змінювати чи видаляти дані про студента. Під час видалення студента з'являється вікно для підтвердження видалення, для запобігання випадкового видалення (рис. Б.13).



Прізвище	Ім'я	По-батькові	Стать	Номер групи	Факультет	Рік набору	Спеціальність	Форма навчання	Кваліфікація	Змінити	Видалити
Слобожанін	Ілля	Іванович	Чоловіча	7	ФФАІС	інженерія програмного забезпечення	2014	Заочна	Бакалавр	Змінити	Видалити
Прокоп'єв	Андрій	Костянтинович	Чоловіча	7	ФФАІС	інженерія програмного забезпечення	2014	Заочна	Бакалавр	Змінити	Видалити
Назаренко	Дмитро	Іванович	Чоловіча	12	ФФАІС	Науки про освіту	2014	Довча	Бакалавр	Змінити	Видалити

Рисунок Б.13. Інтерфейс зміни та видалення даних про студента

У веб-додатку реалізовано можливість виконання звітів, які можна переглянути у відповідному вікні (рис.Б.14) або ж зберегти у вигляді файлу з розширенням «.xls» для подальшої зручної роботи зі звітом поза межами програмного комплексу (рис. Б.15). Звіти відкриваються при натисканні на кнопку «Результати сесії».



Пошук по групі

7 Група, Заочна, кафедра програмної інженерії та інформатичних систем, ФФАІС

7 Група, Заочна, Кафедра програмної інженерії та інформатичних систем, ФФАІС

7 Група, Заочна, Кафедра програмної інженерії та інформатичних систем, ФФАІС

121 група, денна, кафедра програмної інженерії та інформатичних систем, ФФАІС

1 курс 2 курс 3 курс 4 курс

Прізвище	Ім'я	По-батькові	Предмет	Тип оцінювання	Модульна оцінка	Оцінка	Семестр	Навчальні роки
Слобожанін	Ілля	Іванович	Математичний аналіз	Екзамен	87	90	1	2014 - 2015
Прокоп'єв	Андрій	Костянтинович	Математичний аналіз	Екзамен	90	94	1	2014 - 2015
Слобожанін	Ілля	Іванович	Психологія	Залік	75	78	1	2014 - 2015
Назаренко	Дмитро	Іванович	Аналітична геометрія	Екзамен	0	0	2	2014 - 2015
Прокоп'єв	Андрій	Костянтинович	Психологія	Залік	65	63	1	2014 - 2015
Прокоп'єв	Андрій	Костянтинович	Програмування Інтернет	Екзамен	90	90	2	2014 - 2015
Слобожанін	Ілля	Іванович	Програмування Інтернет	Екзамен	89	88	2	2014 - 2015

Рисунок Б.14. Перегляд звіту у вікні веб-додатку

Файл index.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head th:replace="fragment/metaStyles :: meta('Головна')">
</head>

<body>
<div class="wrapper">
  <div th:replace="fragment/sidebar :: sidebar"></div>
  <div id="content">
    <div th:replace="fragment/header :: header"></div>
    <div class="line"></div>
  </div>
</div>
<div th:replace="fragment/scripts :: scripts"></div>
</body>
</html>
```

Файл login.html

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <title>Login page</title>
```

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css"
integrity="sha384-
9gVQ4dYFwwWSjIDZnLEWnxCjeSWFphJiwGPXr1jddIhOegiu1FwO5qRGvFXOdJZ
4" crossorigin="anonymous">
```

```
<link rel="stylesheet" th:href="@{/css/signin.css}" href="../static/css/signin.css" />
</head>
```

```
<body class="text-center">
<form class="form-signin" th:action="@{/login}" method="post">
  
  <div th:if="${param.error}" class="alert alert-error" style="color: red;">
    Некоректно введено логін або пароль
  </div>
  <!-- <div th:if="${param.logout}" class="alert alert-success">-->
  <!-- You have been logged out.-->
  <!-- </div>-->
  <label for="username" class="sr-only">Логін</label>
  <input type="text" id="username" name="username" class="form-control"
placeholder="Логін" autofocus="autofocus"/>
  <label for="password" class="sr-only">Пароль</label>
  <input type="password" id="password" name="password" class="form-control"
placeholder="Пароль" required/>
  <div class="checkbox mb-3">
    <label>
      <input type="checkbox" value="remember-me" name="remember-
me">Запам'ятати мене
```

```
</label>
</div>
<button class="btn btn-lg btn-primary btn-block" style="background: #7386D5;"
type="submit">Увійти</button>
<p class="mt-5 mb-3 text-muted">&copy; 2017-2019</p>
</form>
</body>
</html>
```

Файл report.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head th:replace="fragment/metaStyles :: meta('Вибірка')">
</head>

<body>
<div class="wrapper">
<div th:replace="fragment/sidebar :: sidebar"></div>
<div id="content">
<div th:replace="fragment/header :: header"></div>

<form method="post" th:object="${searchData}">
<div class="form-group">
<label for="facultySelect">Факультет</label>
<select th:field="*{facultyId}" onchange="addCourseTermSelect()"
id="facultySelect">
<option value="0">--</option>
<option th:each="faculty : ${faculties}" th:value="${faculty.facultyId}"
th:utext="${faculty.facultyName}">
</option>
```



```
        </select>
    </div>
    <div class="form-group" id="yearTermSelectDiv">
    </div>
    <div class="form-group" id="yearTermValueSelectDiv">
    </div>
    <div class="form-group" id="groupSelectDiv">
    </div>
    <button type="submit" style="display:none;" id="submitButton" class="btn btn-
primary">Вибірка</button>
</form>
```

```
</div>
```

```
</div>
```

```
<div th:replace="fragment/scripts :: scripts"></div>
```

```
<script>
```

```
function addCourseTermSelect() {
```

```
    var map = {
```

```
        "year": "Рок",
```

```
        "term": "Семестр"
```

```
    };
```

```
    var toDelete = ["yearTermValueSelect", "groupSelect"];
```

```
    createSelect(map, "yearTermSelectDiv", "yearTermSelect", "Виберіть",
    "addCourseOrTerm()", "type", toDelete)
}
```

```
function addCourseOrTerm() {
```

```
var map1 = {  
    "0": "Всі курси",  
    "1": "1 курс",  
    "2": "2 курс",  
    "3": "3 курс",  
    "4": "4 курс",  
    "5": "Магістр 1 курс",  
    "6": "Магістр 2 курс"  
};
```

```
var map2 = {  
    "0": "Всі семестри",  
    "1": "1 курс, 1 семестр",  
    "2": "1 курс, 2 семестр",  
    "3": "2 курс, 1 семестр",  
    "4": "2 курс, 2 семестр",  
    "5": "3 курс, 1 семестр",  
    "6": "3 курс, 2 семестр",  
    "7": "4 курс, 1 семестр",  
    "8": "4 курс, 2 семестр",  
    "9": "Магістр 1 курс, 1 семестр",  
    "10": "Магістр 1 курс 2 семестр",  
    "11": "Магістр 2 курс, 1 семестр",  
    "12": "Магістр 2 курс, 2 семестр"  
};
```

```
var yearTermSelect = document.getElementById("yearTermSelect");  
var value = yearTermSelect.options[yearTermSelect.selectedIndex].value;
```

```
var toDelete = ["groupSelect"];

if (value === "term") {
    createSelect(map2, "yearTermValueSelectDiv", "yearTermValueSelect",
"Виберіть семестр", "addGroups()", "termId", toDelete);
}
if (value === "year") {
    createSelect(map1, "yearTermValueSelectDiv", "yearTermValueSelect",
"Виберіть курс", "addGroups()", "yearId", toDelete);
}
}

function addGroups() {

    var map = {
        "0": "Bci",
        "1": "Група 3",
        "2": "Група 2",
        "3": "Група 12"
    };

    createSelect(map, "groupSelectDiv", "groupSelect", "Виберіть групу",
"addSubmit()", "groupId");
}

function addSubmit() {
    document.getElementById("submitButton").style.display = "block";
}
```



```
function hideSubmit() {
```

```
    document.getElementById("submitButton").style.display = "none";
```

```
}
```

```
function      createSelect(optionContextMap,      divId,      selectId,      labelText,  
onchangeFunction, name, toDelete) {
```

```
    var selectDiv = document.getElementById(divId);
```

```
    if (toDelete !== null) {
```

```
        Array.from(toDelete).forEach(function (i) {
```

```
            checkAndRemove(document.getElementById(i));
```

```
            checkAndRemove(document.getElementById(i + "labelId"));
```

```
        });
```

```
        hideSubmit()
```

```
    }
```

```
    var select = document.getElementById(selectId);
```

```
    checkAndRemove(select);
```

```
    var label = document.getElementById(selectId + "labelId");
```

```
    checkAndRemove(label);
```

```
    var newLabel = document.createElement("label");
```

```
    newLabel.setAttribute("for", selectId);
```

```
    newLabel.setAttribute("id", selectId + "labelId");
```

```
    newLabel.innerText = labelText;
```

```
    selectDiv.appendChild(newLabel);
```

```
var newSelect = document.createElement("select");
newSelect.setAttribute("id", selectId);
if (onChangeFunction != null) {
    newSelect.setAttribute("onchange", onChangeFunction);
}
newSelect.setAttribute("name", name);
selectDiv.appendChild(newSelect);

addOptionToSelect("empty", "---", newSelect);
for (var key in optionContextMap) {
    if (optionContextMap.hasOwnProperty(key)) {
        addOptionToSelect(key, optionContextMap[key], newSelect);
    }
}

function checkAndRemove(element) {
    if (element != null) {
        element.remove();
    }
}

function addOptionToSelect(key, value, selectVar) {
    var option = document.createElement("option");
    option.setAttribute("value", key);
    option.text = value;
    selectVar.appendChild(option);
}
```

</script>

</body>

</html>

Файл **siberal.html**

<nav th:fragment="sidebar" id="sidebar" xmlns:th="http://www.thymeleaf.org"

xmlns:sec="http://www.w3.org/1999/xhtml"

xmlns="http://www.w3.org/1999/html">

<div class="sidebar-header">

</div>

<ul class="list-unstyled components">

<p>Головне меню</p>

<li class="active">

<div sec:authorize="hasAnyRole('ROLE_METHODIST', 'ROLE_ADMIN')">

<a href="#reportSubmenu" data-toggle="collapse" aria-expanded="false"

class="dropdown-toggle">Звіти

<ul class="collapse list-unstyled" id="reportSubmenu">

Звіти

<a href="#subjectSubmenu" data-toggle="collapse" aria-expanded="false"

class="dropdown-toggle">Предмети

<ul class="collapse list-unstyled" id="subjectSubmenu">

Список

Додати

Змінити

Список назв предметів

<a href="#teacherSubmenu" data-toggle="collapse" aria-expanded="false"
class="dropdown-toggle">Викладачі

<ul class="collapse list-unstyled" id="teacherSubmenu">

Список

Додати

Змінити

<a href="#studentSubmenu" data-toggle="collapse" aria-expanded="false"
class="dropdown-toggle">Студенти

<ul class="collapse list-unstyled" id="studentSubmenu">

Список

Додати

Змінити

Факультети

<ul class="collapse list-unstyled" id="facultySubmenu">

Список

Додати

Змінити

Кафедри

<ul class="collapse list-unstyled" id="departmentSubmenu">

Список

Додати

Змінити

```
</li>
</ul>
<a href="#groupSubmenu" data-toggle="collapse" aria-expanded="false"
class="dropdown-toggle">Групи</a>
<ul class="collapse list-unstyled" id="groupSubmenu">
  <li>
    <a href="/group">Список</a>
  </li>
  <li>
    <a href="/groups/add">Додати</a>
  </li>
  <li>
    <a href="/groups/edit">Змінити</a>
  </li>
</ul>
</div>
```

```
<div sec:authorize="hasRole('ROLE_TEACHER')">
  <a href="#ownTeacherSubmenu" data-toggle="collapse" aria-
expanded="false" class="dropdown-toggle">Мої пари</a>
  <ul class="collapse list-unstyled" id="ownTeacherSubmenu">
    <li>
      <a href="/group">??</a>
    </li>
  </ul>
  <a href="#teacherMarksSubmenu" data-toggle="collapse" aria-
expanded="false" class="dropdown-toggle">Оцінки</a>
  <ul class="collapse list-unstyled" id="teacherMarksSubmenu">
    <li>
```



```
<a href="/teacher/subject/">Мої предмети</a>
</li>
<li>
<a href="/teacher/subject/">Виставити</a>
</li>
</ul>
</div>
```

```
<div sec:authorize="hasRole('ROLE_STUDENT')">
  <a href="#ownStudentSubmenu" data-toggle="collapse" aria-
expanded="false" class="dropdown-toggle">Мої пари</a>
  <ul class="collapse list-unstyled" id="ownStudentSubmenu">
    <li>
      <a href="#">Розклад</a>
    </li>
  </ul>
  <a href="#studentMarksSubmenu" data-toggle="collapse" aria-
expanded="false" class="dropdown-toggle">Оцінки</a>
  <ul class="collapse list-unstyled" id="studentMarksSubmenu">
    <li>
      <a href="#">Мої оцінки</a>
    </li>
  </ul>

  <a href="/kek" data-toggle="collapse" aria-expanded="false"
class="dropdown-toggle">Оцінки</a>
  <!-- <ul class="collapse list-unstyled" id="test">-->
  <!-- <li>-->
  <!-- <a href="#">Мої оцінки</a>-->
```

<!-- -->

<!-- -->

</div>

</nav>

Файл subjectTable.html

<div th:fragment="subjectTable" xmlns:th="http://www.thymeleaf.org">

<table id="subjectTable" class="table table-striped table-bordered table-sm">

<thead>

<tr>

<th>Предмет</th>

<th>Кредити</th>

<th>Тип</th>

<th>Лекційні години</th>

<th>Лабораторні години</th>

<th>Практичні години</th>

<th>Група</th>

<th>Семестр</th>

<th>Лектор</th>

<th>Лаборант</th>

<th>Дані про відомість</th>

<th>Закрити відомість</th>

</tr>

</thead>

<tbody>

<tr th:if="\${subjects.empty}">

<td colspan="10">Дані відсутні</td>

</tr>

```

<tr th:each="subject : ${subjects}">
    <td><span th:text="${subject.subjectName.subjectDesc}"></span></td>
    <td><span th:text="${subject.credits}"></span></td>
    <td><span th:text="${subject.assessmentType}"></span></td>
    <td><span th:text="${subject.lecturesHours}"></span></td>
    <td><span th:text="${subject.labsHours}"></span></td>
    <td><span th:text="${subject.practiceHours}"></span></td>
    <td><span th:text="${subject.group.getInfo()}"></span></td>
    <td><span th:text="${subject.term}"></span></td>
    <td><span th:text="${subject.lector.getName()}"></span></td>
    <td><span th:text="${subject.laborant.getName()}"></span></td>
    <td><span
        th:text="${subject.isClosed() ? 'Закрита' : 'Не
закрита'}"></span></td>
    <div th:switch="${subject.isClosed()}">
        <td th:case="true"><span></span></td>
        <td th:case="false">
            <button
                type="button"
                th:onclick="'populateStatementText(\"
                ${subject.subjectId} + '\");'" id="closeStatementButton" class="btn btn-primary" data-
                toggle="modal"
                th:attr="data-id=" +
                ${subject.subjectId}"
                data-
                target="#exampleModal">
                Закрити відомість
            </button>
        </td>
    </div>
</tr>
</tbody>
</table>

```



```
<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-  
labelledby="exampleModalLabel" aria-hidden="true">  
  <div class="modal-dialog" role="document">  
    <div class="modal-content">  
      <div class="modal-header">  
        <h5 class="modal-title" id="exampleModalLabel">Закрити  
відомість</h5>  
        <button type="button" class="close" data-dismiss="modal" aria-  
label="Close">  
          <span aria-hidden="true">&times;</span>  
        </button>  
      </div>  
      <div class="modal-body">  
        Точно закрити відомість?  
      </div>  
      <div class="modal-footer">  
        <button type="button" class="btn btn-primary">  
          <a id="closeStatementHref" href="/subject/close/"  
style="color:inherit">Так</a>  
        </button>  
        <button type="button" class="btn btn-secondary" data-  
dismiss="modal">Hi</button>  
      </div>  
    </div>  
  </div>  
</div>
```

РЕФЕРАТ

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему

**«Розробка інформаційно-аналітичної моделі
діяльності деканату (модуль складання звітів та
створення інтерфейсів користувачів)»**

Студента 2м курсу, 5 групи,
спеціальності 121«Інженерія
програмного забезпечення»,
спеціалізації «Інженерія
програмного забезпечення»

підпис студента

Прокоп'єва Андрія
Костянтиновича

Науковий керівник
кандидат технічних наук,
доцент

підпис керівника

Харченко Олександр
Анатолійович

Гарант освітньої програми
доктор технічних наук,
професор

підпис гаранта

Криворучко Олена
Володимирівна

КИЇВ – 2019

АНОТАЦІЯ

Актуальність даної роботи полягає в тому, що в даний час деканату доводиться оперувати великою кількістю інформації, яку досить складно обробити вже наявними офісними пакетами, до того ж це буває досить незручно. Як наслідок, гостро постає питання автоматизації роботи, що надалі спричинить вивільнення певної кількості трудових ресурсів. Створення веб-додатків з кожним роком набуває все більшої популярності, адже спрямоване на полегшення роботи користувача та надає постійний доступ до потрібних даних, та значно економить час на саму реалізацію.

Метою роботи є створення інтерфейсу користувача та модуля звітів, які дають змогу використовувати функціонал програмного забезпечення та отримувати звітні матеріали в зручному вигляді.

ANNOTATION

The urgency of this work is that in a given moment the dean has to prove a large number of informations, which he suffices to try already with the existing official officials. As a result, the question of the automation of the beat begins to emerge, which will subsequently release some of the number of rigid bins. The stepping-up of the dot-dot with the dot-pockets is getting a higher populapnocte, aga the tap-popping of the poppet of the poppet of the poppa

The purpose of the work is to create a user interface and a reporting module that allow you to use the functionality of the software and receive reporting materials in a convenient form.

Загальна характеристика роботи

Актуальність теми: в даний час деканату доводиться оперувати великою кількістю інформації, яку досить складно обробити вже наявними офісними пакетами, до того ж це буває досить незручно. Як наслідок, гостро постає питання автоматизації роботи, що надалі спричинить вивільнення певної кількості трудових ресурсів.

Створення веб-додатків з кожним роком набуває все більшої популярності, адже спрямоване на полегшення роботи користувача та надає постійний доступ до потрібних даних, та дуже економить час на саму реалізацію. Тому використання веб-додатку для роботи як деканату, так і в інших сферах людської діяльності на даний момент являється актуальною темою.

Об'єкт дослідження: освітня діяльність деканату університету.

Предмет дослідження: розробка пропозицій щодо створення бази даних для веб-додатку моделі діяльності деканату.

Мета роботи: розробити модуль складання звітів та інтерфейс інформаційно-аналітичної моделі діяльності деканату.

Методи і технології розробки: аналіз алгоритмів пошуку, веб-програмування, технологія front-end розробки, візуальне і об'єктно-орієнтоване програмування, порівняння, спостереження, побудова програмного продукту на основі проаналізованих даних.

Результат роботи: Створення інтерфейсу та модуля звітів веб-додатку моделі діяльності деканату.

ОСНОВНИЙ ЗМІСТ ДИПЛОМНОЇ РОБОТИ

У **вступі** випускної кваліфікаційної роботи обґрунтовано актуальність теми, визначено об'єкт, предмет, мету і завдання роботи, також вказані технології розробки та застосовані фреймворки.

Перший розділ «Огляд організації освітнього процесу в КНТЕУ та системи управління у діяльності деканату». У даному розділі розглянуто організацію освітнього процесу в КНТЕУ, визначено за якими формами здійснюється навчання (очна, заочна). Освітній процес у КНТЕУ здійснюється за такими формами: навчальні заняття, самостійна робота, практична підготовка, контрольні заходи, що передбачає різні види його контролю.

Досліджено організаційну структуру університету та визначено, що в КНТЕУ навчається більше 40 тис. студентів. Функціонують 6 факультетів: міжнародної торгівлі та права; економіки, менеджменту та психології; фінансів та банківської справи; обліку, аудиту та інформаційних систем; ресторанно-готельного та туристичного бізнесу; торгівлі та маркетингу.

Проведено аналіз наявних програмних засобів управління в діяльності деканату та визначено, що хоч і існує програмний продукт, який автоматизує управління навчальним процесом, але він має ряд недоліків. Тому було прийнято рішення розробити програмне забезпечення, яке б не мало існуючих недоліків та дозволяло оптимізувати процес управління навчальним процесом.

Другий розділ «Проектування модуля складання звітів та користувацького інтерфейсу» присвячено постановці задачі та виборі технології JasperReports, як основної технології побудови звітів. Також в даному розділі було спроектовано універсальний звіт для користувача. Проаналізовано групи користувачів програмного продукту та виділено такі групи: неавторизований користувач, студент, викладач, співробітник деканату, декан, проректор, адміністратор системи. Детально розглянуто методи створення інтерфейсів.

Третій розділ «Розробка модуля складання звітів та інтерфейсу користувачів». У третьому розділі було виконано початкову розмітку сторінки,

розроблено структуру та стиль інтерфейсу системи. Виділено чотири групи користувачів та розподілено їх в модулі. Також було розроблено модуль складання звітів.

Висновки та пропозиції

Однією з найважливіших задач, які стоять перед ЗВО на сучасному етапі розвитку, постає ефективна організація процесів управління. Широкі можливості для вирішення цієї задачі надають сучасні інформаційні технології. Під час виконання дипломного проекту, було розроблено інтерфейс веб-додатку управління діяльністю деканату, створений за допомогою фреймворку bootstrap. Даний фреймворк був обраний у зв'язку з тим що він має ряд переваг, таких як: кросплатформеність (може працювати на декількох операційних системах), висока швидкодія, зручність написання програмного коду, потужна підтримка, open-source, зручний в поєднанні з іншими мовами програмування.

Вирішено покласти на рівень веб просту задачу відображення даних між HTTP і Java.

Описано можливості, інструменти, контейнери, ряди та блоки bootstrap. Описано технології аналізу інформації. Спроектовано та розроблено модуль складання звітів. Проаналізовано групи користувачів та виділено відповідні їм ролі в інтерфейсі. Виконано повний цикл розробки користувацького інтерфейсу. Головними перевагами розробленого інтерфейсу є інтуїтивність, модульність та кросбраузерність.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Організаційна структура КНТЕУ – Режим доступу: <https://knute.edu.ua/blog/read/?pid=5&uk/>.
2. Положення про організацію освітнього процесу студентів в КНТЕУ <https://knute.edu.ua/file/MjExMzA=/0ce40f69d25d44d4a2619ad857a2edd7.pdf>.
3. Van de Akker J.M., Hoogeveen J.A., van de Velde S.L. Parallel machine scheduling by column generation // Oper. Res.– 1999.– V. 47, N 6.– P. 862 – 872.

4. Van de Akker J.M., Hurkens C.A.J., Savelsbergh M.W.P. Timeindexed formulations for single-machine scheduling problems: column generation // INFORMS J. on Computing.– 2000.– V. 12, N 2.– P. 111 – 124.
5. Baptiste Ph., Le Pape C., Nuijten W. Constraint-based scheduling: applying constraint programming to scheduling problems // Kluwer Academic Publishers, 2001.– 198 p.
6. Brooks G.N., White C.R. An algorithm for finding optimal or near – optimal solutions to the production scheduling problem // J. Ind. Eng.– 1965.– V. 16, N 1.– P. 34 – 40.
7. Brucker P. Scheduling Algorithms. Germany: Springer-Verlag 2001. 365 p.
8. SiteMarker.Ru Інтернет технології. – Режим доступу: <http://sitemaker.ru/technologies/database/mysqlvspostgresql/>.
9. Bootstrap documentation – Режим доступу: <https://getbootstrap.com/docs/4.3/getting-started/introduction/>.
10. Bootstrap 4 Tutorial - Режим доступу: <https://www.w3schools.com/bootstrap4/>.
11. Benjamin J. Bootstrap 4 - Responsive Web Design / J. Benjamin.
12. JasperReports – Режим доступу: <https://uk.wikipedia.org/wiki/JasperReports>.
13. JasperReports® Librar – Режим доступу: <https://community.jaspersoft.com/project/jasperreports-library/>.
14. Короткий посібник зі створення звітів в JasperReports – Режим доступу: <http://ts-soft.ru/blog/jasperreports-quickstart/>.
15. JasperReports® - Free Java Reporting Library – Режим доступу: <https://github.com/TIBCOSoftware/jasperreports/>.
16. 50+ кращих доповнень до Bootstrap – Режим доступу: <https://habr.com/ru/company/dataart/blog/258301/>.
17. Створення призначених для користувача компонент для Bootstrap 4 – Режим доступу: <https://habr.com/ru/post/416235/>.
18. Опенсорс-рішення для автоматизації звітності – Режим доступу: <https://habr.com/ru/company/tinkoff/blog/461719/>.

РЕЦЕНЗІЯ
на випускню кваліфікаційну роботу
студента 2м курсу 5 групи ФОАІС
освітнього ступеня «магістр»
спеціальності 121 «Інженерія програмного забезпечення»
Прокоп'єва Андрія Костянтиновича
на тему: «Розробка інформаційно-аналітичної моделі діяльності деканату
(модуль складання звітів та створення інтерфейсів користувачів)»

Випускна кваліфікаційна робота Прокоп'єва Андрія Костянтиновича, яку подано на рецензію, виконана у відповідності до наукової тематики кафедри та у встановлений термін. Актуальність теми магістерської випускної кваліфікаційної роботи полягає у створенні інтерфейсів користувачів за допомогою сучасних технологій створення інтерфейсів, які дозволяють зручно користуватися всіма функціями програмного забезпечення для управління діяльністю деканату.

Робота складається із вступу, в якому розкриті цілі, актуальність випускної кваліфікаційної роботи, опису проведених досліджень предметної області, розробленого технічного завдання, в якому зазначено вимоги до створеної бази даних, проведено аналіз існуючих технологій розробки інтерфейсів та вибір фреймворку bootstrap для створення інтерфейсу, а також опис процесу розробки, програмна реалізація, інструкції користувача, висновки, перелік джерел інформації. Робота містить: 20 рисунків роботи з інтерфейсом користувача.

Студент використовуючи фреймворк bootstrap розробив інтерфейс веб-додатку для управління діяльністю деканату. Суттєві зауваження щодо оформлення та структури випускної кваліфікаційної роботи відсутні.

Вважаю, що робота виконана на достатньому рівні, задовольняє вимогам Вищої школи та може бути допущена до захисту, а її автору, при успішному захисті, може бути присвоєно кваліфікацію освітнього ступеня «магістр» за спеціальністю «Інженерія програмного забезпечення».

Д.е.н., завідувач кафедри
кібернетики та системного аналізу,
КНТЕУ

Роскладка А.А.