

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
(повна назва інституту/факультету)

Кафедра інформатики та програмної інженерії
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

_____ Едуард ЖАРІКОВ
(підпис) (ім'я прізвище)

“ ” _____ 2022 р.

Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інженерія програмного забезпечення
комп'ютеризованих систем»
спеціальності «121 Інженерія програмного забезпечення»

на тему: Веб-сервіс для відображення даних про підрозділ у системі
«Електронний кампус»

Виконав студент IV курсу, групи ІТ-84
(шифр групи)
Горбик Юлія Олександрівна
(прізвище, ім'я, по батькові) _____ (підпис)

Керівник доцент, к.т.н., доц., Фіногенов О. Д.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) _____ (підпис)

Консультант
з графічної
документації доцент, к.т.н., доц., Ліщук К. І.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) _____ (підпис)

Рецензент проф. каф. ІСТ, д.т.н., доц. Корнага Ярослав Ігорович
(посада, науковий ступінь, вчене звання, прізвище та ініціали) _____ (підпис)

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

Студент _____
(підпис)

Київ –2022

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

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 Інженерія програмного забезпечення

Освітньо-професійна програма – Інженерія програмного забезпечення
комп'ютеризованих систем

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Едуард ЖАРІКОВ
(підпис) (ім'я прізвище)

“ ____ ” _____ 2022 р.

ЗАВДАННЯ
на дипломний проєкт студенту

Горбик Юлії Олександрівні

(прізвище, ім'я, по батькові)

1. Тема проєкту Веб-сервіс для відображення даних про підрозділ у системі «Електронний кампус»

керівник проєкту Фіногенов Олексій Дмитрович, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «10»квітня 2022 р. №1033-с

2. Термін подання студентом проєкту «19» червня 2022 року

3. Вихідні дані до проєкту: технічне завдання

4. Зміст пояснювальної записки

1) Аналіз вимог до програмного забезпечення: аналіз предметної області, аналіз існуючих програмних рішень, аналіз вимог, функціональні та нефункціональні вимоги, постановка задачі

2) Моделювання та конструювання програмного забезпечення: архітектура програмного забезпечення, вибір бази даних, вибір та аналіз існуючих засобів для генерації та відображення графіків.

3) Аналіз якості та тестування програмного забезпечення: опис процесів тестування, опис контрольного прикладу.

4) Впровадження та супровід програмного забезпечення: розгортання програмного забезпечення та робота із програмним забезпеченням.

5. Перелік графічного матеріалу

1) Схема структурна варіантів використань _____

3) Схема бази даних _____

4) Схема структурна програмного продукту _____

6) Креслення вигляду екранних форм _____

6. Консультанти розділів проєкту

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

7. Дата видачі завдання «10» березня 2022 року _____

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Вивчення рекомендованої літератури	10.03.2022	
2	Аналіз існуючих методів розв'язання задачі	19.03.2022	
3	Постановка та формалізація задачі	29.03.2022	
4	Розробка інформаційного забезпечення	11.04.2022	
5	Алгоритмізація задачі	21.04.2022	
6	Обґрунтування вибору використаних технічних засобів	29.04.2022	
7	Розробка програмного забезпечення	16.05.2022	
8	Налагодження програми	23.05.2022	
9	Виконання графічних документів	26.05.2022	
10	Оформлення пояснювальної записки	28.05.2022	
11	Подання ДП на попередній захист	06.06.2022	
12	Подання ДП рецензенту	12.06.2022	
13	Подання ДП на основний захист	19.06.2022	

Студент

(підпис)

Юлія ГОРБИК

(ініціали, прізвище)

Керівник

(підпис)

Олексій ФІНОГЕНОВ

(ініціали, прізвище)

АНОТАЦІЯ

Пояснювальна записка дипломного проєкту складається з чотирьох розділів, містить 20 таблиць, 15 рисунків та 11 джерел – загалом 62 сторінки.

Дипломний проєкт присвячений розробці моніторингової системи для візуалізації даних, де будуть відображатись всі необхідні метрики через користувацький інтерфейс до АІС «Електронний кампус». Система моніторингу та візуалізації даних призначена для адміністративних підрозділів та керівників навчальних підрозділів, зовнішніх користувачів під час акредитації та моніторингу з боку організаторів міжнародних рейтингів.

Метою дипломного проєкту є заснування інтерактивних інструментів графічного представлення бази даних університету для зручного читання та швидкої аналітики.

Об'єкт дослідження: програмне забезпечення для аналітики та інтерактивної візуалізації даних.

Предмет дослідження: процес взаємодії університетської бази даних та застосунку для візуального представлення даних для їх відображення та структуризації.

У розділі «Аналіз вимог до програмного забезпечення» розглянуто актуальність теми, що реалізується; проаналізовано альтернативні рішення, які вже існують; розроблено список варіантів використання системи та описана постановка задачі, яка вирішується.

Розділ «Моделювання та конструювання програмного забезпечення» присвячений моделюванню складових частин для розроблюваного ПЗ; визначенню технологій та інструментів, що слід використати; побудуванню архітектури; опису процесу реалізації.

У розділі «Аналіз та оцінка якості програмного забезпечення» описано підходи для здійснення тестування системи, що було створено.

КЛЮЧОВІ СЛОВА: ДАШБОРД, МОНІТОРИНГОВА СИСТЕМА, БАЗА ДАНИХ, СТАТИСТИКА.

ABSTRACT

The explanatory note of the diploma project consists of four sections, contains 20 tables, 15 figures and 11 sources - a total of 62 pages.

The diploma project is dedicated to the development of a monitoring system for data visualization, which will display all the necessary metrics through the user interface to the AIS "Electronic Campus". The data monitoring and visualization system is designed for administrative units and heads of training units, external users during accreditation and monitoring by the organizers of international rankings.

The aim of the diploma project is to establish interactive tools for graphical presentation of the university database for easy reading and fast analysis.

Object of research: software for analytics and interactive data visualization.

Subject of research: the process of interaction between the university database and the application for visual presentation of data for display and structuring.

In the section "Analysis of software requirements" the relevance of the topic is considered; alternative solutions that already exist are analyzed; the list of variants of use of system is developed and statement of the problem which is solved is described.

The section "Software modeling and design" is devoted to modeling of components for the developed software; identification of technologies and tools to be used; architecture construction; description of the implementation process.

The section "Analysis and evaluation of software quality" describes the approach to testing the system that was created.

KEYWORDS: DASHBOARD, MONITORING SYSTEM, DATABASE, STATISTICS.

Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2022 р.

**ВЕБ-СЕРВІС ДЛЯ ВІДОБРАЖЕННЯ ДАНИХ ПРО ПІДРОЗДІЛ У
СИСТЕМІ «ЕЛЕКТРОННИЙ КАМПУС»**

Технічне завдання

КПІ.ІТ-8408.045440.01.91

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Олексій ФІНОГЕНОВ

Нормоконтроль:

_____ Катерина ЛІЩУК

Виконавець:

_____ Юлія ГОРБИК

Київ – 2022

ЗМІСТ

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ	3
2 ПІДСТАВА ДЛЯ РОЗРОБКИ	4
3 ПРИЗНАЧЕННЯ РОЗРОБКИ	5
4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	6
4.1 Вимоги до функціональних характеристик.....	6
4.2 Вимоги до надійності.....	6
4.3 Умови експлуатації	6
4.4 Вимоги до складу і параметрів технічних засобів.....	7
4.5 Вимоги до інформаційної та програмної сумісності.....	7
4.6 Вимоги до маркування та пакування	7
4.7 Вимоги до транспортування та зберігання.....	7
4.8 Спеціальні вимоги.....	7
5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ.....	9
6 СТАДІЇ І ЕТАПИ РОЗРОБКИ	11
7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....	12

					КПІ.ІТ-8408.045490.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		2

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Веб-сервіс для відображення даних про підрозділ у системі «Електронний кампус».

Галузь застосування:

Наведене технічне завдання поширюється на розробку моніторингової системи для візуалізації даних АІС «Електронний кампус» [КПІ.ІТ-8408.045440], котра використовується для відображення всіх необхідних метрик через користувацький інтерфейс та призначена для аналітики адміністрацією університету та керівників навчальних підрозділів.

					КПІ.ІТ-8408.045490.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки «Веб-сервісу для відображення даних про підрозділ у системі «Електронний кампус»» є завдання на дипломне проєктування, затверджене кафедрою інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

					КПІ.ІТ-8408.045490.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для візуалізації даних, де будуть відображатись всі необхідні метрики через користувацький інтерфейс.

Метою розробки є створення візуального представлення необхідних метрик до АІС «Електронний кампус» для адміністративних підрозділів та керівників навчальних підрозділів.

					КПІ.ІТ-8408.045490.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		5

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

4.1.1 Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

4.1.1.1 Для авторизованого користувача:

- перегляд візуалізованих даних в графіках на дашборді;
- вибір проміжку часу для графічного представлення.

4.1.1.2 Для анонімного користувача:

- перегляд обмежених даних дашборду;

4.1.1.3 Для адміністратора системи:

- створення та конфігурація дашбордів;
- створення нових користувачів;
- створення нових ролей;
- надання користувачам ролі відповідно по посади.

4.1.2 Розробку клієнтської частини представлення користувацьких даних виконати за допомогою сервісу Grafana. Розробку бази даних виконати за допомогою Prometheus.

4.1.3 Додаткові вимоги:

- надання можливості перегляду деяких даних для анонімних користувачів;
- надання адміністратору можливості створення нових ролей, організацій.

					КПІ.ІТ-8408.045490.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		6

4.2 Вимоги до надійності

4.2.1 Передбачити контроль введення інформації.

4.2.2 Передбачити захист від некоректних дій користувача.

4.2.3 Контроль авторизації користувача.

4.2.4 Перевірка наданої адміністратором ролі.

4.2.5 Обмеження наданих даних відповідно до ролі користувача.

4.2.6 Контроль надійності створених паролів.

4.3 Умови експлуатації

4.3.1 Умови експлуатації згідно СанПін 2.2.2.542 – 96.

4.3.2 Обслуговування

Обслуговування виконується адміністратором. Вибрані сервіси є безкоштовними.

4.3.3 Обслуговуючий персонал

Адміністратор здійснює обслуговування моніторингової системи, а саме:

- створення нових користувачів;
- створення нових ролей;
- надання користувачам ролі відповідно по посади;
- надання даних для кожної ролі.

					КПІ.IT-8408.045490.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		7

4.4 Вимоги до складу і параметрів технічних засобів

4.4.1 Програмне забезпечення повинно запускатися у Docker, версії 1.13.1 і вище.

4.4.2 Мінімальна конфігурація технічних засобів:

4.4.2.1 Об'єм ОЗП4096 Мб.

4.4.2.2 Версія Google Chrome..... 80.0.3987.

4.4.2.3 Швидкість інтернету..... 10 Мб/с.

4.4.2.4 Версія Google Chrome..... 80.0.3987.

4.5 Вимоги до інформаційної та програмної сумісності

4.5.1 Програмне забезпечення повинно працювати під управлінням операційних систем Windows 8.1 та новіші версії або OS X El Capitan (10.11) та новіші версії або Ubuntu 18.04 та новіші версії.

4.5.2 Вхідні дані повинні бути представлені в наступному форматі:

- CSV дані;
- дані у XML форматі.

4.5.3 Результати повинні бути представлені в наступному форматі графічних матеріалів користувацьких метрик.

4.5.4 Програмне забезпечення повинно бути реалізовано за використовуючи PromQL, SQL запити до бази даних часових рядів Prometheus або MySQL та сервісу графічного представлення даних Grafana.

4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не пред'являються.

4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не пред'являються.

4.8 Спеціальні вимоги

Згенерувати установчу версію програмного забезпечення.

					КПІ.ІТ-8408.045490.01.91	Арк.
						9
Змін.	Арк.	№ докум.	Підп.	Дата.		

5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

5.1 Програмні модулі, котрі розробляються, повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.

5.2 Програмне забезпечення повинно мати довідникову систему.

5.3 У склад супроводжувальної документації повинні входити наступні документи:

5.3.1 Пояснювальна записка не менше ніж на 50 аркушах формату А4 (без додатків 5.3.2 - 5.3.6).

5.3.2 Технічне завдання.

5.3.3 Керівництво користувача.

5.3.4 Керівництво системного програміста

5.3.5 Керівництво адміністратора

5.3.6 Програма та методика тестування

5.4 Графічна частина повинна бути виконана не менше ніж на 3 аркушах формату А3, котрі включаються у якості додатків до пояснювальної записки:

5.4.1 Схема структурна компонентів програмного забезпечення.

5.4.2 Схема бази даних

5.4.3 Схема структурна варіантів використання

6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

№	Назва етапу	Строк	Звітність
1.	Вивчення рекомендованої літератури	10.03.2022	
2.	Аналіз існуючих методів розв'язання задачі	19.03.2022	Технічне завдання
3.	Постановка та формалізація задачі	29.03.2022	Специфікації програмного забезпечення
4.	Розробка інформаційного забезпечення	11.04.2022	
5.	Алгоритмізація задачі	21.04.2022	
6.	Обґрунтування вибору використаних технічних засобів	29.04.2022	Структурна схема програмного забезпечення та специфікація
7.	Розробка програмного забезпечення	16.05.2022	Тести, результати тестування
8.	Налагодження програми	23.05.2022	Тексти програмного забезпечення
9.	Виконання графічних документів	26.05.2022	Графічний матеріал проекту
10.	Оформлення пояснювальної записки	28.05.2022	Пояснювальна записка
11.	Подання ДП на попередній захист	06.06.2022	
12.	Подання ДП рецензенту	12.06.2022	
13.	Подання ДП на основний захист	19.06.2022	

7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

					КПІ.ІТ-8408.045490.01.91	Арк.
						12
Змін.	Арк.	№ докум.	Підп.	Дата.		

Пояснювальна записка до дипломного проєкту

на тему: “Використання Dashboard для відображення даних про підрозділ у системі «Електронний кампус»”

КПІ.IT-8408.045490.02.81

Київ – 2022

ЗМІСТ

ЗМІСТ	2
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	4
ВСТУП	5
1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	7
1.1 Аналіз предметної області	7
1.2 Аналіз існуючих програмних рішень	8
1.3 Аналіз вимог	9
1.4 Функціональні вимоги системи.....	16
1.5 Нефункціональні вимоги системи	20
1.6 Постановка задачі	21
1.7 Висновки до розділу	21
2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	22
2.1 Архітектура програмного забезпечення.....	22
2.2 Вибір бази даних	24
2.2.1 PostgreSQL СУБД	24
2.2.2 MySQL СУБД.....	24
2.2.3 MongoDB СУБД	25
2.2.4 Prometheus БД	25
2.3 Вибір та аналіз існуючих засобів для генерації та відображення графіків 28	
2.3.1 ChartJs	28
2.3.2 Моніторингова система Graphite	29
2.3.3 Grafana	30
2.4 Висновки до розділу	Ошибка! Закладка не определена.
3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 37	

3.1	Аналіз якості ПЗ.....	37
3.2	Опис процесів тестування.....	39
3.3	Опис контрольного прикладу.....	41
3.4	Висновки до розділу	51
4	ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.	52
4.1	Розгортання програмного забезпечення.....	52
4.2	Робота з програмним забезпеченням.....	53
4.2.1	Загальнодоступний дашборд.....	53
4.2.2	Авторизація	53
4.2.3	Дашборд.....	54
4.2.4	Менеджмент користувачів.....	55
4.2.5	Призначення ролі доступу	56
4.2.6	Редагування графіків дашбордів.....	57
4.2.7	Створення графіків дашборду.....	58
4.3	Висновки до розділу	58
	ВИСНОВКИ.....	59
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	61

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API	– Application programing interface, прикладний програмний інтерфейс.
ЕК	– АІС «Електронний кампус» (система адміністрування навчального процесу КПІ ім. Ігоря Сікорського) [1].
АІС	– Автоматизована інформаційна система.
СУБД	– впорядкована сукупність даних, що призначена для зберігання, обробки і накопичення з використанням комп'ютера [2].
Інтерфейс	– Сукупність засобів, правил і методів, що забезпечують взаємодію користувача та пристроїв обчислювальної системи [3].
Dashboard	– Дашборд (укр. «приборна панель») - візуалізація даних, у вигляді, наближеному до вигляду індикаторів приборів.
Датасорс	– джерело даних.
Use case	– це сценарій по якому користувач взаємодіє із системою [4].

ВСТУП

Зараз веб додаток «Електронний кампус» налічує декілька тисяч користувачів із яких не тільки студенти та викладачі, а і представники адміністрації. Кожного дня користувачі із різними правами виконують певні дії у системі такі як: створення нових користувачів, занесення певних даних у систему, завантаження оцінок і тому подібне для системної роботи університету.

Із певною періодичністю у адміністрації університету виникає потреба в отриманні певного роду статистики по різних показникам для аналізу активності і прийнятті рішень про покращення навчальних процесів навчання. Однією із головних проблем, яка виникає в процесі роботи із системою це оперативний збір та візуалізація всіх необхідних даних, які знадобляться для аналізу та прийняття адміністративних рішень, що наразі повністю відсутня.

Для того щоб була можливість відразу ознайомитись із даними за певний проміжок часу необхідна візуалізація наступних показників :

- кількість користувачів у системі;
- хід голосування опитувань;
- кількість студентів, що взяли участь у опитуванні;
- відсоток назначених кураторів груп по курсам.

Ефективна моніторингова система є важливим елементом для швидкого доступу до статистичних даних. Саме завдяки їй у адміністрації буде можливість бути проінформованими та вчасно втрутитись у процес якщо показники не відповідають очікуваним. Така система є невід’ємним елементом процесу управління будь яких ініціатив, пов’язаних із АІС «Електронний кампус».

Метою дипломної роботи є розробка моніторингової системи для візуалізації даних, де будуть відображатись всі необхідні метрики через користувацький інтерфейс до АІС «Електронний кампус». Система моніторингу та візуалізації даних призначена для адміністративних

					КПІ.ІТ-8408.045490.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		5

підрозділів та керівників навчальних підрозділів, зовнішніх користувачів під час акредитації та моніторингу з боку організаторів міжнародних рейтингів.

					КПІ.ІТ-8408.045490.02.81	Арк.
						6
Змін.	Арк.	№ докум.	Підп.	Дата.		

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Аналіз предметної області

Після створення та розвитку системи «Електронний кампус» виникла проблема відображення даних згідно із певною роллю користувача. Моніторинг широко використовується у веб-технологіях для оцінки продуктивності та збору інформації про використання веб-додатків, що дозволяє відстежувати активність та визначати ефективність програми протягом усього її життєвого циклу.

Система моніторингу дає можливість обробляти, збирати та візуалізувати різного роду метрики, які необхідні адміністрації університету для прийняття рішень та аналізу поточної ситуації в університеті (згідно із кафедрою, факультетом та інститутом). Відображень всіх необхідних метрик дозволить пришвидшити прийняття рішень та ідентифікації критичної ситуації завчасно. Цільовою аудиторією є ректор університету, декани факультетів, завідувачі кафедри, які повинні відслідковувати всі метрики та саме вони є відповідальними за це.

Окрім ідентифікованих користувачів певна інформація буде доступна для анонімних користувачів, котрими можуть виступати представники адміністрації державного рівня експертні групи НАЗЯВО при підготовці до акредитації певних освітніх програм в університеті, розробники міжнародних рейтингів, що перевіряють інформацію стосовно оприлюднення основних даних про університет тощо.

Щоб оцінити можливості системи визначено наступний перелік даних для візуалізації :

- кількість активних користувачів системи із роллю «студент»;
- поточний стан опитування студентів;
- кількість студентів, що взяли участь у опитуванні;

- куратор моніторинг стану призначення кураторів груп в системі АІС «Електронний кампус».

1.2 Аналіз існуючих програмних рішень

“Електронний кампус” - це система, яка використовується університетом для менеджменту всіх навчальних активностей та дозволяє максимально організувати роботу навчально процесу онлайн освіти. У системі “Електронний кампус” є можливість переглядати статистику згідно із ролями користувача по таким параметрам:

- статистика «відвідування»;
- статистика опитувань;
- призначення куратора груп для окремого підрозділу.

На (рис. 1.1) є можливість переглянути користувацький інтерфейс системи та підхід до візуалізації даних.

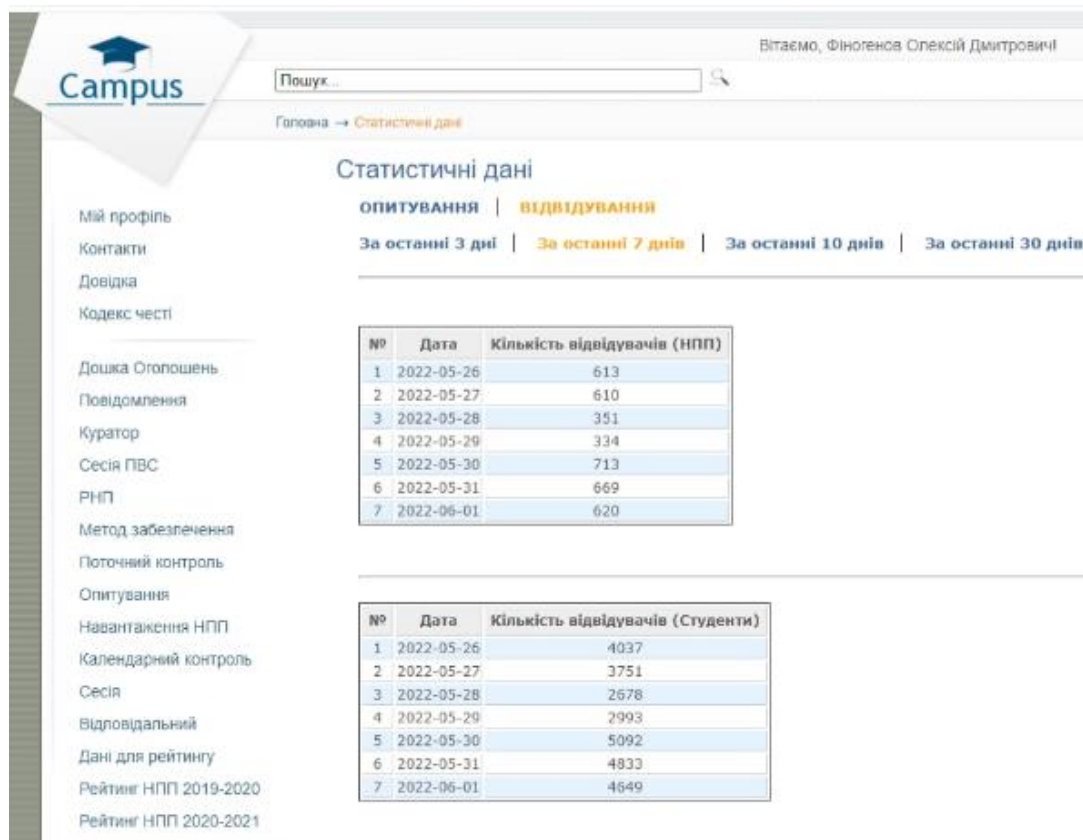


Рисунок 1.1 – Інтерфейс системи АІС «Електронний кампус»

Переваги існуючого рішення:

- забезпечено доступ до необхідних даних користувачам системи;
- забезпечено розділення ролей користувачів;
- наявність сформованих метрик (за запитом адміністративних підрозділів);
- можливість встановлювати певний часовий проміжок для відображення статистичних даних.

Недоліки існуючого рішення:

- немає можливості перегляду обмеженої статистики для неавторизованих користувачів;
- велике навантаження на систему, яке варто обмежити враховуючи технічні показники ТЗ;
- незручний інтерфейс користувача для аналізу метрик, які відображені на сторінці.

1.3 Аналіз вимог

Звичайно, доступ до даних має бути обмежений, тому було визначено список ролей: ректор, декан, завідуючий кафедри та звичайний користувач.

					КПІ.ІТ-8408.045490.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		9

Таблиця 1.1 - Опис ролей та перелік необхідних доступних даних

Роль	Опис
Ректор	Перегляд візуалізованих даних всього університету
Декан	Перегляд візуалізованих даних факультету
Завідуючий кафедри	Перегляд візуалізованих даних по кафедрі
Звичайний користувач	Дані, які відкриті будь якому користувачу.
Адміністратор	Має доступ до: - конфігурації дашборду, папки чи організації; - створення нових користувачів - видачу відповідних до посади прав на перегляд даних - створення та конфігурації графіків, дашбордів, папок та організацій - налаштування джерел даних

У даній системі можемо визначити акторів: адміністратор, декан, ректор, завідуючий кафедри, анонімний користувач.

Необхідно визначити які саме дії (тобто варіанти використання) вони виконують. У таблицях 1.2 – 1.10 описані всі можливі варіанти використання системи.

Таблиця 1.2 – Варіант використання авторизація

Назва	Авторизація
Опис	Неавторизований користувач авторизується в системі
Учасники	Неавторизований користувач
Передумови	Неавторизований користувач попередньо зареєстрований в системі адміністратором. Неавторизований користувач відкрив сторінку авторизації
Пост умови	Користувач авторизований у системі відповідно до призначеної ролі
Основний сценарій	Користувачем введено коректні дані, щоб пройти встановлену валідацію. Користувач натискає кнопку авторизації
Розширений сценарій	Відсутній

Таблиця 1.3 – Варіант використання створення нових користувачів

Назва	Створення нових користувачів
Опис	Після авторизації у системі під обліковий записом із правами адміністратора система надає можливість на сторінці Менеджменту користувачів створювати нових користувачів
Учасники	Адміністратор
Передумови	Користувач авторизований під обліковим записом адміністратора. Користувач перейшов до сторінку менеджменту користувачів
Пост умови	Нового користувача створено

Кінець таблиці 1.3

Основний сценарій	Користувач заповнює логін, пароль та електронну пошту. Новоствореному користувачеві призначається доступ до організації відповідно до ролі створеного користувача
Розширений сценарій	Відсутній

Таблиця 1.4 – Варіант використання перегляд дашборду із загальнодоступними даними

Назва	Перегляд дашборду із загальнодоступними даними
Опис	Після відкриття веб-сайту система надасть доступ до загального дашборду, де будь який користувач може ознайомитись із даними
Учасники	Неавторизований користувач
Передумови	Анонімний користувач перейшов за посиланням дашборду
Пост умови	Користувач може переглянути загальнодоступні метрики на графіку
Основний сценарій	Користувач переглядає дані дашборду
Розширений сценарій	Немає

Таблиця 1.5 – Варіант використання перегляд дашборду із графіками даних всього університету

Назва	Перегляд дашборду із графіками даних всього університету
Опис	Після авторизації під обліковий записом із правами ректора повинен бути відображений дашборд університету

Кінець таблиці 1.5

Учасники	Ректор
Передумови	Користувач авторизований під обліковим записом із правами ректора
Пост умови	У користувача є можливість переглянути графіки із даними всього університету
Основний сценарій	Користувач переходить до сторінки дашборду; Користувач переглядає графіки дашборду
Розширений сценарій	Немає

Таблиця 1.6 – Варіант використання перегляд дашборду із графіками даних факультету

Назва	Перегляд дашборду із графіками даних факультету
Опис	Після авторизації під обліковий записом із правами декана повинен бути відображений дашборд відповідного факультету
Учасники	Декан
Передумови	Користувач авторизований під обліковим записом із правами декана факультету
Пост умови	У користувача є можливість переглянути графіки із даними по факультету
Основний сценарій	Користувач переходить до сторінки дашборду; Користувач переглядає графіки дашборду
Розширений сценарій	Немає

Таблиця 1.7 – Варіант використання перегляд дашборду із графіками даних кафедри

Назва	Перегляд дашборду із графіками даних кафедри
Опис	Після авторизації під обліковий записом із правами завідуючого кафедри повинен бути відображений дашборд відповідної кафедри
Учасники	Завідуючий кафедри
Передумови	Користувач авторизований під обліковим записом із правами завідуючого кафедри
Пост умови	У користувача є можливість переглянути графіки із даними по кафедрі
Основний сценарій	Користувач переходить до сторінки дашборду; Користувач переглядає графіки дашборду
Розширений сценарій	Немає

Таблиця 1.8 – Варіант використання конфігурація відображення графіків (проміжок дат)

Назва	Конфігурація відображення графіків (проміжок дат)
Опис	На сторінці дашборда необхідно мати можливість налаштування проміжок дат, відповідно до яких, генеруються графіка дашборда
Учасники	Адміністратор, декан, ректор, завідуючий кафедри, неавторизований користувач
Передумови	Користувач авторизований під будь якою роллю, або анонімно знаходиться на загальнодоступному дашборді
Пост умови	Відображені графіки відповідно до встановлених дат

Кінець таблиці 1.8

Основний сценарій	Користувач у верхньому правому кутку дашборда натискає на проміжок дат та обирає необхідний йому проміжок; Графіки дашборда оновлюються відповідно до встановленого проміжку користувачем
Розширений сценарій	Немає

Таблиця 1.9 – Варіант використання налаштування, редагування та створення дашбордів

Назва	Налаштування, редагування та створення дашбордів
Опис	У адміністратора системи повинна бути можливість оновлювати, редагувати, створювати додаткові графіки дашборду відповідно до потреб адміністрації університету
Учасники	Адміністратор
Передумови	Користувач авторизований із правами адміністратора
Пост умови	Дашборд відредагований, новий графік дашборда створений,
Основний сценарій	Адміністратор переходить до сторінки налаштування/створення графіка; Змінює/встановлює необхідні параметри відображення даних
Розширений сценарій	Немає

Таблиця 1.10 – Варіант використання доступ до всіх конфігурацій системи

Назва	Доступ до всіх конфігурацій системи
Опис	У адміністратора системи має бути доступ до конфігурацій джерела даних
Учасники	Адміністратор
Передумови	Користувач із правами адміністратора авторизований у системі
Пост умови	Відповідне джерело даних конфігуровано у системі
Основний сценарій	Адміністратор переходить до основних налаштувань системи; Встановлює всі конфігурації для підключення джерела даних для відображення даних
Розширений сценарій	Немає

1.4 Функціональні вимоги системи

Спочатку необхідно згідно із описаними ролями у Таблиці 1.1 розробити діаграму варіантів використання (рис. 1.2) та описати Use case.

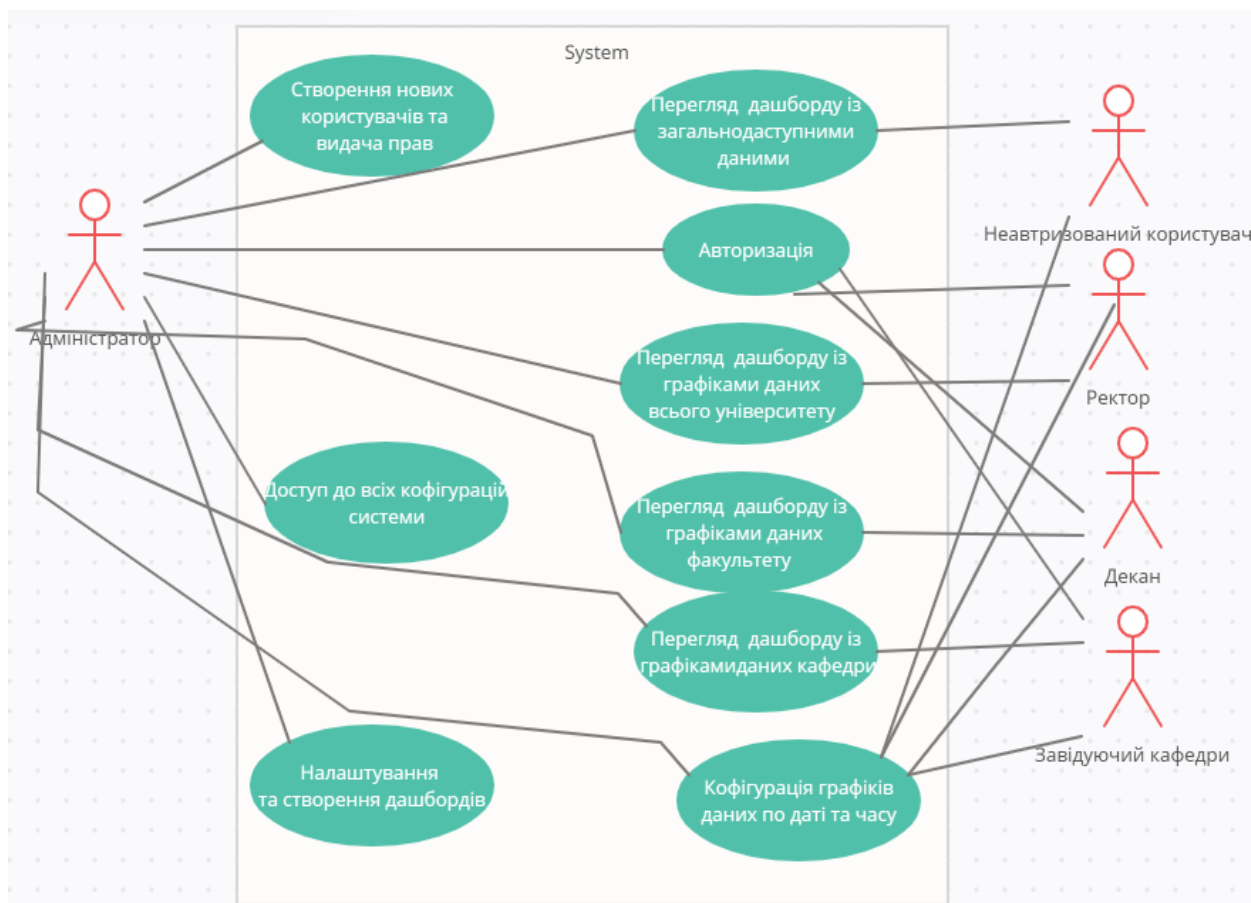


Рисунок 1.2 – Діаграма варіантів використання

Змін.	Арк.	№ докум.	Підп.	Дата.

Таблиця 1.11 – Функціональні вимоги

Варіант використання	Функціональні вимоги
Створення нових юзерів	Недоступна. Потрібно звернутись до адміністратора системи для надання логіну та паролю.
Авторизація	– Система має перевіряти та валідувати введені дані та перевіряти чи був створений обліковий запис раніше. – Коли, якщо користувач забув пароль система має надати йому форму для відновлення. – Якщо дані були введені правильно користувач має бути авторизованим та направленим до сторінки дашборду з приведеними графіками згідно ролі користувача, наданої адміністратором системи.

Кінець таблиці 1.11

1	2
Перегляд даних та конфігурація	– В залежності від приєданої ролі має відображатись доступ до графіків на сторінці дашборду. – Користувач повинен мати можливість налаштування графіка по даті та часу відображення. – Адміністратор системи повинен мати можливість: - налаштовувати дашборд, директорію чи організацію; 1. створювати нових юзерів; 2. видавати відповідно до посади доступ до певного рівня приведених даних; 3. створювати та налаштовувати графіки, дашборди, папки та організації; – Певний дашборд по обмеженим даним повинен бути доступний для неавторизованого користувача.
Вихід	– У користувача має бути можливість виходу із системи. – Після довгої відсутності у системі користувача має бути вилогінено.

1.5 Нефункціональні вимоги системи

Безпека у роботі з користувацькими даними - дуже важливе питання. Для забезпечення її конфіденційності та цілісності необхідно проаналізувати всі ризики та використати заходи для запобігання цих ризиків.

Ключовими нефункціональними вимогами являється:

- швидкість доступу до поточної системи;
- інтуїтивність та зручність інтерфейсу для користувача;
- легкість налаштування.

Таблиця 1.12 – Загрози безпеки даних

№ з/с	Уразливість даних	Загроза безпеці	Вимоги до безпеки даних
1	Поганий захист БД	Несанкціонований доступ користувачів до БД	Надійні (складні для підбору) паролі для БД
2	Дані не шифруються	Крадіжка даних із бази	Використання алгоритмів шифрування які є актуальними в даних умовах
3	Помилково введені дані	Помилки в роботі системи, загроза несанкціонованого доступу до БД	Валідація даних користувача

Приведені загрози є критичними для правильної роботи системи, тому потрібно покращити систему заходів для уникнення загроз, таких як:

- валідація даних;
- використання алгоритмів шифрування які є актуальними в даних умовах;
- складні та надійні паролі, для БД.

1.6 Постановка задачі

Основна мета дипломного проєкту полягає в наданні візуалізованої інформації з АІС «Електронний кампус» різним групам користувачів згідно призначеної ролі. Для досягнення мети потрібно вирішити наступні задачі:

- побудувати архітектуру системи, що забезпечує отримання даних з ЕК та надає UI для перегляду;
- забезпечити можливість розділення ролей користувачів
- забезпечити можливість анонімного перегляду та авторизованого перегляду за ролями;
- видати необхідну інформацію у вигляді наступних графічних елементів, як: стовпчикова діаграма, функція, скалярне значення, теплова діаграма.

Оцінити ризики що до безпеки даних

1.7 Висновки до розділу

В даному розділі були досліджені та описані функціональні і нефункціональні вимоги до програмного забезпечення, та проведений їх аналіз. Також було представлено аналіз ризиків безпеки приведеної системи, та рішення щодо зменшення впливу цих ризиків, при роботі з персональними даними користувачів.

					КПІ.ІТ-8408.045490.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		21

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Архітектура програмного забезпечення

Розроблена система моніторингу представлена у вигляді веб-додатку з архітектурою клієнт-сервер. Ця архітектура домінує у всіх типах розробки додатків. на рис. На (рис. 2.1) наведена архітектура системи.

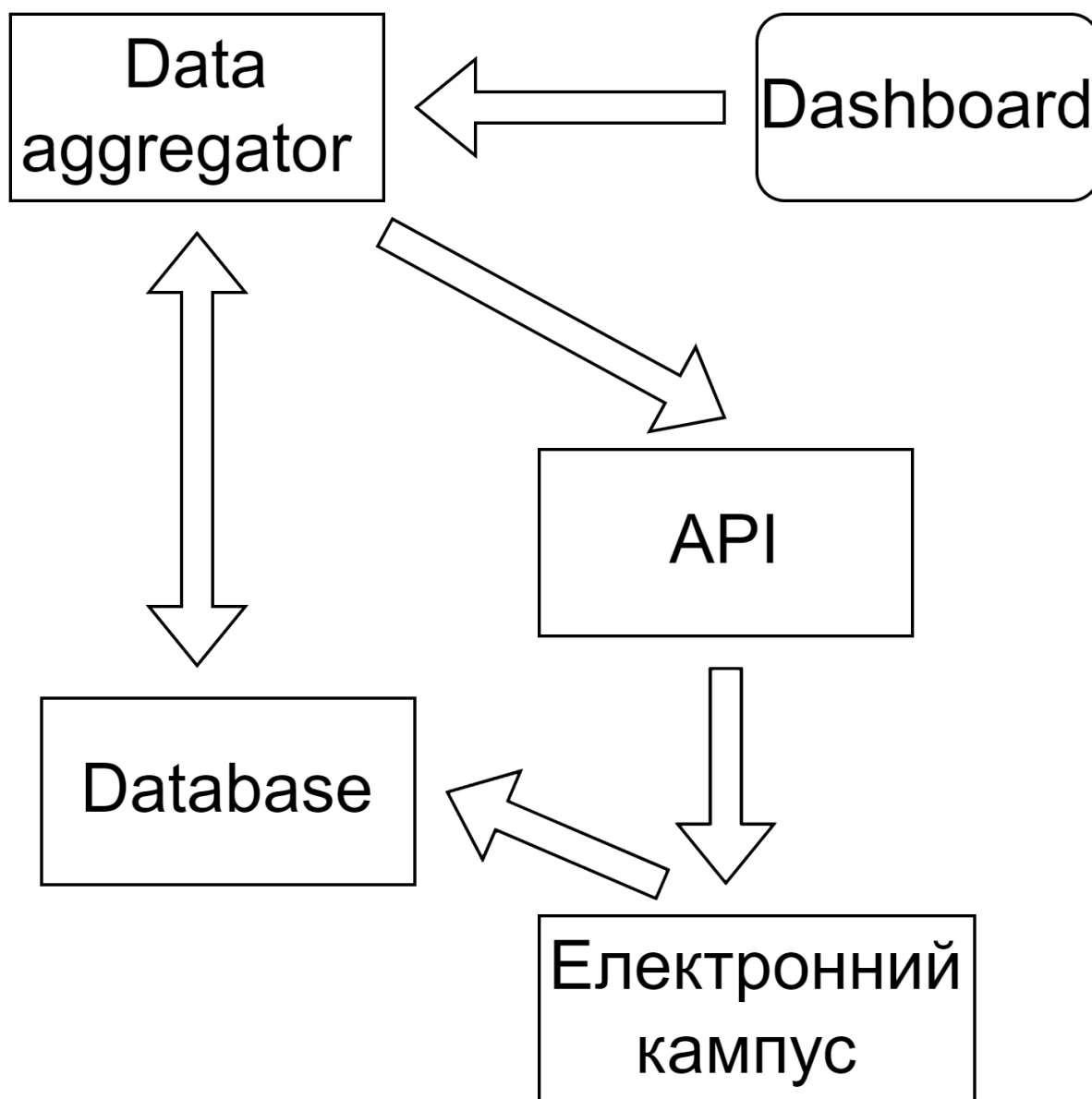


Рисунок 2.1 – Архітектура системи

Така архітектура є дуже гнучкою і дозволяє відокремити клієнтську логіку від серверної частини і може виконуватися як у браузері, так і в додатку на мобільних пристроях.

Компонентами клієнт-серверної архітектури є:

- набір серверів, що відповідають за обробку даних;
- набір клієнтів, які використовують функціональні можливості сервера;
- мережа, яка забезпечує зв'язок між клієнтами і серверами.

Усі частини архітектури клієнт-сервер є окремими блоками і працюють незалежно один від одного. В даному випадку за серверну частину відповідає вже розроблена для системи «Електронний кампус».

З боку клієнтської частини користувачам надається інтерфейс для перегляду їхніх програм і надає доступ до сторінок для перегляду статистики для конкретних програм. Також клієнтська частина представлена у вигляді модуля, вам потрібно підключитися до програми та налаштувати, які дані відображати на діаграмі, а яку статистику розраховувати.

Клієнтська частина системи моніторингу розвитку надає інтерфейс користувача для керування системою та відображення даних користувача:

- реєстрація користувача;
- перегляд додатків, що контролюються, доступ до інформаційних панелей зі статистикою;
- змінити інформацію про користувача;
- керування програмами користувачів.

Також клієнтською частиною надається функціонал підключення до клієнтського додатку, який має свої налаштування:

- збору показників ЕК;
- графіків та метрик дашборду;

2.2 Вибір бази даних

Програмне забезпечення що розробляється зберігає дость велику кількість даних про користувачів, та має складну структуру. У цьому розділі ми розглянемо існуючі системи керування базами даних та проаналізуємо їх.

2.2.1 PostgreSQL СУБД

PostgreSQL – досить широко застосовувана база даних, яка має широкий та потужний функціонал. Вона належить до групи реляційних баз, вона є open source розробкою та має відкритий код, що дозволяє компаніям та різним програмістам постійно розширювати функціонал. Її використовують у розробці складних систем, для яких є важливим безпека. Також, ця БД має внутрішню мову програмування PL/pgSQL, що дозволяє писати більш складні алгоритми.[5]

Переваги цієї БД:

- вертикальне та горизонтальне масштабування;
- можливість працювати з великою кількістю юзер-даних;
- має інтуїтивно зрозумілий інтерфейс користувача PgAdmin.

Недоліки цієї БД:

- налаштування, встановлення є досить складним, а також подальша адміністрація.

2.2.2 MySQL СУБД

MySQL – є однією з найбільш поширених реляційних БД. Також має відкритий код, і постійно оновлюється завдяки своїй популярності серед розробників. Дана система досить легка в налаштуванні. Ця БД має легкий та зручний інтерфейс у консолі який передбачений для управління, а також багато додатків із графічним юзер-інтерфейсом для управління [6].

Переваги цієї СУБД:

- легка при налаштуванні та подальшій адміністрації;

					КПІ.ІТ-8408.045490.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		24

- вона підтримує багато типів даних;
- багато прикладів використання у відкритому доступі
- легко піддається масштабуванню
- має вбудовані функції для створення безпеки.

Недоліки цієї СУБД:

- дії з даними є не дуже оперативними, операції займають порівняно велику кількість часу
- є досить не надійною в роботі з даними

2.2.3 MongoDB СУБД

MongoDb – є нереляційна та документо-орієнтована базою. Використовує текстовий формат для зберігання даних – JSON. Об’єкти бути без чіткої структури. Вона має ефективну можливість масштабованості за рахунок шардингу. Для того щоб використовувати різні мови програмування, необхідно встановити додаткові драйвери, які надають юзеру доступ до БД використовуючи зручну функціональну мову запитів [7].

Переваги MongoDB:

- схем для даних має вільний формат;
- вона досить легко масштабується;
- усі дані зберігаються у JSON форматі;
- швидкість доступу до даних.

До недоліків MongoDB можна віднести не можливість використання мови запитів SQL.

2.2.4 Prometheus БД

Prometheus – це база даних часових рядів, представляє собою набір інструментів для відслідковування та повідомлення систем, також має відкритий вихідний код, але був створений у SoundCloud. Він має досить

широку спільноту розробників та користувачів. На даний момент це відокремлений, самостійний проект який має відкритий вихідний код, та може підтримуватися незалежно від компанії [8].

Функції Prometheus:

- багатомірна модель даних;
- гнучка мова запитів для реалізації переваг великої розмірності;
- незалежна система від розподіленого сховища даних;
- вузли в рамках одного сервера автономні;
- збір часових рядів проходить за допомогою моделі вилучення HTTP;
- цілі знаходяться за допомогою пошуку служби або статичної зміни;
- можливість використання декількох режимів графіки та дашбордингу.

Наведемо порівняння розглянутих СУБД по основних критеріях (табл. 2.1).

					КПІ.ІТ-8408.045490.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		26

Таблиця 2.1 – Порівняння приведених СУБД

Назва	Масштабованість	Мова запитів	Збереження даних	Швидкість обробки запитів
PostgreSQL	горизонтальна масштабованість при використанні додаткових засобів	SQL	Дані зберігаються в таблицях із заданою структурою	Швидка обробка вхідних запитів
MySQL	Масштабується складно	Sql	Зберігання даних в таблицях із введеною структурою	Повільна обробка запитів
Mongo Db	Масштабується легко, через вбудовану реплікацію	Функціональна мова	Дані не мають обмежень по структурі	Швидка обробка вхідних запитів
Prometheus	Легко масштабується	PromQL	Дані мають чітку структуру та прив'язку до дати та часу	Швидка обробка запитів

Проаналізувавши існуючі системи контролю базами даних для розроблюваної моніторингової системи було обрано Prometheus. Вона має функціональну мову запитів, що є зручним для розробки. Особливістю використання саме цього варіанту датасорсу є можливість не на пряму взаємодіяти із базою даних «Електронного кампуса»(не навантажуючи її) та в присвєнні кожному запису даташтампу, який дуже зручний у візуалізації.

2.3 Вибір та аналіз існуючих засобів для генерації та відображення графіків

2.3.1 ChartJs

ChartJs – це бібліотека з відкритим вихідним кодом для візуалізації даних , яка підтримує вісім типів діаграм : стовпчаста , лінія , область , пиріг (пончик), бульбашка , радар , полярна та розсіяна . Створена лондонським веб-розробником Ніком Дауні в 2013 році, тепер вона підтримується спільнотою і є другою за популярністю бібліотекою діаграм JavaScript на GitHub. Chart.js відтворюється на полотні HTML5. Ця бібліотека працює добре, але графічні сторінки генеруються на стороні сервера, що не дуже ефективно рішення [9].

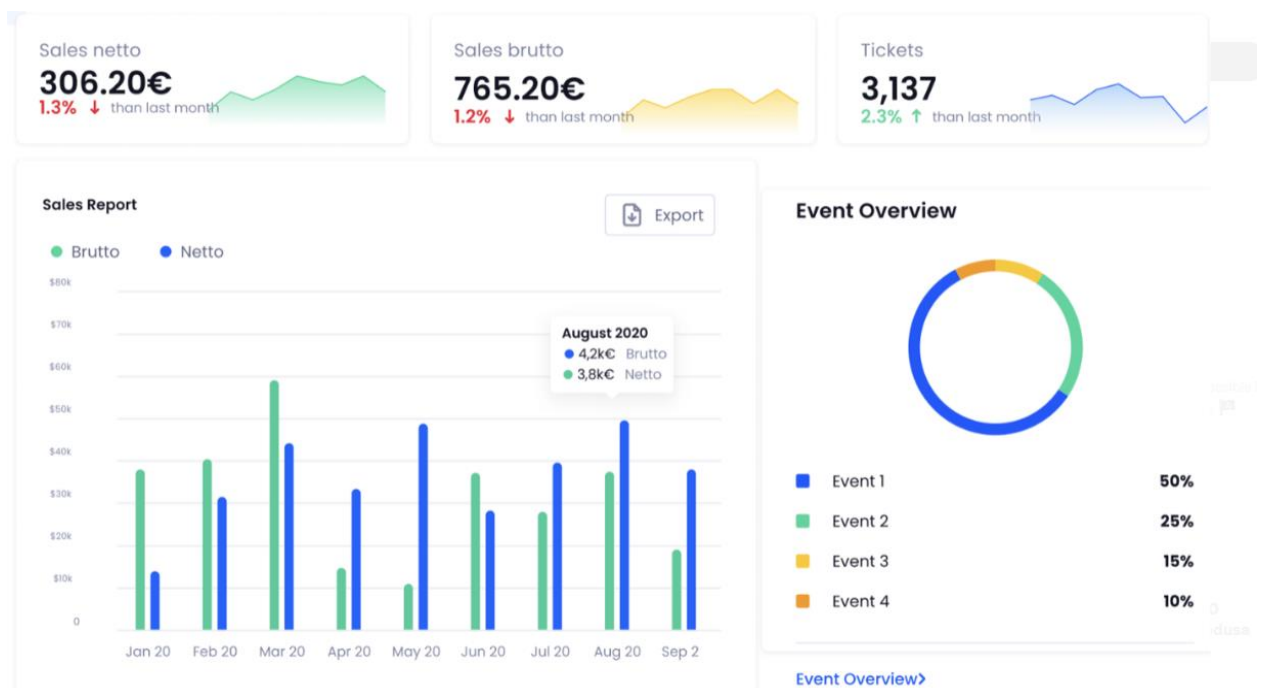


Рисунок 2.2 Користувацький інтерфейс дашборду ChartJS

Плюси:

- простота у використанні;
- генерація графіків на HTML сторінці;
- достатньо широкий список графіків для візуалізації .

Мінуси:

- статичні графіки;
- графіки генеруються на середній стороні.

2.3.2 Моніторингова система Graphite

Graphite – система для онлайн візуалізації графіків.

На рисунку 2.3 відображено зовнішній вигляд інтерфейсу користувача системи моніторингу Graphite [10].

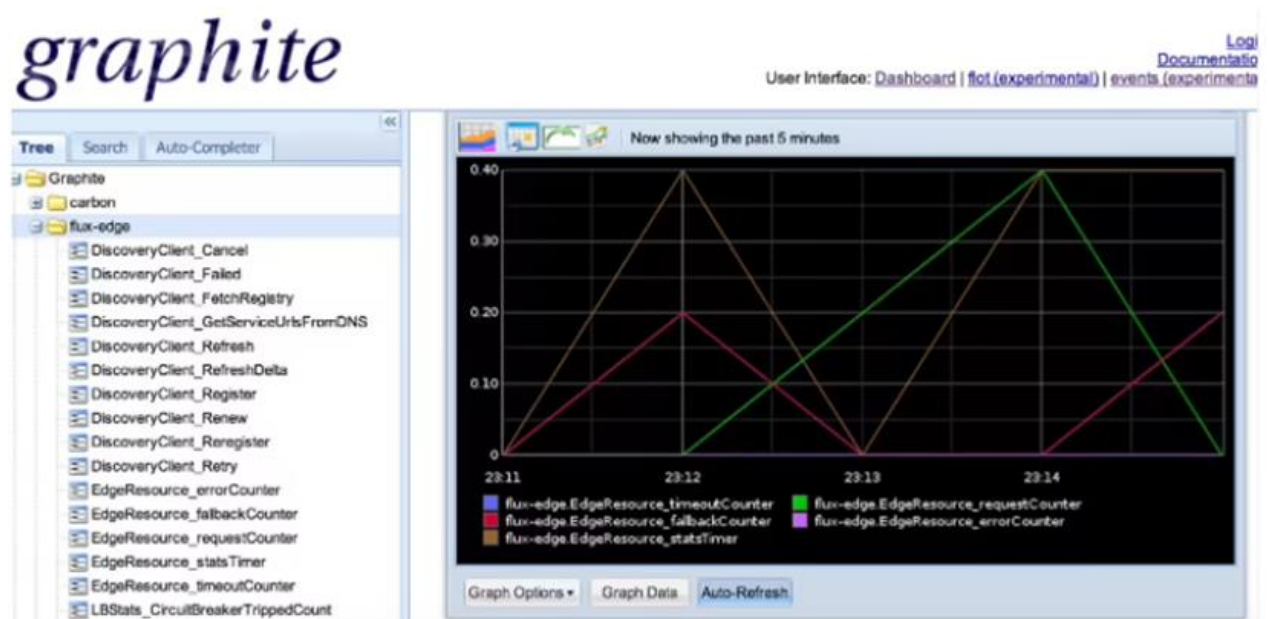


Рисунок 2.3 – Користувацький інтерфейс дашборду
Graphite

Переваги ПЗ:

На дані графіків можна накладати функції;

- можливість візуалізації даних;

- легкість у налаштуванні;
- всі графіки в одному місці відсортовані по папкам;
- дата штамп метрики задається довільно.

Недоліки ПЗ:

- досить обмежені можливості по створенню дашборда;
- не вистачає інтерактивності (немає можливості виставляти конфігурації графіків) ;
- немає можливості переглянути таблицю значень метрик;
- немає можливості відслідковувати метрики систем;
- немає статистичних показників.

2.3.3 Grafana

Grafana – це платформа для моніторингу, аналізу та візуалізації даних. Grafana не збирає і не зберігає дані, а репрезентує «сухі» цифри у вигляді красивих діаграм і графіків. Для Grafana джерело даних - системи моніторингу та сховища систематичних даних. Платформу можна підключити до Graphite, OpenTSDB, Prometheus, InfluxDB, MySQL, PostgreSQL та багатьох інших двигунів.

Платформа була розроблена у 2014 році Торкелем Одегардом. До цього моменту майже ніхто з аналітиків не вмів працювати із СУБД. Щоб отримати дані, проаналізувати, а потім оформити їх у приємному оці вигляд, доводилося запрошувати програмістів.

Grafana – універсальний інструмент. Платформа підтримує десятки різних форматів відображення даних - таблиці, графіки, розподілу, діаграми. Можна побудувати прості та зрозумілі графіки для даних будь-якої складності. А якщо вам потрібно буде оформити щось дуже рідкісне, ви зможете встановити додаткові плагіни [11].



Рисунок 2.4 – Налаштований дашборд Grafana

Grafana надає користувачам такі функції:

- зміна масштабу діаграми;
- відсортувати значення в таблиці;
- великий вибір діаграм;
- виділити колір діаграми;
- можливість розміщення кількох діаграм на приладовій панелі. вигода:
- багато налаштувань;
- відображення індикаторів в режимі реального часу;
- можливість працювати незалежно від роботи серверної частини.

Мінуси:

- складність налаштування.

2.4 Моделювання та аналіз програмного забезпечення

Для моделювання бізнес-процесів була обрана методологія діаграм BPMN.

Серед основних бізнес-процесів можна виділити:

- авторизація адміністратора, ректора, завідуючого кафедри, декана;
- перегляд дашборду

Послідовний опис авторизації користувача, адміністратора та користувача до сайту:

- анонімний користувач вводить логін або електронну адресу та пароль;
- виконується запит на сервер;
- йде порівняння сервером введених даних з даними у БД та видає

помилку у разі відсутності введених даних у БД;

- у випадку успішної валідації система перенаправляє користувача до відповідного дашборда.

На рисунку 2.5 зображена схема бізнес процесу авторизації користувача в системі.

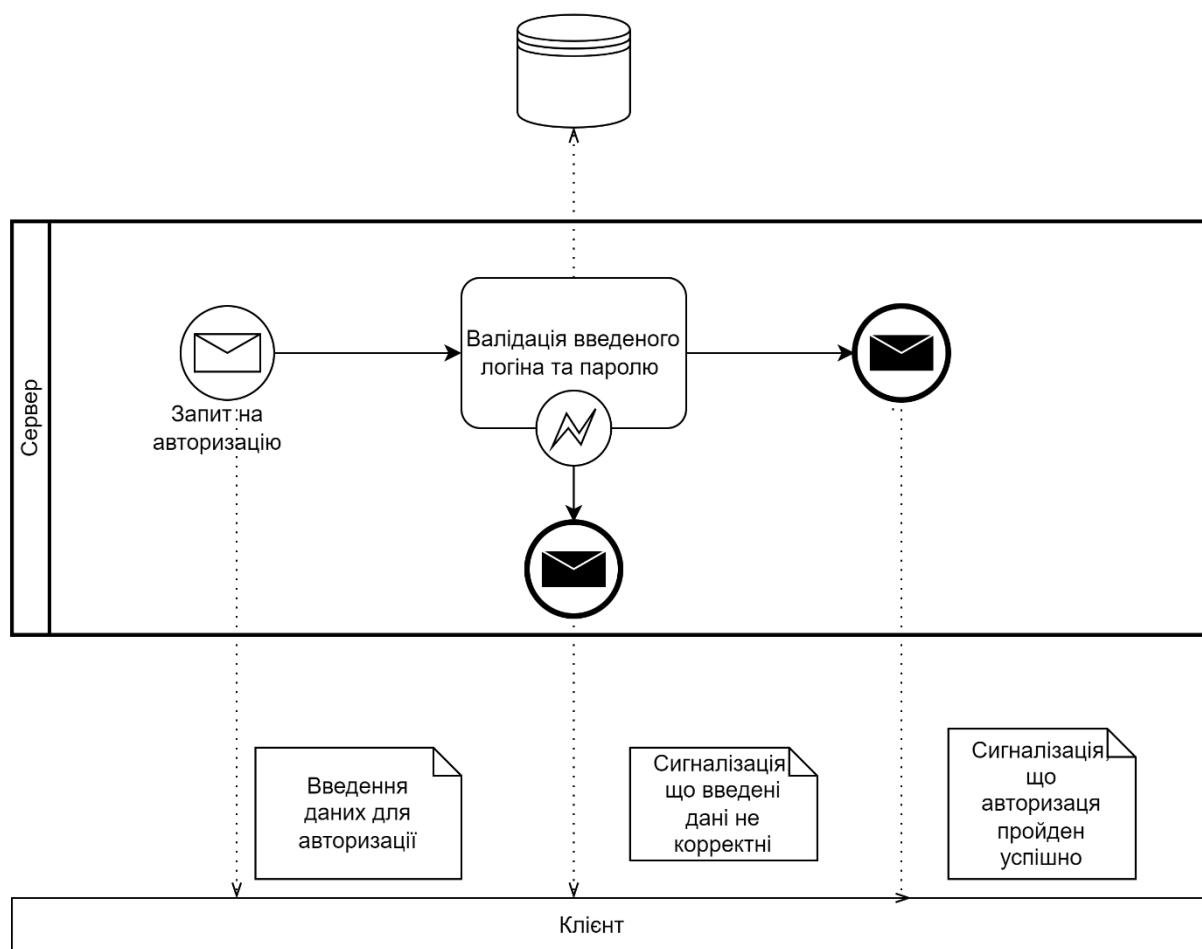


Рисунок 2.5 – Схема бізнес процесу авторизації користувача

Послідовний опис перегляду дашборда:

- користувач переходить на сторінку дашборду;
- виконується запит на сервер для отримання даних із БД;
- сервер завантажує актуальні дані із БД ;
- користувачу відображаються графіки із актуальними даними .

На рисунку 2.6 зображена схема бізнес процесу перегляду даних дашборду.

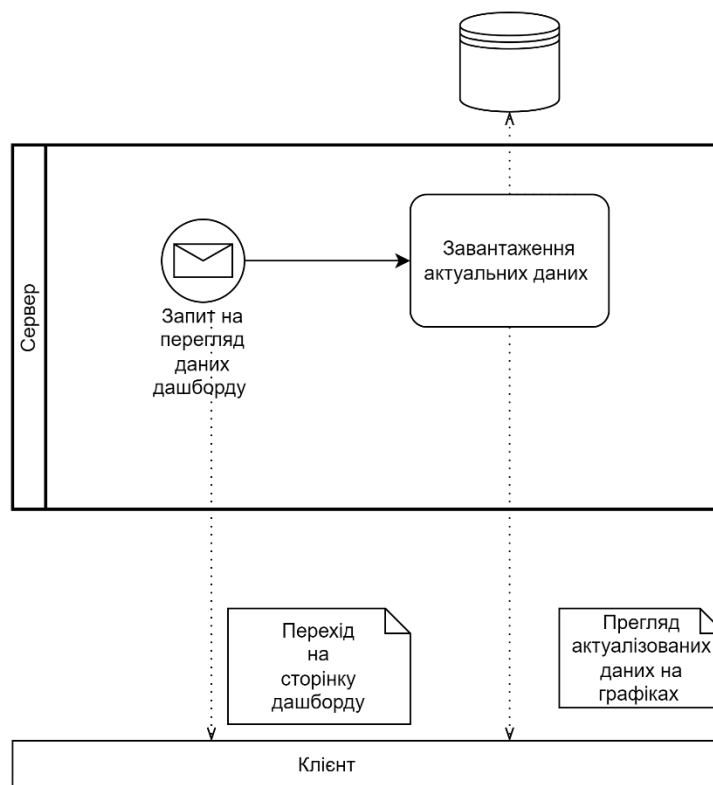


Рисунок 2.6 – Схема бізнес процесу перегляду дашборду

2.5 Структура бази даних

Було проведено 2 експерименти з архітектури роботи програмного забезпечення.

Перший передбачає доступ до даних через API, яке має віддавати необхідні дані по запиті, у відповідному форматі Prometheus, але API на даний момент не дає можливості розділити права користувачів. Тому даний експеримент був проведений лише на тестовому сервері і не відображає необхідні актуальні дані системи.

В подальшому, при додаванні API ЕК із розділенням прав можна буде перейти на дану схему архітектури.

Другий варіант, який розглядався це використання проміжної бази даних куди за запитом адміністратора чи згідно часового інтервалу завантажуються дані, які відображають стан системи. Такий механізм дозволяє зменшити навантаження на ЕК, так як вивантажує лише змінені дані і, у цьому випадку, джерелом даних являється проміжна база даних. Тоді структура таблиць відповідає структурі запитів, що відображається у користувача на графіку.

У ролі проміжної СУБД у цьому програмному забезпеченні було обрано та використано MySQL . На рисунку 2.7 зображена схема результуючих таблиць БД для вігрузки цих таблиць до графіків дашборду.

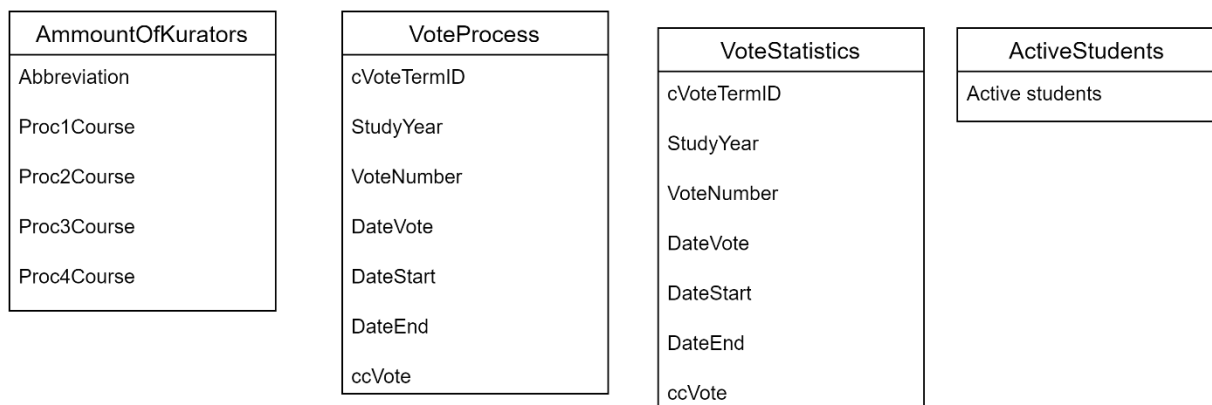


Рисунок 2.7 – Схема структури проміжної бази даних

2.6 Висновки до розділу

У цьому розділі аналізуються технології та програмне забезпечення, які будуть використовуватися в розробленому програмному забезпеченні, базі даних, технології графічного відображення. Серед баз даних: Ptometheus, MongoDB, PostgreSQL, MySQL. Серед технологій відображення діаграм: Chart.js, Grafana, Graphite. Описані їх основні недоліки та переваги, а також

вирішено які саме технології та датасорси будуть використані при розробці програмного забезпечення.

Серед різних варіантів датасорсів було обрано Prometheus. Так як необхідно не на пряму звертатись до записів датасорси для реалізації було обрано найбільш зручне рішення для візуалізації. Саме через особливість зберігання всіх даних із прив'язкою до дати та часу був обраний цей варіант.

Grafana дашборд був обраний як спосіб відображення графіків, оскільки він пропонує більше функціональних можливостей і дозволяє відображати зміни в показниках в режимі реального часу.

					КПІ.ІТ-8408.045490.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		36

3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз якості ПЗ

Якість програмного забезпечення — це сукупність атрибутів, що визначають здатність задовольняти вимоги замовника, виражені у формі вимог до розробки.

Функціональність — це набір атрибутів, які визначають здатність програмного забезпечення виконувати свої функції в певному середовищі, як того вимагає обробка та загальносистемні засоби. Функція — це певна визначена послідовність дій яка задовольняє визначені потреби користувача.

Атрибути функцій програмного забезпечення:

- функціональна повнота - це атрибут, що показує адекватність базової функціональності для вирішення певного завдання відповідно до призначення програмного забезпечення;
- правильність - властивості, що показують, як отримати правильні та послідовні результати;
- інтероперабельність — атрибут, що вказує на здатність програмного забезпечення взаємодіяти з іншими системами та середовищами;
- безпека - атрибут, що представляє здатність запобігати несанкціонований доступ до програм і даних;
- відповідність — атрибути, які вказують на відповідність конкретним стандартам, протоколам, правилам, законам і нормам.

Надійність — це набір властивостей, які представляють здатність програмного забезпечення правильно перетворювати вхідні дані в результати. Зниження надійності програмного забезпечення пов'язане з помилками у вимогах, дизайні та реалізації.

Атрибути надійності програмного забезпечення:

					КПІ.ІТ-8408.045490.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		37

- надійність - властивість, що визначає частоту збоїв через програмні помилки;
- відмовостійкість - властивість, що представляє здатність виконувати функції в ненормальних умовах (апаратні збої, помилки даних та інтерфейсу, порушення поведінки оператора тощо);
- відновлюваність – властивість, що вказує на здатність програми перезавантажити та відновлювати дані після збою;
- відповідність – вказує на ознаки відповідності існуючим стандартам, угодам, правилам, законам і нормам.

Деякі типи систем (реального часу, радіолокаційні системи, системи безпеки, комунікації, медичне обладнання тощо) мають особливі вимоги для забезпечення високої надійності з такими атрибутами, як захист від помилок, безпека, безпека та простота використання та надійність. є основним стандартом надійності.

Зручність користувача — це набір атрибутів, що характеризують умови взаємодії користувача з програмним забезпеченням.

Властивості доступності програмного забезпечення:

- Наочність залежить від зусиль, необхідних для виявлення логічної концепції програмного забезпечення та умов його застосування;
- Легкість навчання залежить від вивчення умов використання;
- Ефективність характеризується швидкістю реагування системи на дії користувача;
- Відповідність залежить від того, чи відповідає розробка вимогам чинних стандартів, протоколів, правил, законів і нормативних актів.

Ефективність – це зв’язок між кількістю задіяних ресурсів та результатів використання ПЗ (матеріали, послуги обслуговуючого персоналу, апаратура тощо).

Технічне обслуговування – це робота, необхідна для налаштування, покращення та адаптації програмного забезпечення через зміни середовища, вимог або функціональних специфікацій.

Властивості обслуговування програмного забезпечення:

- можливість аналізу - показник зусиль, необхідних для діагностики причини відмови або визначення компонента, який потребує модифікації;
- змінність - показник зусиль щодо модифікації, усунення помилок або внесення змін через помилки або нові функції в операційному середовищі;
- стійкість – властивість, що характеризує ймовірність модифікації;
- перевіроочність - властивість, що характеризує перевірку та перевірочну роботу.

Портативність – це здатність програмного забезпечення адаптуватися до мінливого середовища.

Властивості перенесення програмного забезпечення:

- адаптивність;
- легко регулюється;
- сумісність;
- послідовність.

Оцінка якості програмного забезпечення – це дія, яка повинна визначити, наскільки програмне забезпечення відповідає своєму призначенню.

3.2 Опис процесів тестування

Тестування програмного забезпечення являється одним із важливих етапів життєвого циклу розробки продукту, інкременту, задачі, адже саме на цьому етапі перевіряється відповідність функціоналу та вимог і якість програмного продукту, що забезпечує безперебійну роботу системи коректно.

завдяки тестуванню є можливість охоплення всього функціоналу розробленої системи, враховуючи людські і часові ресурси.

Тестування програмного забезпечення - це:

- процес дослідження програмного забезпечення для отримання інформації про якість продукції;
- процес перевірки відповідності продукту заявленим вимогам і реально реалізованої функціональності шляхом спостереження за його роботою в людських умовах та обмеженим набором тестів, підібраних певним чином;
- оцінити систему, щоб виявити різницю між тим, якою має бути система, і тим, яка вона є.

Загалом, тестування є одним із методів контролю якості, включаючи планування, тестування, безпосереднє тестування та аналіз результатів [2].

Виділено наступні основні функції системи, які мають бути протестовані:

- авторизація (Таблиця 3.1)
- доступ до дашборду без авторизації із обмеженими даними (Таблиця 3.2)
- створення користувача адміністратором та надання різних рівнів доступу до системи (Таблиця 3.3)
- авторизація користувача із роллю Декан (Таблиця 3.4)
- авторизація користувача із роллю Ректор (Таблиця 3.5)
- авторизація користувача із роллю Завідуючий кафедри (Таблиця 3.6)
- конфігурація дашбордів для відображення даних згідно із встановленими параметрами часового проміжку (Таблиця 3.7)
- можливість користувачу із роллю адміністратор створювати/редагувати графіки дашборда (Таблиця 3.8)

Критерії успішного тестування розробленого програмного забезпечення:

- коректне відображення дашборду із візуалізованими даними згідно із роллю користувача;
- валідація введених користувачем даних;
- обмеження дій згідно із роллю користувача;
- коректне відображення користувацького інтерфейсу веб застосунку.

3.3 Опис контрольного прикладу

Під час тестування весь ключовий та необхідний функціонал системи було протестовано. Весь ключовий функціонал системи було перевірено. Результати перевірки можна переглянути у звіті нижче:

Таблиця 3.1 – Тестування функціоналу авторизації

Ціль тесту	Перевірити функціонал авторизації у систему
Вихідний стан	Відкрито сторінку авторизації
Вхідні дані	Електронна адреса або юзернейм, встановлений адміністратором системи

Продовження таблиці 3.1

1	2
Алгоритм проведення тесту	Перейти на сторінку авторизації у застосунку; Ввести пароль та електронну адресу/юзернейм існуючого користувача; Ввести пароль та електронну адресу/юзернейм не існуючого користувача; Натиснути на кнопку “Enter” на клавіатурі або “Log in” на користувацькому інтерфейсі
Очікуваний результат	У випадку із існуючим користувачем у системі має відбутися успішна авторизація користувача; У випадку із неіснуючим користувачем система має заборонити вхід у кабінет та відобразити відповідне інформаційне модальне вікно

Продовження таблиці 3.1

1	2
Стан системи після проходження тесту	Дашборд згідно із роллю користувача відображено коректно, авторизація пройшла успішно для існуючого користувача; Під час тестування з не існуючими даними для входу з'явилося інформаційне модальне вікно із текстом "Invalid username or password"

Таблиця 3.2 – Тестування доступу до дашбору без авторизації із обмеженими даними

Ціль тесту	Перевірка доступу до дашборду без авторизації
Вихідний стан	Відкрита головна сторінка системи
Вхідні дані	Немає
Алгоритм проведення тесту	Переглянути наявність відповідних графіків із загальнодоступними даними системи "Електронний кампус"

Кінець таблиці 3.2

1	2
Очікуваний результат	Система успішно відобразить актуальні візуалізовані дані університету
Стан системи після проходження тесту	Наступні графіки із загальнодоступними даними відображено : - Кількість користувачів у системі - Статистика голосування - Хід голосування опитувань - Кількість проголосувавших - Відсоток назначених кураторів групам по курсам

Таблиця 3.3 – Тестування створення користувача адміністратором та надання різних рівнів доступу до системи

Ціль тесту	Тестування створення користувача адміністратором та надання різних рівнів доступу до системи
Вихідний стан	Користувач авторизований під обліковий записом із правами адміністратора системи.
Вхідні дані	Ім'я користувача; Електронна адреса користувача; Юзернейм; Пароль.
Алгоритм проведення тесту	Відкрити сторінку користувацького менеджменту; Натиснути на кнопку “Новий користувач” ; Заповнити ім'я, електронну адресу, юзернейм, пароль. Після створення перейти до профілю користувача та обрати організацію згідно із бажаною роллю користувача; Увійти в систему під створеними користувачем.
Очікуваний результат	Система успішно збереже нового користувача

Кінець таблиці 3.3

1	2
Стан системи після проходження тесту	Створений користувач відображається у списку користувачів

Таблиця 3.4 – Тестування відображення дашборду для ролі Декан

Ціль тесту	Тестування відображення дашборду для ролі Декан
Вихідний стан	Користувач перейшов до сторінки авторизації. Адміністратор системи попередньо створив Обліковий запис із правами доступу до дашборду деканату
Вхідні дані	Електронна адреса або юзернейм; Пароль
Алгоритм проведення тесту	Ввести електронну адресу або юзернейм; Ввести пароль; Переконались що дашборд відображений із даними відповідного факультету
Очікуваний результат	Система успішно авторизує користувача та відобразить дашборд, який був створений для деканату

Кінець таблиці 3.4

1	2
Стан системи після проходження тесту	Дашборд із графіками рівня доступу деканату відображено

Таблиця 3.5 – Тестування відображення дашборду для ролі Ректор

Ціль тесту	Тестування відображення дашборду для ролі Ректор
Вихідний стан	Користувач перейшов до сторінки авторизації. адміністратор системи попередньо створив акаунт із правами доступу до дашборду всього університету
Вхідні дані	Електронна адреса або юзернейм; Пароль
Алгоритм проведення тесту	Ввести електронну адресу або юзернейм; Ввести пароль; Переконатись що дашборд відображений із даними відповідного факультету
Очікуваний результат	Система успішно авторизує користувача та відобразить дашборд університету

Кінець таблиці 3.5

1	2
Стан системи після проходження тесту	Дашборд із графіками рівня доступу Ректор відображено

Таблиця 3.6 – Тестування відображення дашборду для ролі Завідуючий кафедри

Ціль тесту	Тестування відображення дашборду для ролі Завідуючий кафедри
Вихідний стан	Користувач перейшов до сторінки авторизації. адміністратор системи попередньо створив акаунт із правами доступу до дашборду кафедри
Вхідні дані	Електронна адреса або юзернейм; Пароль
Алгоритм проведення тесту	Електронна адреса або юзернейм; Пароль
Очікуваний результат	Ввести електронну адресу або юзернейм; Ввести пароль; Переконались що дашборд відображений із даними відповідної кафедри

Кінець таблиці 3.6

1	2
Стан системи після проходження тесту	Дашборд із графіками рівня доступу Завідуючий кафедри відображено

Таблиця 3.7 – Тестування конфігурації дашбордів для відображення даних згідно із встановленими параметрами часового проміжку

Ціль тесту	Тестування конфігурації дашбордів для відображення даних згідно із встановленими параметрами часового проміжку
Вихідний стан	Відкрита сторінка дашборду
Вхідні дані	Немає
Алгоритм проведення тесту	Перейти до сторінки дашборду (доступний для не авторизованного користувача) або залогінитись та потрапити до сторінки дашборду; У верхньому правому куті натиснути на проміжок дат та вибрати необхідний; Змінити частоту оновлення даних автоматично

Кінець таблиці 3.7

1	2
Очікуваний результат	Графіки будуть оновлені згідно із вибраним проміжком дат; Графіки мають автоматично оновлюватись згідно із обраним часом
Стан системи після проходження тесту	Графіки оновлені та актуалізовані згідно налаштувань

Таблиця 3.8 – Тестування можливості користувача із роллю адміністратор створювати/редагувати графіки дашборда

Ціль тесту	Тестування можливості користувача із роллю адміністратор редагувати графіки дашборда
Вихідний стан	Адміністратор авторизувався у систему та перейшов до певного дашборда
Вхідні дані	Немає

Кінець таблиці 3.8

1	2
Алгоритм проведення тесту	Обрати бажаний графік; Натиснути на нього та обрати Редагування; Змінити вигляд відображеного графіка; Зберегти зміни
Очікуваний результат	Зміни будуть успішно збережені
Стан системи після проходження тесту	Відкрито сторінку дашборду із оновленим графіком

3.4 Висновки до розділу

У цьому розділі було функціонально протестовано систему на коректність та відповідність до функціональних вимог.

Оскільки метою функціонального тестування є виявлення відмінностей між фактичною та очікуваною поведінкою реалізованої функціональності відповідно до специфікацій та вимог. Функціональні тести повинні охоплювати всю реалізовану функціональність і враховувати найбільш ймовірні типи помилок. Тестові сценарії, що об'єднують окремі тести, спрямовані на перевірку якості розв'язання функціональних задач.

Функціональні тести створюються на основі зовнішньої специфікації функції, проектної інформації та тексту на МП, пов'язаних з його функціональними характеристиками, і використовуються для синтезу процесу тестування та перевірки для визначення повноти функціонального завдання та його відповідності вхідним вимогам.

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розгортання програмного забезпечення

Система, описана у дипломній роботі працює на більшості популярних пристроїв: мобільних девайсах, планшетах та браузерях (Google Chrome, Opera, Mozilla Firefox, Safari).

Адміністратор системи має доступ до розробленого програмного забезпечення одразу. Щоб авторизуватись достатньо ввести пароль та електронну адресу, які були встановлені при створенні системи.

Користувачі отримують доступ до системи після того, як системний адміністратор створить профіль кожного користувача із відповідною роллю та ідентифікуючі дані для входу користувачеві.

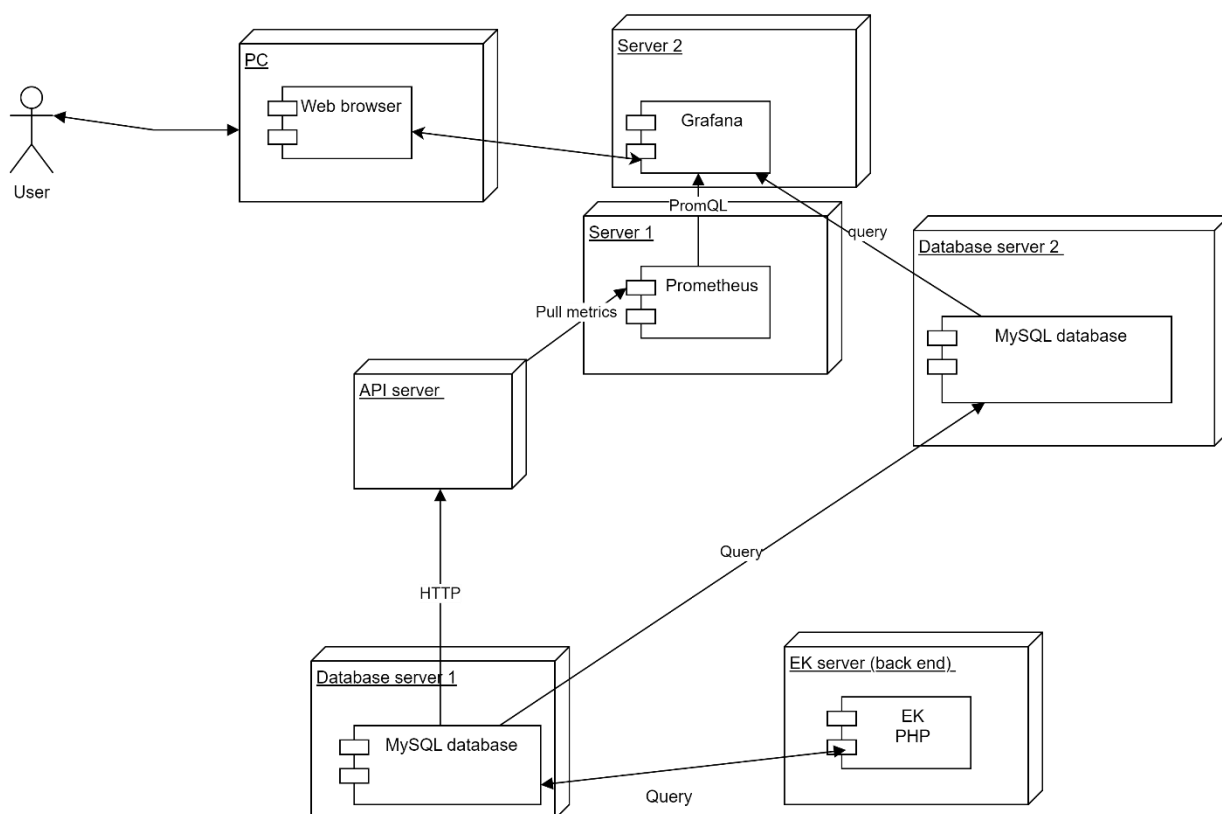


Рисунок 4.1 – Структурна схема програмного продукту

4.2 Робота з програмним забезпеченням

4.2.1 Загальнодоступний дашборд

Для доступу до загальнодоступного дашборду достатньо просто перейти за URL веб застосунку. Для налаштування проміжку дати, за якою відображаються дані на графіках потрібно встановити її див. рис. 4.2.

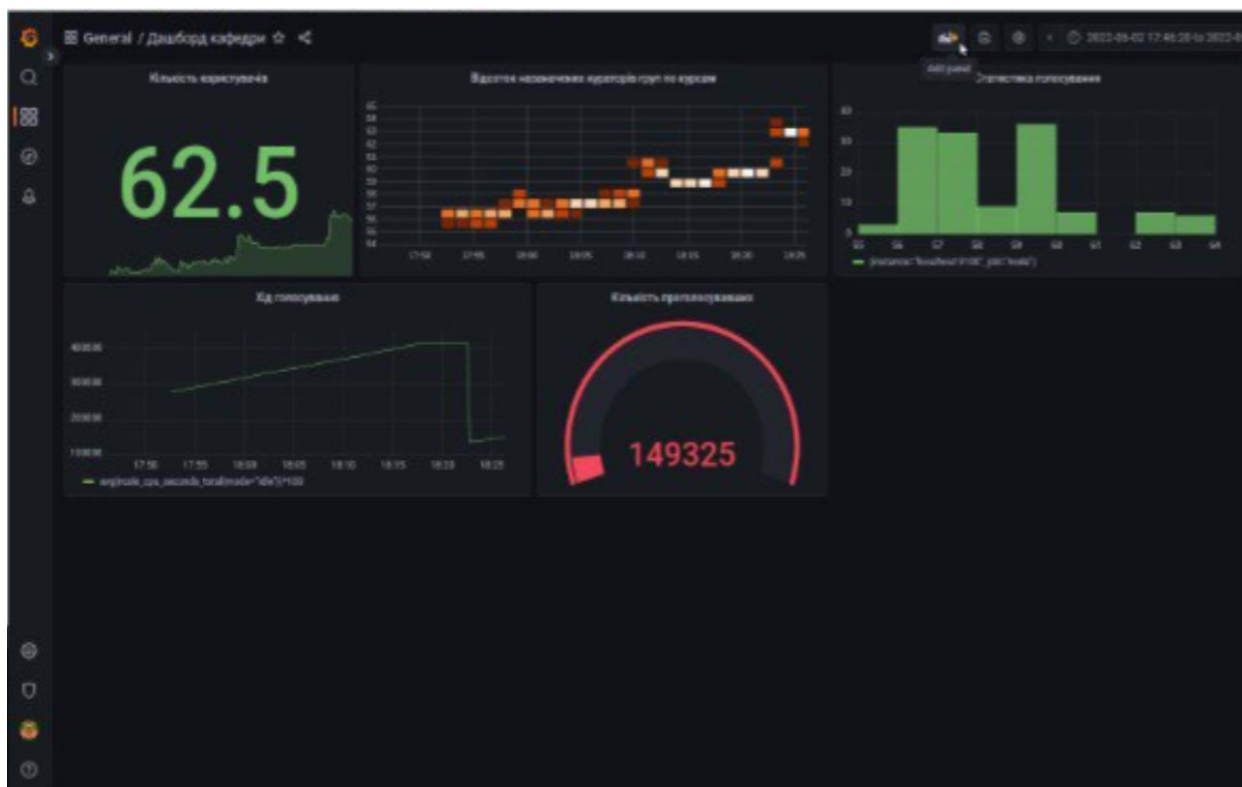


Рисунок 4.2 – Загальнодоступний дашборд

4.2.2 Авторизація

Так як реєстрації у веб застосунку немає, дані для входу можна отримати тільки від адміністратора системи, який має надати відповідні права доступу. Для входу необхідно ввести електронну адресу або юзернейм та пароль.

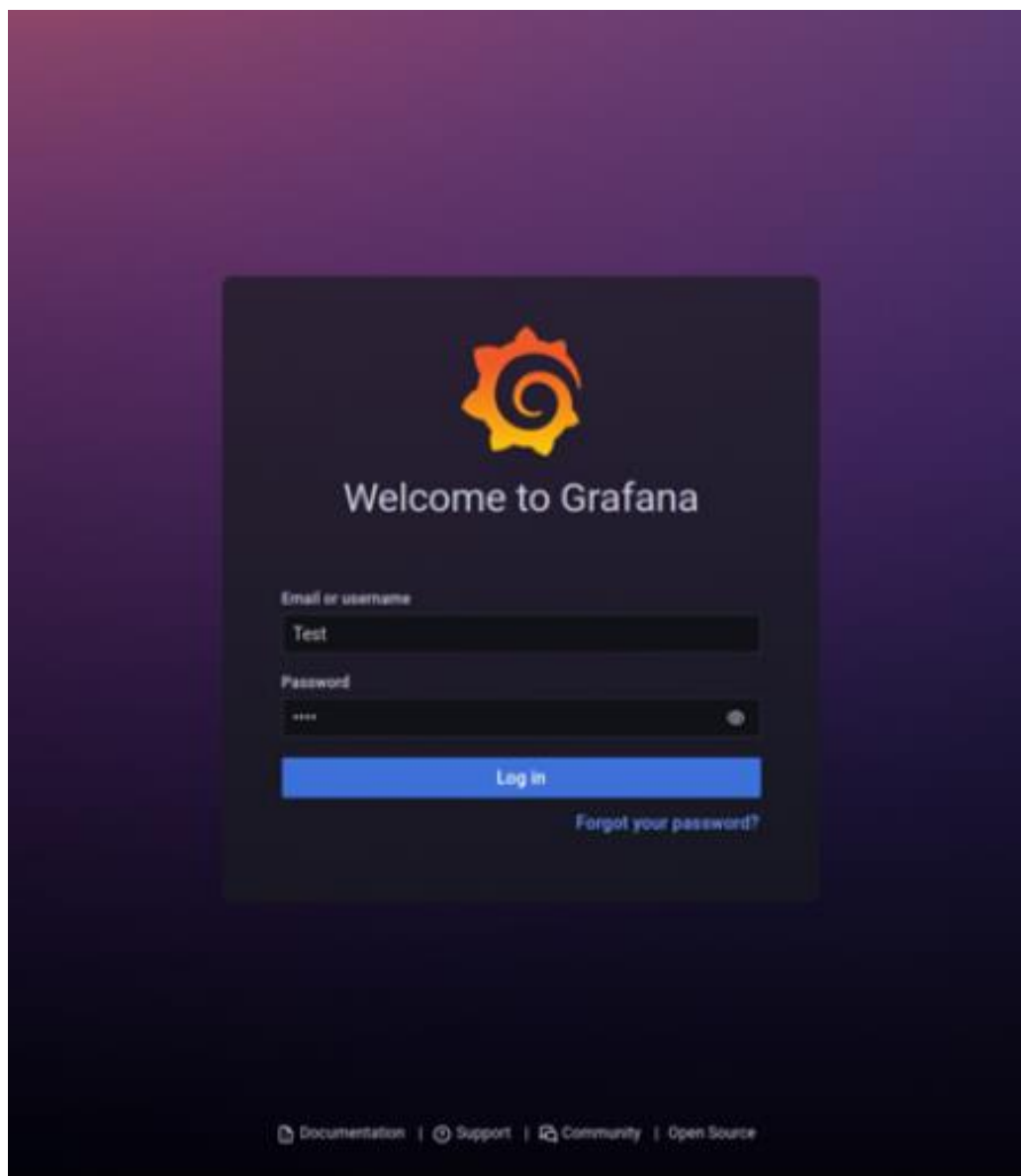


Рисунок 4.3 – Сторінка авторизації

4.2.3 Дашборд

На сторінці дашборда відображено наступні графіки згідно із назначеним рівнем доступу:

- Поточна кількість студентів, які зараз навчаються
- Хід голосування (кількість проголосувавших) у вигляді графіку часу від часу
- Кількість проголосувавших (спідометр)
- Відсоток назначених кураторів групам по курсах (температурна діаграма)

					КПІ.ІТ-8408.045490.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		54

[illegible]

4.2.4 Менеджмент користувачів

- ім'я
- електронна пошта
- юзернейм
- пароль

Необхідно заповнити поля та натиснути кнопку Створити користувача.
Нового користувача збережено та відображено у списку користувачів.

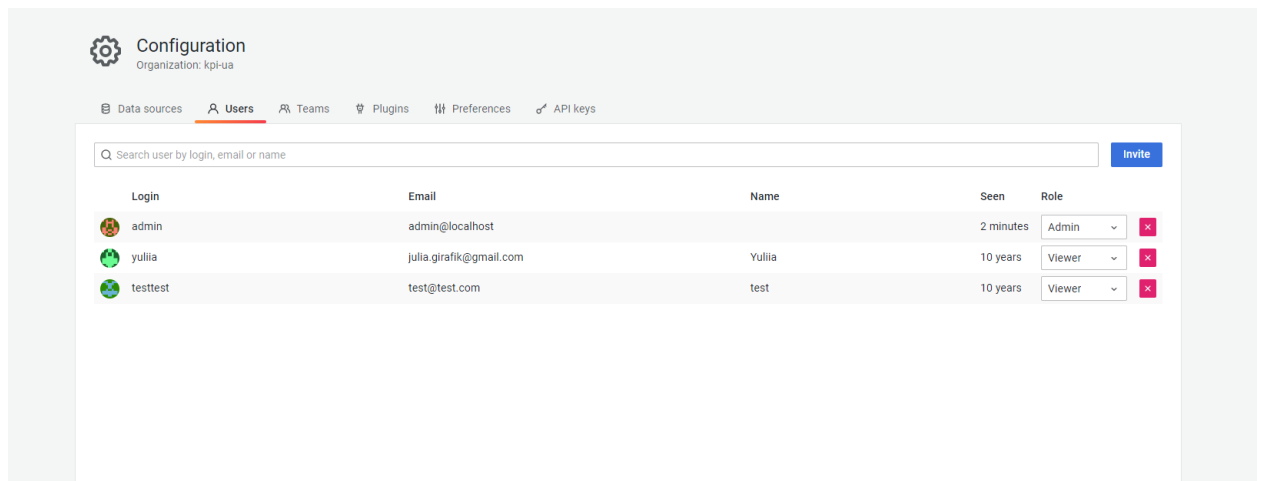


Рисунок 4.5 – Сторінка менеджменту користувачів

4.2.5 Призначення ролі доступу

Для видачі ролі користувачу необхідно натиснути на нього на сторінці Сервер адміністрування у вкладці Користувачі. Натиснути на кнопку Додати користувача до організації. З’явиться форма, де необхідно обрати організацію (для декана має бути обраний факультет, для ректора організація університет, для завідуючого кафедри – кафедра) та роль, де обов’язково має бути обрано Переглядач. Зберегти зміни можна натиснувши на кнопку Додати користувача до організації. Якщо користувача було додано помилково необхідно натиснути на кнопку Видалити користувача із організації, яка знаходиться справа навпроти відповідної організації.

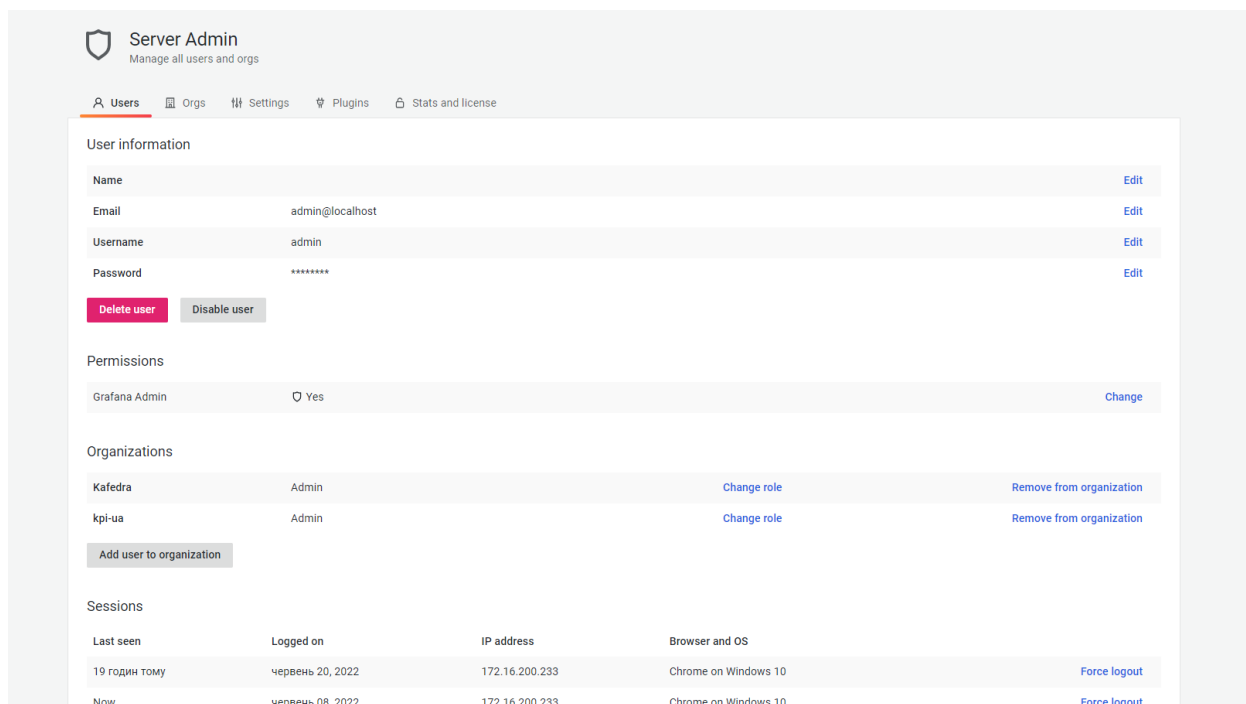


Рисунок 4.6 – Призначення ролі доступу

4.2.6 Редагування графіків дашбордів

Редагування дашбордів доступне тільки для користувача із роллю адміністратор. Щоб відредагувати графік необхідно натиснути на нього та в випадаючому меню вибрати опцію Редагувати. Після чого можна змінити запит до датасорсу, назву графіку, вигляд, у якому дані візуалізуються і т.д. Для збереження змін необхідно натиснути на кнопку Застосувати у верхньому правому куті.

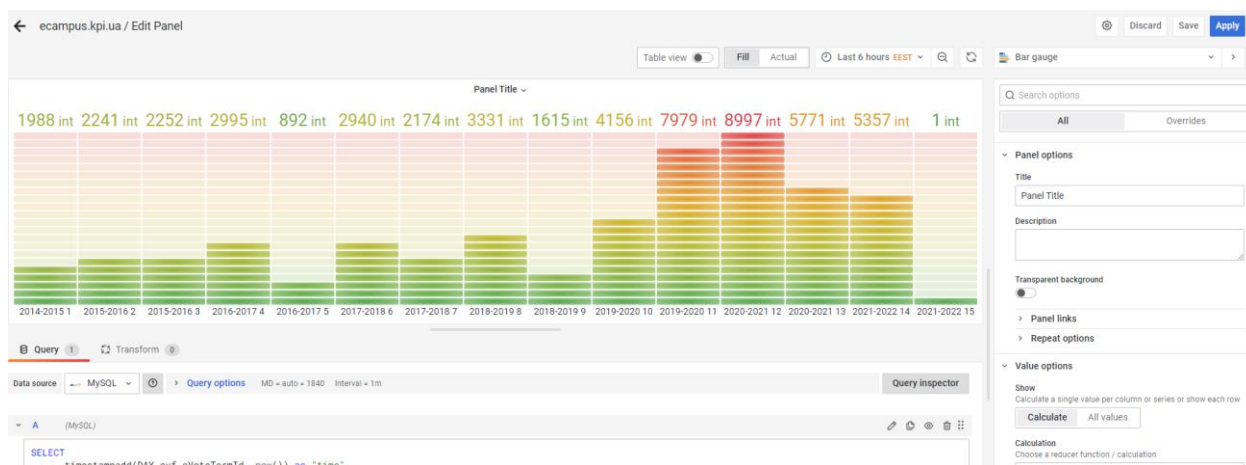


Рисунок 4.7 – Редагування графіка

4.2.7 Створення графіків дашборду

За необхідності є можливість додати графік до дашборду. Для цього необхідно перейти у дашборд та натиснути на кнопку Додати панель у верхньому правому куті. Далі треба перевірити чи встановлений по дефолту датасорс обрано коректно та написати запит до Prometheus для відображення відповідних даних. Далі необхідно назвати графік та обрати вид відображення.

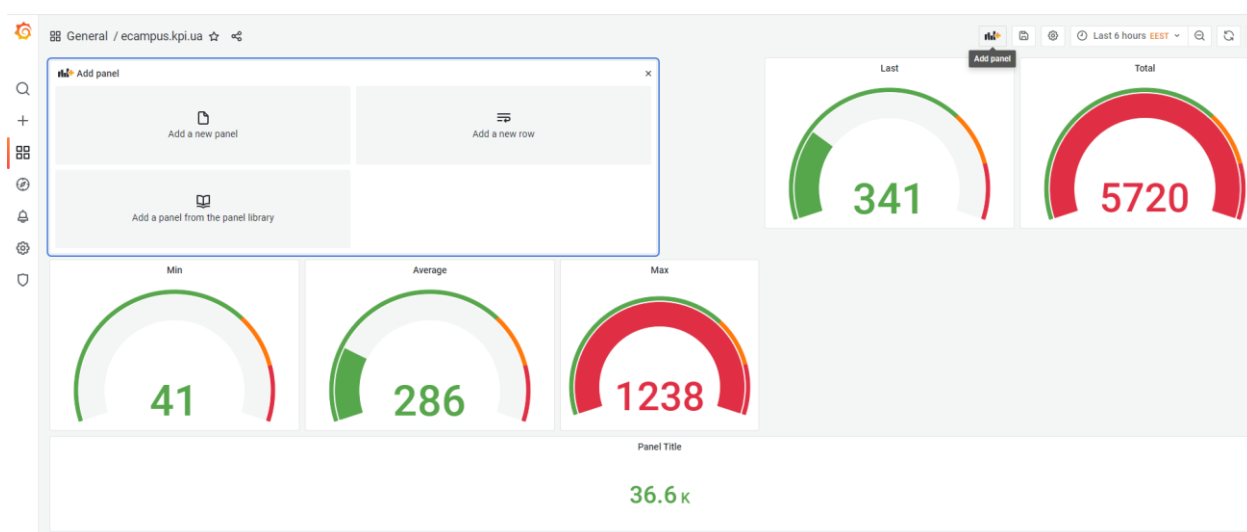


Рисунок 4.8 – Створення нового графіка

4.3 Висновки до розділу

У цьому розділі було описано та візуалізовано взаємодію між компонентами, які необхідні для коректної роботи системи.

Також, було описано коротко як варто користуватись системою як адміністратору так і користувачам.

ВИСНОВКИ

Основна мета проєкту - створення моніторингової системи, яка представляє статистичні дані користувацьких метрик в графічному вигляді та режимі реального часу.

Був виконаний аналіз проблем в реалізованих системах моніторингу веб-додатків, на основі чого були створені функціональні та нефункціональні вимоги до моніторингової систем. Дивлячись на вимоги до системи та проведений аналіз існуючих рішень, обрані основі програмі засоби для реалізації:

- Prometheus – база даних часових рядів
- Grafana – сервіс для графічного представлення користувацьких даних в режимі реального часу.

Розроблена моніторингова система надає користувачам можливість переглядати статистичну інформацію у повному обсязі для її аналізу та прийняття відповідних рішень.

Була розроблена архітектура системи, що забезпечує відображення даних. Вона включає в себе компоненти Grafana та Prometheus, але недоліком є неспроможність API та ЕК забезпечити безпеку даних.

Створена моніторингова система має такі можливості:

- збереження та графічне представлення користувацьких метрик;
- перегляд візуалізованих даних відповідно до ролі користувача;
- доступ до візуалізованих даних анонімним користувачам.

Реалізована візуалізація наступних показників :

- кількість користувачів у системі;
- хід голосування опитувань;
- кількість студентів, що взяли участь у опитуванні;
- відсоток назначених кураторів груп по курсам.

Напрямок подальших розробок є уніфікація доступу до єдиного

					КПІ.ІТ-8408.045490.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		59

інформаційного середовища КПІ ім. Ігоря Сікорського. Система автоматичного створення користувачів на базі єдиної системи обліку співробітників із розподіленням ролей, подальший розвиток API ЕК.

Працездатність моніторингової системи була перевірена на тестових метриках для користувачів із різними ролями. Система дашбордів готова до впровадження.

Створена відповідно до сформованих вимог і є універсальним рішенням для моніторингу показників системи «Електронний кампус».

					КПІ.ІТ-8408.045490.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		60

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. АІС «Електронний кампус» [Електронний ресурс]. – Режим доступу: <https://ecampus.kpi.ua>
2. СУБД [Електронний ресурс]. – Режим доступу: <https://sites.google.com/site/bazidanihsbdaccess/bazi-danih-sbd>
3. Інтерфейс [Електронний ресурс]. – Режим доступу: <https://www.jnsm.com.ua/cgi-bin/u/book/sis.pl?Qry=%B2%ED%F2%E5%F0%F4%E5%E9%F1>
4. Use case [Електронний ресурс]. – Режим доступу: <https://software-testing.org/testing/chto-takoe-yuzkeys-use-case-ili-scenariy-ispolzovaniya-v-testirovanii-po.html> <https://software-testing.org/testing/chto-takoe-yuzkeys-use-case-ili-scenariy-ispolzovaniya-v-testirovanii-po.html>
5. Салахалдін Джуба, Андрій Волков. Learning PostgreSQL 10 - Second Edition, 2018. – 448 с.
6. Берон Шварц, Петр Зайцев, Вадим Ткаченко. MySQL по максимуму, 2018. – 864 с.
7. Шенон Бредшоу, Еоін Бразіл, Христина Ходоров. MongoDB: The Definitive Guide: Powerful and Scalable Data Storage, 2019. – 514 с.
8. Prometheus [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/company/southbridge/blog/455290/>
9. ChartJS [Електронний ресурс]. – Режим доступу: <https://code.tutsplus.com/ru/tutorials/getting-started-with-chartjs-introduction--cms-28278>
10. Grafiti [Електронний ресурс]. – Режим доступу: <https://graphite.readthedocs.io/en/latest/dashboard.html>
11. Grafana [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/post/232767/>

Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2022 р.

**ВЕБ-СЕРВІС ДЛЯ ВІДОБРАЖЕННЯ ДАНИХ ПРО ПІДРОЗДІЛ У
СИСТЕМІ «ЕЛЕКТРОННИЙ КАМПУС»**

Опис програми

КПІ.ІТ-8408.045490.03.13

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Олексій ФІНОГЕНОВ

Нормоконтроль:

_____ Катерина ЛІЩУК

Виконавець:

_____ Юлія ГОРБИК

Київ – 2022

ТЕКСТ ПРОГРАМНОГО КОДУ

Тексти програмного коду

Веб-сервіс для відображення даних про підрозділ у системі «Електронний кампус»

(Найменування програми (документа))

DVD-R

(Вид носія даних)

28 арк, 15.6 Кб

(Обсяг програми, арк., Кб)

Київ – 2022

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		2

```

{
  "annotations": {
    "list": [
      {
        "builtIn": 1,
        "datasource": "-- Grafana --",
        "enable": true,
        "hide": true,
        "iconColor": "rgba(0, 211, 255, 1)",
        "name": "Annotations & Alerts",
        "target": {
          "limit": 100,
          "matchAny": false,
          "tags": [],
          "type": "dashboard"
        },
        "type": "dashboard"
      }
    ]
  },
  "editable": true,
  "fiscalYearStartMonth": 0,
  "graphTooltip": 0,
  "id": 1,
  "links": [],
  "liveNow": false,
  "panels": [
    {
      "datasource": {
        "type": "mysql",
        "uid": "TahU9ACnk"
      },
      "fieldConfig": {

```

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

```

    "defaults": {
      "mappings": [],
      "thresholds": {
        "mode": "percentage",
        "steps": [
          {
            "color": "green",
            "value": null
          },
          {
            "color": "orange",
            "value": 70
          },
          {
            "color": "red",
            "value": 85
          }
        ]
      }
    },
    "overrides": [],
  },
  "gridPos": {
    "h": 7,
    "w": 5,
    "x": 0,
    "y": 0
  },
  "id": 16,
  "options": {
    "orientation": "auto",
    "reduceOptions": {
      "calcs": [

```

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

```

        "min"
    ],
    "fields": "",
    "values": false
},
"showThresholdLabels": false,
"showThresholdMarkers": true
},
"pluginVersion": "8.3.6",
"targets": [
    {
        "datasource": {
            "type": "mysql",
            "uid": "TahU9ACnk"
        },
        "format": "time_series",
        "group": [],
        "metricColumn": "none",
        "rawQuery": true,
        "rawSql": "SELECT \n            cvf.DateVote as \"time\", \n
count(DISTINCT cvf.sPersonalityId) as \"value\" \n        FROM \n
cVoteFact cvf \n            INNER JOIN cVoteTerm cvt ON
(cvf.cVoteTermId = cvt.cVoteTermId) \n        WHERE \n
cvt.cVoteTermId = 14 \n        GROUP BY \n            cvf.DateVote",
        "refId": "A",
        "select": [
            [
                {
                    "params": [
                        "AgreementId"
                    ],
                    "type": "column"
                }
            ]
        ]
    }
]

```

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		5

```

    ],
    "table": "Agreement",
    "timeColumn": "vcChangeDate",
    "timeColumnType": "timestamp",
    "where": [
        {
            "name": "$__timeFilter",
            "params": [],
            "type": "macro"
        }
    ]
    },
    "title": "Min",
    "type": "gauge"
},
{
    "datasource": {
        "type": "mysql",
        "uid": "TahU9ACnk"
    },
    "fieldConfig": {
        "defaults": {
            "mappings": [],
            "thresholds": {
                "mode": "percentage",
                "steps": [
                    {
                        "color": "green",
                        "value": null
                    },
                    {
                        "color": "orange",

```

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		6

```

        "value": 70
      },
      {
        "color": "red",
        "value": 85
      }
    ]
  }
},
"overrides": []
},
"gridPos": {
  "h": 7,
  "w": 5,
  "x": 5,
  "y": 0
},
"id": 17,
"options": {
  "orientation": "auto",
  "reduceOptions": {
    "calcs": [
      "mean"
    ],
    "fields": "",
    "values": false
  },
  "showThresholdLabels": false,
  "showThresholdMarkers": true
},
"pluginVersion": "8.3.6",
"targets": [
  {

```

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		7

```

"datasource": {
  "type": "mysql",
  "uid": "TahU9ACnk"
},
"format": "time_series",
"group": [],
"metricColumn": "none",
"rawQuery": true,
"rawSql": "SELECT \n          cvf.DateVote as \"time\", \n
count(DISTINCT cvf.sPersonalityId) as \"value\" \n      FROM \n
cVoteFact cvf \n          INNER JOIN cVoteTerm cvt ON
(cvf.cVoteTermId = cvt.cVoteTermId) \n      WHERE \n
cvt.cVoteTermId = 14 \n      GROUP BY \n          cvf.DateVote",
"refId": "A",
"select": [
  [
    {
      "params": [
        "AgreementId"
      ],
      "type": "column"
    }
  ]
],
"table": "Agreement",
"timeColumn": "vcChangeDate",
"timeColumnType": "timestamp",
"where": [
  {
    "name": "$__timeFilter",
    "params": [],
    "type": "macro"
  }
]

```

```

    }
  ],
  "title": "Average",
  "type": "gauge"
},
{
  "datasource": {
    "type": "mysql",
    "uid": "TahU9ACnk"
  },
  "fieldConfig": {
    "defaults": {
      "mappings": [],
      "thresholds": {
        "mode": "percentage",
        "steps": [
          {
            "color": "green",
            "value": null
          },
          {
            "color": "orange",
            "value": 70
          },
          {
            "color": "red",
            "value": 85
          }
        ]
      }
    },
    "overrides": []
  },

```

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		9

```

"gridPos": {
  "h": 7,
  "w": 4,
  "x": 10,
  "y": 0
},
"id": 18,
"options": {
  "orientation": "auto",
  "reduceOptions": {
    "calcs": [
      "max"
    ],
    "fields": "",
    "values": false
  },
  "showThresholdLabels": false,
  "showThresholdMarkers": true
},
"pluginVersion": "8.3.6",
"targets": [
  {
    "datasource": {
      "type": "mysql",
      "uid": "TahU9ACnk"
    },
    "format": "time_series",
    "group": [],
    "metricColumn": "none",
    "rawQuery": true,
    "rawSql": "SELECT \n          cvf.DateVote as \"time\", \n          count(DISTINCT cvf.sPersonalityId) as \"value\" \n          FROM \n          cVoteFact cvf \n          INNER JOIN cVoteTerm cvt ON

```

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		10

```

(cvf.cVoteTermId = cvt.cVoteTermId)\n      WHERE\n      cvt.cVoteTermId = 14\n      GROUP BY \n      cvf.DateVote",

      "refId": "A",

      "select": [

        [

          {

            "params": [

              "AgreementId"

            ],

            "type": "column"

          }

        ]

      ],

      "table": "Agreement",

      "timeColumn": "vcChangeDate",

      "timeColumnType": "timestamp",

      "where": [

        {

          "name": "$__timeFilter",

          "params": [],

          "type": "macro"

        }

      ]

    },

    "title": "Max",

    "type": "gauge"

  },

  {

    "datasource": {

      "type": "mysql",

      "uid": "TahU9ACnk"

    },

```

```

"fieldConfig": {
  "defaults": {
    "mappings": [],
    "thresholds": {
      "mode": "percentage",
      "steps": [
        {
          "color": "green",
          "value": null
        },
        {
          "color": "orange",
          "value": 70
        },
        {
          "color": "red",
          "value": 85
        }
      ]
    }
  },
  "overrides": []
},
"gridPos": {
  "h": 7,
  "w": 5,
  "x": 14,
  "y": 0
},
"id": 19,
"options": {
  "orientation": "auto",
  "reduceOptions": {

```

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		12

```

    "calcs": [
        "last"
    ],
    "fields": "",
    "values": false
},
"showThresholdLabels": false,
"showThresholdMarkers": true
},
"pluginVersion": "8.3.6",
"targets": [
    {
        "datasource": {
            "type": "mysql",
            "uid": "TahU9ACnk"
        },
        "format": "time_series",
        "group": [],
        "metricColumn": "none",
        "rawQuery": true,
        "rawSql": "SELECT \n          cvf.DateVote as \"time\", \n          count(DISTINCT cvf.sPersonalityId) as \"value\" \n          FROM \n          cVoteFact cvf \n          INNER JOIN cVoteTerm cvt ON \n          (cvf.cVoteTermId = cvt.cVoteTermId) \n          WHERE \n          cvt.cVoteTermId = 14 \n          GROUP BY \n          cvf.DateVote",
        "refId": "A",
        "select": [
            [
                {
                    "params": [
                        "AgreementId"
                    ],
                    "type": "column"
                }
            ]
        ]
    }
]

```

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		13

```

        ]
    ],
    "table": "Agreement",
    "timeColumn": "vcChangeDate",
    "timeColumnType": "timestamp",
    "where": [
        {
            "name": "$__timeFilter",
            "params": [],
            "type": "macro"
        }
    ]
}

],
"title": "Last",
"type": "gauge"
},
{
    "datasource": {
        "type": "mysql",
        "uid": "TahU9ACnk"
    },
    "fieldConfig": {
        "defaults": {
            "mappings": [],
            "thresholds": {
                "mode": "percentage",
                "steps": [
                    {
                        "color": "green",
                        "value": null
                    },
                    {

```

```

        "color": "orange",
        "value": 70
    },
    {
        "color": "red",
        "value": 85
    }
]
}
},
"overrides": []
},
"gridPos": {
    "h": 7,
    "w": 5,
    "x": 19,
    "y": 0
},
"id": 20,
"options": {
    "orientation": "auto",
    "reduceOptions": {
        "calcs": [
            "sum"
        ],
        "fields": "",
        "values": false
    },
    "showThresholdLabels": false,
    "showThresholdMarkers": true
},
"pluginVersion": "8.3.6",
"targets": [

```

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		15

```

{
  "datasource": {
    "type": "mysql",
    "uid": "TahU9ACnk"
  },
  "format": "time_series",
  "group": [],
  "metricColumn": "none",
  "rawQuery": true,
  "rawSql": "SELECT \n          cvf.DateVote as \"time\", \n count(DISTINCT cvf.sPersonalityId) as \"value\" \n          FROM \n cVoteFact cvf \n          INNER JOIN cVoteTerm cvt ON\n (cvf.cVoteTermId = cvt.cVoteTermId) \n          WHERE \n cvt.cVoteTermId = 14 \n          GROUP BY \n          cvf.DateVote",
  "refId": "A",
  "select": [
    [
      {
        "params": [
          "AgreementId"
        ],
        "type": "column"
      }
    ]
  ],
  "table": "Agreement",
  "timeColumn": "vcChangeDate",
  "timeColumnType": "timestamp",
  "where": [
    {
      "name": "$__timeFilter",
      "params": [],
      "type": "macro"
    }
  ]
}

```

```

        ]
    }
],
"title": "Total",
"type": "gauge"
},
{
    "datasource": {
        "type": "mysql",
        "uid": "TahU9ACnk"
    },
    "description": "",
    "fieldConfig": {
        "defaults": {
            "color": {
                "mode": "continuous-GrYlRd"
            },
            "mappings": [],
            "thresholds": {
                "mode": "absolute",
                "steps": [
                    {
                        "color": "green",
                        "value": null
                    },
                    {
                        "color": "red",
                        "value": 80
                    }
                ]
            }
        },
        "unit": "int"
    },

```

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		17

```

    "overrides": [],
  },
  "gridPos": {
    "h": 9,
    "w": 24,
    "x": 0,
    "y": 7
  },
  "id": 12,
  "options": {
    "displayMode": "lcd",
    "orientation": "auto",
    "reduceOptions": {
      "calcs": [
        "lastNotNull"
      ],
      "fields": "",
      "values": false
    },
    "showUnfilled": true
  },
  "pluginVersion": "8.3.6",
  "targets": [
    {
      "datasource": {
        "type": "mysql",
        "uid": "TahU9ACnk"
      },
      "format": "time_series",
      "group": [],
      "hide": false,
      "metricColumn": "none",
      "rawQuery": true,

```

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		18

```

        "rawSql": "SELECT \n
timestampadd(DAY,cvfc.cVoteTermId, now()) as \"time\", \n
count(DISTINCT cvf.sPersonalityId) as \"\", \n
concat(cvt.StudyYear, ' ', cvf.cVoteTermId) as mode \n      FROM
\n      cVoteFact cvf inner join cVoteTerm cvt on
cvt.cVoteTermId = cvf.cVoteTermId \n      GROUP BY \n
cvf.cVoteTermId, \n      cvt.StudyYear",

        "refId": "A",

        "select": [

            [

                {

                    "params": [

                        "AgreementId"

                    ],

                    "type": "column"

                }

            ]

        ],

        "table": "Agreement",

        "timeColumn": "vcChangeDate",

        "timeColumnType": "timestamp",

        "where": [

            {

                "name": "$__timeFilter",

                "params": [],

                "type": "macro"

            }

        ]

    },

    "title": "Panel Title",

    "type": "bargauge"

},

{

    "datasource": {

```

```

    "type": "mysql",
    "uid": "TahU9ACnk"
  },
  "fieldConfig": {
    "defaults": {
      "mappings": [],
      "thresholds": {
        "mode": "percentage",
        "steps": [
          {
            "color": "red",
            "value": null
          },
          {
            "color": "yellow",
            "value": 50
          },
          {
            "color": "light-green",
            "value": 80
          },
          {
            "color": "dark-green",
            "value": 100
          }
        ]
      },
      "unit": "short"
    },
    "overrides": []
  },
  "gridPos": {
    "h": 8,

```

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		20

```

        "w": 24,
        "x": 0,
        "y": 16
    },
    "id": 10,
    "options": {
        "colorMode": "background",
        "graphMode": "none",
        "justifyMode": "auto",
        "orientation": "auto",
        "reduceOptions": {
            "calcs": [
                "lastNotNull"
            ],
            "fields": "",
            "values": false
        },
        "text": {
            "titleSize": 12
        },
        "textMode": "auto"
    },
    "pluginVersion": "8.3.6",
    "targets": [
        {
            "datasource": {
                "type": "mysql",
                "uid": "TahU9ACnk"
            },
            "format": "time_series",
            "group": [],
            "metricColumn": "none",
            "rawQuery": true,

```

```

"rawSql": "SELECT\n  now() as \"time\", \n
T1_1.ccGroup1CourseKurator / T1.ccGroup1Course * 100 as \"1
kypc\", \n
T2_1.ccGroup2CourseKurator / T2.ccGroup2Course * 100
as \"2 kypc\", \n
T3_1.ccGroup3CourseKurator / T3.ccGroup3Course
* 100 as \"3 kypc\", \n
T4_1.ccGroup4CourseKurator /
T4.ccGroup4Course * 100 as \"4 kypc\", \n
ds.Abbreviation\nFROM\n  DcSubdivision ds\n  INNER JOIN
DcSubdivision ds1 ON (ds1.ParentId = ds.DcSubdivisionId)\n
INNER JOIN DcSubdivision ds2 ON (ds2.ParentId =
ds1.DcSubdivisionId AND ds2.DcSubdivisionLevel = 4)\n  LEFT
JOIN\n  (\n    SELECT\n      dsg.CathedralId, \n
count(RtStudyGroupId) as ccGroup1Course\n    FROM\n
DcStudyGroup dsg\n    LEFT JOIN RtStudyGroup rsg ON
(rsg.DcStudyGroupId = dsg.DcStudyGroupId)\n    WHERE\n
rsg.StudyCourse = 1\n    AND DcStudyingYearId = 8\n    GROUP
BY\n    dsg.CathedralId\n  ) T1 ON (ds2.DcSubdivisionId =
T1.CathedralId)\n  LEFT JOIN\n  (\n    SELECT\n
dsg.CathedralId, \n
count(RtStudyGroupId) as ccGroup2Course\n  FROM\n
DcStudyGroup dsg\n  LEFT JOIN RtStudyGroup rsg
ON (rsg.DcStudyGroupId = dsg.DcStudyGroupId)\n  WHERE\n
rsg.StudyCourse = 2\n  AND DcStudyingYearId = 8\n  GROUP
BY\n  dsg.CathedralId\n  ) T2 ON (ds2.DcSubdivisionId =
T2.CathedralId)\n  LEFT JOIN\n  (\n    SELECT\n
dsg.CathedralId, \n
count(RtStudyGroupId) as ccGroup3Course\n  FROM\n
DcStudyGroup dsg\n  LEFT JOIN RtStudyGroup rsg
ON (rsg.DcStudyGroupId = dsg.DcStudyGroupId)\n  WHERE\n
rsg.StudyCourse = 3\n  AND DcStudyingYearId = 8\n  GROUP
BY\n  dsg.CathedralId\n  ) T3 ON (ds2.DcSubdivisionId =
T3.CathedralId)\n  LEFT JOIN\n  (\n    SELECT\n
dsg.CathedralId, \n
count(RtStudyGroupId) as ccGroup4Course\n  FROM\n
DcStudyGroup dsg\n  LEFT JOIN RtStudyGroup rsg
ON (rsg.DcStudyGroupId = dsg.DcStudyGroupId)\n  WHERE\n
rsg.StudyCourse = 4\n  AND DcStudyingYearId = 8\n  GROUP
BY\n  dsg.CathedralId\n  ) T4 ON (ds2.DcSubdivisionId =
T4.CathedralId)\n  \n \n LEFT JOIN\n  (\n    SELECT\n
dsg.CathedralId, \n
count(rsg.RtStudyGroupId) as
ccGroup1CourseKurator\n    FROM\n      DcStudyGroup dsg\n
LEFT JOIN RtStudyGroup rsg ON (rsg.DcStudyGroupId =
dsg.DcStudyGroupId)\n    LEFT JOIN StudyGroupMember sgm ON
(sgm.RtStudyGroupId = rsg.RtStudyGroupId)\n    WHERE\n
rsg.StudyCourse = 1\n    AND rsg.DcStudyingYearId = 8\n
AND sgm.DcProfileId = 8\n    AND sgm.vcActuality = 0\n
GROUP BY\n    dsg.CathedralId\n  ) T1_1 ON (ds2.DcSubdivisionId
= T1_1.CathedralId)\n  \n LEFT JOIN\n  (\n    SELECT\n
dsg.CathedralId, \n
count(rsg.RtStudyGroupId) as
ccGroup2CourseKurator\n    FROM\n      DcStudyGroup dsg\n
LEFT JOIN RtStudyGroup rsg ON (rsg.DcStudyGroupId =
dsg.DcStudyGroupId)\n    LEFT JOIN StudyGroupMember sgm ON
(sgm.RtStudyGroupId = rsg.RtStudyGroupId)\n    WHERE\n
rsg.StudyCourse = 2\n    AND rsg.DcStudyingYearId = 8\n
AND sgm.DcProfileId = 8\n    AND sgm.vcActuality = 0\n
GROUP BY\n    dsg.CathedralId\n  ) T2_1 ON (ds2.DcSubdivisionId

```

					KPI.IT-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		22

```
= T2_1.CathedraId)\n LEFT JOIN\n (\n SELECT\n
dsg.CathedraId,\n count(rsg.RtStudyGroupId) as
ccGroup3CourseKurator\n FROM\n DcStudyGroup dsg \n
LEFT JOIN RtStudyGroup rsg ON (rsg.DcStudyGroupId =
dsg.DcStudyGroupId)\n LEFT JOIN StudyGroupMember sgm ON
(sgm.RtStudyGroupId = rsg.RtStudyGroupId)\n WHERE\n
rsg.StudyCourse = 3\n AND rsg.DcStudingYearId = 8\n
AND sgm.DcProfileId = 8\n AND sgm.vcActuality = 0\n
GROUP BY\n dsg.CathedraId\n ) T3_1 ON (ds2.DcSubdivisionId
= T3_1.CathedraId) \n LEFT JOIN\n (\n SELECT\n
dsg.CathedraId,\n count(rsg.RtStudyGroupId) as
ccGroup4CourseKurator\n FROM\n DcStudyGroup dsg \n
LEFT JOIN RtStudyGroup rsg ON (rsg.DcStudyGroupId =
dsg.DcStudyGroupId)\n LEFT JOIN StudyGroupMember sgm ON
(sgm.RtStudyGroupId = rsg.RtStudyGroupId)\n WHERE\n
rsg.StudyCourse = 4\n AND rsg.DcStudingYearId = 8\n
AND sgm.DcProfileId = 8\n AND sgm.vcActuality = 0\n
GROUP BY\n dsg.CathedraId\n ) T4_1 ON (ds2.DcSubdivisionId
= T4_1.CathedraId) \nWHERE\n ds.DcSubdivisionLevel
= 5 \n AND ds.vcActualityType = 0\n AND ds1.vcActualityType =
0\n AND ds2.vcActualityType = 0\nGROUP BY\n
ds.Abbreviation\nORDER BY\n ds.Abbreviation\n ",
```

```
"refId": "A",
```

```
"select": [
```

```
[
```

```
{
```

```
"params": [
```

```
"AgreementId"
```

```
],
```

```
"type": "column"
```

```
}
```

```
]
```

```
],
```

```
"table": "Agreement",
```

```
"timeColumn": "vcChangeDate",
```

```
"timeColumnType": "timestamp",
```

```
"where": [
```

```
{
```

```
"name": "$__timeFilter",
```

```
"params": [],
```

```
"type": "macro"
```

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		23

```

        }
    ]
}
],
"title": "Panel Title",
"type": "stat"
},
{
    "datasource": {
        "type": "mysql",
        "uid": "TahU9ACnk"
    },
    "fieldConfig": {
        "defaults": {
            "color": {
                "mode": "palette-classic"
            },
            "custom": {
                "axisLabel": "",
                "axisPlacement": "auto",
                "barAlignment": 0,
                "drawStyle": "line",
                "fillOpacity": 0,
                "gradientMode": "none",
                "hideFrom": {
                    "legend": false,
                    "tooltip": false,
                    "viz": false
                },
                "lineInterpolation": "linear",
                "lineWidth": 1,
                "pointSize": 5,
                "scaleDistribution": {

```

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		24

```

        "type": "linear"
    },
    "showPoints": "auto",
    "spanNulls": false,
    "stacking": {
        "group": "A",
        "mode": "none"
    },
    "thresholdsStyle": {
        "mode": "off"
    }
},
"mappings": [],
"thresholds": {
    "mode": "absolute",
    "steps": [
        {
            "color": "green",
            "value": null
        },
        {
            "color": "red",
            "value": 80
        }
    ]
}
},
"overrides": []
},
"gridPos": {
    "h": 6,
    "w": 24,
    "x": 0,

```

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		25

```

        "y": 24
    },
    "id": 8,
    "options": {
        "legend": {
            "calcs": [],
            "displayMode": "list",
            "placement": "bottom"
        },
        "tooltip": {
            "mode": "single"
        }
    },
    "pluginVersion": "8.3.6",
    "targets": [
        {
            "datasource": {
                "type": "mysql",
                "uid": "TahU9ACnk"
            },
            "format": "time_series",
            "group": [],
            "metricColumn": "none",
            "rawQuery": true,
            "rawSql": "SELECT \n            cvf.DateVote as \"time\", \n            count(DISTINCT cvf.sPersonalityId) as value \n            FROM \n            cVoteFact cvf \n            GROUP BY \n            cvf.DateVote",
            "refId": "A",
            "select": [
                [
                    {
                        "params": [
                            "AgreementId"
                        ]
                    }
                ]
            ]
        }
    ]
}

```

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		26

```

        "type": "column"
      }
    ]
  ],
  "table": "Agreement",
  "timeColumn": "vcChangeDate",
  "timeColumnType": "timestamp",
  "where": [
    {
      "name": "$__timeFilter",
      "params": [],
      "type": "macro"
    }
  ]
}

],
"title": "Графік опитування",
"type": "timeseries"
}

],
"schemaVersion": 34,
"style": "dark",
"tags": [],
"templating": {
  "list": []
},
"time": {
  "from": "2022-01-17T13:09:32.863Z",
  "to": "2022-02-06T09:45:02.203Z"
},
"timepicker": {},
"timezone": "",
"title": "ecampus.kpi.ua",

```

					КПІ.IT-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		27

```
"uid": "CEj3JY-nz",  
"version": 27,  
"weekStart": ""  
} JElmzMq7k  
  
"timepicker": {},  
"timezone": "",  
"title": "Факультет ФІОТ",  
"uid": "JElmzMq7k",
```

					КПІ.ІТ-8408.045440.03.13	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		28

					КПШ.ІТ-8408.045440.03.13	Арк.
						29
Змін.	Арк.	№ докум.	Підп.	Дата.		

Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2022 р.

**ВЕБ-СЕРВІС ДЛЯ ВІДОБРАЖЕННЯ ДАНИХ ПРО ПІДРОЗДІЛ У
СИСТЕМІ «ЕЛЕКТРОННИЙ КАМПУС»**

Програма та методика тестування

КПІ.ІТ-8408.045440.04.51

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Олексій ФІНОГЕНОВ

Нормоконтроль:

_____ Катерина ЛІЩУК

Виконавець:

_____ Юлія ГОРБИК

Київ – 2022

ЗМІСТ

1 Об'єкт випробувань	3
2 Мета тестування	4
3 Методи тестування.....	5
4 Засоби та порядок тестування.....	6

					КПІ.ІТ-8408.045490.04.51	Арк.
						2
Змін.	Арк.	№ докум.	Підп.	Дата.		

1 ОБ'ЄКТ ВИПРОБУВАНЬ

Об'єктом випробування є веб-сервіс для відображення даних про підрозділ у системі «Електронний кампус», який реалізований у вигляді клієнтської бібліотеки та накопичувача необхідних даних із бази даних системи ЕК. Користувачі цієї системи мають можливість відслідковувати необхідні метрики відповідно до ролі користувача і переглядати їх у вигляді графіків.

					КПІ.ІТ-8408.045490.04.51	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

2 МЕТА ТЕСТУВАННЯ

Метою тестування є наступне:

- перевірка правильності роботи програмного забезпечення у відповідно до функціональних вимог;
- перевірка налаштувань системи;
- перевірка підрахункового модулю статистичних показників;
- коректність та стабільність роботи системи;
- наявність всіх необхідних метрик;
- знаходження проблем та недоліків з метою їх усунення;
- перевірка зручності графічного інтерфейсу.

					КПІ.ІТ-8408.045490.04.51	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

3 МЕТОДИ ТЕСТУВАННЯ

Для тестування програмного забезпечення використовуються такі методи:

- функціональне тестування, за допомогою якого можна перевірити правильність виконання поставлених задач та відповідність усім вимогам до програмного продукту;
- тестування зручності використання;
- тестування інтерфейсу.

					КПІ.ІТ-8408.045490.04.51	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		5

4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Тестування виконується вручну, з метою знаходження помилок та недоліків як у функціональній частині програмного забезпечення так і в зручності в користуванні. Для того, щоб перевірити працездатність та відмовостійкість додатку, необхідно провести наступні тестування:

- динамічне тестування на відповідність функціональним вимогам;
- тестування на виведення повідомлень про помилку, коли це необхідно;
- тестування працездатності програми у випадку відсутності з'єднання до мережі;
- тестування інтерфейсу користувача;
- тестування стабільності роботи при різних зовнішніх умовах;
- тестування зручності використання.

					КПІ.ІТ-8408.045490.04.51	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		6

Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2022 р.

**ВЕБ-СЕРВІС ДЛЯ ВІДОБРАЖЕННЯ ДАНИХ ПРО ПІДРОЗДІЛ У
СИСТЕМІ «ЕЛЕКТРОННИЙ КАМПУС»**

Керівництво користувача

КПІ.ІТ-8408.045440.05.34

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Олексій ФІНОГЕНОВ

Нормоконтроль:

_____ Катерина ЛІЩУК

Виконавець:

_____ Юлія ГОРБИК

Київ – 2022

ЗМІСТ

1 Загальні відомості	3
2 Підготовка до роботи.....	4
2.1 Системні вимоги для коректної роботи	4
2.2 Завантаження застосунку.....	4
2.3 Перевірка коректної роботи.....	4
3 Робота із застосунком.....	5

					КПІ.ІТ-8408.045440.05.34	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		2

1 ЗАГАЛЬНІ ВІДОМОСТІ

Веб-сервіс для відображення даних про підрозділ у системі «Електронний кампус», який реалізований у вигляді веб-додатку та накопичувача необхідних даних із бази даних системи ЕК. Користувачі цієї системи мають можливість відслідковувати необхідні метрики відповідно до ролі користувача і переглядати їх у вигляді графіків.

					КПІ.ІТ-8408.045440.05.34	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

2 ПІДГОТОВКА ДО РОБОТИ

2.1 Системні вимоги для коректної роботи

Для успішної роботи даного застосунку необхідне виконання наступних вимог:

- наявність персонального комп'ютера зі встановленим сучасним браузером;
- наявність доступу до Інтернету;

2.2 Завантаження застосунку

На даний момент системою можна користуватись власноруч анонімно або авторизувавшись за виданими даними адміністратором, просто використовуючи веб інтерфейс. Для цього потрібно зайти на головну сторінку сайту системи за адресою <https://monitoring.campus.kpi.ua/d/CEj3JY-nz/ecampus-kpi-ua?orgId=1>.

2.3 Перевірка коректної роботи

Якщо система працює коректно, то перейшовши за посиланням має відобразитись головна сторінка дашборду, яка доступна для анонімних користувачів.

					КПІ.ІТ-8408.045440.05.34	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

3 РОБОТА ІЗ ЗАСТОСУНКОМ

3.1.1 Загальнодоступний дашборд

Для доступу до загальнодоступного дашборду достатньо просто перейти за URL веб застосунку. Для налаштування проміжку дати, за якою відображаються дані на графіках потрібно встановити її див. рис. 4.2.



Рисунок 1.1 – Загальнодоступний дашборд

3.1.2 Авторизація

Так як реєстрації у веб застосунку немає, дані для входу можна отримати тільки від адміністратора системи, який має надати відповідні права доступу. Для входу необхідно ввести електронну адресу або юзернейм та пароль.

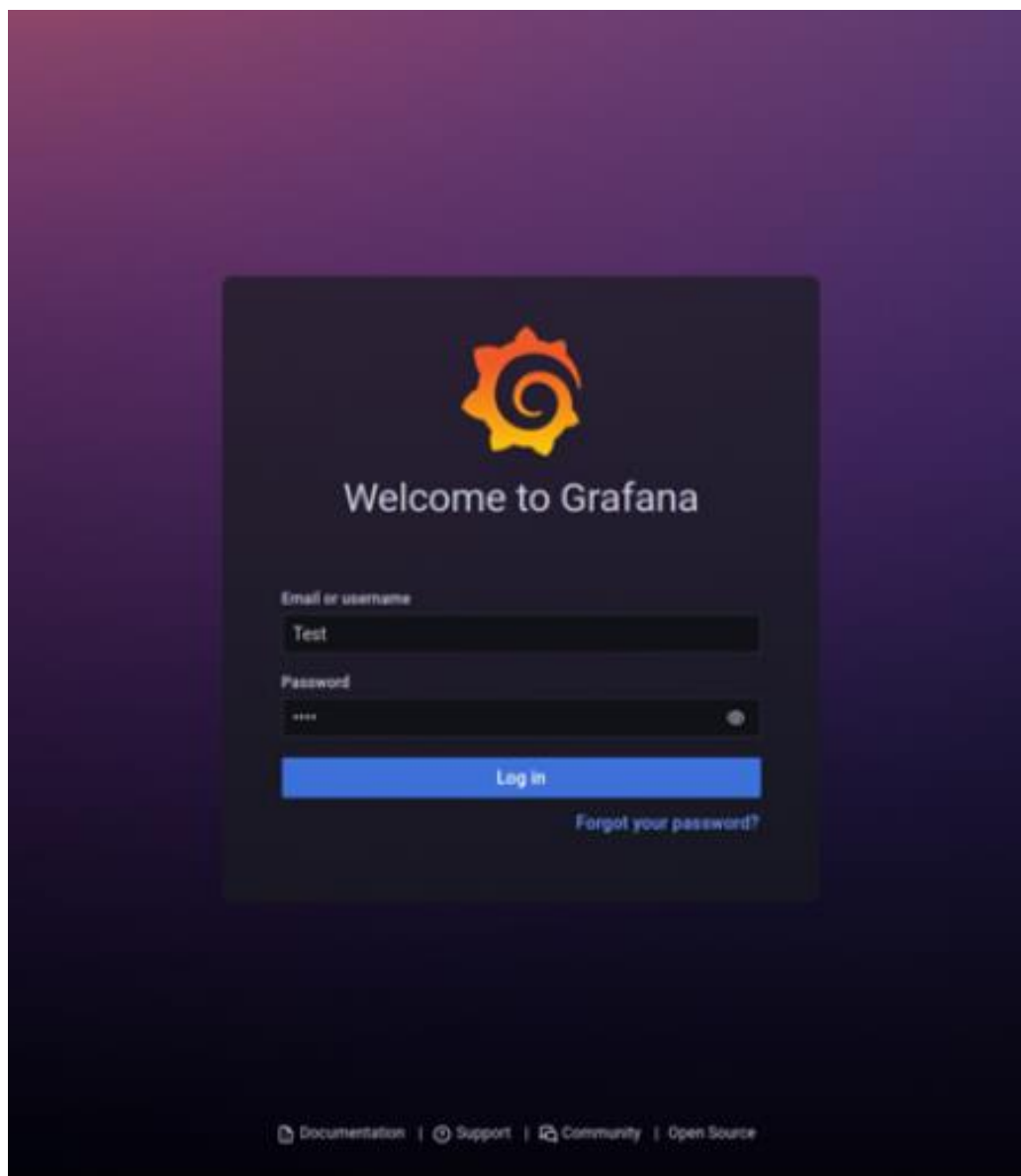


Рисунок 1.2 – Сторінка авторизації

3.1.3 Дашборд

На сторінці дашборда відображено наступні графіки згідно із назначеним рівнем доступу:

- Поточна кількість студентів, які зараз навчаються
- Хід голосування (кількість проголосувавших) у вигляді графіку часу від часу
- Кількість проголосувавших (спідометр)
- Відсоток назначених кураторів групам по курсах (температурна діаграма)

					КПІ.ІТ-8408.045440.05.34	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		6

Для налаштування проміжку дати, за якою відображаються дані на графіках потрібно встановити її див. рис. 4.4.

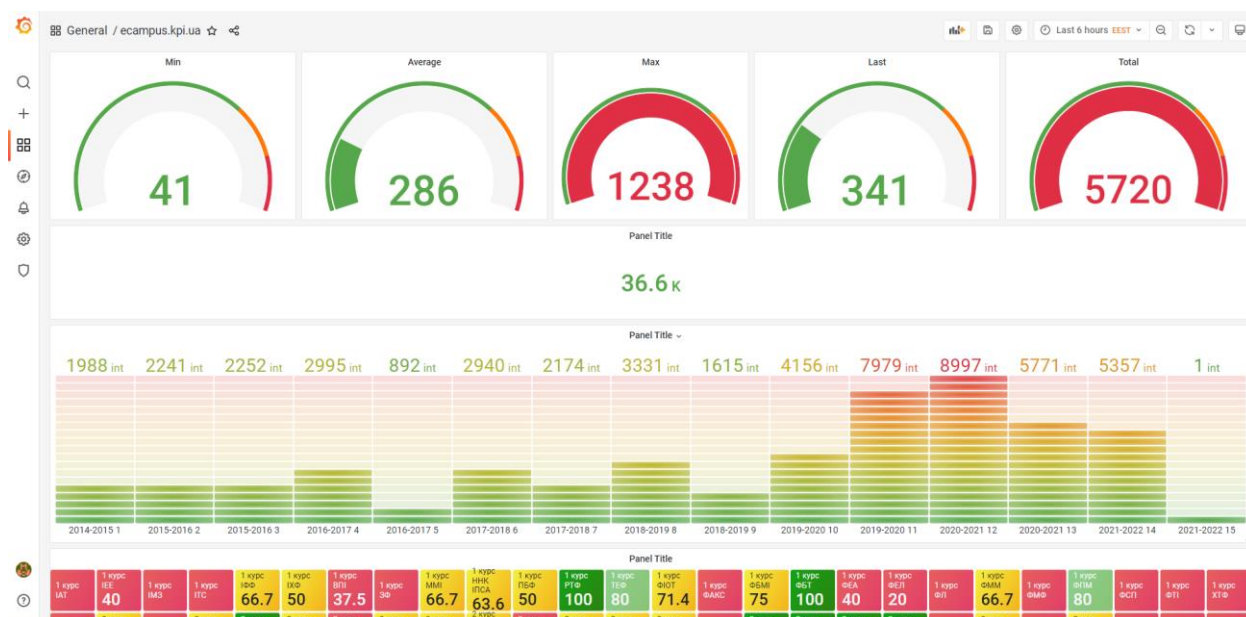


Рисунок 1.3 – Дашборд університету

3.1.4 Менеджмент користувачів

Сторінка доступна тільки користувачу із роллю адміністратора. Для створення користувача необхідно перейти до розділу Сервер адміністратор та знаходитись у вкладці Користувачі. Далі необхідно натиснути на кнопку “Новий користувач”. Після натискання з’явиться форма із полями:

- ім’я
- електронна пошта
- юзернейм
- пароль

Необхідно заповнити поля та натиснути кнопку Створити користувача. Нового користувача збережено та відображено у списку користувачів.

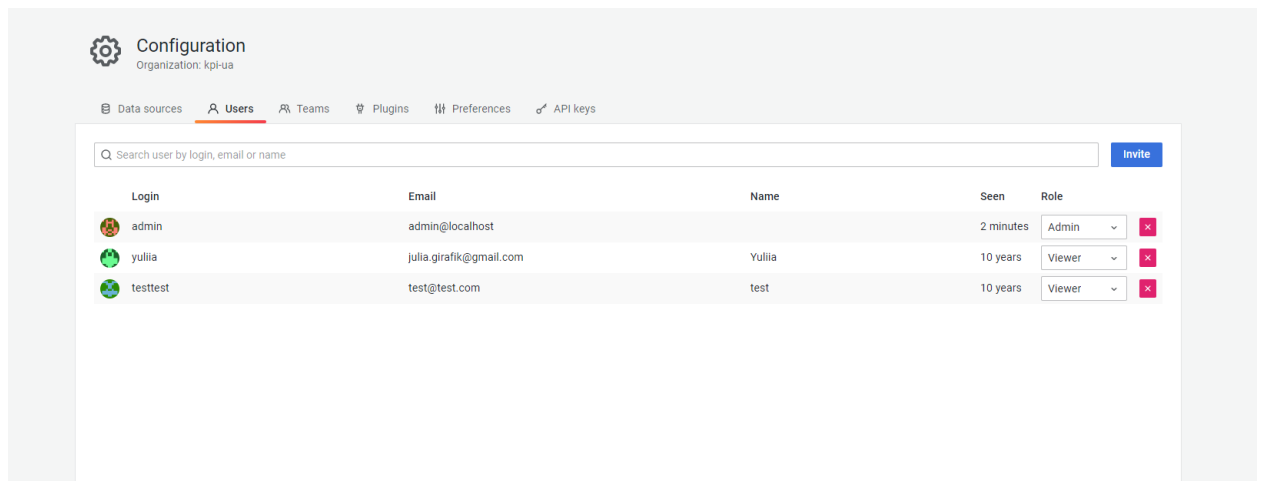


Рисунок 1.4 – Сторінка менеджменту користувачів

3.1.5 Призначення ролі доступу

Для видачі ролі користувачу необхідно натиснути на нього на сторінці Сервер адміністрування у вкладці Користувачі. Натиснути на кнопку Додати користувача до організації. З'явиться форма, де необхідно обрати організацію (для декана має бути обраний факультет, для ректора організація університет, для завідуючого кафедри – кафедра) та роль, де обов'язково має бути обрано Переглядач. Зберегти зміни можна натиснувши на кнопку Додати користувача до організації. Якщо користувача було додано помилково необхідно натиснути на кнопку Видалити користувача із організації, яка знаходиться справа навпроти відповідної організації.

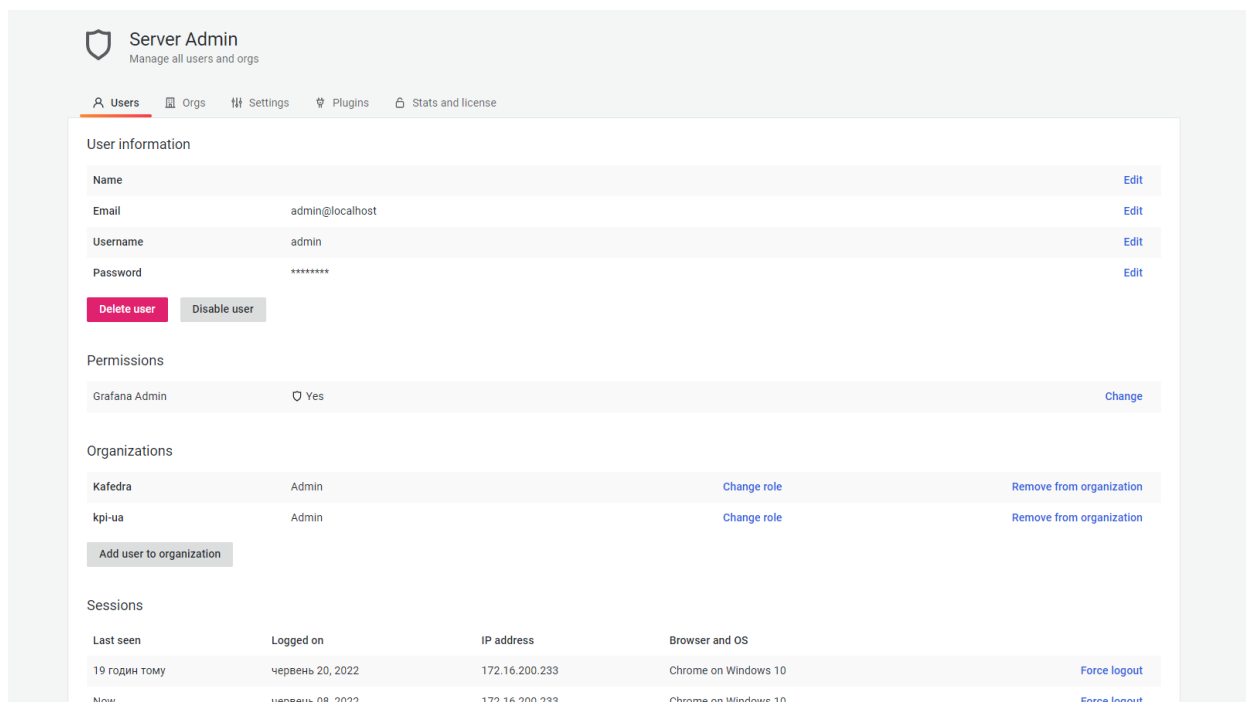


Рисунок 1.5 – Призначення ролі доступу

3.1.6 Редагування графіків дашбордів

Редагування дашбордів доступне тільки для користувача із роллю адміністратор. Щоб відредагувати графік необхідно натиснути на нього та в випадаючому меню вибрати опцію Редагувати. Після чого можна змінити запит до датасорсу, назву графіку, вигляд, у якому дані візуалізуються і т.д. Для збереження змін необхідно натиснути на кнопку Застосувати у верхньому правому куті.

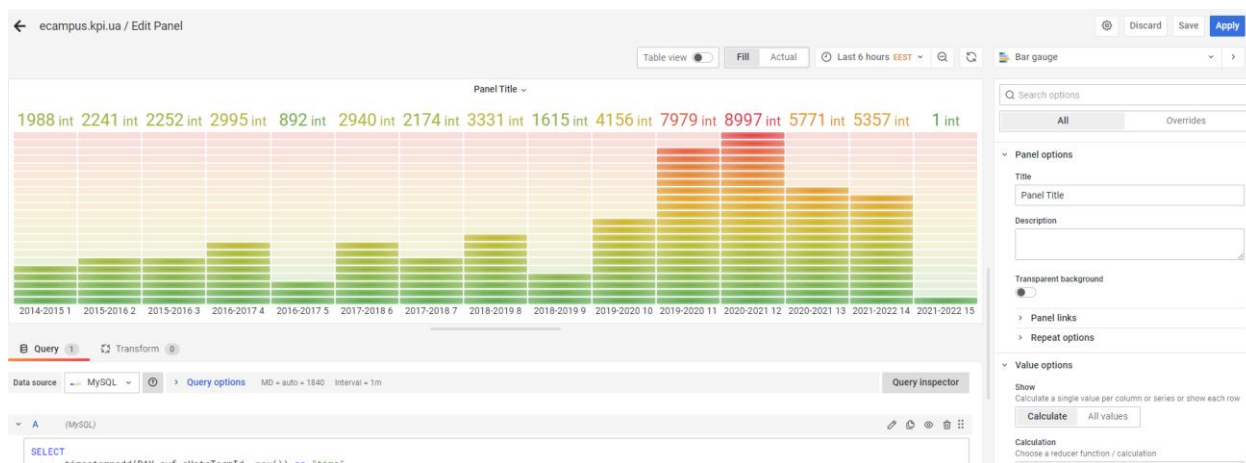


Рисунок 1.6 – Редагування графіка

3.1.7 Створення графіків дашборду

За необхідності є можливість додати графік до дашборду. Для цього необхідно перейти у дашборд та натиснути на кнопку Додати панель у верхньому правому куті. Далі треба перевірити чи встановлений по дефолту датасорс обрано коректно та написати запит до Prometheus для відображення відповідних даних. Далі необхідно назвати графік та обрати вид відображення.

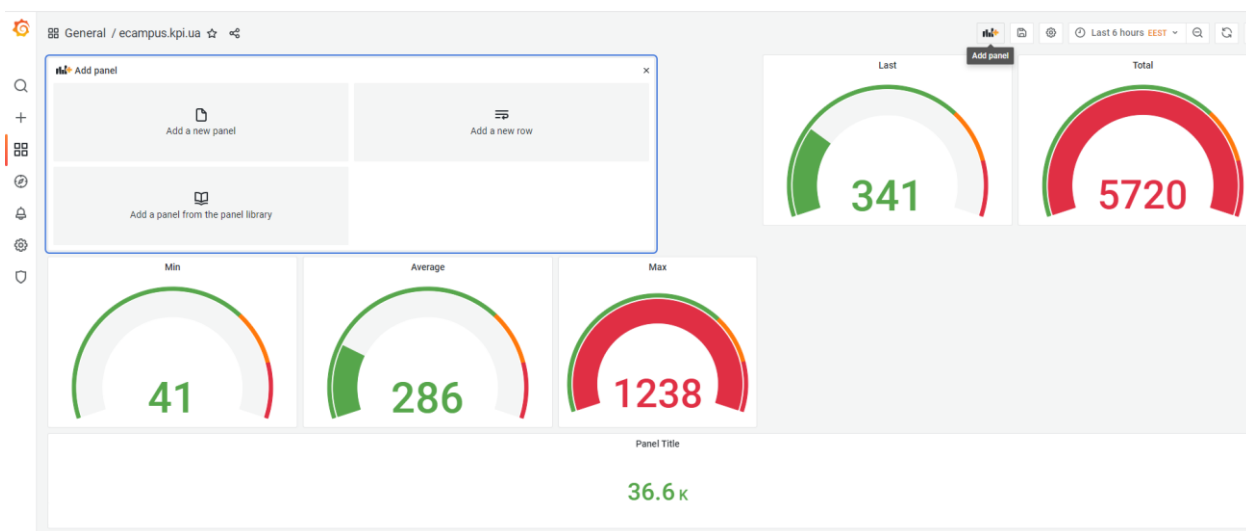


Рисунок 1.7 – Створення нового графіка

Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2022 р.

**ВЕБ-СЕРВІС ДЛЯ ВІДОБРАЖЕННЯ ДАНИХ ПРО ПІДРОЗДІЛ У
СИСТЕМІ «ЕЛЕКТРОННИЙ КАМПУС»**

Графічний матеріал

КПІ.IT-8408.045490.05.99

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Олексій ФІНОГЕНОВ

Нормоконтроль:

_____ Катерина ЛІЩУК

Виконавець:

_____ Юлія ГОРБИК

Київ – 2022

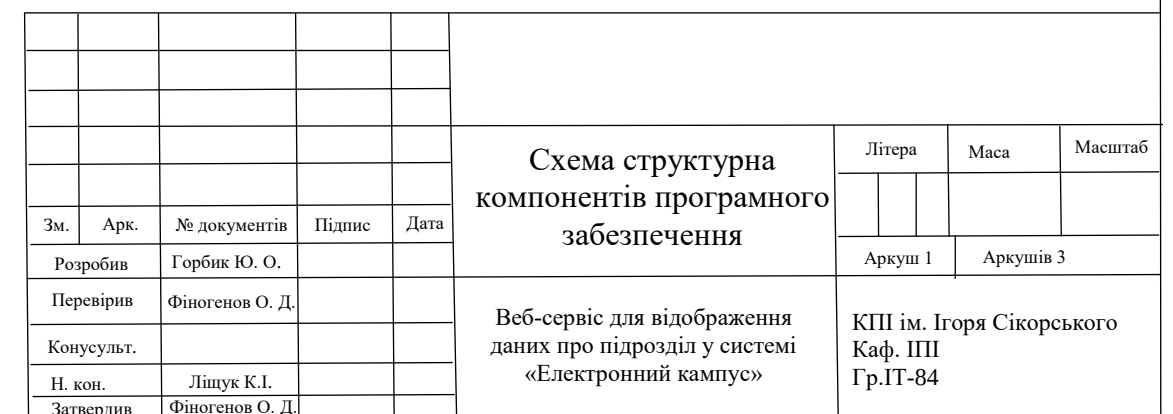
AmmountOfKurators
Abbreviation
Proc1Course
Proc2Course
Proc3Course
Proc4Course

VoteProcess
cVoteTermID
StudyYear
VoteNumber
DateVote
DateStart
DateEnd
ccVote

VoteStatistics
cVoteTermID
StudyYear
VoteNumber
DateVote
DateStart
DateEnd
ccVote

ActiveStudents
Active students

					Схема бази даних			Літера		Маса	Масштаб	
								Аркуш 1		Аркушів 3		
Зм.	Арк.	№ документів	Підпис	Дата	Веб-сервіс для відображення даних про підрозділ у системі «Електронний кампус»			КПІ ім. Ігоря Сікорського Каф. ІІІ Гр.ІТ-84				
Розробив		Горбик Ю..О.										
Перевірив		Фіногенов О.Д.										
Консуульт.												
Н. кон.		Ліщук К.І.										
Затвердив		Фіногенов О. Д.										





Имя пользователя:
Лісовиченко Олег Іванович

Дата проверки:
21.06.2022 08:39:16 EEST

Дата отчета:
21.06.2022 08:39:29 EEST

ID проверки:
1011624797

Тип проверки:
Doc vs Internet + Library

ID пользователя:
76913

Название файла: IT-84_Горбик_ПЗ

Количество страниц: 55 Количество слов: 6288 Количество символов: 49962 Размер файла: 3.04 MB ID файла: 1011492628

11.4% Совпадения

Наибольшее совпадение: 8.14% с источником из Библиотеки (ID файла: 1011466185)

0.33% Источники из Интернета 2 Страница 57

11.4% Источники из Библиотеки 100 Страница 57

0% Цитат

Исключение цитат выключено

Исключение списка библиографических ссылок выключено

0% Исключений

Нет исключенных источников