

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

**КУРСОВА РОБОТА**  
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Бази даних»  
на тему: «**БАЗА ДАНИХ САЙТУ ТЕЛЕКАНАЛУ**»

студента II курсу групи ІПЗ-20-3  
спеціальності 121 «Інженерія програмного  
забезпечення»

Гаєвський Денис Олегович  
(прізвище, ім'я та по-батькові)

Керівник: к.т.н., доц. Сугоняк.І.І.

Дата захисту: " \_ " \_\_\_\_\_ 2024р.

Національна шкала \_\_\_\_\_

Кількість балів: \_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

Члени комісії

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
(підпис)

О.В. Коротун  
(прізвище та ініціали)

С.М. Кравченко  
(прізвище та ініціали)

О.В. Чижмотря  
(прізвище та ініціали)

Житомир – 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»  
Факультет інформаційно-комп'ютерних технологій  
Кафедра інженерії програмного забезпечення  
Освітній рівень: бакалавр  
Спеціальність 121 «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

В.о.зав. кафедри ІПЗ

\_\_\_\_\_ А.В.Морозов

“ \_\_\_\_\_ ” \_\_\_\_\_ 2024 р.

ЗАВДАННЯ  
НА КУРСОВУ РОБОТУ СТУДЕНТУ  
Гаєвському Денису Олеговичу

1. Тема роботи: **База даних сайту телеканалу**,  
керівник роботи: к.т.н., доц. Сугоняк І.І.
2. Строк подання студентом: “1” лютого 2024р.
3. Вихідні дані до роботи: Розробити базу даних для сайту телеканалу
4. Зміст розрахунково-пояснювальної записки(перелік питань. Які підлягають розробці)
  - Аналіз інформаційних потоків та особливостей предметної області дослідження
  - Проектування бази даних за напрямком курсової роботи
  - Реалізація підсистеми обробки даних за напрямком курсової роботи
  - Адміністрування баз даних
5. Перелік графічного матеріалу(з точним зазначенням обов'язкових креслень)
  1. Презентація до КР: [файл](#)
  2. Посилання на репозиторій: [https://github.com/dengaevsky/BD\\_TETaTET](https://github.com/dengaevsky/BD_TETaTET)
6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посади консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1,2,3	Сугоняк І.І., к.т.н., доц.		

7. Дата видачі завдання “ 10 ” лютого 2022 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/П	Назва етапів курсового проекту	Строк виконання етапів проекту	Примітки
1	Постановка задачі	10.02.2022	Виконано
2	Пошук, огляд та аналіз аналогічних розробок	20.12.2023	Виконано
3	Формулювання технічного завдання	04.12.2023	Виконано
4	Опрацювання літературних джерел	08.12.2023	Виконано
5	Проектування бази даних	28.01.2024	Виконано
6	Написання програмного коду	13.01.2024	Виконано
7	Відлагодження	26.01.2024	Виконано
8	Написання пояснювальної записки	31.01.2024	Виконано
9	Захист		

**Студент**

\_\_\_\_\_

(підпис)

**Гаєвський Д. О.**

(прізвище та ініціали)

**Керівник проекту**

\_\_\_\_\_

(підпис)

**Сугодняк І.І.**

(прізвище та ініціали)

## РЕФЕРАТ

Завданням на курсову роботу було створити базу даних до сайту телеканалу.

Пояснювальна записка до курсового проекту (роботи) на тему “Сайт телеканалу” складається з вступу, чотирьох розділів, висновків, списку використаної літератури та додатку.

Текстова частина викладена на 49 сторінках друкованого тексту.

Пояснювальна записка має 9 сторінок додатків. Список використаних джерел містить 10 найменувань і займає 1 сторінку. В роботі наведено 27 рисунків. Загальний обсяг роботи – 58 сторінок.

Ключові слова: PHPMYADMIN, JAVASCRIPT, КЛІЄНТ, АДМІН, АВТОРИЗАЦІЯ, ЧАТ, КОМЕНТАРІ, РОЛІ, ДАНІ, МОДЕЛЬ ДАНИХ, БАЗА ДАНИХ, КЛЮЧОВЕ ПОЛЕ, ЗАПИТ.

					ДУ «Житомирська політехніка».24.121.05.000 - ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.	Гаєвський Д.О.				База даних телеканалу "ТЕТАТЕТ"	Літ.	Арк.	Аркушів
Керівник	Суходняк І.І.						4	60
Н. контр.								
Зав. каф.						ФІКТ Гр. ІПЗ-20-3[1]		

## ЗМІСТ

ВСТУП .....	7
РОЗДІЛ 1. ТЕОРЕТИЧНИЙ АНАЛІЗ ІНФОРМАЦІЙНИХ ПОТОКІВ ТА ОСОБЛИВОСТЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ .....	7
1.1 Аналіз інформаційних потреб та визначення предметної області дослідження .....	7
1.2 Архітектура та засоби реалізації бази даних .....	9
Висновки до першого розділу .....	15
РОЗДІЛ 2. ПРОЕКТУВАННЯ БАЗИ ДАНИХ КАРТИННОЇ ГАЛЕРЕЇ .....	16
2.1 Аналіз структури інформаційних процесів адміністраторів .....	16
2.2 Проектування бази даних галереї .....	18
2.3 Розробка математичної моделі та алгоритмів обробки інформації про галерею .....	21
Висновки до другого розділу .....	22
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ КАРТИННОЇ ГАЛЕРЕЯ .....	23
3.1 Проектування інтерфейсу системи картинної галереї .....	23
3.2. Реалізації операцій обробки даних в БД .....	33
Висновок до третього розділу .....	39
РОЗДІЛ 4. АДМІНІСТРУВАННЯ БАЗ ДАНИХ .....	40
4.1 Розробка заходів захисту інформації в БД .....	40
4.2 Налаштування параметрів роботи Firebase .....	43
Висновок до четвертого розділу .....	48
ВИСНОВКИ .....	49
СПИСОК ЛІТЕРАТУРИ .....	50
ДОДАТКИ .....	51

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ:

ТЗ – Технічне завдання

ПКМ – права клавіша миши

Рис – рисунок

СУБД – система управління базою даних

БД - база даних

Б-ка – бібліотека

ІС – інформаційна система

СУБД – система управління базою даних

CRUD – create, read, update, delete.

ПЗ – Програмне забезпечення

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сугодняк І. І.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

# РОЗДІЛ 1. ТЕОРЕТИЧНИЙ АНАЛІЗ ІНФОРМАЦІЙНИХ ПОТОКІВ ТА ОСОБЛИВОСТЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ

## 1.1 Аналіз інформаційних потреб та визначення предметної області дослідження

В даному курсовому проекті за предметну область взято базу даних, яка буде зберігати усю необхідну інформацію для сайту телеканалу. Цей сайт надає можливість розмістити новину, перейшовши за яким, можна її переглянути детальніше. Розмістити проекти, які показуються на цьому телеканалі, телепрограму, розписану на тиждень.

. Щоб користувачі мали змогу одержати для себе більш повну інформацію про новину, залишити повідомлення в чаті чи коментар до товару, мати можливість увійти/zaregistruvatis'я і тд... необхідно створити базу даних, яка зберігатиме усі необхідні дані. А для адміністраторів, потрібно зробити можливість видаляти й редагувати окремі новини.

База даних повинна бути зручною у використанні та задовольняти усім стандартам розробки баз даних. Однією з найважливіших функцій бази даних є швидкість обробки запитів від користувача, швидке видання інформації для нього та швидке оновлення даних.

Для коректного виконання запитів на отримання даних, дані повинні вноситись та змінюватись у БД. За це будуть відповідати деякий перелік операцій:

### Реєстрація користувача

Додаток повинен розрізняти користувачів, щоб давати йому доступ до певних даних та функцій, тому потрібно, створити таблицю, з користувачами. Для заповнення цієї таблиці потрібне створення нового користувача, за це і відповідатиме дана функція.

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

### **Вхід користувача**

Логічний опис використано у попередньому абзаці (функція, що дозволить розрізняти користувачів та увійти на сайт).

### **Роль адміністратора**

Необхідна для доступу користувача до певних функцій, що доступні лише адміністраторам. Серед тих функцій є редагування та видалення новини.

### **Додавання новини**

Необхідно для того, щоб сайт міг виводити список оголошень, відкривати, шукати, редагувати та видаляти їх. Для цього функціоналу потрібно створити певну таблицю – news. Вона буде містити інформацію про новину.

### **Видалення новини**

Коли адміністратор вирішив, що дані новини є не коректними чи за інших потреб, має бути реалізована функція видалення новини.

### **Редагування новини**

Може трапитись ситуація, що потрібно буде змінити якусь інформацію про новину, наприклад десь щось не так записали, або не ту фотографію подали при збереженні, тому потрібно реалізувати дану функцію.

### **Перегляд новини**

Щоб дізнатись більше про новину, користувач має мати можливість перейти на сторінку, де знаходиться детальний опис новини.

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				8
Змн.	Арк.	№ докум.	Підпис	Дата		



### **Коментування новини**

Щоб залишити відгук про новину, авторизований користувач має мати можливість залишити коментар.

### **Додавання проекту**

Необхідно для того, щоб сайт міг виводити список телепроектів, відкривати, шукати, редагувати та видаляти їх. Для цього функціоналу потрібно створити певну таблицю – projects. Вона буде містити інформацію про проект.

### **Видалення проекту**

Коли адміністратор вирішив, що дані проекту є не коректними чи за будь-яких інших потреб вона має бути видалена, реалізована функція видалення.

### **Редагування проекту**

Може трапитись ситуація, що потрібно буде змінити якусь інформацію про проект, наприклад десь щось не так записали або не ту фотографію подали при збереженні, тому потрібно реалізувати дану функцію.

### **Перегляд проекту**

Щоб дізнатись більше про передачу, користувач має мати можливість перейти на сторінку, де знаходиться детальний опис.

### **Коментування проекту**

Щоб залишити відгук про проект, авторизований користувач має мати можливість залишити коментар.

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

### **Додавання телепрограми**

Необхідно для того, щоб на сайті була можливість переглядати графік перегляду телепередач. Для цього функціоналу потрібно створити певну таблицю – teleprogram. Вона буде містити інформацію про це.

### **Видалення телепрограми**

Коли адміністратор вирішив, що дані телепрограми є не коректними чи за будь-яких інших потреб вона має бути видалена, реалізована функція видалення.

### **Редагування телепрограми**

Може трапитись ситуація, що потрібно буде змінити якусь інформацію про телепрограму, тому потрібно реалізувати дану функцію.

### **Перегляд проекту**

Щоб дізнатись більше про передачу, користувач має мати можливість перейти на сторінку, де знаходиться детальний опис.

### **Пошук**

Щоб віднайти необхідний новину за назвою, був створений функціонал пошуку. Відсилає запит до БД, де за назвою новини знаходить потрібну.

### **Повідомлення у чаті**

Щоб залишити повідомлення у глобальному чаті.

### **Системні повідомлення**

Щоб проінформувати користувача про помилку чи про вдалі події.

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

## 1.2 Архітектура та засоби реалізації бази даних

Аналіз та вибір СУБД проведемо з урахуванням того, що число клієнтських місць може містити в собі більше тисячі користувачів, а доступ до даних має бути максимально ефективним. В таблицях наведено порівняльні характеристики СУБД.

Таблиця №1

### Функціональне порівняння СУБД

Характеристика	Microsoft SQL Server	Oracle	Firebase
Робота під керівництвом ОС	Задовільно	Добре	Відмінно
Складність	Відмінно	Задовільно	Добре
Доступ до даних	Відмінно	Добре	Добре
Використання у великих проектах	Відмінно	Відмінно	Добре
Використання у малих проектах	Добре	Добре	Відмінно
Схема даних	Відмінно	Відмінно	Добре
Підтримувані мови програмування	Відмінно	Добре	Задовільно
Підключення до Web	Відмінно	Відмінно	Відмінно
Графічні	Добре	Добре	Добре
Тригери	Відмінно	Відсутній критерій	Добре
Паралелізм	Відмінно	Відмінно	Відмінно
Одночасний доступ декількох користувачів	Добре	Добре	Відмінно
Обробка даних мультимедіа	Слабо	Відмінно	Відмінно

Пошук за регулярними виразами	Відмінно	Відмінно	Задовільно
Підтримка об'єктно-орієнтованої парадигми	Добре	Відмінно	Задовільно
Засоби аналізу	Відмінно	Відмінно	Відмінно

### Пояснення таблиці:

Крім того, що Microsoft SQL Server та Oracle зберігають дані реляційно, а Firebase використовує сховище документів, дані субд мають інші різні характеристики, більше того, ці характеристики можна оцінити, що і зображено у таблиці номер 1.

### Робота під керівництвом ОС

Серверними операційними системами для Oracle Windows, Linux, Solaris, HP-UX, OS X, тоді як для SQL Server можуть бути лише Windows та Linux, а для Firebase розміщувати не потрібно. Сервери для роботи з даною СУБД уже розміщені, що дає значну перевагу у простоті налаштування та використання створеної програмістом БД.

### Складність

Під час аналізування, вивчення та розробки за допомогою СУБД Oracle виникає багато труднощів, у той момент, як при використанні SQL Server більшість проблем виникає лише при навчанні та розуміння логіки мови, що дає даних СУБД оцінку відмінно в даному критерії. Якщо звернути увагу на простоту до Firebase, то її аналізувати тяжко. Дана СУБД містить специфіку для роботи з певним функціоналом, наприклад, для збереження медіа потрібно налаштувати свій проект, для роботи з валідацією користувачів потрібно налаштувати свій проект, однак, попри те, що потрібно багато чого налаштовувати і багато у чому

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

розбиратись, дана СУБД цього варта. Якщо Ви освоїте повністю дану нереляційн БД, то ваші проекти, які будуть містити бази, стануть значно швидше розроблятись, тому даному критерію варто поставити оцінку добре.

### **Доступ до даних**

Оцінимо те, як використовується база даних користувачами. У СУБД Oracle користувачі повністю можуть переглядати базу, у той момент як у SQL SERVER та Firebase користувач отримує доступ лише туди, куди йому його надають, однак Firebase все одно не на стільки безпечний як ORACLE.

### **Використання у великих проектах**

Оскільки для доступу до Firebase з додатку потрібно відправляти запити, ідані, що приходять від сервера, не можна запитом отримати відразу згруповано(мається на увазі відсутність аналога join в СУБД), то це може призвести до витрачання великої кількості часу та пам'яті, що не дуже добре для веб-розробки. У той момент, як за допомогою реляційних БД, можна отримувати згруповані дані, і зменшувати кількість запитів на сервер, що є великим плюсом для великих проектів, де іноді, потрібно швидко отримати дані відразу з кількох десятків, а то, і сотень таблиць.

### **Використання у малих проектах**

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

Для невеликих проектів, потрібно виконати багато роботи та шаблонних налаштувань, щоб коректно працювала база даних, розроблена на SQL Server, також потрібно зробити багато налаштувань безпеки, щоб не виникло різних казусів з витіком даних, у той момент, як Firebase зробить ці налаштування сам, і не пропустить зловмисника так просто, та й підключити Firebase у проект значно легше, ніж SQL Server чи Oracle. XPRS - З одного боку, він підтримує класичний підхід з використанням макетів і тим, з іншого дозволяє «нарізати» сторінку на блоки-смужки, які можна переміщувати, вимикати, міняти місцями тощо.

### Схема даних

Схема даних (Схема баз даних — це структура системи баз даних описана мовою, яка підтримується системою керування баз даних і відноситься до організації даних для створення плану побудови бази даних з розподілом на таблиці.). Підтримка фіксованої схеми та розгортання без схеми з можливістю взаємодії між ними реалізована лише у Oracle, у той час як в SQL Server потрібно розгортати лише з описаною схемою даних, а firebase взагалі цього не використовує.

### Підтримування мов програмування

Firebase підтримує розробку за допомогою мови C, JS та Objective-C, у той момент, як Oracle підтримує C, C#, Go, Java, JavaScript (Node.js) та Python, а SQL Server – ще більше. Це говорить про те, що firebase не дуже добре підтримується мовами програмування, однак, якщо проаналізувати мови, якими він може використовуватись, то спаде на думку, що він корисний саме у веб-розробці, яка і буде виконана в КР. Решта СУБД теж, добре себе проявляють у цій сфері розробки.

### Графічні

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

Кожна з перчислених СУБД має додаток, або сайт за допомогою якого можна керувати базою та налаштовувати її(тобто є графічні інтерфейси), кожен з даних інтерфейсів є зручним та простим у використанні(однак іноді тяжко розібратись де знайти певний функціонал та як його заповнити, без допомоги інтернету чи інших джерел інформації).

### Тригери

Звернемо увагу на тригери(події, що дозволяють перехоплювати дії при створенні, видаленні чи оновленні даних). У Firebase дана функціональність виконується та підключається автоматично, відбувається відслідковування змін у БД, та при коректному підключенні до проекту, зміни буде відразу передаватись та аналізуватись. Тригери також реалізовані у SQL SERVER, даний функціонал призначений для тих же дій, що і в Firebase, однак він може виконуватись не лише після певних змін у бд, а й замість них, що є дуже зручним, для валідацій даних. У той момент, як Oracle взагалі не реалізував дані можливості.

### Паралелізм

Підтримка одночасного маніпулювання даними(паралелізм), реалізовано у всіх СУБД, що є великим плюсом для цих СУБД.

### Одночасний доступ декількох користувачів

Кожна з даних СУБД підтримує одночасний доступ кільком користувачам, що є теж неабияким плюсом. Однак кожна з них має різні ліміти та різне розподілення ресурсів для кількох користувачів, в результаті аналізування цих критеріїв і було виставлено оцінки в таблиці 1.

### Обробка даних мультимедіа

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

Обробка медіаданих реалізована у всіх СУБД, але в SQL SERVER для цього потрібно вручну створити таблицю, яка буде зберігати не лише сам файл, а й ім'я, щоб не втратити цю інформацію.

Приклад таблиці виглядає так:

```
CREATE TABLE ImageAsVarBinary (ID int IDENTITY(1,1),  
    ImageName varChar(50), varBinaryData varBinary(MAX));
```

У той момент, як для Firebase потрібно просто реалізувати певний скрипт, що буде зберігати інформацію в Firebase/storage.

### Пошук за регулярними виразами

Щодо пошуку даних за певним значенням, що має зустрітись в тексті (регулярні вирази), то Firebase явно програє SQLServer та Oracle, так як дана СУБД не може використовувати оператор Like, однак, за допомогою певних маніпуляцій можна відфільтрувати лише дані, що будуть починатись лише з певних символів, у той момент, як дві інші СУБД можуть використовувати оператор Like і знаходити дані коректно і не лише з однієї колонки.

### Підтримка об'єктно-орієнтованої парадигми

Підтримка ООП, за допомогою мов програмування можна підганяти отримані дані в об'єкти або класи. На жаль, курсова робота не буде містити тотального використання ООП.

### Засоби аналізу

Аналізи та звіти дуже зручно реалізовані в Firebase, так як на їхньому офіційному сайті, можна зайти на проект і отримати повну інформацію про базу. Oracle теж має таку можливість, у той момент, як для SQL Server потрібно виконати певні скрипти, що не завжди є зручним та можливим. Отже можна

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				16
Змн.	Арк.	№ докум.	Підпис	Дата		



зробити висновок, Firebase варто використовувати для простих задач, а SQL Server чи Oracle краще підходить для складних.

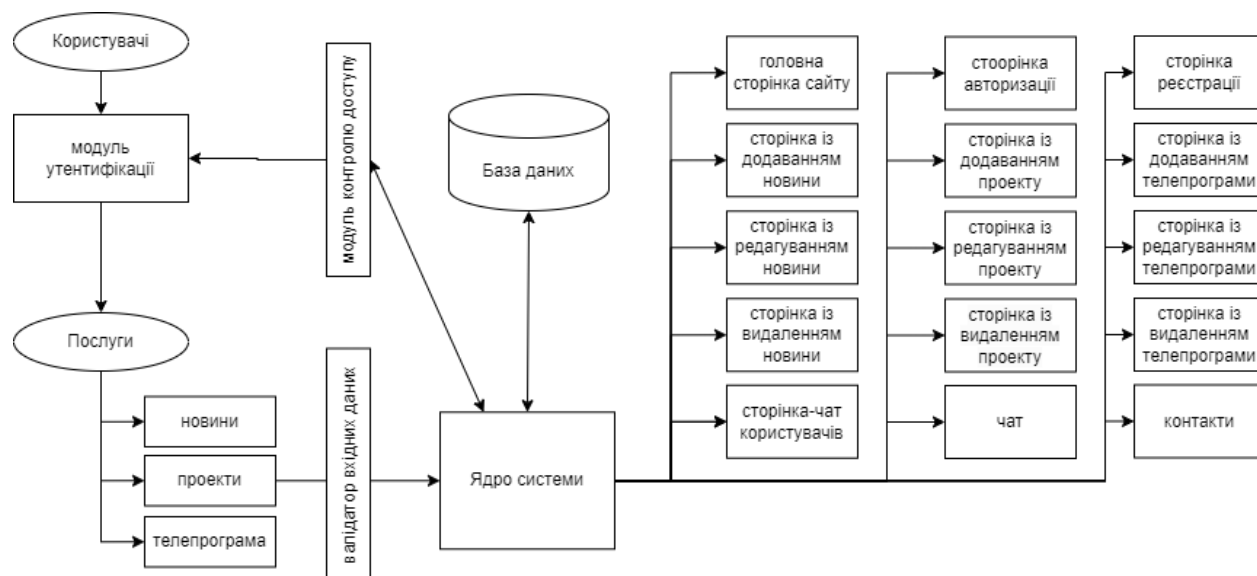
### **Висновки до першого розділу**

Це дозволить реалізувати БД якісно, надійно, швидко та написати клієнтський код до додатку швидко і не використовувати багато шаблонного коду.

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сугодняк І. І.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 2. ПРОЕКТУВАННЯ БАЗИ ДАНИХ САЙТУ ТЕЛЕКАНАЛУ

### 2.1 Аналіз структури інформаційних процесів адміністраторів



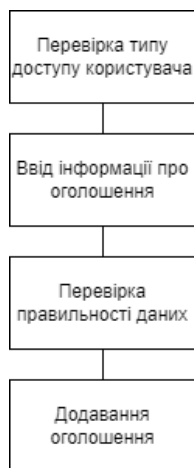
**Рис.1.** Структура сайту новинних оголошень

Користувачі будуть мати змогу працювати з веб-додатком як після аутентифікації, так і без неї. Наприклад, переглядати якісь дані, новини, телепрограми, проекти, переглядати чат, можна буде і без аутентифікації, а ось отримувати якісь більші можливості, наприклад, для звичайного користувача – це буде можливість залишити коментар чи повідомлення, виставити власне якісь новини, буде недоступним.

Далі наведено деякі приблизні структурні схеми функцій системи. Приблизні, так як під час розробки додатку може виникнути необхідність розділити деякі функції на більшу кількість функцій, для спрощення написання коду та роботи з функціоналом.



**Рис.2.** Структурна схема кількох функцій системи

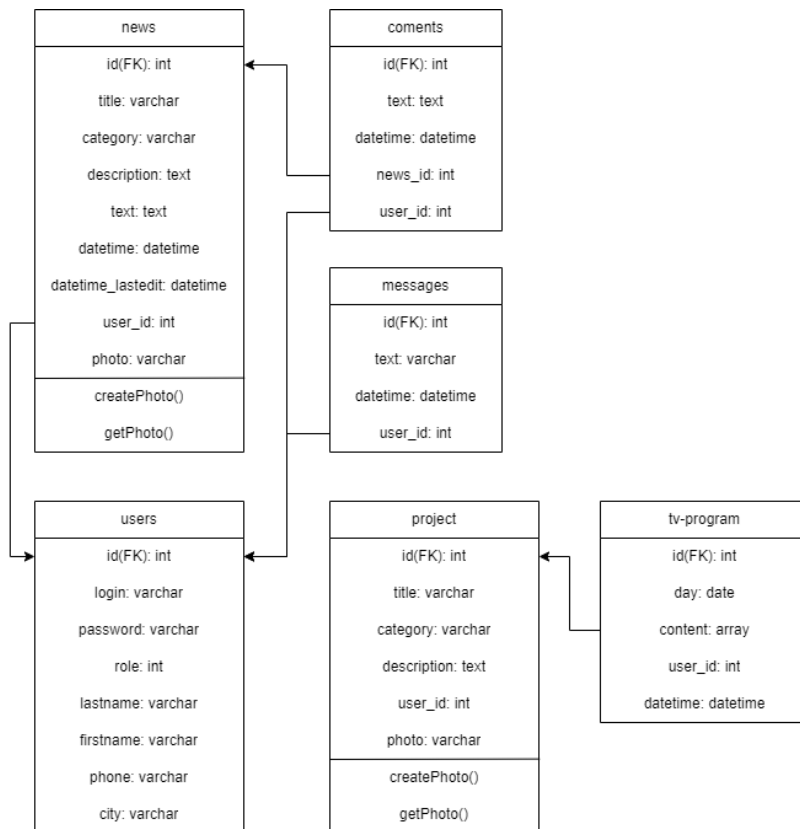


**Рис.3.** Структурна схема створення новини



**Рис.4.** Структурна схема пошуку новини

## 2.2 Проектування бази даних сайту оголошень



**Рис.5.** Діаграма „сутність-зв’язок” (логічний рівень) системи

Внаслідок проектування до бази даних включено таблиці:

- News
- Users
- Messages
- Chat
- Tv-Program
- Project

Для збереження користувачів призначена таблиця "Users ". В дану таблицю вводяться всі необхідні дані користувача. До них належать ім'я, пошта, пароль, доступ, фамілія, і тд... Структура таблиці наведена нижче:

**Таблиця 2** Структура таблиці "User"

Назва	Тип даних	ПК	ЗК	Опис поля
id	int	+	-	Id користувача
login	varchar	-	-	Ел.адреса
password	varchar	-	-	Пароль
access	int	-	-	Рівень доступу
lastname	varchar	-	-	Фамілія
firstname	varchar	-	-	Ім'я
phone	varchar	-	-	Номер телефону
city	varchar	-	-	Місто проживання

Для збереження інформації для новини призначена таблиця "News". В дану таблицю вводяться всі необхідні дані. До них належать назва, опис, текст і тд.

Структура таблиці наведена нижче:

**Таблиця 3** Структура таблиці "News"

Назва	Тип даних	ПК	ЗК	Опис поля
id	int	+	-	Id новини
title	varchar	-	-	Назва
type	varchar	-	-	Тип
description	text	-	-	Опис
text	text	-	-	Текст
datetime	datetime	-	-	Дата і час опублікування
datetime_lastedit	datetime	-	-	Дата і час редагування
photo	varchar	-	-	Фото
user_id	int	-	+	Id автора

Для збереження інформації для коментарів до оголошень призначена таблиця "Messages". В дану таблицю вводяться всі необхідні дані. До них належать коментар та дата, айді новини, айді автора.

Структура таблиці наведена нижче:

**Таблиця 4** Структура таблиці "Messages"

Назва	Тип даних	ПК	ЗК	Опис поля
id	int	+	-	Id повідомлення
message	text	-	-	Коментар
datetime	datetime	-	-	Дата і час опублікування
car_id	int	-	+	Айді новини
user_id	int	-	+	Айді автора

Для збереження інформації для повідомлень у чаті користувачів призначена таблиця "Chat". В дану таблицю вводяться всі необхідні дані. До них належать повідомлення та дата й айді автора.

Структура таблиці наведена нижче:

**Таблиця 5** Структура таблиці "Chat"

Назва	Тип даних	ПК	ЗК	Опис поля
id	int	+	-	Id повідомлення
message	text	-	-	Коментар
datetime	datetime	-	-	Дата і час опублікування
user_id	int	-	+	Айді автора

Для збереження інформації про телепередачі призначена таблиця "Project". В дану таблицю вводяться всі необхідні дані. До них належать айді, фото, заголовок, жанр, опис.

Структура таблиці наведена нижче:

**Таблиця 6** Структура таблиці "Project"

Назва	Тип даних	ПК	ЗК	Опис поля
id	int	+	-	Id
title	varchar	-	-	Назва
category	varchar	-	-	Категорія
description	varchar	-	-	Опис проекту
user_id	int	-	-	Айді автора
photo	varchar	-	-	Фото проекту

Для збереження інформації про телепрограму призначена таблиця "Teleprogram". В дану таблицю вводяться всі необхідні дані. До них належать айді, дата, айді проекту, час початку й закінчення показу, й айді автора

Структура таблиці наведена нижче:

**Таблиця 7** Структура таблиці "teleprogram"

Назва	Тип даних	ПК	ЗК	Опис поля
id	int	+	-	Id повідомлення
premiere	int	-	-	Номер тижня
Project_id	int	-	-	Айді передачі
Start_time	time	-	-	Час початку
End_time	time	-	-	Час закінчення
User_id	int	-	-	Автор

## 2.3 Розробка математичної моделі та алгоритмів обробки інформації про базу даних сайту телеканалу

Для створення нового користувача, потрібно буде пройти реєстрацію на сайті, заповнити форми із ел.адресою, фамілією і тд.. Веб-додаток перевірить, чи не існує користувача із такою самою ж електронною адресою, якщо все гаразд, відправить дані до БД.

Для створення новини, проекту, телепрограми, коментаря, повідомлення потрібно буде просто заповнити відповідні форми, після чого, дані будуть перевірені на правильність, та відправлені у спеціальний метод, що зробить додання нового документу у БД. Дані, які потрібні для новини – це назва, підзаголовок, текст, фото. Дані, які потрібні для коментаря – це сам коментар, час, айді новини та айді автора, з форми будуть приходити лише коментар, все інше створюється автоматично. Для повідомлення у чаті користувачів потрібно саме повідомлення, час й айді користувача.

Id користувача будуть створюватись автоматично.

Для повідомлення у чаті адміністраторів все аналогічно, як і в чаті звичайних користувачів.

Редагувати новини, проекти, телепередачі можуть лише автори або адміністратори.

Для оновлення даних новини, проекту чи телепрограми потрібно буде просто вибрати необхідний об'єкт та відредагувати застарілу інформацію.

Після цього дані будуть оновлені, за допомогою id документа, який буде отримуватись при перегляді форми для оновлення документа.

Автори оголошень та адміністратори можуть видалити новини, вибравши його та підтвердивши свій вибір.

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				
Змн.	Арк.	№ докум.	Підпис	Дата		24



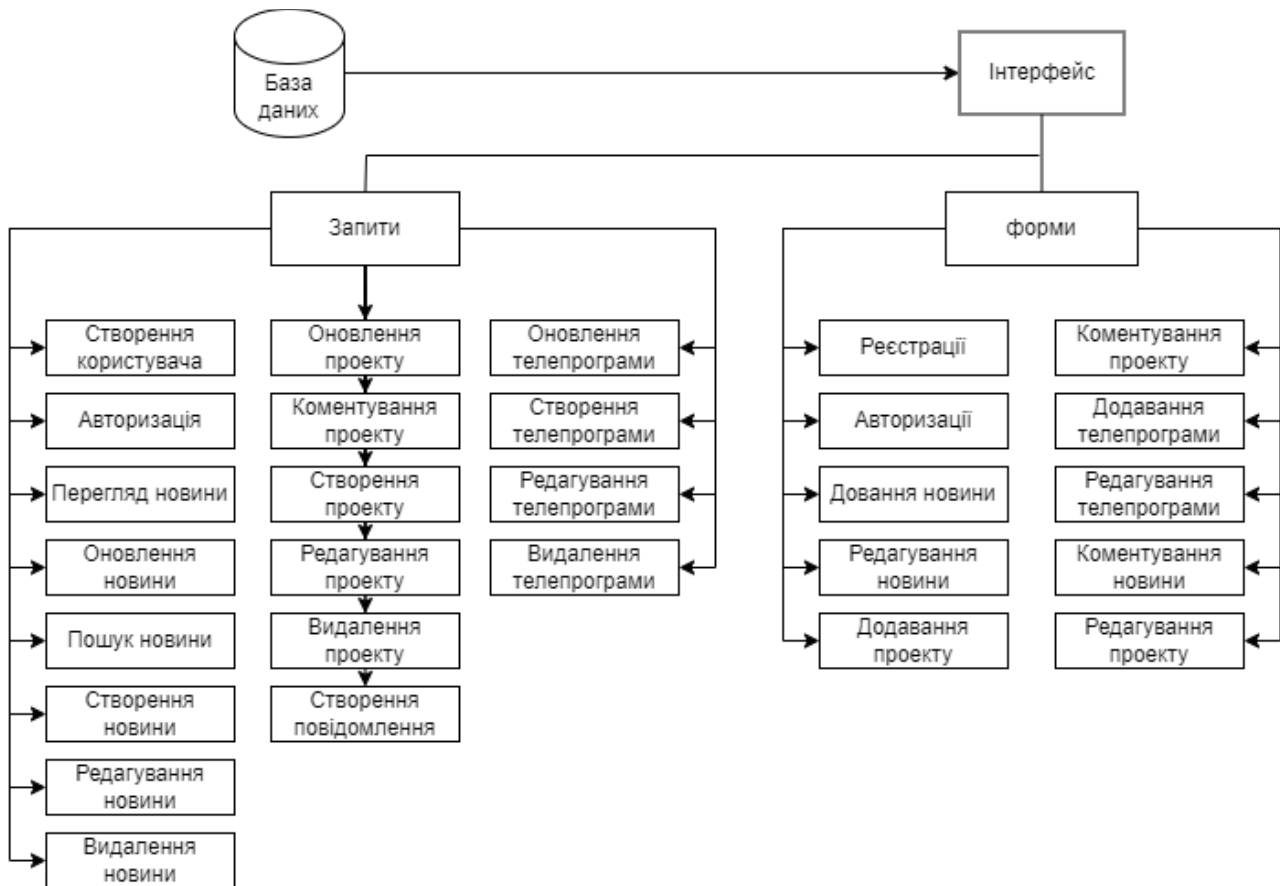
## Висновки до другого розділу

У результаті виконання даного розділу було проаналізовано інформаційні процеси, спроектовано структуру бази даних, описано приблизну логіку функцій для роботи з базою на клієнті, описано деяку логіку обробки, збереження та генерації даних для бази. Також було описано як буде відбуватись побудова та генерація статистичної інформації про сайт. Крім цього, було описано логіку взаємодії між даними

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сугодняк І. І.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 3. РЕАЛІЗАЦІЯ БАЗИ ДАНИХ САЙТУ ТЕЛЕКАНАЛУ

### 3.1. Проектування інтерфейсу системи сайту телеканалу



**Рис.6.** Структура інтерфейсу

На рисунку 6 було зображено загальну схему інтерфейсу сайту телеканалу. Далі будуть зображені форми та сторінки, що показують фінальний результат виконання проектування інтерфейсу веб-сайту. Кожна з форм чи сторінок супроводжується розробкою функцій для роботи з БД. Більшість випадків побудови форми редагування та додавання були об'єднані з метою зменшення коду.

## Форми та сторінки інтерфейсів:

Відкриваємо наш сайт за адресою: <http://tetatet/>. Відкривається головна сторінка сайту, на ній зображені основні блоки сайту: карусель, в якій із бази даних підтягуються 5 останніх оголошень, про нас, відгуки і тд...



		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

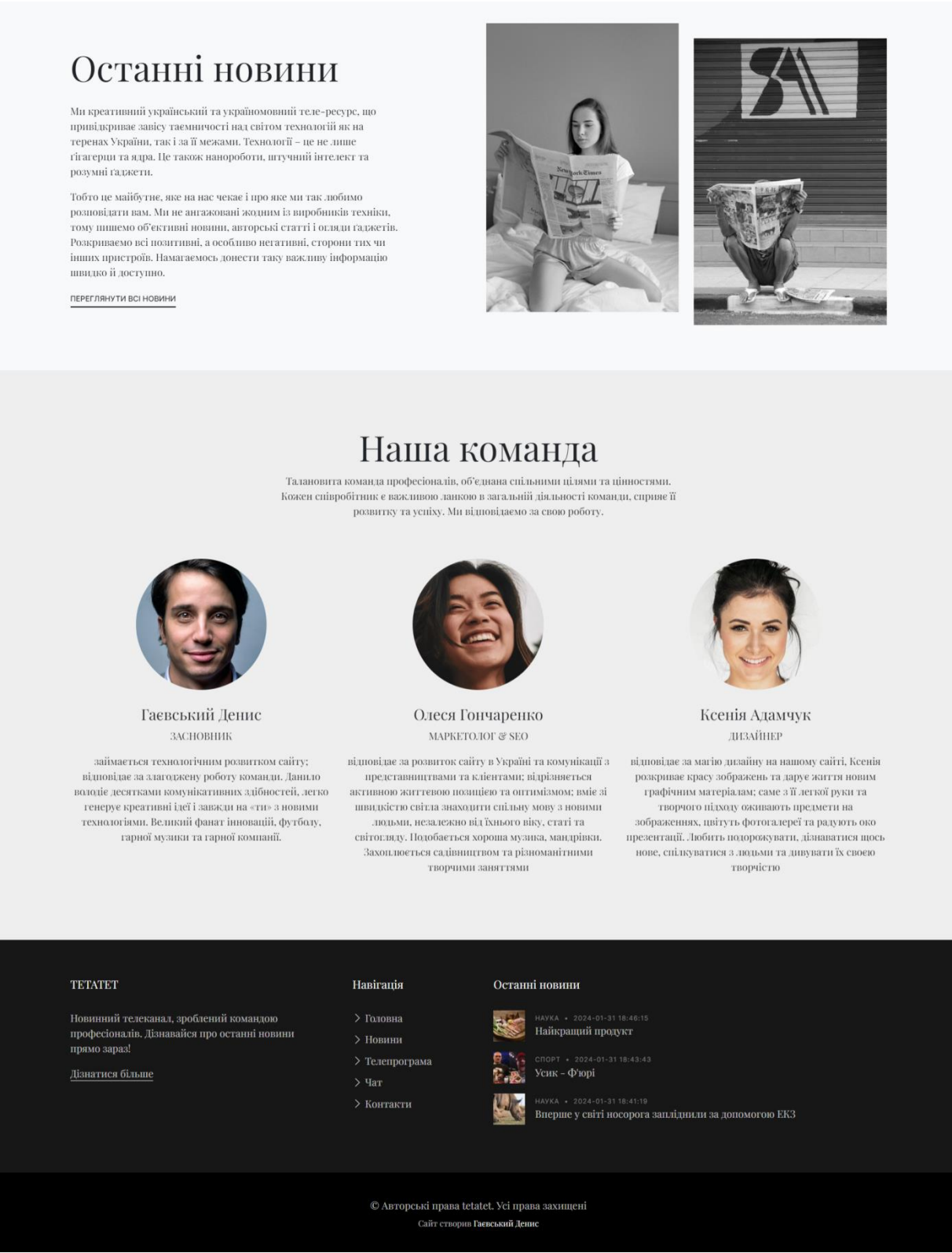


Рис.7 Головна сторінка сайту

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				28
Змн.	Арк.	№ докум.	Підпис	Дата		

Реєструємось на сайті. Форма реєстрації приймає ім'я, прізвище, пошту, місто, номер телефону, пароль та повторний пароль. Майже всі перераховані поля є обов'язковими. Перед відправкою форми на сервер відбувається перевірка того, чи співпадають паролі та чи існує користувач із такою самою поштою, в негативному випадку, користувача повідомляють про це і просять змінити дані.

**TETATET**      Головна   Новини   Телепрограма   Телепроекти   Чат   Контакти   🔍   Увійти   Реєстрація

### Реєстрація

Аккаунт необхідний для того, щоб користуватись чатом, коментувати новини й телепередачі, тощо...

Увійти можна тут

Ім'я\*

Прізвище\*

Звідки ви?

Номер телефону

Е.пошта\*

Пароль\*

Повторіть пароль\*

☐ Я згоден з Правилами користування\*

**Створити аккаунт**

© Авторські права tetatet. Усі права захищені  
Сайт створив Гаєвський Денис

**Рис.8** Форма для реєстрації користувача в систему

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

Заходимо на сайт. Дана форма приймає логін та пароль. При відправці запиту на під'єднання користувача до веб-додатку, відбувається перевірка чи співпадають пароль та логін користувача. Якщо ж веденого користувача не виявлено чи навпаки увійти вдалось, буде висвітлено сповіщення про невдачу чи вдалу авторизацію.

**Рис.9** Сторінка авторизації

**Рис.10** Сповіщення про вдалу авторизацію

**Рис.11** Сповіщення про невдачу авторизацію

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

Переглядаємо сторінку Контакти, тут кожен бажаючий може задати питання чи залишити відгук, присутня форма, яка приймає ім'я, ел.адресу, телефон та саме повідомлення:

**Рис.12** Сторінка “контакти”

Переглядаємо сторінку Чат, тут реалізована можливість залишити повідомлення в чат користувачів, для цього створена окрема таблиця в базі даних **chat**, форма приймає лише повідомлення, можливість залишити його мають лише авторизовані користувачі:

**Рис.13** Сторінка “чат користувачів

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				31
Змн.	Арк.	№ докум.	Підпис	Дата		

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сугодняк І. І.				32
Змн.	Арк.	№ докум.	Підпис	Дата		



Переглядаємо сторінку Новини, на ній висвітлюється список із оголошень, і оскільки ми зайшли через роль адміністратора, в нас є функція відредагувати чи видалити повідомлення:

ГОЛОВНА

НОВИНИ

ТЕЛЕПРОГРАМА

ТЕЛЕПРОЕКТИ


ЧАТ

КОНТАКТИ

Увійти

Реєстрація

КАТЕГОРІЯ: ОСТАННІ НОВИНИ




НАУКА

• 2024-01-31 18:46:15

Найкращий продукт

З ним можна прожити більше 100 років: улюблений продукт довгожителів і всіх українців

Автор: Артем Гаєвський




СПОРТ

• 2024-01-31 18:43:43

Усик - Ф'юрі

Усик - Ф'юрі: боксери назвали фаворита бою за звання абсолютного чемпіона світу

Автор: Артем Гаєвський




НАУКА

• 2024-01-31 18:41:19

Вперше у світі носорога запліднили за допомогою ЕКЗ

Не даю надію на порятунок північного білого носорога від вимирання - на планеті залишилися лише дві такі тварини.

Автор: Артем Гаєвський




ІТ

• 2023-10-14 13:28:47

Мільйони за ніч. Користувачі помітили, що їхні підписники у Facebook різко зникли

Користувачі з усього світу помічають, що кількість підписників у популярних акаунтів різко впала.

Автор: Василь Поліщук




БІЗНЕС

• 2023-10-14 13:23:46

Данілов спростував заяви про те, що російські обстріли пошкодили третину енергетичної інфраструктури України

Секретар РНБО Олександр Данілов спростував заяви глави Міненерго про те, що російські ракетні удари 10 жовтня пошкодили близько 30% енергетичної інфраструктури України.

Автор: Василь Поліщук




ВІЙНА

• 2023-10-14 13:15:01

Зеленський прокоментував рішення ПАРЄ про терористичний режим Росії:

Зеленський заявив, що з цієї терористичною групою, нема про що говорити.

Автор: Наталія Коваль




ПОЛІТИКА

• 2023-10-14 13:10:34

Курс валют на 1 лютого

Курс валют на 1 лютого: скільки коштуватимуть долар, євро і злотий

Автор: Наталія Коваль



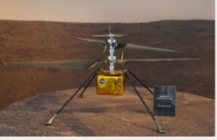
ВІЙНА

• 2023-10-14 13:05:01

У ТЦК Києва розповіли, чи будуть силовіць затримувати чоловіків на вулицях

У ТЦК Києва заявили, що не зможуть силовіць затримувати військовозобов'язаних, бо таких повноважень у співробітників військоматів відповідно до законодавства немає, а це...

Автор: Наталія Коваль



ІТ

• 2023-10-14 12:58:12

Вертоліт NASA на Марсі здійснив останній політ і вийшов із ладу

Коли вертоліт Ingenuity опинився на висоті 1 метр над поверхнею Марса, вчені з NASA втратили з ним зв'язок.

Автор: Артем Гаєвський

ОСТАННІ

ВІЙНА

ІТ

БІЗНЕС

ПОЛІТИКА

ВІЙНА

• 2023-10-14 12:36:03

Маск вирішив не платити за "Старлінк" в Україні і відмовляється постачати нові

Артем Гаєвський

ІТ

• 2023-10-14 12:58:12

Вертоліт NASA на Марсі здійснив останній політ і вийшов із ладу

Артем Гаєвський

ВІЙНА

• 2023-10-14 13:05:01

У ТЦК Києва розповіли, чи будуть силовіць затримувати чоловіків на вулицях

Наталія Коваль

ПОЛІТИКА

• 2023-10-14 13:10:34

Курс валют на 1 лютого

Наталія Коваль

ВІЙНА

• 2023-10-14 13:15:01

Зеленський прокоментував рішення ПАРЄ про терористичний режим Росії:

Наталія Коваль

БІЗНЕС

• 2023-10-14 13:23:46

Данілов спростував заяви про те, що російські обстріли пошкодили третину енергетичної інфраструктури України

Василь Поліщук

ІТ

• 2023-10-14 13:28:47

Мільйони за ніч. Користувачі помітили, що їхні підписники у Facebook різко зникли

Василь Поліщук

НАУКА

• 2024-01-31 18:41:19

Вперше у світі носорога запліднили за допомогою ЕКЗ

Артем Гаєвський

СПОРТ

• 2024-01-31 18:43:43

Усик - Ф'юрі

Артем Гаєвський


НАУКА

• 2024-01-31 18:46:15

Найкращий продукт

Артем Гаєвський

ВІДЕО



		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				33
Змн.	Арк.	№ докум.	Підпис	Дата		

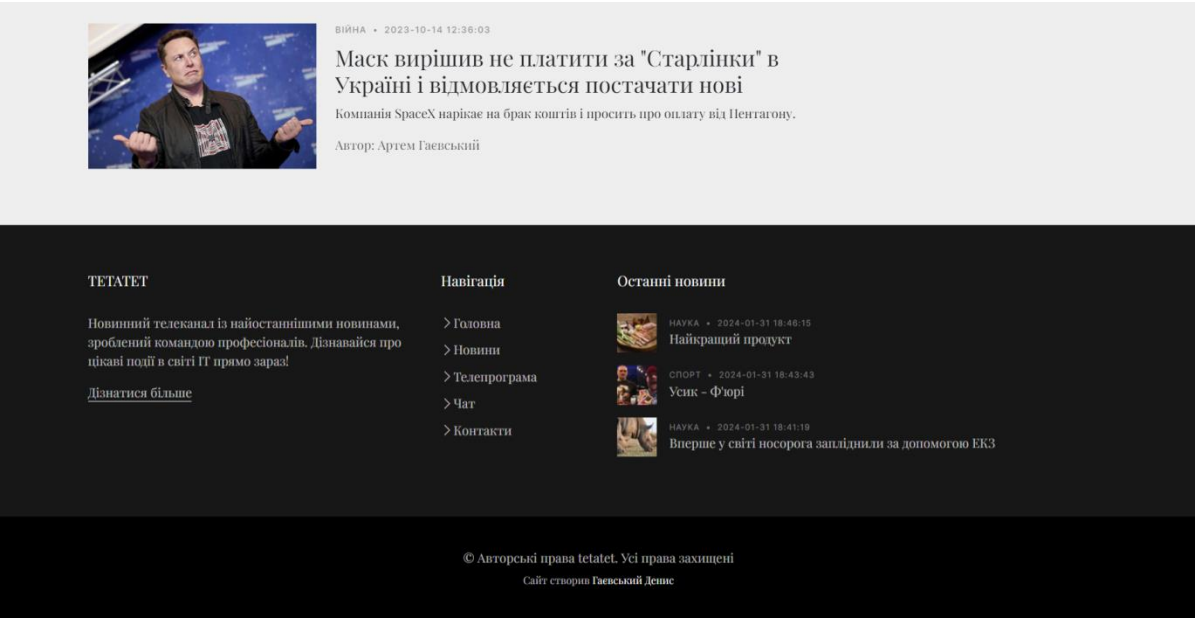


Рис.14 Сторінка “новини”

Переглянемо одну із новин детальніше, із бази даних підтягується вся інформація про цю новину, можна залишити коментар до неї:

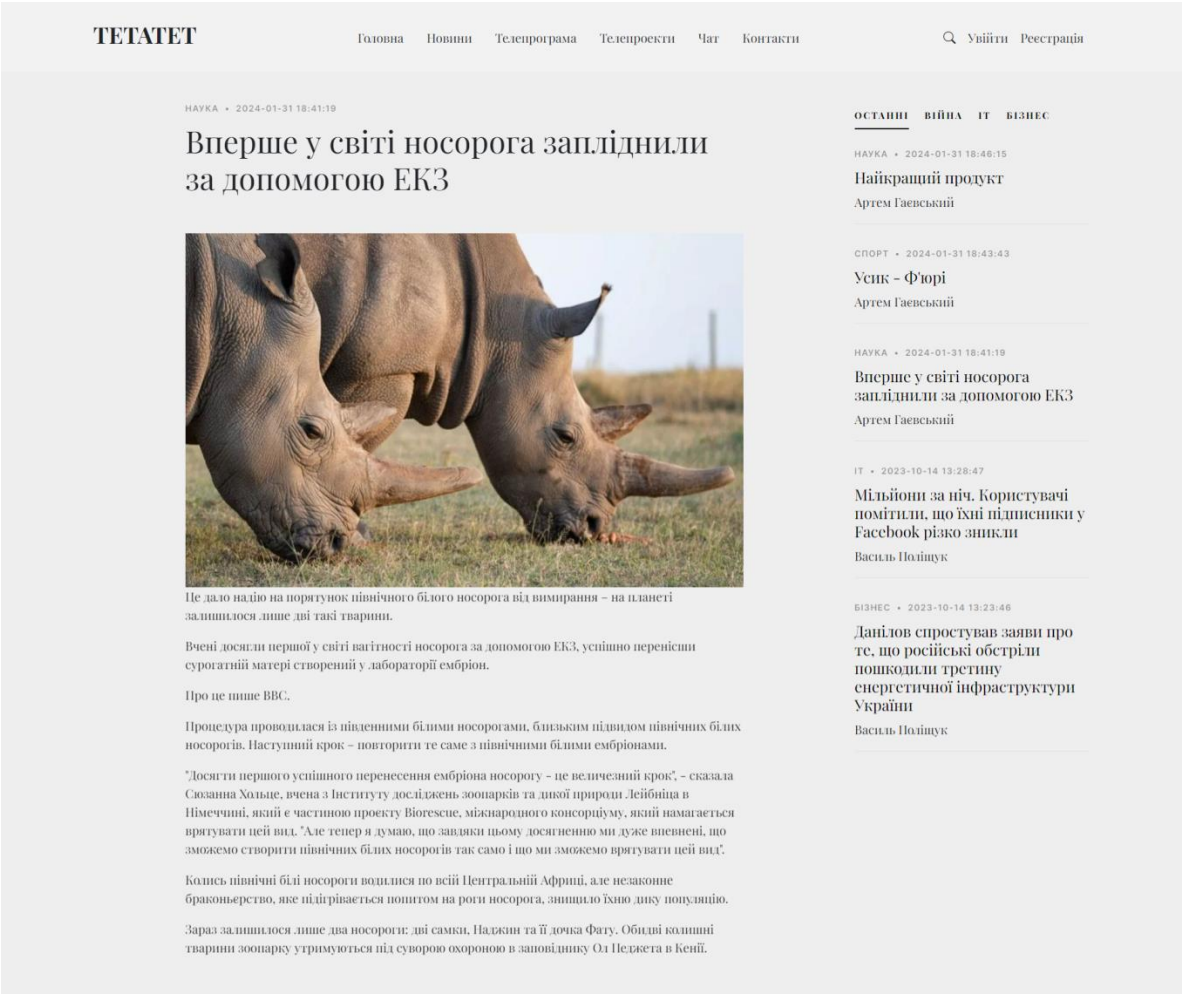
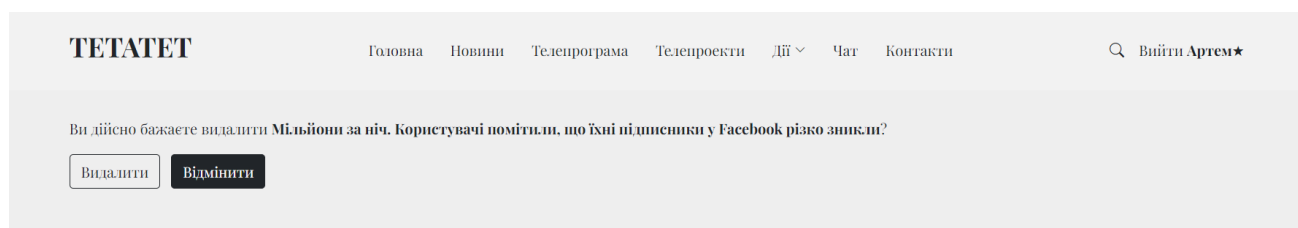


Рис.15 Сторінка новини

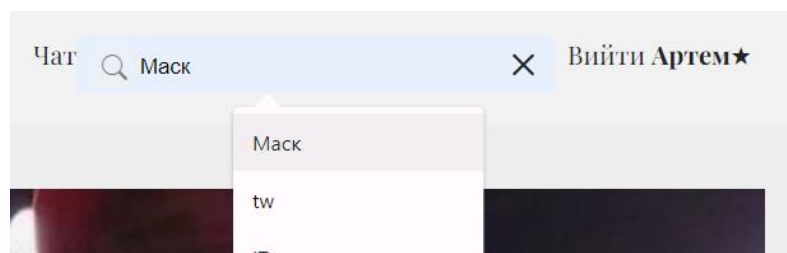
		Гаєвський Д.О.				Арк.
		Сьогодні І. І.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

Ті новини, які були створені активним аккаунтом або коли користувач це адміністратор, можна редагувати або видаляти дописи. Ось приклад редагування або створення допису. Форма приймає заголовок, назву, текст, і фотографію:

**Рис.16** Сторінка “новини”. Приклад редагування новини



**Рис.17** Сторінка “новини”. Приклад видалення допису



**Рис.18** Пошук новини

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				35
Змн.	Арк.	№ докум.	Підпис	Дата		

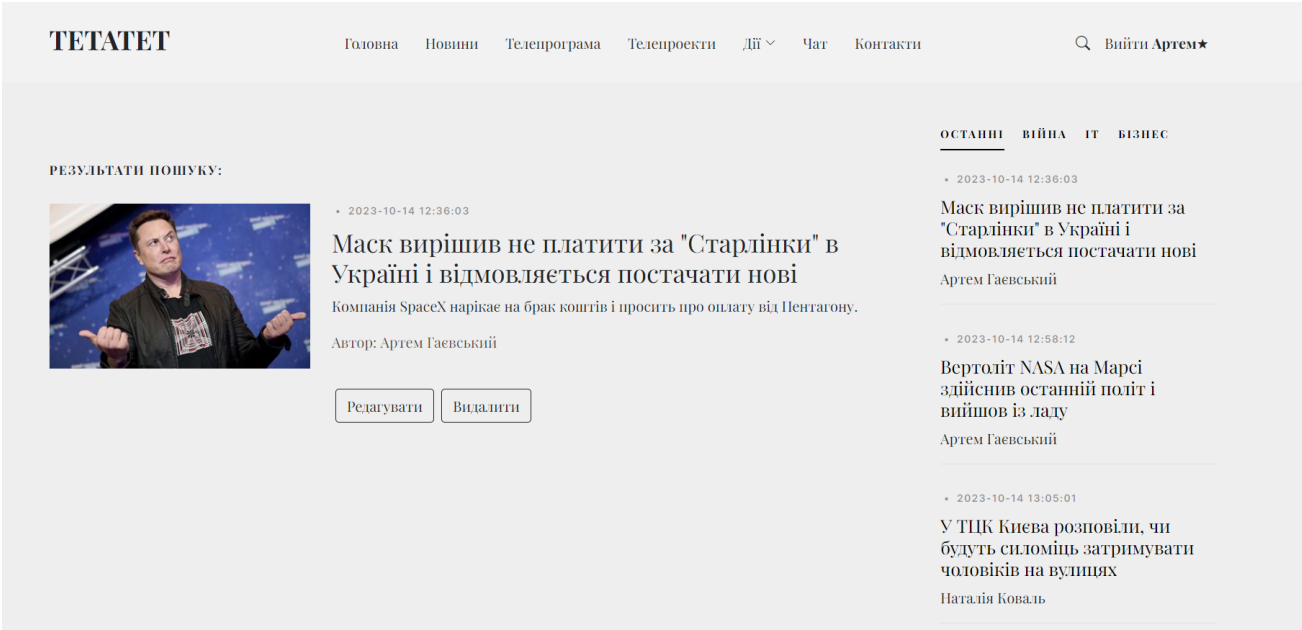


Рис.19 Пошук новини

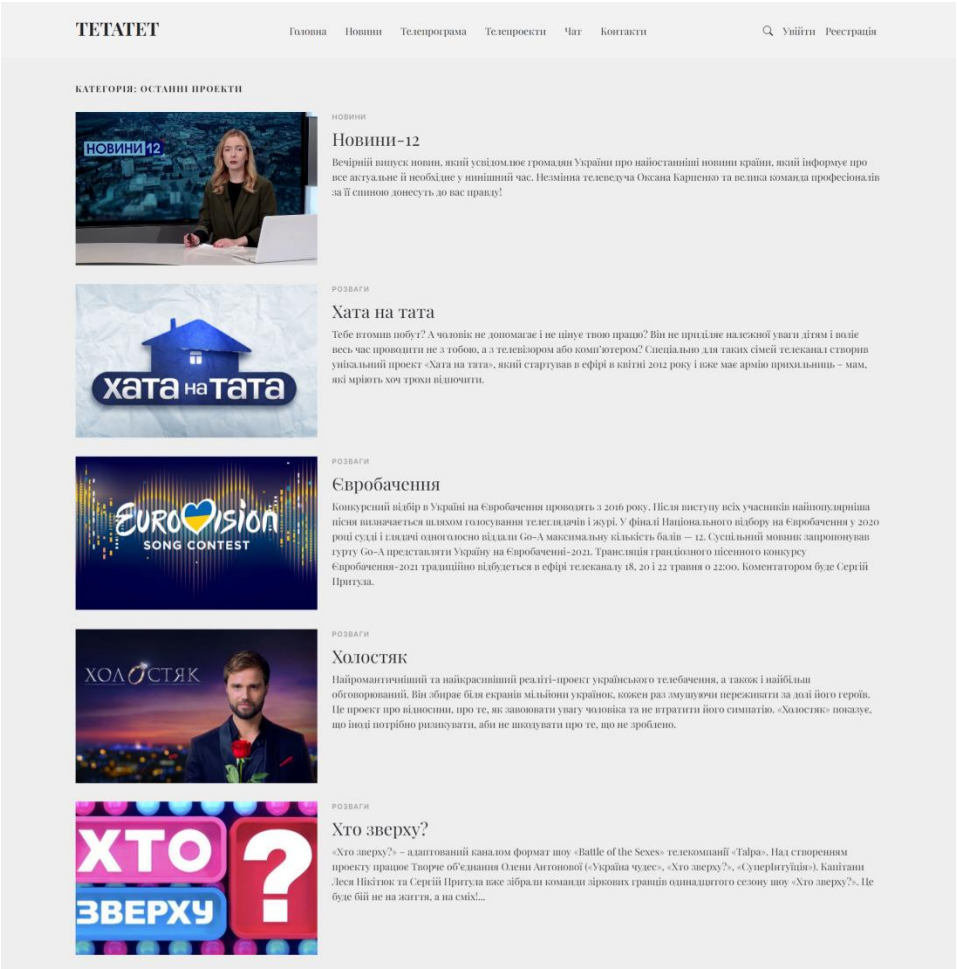


Рис.20 Проекти

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				36
Змн.	Арк.	№ докум.	Підпис	Дата		



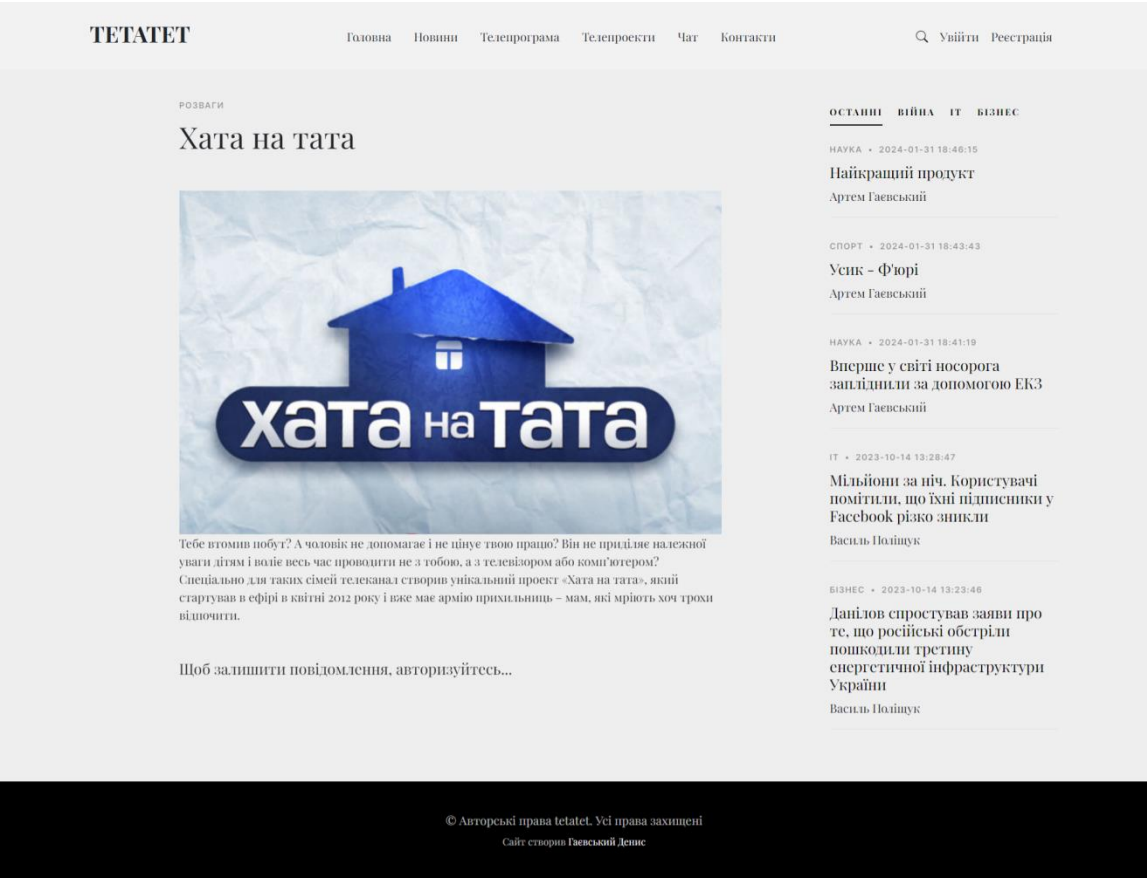


Рис.21 Сторінка проекту

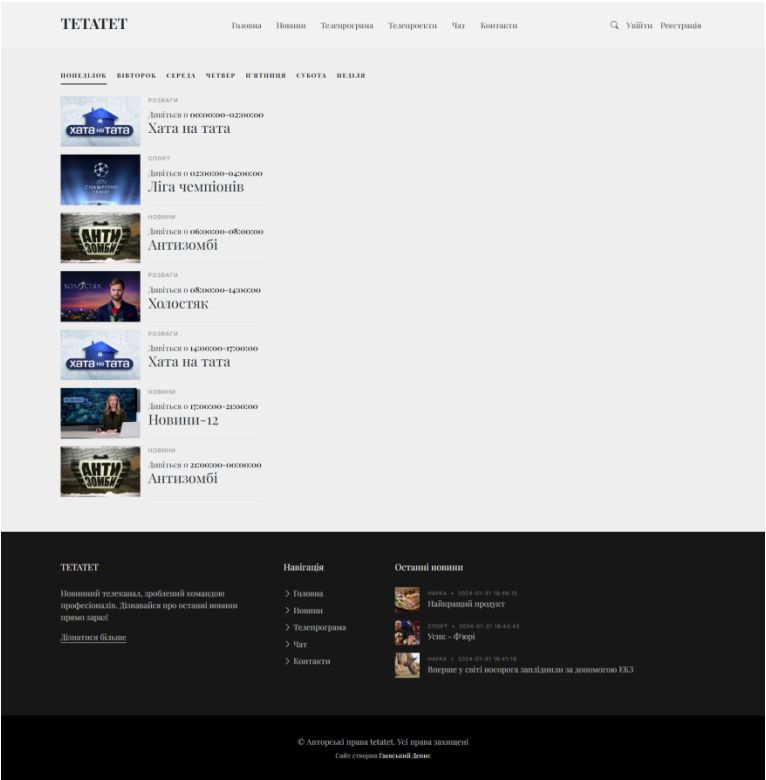
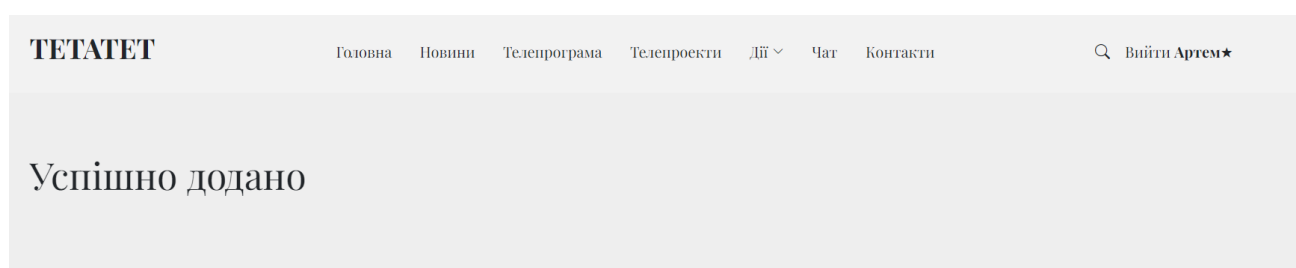


Рис.22 Телепрограма

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк. 37
		Сьогодні І. І.				
Змн.	Арк.	№ докум.	Підпис	Дата		

**Рис.24** Додавання телепрограми



**Рис.25** Успішне додавання телепрограми

### 3.2. Реалізації операцій обробки даних в БД

Оскільки будь-яка програма розпочинається з аутентифікації та контролю доступу, то потрібно розглянути, як саме користувач реєструється та входить на веб-сайт під своїм логіном.

Функція додавання:

```
public function AddUser($userRow)
{
    $validateResult = $this->Validate($userRow);
    if (is_array($validateResult))
        return $validateResult;
    $fields = ['login', 'password', 'firstname', 'lastname', 'phone', 'city'];
    $userRowFiltered = Utils::ArrayFilter($userRow, $fields);
    $userRowFiltered['password'] = md5($userRowFiltered['password']);
    \core\Core::getInstance()->getDB()->insert('users', $userRowFiltered);
    return true;
}
```

Функція приймає введені дані із форми реєстрації, валідує їх за допомогою методу Validate, який в свою чергу перевіряє чи існує вже даний користувач та чи були введені всі поля форми, якщо валідація пройшла успішно, ми сортируємо масив та передаємо його до таблиці users в БД.

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

## Функція аутенфікації:

```
public function AuthUser($login, $password)
{
    $password = md5($password);
    $users = \core\Core::getInstance()->getDB()->select('users', "",
    [
        'login' => $login,
        'password' => $password,
    ]));
    if (count($users) == 1) /** якщо 1 користувач**/
    {
        $user = $users[0];
        return $user;
    } else
        return false;
}
```

Функція приймає введений пароль та логін, шифрує пароль за тим самим кодуванням, який і його додавав при реєстрації, та шукає в БД співпадіння, якщо користувач був знайдений, функція повертає нам його об'єкт, якщо ж ні то **false**.

## Давайте розглянемо функцію додавання новини

```
public function AddNews($row)
{
    $userModel = new \models\Users();
    $user = $userModel->GetCurrentUser();
    if ($user == null) {
        $result = [
            'error' => true,
            'messages' => ['Користувач не аутентифікований']
        ];
        return $result;
    }
    $validateResult = $this->Validate($row); /** валідація **/
    if (is_array($validateResult)) {
        $result = [
            'error' => true,
            'messages' => $validateResult
        ];
        return $result;
    }
    $fields = ['title', 'type', 'description', 'text']; /** дані для таблиці **/
    $rowFiltered = Utils::ArrayFilter($row, $fields);
    $rowFiltered['datetime'] = date('Y-m-d H:i:s');
    $rowFiltered['user_id'] = $user['id'];
    $rowFiltered['photo'] = '...photo...';
    $id = \core\Core::getInstance()->getDB()->insert('news', $rowFiltered); /**
    відправляємо в бд**/
}
```

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І.І.				39
Змн.	Арк.	№ докум.	Підпис	Дата		

```

return [
    'error' => false,
    'id' => $id
];
}

```

Функція спочатку звіряє чи аутентифікований користувач, якщо ж ні, то виводить помилку, якщо так, то вона працює далі, валідує введені дані з форми(принцип валідації був розписаний у дописі до минулої функції), але поки без самої фотографії(вона буде додана за допомогою наступної функції) та заносить ці дані до БД.

Функція додавання фото в новині:

```

public function ChangePhoto($id, $file)
{
    $folder = 'files/news/';
    $file_path = pathinfo($folder.$file);

    /** створення назв**/
    $file_big = $file_path['filename'].'_b.jpg';
    $file_middle = $file_path['filename'].'_m.jpg';
    $file_small = $file_path['filename'].'_s.jpg';
    $news = $this->GetNewsById($id);
    if (is_file($folder.$news['photo'].'_b.jpg') && is_file($folder.$file))
        unlink($folder.$news['photo'].'_b.jpg');
    if (is_file($folder.$news['photo'].'_m.jpg') && is_file($folder.$file))
        unlink($folder.$news['photo'].'_m.jpg');
    if (is_file($folder.$news['photo'].'_s.jpg') && is_file($folder.$file))
        unlink($folder.$news['photo'].'_s.jpg');
    $news['photo'] = $file_path['filename'];

    /** зчитування, переробка й зберігання фото **/
    $im_b = new Imagick();
    $im_b->readImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.$file);
    $im_b->cropThumbnailImage(900, 571, true);
    $im_b->writeImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.'/'.$file_big.'.jpg');

    $im_m = new Imagick();
    $im_m->readImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.$file);
    $im_m->cropThumbnailImage(340, 215, true);
    $im_m->writeImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.'/'.$file_middle.'.jpg');

    $im_s = new Imagick();
    $im_s->readImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.$file);
    $im_s->cropThumbnailImage(50, 50, true);
}

```

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І.І.				40
Змн.	Арк.	№ докум.	Підпис	Дата		



```
//or $im_s->ThumbnailImage(180, 180, true, false); із збереженням пропорцій
$im_s-
>writeImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.'/'.$file_small.'.jpg');
unlink($folder.$file);
$this->UpdateNews($news, $id);
}
```

Функція отримує сам файл фотографії та айді новини, створює каталог із фото, та формлює назви майбутніх фотографій. Перевіряє чи існують файли із такою самою ж назвою, якщо так, то просто їх перезаписує, далі він тричі зчитує, обрізає фотографії до потрібних нам розмірів(велике, середнє, мале) та зберігає його в каталозі із фото, потім оновлює новини.

Давайте також розглянемо функцію, які відповідає за валідацію даних

```
public function Validate($formrow)
{
    $errors = [];
    if (empty($formrow['title']))
        $errors [] = 'Поле "Назва" не може бути порожнім!';
    if (empty($formrow['type']))
        $errors [] = 'Поле "Тип" не може бути порожнім!';
    if (empty($formrow['description']))
        $errors [] = 'Поле "Короткий опис" не може бути порожнім!';
    if (empty($formrow['text']))
        $errors [] = 'Поле "Повний текст" не може бути порожнім!';
    if (empty($formrow['price']))
        if(count($errors)>0)
            return $errors;
    else
        return true;
}
```

## Висновок до третього розділу

В цьому розділі була описана документація до розробленого веб-додатку, яке описує роботу з додатком для звичайного користувача. Сайт має загалом інтуїтивно зрозумілий інтерфейс.

Відбулось написання повного коду для взаємодії користувача з БД. Було спроектовано інтерфейс веб-сайту новинного блогу. Були розроблені форми для створення оголошень та їх редагувань, повідомлень, коментарів. Були створені сторінки з можливістю перегляду, виконання запитів на фільтрації даних, виконання запитів на підгрузку даних.

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 4. АДМІНІСТРУВАННЯ БАЗ ДАНИХ

### 4.1. Розробка заходів захисту інформації в БД

Отже, розглянемо категорії користувачів веб-сайту телеканалу “ТЕТАТЕТ”: адміністратор, автор, авторизовані користувачі, неавторизовані користувачі. Найбільші права доступу до даних має адміністратор, що необхідно для організації роботи сайту. Він можуть переглядати чат адміністраторів та залишати там власні повідомлення, також адміністратори мають можливість редагування та видалення будь-якого новини. Більш детально про роль надано ролі "Адміністратор" наведено в табл. 6

Таблиця 6

Матриця доступу для ролі "Адміністратор"

Роль:	Адміністратор
Опублікувати новини	+
Редагувати всі новини	+
Видаляти всі новини	+
Опублікувати проекти	+
Редагувати всі проекти	+
Видаляти всі проекти	+
Опублікувати телепрограми	+
Редагувати всі телепрограми	+
Видаляти всі телепрограми	+

Звичайні користувачі поділяються на авторизованих та на не авторизованих, так як у авторизованих є доступ до більшого функціоналу.

Більш детально про роль " Авторизований користувач " наведено в табл. 7

Таблиця 7

Матриця доступу для ролі "Авторизований користувач"

	Авторизований користувач
Опублікувати новини	-
Редагувати всі новини	-
Видаляти всі новини	-
Опублікувати коментар під новинами	+
Опублікувати коментар під проектами	+
Опублікувати повідомлення у чаті користувачів	+

Також розглянемо функціонал для не авторизованого користувача:

Таблиця 8

Матриця доступу для ролі "Не авторизований користувач"

	Не авторизований користувач
Опублікувати новини	-
Видаляти всі новини	-
Опублікувати коментар під новиним	-
Опублікувати повідомлення у чаті користувачів	-
Доступ до чату адміністраторів	-

## 4.2. Налаштування параметрів роботи Firebase.

Для того, щоб підключити Firebase в проект потрібно імпортувати та встановити бібліотеку firebase за допомогою команди npm і firebase.

Далі потрібно підключитись до проекту, створеного на офіційному сайті даного продукту <https://firebase.google.com/>.

Далі потрібно імпортувати(або скопіювати) дані про створений firestore проект. У моєму випадку об'єкт з конфігураціями має наступний вигляд:

```
const firebaseConfig = {
  apiKey:
    "AIzaSyBBm8KHmG5w4iCNHRT6APuHxMLuh3OI3t4k",
  authDomain: "cursova-5b006.firebaseio.com",
  projectId:
    "cursova-5b006",
  storageBucket: "cursova-5b006.appspot.com",
  messagingSenderId: "378637937949",
  appId:
    "1:378637937949:web:7c38e86d45d0059e64c5f4",
  measurementId:
    "G-4PPJZCJVFK",
  databaseURL: "https://cursova-5b006.firebaseio.com"
};
```

Після отримання даних конфігурації, потрібно зробити підключення до проекту та підключити можливість роботи з базою та з аутентифікацією користувачів.

Лістинг коду:

```
import { initializeApp } from "firebase/app";
import { getAuth } from "firebase/auth";
import { getFirestore } from
  "firebase/firestore";

const firebaseConfig = {...};

export const app = initializeApp(firebaseConfig); export const db = getFirestore(app) export const
auth = getAuth(app);
```

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				44
Змн.	Арк.	№ докум.	Підпис	Дата		

Після того, як було коректно підключено та налаштовано роботу з базою, потрібно створити колекції(або таблиці по релятивному), у яких будуть зберігатись дані про сайт. Для цього потрібно викристати метод collection, що є у встановленій бібліотеці, передати туди об'єкт з інформацією про базу та назву колекції.

Лістинг коду:

```
import { db } from './config' import { collection } from "firebase/firestore"
export const users = collection(db, 'users') export const pictures =
collection(db, 'pictures') export const hols = collection(db, 'hols') export
const roles = collection(db, 'roles') export const holsPlans = collection(db,
'holsPlans') export const guides = collection(db, 'guides') export const
ticketsTemplates = collection(db, 'ticketsTemplates') export const
userTickets = collection(db, 'userTickets') export const
timesTicketTemplates = collection(db, 'timesTicketTemplates') export
const systemMessages = collection(db, 'systemMessage')
```

У даному коді було створено усі потрібні для бази колекції

Далі варто зробити спрощену логіку додавання, оновлення, видалення та считування даних, для того, щоб уникнути великої кількості шаблонного коду. Для цих дій потрібно використовувати функції `oc`, `setDoc`, `addDoc`, `getDoc`, `deleteDoc`, зі встановленої бібліотеки. Ці функції приймають в себе колекцію та певні дані. Лістинг спрощених команд наведено нижче:

```
export const create = async (collection: CollectionReference, data: any) => {
  return await addDoc(collection, data)
} export const update = async (collection: CollectionReference, id: string, data: any)
=>
{
  const ref = doc(collection, id)

  return await setDoc(ref, data, { merge: true })
} export const get = async (collection: CollectionReference, id: string)
=> {
  const ref = doc(collection, id)
  const snap = await getDoc(ref)
  return snap.exists() ? snap.data() : undefined
} export const drop = async (collection: CollectionReference, id: string)
=> {
  const ref = doc(collection, id)
  return await deleteDoc(ref)
```

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				45
Змн.	Арк.	№ докум.	Підпис	Дата		

}

```
export const createDocWithId = async (collection: CollectionReference, id: string,
data: any) => {    await setDoc(doc(collection, id), data).then(() => id) }
```

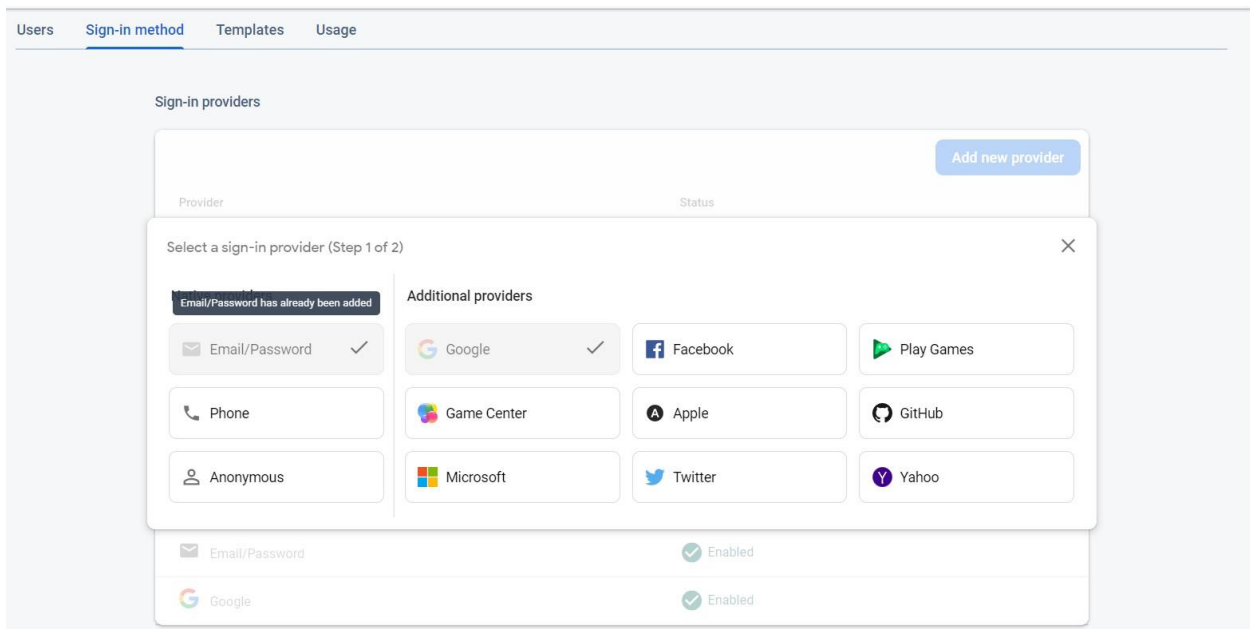
Далі варто розглянути роботу з медіа. Якщо нам потрібно зберігати якісь медіа-файли, то для цього потрібно підключити ще один тип збереження даних у Firebase – Storage.

Для того, щоб у цьому сховищі даних, всі файли зберігались коректно і в коректних папках(щоб шлях, при збереженні в Firestore був коректним і повертав медіа), потрібно прописати певні правила. Для цього потрібно перейти в Storage, далі натиснути на вкладку rules і написати свої правила для збереження.

Для того, щоб користувачі могли коректно реєструватись, та логінитись, потрібно підключити додаткову функцію з назвою –

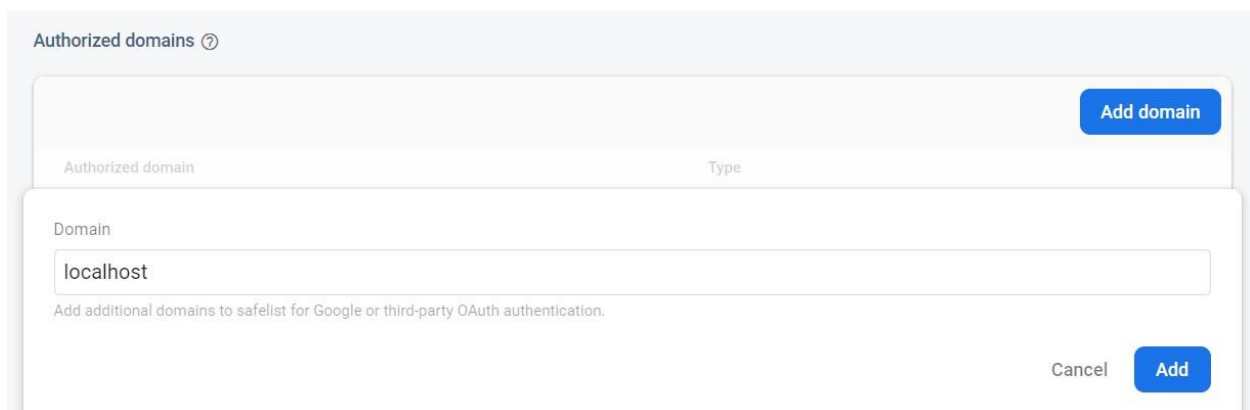
Authentication. Для того, щоб дозволити юзерам конектитись до програми за допомогою логіну та пароллю, потрібно відкрити вкладку Authentication далі натиснути на Sign-in method і в провайдерах додати новий, що має назву Email\Password. Після додання цього провайдера, користувачі зможуть реєструватись та логінитись до додатку “сайт телеканалу”. Приклад додання провайдера наведено на рисунку 18.

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				46
Змн.	Арк.	№ докум.	Підпис	Дата		



**Рис.26.** Приклад налаштування можливості входу користувача до розробленого додатку

Також для того, щоб мати змогу коректно підключатись доданої функції, варто внести домену ім'я додатку до Authorized domains, це потрібно для того, щоб Firebase не блокував запити до даної функції. Приклад додання нового домену наведено на Рис.21.



**Рис.27.** Додання домену до “білого списку” Firebase Authentication

Розглянемо код, що виконує реєстрацію користувача:

```
import { createUserWithEmailAndPassword, getAuth, updateProfile, signInWith,
  signInWithEmailAndPassword } from "firebase/auth";
```

```
export const signUp = async ({ email, password, firstName, lastName,
  avatar, repeatedPassword }: IUserRegister) => {    const { user } = await
```

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				47
Змн.	Арк.	№ докум.	Підпис	Дата		

```

createUserWithEmailAndPassword(getAuth(), email, password);
let photoURL = ""; if (avatar) photoURL =
await uploadFile("user", avatar, user.uid); const displayName =
`${firstName} ${lastName}`; await updateProfile(user, {
displayName, photoURL}); return user;
}

```

Перший рядок імпортує всі необхідні функції для коректної взаємодії з аутентифікацією користувача за допомогою паролю та пошти. Далі відбувається створення функції, яка буде приймати об'єкт, що буде складатись з пошти, імені, паролів та аватару. Далі Створюємо користувача за допомогою даних (асинхронна функція createUserWithEmailAndPassword). Далі, при наявності фото, при створенні акаунту, відбувається його збереження, після чого згенероване посилання передається у відповідну змінну, далі відбувається генерація повного імені користувача з імені та прізвища, після цього відбувається оновлення створеного акаунту

Після реєстрації користувач може залогінитись, дія подібна до реєстрації, за кількома відмінностями: користувач має внести лише пошту та пароль, і не потрібні ніякі оновлення даних, різні генерації даних, потрібно просто використати метод signInWithEmailAndPassword, він дозволить використовувати далі метод getAuth, який буде повертати інформацію про акаунт, під яким зайшов користувач.

Також, в js можна слідкувати за змінами поточного користувача, для цього є спеціальний метод onAuthStateChanged. Даний метод приймає об'єкт, що посилається на Authentication та колбек, що приймає поточного користувача, і в колбеці можна виконувати певні дії з інформацією про нового користувача, під яким відбувався вхід, чи вихід з акаунту(якщо вихід, то в поточному користувачі нічого не буде).

Ось як відбувалось налаштування та написання функцій для коректної авторизації користувача в додатку, збереження медіа-файлів та налаштування доступу до цих функцій.

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				48
Змн.	Арк.	№ докум.	Підпис	Дата		



## Висновки до четвертого розділу

У ході виконання даного розділу проведено налаштування прав доступу користувача до бази, як зі сторони сайту Firebase так і зі сторонни клієнта веб-додатку.

Також у ході виконання цього розділу було показано, як саме відбулось підключення додатку до БД, як відбувалось налаштування можливості реєстрації користувача, авторизації та виходу з акаунту, як відбулось налаштування збереження медіа, та як саме медіа-файли почали зберігатись та як генерується шлях(посилання), цих медіа-файлів на даний момент для певних колекцій даних.

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				49
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

У результаті виконання даного проекту було спроектовано та розроблено базу даних сайту телеканалу.

Під час виконання курсового проекту було проаналізовано головні, можливості та особливості різних БД, що допомогло визначити основні плюси та мінуси їх та вибрати ту, за допомогою якої можна якісно та швидко виконати дану роботу.

Після того, як було вибрано та пояснено причину та логіку вибору БД було виконано проектування схеми бази. У результаті виконання цього етапу, було розроблено коректну загальну схему бази, що дозволило отримати розуміння того, що саме має відбуватись в коді та як це потрібно реалізувати.

Після проектування загальної структури бази, було розпочато проектування інтерфейсу веб-додатку. Під час цього етапу було реалізовано підключення до БД, створені різні методи для коректної взаємодії з базою. Наприклад, було написано методи для вставок даних, для оновлен даних, для їх видалення й пошуку.

При виконання загальної структури коду, було виконано налаштування деяких додаткових сервісів для коректної роботи додатку, а саме для збереження медіа-файлів та можливості валідного входу та реєстрації користувача в додатку.

Було проведено тестування у ручному режимі, після виявлення помилок, їх було виправлено та пере-протестовано додаток, що дає можливість визнати, що додаток є стабільним. Таким чином була розроблена програма з використанням мови PHP, SQL, HTML, CSS, JavaScript використаний Bootstrap та phpMyAdmin.

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				50
Змн.	Арк.	№ докум.	Підпис	Дата		

## Список використаних джерел

1. А.Зандстра, Ф.Матт. PHP Objects, Patterns, and Practice, 2013. Режим доступу: <https://www.apress.com/la/book/9781430260325>
2. Виразний JavaScript - 2017. [Електронний ресурс]. Режим доступу: [https://legacy.gitbook.com/book/karmazzin/eloquentjavascript\\_ru/details](https://legacy.gitbook.com/book/karmazzin/eloquentjavascript_ru/details)
3. Всі операції с БД в графічному додатку [Електронний ресурс]  
Режим доступу до ресурсу: <https://metanit.com/sharp/adonet/3.5.php>
4. Технічна документація MySQL [Електронний ресурс] - 2006.  
Режим доступу: <https://dev.mysql.com/doc/>
5. Завантажувати файли за допомогою хмарного сховища в Інтернеті.  
[Електронний ресурс] – 2022. Режим доступу:  
<https://firebase.google.com/docs/storage/web/upload-files>
6. Основи наслідування - 2015. Режим доступу: [https://professorweb.ua/my/level7/7\\_1.php](https://professorweb.ua/my/level7/7_1.php)
7. Повна документація phpMyAdmin [електронний ресурс] - 2022. Режим доступу: <https://www.phpmyadmin.net/docs/>
8. Створення SQL запитів - 2016. Режим доступу: [https://www.politerm.com/zuludoc/spatial\\_query\\_sql.htm#\\_sql\\_overview](https://www.politerm.com/zuludoc/spatial_query_sql.htm#_sql_overview)
9. Тим Кедлек. “Адаптивний дизайн. Робимо сайти для будь-яких девайсів”  
- 2013. Режим доступу: <https://www.ozon.ua/context/detail/id/21098804/>
10. SQL - 2016. Режим доступу: <https://uk.wikibooks.org/wiki/SQL>

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				51
Змн.	Арк.	№ докум.	Підпис	Дата		

# ДОДАТКИ

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сугодняк І. І.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

## Вихідний код проекту

**DB.php**

```

<?php

namespace core;

class DB
{
    protected $pdo;
    public function __construct($server, $login, $password, $database)
    {
        $this->pdo = new
        \PDO("mysql:host={$server};dbname={$database};charset=UTF8", $login, $password);
    }

    //CRUD
    /**
     * Create
     * Read
     * Update
     * Delete
     */
    public function select($table, $fields = '*', $where = null, $orderBy = null,
        $limit = null, $start = null)
    {
        $fieldsStr = '*';
        if (is_string($fields))
            $fieldsStr = $fields;

        if (is_array($fields))
            $fieldsStr = implode(", ", $fields);
        $sql = "SELECT {$fieldsStr} FROM {$table}";

        if (is_array($where) && count($where) > 0) {
            $whereParts = [];
            foreach($where as $key => $value)
                $whereParts [] = "{$key} = ?";
            $whereStr = implode(" AND ", $whereParts);
            $sql .= " WHERE ".$whereStr;
        }

        if(is_string($where))
            $sql .= " WHERE ".$where;
    }
}

```

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І.І.				53
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if (is_array($orderBy))
{
    $orderByParts = [];
    foreach($orderBy as $key => $value)
    {
        $orderByParts [] = "{$key} {$value}";
    }
    $sql .= ' ORDER BY '.implode( ' ', $orderByParts);
}

if (!empty($limit))
{
    if (!empty($offset))
        $sql .= " LIMIT {$offset}, {$limit}";
    else
        $sql .= " LIMIT {$limit}";
}

$stmt = $this->pdo->prepare($sql);
if (is_array($where) && count($where) > 0)
    $stmt->execute(array_values($where));
else
    $stmt->execute();
return $stmt->fetchAll();
}

public function insert($table, $row)
{
    $fieldsStr = implode(',', array_keys($row));
    $valuesParts = [];
    foreach ($row as $key => $value)
    {
        $valuesParts [] = "?";
    }
    $valuesStr = implode(' ', $valuesParts);
    $sql = "INSERT INTO {$table} ($fieldsStr) VALUES ($valuesStr)";
    $stmt = $this->pdo->prepare($sql);
    $stmt->execute(array_values($row));
    return $this->pdo->lastInsertId();
}

public function delete($table, $where = null)
{
    $sql = "DELETE FROM {$table}";
    if (is_array($where) && count($where) > 0){
        $whereParts = [];
        foreach($where as $key => $value)
            $whereParts [] = "{$key} = ?";
    }

```

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І.І.				54
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $whereStr = implode(' AND ', $whereParts);
        $sql.= ' WHERE '.$whereStr;
    }

    if(is_string($where))
        $sql .= ' WHERE '.$where;
    $sth = $this->pdo->prepare($sql);
    if(is_array($where) && count($where) > 0)
        $sth->execute(array_values($where));
    else
        $sth->execute();
}

public function update($table, $newRow, $where)
{
    $sql = "UPDATE {$table} SET ";
    $setParts = [];
    $paramsArr = [];
    foreach($newRow as $key => $value)
    {
        $setParts [] = "{$key} = ?";
        $paramsArr [] = $value;
    }
    $sql .= implode(' ', $setParts);
    if (is_array($where) && count($where) > 0){
        $whereParts = [];
        foreach($where as $key => $value)
            $whereParts [] = "{$key} = ?";
        $paramsArr [] = $value;
        $whereStr = implode(' AND ', $whereParts);
        $sql.= ' WHERE '.$whereStr;
    }

    if(is_string($where))
        $sql .= ' WHERE '.$where;
    $sth = $this->pdo->prepare($sql);
    $sth->execute($paramsArr);
}
}

```

## Core.php

```
<?php
```

```
namespace core;
```

```
/**
```

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				55
Змн.	Арк.	№ докум.	Підпис	Дата		

```

* Головний клас ядра системи
* (синглтон)
*/
class Core
{
    private static $instance;
    private static $mainTemplate; /** зразок сторінки**/
    private static $db; /** бд**/
    private function __construct()
    {
        global $Config; /** приймаємо конфіг бд**/
        spl_autoload_register('\core\Core::__autoload'); /** підключаємось**/
        self::$db = new \core\DB($Config['Database']['Server'],
            $Config['Database']['Username'],
            $Config['Database']['Password'],
            $Config['Database']['Database']);
    }

    /**
     * Повертає екземляр ядра системи
     * @return Core
     */
    public static function getInstance()
    {
        if (empty(self::$instance)){ /** якщо немає**/
            self::$instance = new Core(); /** створити**/
            return self::getInstance(); /** повернути**/
        }
        else
            return self::$instance; /** якщо є, то повернути**/
    }

    /**
     * Отримати об'єкт-з'єднання з базою даних
     */
    public function getDB()
    {
        return self::$db;
    }

    /**
     * Ініціалізація системи
     */
    public function init()
    {

```

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				56
Змн.	Арк.	№ докум.	Підпис	Дата		



```

        session_start();
        self::$mainTemplate = new Template();
    }

    /**
     * Виконує основний процес роботи CMS-системи
     */
    public function run()
    {
        $path = $_GET['path'];
        $pathParts = explode('/', $path);

        $className = ucfirst($pathParts[0]); /** створюємо назви для класів**/
        if (empty($className))
            $fullClassName = 'controllers\\Site';
        else
            $fullClassName = 'controllers\\'.$className;

        $methodName = ucfirst($pathParts[1]); /** створюємо назви для методів**/
        if (empty($methodName))
            $fullMethodName = 'actionIndex';
        else
            $fullMethodName = 'action'.$methodName;

        if (class_exists($fullClassName)){
            $controller = new $fullClassName(); /** створюємо контроллер**/
            if (method_exists($controller, $fullMethodName)){
                $method = new \ReflectionMethod($fullClassName, $fullMethodName);
                $paramsArray = [];
                foreach ($method->getParameters() as $parameter)
                {
                    array_push($paramsArray, isset($_GET[$parameter->name]) ?
$_GET[$parameter->name] : null);
                }
                $result = $method->invokeArgs($controller, $paramsArray);
                if (is_array($result))
                {
                    self::$mainTemplate->setParams($result); /** задаємо
параметри**/
                }
            }
            else /** інакше помилка**/
                throw new \Exception('404 Not Found');}
        else
            throw new \Exception('404 Not Found');
        //echo "Class: {$fullClassName}, method: {$fullMethodName}";
    }

```

		Гаєвський Д.О.			ДУ «Житомирська політехніка».24.121.05.000 - ПЗ	Арк.
		Сьогодні І. І.				57
Змн.	Арк.	№ докум.	Підпис	Дата		

```

/**
 * Завершення роботи системи та виведення даних
 */
public function done(){
    self::$mainTemplate->display('views/layout/index.php'); /** виводимо
сторінку за назвою**/
}

/**
 * Автозавантажувач класів
 * @param $fullClassName string Назва класу
 */
public static function __autoload($fullClassName)
{
    $fileName = $fullClassName.'.php';
    if (is_file($fileName))
        include($fileName);
}
}

```

## Core.php

<?php

```
namespace models;
```

```
use core\Utils;
```

```
use Imagick;
```

```
class News extends \core\Model
```

```
{
```

```
    public function ChangePhoto($id, $file)
```

```
    {
```

```
        $folder = 'files/news/';
```

```
        $file_path = pathinfo($folder.$file);
```

```
        /** створення назв**/
```

```
        $file_big = $file_path['filename'].'_b';
```

```
        $file_middle = $file_path['filename'].'_m';
```

```
        $file_small = $file_path['filename'].'_s';
```

```
        $news = $this->GetNewsById($id);
```

```
        if (is_file($folder.$news['photo'].'_b.jpg') && is_file($folder.$file))
            unlink($folder.$news['photo'].'_b.jpg');
```

```
        if (is_file($folder.$news['photo'].'_m.jpg') && is_file($folder.$file))
            unlink($folder.$news['photo'].'_m.jpg');
```

```
        if (is_file($folder.$news['photo'].'_s.jpg') && is_file($folder.$file))
```

|      |      |                |        |      |   |      |
|------|------|----------------|--------|------|---|------|
|      |      | Гаєвський Д.О. |        |      | ДУ «Житомирська політехніка».24.121.05.000 - ПЗ | Арк. |
|      |      | Сьогодні І.І.  |        |      |   | 58   |
| Змн. | Арк. | № докум.       | Підпис | Дата |   |      |

```

        unlink($folder.$news['photo'].'_s.jpg');
        $news['photo'] = $file_path['filename'];

        /** зчитування, переробка й зберігання фото */
        $im_b = new Imagick();
        $im_b->readImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.$file);
        $im_b->cropThumbnailImage(900, 571, true);
        $im_b->
>writeImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.'/'.$file_big.'.jpg');

        $im_m = new Imagick();
        $im_m->readImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.$file);
        $im_m->cropThumbnailImage(340, 215, true);
        $im_m->
>writeImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.'/'.$file_middle.'.jpg');

        $im_s = new Imagick();
        $im_s->readImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.$file);
        $im_s->cropThumbnailImage(50, 50, true);
        //or $im_s->ThumbnailImage(180, 180, true, false); із збереженням пропорцій
        $im_s->
>writeImage($_SERVER['DOCUMENT_ROOT'].'/'.$folder.'/'.$file_small.'.jpg');
        unlink($folder.$file);
        $this->UpdateNews($news, $id);
    }

    public function AddNews($row)
    {
        $userModel = new \models\Users();
        $user = $userModel->GetCurrentUser();
        if ($user == null) {
            $result = [
                'error' => true,
                'messages' => ['Користувач не аутентифікований']
            ];
            return $result;
        }
        $validateResult = $this->Validate($row); /** валідація */
        if (is_array($validateResult)) {
            $result = [
                'error' => true,
                'messages' => $validateResult
            ];
            return $result;
        }
        $fields = ['title', 'type', 'description', 'text', 'price', 'drive',
'engine', 'mileage', 'color']; /** дані для таблиці */
        $rowFiltered = Utils::ArrayFilter($row, $fields);
    }

```

|      |      |                |        |      |   |      |
|------|------|----------------|--------|------|---|------|
|      |      | Гаєвський Д.О. |        |      | ДУ «Житомирська політехніка».24.121.05.000 - ПЗ | Арк. |
|      |      | Сьогодні І.І.  |        |      |   | 59   |
| Змн. | Арк. | № докум.       | Підпис | Дата |   |      |

```

        $rowFiltered['datetime'] = date('Y-m-d H:i:s');
        $rowFiltered['user_id'] = $user['id'];
        $rowFiltered['photo'] = '...photo...';
        $id = \core\Core::getInstance()->getDB()->insert('news', $rowFiltered); /**
Відправляємо в бд**/
        return [
            'error' => false,
            'id' => $id
        ];
    }

    public function GetLastNews($count)
    {
        return \core\Core::getInstance()->getDB()->select('news', '*', null,
['datetime'=>'DESC'], $count);
    }

    public function GetNewsByTitle($title)
    {
        $news = \core\Core::getInstance()->getDB()->select('news', '*',
['title'=>$title]);
        if (!empty($news))
            return $news[0];
        else
            return null;
    }

    public function UpdateNews($row, $id)
    {
        $userModel = new \models\Users();
        $user = $userModel->GetCurrentUser();
        if ($user == null)
            return false;
        $validateResult = $this->Validate($row);
        if (is_array($validateResult))
            return $validateResult;
        $fields = ['title', 'type', 'description', 'text', 'price', 'drive',
'engine', 'mileage', 'color', 'photo'];
        $rowFiltered = Utils::ArrayFilter($row, $fields);
        $rowFiltered['datetime_lastedit'] = date('Y-m-d H:i:s');
        //$rowFiltered['user_id'] = $user['id'];
        //$rowFiltered['photo'] = '...photo...';
        \core\Core::getInstance()->getDB()->update('news', $rowFiltered, ['id' =>
$id]);
        return true;
    }

```

|      |      |                |        |      |   |      |
|------|------|----------------|--------|------|---|------|
|      |      | Гаєвський Д.О. |        |      | ДУ «Житомирська політехніка».24.121.05.000 - ПЗ | Арк. |
|      |      | Сьогодні І.І.  |        |      |   | 60   |
| Змн. | Арк. | № докум.       | Підпис | Дата |   |      |