

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ	6
ВСТУП.....	8
1 ОГЛЯД І АНАЛІЗ АНАЛОГІЧНИХ СИСТЕМ.....	11
1.1 Характеристика предметної області.....	11
1.2 Огляд і аналіз аналогічних систем	13
1.2.1 Додаток Metview	13
1.2.2 Онлайн сервіс OpenWeatherMap	15
1.2.3 Система GrADS.....	17
1.2.4 Веб-система для обробки і візуалізації метеорологічних даних для дослідження навколишнього середовища.....	19
1.2.5 Веб-портал з екологічних наук «АТМОС».....	20
1.2.6 Порівняння аналогів.....	22
1.3 Постановка завдання	24
2 ВИБІР АРХІТЕКТУРИ СИСТЕМИ.....	26
2.1 Модель «файл-сервер» в технології БД.....	26
2.2 Дворівневі архітектура «клієнт-сервер».....	28
2.3 Архітектура сервера додатків	31
3 ВИБІР ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОГРАМ СИСТЕМИ.....	34
3.1 Вибір мови програмування.....	34
3.2 Вибір СУБД.....	35
3.3 Вибір системи управління контентом (CMS).....	36
3.4 Вибір хостингу	40
4 ПРОЕКТУВАННЯ І РОЗРОБКА БАЗИ ДАНИХ СИСТЕМИ	42
4.1 Побудова функціональної моделі.....	42
4.2 Побудова концептуальної моделі	49
5 ПРОЕКТУВАННЯ І РОЗРОБКА БАЗИ МОДЕЛЕЙ СИСТЕМИ	59
5.1 Загальні відомості про бази моделей і системи управління базами моделей.....	59
5.2 Deductor Studio Academic	60
5.2.1 Обробка і візуалізація гідрометеорологічних даних.....	63
5.2.2 Data mining	67
6 РОЗРОБКА ДОДАТКІВ СИСТЕМИ.....	74
ВИСНОВКИ.....	81
ПЕРЕЛІК ПОСИЛАНЬ	82

ДОДАТОК А АДМІНСТРУВАННЯ У WORDPRESS	85
А.1 Консоль WordPress	85
А.2 Створення контенту.....	89
ДОДАТОК Б КОД ВЕБ-СИСТЕМИ	Ошибка! Закладка не определена.

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

АІС	– автоматизована інформаційна система – це взаємозв'язана сукупність даних, обладнання, програмних засобів, персоналу, стандартних процедур, які призначені для збору, обробки, розподілу, зберігання, представлення інформації згідно з вимогами, які впливають з цілей організації
ПЗ	– програмне забезпечення
СППР	– система підтримки прийняття рішень
СУБМ	– системи управління базами моделей
Тайли	– невеликі зображення однакових розмірів, які служать фрагментами великої картини.
Тригер в БД	– це збережена процедура особливого типу, яку користувач не викликає явно, а використання якої обумовлено настанням визначеної події (дії) у реляційній базі даних
API	– Application Programming Interface, інтерфейс програми програмування
ASCII	– American Standard Code for Information Interchange, американський стандартний код для обміну інформацією
BUFR	– Binary Universal Form for the Representation of meteorological data, формат двійкових даних, що зберігається Всесвітньою метеорологічною організацією
CCIS	– Customer Control Information System, інформаційна система управління клієнтами
CMS	Content management system, система управління вмістом (контентом)
DNS	– Domain Name System, система доменних імен
ECMWF	European Centre for Medium-Range Weather Forecasts, – Європейський центр середньострокових прогнозів
ERP	– Enterprise Resource Planning System, система планування ресурсів підприємства
FORTTRAN	– це імперативна мова програмування загального призначення, яка особливо підходить для інтенсивних чисельних та наукових обчислень.
GRIB	– GRIdded Binary, бінарні сітки, математичний формат стислих даних, що часто використовується в метеорології для зберігання історичних і прогнозованих даних про

	погоду
GUI	– Graphical User Interface, графічний інтерфейс користувача
HDF	– Hierarchical Data Format, ієрархічний формат даних – формат файлів, розроблений для зберігання великого обсягу цифрової інформації.
IMS/DC	– Information Management System, Система управління інформацією
OLAP	– Online Analytical Processing, онлайн аналітична обробка – це технологія обробки інформації, що дозволяє швидко отримувати відповіді на багатовимірні аналітичні запити.
POST метод	– метод запиту, котрий використовується для завантаження файлу або подання заповненої веб-форми
RSS-фід	– Rich Site Summary, збагачена зведення сайту
TSO	– Time Sharing Option, варіант обміну часом
XA-протокол	– extended architecture, X/Open transaction interface protocol, розширена архітектура, протокол інтерфейсу X / Open транзакції – специфікація розподілених транзакцій, що визначає принципи спільної участі ресурсів в транзакції в умовах розподілених обчислень

ВСТУП

Згадки про різні атмосферні явища зустрічаються у більшості культурних народів давнини. З літописів Середньовіччя до нас дійшли відомості про різні явища природи, в тому числі бурях, грозах, ранніх снігопадах, сильних морозах, повенях і т.п. Записи про стан погоди, катастрофічних її явищах і кліматі містяться в давньоруських літописах. В епоху великих географічних відкриттів (XV і XVI ст.) з'явилися кліматичні описи країн, котрі відкривались. Однак метеорологія як наука тоді ще не сформувалася.

Перша найбільш чітка система знань про атмосферні явища була розроблена Аристотелем. Зародження метеорології як самостійної науки пов'язане з появою спеціальних приладів, таких як термометр, барометр, дощомір, адаптовані прилади для визначення швидкості і напрямку вітру. Початок спеціальних вимірювань за допомогою цих приладів пов'язують з першою половиною XVII ст., коли Галілеєм і його учнями були використані прилади та запропоновані методи вимірювань.

У другій половині XVIII ст. розпочало свою діяльність Мангеймське метеорологічне суспільство, яке створило мережу з 39 станцій з єдиними приладами.

У 1849 році в Петербурзі була організована Головна геофізична обсерваторія. У 1873 р. відбувся Перший міжнародний метеорологічний конгрес у Відні, а в 1879 р. був проведений вже другий конгрес, учасником якого був Д. І. Менделєєв. Після Другої світової війни при ООН була створена Всесвітня метеорологічна організація (ВМО), яка діє і в даний час.

Всесвітня метеорологічна організація сприяє розширенню гідрометеорологічного забезпечення народів усього світу, стежить за стандартами складу і термінів метеорологічних спостережень, приладів, точності вимірювань, виконує ряд інших важливих функцій.

Метеорологія – наука про земну атмосферу, її будову, властивості та явища і процеси, котрі в ній відбуваються. Предметом вивчення метеорології є атмосфера – повітряна оболонка, що оточує земну кулю [1].

Ключовими параметрами, які характеризують загальний стан атмосфери або основні тенденції в її розвитку, є: температура повітря, його тиск і вологість – абсолютна і відносна, напрямок і швидкість переміщення повітряних мас, стан неба і кількість опадів, які випадають на поверхню

землі з атмосфери у вигляді дощу, снігу, граду та ін., так званих гідрометеорів.

Метеорологічні спостереження здійснюються на спеціальних гідрометеорологічних станціях і постах. Мережа метеорологічних станцій і постів отримує інформацію про основні параметри, що характеризують фізичний стан атмосфери. Ця інформація служить основою для складання гідрометеорологічних прогнозів, визначення характеристик погоди і клімату, а також їх зміни під впливом природних факторів і господарської діяльності людини.

Структура багаторівневої інформаційно-обчислювальної системи складається з чотирьох основних систем: отримання інформації, збору і передачі інформації, обробки інформації, подання та доведення інформації до споживача.

Сільське господарство, авіація, автомобільний і морський транспорт, водозабезпечення, будівництво і боротьба з надзвичайними ситуаціями – всі ці сфери в тій чи іншій мірі залежать від погоди. А значить, потребують правильних і точних прогнозів.

Фахівці – гідрометеорологи забезпечують інформацією про погоду практично всі галузі економіки, щодня прогноз погоди надається в органи державної влади, МНС, а також в самих різних життєвих ситуаціях організації та приватні особи звертаються в Гідрометцентр за довідками про погодні умови і стан навколишнього середовища.

Тема є актуальною і може становити інтерес для спеціалістів комп'ютерних та гідрометеорологічних наук. Обробка гідрометеорологічних даних дає змогу в подальшому використовувати їх для машинного навчання. Аналіз гідрометеорологічних даних дозволяє виявляти певні тенденції у зміні клімату.

Метою магістерської роботи є розробка веб-системи для обробки і аналізу гідрометеорологічних даних.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- 1) огляд і аналіз аналогічних систем;
- 2) вибір архітектури системи;
- 3) вибір програмних засобів реалізації програм системи;
- 4) проектування і розробка бази даних системи;
- 5) проектування і розробка бази моделей системи;
- 6) розробка додатків системи.

Загальна кількість сторінок – 102, рисунків – 50, таблиць – 17, сторінок додатків – 22.

1 ОГЛЯД І АНАЛІЗ АНАЛОГІЧНИХ СИСТЕМ

1.1 Характеристика предметної області

У метеорології погода визначається як фізичний стан атмосфери в даний момент часу (при цьому, під «даним моментом часу» для різних характеристик маються на увазі різні терміни).

Фізичний стан атмосфери характеризують такі метеорологічні величини, як температура, тиск, вологість повітря, швидкість і напрям вітру, кількість хмар і опадів, що випали (дощу, снігу, граду), і такі атмосферні явища, як гроза, заметіль, туман, пилова буря. Сюди ж відносяться і оптичні явища – блакитний колір неба, веселка, гало, вінці. Повітря весь час рухається, і його фізичний стан безперервно змінюється. А це означає, що безперервно змінюються і метеорологічні величини в кожній точці Землі, тобто змінюється погода. Іншими словами, безперервно змінюються температура, тиск і вологість повітря, швидкість і напрям вітру, кількість і форма хмар, опадів. Зазвичай людей цікавить температура повітря, діапазон величин якої на Землі дуже великий.

Найбільш приємні для людини в середніх широтах температури від $+20^{\circ}\text{C}$ до $+25^{\circ}\text{C}$, але відчуття тепла залежить не тільки від температури, але і від вологості повітря, і від швидкості вітру. Тиск повітря то зростає, то падає величина його залежить і від географічної широти місцевості. Тиск повітря має найважливіше значення, тому що від його розподілу по земній кулі в кожен даний момент залежить рух повітря відносно земної поверхні, тобто вітер. Невеликий вітер (10-12 м/с) – приносить користь людині: вітер очищає міста від забруднення, надуває вітрила судна, обертає лопаті вітродвигунів, несе прохолоду в задушливий жаркий день. Однак як тільки його швидкість перевищує 15 м/с. вітер стає небезпечним і навіть руйнівним.

Вологість повітря важлива для тепло-почуття людини. Саме температура, вологість повітря і вітер визначають комфортні умови для людини.

Для характеристики погоди важлива хмарність, тобто ступінь закритості неба хмарами, а також форми хмар, які є свого роду поплавками, які показують рух повітря.

Безперервно відбуваються зміни в стані погоди обумовлені в першу чергу процесами загальної циркуляції атмосфери. Зміна дня і ночі визначає добовий хід метеорологічних елементів. Однак для погоди набагато більш

характерні різкі і нерегулярні зміни в результаті зміни повітряних мас, проходження атмосферних фронтів і еволюції циклонів і антициклонів. Повітряні маси, як і атмосферні вихори (циклони і антициклони), захоплюють відразу великі райони: їх горизонтальні розміри – тисячі кілометрів, тому погода в межах однієї і тієї ж повітряної маси на досить значному просторі матиме подібний характер. Якщо на територію вторгається нова повітряна маса, то відповідні зміни погоди відбуваються одночасно в багатьох районах. Отже, погода на даній території пов'язана з попереднім і наступним станами погоди в інших районах. Спостереження в одному пункті не можуть дати повного уявлення ні про причини, ні про сутність змін погоди.

Просторовий масштаб процесів, які формують погоду, такий, що їх найкраще вивчати за допомогою географічних карт. На карту в точках розташування метеорологічних станцій цифрами і умовними значками наносять результати спостережень, зроблені в один момент часу. Така карта називається синоптичної (тобто оглядовою) картою погоди, вона характеризує стан погоди одночасно на великій території, наприклад в Європі або у всій Північній півкулі. Синоптичні карти складаються через певні проміжки часу (наприклад, через кожні 12, 6 або 3 год.) і дають уявлення про розгортання процесів в часі. Синоптичні карти складаються як для поверхні Землі (приземні карти погоди), так і для різних шарів атмосфери (висотні карти погоди). Разом з вертикальними розрізами атмосфери і з графіками, побудованими за даними радіозондування, вони дають уявлення про тривимірну структуру атмосфери до висоти 25-30 км. Простежуючи по синоптичних картах, як і куди рухалися раніше і рухаються зараз повітряні маси, фронти, циклони і антициклони і що з ними відбувається, синоптик може скласти уявлення про загальний характер і механізм погоди у всьому географічному районі і на основі своїх висновків сформулювати попередній прогноз майбутньої погоди.

Існують самописні прилади, безперервно фіксують величини тиску, температури і відносної вологості. Однак неможливо, та й не потрібно передавати кожен секунду значення цих величин. На аеродромах спостереження за погодою виконують кожні 15 хв. На метеорологічних станціях всієї земної кулі прийнято вести спостереження за погодою кожні 3 год., рахуючи від 00 годин по Грінвічському середньому часу. Спостерігачі всієї Землі в 0, 3, 6, 9, 12, 15, 18 і 21 год. виходять на метеорологічні

майданчики і роблять відлік за приладами, що вимірює вищевказані метеорологічні величини [2].

На даний момент у всьому світі функціонує майже 12 000 [3] метеостанцій, які постійно фіксують зміни в погоді. Якщо брати виключно вимірювання температури по 8 разів на добу, ми отримаємо трохи менше 96 000 вимірювань, що стосуються тільки температури повітря. Для того, щоб обробити цю інформацію, людині необхідна наявність засобів для обробки і аналізу інформації.

1.2 Огляд і аналіз аналогічних систем

У якості аналогів для розгляду було обрано наступні системи: Metview, OpenWeatherMap, система GrADS, веб-система для обробки і візуалізації метеорологічних даних для дослідження навколишнього середовища, веб-портал з екологічних наук «АТМОС». Далі представлений огляд кожної системи.

1.2.1 Додаток Metview

Metview – додаток метеорологічної робочої станції, призначений для створення повного робочого середовища як для оперативного, так і для дослідницького метеоролога. Його можливості включають в себе потужний доступ до даних, обробку та візуалізацію.

Він має багатofункціональний користувальницький інтерфейс (рис. 1.1) на основі значків для інтерактивної роботи та мову сценаріїв для пакетної обробки. Ці дві складові зв'язані за допомогою можливості автоматичного перетворення значків в їх еквівалентний код сценарію [4].

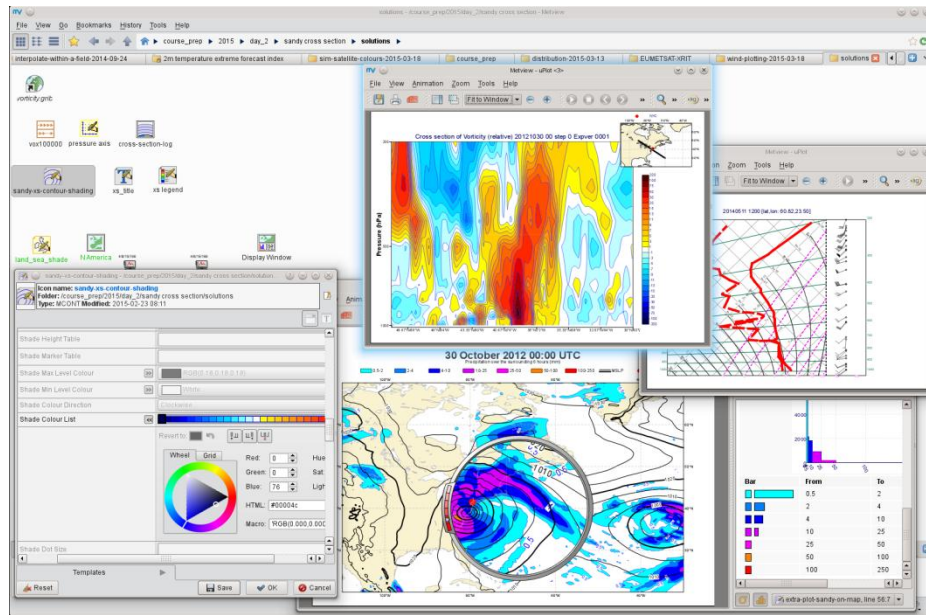


Рисунок 1.1 – Графічний інтерфейс Metview

Metview може приймати вхідні дані з різних джерел, в тому числі:

- Файли GRIB (видання 1 і 2);
- Файли BUFR;
- MARS (метеорологічний архів ECMWF);
- ODB (База даних спостережень);
- Місцеві бази даних;
- Файли даних ASCII (CSV, сітки і розкидані дані);
- NetCDF.

Також доступні потужні засоби фільтрації і обробки даних, і якщо потрібно графічний вихід, Metview може створювати безліч типів графіків, в тому числі:

- відображення в різних проекціях;
- поперечним перерізом;
- вертикальні профілі;
- графічні графіки x/y;
- інтелектуальне накладення даних з різних джерел на одній карті;
- розташування декількох графіків на одній сторінці.

Для того, щоб вирішити будь-яку задачу в додатку Metview необхідно провести певну послідовність операцій над іконками. Наприклад, є іконка, що представляє властивості географічної підкладки (включаючи налаштовані властивості проекції, берегової лінії, кордонів держав і т.п. із заданими товщиною ліній і квітами і т.п.), а також група іконок, що представляє дані

(наприклад, поля тиску і температури на різних висотах). Для відтворення підкладки потрібно вибрати з меню першої іконки метод «намалювати». Для зображення того чи іншого поля необхідно мишкою перетягнути потрібну іконку з даними на малюнок підкладки і на ньому буде відображене відповідне поле. Для відтворення іншого поля треба перенести на малюнок іконку з іншими даними, тим самим отримаємо нове зображення і т.д. Є ряд підготовлених розробниками іконок, для полегшення початку роботи з програмою [5].

Є велика технічна документація та навчальні курси як з графічного інтерфейсу, так і за мовою написання коду. Частина функціональності для зовнішніх користувачів (особливо для тих, хто не проживає в країнах учасницях ECMWF) обмежена. Зокрема це стосується вільного доступу до баз даних ECMWF через систему MARS.

1.2.2 Онлайн сервіс OpenWeatherMap

OpenWeatherMap – онлайн сервіс (рис. 1.2), який надає платний (є функціонально обмежена безкоштовна версія) API для доступу до даних про поточну погоду, прогнозами, для web-сервісів і мобільних додатків. Архівні дані доступні тільки на комерційній основі. Як джерело даних використовуються офіційні метеорологічні служби, дані з метеостанцій аеропортів, і дані з приватних метеостанцій.



Рисунок 1.2 – Зовнішній вигляд веб-системи OpenWeatherMap

Інформація обробляються OpenWeatherMap, після чого, на основі даних будується прогноз погоди і погодні карти, наприклад карти хмарності та опадів. Основною ідеєю сервісу OWM є використання приватних погодних станцій, які допомагають підвищити точність вихідної погодної інформації і, як наслідок, точність прогнозів погоди [6, 7].

В даний час OWM націлений на отримання прибутку і є комерційною організацією. Сервіс використовує карти OpenStreetMap для побудови погодних карт.

API OpenWeatherMap використовує платний API (є функціонально обмежений безкоштовний доступ), щоб надати дані поточної погоди, прогнозу і карт з погодними явищами, такими як хмари, вітер, тиск і опади. Всі погодні дані можуть бути отримані в форматах JSON, XML або HTML.

Відправка даних метеостанції проводиться по протоколу POST. Система OpenWeatherMap приймає наступні параметри: напрямок вітру, швидкість вітру, швидкість поривів вітру, температуру повітря, відносну вологість, атмосферний тиск, переважно без опадів (за останню годину, 24

години, з півночі), сніг за останні 24 години, яскравість, широту, довготу, висота, радіацію, точку роси, UV-індекс, назва метеостанції.

Поточні погодні дані можуть бути знайдені по місту (понад 200.000 доступних міст) або географічними координатами. Дані оновлюються кожні 10 хвилин.

Прогнози можуть бути знайдені по місту (понад 200.000 доступних міст) або географічними координатами. 3-годинні прогнози складаються на термін до 5 днів. Щоденні прогнози доступні на 16 днів.

Система геокодування OpenWeatherMap дозволяє знайти міста за назвою, країною, поштовим індексом або географічними координатами. Пошук можливий за частиною імені міста. Для того, щоб результати пошуку були більш точними, назва міста і країни повинні бути розділені комою.

OpenWeatherMap надає безліч карт погоди, включаючи карти опадів, хмарності, атмосферного тиску, температури, вітру та багато інших. Карти можуть бути підключені до мобільних додатків і веб-систем. Погодні карти можуть бути підключені в якості шарів для багатьох постачальників карт, в тому числі статичних тайлів, WMS, OpenLayers, Leaflet, карт Google, і Яндекс карт.

Метеостанції можуть бути підключені до OpenWeatherMap в 3 етапи:

- 1) необхідно зареєструватися в проєкті;
- 2) дані повинні відправлятися відповідно до API;
- 3) після підключення метеостанції слід перевірити правильність передачі даних на персональній сторінці OpenWeatherMap.org/my.

При відправці метеоданих в OpenWeatherMap методом POST, до HTTP заголовках, для авторизації, необхідно додати заголовок x-api-key зі значенням API ключа який можна знайти в особистому кабінеті.

1.2.3 Система GrADS

Система аналізу і відображення сітки (GrADS) являє собою інтерактивний інструмент для робочого столу, який використовується для легкого доступу, маніпуляції і візуалізації даних про земну науку. GRADS має дві моделі даних для обробки даних з прив'язкою до сітки і станції. GrADS підтримує безліч форматів файлів даних, включаючи виконавчі (потоківі або послідовні), GRIB (редакція 1 і 2), NetCDF, HDF (версії 4 і 5) і BUFR (для даних станції). GrADS впроваджується в усьому світі в різних широко використовуваних операційних системах і вільно поширюється через Інтернет.

GRADS використовує 5-мірну середу даних: чотири звичайних вимірювання (довгота, широта, вертикальний рівень і час) плюс додатковий 5-й вимір для сіток, який звичайно реалізується, але призначений для використання в ансамблях. Набори даних поміщаються в 5-мірний простір з використанням файлу дескриптора даних. GrADS обробляє сітки, які є регулярними, нелінійно рознесеними, гауссовими або зі змінною роздільною здатністю. Дані з різних наборів даних можуть бути графічно накладені, з правильною просторовою і тимчасовою реєстрацією. Операції виконуються в інтерактивному режимі, вставляючи FORTRAN-подібні вирази в командному рядку. Надається багатий набір вбудованих функцій, але користувачі можуть також додавати свої власні функції в якості зовнішніх процедур, написаних на будь-якій мові програмування.

Дані можуть відображатися з використанням різних графічних методів (рис 1.3): лінійних і гістограм, графіків розкиду, згладжених контурів, заштрихованих контурів, ліній струму, векторів вітру, графіків моделі станції і т.д. Графіка може виводитися в форматах PostScript або зображень. GrADS надає геофізично інтуїтивні значення за замовчуванням, але користувач має можливість контролювати всі аспекти виведення графіки.

GrADS має програмований інтерфейс (мова сценаріїв), який дозволяє використовувати складні програми для аналізу і відображення. Можливе використання сценаріїв для відображення кнопок і спадаючих меню, а також графіки, а потім виконання дії на основі призначених для користувача точок і клацань. GRADS можна запускати в пакетному режимі, а мова сценаріїв полегшує використання GrADS для виконання тривалих пакетних завдань [8].

Examples of GrADS Used on the Web to Deliver Images

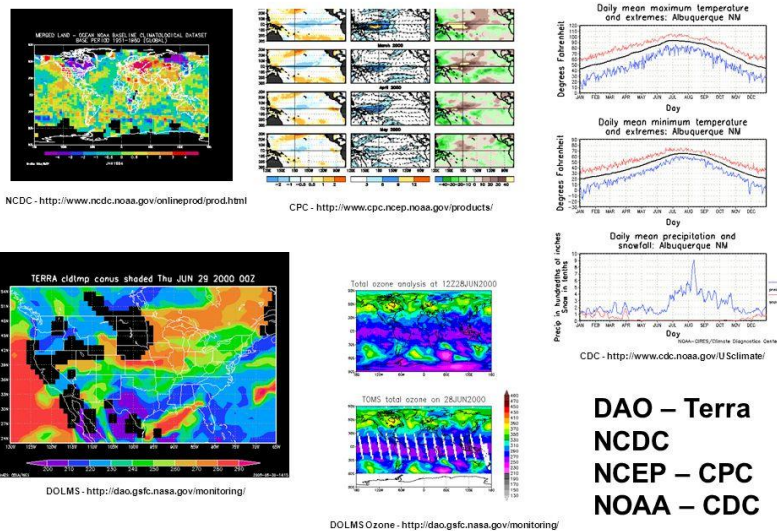


Рисунок 1.3 – Результати роботи системи GrADS

1.2.4 Веб-система для обробки і візуалізації метеорологічних даних для дослідження навколишнього середовища

Інформаційно-обчислювальна система для візуалізації та обробки метеорологічних даних є спеціалізованим веб-додатком, створеним на основі ядра порталу ATMOS. Вона дозволяє проводити різні математичні та статистичні операції над даними з метою дослідження тенденцій зміни глобальних і регіональних характеристик клімату. На даний момент для обробки доступні дані реаналізів NCEP/NCAR і NCEP/DOE AMIP II. Реалізована можливість побудови тимчасових трендів метеорологічних і кліматичних параметрів для обох реаналізів і їх порівняння. Для візуалізації результатів використовувався пакет GrADS з відкритим кодом. Графічний інтерфейс розроблений з використанням мов HTML, PHP і JavaScript і являє собою динамічну форму для введення параметрів розрахунку і візуалізації. Програми представляють з себе незалежні модулі, що підключаються за допомогою PHP і виконуються системою GrADS, яка після закінчення розрахунків виробляє графічний висновок результатів в файл. Цей файл в подальшому передається в ядро системи для відображення на веб-сторінці. Методані в структурованому вигляді зберігаються на жорсткому диску сервера і доступні тільки для обробки системою. Користувач не має безпосереднього доступу до даних і не може завантажити їх, однак вільно може отримати результати графічного відображення, як даних, так і результатів їх обробки [9-11].

В системі обробляється кілька ключових параметрів, що характеризують загальний стан клімату або основні тенденції в його змінах (температура, тиск, вологість атмосфери, температура і вологість ґрунту, а також рівень опадів).

Система дозволяє виконувати різні математичні та статистичні операції над метеоданими, а саме – обчислення середнього значення, стандартного відхилення, визначення першого (останнього) теплого (холодного) дня (тижня, місяця) на рік, підрахунок числа днів з кількістю опадів із заданого інтервалу значень та ін. Наявність декількох масивів даних дає можливість для порівняння ключових метеорологічних і кліматичних характеристик. Інтерфейс (рис 1.4) дає можливість задавати період осереднення даних

(попередні установки: для дня, тижня, місяця і т.д., а також задається користувачем), необхідну географічну область і часовий діапазон (з можливістю задати кілька безперервних тимчасових інтервалів). Крім того, є можливість задати набір параметрів візуалізації даних і результатів обробки: два типи відображення карти (контурна і контурна зафарбована) і розмір зображення. Після вибору потрібного параметра, типу обробки і візуалізації система видає користувачеві результат розрахунків у вигляді графічного зображення, побудованого за допомогою системи GrADS.

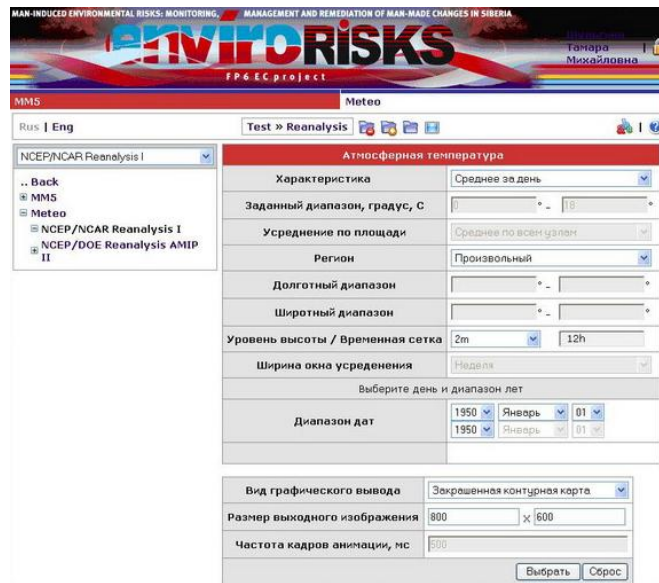


Рисунок 1.4 – Інтерфейс для роботи з веб-системою

Також в системі доступна функція побудови часових трендів метеорологічних і кліматичних параметрів, що є однією з основних характеристик, що сприяє виявленню загальної довгострокової тенденції в зміні метеорологічних величин в часовому ряді.

1.2.5 Веб-портал з екологічних наук «АТМОС»

Розроблений в рамках грантового веб-порталу INTAS ATMOS (рис. 1.5) [13] надає доступ до міжнародної науково-дослідницької спільноти, екологічним менеджерам і зацікавленій громадськості, двомовному джерелу інформації для області фізики атмосфери і хімії і пов'язаній області застосування оцінки та управління якістю повітря.

Він пропонує доступ до інтегрованої тематичної інформації, експериментальними даними, аналітичних інструментів і моделей, тематичних досліджень та пов'язаних з ними інформації та освітніх ресурсів, зібраним, структурованим і відредагованим партнерами в узгоджений і послідовний тематичний інформаційний ресурс. Пропонуючи звичайні компоненти тематичної системи, такі як колекції посилань, реєстрація груп користувачів, дискусійний форум, розділ новин і т.д. Система відрізняється своїми науковими інформаційними сервісами та інструментами: он-лайн моделі і аналітичні інструменти, а також збір даних і тематичні дослідження разом з навчальними матеріалами.

Климатическая модель

Guest |


География | Климат | **Реки, озера, болота** | Почва | Растительность | Животный мир | Экология | ГИС | Природные комплексы

Rus | Eng

Портал ATMOS
Западная Сибирь
Байкал
Оценка и управление качеством воздуха
Климат
Новости

Обширный край, издавна называющийся Западной Сибирью, расположен в срединной части территории России. Его составляют одна из самых крупных низменных равнин земного шара – Западно-Сибирская равнина – и части гор Южной Сибири: Кузнецкий Алатау, Салаирского кряжа, Горной Шории и расположенной южнее Алтайской горной страны. Площадь Западной Сибири составляет 2427200 км², т.е. 10,8 % от всей территории бывшего СССР. (Западная Сибирь, 1963. С. 9).

Климат огромной территории Западной Сибири, которая простирается с севера на юг почти на 3000 км (от вечных льдов до сухих казахстанских и монгольских степей), отличается огромным разнообразием.



Реки Западной Сибири принадлежат бассейну Карского моря. Самая крупная водная артерия – Обь с притоком Иртыш – относится к числу величайших рек земного шара. Река Обь образуется из слияния Бии и Катунь, берущих начало в Алтае, и впадает в Обскую губу Карского моря. Она представляет собой типичную равнинную реку с малым падением русла, обширной долиной и поймой, изобилующей протоками, а местами озёрами и болотами. Иртыш – самый большой левый приток Оби берёт начало на территории Китайской Народной Республики, на склонах Монгольского Алтая.

Обширная территория Западной Сибири, её равнинность, суровый климат определяют большое разнообразие типов почв, существенно отличающихся от Европейской части России или Восточной Сибири.

Рисунок 1.5 – Зовнішній вигляд порталу

Портал організований як набір взаємозв'язаних наукових систем, в яких розглядаються основні напрямки досліджень в області атмосферних наук і моделювання клімату, а також прикладні області оцінки та управління якістю повітря, моделювання та оцінки впливу на навколишнє середовище. Кожна наукова система є відкритою для інформаційно-обчислювальної системи зовнішнього доступу, що реалізовується за допомогою інтернет-технологій. Основні наукові теми присвячені атмосферній хімії, атмосферній спектроскопії і радіації, атмосферним аерозолям, атмосферній динаміці і атмосферним моделям, включаючи кліматичні моделі.

Портал ATMOS відображає поточну тенденцію перетворення наук про навколишнє середовище в точні (кількісні) науки і є досить ефективним

прикладом інтеграції сучасних інформаційних технологій та екологічних наук. Це робить портал і допоміжним інструментом для підтримки міждисциплінарних проектів регіонального середовища та великого освітнього ресурсу в цій галузі [12].

1.2.6 Порівняння аналогів

Переваги та недоліки представлених раніше аналогів наведені нижче (табл. 1.1).

Таблиця 1.1 – Переваги і недоліки систем Metview, OpenWeatherMap, GrADS

Назва системи	Переваги	Недоліки
1	2	3
Веб-система для обробки і візуалізації метеорологічних даних	<ul style="list-style-type: none"> – веб-система; – добре розвинений інтерфейс; – дозволяє різноманітні обчислення; – наявність декількох масивів даних дає можливість для порівняння ключових метеорологічних і кліматичних характеристик; – є можливою побудова часових трендів метеорологічних і кліматичних параметрів, що є однією з основних характеристик, що сприяє виявленню загальної довгострокової тенденції в зміні метеорологічних величин в часовому ряді. 	<ul style="list-style-type: none"> – немає можливості завантажити оброблені данні з системи.

Продовження таблиці 1.1

1	2	3
---	---	---

OpenWeatherMap	<ul style="list-style-type: none"> – он-лайн сервіс; – API для доступу до даних про поточну погоду, прогнозами, для web-сервісів і мобільних додатків – джерелом даних є офіційні метеорологічні служби, дані з метеостанцій аеропортів, і дані з приватних метеостанцій; – безліч карт погоди, які можуть бути підключені до мобільних додатків і веб-систем; – потужний пошук метеоданих по містам та координатам. 	<ul style="list-style-type: none"> – платний API; – архівні дані доступні тільки на комерційній основі.
GrADS	<ul style="list-style-type: none"> – легкий доступ, маніпуляція і візуалізація даних про земну науку; – підтримує безліч форматів файлів даних; – впроваджується в усьому світі в різних широко використовуваних операційних системах і вільно поширюється через Інтернет; – дані можуть відображатися з використанням різних графічних методів. 	<ul style="list-style-type: none"> – десктопний додаток.
Веб-портал «АТМОС»	<ul style="list-style-type: none"> – он-лайн моделі і аналітичні інструменти; – збір даних; – тематичні дослідження разом з навчальними матеріалами; – дискусійний форум. 	<ul style="list-style-type: none"> – погано розроблений інтерфейс користувача
Metview	<ul style="list-style-type: none"> – велика різноманітність форматів даних, що приймаються на вході; – зручний призначений для користувача інтерфейс; – потужні засоби фільтрації та обробки даних; – велика кількість типів графіків для відображення даних; – детальна документація. 	<ul style="list-style-type: none"> – працює тільки під Linux; – десктопний додаток.

Виходячи з наведених вище характеристик було прийнято рішення створити веб-систему (назва системи *meteodata processing*) для обробки і аналізу гідрометеорологічних даних.

1.3 Постановка завдання

Далі наведені головні завдання, котрі необхідно виконати в ході розробки системи.

Як було зазначено раніше, головною метою магістерської роботи є розробка веб-системи для обробки і аналізу гідрометеорологічних даних.

Веб-система повинна дозволяти завантажувати користувачеві гідрометеорологічні дані в систему у вигляді файлу, виконувати обробку завантажених даних і надавати можливість завантажити оброблені дані для подальшого використання. Також система буде забезпечувати користувачів можливістю обговорення будь-яких питань стосовно веб-системи та метеорології на форумі системи.

Для того щоб система залишалася актуальною тривалий час, необхідно забезпечувати взаємодію адміністрації ресурсу і його користувачів. Для цього на ньому необхідно підтримувати форум і розділ новин, який передбачав би публікацію і модерацію інформації. Також буде створено розділ, в якому користувач зможе звернутися до адміністрації системи на пряму без додаткової реєстрації.

Для підтримки системи в належному стані, а саме, наявність актуальної інформації, модерації форумів, призначення прав доступу, необхідний адміністратор і модератор системи.

Навігація у системі повинна бути інтуїтивно зрозумілою, для того, щоб користувач зміг швидко адаптуватися і вирішити необхідне йому завдання.

Будуть визначені такі категорії користувачів як гість, зареєстрований користувач, адміністратор.

Гість зможе переглядати загальну інформацію і новини у системі, коментувати сторінки системи, також буде мати доступ до перегляду форумів, однак не зможе залишити в них повідомлення.

Зареєстрований користувач отримає можливість завантажувати в систему файли з гідрометеорологічними даними, які необхідно обробити, вибирати спосіб обробки, отримувати оброблені дані у вигляді графіків, а також завантажувати оброблені дані на свій комп'ютер. Крім всіх

перерахованих можливостей гостя, користувач також може залишати повідомлення на форумі.

Адміністратор має доступ до адміністраторської панелі, де він керує БД, користувачами, новинами і всім наповненням системи.

Система повинна підтримувати на багато користувачів режим роботи, для зберігання і маніпуляції інформацією і всіх відомостей необхідна

СУБД, для зручної роботи користувачів потрібно створити додаток з використанням веб-технологій.

2 ВИБІР АРХІТЕКТУРИ СИСТЕМИ

Далі були обрані архітектура системи та її функціональні можливості.

2.1 Модель «файл-сервер» в технології БД

В архітектурі «файл-сервер» (рис. 2.1) засоби організації і управління БД (в тому числі СУБД) розташовуються на машині клієнта, а БД, що представляє собою набір спеціалізованих файлів – на машині-сервері.

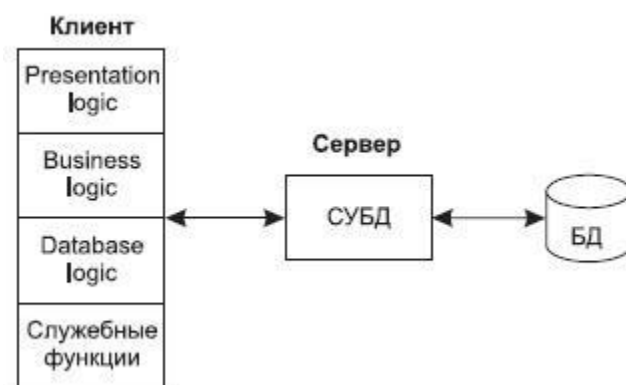


Рисунок 2.1 – Архітектура файл-сервер

В цьому випадку серверна компонента представлена мережевими складовими ОС, що забезпечують віддалений розподілений доступ до файлів. Запит до БД, сформульований на мові маніпулювання даними, перетворюється СУБД в послідовність команд введення-виведення, які обробляються операційною системою машини-сервера.

Функції роботи з базами даних можна розділити на 5 груп, що мають різну природу:

- 1) функції введення і відображення даних (Presentation Logic);
- 2) прикладні функції, що визначають основні алгоритми вирішення завдань програми (Business Logic);
- 3) функції обробки даних всередині програми (Database Logic);
- 4) функції управління інформаційними ресурсами (Database Manager System);
- 5) службові функції, які відіграють роль зв'язок між функціями перших чотирьох груп.

Презентаційна логіка (Presentation Logic) як частина програми визначається тим, що користувач бачить на своєму екрані, коли працює додаток. Сюди відносяться всі екранні форми інтерфейсу, які користувач бачить або заповнює в ході роботи програми, до цієї ж частини відносяться все те, що виводиться користувачеві на екран як результати вирішення деяких проміжних завдань або як довідкова інформація. Тому основними завданнями презентаційної логіки є:

- формування екранних зображень;
- читання і запис в екранні форми інформації;
- управління екраном;
- обробка рухів миші і натиснення клавіш клавіатури.

Деякі можливості для організації презентаційної логіки додатків надає знако-орієнтований інтерфейс, що задається моделями CCIS (Customer Control Information System) і IMS/DC фірми IBM і моделлю TSO (Time Sharing Option) для централізованої main-фреймової архітектури. Модель GUI – графічного користувацького інтерфейсу, підтримується в операційних середовищах Microsoft's Windows, Windows NT, в OS/2 Presentation Manager, X-Windows і OSF/Motif.

Бізнес-логіка, або логіка власне додатків (Business processing Logic), – це частина коду програми, яка визначає власне алгоритми вирішення конкретних завдань програми. Зазвичай цей код пишеться з використанням різних мов програмування, таких як C, C++, Cobol, SmallTalk, Visual-Basic.

Логіка обробки даних (Data manipulation Logic) – це частина коду програми, яка пов'язана з обробкою даних всередині програми. Даними управляє власне СУБД. Для забезпечення доступу до даних використовують мову запитів і засоби маніпулювання даними стандартної мови SQL.

Процесор управління даними (Database Manager System Processing) – це власне СУБД, яка забезпечує зберігання і управління базами даних. В ідеалі функції СУБД повинні бути приховані від бізнес-логіки додатка.

У централізованій архітектурі (Host-based processing) ці частини програми розташовуються в єдиному середовищі і комбінуються усередині однієї програми, що виконується [14].

Недоліки архітектури:

- високе завантаження мережі і машин-клієнтів, так як обмін йде на рівні одиниць інформації файлової системи (фізичних записів, блоків, файлів);

- низький рівень захисту даних, так як доступ до файлів БД керується загальними засобами ОС сервера;
- бізнес-правила функціональної обробки, зосереджені в клієнтській частині, можуть бути суперечливими.

2.2 Дворівневі архітектура «клієнт-сервер»

Дворівнева модель фактично є результатом розподілу п'яти зазначених функцій між двома процесами, які виконуються на двох платформах: на клієнті і на сервері. Було розглянуто наступні моделі:

- виділений сервер бази даних;
- активний сервер баз даних.

В архітектурі «виділений сервер бази даних» (рис. 2.2) засоби управління базою даних і база даних розміщені на машині-сервері (DB-сервер).

У такій моделі база даних зберігається на сервері. На сервері ж знаходиться ядро СУБД. На клієнті розташовується презентаційна логіка і бізнес-логіка програми. Клієнт звертається до сервера з запитом на мові SQL.

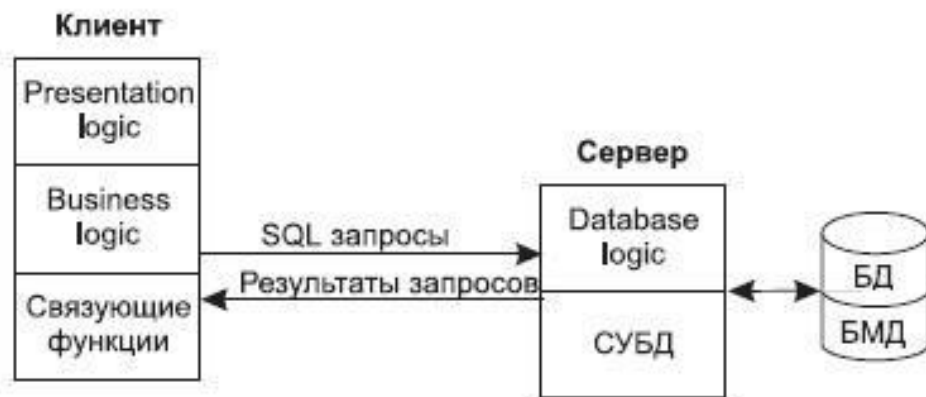


Рисунок 2.2 – Архітектура з виділеним сервером бази даних

Звернення до БД здійснюється на мові SQL, тому сервер БД часто називають SQL-сервером. Він підтримується всіма реляційними СУБД (Oracle, Informix, MS SQL, DB2, ADABAS D, InterBase, SyBase). Сервер БД здійснює пошук записів і аналізує їх. Записи, що задовольняють умовам, можуть накопичуватися на сервері і після обробки запиту передаватися

користувачу. Клієнтську програму може бути реалізовано на мові настільних СУБД (MS Access, FoxPro, Paradox, Clipper). Взаємодія клієнтського додатка з SQL-сервером здійснюється через ODBC-драйвер (Open DataBase Connectivity). ODBC став стандартом де-факто на алгоритм доступу до різномірних БД.

Переваги архітектури:

- зниження навантаження на машини сервера і клієнтів;
- зниження мережевого трафіку і підвищення ефективності обробки за рахунок оптимізації і буферизації введення-виведення;
- захист даних засобами СУБД, що дозволяє блокувати недозволені користувачеві дії;
- сервер реалізує управління транзакціями і може блокувати спроби одночасного зміни одних і тих же записів.

Недоліки архітектури:

- бізнес-логіка функціональної обробки і представлення даних можуть бути однаковими для кількох клієнтських додатків, що збільшує потреби в ресурсах (повторення коду програм і запитів);

Архітектура «активний сервер баз даних». Для того щоб позбутися недоліків моделі віддаленого доступу, повинні бути дотримані наступні умови:

- 1) необхідно, щоб БД в кожен момент відображала поточний стан предметної області, яке визначається не тільки власне даними, але і зв'язками між об'єктами даних. Тобто дані, які зберігаються в БД, в кожен момент часу повинні бути не суперечливими;
- 2) БД повинна відображати деякі правила предметної області, закони, за якими вона функціонує (business rules);
- 3) необхідний постійний контроль за станом БД, відстеження всіх змін і адекватна реакція на них;
- 4) необхідно, щоб виникнення деякої ситуації в БД чітко і оперативно впливало на хід виконання прикладної задачі.

Однією з найважливіших проблем СУБД є контроль типів даних. На даний момент СУБД контролює синтаксично тільки стандартно-допустимі типи даних, тобто такі, які визначені в DDL (data definition language) – мові опису даних, який є частиною SQL. Однак в реальних предметних областях у нас діють дані, які несуть в собі ще й семантичну складову, наприклад, це координати об'єктів або одиниці різних метрик, наприклад робочий тиждень на відміну від реальної має відразу після п'ятниці понеділок.

Дану модель підтримують більшість сучасних СУБД: Informix, Ingres, Sybase, Oracle, MS SQL Server. Основу даної моделі становить механізм збережених процедур як засіб програмування SQL-сервера, механізм тригерів як механізм відстежування поточного стану інформаційного сховища і механізм обмежень на призначені для користувача типи даних, який іноді називається механізмом підтримки доменної структури. Модель сервера баз даних представлена на рис. 2.3.



Рисунок 2.3 – Модель активного сервера БД

У цій моделі бізнес-логіка розділена між клієнтом і сервером. На сервері бізнес-логіка реалізована у вигляді збережених процедур – спеціальних програмних модулів, які зберігаються в БД і управляються безпосередньо СУБД. Клієнтська програма звертається до сервера з командою запуску збереженої процедури, а сервер виконує цю процедуру і реєструє всі зміни в БД, які в ній передбачені. Сервер повертає клієнту дані, релевантні його запиту, які потрібні клієнту або для виведення на екран, або для виконання частини бізнес-логіки, яка розташована на клієнті. Трафік обміну інформацією між клієнтом і сервером різко зменшується.

Централізований контроль в моделі сервера баз даних виконується з використанням механізму тригерів. Тригери також є частиною БД.

Термін «тригер» взято з електроніки і семантично дуже точно характеризує механізм відстеження спеціальних подій, які пов'язані зі станом БД. Тригер в БД є як би деяким тумблером, який спрацьовує при виникненні певної події в БД. Ядро СУБД проводить моніторинг всіх подій, які викликають створені і описані тригери в БД, і при виникненні відповідної події сервер запускає відповідний тригер. Кожен тригер являє собою також

деяку програму, яка виконується над базою даних. Тригери можуть викликати збережені процедури.

Механізм використання тригерів передбачає, що при спрацьовуванні одного тригера можуть виникнути події, які викличуть спрацьовування інших тригерів. Цей потужний інструмент вимагає тонкого і узгодженого застосування, щоб не вийшов нескінченний цикл спрацьовування тригерів.

У даній моделі сервер є активним, тому що не тільки клієнт, але і сам сервер, використовуючи механізм тригерів, може бути ініціатором обробки даних в БД.

І збережені процедури, і тригери зберігаються в словнику БД, вони можуть бути використані декількома клієнтами, що істотно зменшує дублювання алгоритмів обробки даних в різних клієнтських додатках.

Недоліком даної моделі є дуже велике завантаження сервера. Дійсно, сервер обслуговує безліч клієнтів і виконує наступні функції:

- 1) здійснює моніторинг подій, пов'язаних з описаними тригерами;
- 2) забезпечує автоматичне спрацьовування тригерів при виникненні пов'язаних з ними подій;
- 3) забезпечує виконання внутрішньої програми кожного тригера;
- 4) запускає збережені процедури по запитам користувачів;
- 5) запускає збережені процедури з тригерів;
- 6) повертає необхідні дані клієнта;
- 7) забезпечує всі функції СУБД: доступ до даних, контроль і підтримку цілісності даних в БД, контроль доступу, забезпечення коректної паралельної роботи всіх користувачів з єдиною БД.

Якщо ми переклали на сервер більшу частину бізнес-логіки додатків, то вимоги до клієнтів в цій моделі різко зменшуються. Іноді таку модель називають моделлю з «тонким клієнтом», на відміну від попередніх моделей, де на клієнта покладалися набагато серйозніші завдання. Ці моделі називаються моделями з «товстим клієнтом» [14].

2.3 Архітектура сервера додатків

Для розвантаження сервера була запропонована трирівнева модель.

Ця модель є розширенням дворівневої моделі і в ній вводиться додатковий проміжний рівень між клієнтом і сервером. Архітектура трирівневої моделі приведена на рис. 2.4. Цей проміжний рівень містить один або кілька серверів додатків.



Рисунок 2.4 – Модель сервера додатків

У цій моделі компоненти програми діляться між трьома виконавцями:

- 1) клієнтами;
- 2) серверами додатків;
- 3) серверами БД.

Клієнт забезпечує логіку уявлення, включаючи графічний користувальницький інтерфейс, локальні редактори; клієнт може запускати локальний код додатка клієнта, який може містити звернення до локальної БД, розташованої на комп'ютері-клієнті. Клієнт виконує комунікаційні функції front-end частини програми, які забезпечують доступ клієнту в локальну або глобальну мережу. Додатково реалізація взаємодії між клієнтом і сервером може включати в себе управління розподіленими транзакціями, що відповідає тим випадкам, коли клієнт також є клієнтом менеджера розподілених транзакцій.

Сервери додатків складають новий проміжний рівень архітектури. Вони спроектовані для виконання загальних не завантажуваних функцій для клієнтів. Сервери додатків підтримують функції клієнтів як частин взаємодіючих робочих груп, підтримують мережеву доменну операційну середу, зберігають і виконують найбільш загальні правила бізнес-логіки, підтримують каталоги з даними, забезпечують обмін повідомленнями і підтримку запитів, особливо в розподілених транзакцій.

Сервери баз даних в цій моделі займаються виключно функціями СУБД: забезпечують функції створення та ведення БД, підтримують цілісність реляційної БД, забезпечують функції сховищ даних (warehouse services). Крім того, на них покладаються функції створення резервних копій БД і відновлення БД після збоїв, управління виконанням транзакцій і підтримки застарілих (успадкованих) додатків (legacy application).

Відзначимо, що ця модель має більшу гнучкість, ніж дворівневі моделі. Найбільш помітні переваги моделі сервера додатків в тих випадках, коли клієнти виконують складні аналітичні розрахунки над базою даних, які відносяться до області OLAP-додатків. (On-line analytical processing.) У цій моделі велика частина бізнес-логіки клієнта ізольована від можливостей вбудованого SQL, реалізованого в конкретній СУБД, і може бути виконана на стандартних мовах програмування, таких як C, C ++, SmallTalk, Cobol. Це підвищує переносимість системи, її масштабованість.

Функції проміжних серверів можуть бути в цій моделі розподілені в рамках глобальних транзакцій шляхом підтримки XA-протоколу (X/Open transaction interface protocol), який підтримується більшістю постачальників СУБД [14].

В нашому випадку буде використана дворівнева архітектура.

3 ВИБІР ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОГРАМ СИСТЕМИ

3.1 Вибір мови програмування

PHP – скриптова мова загального призначення, інтенсивно застосовується для розробки веб-додатків. В даний час підтримується переважною більшістю хостинг-провайдерів і є одним з лідерів серед мов, що застосовуються для створення динамічних веб-сайтів та веб-систем.

В області веб-програмування, зокрема, серверної частини, PHP – один з популярних сценарних мов (разом з JSP, Perl, Python і мовами, використовуваними в ASP.NET).

Популярність в області побудови веб-систем визначається наявністю великого набору вбудованих засобів для розробки веб-додатків. Основні з них:

- автоматичне вилучення POST і GET-параметрів, а також змінних оточення веб-сервера в зумовлені масиви;
- взаємодія з великою кількістю різних СУБД;
- автоматизована відправка HTTP-заголовків;
- робота з HTTP-авторизацією;
- робота з cookies і сесіями;
- робота з локальними і віддаленими файлами, сокетом;
- обробка файлів, що завантажуються на сервер;
- робота з XForms.

В даний час PHP використовується сотнями тисяч розробників. Згідно з рейтингу корпорації ТЮВЕ, що базується на даних пошукових систем, в травні 2016 року PHP знаходився на 6-му місці серед мов програмування. До найбільших сайтів, які використовують PHP, відносяться Facebook, Wikipedia та ін.

Входить в LAMP – поширений набір програмного забезпечення для створення та хостингу веб-сайтів (Linux, Apache, MySQL, PHP).

Хоча PHP і не дуже поширений в створення GUI-додатків, його можна використовувати і в даній області.

Для створення кроссплатформених додатків служать пакети PHP-GTK і PHP-Qt, що представляють собою обгортки для відповідних популярних бібліотек віджетів. Також існує середовище розробки кроссплатформених додатків Devel Next.

Для створення графічних додатків для Windows існують вільні пакети WinBinder (написаний на Cі, фактично – обгортка для WinAPI), PQBuilder (написаний на PHP з використанням бібліотеки PHPQt5), а також попередник Devel Next – середовище швидкої розробки Devel Studio.

Крім цього існує реалізація PHP для .NET/Mono – Phalanger і для JVM – JPHP, результатом компіляції PHP-коду в Phalanger може бути будь-який .NET-додаток, в той же час JPHP підтримує розширення Swing, майже повністю портоване з середовища Java [15].

3.2 Вибір СУБД

MySQL – це найпоширеніша повноцінна серверна СУБД. MySQL дуже функціональна, СУБД вільно розповсюджується та успішно працює з різними сайтами і веб-додатками. Навчитися використанню цієї СУБД досить просто, так як на просторах інтернету ви легко знайдете більшу кількість інформації.

Замітка: варто зауважити, що завдяки популярності цієї СУБД, існує величезна кількість різних плагінів і розширень, що полегшують роботу з системою.

Незважаючи на те, що в ній не реалізований весь SQL функціонал, MySQL пропонує досить багато інструментів для розробки додатків. Так як це серверна СУБД, додатки для доступу до даних, на відміну від SQLite працюють зі службами MySQL.

Переваги MySQL

- 1) простота в роботі – встановити MySQL досить просто. Існують різноманітні програми, наприклад GUI, дозволяє досить легко працювати з БД;
- 2) багатий функціонал – MySQL підтримує більшість функціоналу SQL;
- 3) безпека – велика кількість функцій забезпечують безпеку, які підтримується за замовчуванням;
- 4) масштабованість – MySQL легко працює з великими обсягами даних і легко масштабується;
- 5) швидкість – спрощення деяких стандартів дозволяє MySQL значно збільшити продуктивність.

Недоліки MySQL

- 1) Відомі обмеження – за задумом в MySQL закладені деякі обмеження функціонала, які іноді необхідні в особливо вимогливих додатках.
- 2) Проблеми з надійністю – через деяких способів обробки даних MySQL (зв'язку, транзакції, аудити) іноді поступається іншим СУБД по надійності.
- 3) Повільна розробка – Хоча MySQL технічно відкрите ПЗ, існують скарги на процес розробки. Варто зауважити, що існують інші досить успішні СУБД створені на базі MySQL, наприклад MariaDB.

Коли слід використовувати MySQL

- розподілені операції – якщо функціоналу SQLite не вистачає, то варто розглянути MySQL. Так як ця СУБД поєднує в собі просунутий функціонал і вільний доступ до вихідного коду.
- високий рівень безпеки – система безпеки MySQL включає в себе прості і в той же час гідні способи захисту доступу до даних
- веб-сайти і веб-додатки – більшість сайтів та онлайн додатків спокійно працюють з MySQL незважаючи на деякі обмеження. Будучи легкою в налаштуванні і масштабується – MySQL перевірена часом.
- індивідуальні рішення – якщо ви працюєте з якихось специфічних проектом, MySQL легко зможе вам допомогти завдяки широким можливостям в налаштуванні і функціоналом [16].

3.3 Вибір системи управління контентом (CMS)

Система управління вмістом (контентом) (Content management system, CMS) – інформаційна система або комп'ютерна програма, яка використовується для забезпечення і організації спільного процесу створення, редагування і управління вмістом, інакше – контентом.

Основні функції CMS:

- надання інструментів для створення вмісту, організація спільної роботи над вмістом;
- управління вмістом: зберігання, контроль версій, дотримання режиму доступу, управління потоком документів і т.д.;
- публікація вмісту;

- подання інформації у вигляді, зручному для навігації, пошуку.

В системі керування вмістом можуть перебувати найрізноманітніші дані: документи, фільми, фотографії, номери телефонів, наукові дані і так далі. Така система часто використовується для зберігання, управління, перегляду і публікації документації. Контроль версій є одним з основних її переваг, коли вміст змінюється групою осіб [17].

Більшість сучасних веб-систем працюють з використанням систем управління контентом. Найпопулярніші з них – WordPress [18], Joomla [19] і Drupal [20]. Ці CMS пропонують безліч налаштувань та додаткових функцій, що дозволяють змінювати структуру системи без написання спеціального коду. Всі ці системи управління контентом входять в топ-5 за популярністю в сегменті інтернету.

WordPress – найпопулярніша система управління контентом, розібратися з якою під силу навіть новачкові. Сьогодні 29% сайтів в Інтернеті працюють на WordPress. Друга за популярністю Joomla досить потужна, щоб запустити більшість інтернет-ресурсів без особливих проблем, і не вимагає від веб-майстра високого рівня спеціальних навичок для роботи, як наступний за популярністю движок – Drupal. Це одна з найбільш потужних доступних CMS з відкритим кодом, для управління якої знадобляться певні технічні вміння.

Всі ці CMS поширюються вільно і мають відкритий код, який при наявності необхідних умінь можна редагувати.

Загальні параметри присутні у WordPress, Joomla і Drupal:

- ці двигуни створені на PHP;
- всі вони дозволяють застосовувати MySQL як вбудовану систему керування базами даних. Крім MySQL, Joomla і Drupal можуть працювати і з іншими базами;
- для кожного з движків доступна технічна підтримка – офіційні спільноти, блоги і форуми розробників на цьому движку;
- ці CMS застосовують шаблони для відображення сайтів та веб-систем, а також модулі або плагіни для збільшення функціональних можливостей;
- кожен движок має безліч додаткових інструментів, щоб додавати в систему унікальні функції;
- дозволяють користувачам змінювати візуальне відображення системи із застосуванням шаблонів.

Хоча подібностей між цими трьома CMS багато, в деяких аспектах вони відрізняються.

WordPress оснащений значно більшим числом плагінів і шаблонів, ніж будь-який інший движок. Тільки в офіційному каталозі пропонується близько 50 тисяч плагінів, використання яких дозволяє створити ресурс від простого блогу до великого порталу. Завдяки широкій популярності WordPress, її розробники створюють безкоштовні або недорогі нові розширення, які допоможуть оснастити тематичний сайт або веб-систему, SEO-оптимізований для просування в інтернеті. Доступні преміум-версії плагінів, придбання яких дозволить звертатися за підтримкою до розробників і отримати додаткові можливості для налаштування функціоналу системи (тобто базовий функціонал плагіна доступний безкоштовно, а вже додаткові функції – в платній версії).

Для систем, створених на Joomla, також розроблено безліч додаткових розширень, що дозволяють робити електронні магазини, великі портали і управляти розсилками по електронній пошті. Joomla чудово поєднує в собі міць і функціональність, які забезпечує Drupal, при цьому зберігає візуально зрозуміле управління, як у WordPress.

Кількість розширень у Joomla не така значна, як у WordPress, це іноді ускладнює підбір відповідного модуля. Крім того, стандартна установка Joomla не дозволяє виконувати пошук і установку розширень і шаблонів безпосередньо з адмінки. З цією метою потрібно поставити спеціальне розширення, але пошук і налаштування доведеться виконувати вручну.

У Drupal відома аналогічна складність з установкою шаблонів і модулів. Веб-майстер повинен знайти потрібні доповнення у сторонній системі, а потім поставити їх адресу в розділі «Модулі» або «Теми», щоб активувати. Drupal легко налаштовується за допомогою безлічі доступних модулів – на офіційному сайті їх представлено близько 37 тисяч. Припустимо безпосередньо правити файли в корені сайту, що зручно для тих користувачів, які вносять значні зміни в структуру сайту у міру його розробки.

Сайти на Drupal, як правило, характеризуються поліпшеною продуктивністю – сторінки завантажуються швидше і показують менший час відгуку, ніж ресурси, розроблені на WordPress і Joomla. Частково це пояснюється тим, що ця CMS не вимагає значних ресурсів, дорогого сервера або хостингу [21].

Для використання в системі було обрано WordPress методом аналізу ієрархій.

3.4 Вибір хостингу

Хостинг – послуга з надання ресурсів для розміщення інформації на сервері, що постійно перебуває в мережі.

Зазвичай хостинг входить в пакет з обслуговування веб-системи і має на увазі як мінімум послугу розміщення файлів системи на сервері, на якому запущене ПЗ, необхідне для обробки запитів до цих файлів (веб-сервер). Як правило, в обслуговування вже входить надання місця для поштової кореспонденції, баз даних, DNS, файлового сховища на спеціально виділеному файл-сервері і т.п., а також підтримка функціонування відповідних сервісів.

Хостинг бази даних, розміщення файлів, хостинг електронної пошти, послуги DNS можуть надаватися окремо як самостійні послуги або входити в комплексну послугу [22].

Для розміщення веб-системи були розглянуті наступні хостинги:

- 1) 000webhost [23];
- 2) Beget [24];
- 3) Timeweb [25].

В табл. 3.1 наведені характеристики тарифних планів розглядуваних хостингів.

Таблиця 3.1 – Характеристики хостингів 000webhost, Beget та Timeweb

Характеристики	000webhost	Beget	Timeweb
Дисковий простір	1000 Мб	1000 Мб	5000 Мб
Бази даних MySQL	2	1	1
Кількість систем	2	1	1
Ціна	Безкоштовно	Безкоштовно	Безкоштовний тестовий період 10 днів
Недоліки	Не працює 1 годину на добу		Платний

Виходячи з наведених характеристик, методом аналізу ієрархій було обрано хостинг Veget.

4 ПРОЕКТУВАННЯ І РОЗРОБКА БАЗИ ДАНИХ СИСТЕМИ

4.1 Побудова функціональної моделі

Завдання функціонального моделювання полягає в представленні системи у вигляді сукупності взаємопов'язаних функцій. Як методологічний інструменту функціонального моделювання розглянемо методологію IDEF0, яка включає в себе метод IDEF0, а також методи і процедури, котрі його підтримують.

У методі IDEF0 можна виділити такі складові, як концепція методу, графічна мова, процедура читання діаграми, метод побудови моделі, критерії оцінки якості і інші. В організаційну підтримку методу IDEF0 входять: процедура збору даних (інтерв'ювання), метод групової роботи, форми документування моделі, процедури погодження та затвердження моделі.

Концепція IDEF0-моделі, загальні положення.

Будь-яка система складається з взаємопов'язаних частин (елементів системи), які, працюючи спільно, виконують деяку функцію. Як елементи системи можуть виступати: люди, інформація, програмне забезпечення, обладнання, продукти або сировина. IDEF0-модель описує:

- що система робить;
- що вона виробляє;
- як управляється;
- які ресурси і кошти використовуються для виконання її функцій.

Одним з достоїнств IDEF0-моделей є те, що вони забезпечують можливість обміну інформацією про даний об'єкт мовою, зрозумілою не тільки аналітику і розробнику системи, а й фахівцю-експерту в предметній області, користувачеві, керівнику. В основі методу IDEF0 лежать такі концептуальні положення:

- графічне представлення моделі у вигляді ієрархії діаграм, що забезпечує компактність подання інформації;
- максимальна виразність, тобто здатність найкращим чином забезпечити зрозумілість моделі;
- строгість і точність представлення;
- покрокові процедури розробки моделі, її перегляду і об'єднання;
- відділення організації від функції – виключення впливу організаційної структури на функціональну модель.

IDEF0-модель складена з ієрархічного ряду діаграм, які поступово відображають рівні все більш докладних описів функцій і їх інтерфейсів в межах системи. Діаграма, що знаходиться на вершині моделі, узагальнює всю розглянуту систему. Діаграми першого рівня представляють найважливіші підсистеми з їх взаємозв'язками, а діаграми самого нижнього рівня представляють деталізовані функції, за допомогою яких, власне, і працює система.

Можна назвати три основні типи діаграм, які використовуються в IDEF0-моделях: графічні, текстові і глосарії.

Графічні діаграми – головний компонент моделі – визначають функції і функціональні відносини. Блоки представляють головні функції конкретної підсистеми поточного рівня декомпозиції. Ці функції розбиваються (декомпозиціануються) на більш детальні діаграми, поки підсистема не буде описана на рівні, що задовольняє цілі проекту.

Текст і глосарій забезпечують додаткову інформацію для графічних діаграм. При цьому всі діаграми перехресно посилаються одна на одну.

Однак до початку побудови моделі необхідно визначитися з метою моделювання, кордоном системи і точкою зору моделі.

Мета моделювання.

Створювана IDEF0-модель має конкретне призначення, зване метою моделі. Мета моделювання можна зрозуміти з наступного формального визначення моделі: M є модель системи S , якщо M може бути використана для отримання відповідей на питання щодо S з точністю A .

Таким чином, метою моделі є отримання відповідей на деяку сукупність питань. Ці питання неявно присутні (маються на увазі) в процесі аналізу і, отже, вони керують створенням моделі і направляють його. Якщо модель відповідає не на всі питання або її відповіді були не зовсім точні, то ми говорим, що модель не досягла своєї мети.

Зазвичай питання для IDEF0-моделі формулюються на самому ранньому етапі аналізу або проектування, при цьому основна суть цих питань повинна бути виражена в одній-двох фразах [26].

Межі системи.

Модельована система є частиною навколишнього середовища і завжди пов'язана з нею, тому часто важко сказати, де закінчується система і починається середовище. У зв'язку з цим в методології IDEF0 потрібно точно визначати межі системи: модель встановлює точно, що є і що не є об'єктом

моделювання, описуючи те, що входить в систему, і маючи на увазі те, що лежить за її межами.

Точка зору моделі.

З визначенням мети моделі тісно пов'язана позиція, з якої спостерігається система і створюється її модель. IDEF0 вимагає, щоб конкретна модель завжди розглядалася з однієї і тієї ж точки зору. Ця позиція називається точкою зору даної моделі.

Точку зору найкраще уявляти собі як місце (позицію) людини або об'єкта, з якої можна побачити систему в дії. Залежно від мети моделювання, можуть бути прийняті різні положення точок зору, що підкреслює різні аспекти опису об'єкта. Те, що є важливими з однієї точки зору, може навіть не з'явитися в моделі, представленій під іншим кутом зору для однієї і тієї ж системи. Для відображення в моделі інших точок зору зазвичай використовують ГЕО-діаграми.

На цьому етапі розподіляються можливості категорій користувачів, які дії і над якими даними можна робити. У проекті передбачені наступні категорії користувачів:

Гість – ознайомлення з системою, перегляд інформації.

Зареєстрований користувач – додавання коментарів, обробка даних.

Модератор – перевірка коментарів на предмет порушення правил веб-системи.

Адміністратор – адміністрування.

Категорія користувачів Гості увійшовши в систему можуть:

- отримувати інформацію про призначення системи;
- ознайомитися з тестовим прикладом обробки даних;
- переглядати новини;
- переглядати повідомлення в форумі;
- зареєструватися.

Категорія зареєстрованих користувачів, крім всіх перерахованих вище можливостей гостя, можуть:

- залишати коментарі – після новини, на форумі;
- завантажувати свої дані для обробки.

Категорія адміністратор, призначена безпосередньо для адміністрування ресурсу:

- робота з новинами – додавання, зміна, видалення новин;

- робота з інформацією-редагування вмісту головної сторінки, і кожного розділу системи;
- доступ до інформації про користувача – при необхідності, є можливість отримати особисті дані користувача (наприклад для складання статистики);
- робота з файлами (блокування, видалення), завантаженими користувачами;
- розробка додаткових функцій з урахуванням побажань користувачів.

Категорія модератор призначена для підтримки змісту коментарів та відгуків в межах, обговорених правилами ресурсу.

Контекстна діаграма верхнього рівня.

Кожна модель повинна мати контекстну діаграму верхнього рівня, на якій об'єкт опису представлений поодиноким блоком, оточеним дугами. Дуги на цій діаграмі пов'язують об'єкт опису з навколишнім середовищем (зовнішніми об'єктами). Контекстна діаграма встановлює область або кордони моделювання, і так як контекстна діаграма представляє весь об'єкт, то діаграма має мати саме загальне ім'я. Те ж відноситься і до імен інтерфейсних дуг – вони також представляють закінчений набір зовнішніх інтерфейсів.

На контекстній діаграмі також повинні бути вказані точка зору моделі і мета її створення, які допомагають направляти створення моделі. Точка зору визначає те, що може бути «видно» в межах контекстної моделі і з якої перспектива або «кута». Формулювання мети висловлює причину створення моделі і фактично визначає її структуру. Найбільш важливі особливості системи представлені в ієрархії, оскільки функція верхнього рівня декомпозиціануються на підфункції.

Поділ об'єкта на його структурні частини (блоки і дуги, складові діаграми) називається декомпозицією.

Підфункції, в свою чергу, декомпозиціануються далі, поки не будуть показані всі деталі, важливі з цієї точки зору. Кожна підфункція зображується окремим блоком, який є дочірньою деталізацією батьківського блоку на наступному (нижчому) рівні. Области опису дочірніми діаграмами не повинні виходити за межі опису контекстної діаграми верхнього рівня.

На рис. 4.1 представлена контекстна діаграма верхнього рівня проєктованої системи.

Головною функцією системи є управління інформаційною системою обробки і аналізу гідрометеорологічних даних. На вході в систему зображено

п'ять стрілок: відвідувач, дані користувача, файли, коментарі, контент. В якості керуючих стрілок зображені: методи обробки даних, бюджет, правила ресурсу. До механізмів відносяться стрілки: адміністратор, модератор. На виході з системи маємо: оброблені дані у вигляді діаграми, файли для завантаження, оновлений контент, новий функціонал.

Створення А0-діаграми.

Всі системні функції лежать в межах одиночного блоку, показаного на діаграмі А-0. Діаграма А0 декомпозиційною функцією на діаграмі А0 на тринадцять головних підфункцій.

Реальна «вершина» моделі – діаграма А0. Її структура ясно показує те, що діаграма А-0 тільки намітила показати. Зміст і структура А0 також обмежують кожний наступний рівень, тому що це закінчений опис вибраного об'єкту. Нижні рівні тільки розкривають (але не доповнюють) функції А0. Діаграма А0 змушує автора підтримувати обраний рівень абстракції, тримати однакову глибину моделювання і відносити подробиці до більш низького рівня.

На рис. 4.2 зображена А0-діаграма нашої системи.

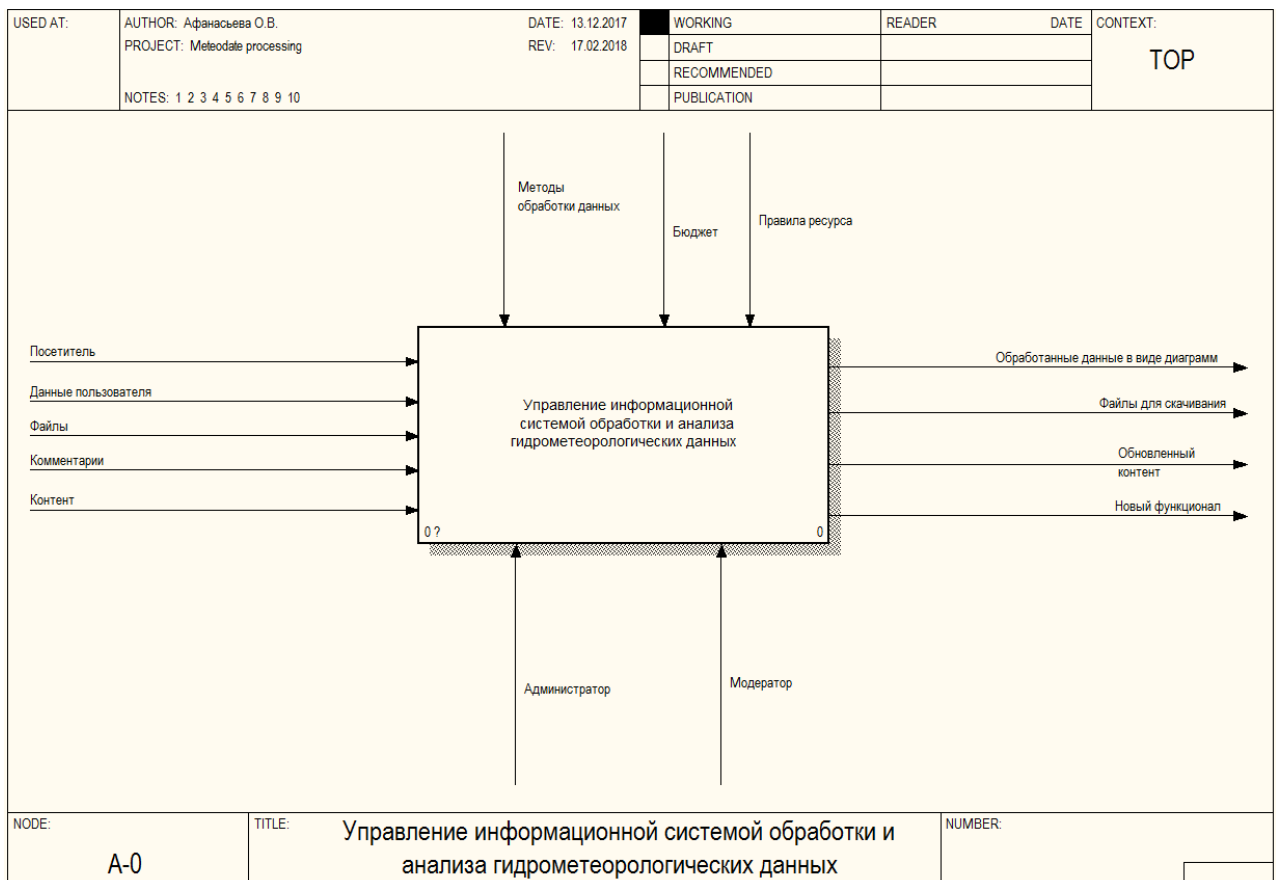


Рисунок 4.1 – Контекстна діаграма верхнього рівня

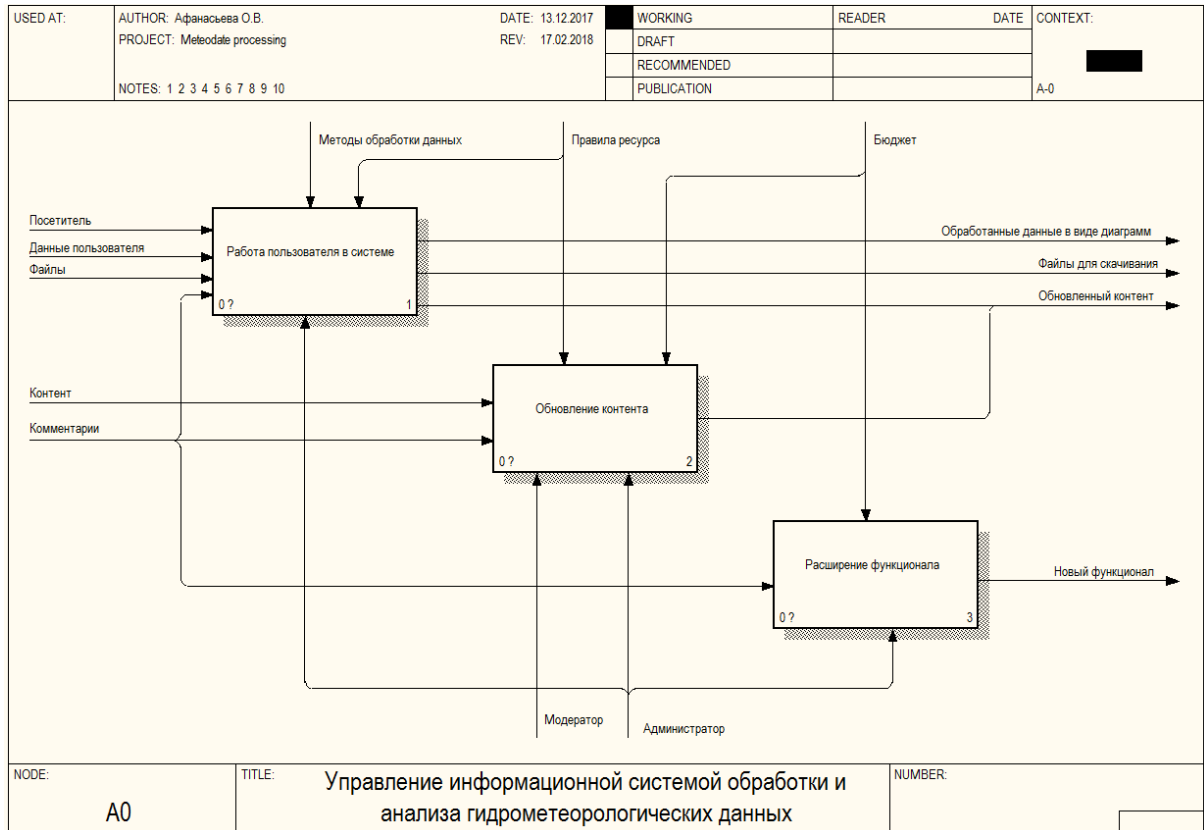


Рисунок 4.2 – A0-диаграмма

Створення дочірніх діаграм.

Формується нова діаграма для кожного блоку, який охоплює ту ж саму тему, що і її батьківський блок, але більш докладно. Створюється спочатку чорнова діаграма шляхом запису всіх даних/об'єктів, пов'язаних з аналізованою функцією. Необхідно, щоб цей список охоплював всю тему батьківського блоку без втрати будь-якої частини при декомпозиції. Потім малюються блоки, які є кандидатами в якості підфункцій з відповідними даними та об'єктами зі списку, і показуються дуги між блоками.

При створенні будь-якої IDEF0-діаграми, повинні бути задоволені наступні вимоги:

- мета діаграми і точка зору повинна відповідати заявленій меті та точці зору повної моделі;
- граничні дуги повинні відповідати дугам батьківського блоку;
- зміст блоку має точно відповідати змісту батьківського блоку.

На рис. 4.3-4.5 зображені дочірні діаграми нашої системи.

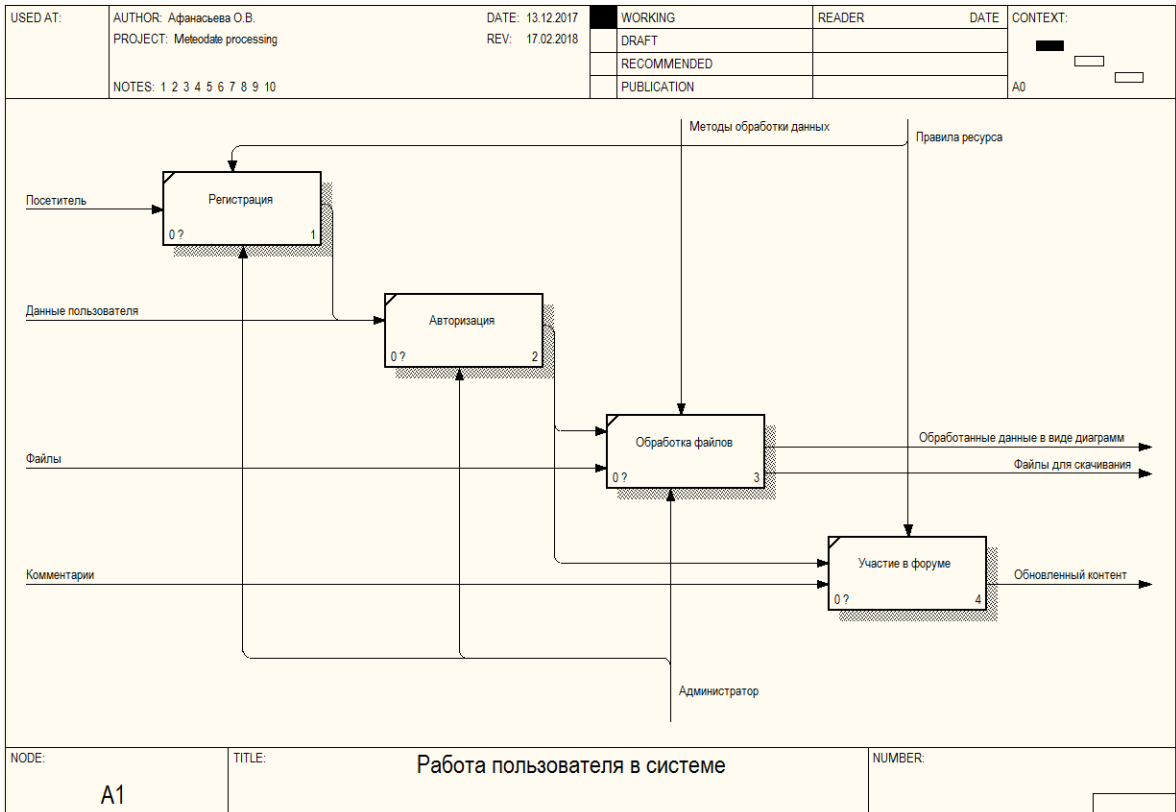


Рисунок 4.3 – Дочірня діаграма блоку «Робота користувача в системі»

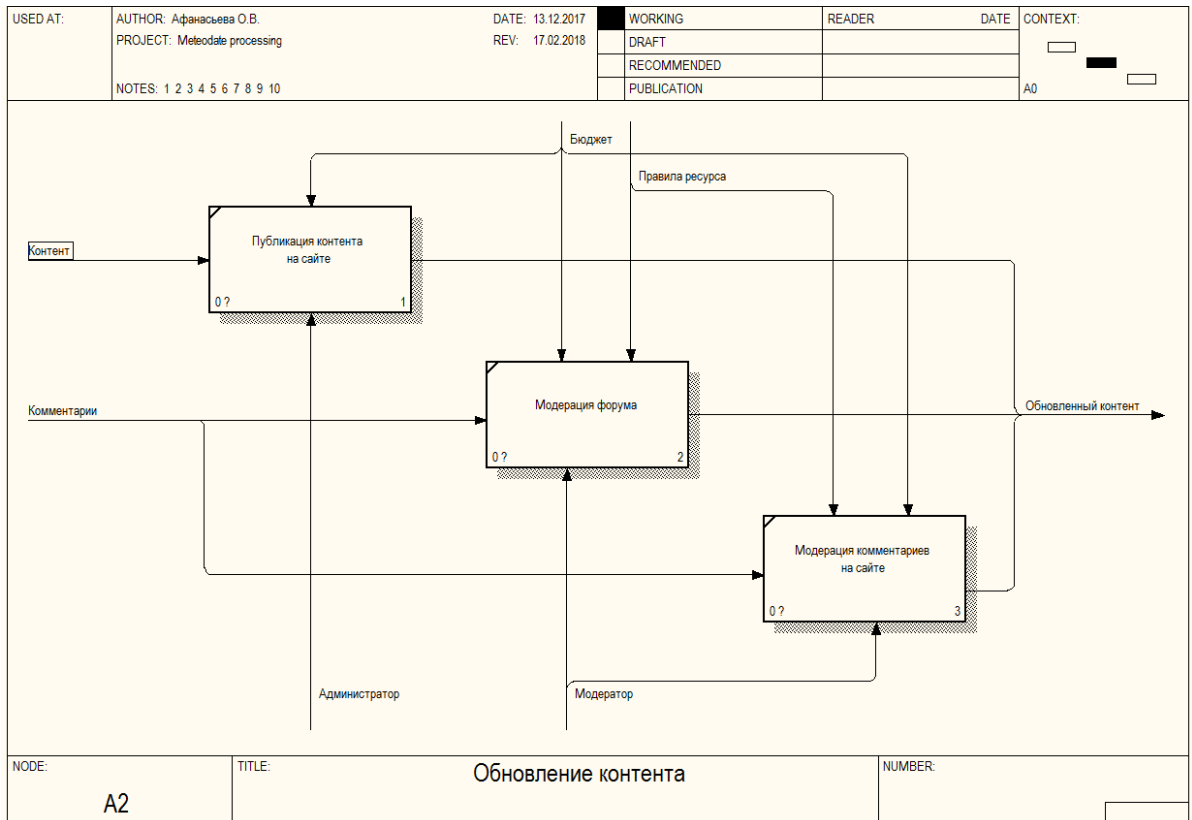


Рисунок 4.4 – Дочірня діаграма блоку «Оновлення контенту»

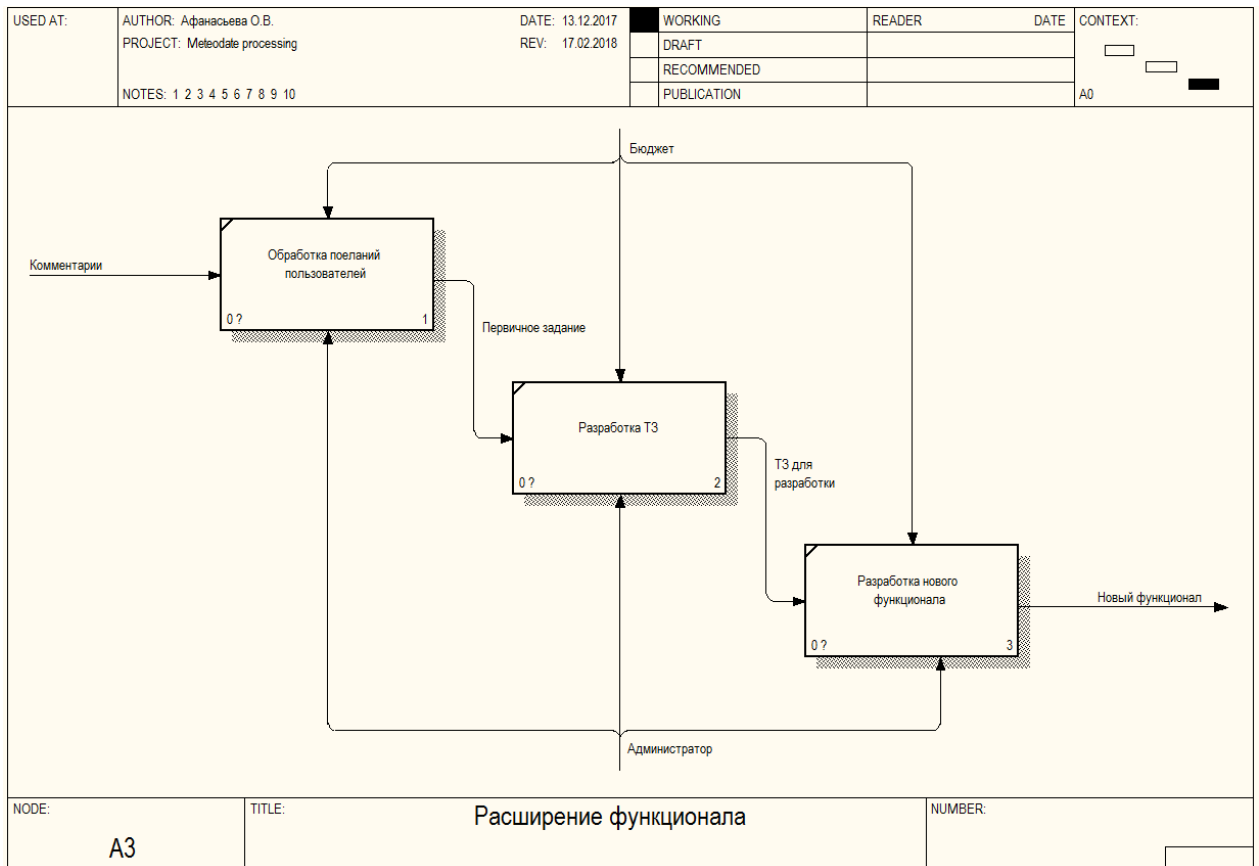


Рисунок 4.5 – Дочірна діаграма блоку «Розширення функціоналу»

4.2 Побудова концептуальної моделі

Початковою стадією проектування системи баз даних є побудова семантичної моделі предметної області, яка базується на аналізі властивостей і природи об'єктів предметної області та інформаційних потреб майбутніх користувачів системи, що розробляється. Цю стадію прийнято називати концептуальним проектуванням системи, а її результат – концептуальною моделлю предметної області (об'єктом моделювання тут є предметна область майбутньої системи).

Такі моделі узагальнено представляють інформаційні потреби користувачів створюваної системи в частині використання збережених даних і по суті є засобом комунікації як розробників, так і користувачів на різних стадіях життєвого циклу бази даних. Призначення концептуальних моделей визначає і деякі специфічні вимоги до засобів їх подання. Крім зв'язку між об'єктом і його властивостями, концептуальна модель відображає зв'язки між об'єктами різних класів. Кількісний характер участі сутностей (один або багато) задається типом зв'язку (або потужністю зв'язку). Можливі наступні

типи: «один до одного» (1: 1), «один до багатьох» (1: M), «багато до одного» (M: 1), «багато до багатьох» (M: M).

Створення зв'язків між таблицями – останній етап проектування системи таблиць. На цьому етапі фактично реєструються зв'язку між первинними і зовнішніми ключами, заплановані при конструюванні таблиць. Після цього можна приступати до розробки інтерфейсу майбутнього програми.

Існує дві причини для створення зв'язків між таблицями: підтримка посилальної цілісності і завдання способу вибірки даних з декількох таблиць. Зв'язок задає відношення між полями таблиць, що мають однакові за змістом значення, наприклад, між первинним ключем однієї таблиці і зовнішнім ключем іншої таблиці. Зв'язки можна задати на рівні бази даних (в цьому випадку можлива підтримка посилальної цілісності даних) і на рівні запитів. Зв'язки на рівні бази даних є постійними зв'язками і є актуальними при будь-яких діях, які виконуються з таблицями. Зв'язки на рівні запитів є тимчасовими і актуальні тільки на момент виконання запиту, в якому вони задані.

Один-до-багатьох (one-to-many). Є найбільш часто вживаним видом зв'язку. У цьому випадку кожного запису таблиці А може відповідати багато записів таблиці Б (або жодної). У свою чергу, кожного запису таблиці Б відповідає в точності один запис таблиці А. Таблиця А в такого зв'язку називається головною, а таблиця Б – пов'язаної або підлеглої.

Багато-до-багатьох (many-to-many). Багатьом із записами з таблиці А може відповідати багато записів з таблиці Б (і навпаки). По суті, зв'язок типу багато-до-багатьох (many-to-many) являє собою дві зв'язку типу один до багатьох (one-to-many). При цьому таблиці А і Б розташовані з боку один (one), а допоміжна таблиця – з боку багато (many). Такий тип зв'язку використовується рідше, але існують ситуації, коли без неї не обійтися.

Один-до-одного (one-to-one). Однією записи таблиці А відповідає в точності один запис таблиці Б і навпаки. Цей тип зв'язку практично ніколи не застосовується. Єдиний випадок, коли застосування цього типу зв'язку виправдано – розбивка таблиці, яка містить дуже велику кількість полів, на кілька частин.

У проєкті використовується зв'язок «один до багатьох» (1: M) це такий тип зв'язку між сутностями А і В, коли одному екземпляру сутності А може відповідати нуль, один або кілька екземплярів сутності В, проте кожному екземпляру сутності В відповідає тільки один екземпляр сутності А.

Приклад зв'язку «один до багатьох» – зв'язок між сутностями Країна і Місто. Країна може вміщати декілька міст, але кожне місто обов'язково має країну, і тільки одну. Зв'язки «один до багатьох» найбільш широко поширені.

Важливим процесом є нормалізація бази даних. Головна мета нормалізації бази даних – усунення надмірності та дублювання інформації. В ідеалі при нормалізації треба домогтися, щоб будь-яке значення зберігалось в базі в одному екземплярі, причому значення це не має бути отримано розрахунковим шляхом з інших даних, що зберігаються в базі.

База даних вважається нормалізованою, якщо її таблиці (по крайній мере, більшість таблиць) представлені як мінімум в третій нормальній формі.

Перша нормальна форма: забороняє повторювані стовпці (що містять однакову за змістом інформацію) забороняє множинні стовпці (що містять значення типу списку і т.п.) вимагає визначити первинний ключ для таблиці, тобто той стовпець або комбінацію стовпців, які однозначно визначають кожен рядок.

Друга нормальна форма: вимагає, щоб не ключові стовпці таблиць залежали від первинного ключа в цілому, але не від його частини. Маленька ремарочка: якщо таблиця знаходиться в першій нормальній формі і первинний ключ у неї складається з одного стовпця, то вона автоматично знаходиться і в другій нормальній формі.

Третя нормальна форма вимагає щоб не ключові стовпці в ній не залежали від інших не ключовим стовпців, а залежали лише від первинного ключа. Найпоширеніша ситуація в даному контексті – це розрахункові стовпці, значення яких можна отримати шляхом будь-яких маніпуляцій з іншими стовпцями таблиці. Для приведення таблиці в третю нормальну форму такі стовпці з таблиць треба видалити.

Мета проведення нормалізації бази даних – це усунення (або, принаймні, серйозне скорочення) надмірності, дублювання даних. Як наслідок, значно скорочується вірогідність появи суперечливих даних, полегшується адміністрування бази і оновлення інформації в ній, скорочується обсяг дискового простору.

Проаналізувавши вимоги до інформаційної системи спроектували базу даних, в якій буде зберігатися весь контент розміщений в системі.

WordPress використовує кілька взаємопов'язаних таблиць. Між ними встановлені зв'язку один до багатьох. Наприклад, до однієї сторінки може бути багато коментарів.

Стандартний префікс для WordPress – wp_. Можна поставити й інше префікс, але це потрібно робити при створенні нової БД: в старій базі даних зробити це не вийде – система перестане функціонувати. Префікс можна задати абсолютно будь-який, і робиться це для поліпшення захисту від злому.

Отже, моя база даних складається з 14 таблиць:

- wp_commentmeta;
- wp_comments;
- wp_links;
- wp_options;
- wp_postmeta;
- wp_posts;
- wp_termmeta;
- wp_terms;
- wp_term_relationships;
- wp_term_taxonomy;
- wp_usermeta;
- wp_users;
- wp_file;
- wp_userfiles.

Нижче, в табл. 4.1 наведений короткий опис кожної таблиці. В табл. 4.2-4.15 наведена структура кожної таблиці БД.

Таблиця 4.1 – Опис таблиць

Таблиця	Дані
1	2
wp_posts	Записи, сторінки, вкладення, редакції, призначені для користувача записи
wp_postmeta	Метадані записів, сторінок і т.д.
wp_comments	Коментарі
wp_commentmeta	Метадані коментарів
wp_term_relationships	Зв'язки між таксономії і записами, сторінками і т.д.
wp_term_taxonomy	Таксономії (включаючи категорії і мітки)
wp_terms	Категорії, мітки і терміни призначених для користувача таксономій
wp_termmeta	Метадані для кожної таксономії

Продовження таблиці 4.1

1	2
wp_links	Посилання в блоці (як правило, зараз не використовуються)
wp_users	Користувачі
wp_user_meta	Метадані для кожного користувача
wp_options	Опції і налаштування системи (встановлюються в адмінки на сторінці налаштувань і в темах/плагінах)
wp_file	Загальна інформація про файл
wp_userfiles	Інформація про приналежність файла певному користувачеві
wp_file	Інформація про файл
wp_userfiles	Інформація про приналежність файлу певному користувачеві

Таблиця 4.2 – Таблиця wp_commentmeta бази даних

Назва	Тип
meta_id	bigint(20)
comment_id	bigint(20)
meta_key	varchar(255)
meta_value	longtext

Таблиця 4.3 – Таблиця wp_comments бази даних

Назва	Тип
1	2
comment_ID	bigint(20)
comment_post_ID	bigint(20)
comment_author	tinytext
comment_author_email	varchar(100)
comment_author_url	varchar(200)
comment_author_IP	varchar(100)
comment_date	datetime
comment_date_gmt	datetime
comment_content	text

Продовження таблиці 4.3

1	2
comment_karma	int(11)
comment_approved	varchar(20)
comment_agent	varchar(255)
comment_type	varchar(20)
comment_parent	bigint(20)
user_id	bigint(20)

Таблиця 4.4 – Таблиця wp_links бази даних

Назва	Тип
link_id	bigint(20)
link_url	varchar(255)
link_name	varchar(255)
link_image	varchar(255)
link_target	varchar(25)
link_description	varchar(255)
link_visible	varchar(20)
link_owner	bigint(20)
link_rating	int(11)
link_updated	datetime
link_rel	varchar(255)
link_notes	mediumtext
link_rss	varchar(255)

Таблиця 4.5 – Таблиця wp_options бази даних

Назва	Тип
option_id	bigint(20)
option_name	varchar(191)
option_value	longtext
autoload	varchar(20)

Таблиця 4.6 – Таблиця wp_postmeta бази даних

Назва	Тип
meta_id	bigint(20)
post_id	bigint(20)
meta_key	varchar(255)
meta_value	longtext

Таблиця 4.7 – Таблиця wp_posts бази даних

Назва	Тип
ID	bigint(20)
post_author	bigint(20)
post_date	datetime
post_date_gmt	datetime
post_content	longtext
post_title	text
post_excerpt	text
post_status	varchar(20)
comment_status	varchar(20)
ping_status	varchar(20)
post_password	varchar(255)
post_name	varchar(200)
to_ping	text
pinged	text
post_modified	datetime
post_modified_gmt	datetime
post_content_filtered	longtext
post_parent	bigint(20)
guid	varchar(255)
menu_order	int(11)
post_type	varchar(20)
post_mime_type	varchar(100)
comment_count	bigint(20)

Таблица 4.8 – Таблица wp_termmeta базы данных

Назва	Тип
meta_id	bigint(20)
term_id	bigint(20)
meta_key	varchar(255)
meta_value	longtext

Таблица 4.9 – Таблица wp_terms базы данных

Назва	Тип
term_id	bigint(20)
name	varchar(200)
slug	varchar(200)
term_group	bigint(10)

Таблица 4.10 – Таблица wp_term_relationships базы данных

Назва	Тип
object_id	bigint(20)
term_taxonomy_id	bigint(20)
term_order	int(11)

Таблица 4.11 – Таблица wp_user_meta базы данных

Назва	Тип
umeta_id	bigint(20)
user_id	bigint(20)
meta_key	varchar(255)
meta_value	longtext

Таблица 4.12 – Таблица wp_userfiles базы данных

Назва	Тип
file_id	bigint(20)
user_id	bigint(20)

Таблиця 4.13 – Таблиця wp_term_taxonomy бази даних

Назва	Тип
term_taxonomy_id	bigint(20)
term_id	bigint(20)
taxonomy	varchar(32)
description	longtext
parent	bigint(20)
count	bigint(20)

Таблиця 4.14 – Таблиця wp_users бази даних

Назва	Тип
ID	bigint(20)
user_login	varchar(60)
user_pass	varchar(255)
user_nicename	varchar(50)
user_email	varchar(100)
user_url	varchar(100)
user_registered	datetime
user_activation_key	varchar(255)
user_status	int(11)
display_name	varchar(250)

Таблиця 4.15 – Таблиця wp_file бази даних

Назва	Тип
file_id	bigint(20)
file_name	varchar(60)
file_path	varchar(255)
file_size	int(11)

Фізична схема спроектованої бази даних відображена на рис. 4.6.

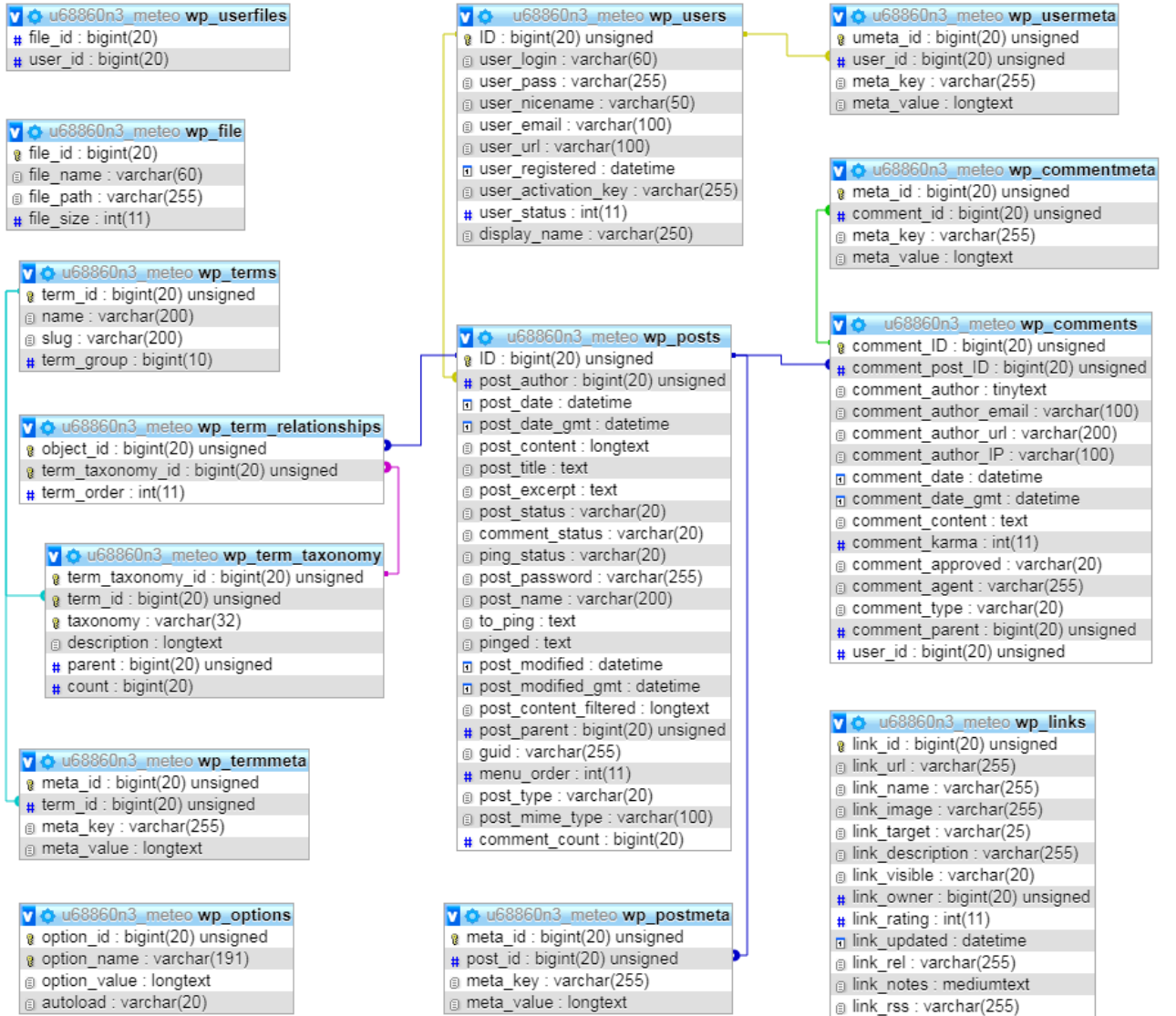


Рисунок 4.6 – Фізична схема БД

5 ПРОЕКТУВАННЯ І РОЗРОБКА БАЗИ МОДЕЛЕЙ СИСТЕМИ

5.1 Загальні відомості про бази моделей і системи управління базами моделей

Дані та моделі є центральними елементами систем підтримки прийняття рішень (СППР). Фактичне СППР відрізняється від автоматизованих інформаційних систем (АІС) наявністю інтерактивних програм (з їх допомогою користувач може досліджувати і «мандрувати» по базах даних різних форм, розмірів і типів) та бази моделей (усередині її користувач може конструювати, аналізувати, інтерпретувати одну чи кілька моделей).

Протягом 60-х років управлінські системи розроблялися на базі процедурних елементів, причому управління моделями полягало в керуванні бібліотеками процедур розв'язків, поданих у формі програм і підпрограм. На початку 70-х років набула стабільності концепція баз даних, на основі якої і створювалися інформаційні системи. Проте тепер визнаною є думка, що саме моделі визначають відношення між даними і суто базовий підхід до проектування систем призводить до нехтування зв'язків, які відповідають процесам і процедурам у середовищі розв'язуваних задач. Тому спостерігається міграція від підходу даних до підходу моделей, що стають джерелом тверджень і засновків як підґрунтя для усвідомлення суті інформаційних відношень. Концепція управління моделями усвідомлюється дедалі ширшим загальною дослідників і спеціалістів як передній край у галузі інформаційних систем і систем підтримки прийняття рішень.

Система управління моделями є одним із компонентів універсальної СППР. Функціями цього компонента є класифікація, організація і доступ до моделей, тобто ці функції аналогічні функції системи управління базами даних.

База моделей СППР містить оптимізаційні і не оптимізаційні моделі. До складу оптимізаційних моделей належать моделі математичного програмування – лінійного (розподіл ресурсів, оптимальне планування, аналіз сіткових графіків, транспортна задача), нелінійного, динамічного; моделі обліку; моделі аналізу цінних паперів для визначення інвестиційної стратегії; моделі маркетингу і та ін.

До не оптимізаційних моделей належать статистичні моделі (лінійний і нелінійний аналіз регресій); методи прогнозування (аналізу) часового ряду; альтернативні методи моделювання (наприклад, машинна імітація) тощо.

У більшості відомих з літературних джерел систем для маніпулювання і зберігання моделей використовувались поняття і методи подання знань (формальна логіка, моделі продукції, семантичні сітки, фрейми та їхні гібриди), а також реляційний базис, аналогічні реляційні моделі даних.

Системи управління базами моделей (СУБМ), як узагальнені програмні засоби, забезпечують користувачам широкий набір моделей і дають змогу проводити гнучкий доступ, оновлення та зміну бази моделей.

Основні функції СУБМ такі:

- створення нових моделей;
- каталогізація і оцінювання широкого діапазону моделей;
- зв'язування компонентів моделей у базі моделей;
- інтеграція складових елементів моделей;
- виконання набору загальних функцій управління СУБМ.

Програмне забезпечення для СУБМ розроблене значно менше, ніж для СУБД чи користувацького інтерфейсу; наявним системам СУБМ притаманне розмаїття, а комерційні пакети СППР нерідко містять основні комбінації аналітичних методів розв'язування, статистичних пакетів та інших засобів моделювання. Повний комплект усіх сімей і підсімей методів моделювання зустрічається рідко, а частіше вмонтовані в систему процедури і засоби користувацького інтерфейсу; СУБМ для систематизованого формулювання, аналізу та інтерпретації моделей часто бувають спрощеними і обмеженими за своєю природою. Перспективним напрямком створення ефективних СУБМ є структурне моделювання. Розроблено також кілька мов програмування досить високого рівня, спеціально пристосованих для створення елементів СУБМ для СППР [27].

5.2 Deductor Studio Academic

Мета роботи Deductor – формалізувати процес прийняття рішень і поставити його «на потік».

Ключовою особою в даному процесі є аналітик. Для нього необхідний інструмент, що дозволяє формалізувати і гнучко перебудовувати логіку прийняття рішень. Таким інструментом є Deductor Studio – робоче місце аналітика.

Робота аналітика зі Studio зводиться до візуальної побудови сценаріїв. Сценарій – послідовність дій, що дозволяють отримати з даних знання. Вся робота виконується за допомогою майстрів і зводиться до комбінування всього 5 операцій:

- 1) Підключення коннектора. Для взаємодії зі сторонніми системами необхідно налаштувати параметри доступу: місце розташування, користувач, пароль та інше. Через одного разу налаштоване підключення здійснюється вивантаження і завантаження даних в зовнішню систему. Deductor підтримує десятки систем: СУБД, сховища даних, облікові системи, веб-сервіси, офісні програми, файли і т.д.
- 2) Імпорт даних. Аналіз в Deductor починається з отримання набору даних. За допомогою майстра вибираються необхідні таблиці, об'єкти, файли і запускається процес імпорту.
- 3) Обробка. Під обробкою мається на увазі будь-яке перетворення даних. Підтримуються десятки методів обробки від розрахунку за формулами до самонавчальних алгоритмів: очищення даних, трансформація, Data Mining. Механізми обробки можна комбінувати довільним чином, реалізуючи як завгодно складну логіку аналізу.
- 4) Візуалізація. Переглянути дані в Deductor Studio можна на будь-якому етапі обробки. Програма самостійно аналізує, яким чином можна відобразити інформацію, користувач повинен лише вибрати потрібний варіант. Deductor включає безліч зручних інтерактивних візуалізаторів: OLAP, таблиці, графіки, дерева, карти...
- 5) Експорт результатів. Завершальним кроком у сценарії обробки найчастіше є експорт даних. Результати вивантажуються для подальшого використання в інших програмах, наприклад, прогноз продажів передається в систему для формування замовлення на поставку або у корпоративній веб-системі публікуються розраховані KPI.

Deductor Studio дозволяє аналітику автоматизувати рутинні операції по обробці даних і зосередитися на інтелектуальній роботі: формалізація логіки прийняття рішень, побудова моделей, прогнозування. Інші співробітники компанії можуть легко скористатися готовими результатами, не вникаючи в складності аналізу:

Аналітична звітність. Аналітик перетягує мишкою на спеціальну панель необхідні звіти. Кінцевий користувач за допомогою Deductor Viewer просто вибирає цікавий звіт зі списку і отримує результат. Ніяких додаткових дій робити не потрібно. Вся складна аналітична обробка виконується автоматично.

Інтеграція в бізнес-процес. Аналітик експортує результати в сторонню систему: веб-систему, ERP, CRM і т.п., а кінцевий користувач побачить у звичній йому програмі підсумок складної аналітичної обробки. Обмін даними може здійснюватися в режимі online або по регламенту. Для вбудовування в бізнес-процес необхідно скористатися Analytic або Integration Server [28].

Через те, що ПО Deductor є платним, для ознайомлення було використано безкоштовну версію. Академічна версія Deductor Academic призначена для некомерційного використання, в ній присутній тільки інтерактивний режим побудови сценаріїв і імпорт з двох джерел даних [29].

Обробка і візуалізація – дві основні операції з даними в Deductor. Під обробкою маються на увазі будь-які маніпуляції з набором даних: від найпростіших (наприклад, сортування) до складних (побудова нейронної мережі). Обробник можна уявити у вигляді «чорної скриньки», на вхід якого подається набір даних, а на виході формується перетворений набір даних

Реалізовані в Deductor обробники забезпечують основну потребу в аналізі даних і створенні закінчених аналітичних рішень на базі Data Mining. Їх класифікація наведена на рис. 5.1.

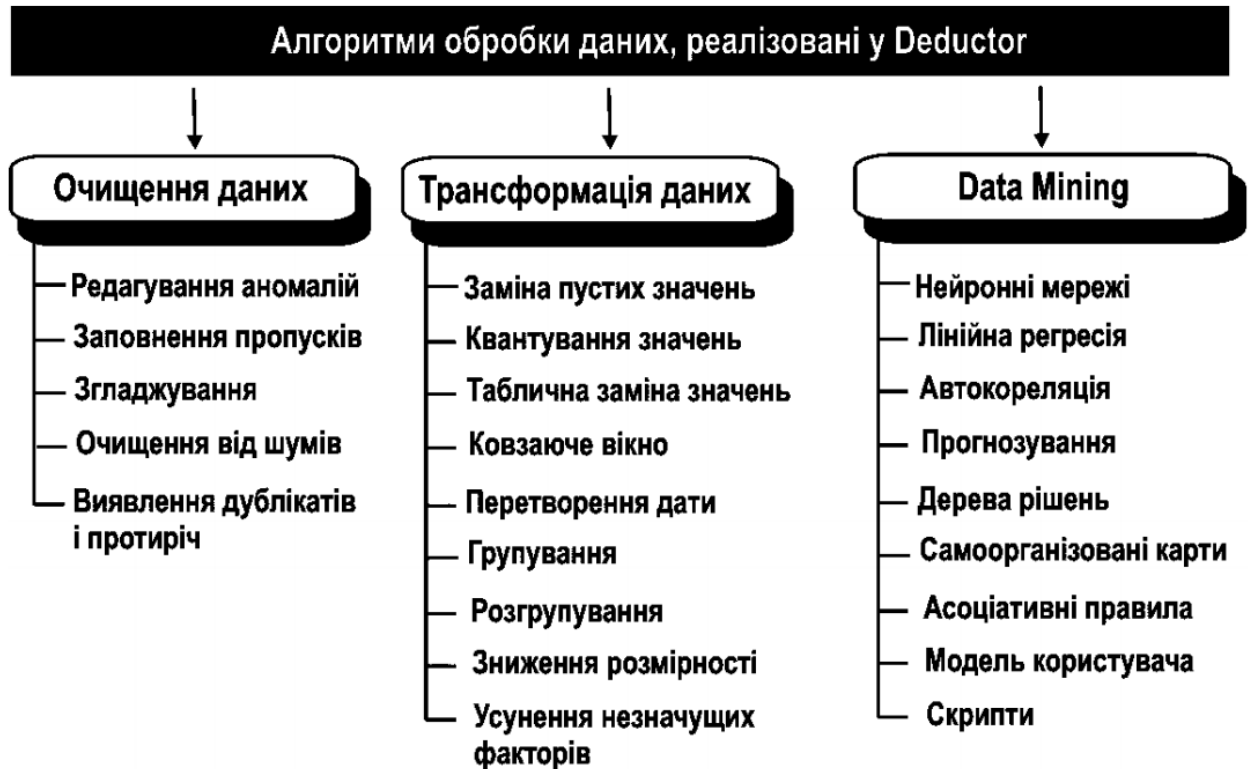


Рисунок 5.1 – Схема класифікації алгоритмів (обробників) у Deductor

5.2.1 Обробка і візуалізація гідрометеорологічних даних

По-перше необхідно створити нове сховище. Далі необхідно завантажити в сховище даних значення температури, тиску, відносної вологості та швидкості вітру у форматі .txt. Дані були взяті за період з 01.01.2007 по 31.12.2016 з джерела [30] у форматі .xls і конвертовані у .txt. На рис. 5.2 наведено фрагмент таблиці, завантаженої в Deductor.

Местное время в Одессе	T	Po	U	Ff
01.01.2007 2:00:00	2	766,1	78	3
01.01.2007 5:00:00	3	765,7	72	2
01.01.2007 8:00:00	2,2	765,9	75	3
01.01.2007 11:00:00	8,8	766	48	2
01.01.2007 14:00:00	10,6	763,6	47	2
01.01.2007 17:00:00	6,8	762,7	73	3
01.01.2007 20:00:00	7,6	761	66	4
01.01.2007 23:00:00	7,1	759,2	71	4
02.01.2007 2:00:00	6,8	757,2	71	4
02.01.2007 5:00:00	6,4	755,7	72	2
02.01.2007 8:00:00	5	755,6	81	3
02.01.2007 11:00:00	8,1	755,8	70	1

Рисунок 5.2 – Фрагмент первинних даних

При завантаженні файлу також можна вибрати способи відображення даних. На рис. 5.3 і 5.4 приведена статистика, котра була автоматично сформована програмою і гістограма розподілу значень температури.

Метка столбца	Статистика: Кол-во значений = 29415								
	Гистогра...	Мини...	Макс...	Сред...	Стан...	Σ Сумма	Σ ² Сумм...	s Кол-в...	σ Кол-в...
1 Местное время ...		007 2:00:00	16 23:00:00	12 17:59:27	дн. 13:28:08				0
2 9.0 T		-20,3	38,4	1,73668369	,671407605	344706,4	6792772,44		45
3 9.0 Po		731,2	779,6	58,5536321	,811504765	22273410,3	656794E10		52
4 9.0 U		15	100	5,69272256	7,61445578	2217494	176937490		119
5 9.0 Ff		0	18	,880480664	,013159833	84617	362789		39

Рисунок 5.3 – Статистика даних

9.0 T	Диапазон	Частота
	-20,3 : -16,38666667	17 (0,1%)
	-16,38666667 : -12,47333333	89 (0,3%)
	-12,47333333 : -8,56	224 (0,8%)
	-8,56 : -4,64666667	781 (2,7%)
	-4,64666667 : -0,7333333333	1684 (5,7%)
	-0,7333333333 : 3,18	3466 (11,8%)
	3,18 : 7,093333333	4376 (14,9%)
	7,093333333 : 11,00666667	3726 (12,7%)
	11,00666667 : 14,92	3266 (11,1%)
	14,92 : 18,83333333	3426 (11,6%)
	18,83333333 : 22,74666667	3822 (13,0%)
	22,74666667 : 26,66	3003 (10,2%)
	26,66 : 30,57333333	1226 (4,2%)
	30,57333333 : 34,48666667	233 (0,8%)
	34,48666667 : 38,4	31 (0,1%)

Рисунок 5.4 – Гістограма значень температури

Після завантаження даних, їх необхідно обробити для того, щоб ці дані можливо було в подальшому використовувати для прогнозу погоди.

Спочатку засобами Deductor виконуємо аналіз якості даних (рис.5.5). Бачимо, що в стовпці температура відсутні 45 значень (ця інформація також була наведена в статистиці в останньому стовпчику на рис. 5.3). Зараз ми можемо обрати спосіб, котрим будуть заповнюватися/розраховуватися відсутні значення. В нашому випадку було обрано значення медіани для заповнення.

№	Столбец	Тип данных	Вид данных	Пропуски		Выбросы		Экстремальные		Колво уникальных	Качество данных	Резюме
				Колво	Действие	Колво	Действие	Колво	Действие			
1	Местное время в Одессе	Дата/Время	Непрерывный								0,9997	Пригоден
2	T	9.0 Вещественный	Непрерывный	45	Заменять медианой	12	Ограничивать				0,8482	Предобработка
3	Po	9.0 Вещественный	Непрерывный	52	Заменять медианой	115	Ограничивать				0,7493	Предобработка
4	U	9.0 Вещественный	Непрерывный	119	Заменять медианой	14	Ограничивать				0,8781	Предобработка
5	Ff	9.0 Вещественный	Непрерывный	39	Заменять медианой	554	Ограничивать	87	Ограничивать		0,6549	Предобработка

Рисунок 5.5 – Аналіз якості даних

Для того, щоб в подальшому була можлива побудова діаграм, необхідно перетворити стовпець з датою, а саме розділити його на 3 стовпця рік, місяць, дата (рис 5.6).

Местное время в Одессе	Местное время в Одессе (Год)	Местное время в Одессе (Месяц)	Местное время в Одессе (День месяца)	T	Po	U	Ff
01.01.2007 2:00:00	2007	1	1	2	766,1	78	3
01.01.2007 5:00:00	2007	1	1	3	765,7	72	2
01.01.2007 8:00:00	2007	1	1	2,2	765,9	75	3
01.01.2007 11:00:00	2007	1	1	8,8	766	48	2
01.01.2007 14:00:00	2007	1	1	10,6	763,6	47	2
01.01.2007 17:00:00	2007	1	1	6,8	762,7	73	3
01.01.2007 20:00:00	2007	1	1	7,6	761	66	4
01.01.2007 23:00:00	2007	1	1	7,1	759,2	71	4
02.01.2007 2:00:00	2007	1	2	6,8	757,2	71	4
02.01.2007 5:00:00	2007	1	2	6,4	755,7	72	2
02.01.2007 8:00:00	2007	1	2	5	755,6	81	3
02.01.2007 11:00:00	2007	1	2	8,1	755,8	70	1

Рисунок 5.6 – Фрагмент перетворених даних

Після перетворення дати можливо побудувати OLAP-куб для подальшого аналізу даних. Для відображення в OLAP-кубі були обрані середнє, мінімальне і максимальне значення температури розраховані по місяцях і роках (рис. 5.7).

		Местное время в Одессе (Месяц)																							
		1			2			3			4			5			6			7					
		Т			Т			Т			Т			Т			Т			Т					
Местное время в Одессе (Год)	Т	Ср	Мин	Мак	Т	Ср	Мин	Мак	Т	Ср	Мин	Мак	Т	Ср	Мин	Мак	Т	Ср	Мин	Мак	Т	Ср	Мин	Мак	
2007	5,03	-8,20	12,00	1,29	-12,70	11,20	6,06	0,20	13,60	10,29	2,20	17,80	18,03	5,20	32,30	23,09	17,00	31,10	24,59	11,40	38,40				
2008	-0,50	-12,30	11,40	1,81	-8,10	15,50	6,95	0,90	13,80	10,19	5,80	23,10	15,13	5,20	22,60	21,32	11,30	31,70	22,51	15,50	29,80				
2009	0,23	-9,60	11,40	1,84	-4,30	11,40	4,27	-4,40	11,90	11,06	1,50	21,10	16,32	8,30	25,80	21,42	12,70	31,40	24,30	11,30	36,20				
2010	-2,71	-20,30	10,20	-0,39	-8,10	8,50	3,88	-5,10	19,80	10,29	4,20	17,60	16,67	10,10	24,20	21,96	14,60	30,20	24,48	11,40	32,20				
2011	-0,21	-10,20	8,20	-2,19	-11,00	11,60	3,29	-8,90	17,30	9,76	0,70	20,80	16,29	6,00	26,00	20,74	11,40	29,80	23,33	11,40	33,40				
2012	-0,93	-14,30	7,60	-5,31	-18,00	9,40	4,73	-5,60	20,30	10,94	0,80	27,80	19,41	10,60	29,20	22,35	13,30	33,10	25,60	11,40	34,10				
2013	0,36	-12,30	11,10	2,69	-2,40	9,90	3,15	-7,80	12,10	11,21	3,20	25,60	19,08	13,10	24,90	21,88	11,40	30,40	23,05	15,00	31,70				
2014	0,27	-16,20	11,40	0,49	-11,40	10,40	7,29	1,00	18,60	11,13	1,00	20,20	17,22	5,70	28,40	20,75	12,00	30,90	24,22	16,50	32,00				
2015	0,42	-14,50	7,40	0,96	-9,00	9,60	5,24	-0,60	11,80	9,61	1,60	21,50	16,70	9,20	26,30	21,12	14,10	28,60	23,31	13,90	32,30				
2016	-2,16	-16,10	7,70	4,33	-2,70	16,80	6,00	-3,30	14,90	11,25	3,30	25,00	15,69	8,30	25,60	21,69	10,70	32,20	24,06	14,90	32,50				
Итого:	0,01	-20,30	12,00	0,70	-18,00	16,80	5,05	-8,90	20,30	10,57	0,70	27,80	17,07	5,20	32,30	21,63	10,70	33,10	23,95	11,30	38,40				

Рисунок 5.7 – OLAP-куб

На основі даних, приведених в OLAP-кубі було побудовано діаграми, представлені на рис. 5.8-5.9. З цих діаграм можна зробити певні висновки про деякі тенденції в зміні температури за 10 років. Наприклад, на рис. 5.7 бачимо нехарактерно холодну середню температуру у лютому 2012 року,

середня температура в січні 2007 року була надзвичайно високою на протязі 10 років, взятих для аналізу.

Використовуючи OLAP-куб і будуючи на його основі різноманітні графіки зручно відстежувати певні тенденції, робити фільтрацію даних.

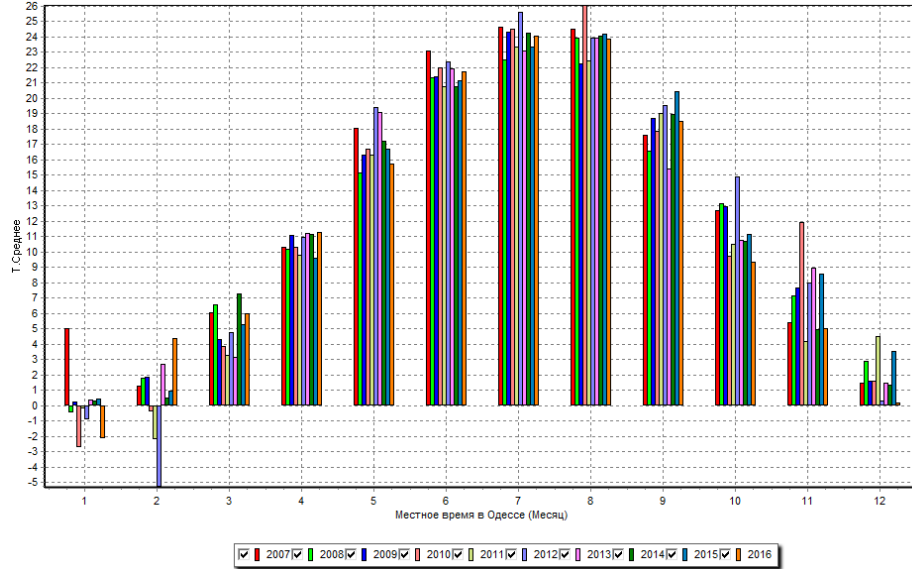


Рисунок 5.8 – Середнє значення температури в місяць з групуванням по місяцях

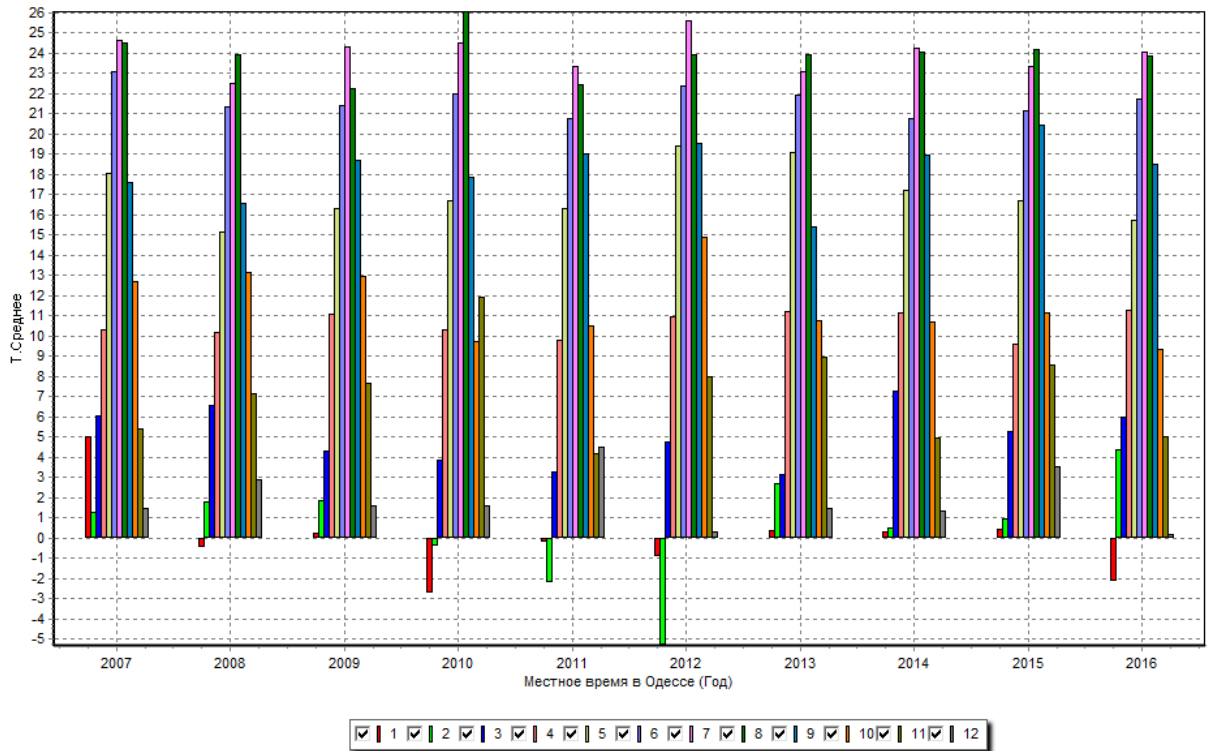


Рисунок 5.9 – Середнє значення температури в місяцях з групуванням по роках

5.2.2 Data mining

Data Mining – це процес виявлення в необроблених даних раніше невідомих нетривіальних практично корисних і доступних інтерпретацій знань, необхідних для прийняття рішень в різних сферах людської діяльності. Data Mining є одним з кроків Knowledge Discovery in Databases.

Інформація, знайдена в процесі застосування методів Data Mining, повинна бути нетривіальною і раніше невідомою. Знання повинні описувати нові зв'язки між властивостями, передбачати значення одних ознак на основі інших і т.д. Знайдені знання повинні бути застосовними і на нових даних з деякою мірою вірогідності. Корисність полягає в тому, що ці знання можуть приносити певну вигоду при їх застосуванні. Знання повинні бути в зрозумілому для користувача не математичному вигляді. Наприклад, найпростіше сприймаються людиною логічні конструкції «якщо... то...». Більш того, такі правила можуть бути використані в різних СУБД в якості SQL-запитів. У разі, коли витягнуті знання непрозорі для користувача, повинні існувати методи постобробки, що дозволяють привести їх до інтерпретованого виду.

Алгоритми, що використовуються в Data Mining, вимагають великої кількості обчислень. Раніше це було стримуючим фактором широкого практичного застосування Data Mining, проте сьогоденнє зростання продуктивності сучасних процесорів зняло гостроту цієї проблеми. Тепер за прийнятний час можна провести якісний аналіз сотень тисяч і мільйонів записів.

Завдання, які вирішуються методами Data Mining:

- Класифікація – це віднесення об'єктів (спостережень, подій) до одного з заздалегідь відомих класів.
- Регресія, в тому числі завдання прогнозування. Встановлення залежності безперервних вихідних від вхідних змінних.
- Кластеризація – це групування об'єктів (спостережень, подій) на основі даних (властивостей), що описують сутність цих об'єктів. Об'єкти усередині кластера повинні бути «схожими» один на одного і відрізнятися від об'єктів, які увійшли в інші кластери. Чим

більше схожі об'єкти усередині кластера і чим більше відмінностей між кластерами, тим точніше кластеризація.

- Асоціація – виявлення закономірностей між пов'язаними подіями. Прикладом такої закономірності служить правило, яке вказує, що з події X слідує подія Y. Такі правила називаються асоціативними.
- Послідовні шаблони – встановлення закономірностей між пов'язаними в часі подіями, тобто виявлення залежності, що якщо відбудеться подія X, то через заданий час відбудеться подія Y.
- Аналіз відхилень – виявлення найбільш нехарактерних шаблонів.

Для вирішення вищеописаних завдань використовуються різні методи і алгоритми Data Mining. З огляду на те, що Data Mining розвивалася і розвивається на стику таких дисциплін, як статистика, теорія інформації, машинне навчання, теорія баз даних, цілком закономірно, що більшість алгоритмів і методів Data Mining були розроблені на основі різних методів з цих дисциплін. Велику популярність отримали такі методи Data Mining: нейронні мережі, дерева рішень, алгоритми кластеризації, в тому числі і масштабовані, алгоритми виявлення асоціативних зв'язків між подіями і т.д.

Deductor є аналітичною платформою, в яку включено повний набір інструментів для вирішення завдань Data Mining: лінійна регресія, нейронні мережі з учителем, нейронні мережі без вчителя, дерева рішень, пошук асоціативних правил і безліч інших. Для багатьох механізмів передбачені спеціалізовані візуалізатори, що значно полегшують використання отриманої моделі і інтерпретацію результатів. Сильною стороною платформи є не тільки реалізація сучасних алгоритмів аналізу, але і забезпечення можливості довільним чином комбінувати різні механізми аналізу [31].

Для обробки даних було використано наступні методи Data Mining: автокореляція та кореляційний аналіз, карти Кохонена.

Автокореляція використовується при аналізі часового ряду з метою виявлення закономірності поведінки ряду (наприклад, сезонної періодичності або тренда), а також побудови прогнозу його подальшого розвитку.

В процесі автокореляційного аналізу розраховуються коефіцієнти кореляції (міра взаємної залежності) для двох значень вибірки, що знаходяться один від одного на певній кількості відліків, званих також лагом. Сукупність коефіцієнтів кореляції за всіма лагами є автокореляційною функцією ряду (АКФ)

При лагу рівному 0, автокореляційна функція буде максимальною і дорівнюватиме 1, тобто значення послідовності повністю корельовано саме з собою – ступінь статистичної взаємозалежності максимальна.

У міру збільшення числа лагов, тобто збільшення відстані між двома значеннями, для яких обчислюється коефіцієнт кореляції, значення АКФ будуть спадати через зменшення статистичної взаємозалежності між цими значеннями (ймовірність появи одного з них все менше впливає на ймовірність появи іншого).

Якщо у вихідній вибірці має місце тренд (плавне збільшення або зменшення значень ряду), то плавна зміна АКФ також матиме місце. При наявності сезонних коливань в вихідному наборі даних, АКФ також матиме періодичні сплески [3231].

Після обробки даних з використання методу автокореляції одержали значення автокореляційної функції наведені на рис 5.10. Звідси можна зробити висновок, що весь ряд даних має сильну статистичну взаємозалежність.

Після застосування метода автокореляції можливо виконати кореляційний аналіз. Кореляційний аналіз застосовується для оцінки ступеня лінійної залежності між парами факторів, проводиться з метою відбору та попередньої обробки вхідних полів для використання в тих, хто навчається на даних моделях. Наприклад, наявність кореляції між вхідними факторами вкрай негативно позначається при побудові лінійної регресії.

В налаштуваннях обробника вказується вхідні і вихідні поля. В результаті виходить таблиця з коефіцієнтами кореляції для кожної можливої пари з комбінації вхідного і вихідного факторів. Коефіцієнт кореляції приймає значення від -1 до 1.

Лаг	T	Po	U	Ff
0	0,999999999999984	1,000000000000008	1,000000000000003	1,000000000000013
1	0,979483980043832	0,989343736340077	0,850770388907629	0,785622731864996
2	0,949821166850157	0,969713957793535	0,688190478185987	0,672052250290259
3	0,924288236557381	0,943595045873816	0,555387482258039	0,593115896078282

Рисунок 5.10 – Результати виконання автокореляції над даними

Модуль коефіцієнта свідчить про ступінь залежності: чим ближче його значення до 0, тим слабкіше лінійна залежність. Чим ближче коефіцієнт кореляції від 0 до 1, тим сильніше пряма лінійна залежність, чим ближче від 0 до -1, тим сильніше зворотна лінійна залежність. На практиці вважається, що якщо модуль коефіцієнта кореляції більше 0,6, то лінійна залежність сильна, а якщо менше 0,3, то майже відсутня.

Варто зауважити, що низький ступінь кореляції між вхідним і прогнозованим полями не означає відсутність інших, нелінійних залежностей. Крім того, при побудові лінійних моделей варто розглянути такий вхідний фактор уважніше, так як він може бути використаний для проектування ознак (Feature Engineering) [33].

На рис. 5.11 зображені результати кореляційного аналізу.

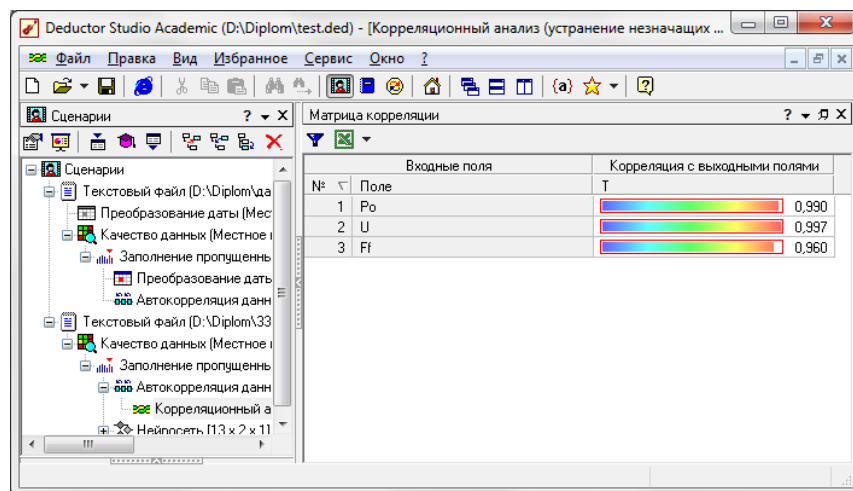


Рисунок 5.11 – Результати кореляційного аналізу

Карти Кохонена що самоорганізуються – потужний самонавчальний механізм кластеризації, що дозволяє відобразити результати у вигляді компактних і зручних для інтерпретації двовимірних карт.

Даний обробник використовується для пошуку закономірностей у великих масивах даних. Це дозволяє проводити розвідувальний аналіз даних, що відрізняється від класичних статистичних процедур, в ході яких перевіряється деякий набір висунутих гіпотез.

Основні переваги алгоритму:

- стійкість до зашумлених даних
- швидке і некероване навчання

– можливість візуалізувати багатовимірні вхідні дані

Опис алгоритму

Алгоритм функціонування карт, що самоорганізуються є один з варіантів кластеризації багатовимірних векторів – алгоритм проектування зі збереженням топологічного подібності. Тобто якщо були значно віддалені один від одного в вихідному просторі, то і на карті вони будуть значно віддалені один від одного.

Перевагою самоорганізованих карт Кохонена є те, що при навчанні використовується метод навчання без учителя, тобто результат навчання залежить тільки від структури вхідних даних [34].

На рис. 5.12-5.15 наведені процес побудова карти Кохонена, результати побудови, діаграма розсіювання та прогноз на основі карти Кохонена.

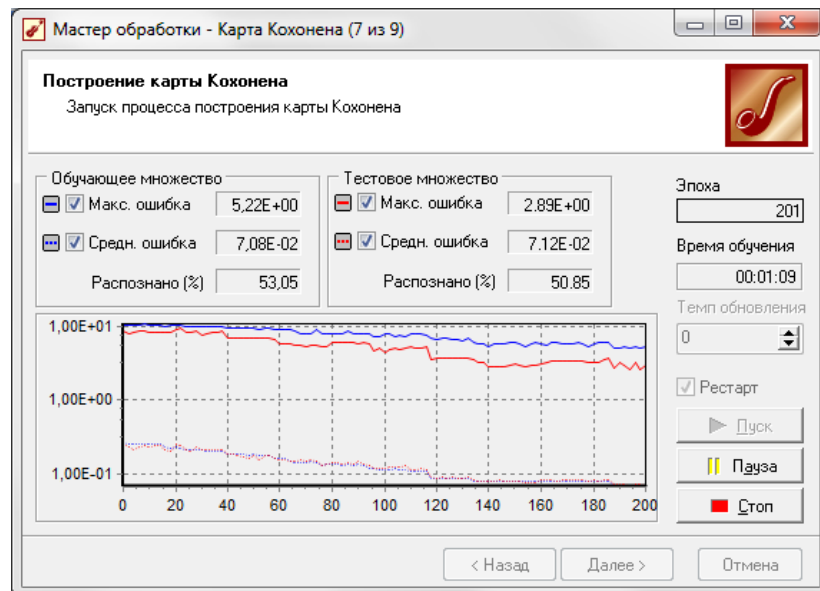


Рисунок 5.12 – Побудова карти Кохонена

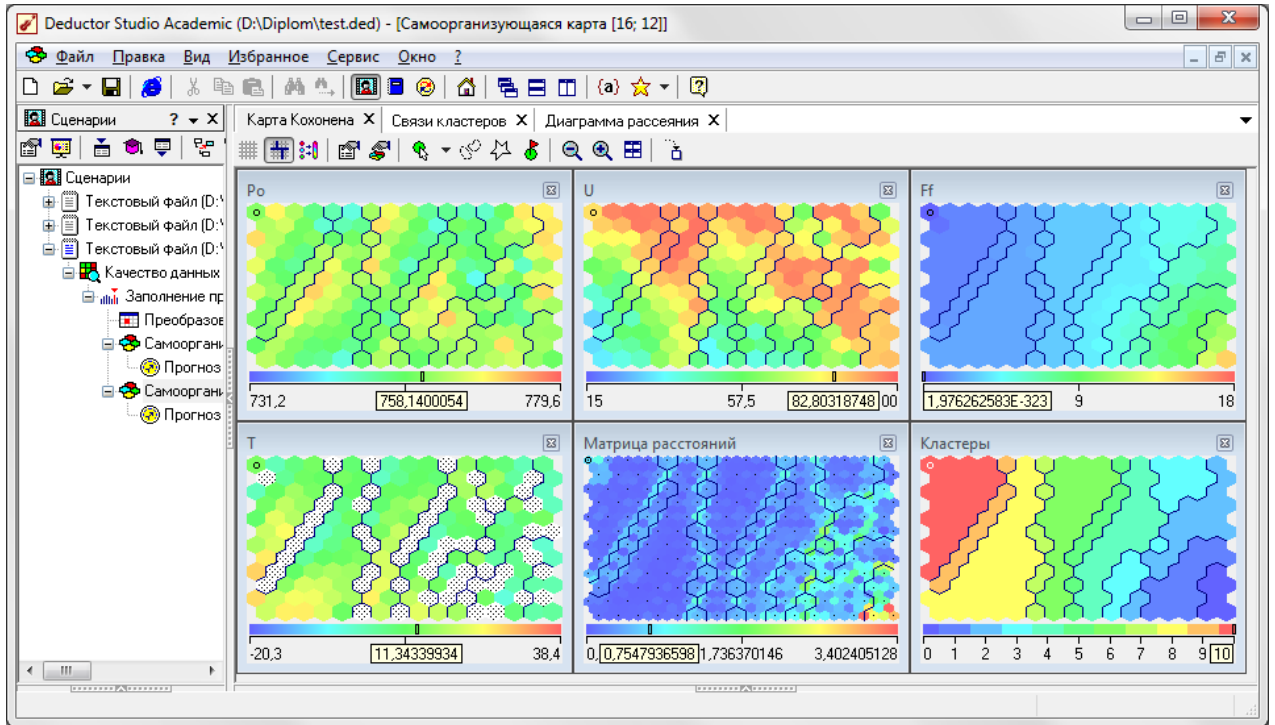


Рисунок 5.13 – Карта Кохонена

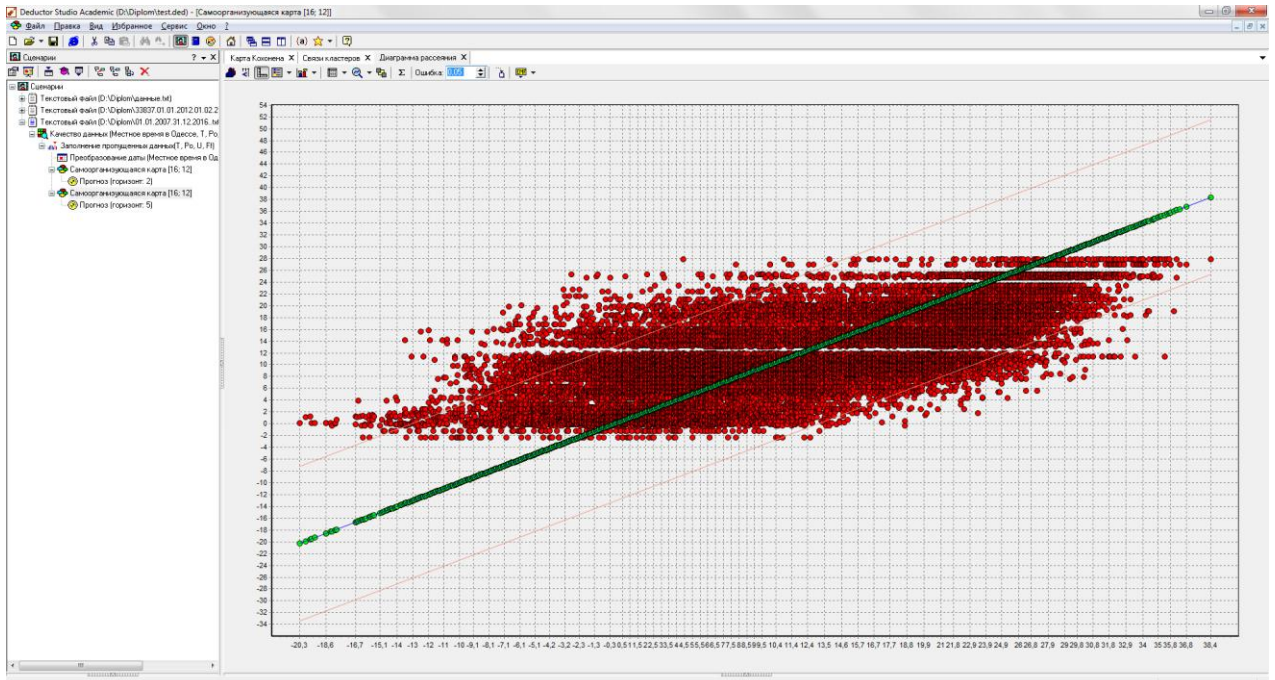


Рисунок 5.14 – Діаграма розсіювання

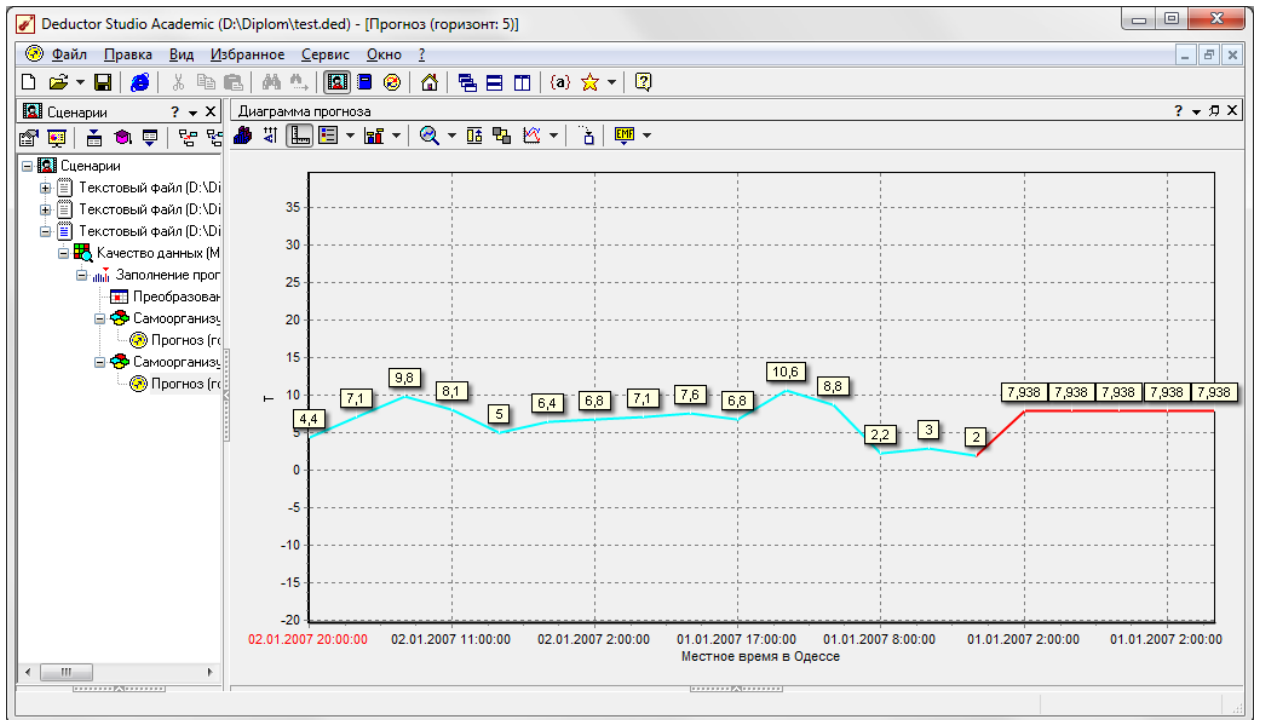


Рисунок 5.15 – Диаграмма прогнозу

6 РОЗРОБКА ДОДАТКІВ СИСТЕМИ

Навігація веб-системи один з найважливіших етапів при проектуванні системи. Кожен відвідувач, переглядаючи систему повинен, так чи інакше, до неї пристосовуватися. Це пов'язано з великою кількістю технічних рішень і дизайнерських можливостей при проектуванні веб-систем. Тим більше що в розпорядженні будь-якого користувача за статистикою всього 7 - 12 секунд, після чого, якщо йому веб-система не сподобатися він просто піде з неї, закривши сторінку браузера. Тому буде проста і зрозуміла для користувача навігація. Посилання в навігації повинні бути легко пізнавані. Користувач повинен відразу розуміти, що посилання, по якому він клацнув, є навігацією, а не простим посиланням в тексті. Все те, що створено у веб-системі, а це шапка, нижня її частина, бічні панелі, повинні бути присутніми в незмінному вигляді у веб-системі. Щоб не лякати відвідувачів і не змушувати їх шукати той чи інший блок при переході від сторінки до сторінки ситсем. Як тільки користувач зайшов до веб-системи він завжди оцінює перші 3 секунди сторінку поглядом і якщо його щось зацікавило в системі він повинен відразу ж прийняти рішення кликати на посилання чи ні. Всі номери в навігації повинні вести на будь-яку сторінку системи. Вони повинні бути протестовані неодноразово. Цим підвищується довіра з боку користувачів до проекту, тому що він буде впевнений, що натискаючи на посилання, він не побачить повідомлення про 404 помилку. Питання побудови ідеальної навігації завжди будуть присутні при створенні веб-системи. Але прагнути до її створення варто. Цим підвищується юзабіліті системи, а значить, є ймовірність, що відвідувачі будуть на нього повертатися знову і знову.

В якості дизайну веб-системи був обраний один з шаблонів, котрі запропоновані у WordPress.

Було зроблено горизонтальне меню у шапці і нижній частині (рис. 6.1-6.2). В нижній частині також присутні віджети «Про нас», «Календар», «Знайдіть нас» (рис. 6.2).

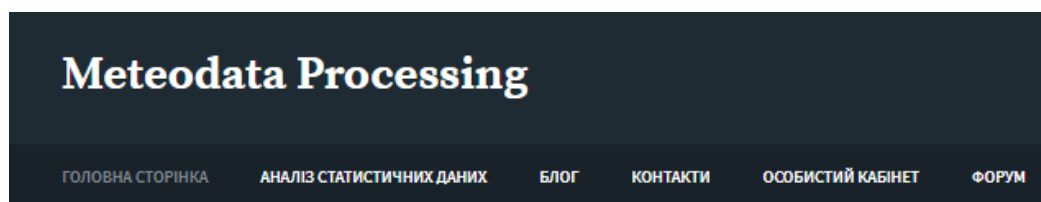


Рисунок 6.1 – Горизонтальне меню у хедері

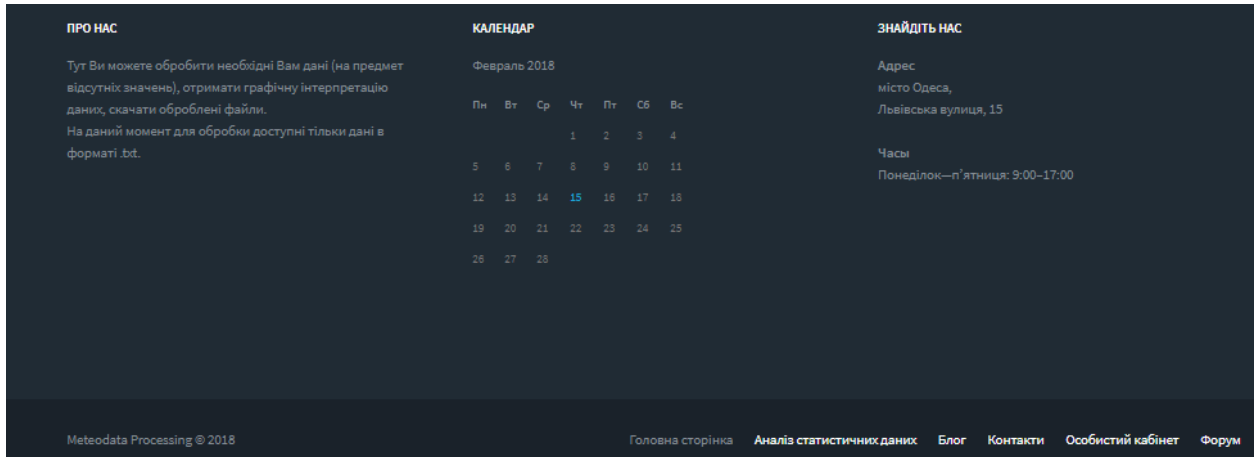


Рисунок 6.2 – Горизонтальне меню та віджети у футері

На рис. 6.3 зображена головна сторінка веб-системи Meteodata Processing. В системі присутня зручна навігація. В боковій панелі є пошук у системі, відображаються останні новини та коментарі, архів.

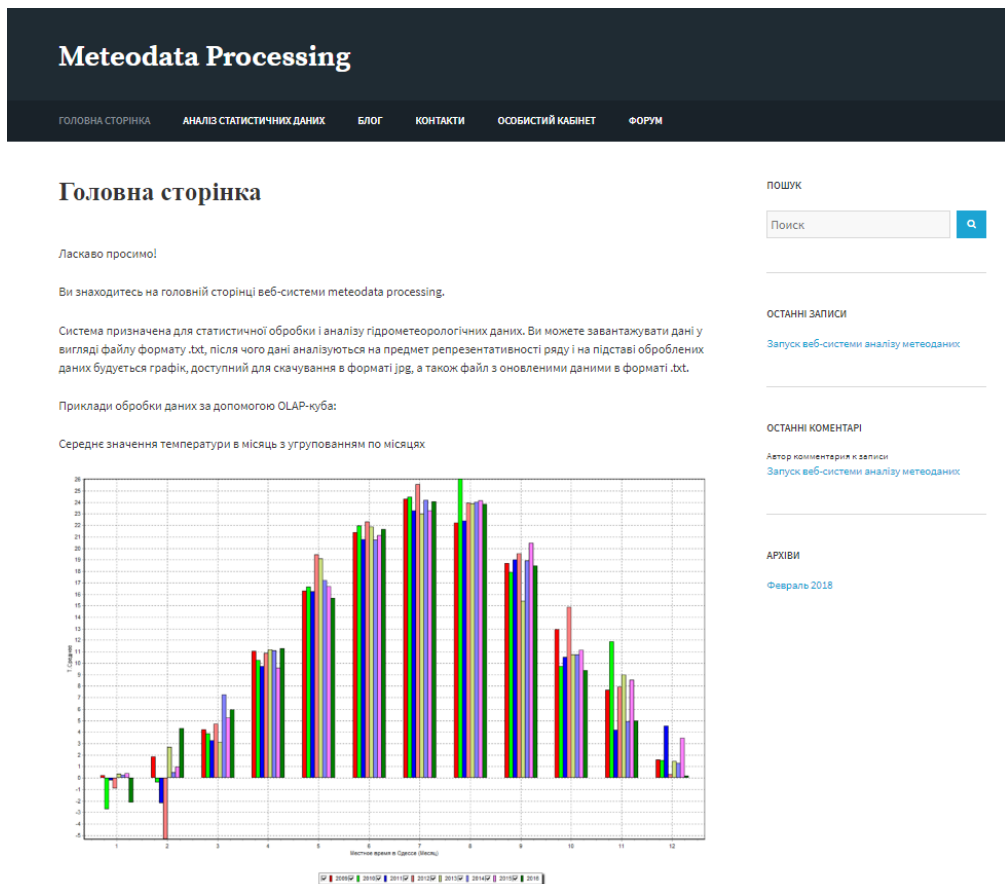


Рисунок 6.3 – Головна сторінка веб-системи Meteodata Processing

На рис. 6.4-6.7 зображені сторінки системи Meteodata Processing Аналіз статистичних даних, Блог, Контакти та Особистий кабінет, на яких можна знайти інформацію стосовно роботи з веб-системою.

Meteodata Processing

ГОЛОВНА СТОРІНКА АНАЛІЗ СТАТИСТИЧНИХ ДАНИХ БЛОГ КОНТАКТИ ОСОБИСТИЙ КАБІНЕТ ФОРУМ

Аналіз статистичних даних

Даний ресурс призначений для навчальних цілей.

В системі використовується один з методів Data mining — карти Кохонена що самоорганізуються – це потужний самонавчальний механізм кластеризації, що дозволяє відобразити результати у вигляді компактних і зручних для інтерпретації двовимірних карт.

Приклад обробки з використанням карт Кохонена:

Карта Кохонена

Р0 U R1 T Матриця розподілу Складові

Прогноз наступного значення на основі карти кохонена:

Рисунок 6.4 – Сторінка «Аналіз статистичних даних»

Meteodata Processing

ГОЛОВНА СТОРІНКА АНАЛІЗ СТАТИСТИЧНИХ ДАНИХ БЛОГ КОНТАКТИ ОСОБИСТИЙ КАБІНЕТ ФОРУМ

Запуск веб-системи аналізу метеоданих

admin
15.02.2018
1 коментарій
Редактировать

Ласкаво просимо в систему Meteodata Processing.

Система призначена для статистичної обробки і аналізу гідрометеорологічних даних. Ви можете завантажувати дані у вигляді файлу формату .txt, після чого дані аналізуються на предмет репрезентативності ряду і на підставі оброблених даних будується графік, доступний для скачування в форматі jpg, а також файл з оновленими даними в форматі .txt.

пошук

Поиск

Останні записи

Запуск веб-системи аналізу метеоданих

Рисунок 6.5 – Сторінка «Блогу»

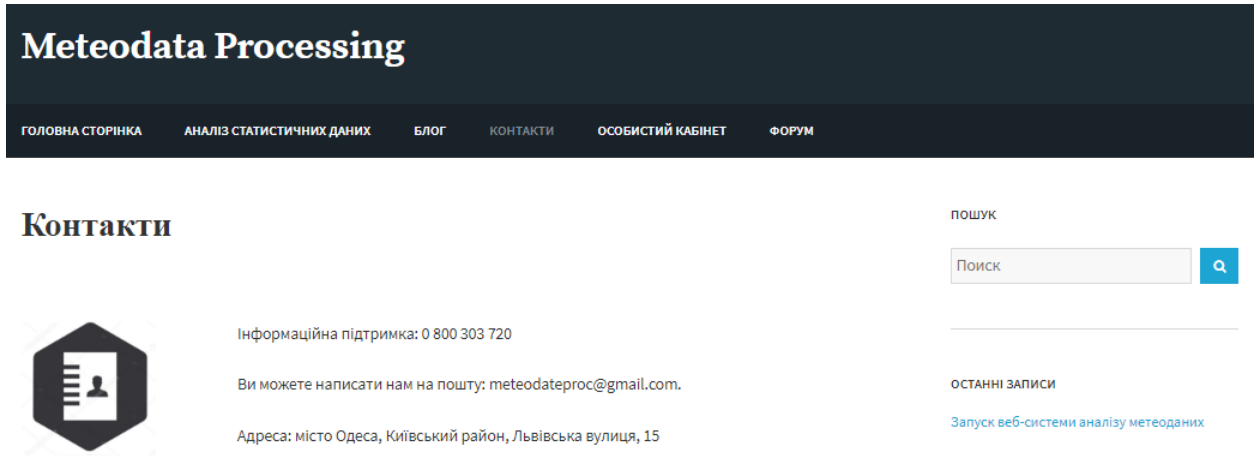


Рисунок 6.6 – Сторінка «Контакти»

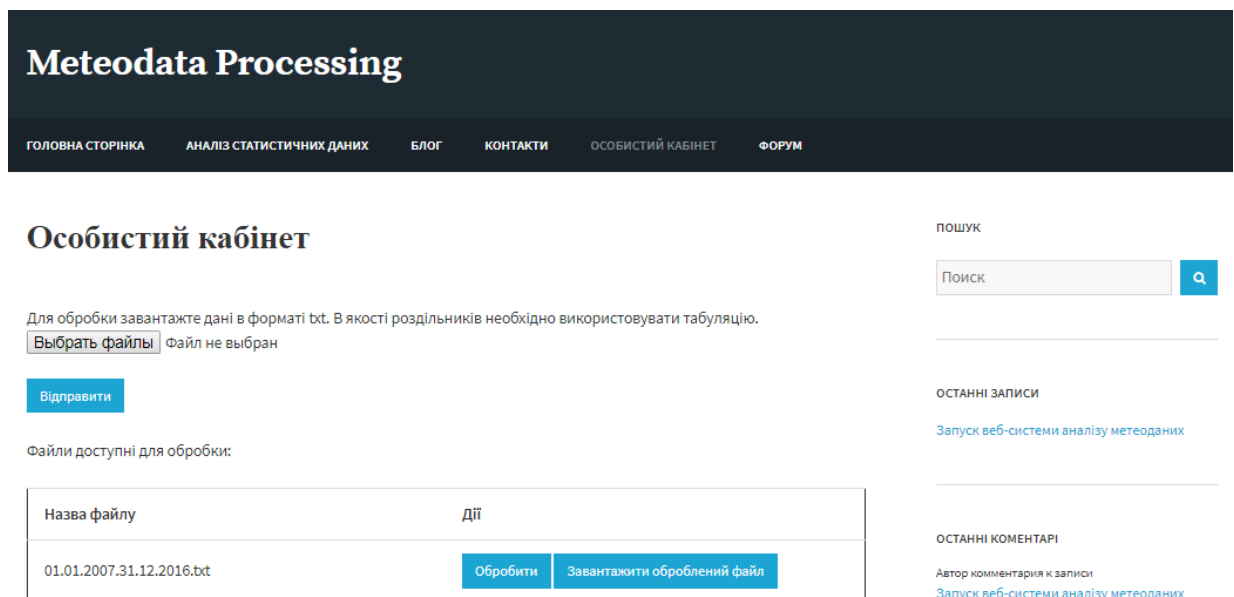


Рисунок 6.7 – Особистий кабінет користувача

У власному кабінеті користувач може завантажити дані в систему, переглянути раніше завантажені файли, зробити повторну обробку раніше завантажених даних, завантажити оброблені дані з системи на власний пристрій.

На рис. 6.8-6.9 наведений зовнішній вигляд власного кабінету після обробки даних. Дані представлені у вигляді стовпчастих діаграм (рис. 6.8), а також представлений проноз температури (рис. 6.9).

Meteodata Processing

ГОЛОВНА СТОРІНКА ПРО НАС БЛОГ КОНТАКТИ ОСОБИСТИЙ КАБІНЕТ **ФОРМ**

Особистий кабінет

Для обробки завантажте дані в форматі **txt**. В якості роздільників необхідно використовувати табуляцію.
 Файл не вибран

Файли доступні для обробки:

Назва файлу	Дії
01.01.2007.31.12.2016.txt	<input type="button" value="Обробити"/> <input type="button" value="Завантажити оброблений файл"/>

Результати обробки файлу:

Середнє значення температури в місяць з угрупованням по місяцях

ПОШУК

ОСТАННІ ЗАПИСИ

[Запуск веб-системи аналізу метеоданих](#)

ОСТАННІ КОМЕНТАРІ

[Автор коментаря к записи](#)
[Запуск веб-системи аналізу метеоданих](#)

АРХІВИ

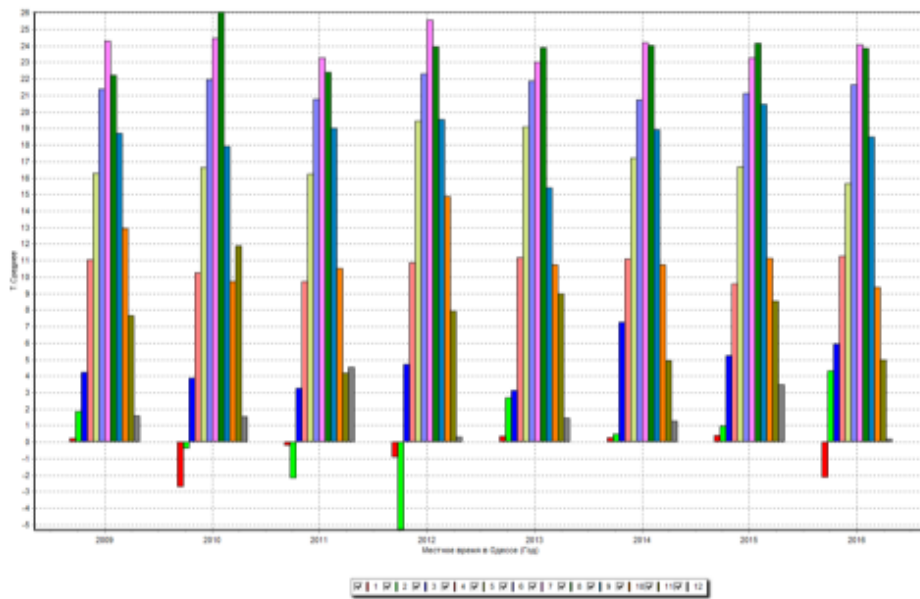
[Февраль 2018](#)

Рисунок 6.8 – Результат обробки даних, завантажених користувачем

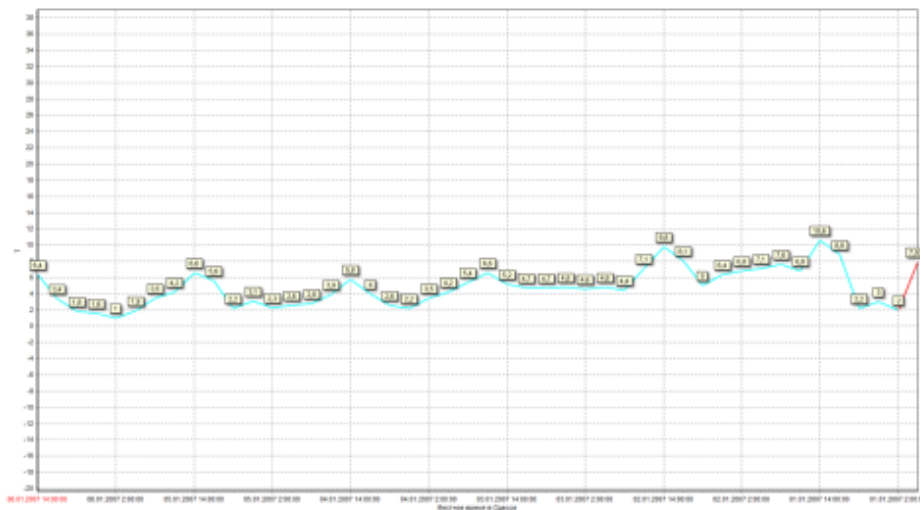
Прогноз був зроблений використанням карт Кохонена. На графіку зображено прогноз на один крок вперед (рис. 6.9).

На рисунку 6.10 зображено зовнішній вигляд панелі адміністрування системи на WordPress. Робота з панеллю адміністрування буде розглянута в додатку А.

Середнє значення температури в місяцях з групуванням по роках



Результати прогнозу на основі карт Кохонена:



ПРО НАС

Тут Ви можете обробити необхідні Вам дані (на предмет відсутніх значень), отримати графічну інтерпретацію даних, скачати оброблені файли.
На даний момент для обробки доступні тільки дані в форматі .csv.

КАЛЕНДАР

Февраль 2018

Пн	Вт	Ср	Чт	Пт	Сб	Вс
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18

ЗНАЙДІТЬ НАС

Адрес
місто Одеса,
Левітська вулиця, 15

Часи
Понеділок—п'ятниця: 9:00–17:00

Рисунок 6.9 – Результат прогнозу на основі оброблених даних користувача

The image shows the WordPress dashboard interface. At the top, there's a navigation bar with the site name 'Metoodata Processing', a user profile 'Привет, admin', and options for 'Настройки экрана' and 'Помощь'. The main content area is titled 'Консоль' and features a large welcome message: 'Добро пожаловать в WordPress! Мы собрали несколько ссылок для вашего удобства:'. Below this, there are three columns of quick actions: 'Для начала' with a 'Настройте свой сайт' button, 'Следующие шаги' with tasks like editing the homepage and adding pages, and 'Другие действия' with options to manage widgets and comments. The dashboard is divided into several widgets: 'На виду' (Overview) showing 1 post and 17 pages; 'Активность' (Activity) with a recent post 'Запуск веб-системы анализу метеоданих'; 'Быстрый черновик' (Quick Draft) with a text input and a 'Сохранить' button; and 'Новости и мероприятия WordPress' (WordPress News and Events) with a link to participate in an event. A sidebar on the left contains a menu with items like 'Главная', 'Обновления', 'Записи', 'Медиафайлы', 'Страницы', 'Комментарии', 'WPTables', 'Форумы', 'Темы', 'Ответы', 'Внешний вид', 'Плагины', 'Пользователи', 'Инструменты', 'Настройки', 'TML', 'WP File Manager', and 'Свернуть меню'.

Рисунок 6.10 – Панель адміністрування системи

ВИСНОВКИ

Тема є актуальною і може становити інтерес для спеціалістів комп'ютерних та гідрометеорологічних наук. Обробка гідрометеорологічних даних дає змогу в подальшому використовувати їх для машинного навчання. Аналіз гідрометеорологічних даних дозволяє виявляти певні тенденції у зміні клімату.

Для досягнення поставленої мети було розглянуто і проаналізовано аналогічні системи, на підставі результатів дослідження аналогів та вимог користувачів були розроблені основні вимоги до веб-системи, для побудови веб-системи була обрана дворівнева архітектура, для реалізації веб-системи були обрані СУБД MySQL, мова програмування PHP, система керування контентом WordPress, розроблена функціональна модель системи, яка дозволила в подальшому розробити фізичну модель БД, в якості бази даних було використано систему Deductor Studio Academic, для обробки і аналізу використовувалися гідрометеорологічні дані (час і дата спостереження, температура повітря, тиск, швидкість вітру та відносна вологість повітря) за період з 01.01.2007 по 31.12.2016 з джерела [30] у форматі .xls і конвертовані у .txt, дані було проаналізовано на предмет відсутніх значень, котрі були замінені медіанами, на основі метода побудови карт Кохонена було зроблено прогноз температури. Була розроблена веб-система з використанням система керування контентом WordPress.

Серед переваг системи в порівнянні з аналогами є можливість обробляти набір даних, необхідний саме користувачеві, можливість завантажити оброблені дані в універсальному форматі.txt. Дані користувача також підлягають статистичній обробці (побудова діаграм з розрахованими середніми, максимальними та мінімальними значеннями параметри з групуванням по місяцям та рокам). Результати статистичної обробки відображаються в особистому кабінеті у вигляді діаграм. Проводиться прогноз на основі даних користувача. Результати прогнозу також відображаються у вигляді графіку.

Серед недоліків системи є підключення до Deductor Studio через API, так як API доступне лише в платній версії програми, через що функціонал системи було обмежено. Система не підтримує стандартні формати гідрометеорологічних даних: GRIB (редакція 1 і 2), NetCDF, BUFR (для даних станції).

ПЕРЕЛІК ПОСИЛАНЬ

1. Ходжаева Г.К. Метеорологические методы и приборы наблюдений: Учебное пособие. – Нижневартовск: Изд-во Нижневарт. гос. ун-та, 2013. – 189 с.
2. Погода и наблюдения за ней. [Электронный ресурс] Режим доступа: <https://geographyofrussia.com/pogoda-i-nablyudeniya-za-nej/>
3. Список метеостанций России и мира. [Электронный ресурс] Режим доступа: http://meteocenter.net/_world_weather_stations.htm
4. Metview. [Электронный ресурс] Режим доступа: <https://software.ecmwf.int/wiki/display/METV/Metview>
5. Metview – инструмент для визуализации метеорологических данных. [Электронный ресурс] Режим доступа: <http://koldunov.net/?p=672>
6. OpenWeatherMap. [Электронный ресурс] Режим доступа: <https://ru.wikipedia.org/wiki/OpenWeatherMap>
7. OpenWeatherMap. [Электронный ресурс] Режим доступа: <https://openweathermap.org/>
8. Grid Analysis and Display System (GrADS). [Электронный ресурс] Режим доступа: <http://cola.gmu.edu/grads/>
9. Окладников И.Г., Титов А.Г. Веб-система для обработки и визуализации метеорологических данных ИМКЭС, Академический пр., 10/3, г.Томск, 634021
10. Шульгина Т.М., Окладников И.Г., Титов А.Г. Развитие функциональности веб-системы для обработки и визуализации метеорологических данных для исследования окружающей среды Сибири. Томск, СЦ КЛИО, ИМКЭС СО РАН
11. Веб-система для обробки і візуалізації метеорологічних даних для дослідження навколишнього середовища [Электронный ресурс] Режим доступа: <http://risks.scert.ru/>
12. Web portal on environmental sciences «ATMOS» E. P. Gordov¹, V. N. Lykosov², and A. Z. Fazliev³ [Электронный ресурс] Режим доступа: <https://www.adv-geosci.net/8/33/2006/>
13. ATMOS [Электронный ресурс] Режим доступа: <http://atmos.scert.ru>
14. Компоненты сетевого приложения. Клиент-серверное взаимодействие и роли серверов. [Электронный ресурс] Режим доступа: <http://www.4stud.info/networking/lecture5.html>

15. PHP [Электронный ресурс] Режим доступа:
<https://ru.wikipedia.org/wiki/PHP>
16. SQLite vs MySQL vs PostgreSQL: сравнение систем управления базами данных. [Электронный ресурс] Режим доступа:
<http://devacademy.ru/posts/sqlite-vs-mysql-vs-postgresql/>
17. Система управления содержимым. [Электронный ресурс] Режим доступа: https://ru.wikipedia.org/wiki/Система_управления_содержимым
18. WordPress [Электронный ресурс] Режим доступа:
<https://ru.wordpress.com/>
19. Joomla [Электронный ресурс] Режим доступа: <http://joomla.ru/>
20. Drupal [Электронный ресурс] Режим доступа: <https://www.drupal.org/>
21. WordPress, Joomla или Drupal – что лучше выбрать. [Электронный ресурс] Режим доступа: <https://www.ipipe.ru/info/wordpress-joomla-drupal-sravnenie.html>
22. Хостинг. [Электронный ресурс] Режим доступа:
<https://ru.wikipedia.org/wiki/Хостинг>
23. 000webhost. [Электронный ресурс] Режим доступа:
<https://ru.000webhost.com/>
24. Beget. [Электронный ресурс] Режим доступа:
<https://beget.com/ru/free-hosting;>
25. Timeweb. [Электронный ресурс] Режим доступа:
[https://timeweb.com/ru/services/hosting/.](https://timeweb.com/ru/services/hosting/)
26. Функциональное моделирование [Электронный ресурс] Режим доступа: <https://code.google.com/archive/p/ide-editor/wikis/IDEF0.wiki>
27. Бази моделей і системи управління базами моделей [Электронный ресурс] Режим доступа: <http://diplomukr.com.ua/news/2009/02/09/303>
28. Studio [Электронный ресурс] Режим доступа:
<https://basegroup.ru/deductor/components/studio>
29. Deductor Руководство администратора Версия 5.2 [Электронный ресурс] Режим доступа:
https://basegroup.ru/system/files/documentation/guide_admin_5.2.0.pdf
30. Архив погоды в Одессе [Электронный ресурс] Режим доступа:
http://rp5.ua/Архив_погоды_в_Одессе
31. Data Mining – добыча данных [Электронный ресурс] Режим доступа:
<https://basegroup.ru/community/articles/data-mining>
32. Автокорреляция [Электронный ресурс] Режим доступа:
<https://basegroup.ru/deductor/function/algorithm/autocorrect>

33. Корреляционный анализ [Электронный ресурс] Режим доступа:
<https://basegroup.ru/deductor/function/algorithm/correlation-analysis>

34. Карта Кохонена [Электронный ресурс] Режим доступа:
<https://basegroup.ru/deductor/function/algorithm/kohonen>

35. Инструкция по работе с WordPress [Электронный ресурс] Режим
доступа: <http://oddstyle.ru/instrukciya-po-rabote-s-wordpress-rukovodstvo-dlya-novichkov>

ДОДАТОК А АДМІНСТРУВАННЯ У WORDPRESS

А.1 Консоль WordPress

Щоб почати роботу з WordPress-системою, вам потрібно увійти до веб-системи, використовуючи свій логін і пароль, який ви створили при установці системи. Увійти до системи можна за адресою <http://www.vashdomain.com/wp-admin>, де ви побачите форму представлену на рис. А.1.

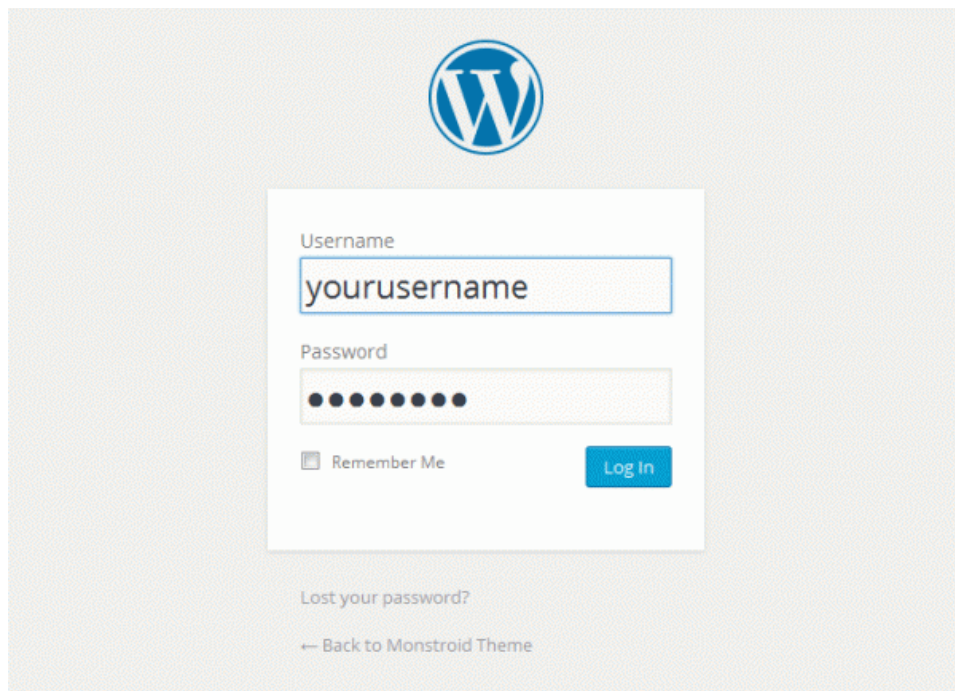


Рисунок А.1 – Форма авторизації WordPress

Після того як ви введете свої облікові дані, ви увійдете в консоль WordPress. Консоль – це головна область адміністрування, де ви можете налаштовувати параметри вашої системи, створювати записи і сторінки, встановлювати теми і плагіни і робити багато всього іншого. При першій установці WordPress дуже корисним виявляється розділ «Welcome to WordPress» (рис. А.2), що знаходиться в самому верху екрана, який має посилання на найпопулярніші інструменти і дії.

Ви можете також подивитися, як виглядає ваша система в даний момент в браузері.

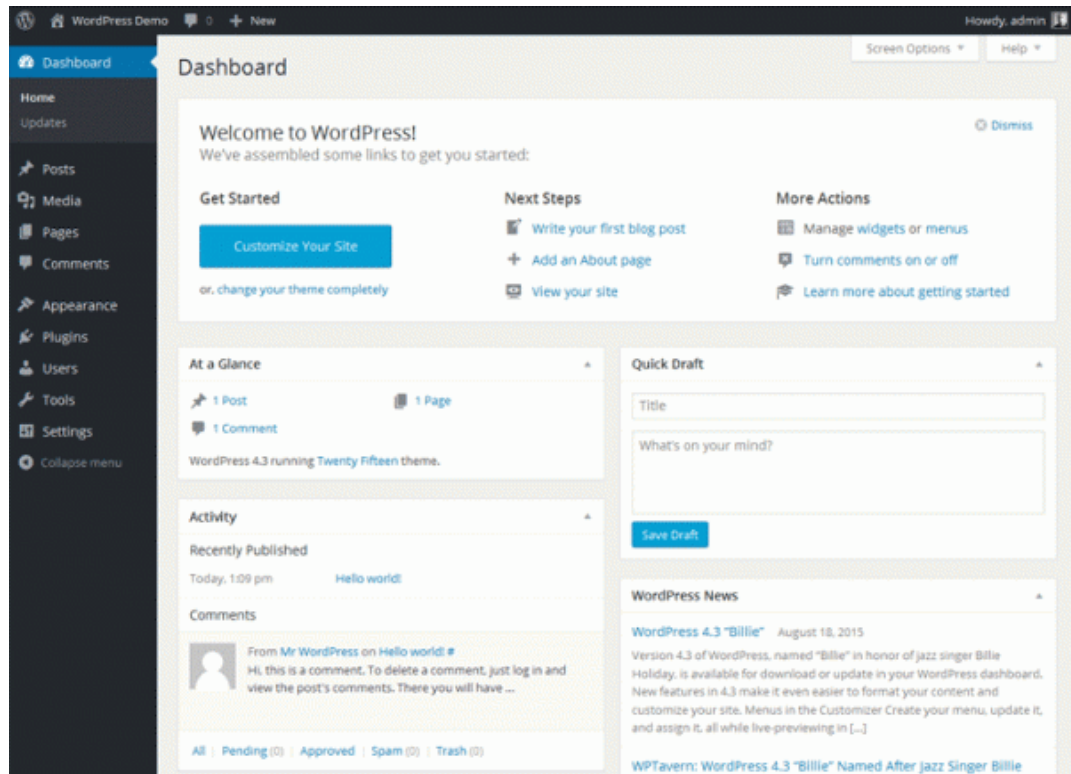


Рисунок А.2 – Розділ «Welcome to WordPress»

Загальні параметри

Є деякі параметри, які повинні бути налаштовані для кожної нової системи WordPress. Також це хороший спосіб познайомитися з консоллю. Для початку відвідаємо розділ «Settings» – «General» («Параметри» – «Загальні») в меню адміністратора і подивимося, які налаштування можемо поставити для системи (рис. А.3):

- Site Title (назва вашої системи)
- Site Tagline (опис вашої системи)
- WordPress Address (розташування WordPress)
- Site Address (домашня сторінка вашої системи, якщо ви встановлювали WordPress в підпапку)
- Email Address (поштову адресу адміністратора)
- Membership (поставте галочку навпроти цього пункту, якщо ви хоти-ті, щоб користувачі могли реєструватися у вашій системі)
- New user default role (нехай залишиться як Subscriber (Підписчик))
- Date Format (формат дати)
- Time Format (формат часу)
- Day the Week Starts (Перший день тижня)

– Site Language (мова системи)

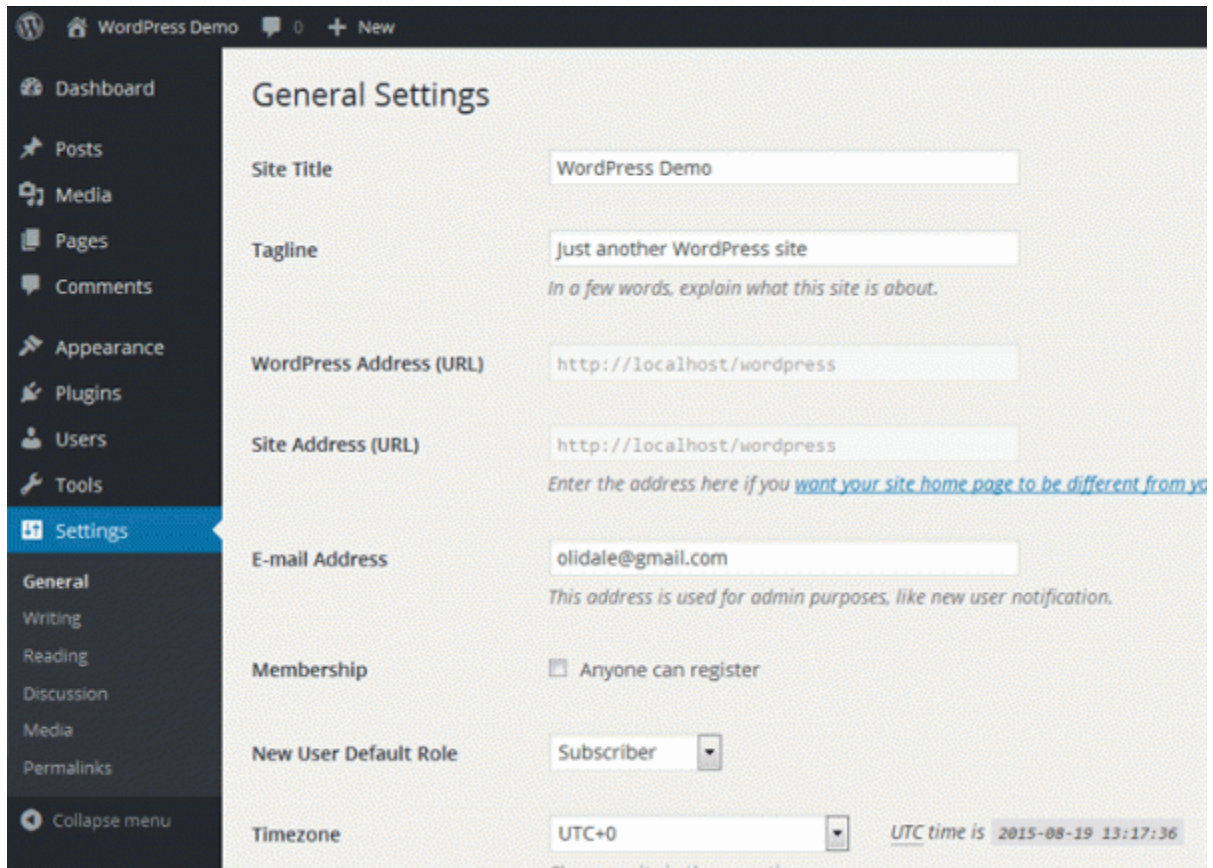


Рисунок А.3 – Розділ Settings – General

Постійні посилання

Наступна річ, яку потрібно буде зробити – це налаштувати ваші постійні посилання (Permalinks). Постійні посилання – це структура URL-адресу, яка буде використовуватися у вашій системі. За замовчуванням WordPress використовує постійні посилання наступного формату (рис. А.4):

`http://www.yourdomain.com/?p=123`

Такий формат є не найкрасивішим і не самим ласкавим до користувачів і пошукових систем, тому ви можете перейти до формату з датами, як, наприклад:

`http://www.yourdomain.com/2015/08/19/sample-post/`

Даний формат виглядає набагато краще. Проте ідеальним рішенням буде варіант Post Name, який є досить коротким і зручним:

`http://www.yourdomain.com/sample-post/`

Щоб перейти до цього формату, необхідно вибрати Post Name, після чого натиснути на кнопку «Save Changes».

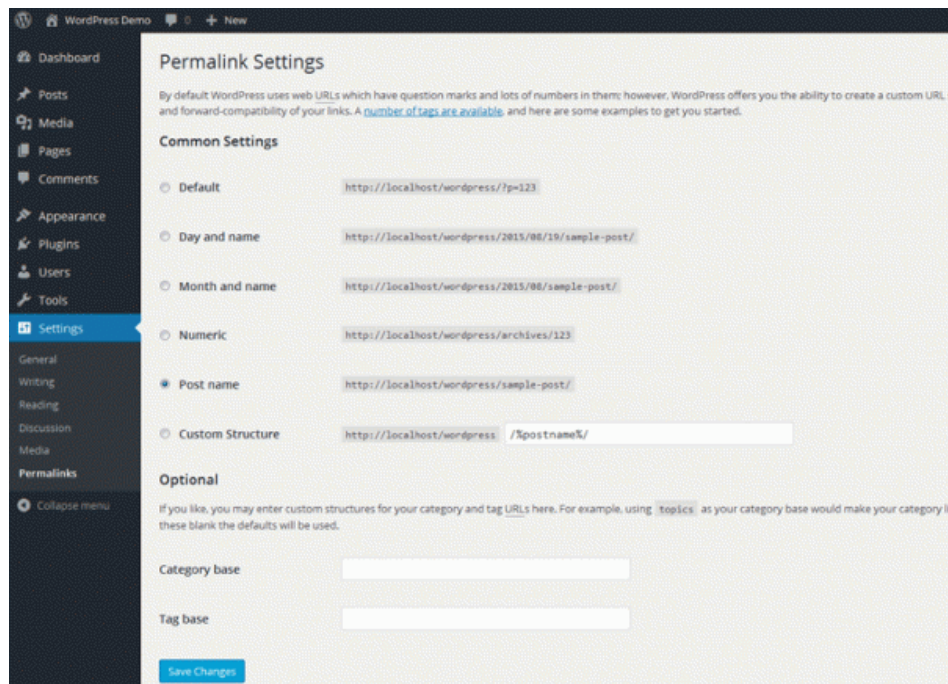


Рисунок А.4 – Розділ Settings – Permalinks

Параметри обговорення

Наступний крок – параметрів параметрів «Discussion» (Обговорення), які відповідають за можливості коментування. Люди можуть залишати свої коментарі до ваших записів. Опції Discussion знаходяться в розділі Settings (рис. А.5).

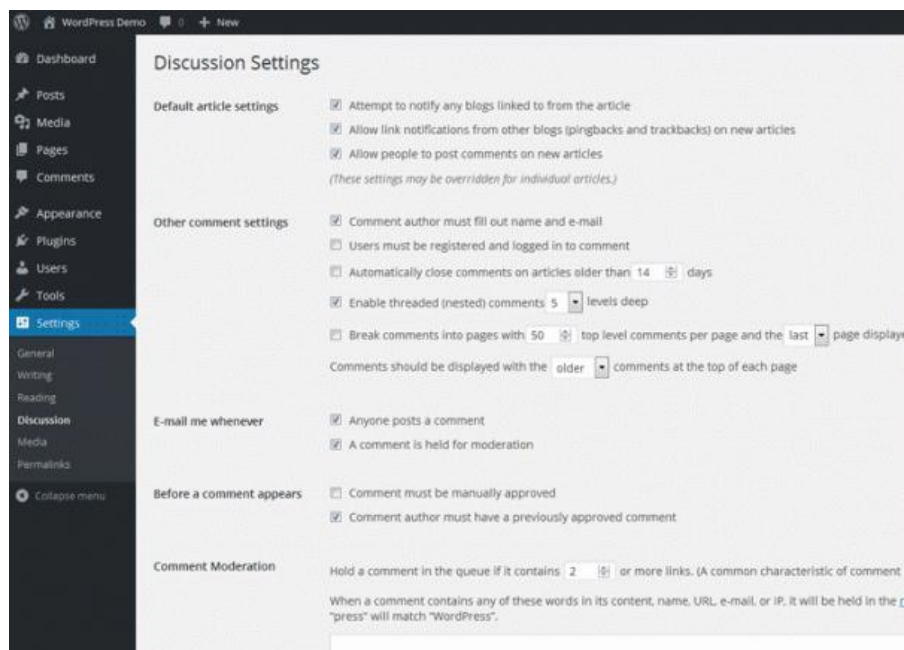


Рисунок А.5 – Розділ Settings – Discussion

У самому низу сторінки ви можете знайти опції для аватарів (Avatars) (рис. А.6). Аватари – це невеликі іконки, які виводяться на екран в коментарі поруч з ім'ям людини, що опублікував його. Ви можете вибрати стандартні зображення для тих користувачів, які не мають своїх аватарів, або ви можете взагалі відключити їх при бажанні.

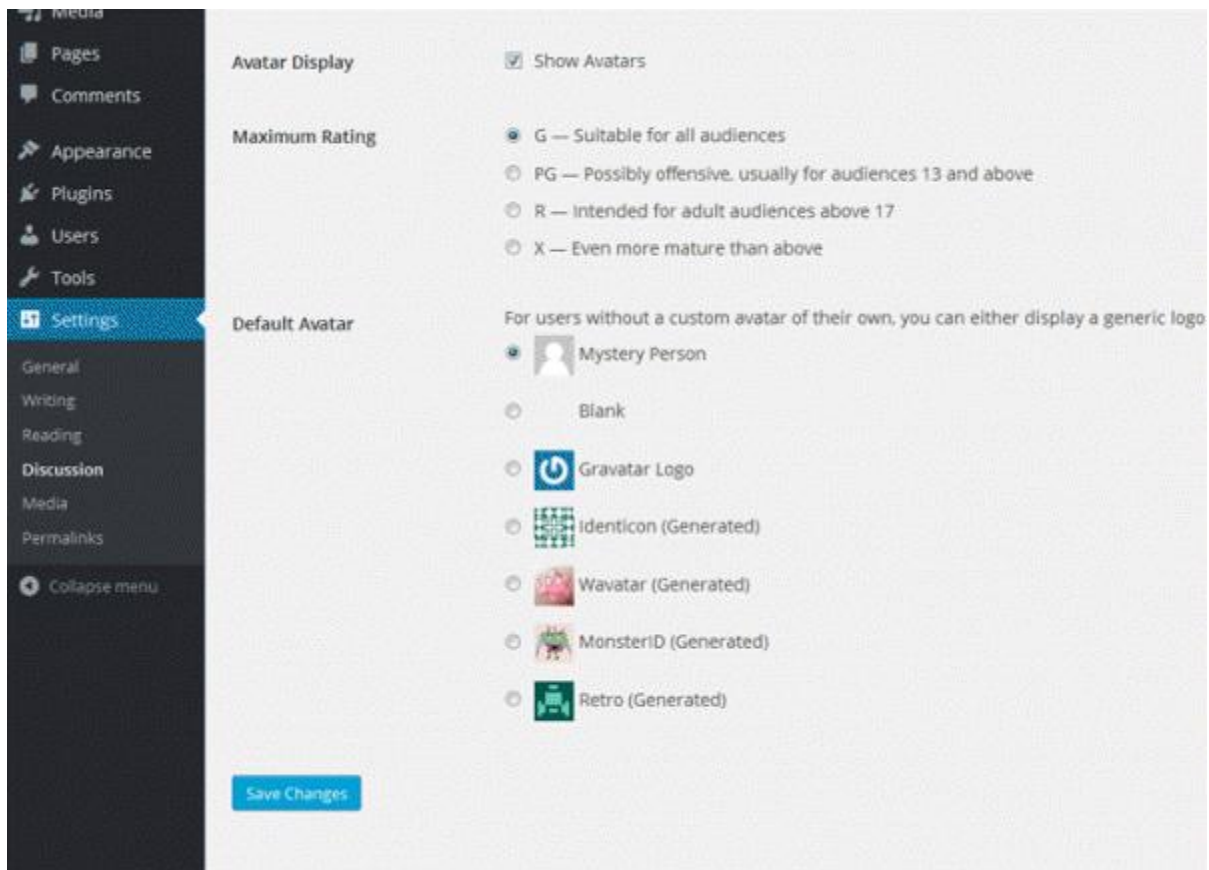


Рисунок А.6 – Розділ Settings – Discussion

А.2 Створення контенту

Різниця між записами і сторінками

За замовчуванням в WordPress ви можете створювати два типи контенту: записи і сторінки.

Сторінки – це елементи системи, котрі використовуються для розміщення статичного контенту. Вони використовуються для створення таких речей, як сторінки про себе, контактні сторінки, сторінки з положеннями та умовами використання і т.д. За замовчуванням коментарі відключені для сторінок, оскільки вони там не потрібні. Сторінки не

виводяться на головній сторінці вашої системи, коли ви додаєте контент до них; вони розроблені для того, щоб виводити посилання на них в меню, в хедері, в футері або в інших місцях.

Записи складають основний контент вашого блогу. При публікації записи вона буде виводитися в самому верху головної сторінки. У записах коментарі включені за замовчуванням. Ви можете сортувати записи по рубриках, за датою і по мітках. Ваші записи виводяться в зворотному хронологічному порядку і потрапляють в ваш RSS-фід веб-системи, в той час як сторінки – немає.

Рубрики і мітки

Щоб згрупувати ваші записи, ви можете використовувати рубрики і мітки. При публікації записи ви можете вибрати рубрику, яка буде містити всі записи, пов'язані з даною тематикою. Користувач може вибрати цю рубрику у вашій системі і подивитися всі записи, котрі входять в неї.

Ви можете додати рубрики на екрані Post Editing (Редагування запису).

Щоб зробити це, відвідайте «Posts» – «Categories» («Записи» – «Теми») і додайте нову рубрику (рис. А.7).

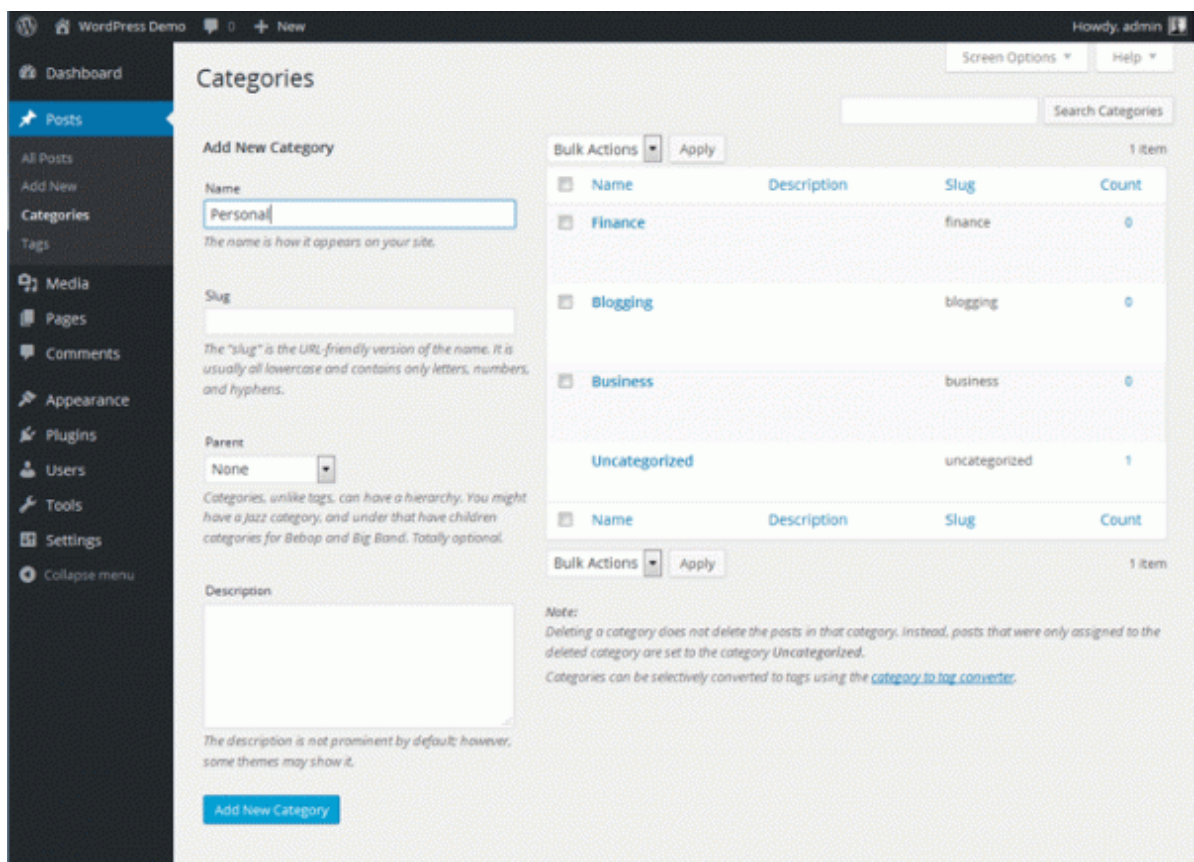


Рисунок А.7 – Розділ Posts – Categories

Інший спосіб згрупувати контент – це використовувати мітки (tags). Коли ви публікуєте запис, ви можете додати до нього деякі мітки, які будуть описувати статтю. Потім, ці мітки можуть бути використані в інших записах.

Використання міток не є обов'язковим.

Додавати мітки ви можете точно так само, як і рубрики – або ще до того, як ви будете писати контент («Posts» – «Tags»), або в процесі його написання.

Редактор записів

У редакторі записів ви можете створювати нові записи, редагувати їх, публікувати їх в системі, планувати їх публікацію і т.д. Перейдіть до розділу «Posts» – «Add New» (рис. А.8), щоб почати процес додавання записів.

Екран Add New Post містить невелике поле, де ви можете ввести заголовок вашого запису, і широке поле під ним, де ви можете ввести контент вашого запису. Трохи вище цього великого поля ви можете побачити кнопку візуального редактора (Visual Editor), який використовується для візуального форматування тексту: присвоювання напівжирного, курсивного виділення, розміщення списків, цитат, посилань і т.д. На одному рівні з вкладкою візуального редактора знаходиться кнопка Add Media, котра дозволяє вам додавати зображення до публікації.

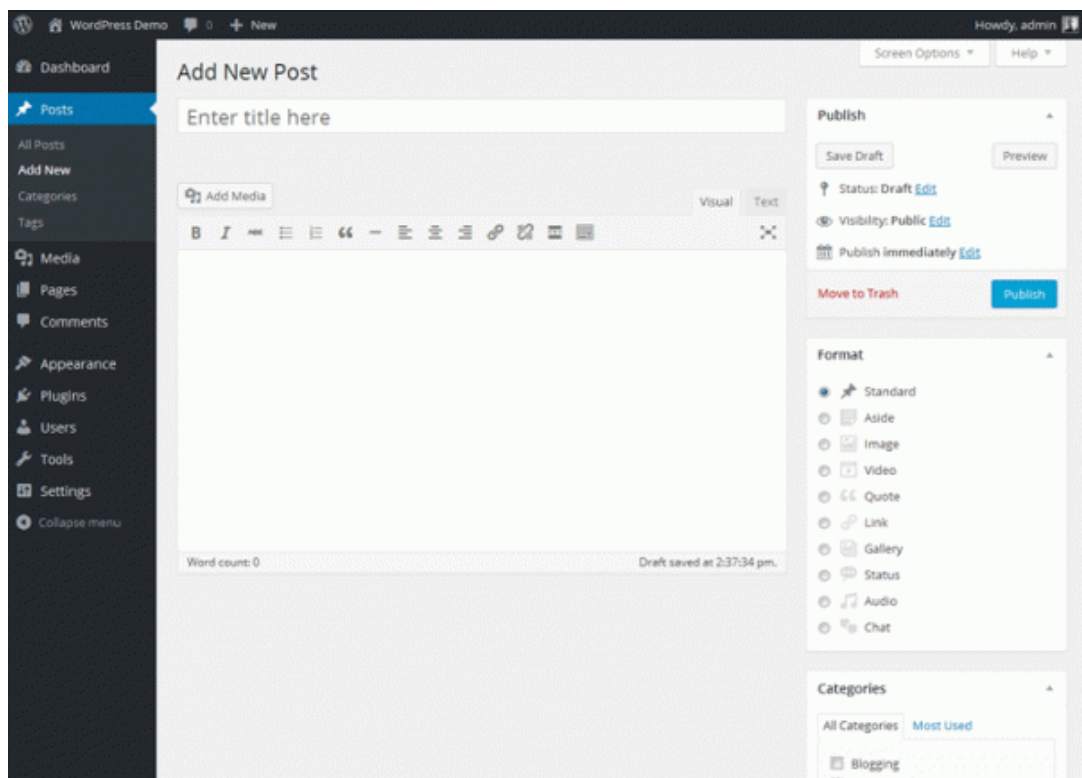


Рисунок А.8 – Розділу Posts – Add New

Якщо ви хочете вставити посилання, виділіть фрагмент тексту, після чого натисніть на кнопку із посиланням і введіть URL (рис. А.9) для вашого посилання. Ви можете потім клацнути по кнопці «Preview», щоб подивитися, як буде виглядати запис у вашій системі.

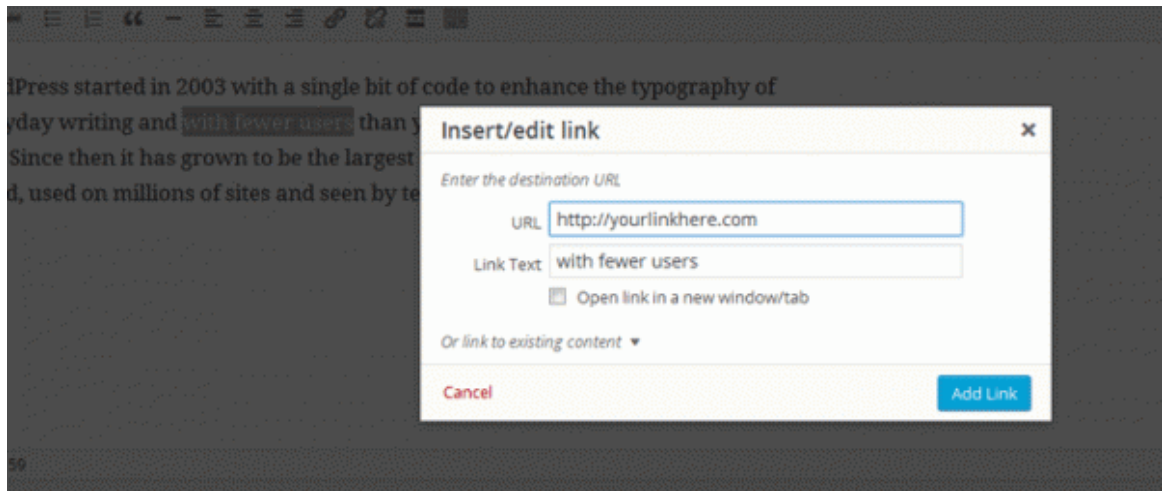


Рисунок А.9 – Додавання URL посілань

За замовчуванням ви маєте в своєму розпорядженні лише деякі обмежені опції форматування, доступні в візуальному редакторі. Якщо ви клацнете по кнопці «Toolbar Toggle» (рис. А.10), з'явиться ще один рядок з різними опціями, де будуть знаходитися додаткові елементи форматування, які ви можете використовувати при наборі тексту вашої запису.

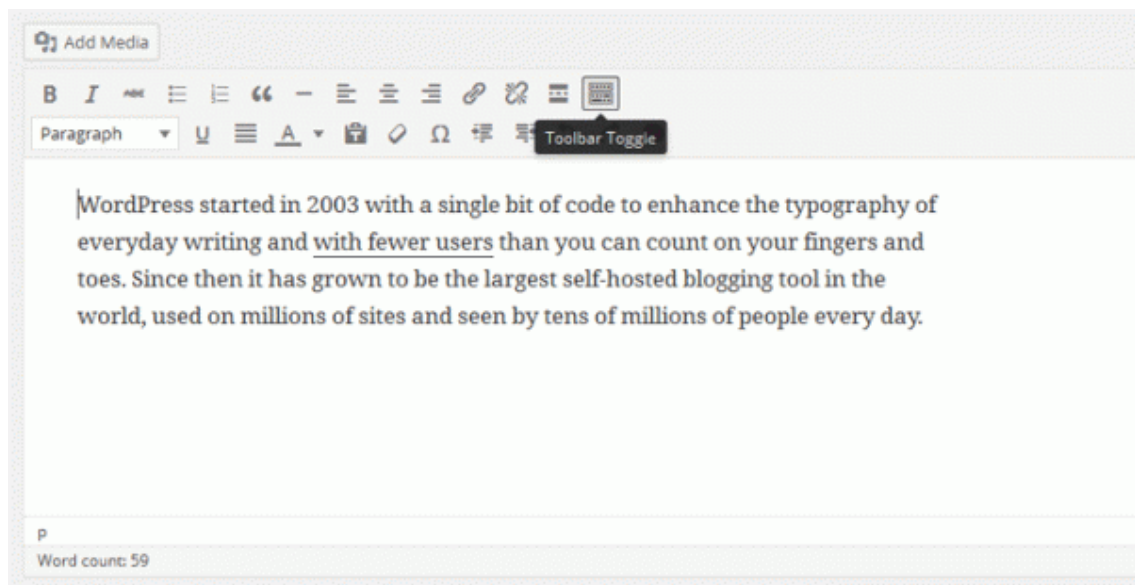


Рисунок А.10 – Toolbar Toggle

Медіа-бібліотека (або медіатека)

Важлива частина будь-якого запису блогу – зображення. Всі зображення, відеофайли і аудіофайли в WordPress знаходяться під контролем медіа-бібліотеки. При написанні записи ви можете звернутися до медіа-бібліотеці, завантаживши зображення або вибравши будь-який з вже завантажених раніше. На екрані редагування записи натисніть на кнопку Add Media, розташованої над головним полем візуального редактора. З'явиться вікно, в якому будуть виведені зображення, котрі вже використовувалися в системі (рис. А.11).

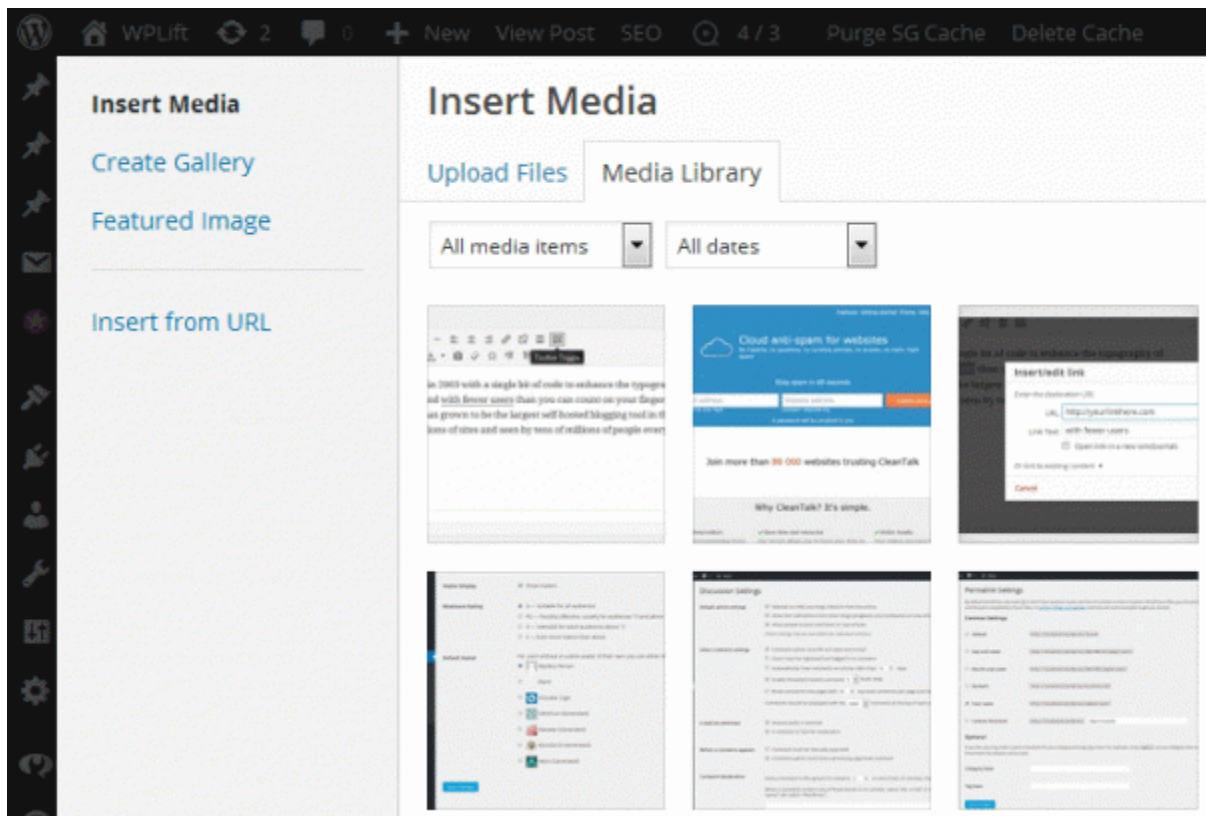


Рисунок А.11 – Медіа-бібліотека

Щоб додати нове зображення, клацніть по вкладці «Upload Files», яка призведе до появи медіа-завантажувача. Тут ви можете або перетягнути ваш набір зображень з комп'ютера, або клацнути по кнопці Select Images і вибрати зображення вручну, розташовані на вашому комп'ютері. Після того як файли будуть завантажені, вам будуть запропоновані деякі додаткові опції, які стоять з правої сторони вікна (рис. А.12).

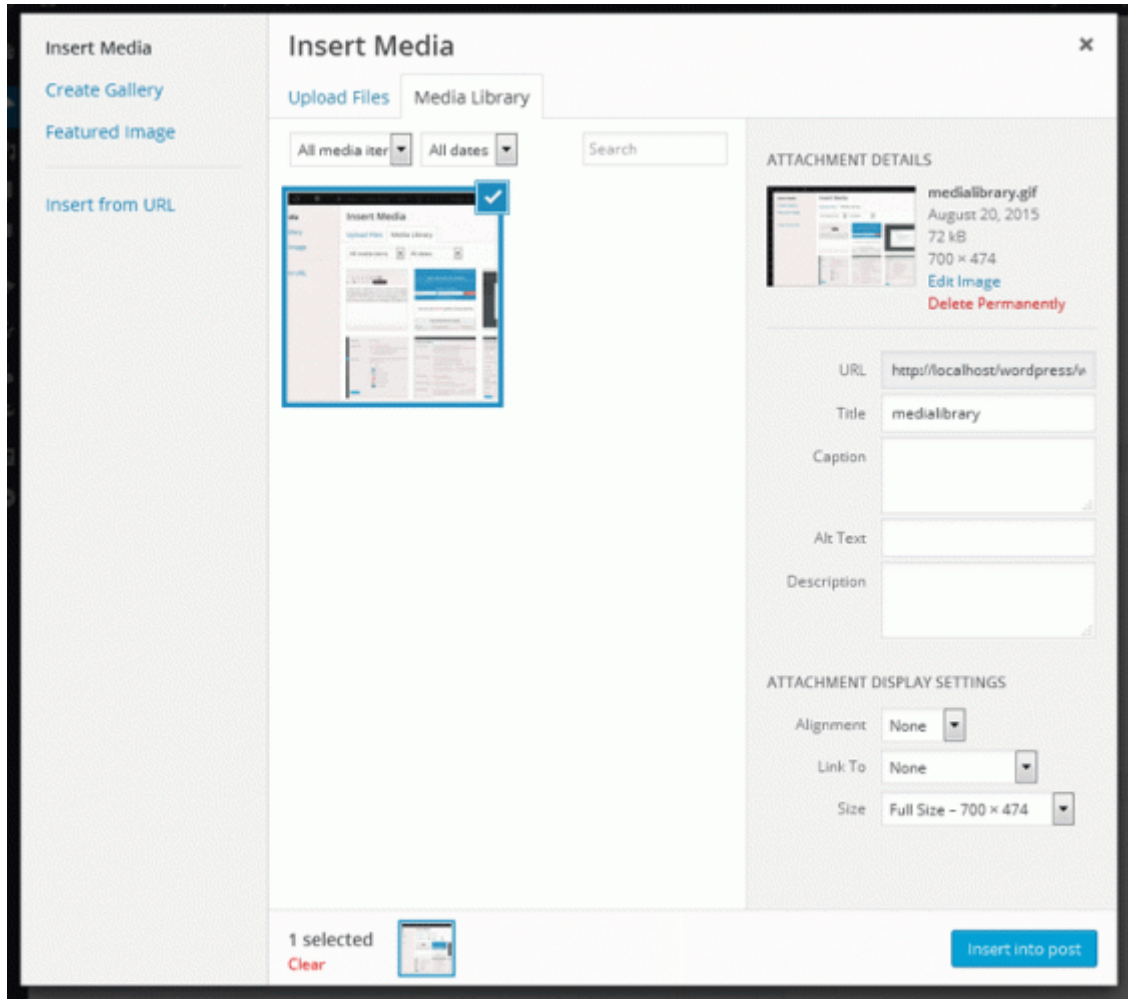


Рисунок А.12 – Налаштування меді-файлу

Ви можете привласнити кожному зображенню свій заголовок і підпис, задати текст Alt і опис, а також вибрати деякі опції відображення для зображень. Ви можете вибрати вирівнювання (без вирівнювання, по лівому краю, по правому краю, по центру), задати посилання (без посилання, посилання на довільний URL, посилання на сторінку вкладення, посилання на оригінальний файл), а також вибрати розмір (повний розмір, середній розмір або мініатюра).

Після натискання кнопки «Insert into Post», ваше зображення буде додано до запису. Якщо вам знадобиться згодом змінити будь-яке з цих налаштувань, ви можете один раз клацнути по необхідному зображенню, після чого з'явиться панель інструментів, де ви зможете відредагувати зображення.

Публікація записи

Тепер, коли ви написали свій запис, вставили в нього посилання і зображення, ви можете зробити її доступною у своїй веб-системі. Перед тим як публікувати свій запис, вам потрібно буде внести в нього деякі фінальні штрихи. З правого боку екрану редагування записи є панелі, які ви повинні заповнити. Виберіть рубрику для вашого запису зі списку, який ви створили раніше. Ви можете також додати нову рубрику безпосередньо в цьому полі. Введіть деякі мітки для запису в панелі Tags, клацаючи Add після введення кожної нової мітки. І панель «Featured Image» (Мініатюра); тут ви можете завантажити зображення, яке буде пов'язане з даним записом (рис. А.13). Різні теми виводять мініатюри по-різному, проте зазвичай вони представлені як великі зображення в самому верху записів. Завантажуються вони точно так само за допомогою завантажувача медіа-бібліотеки.

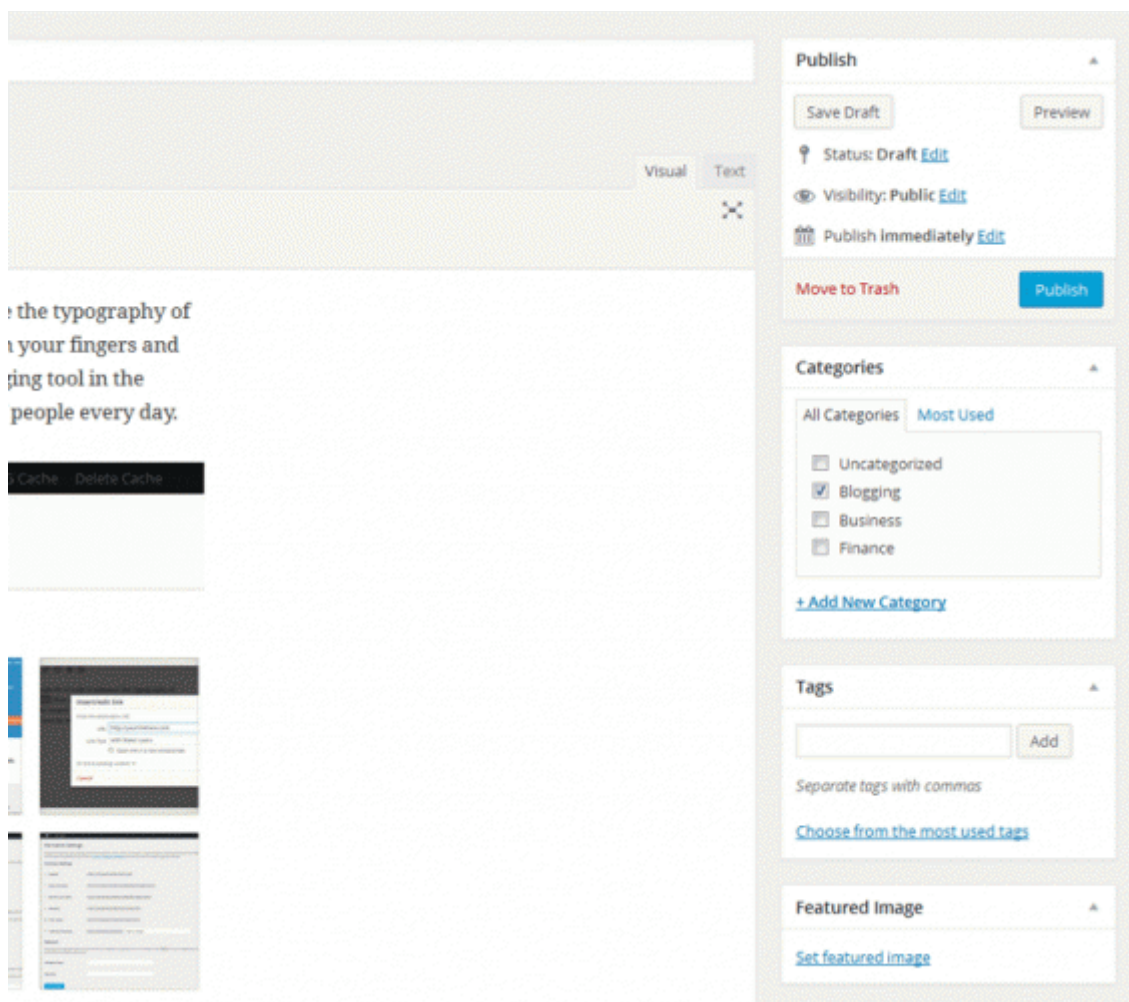


Рисунок А.13 – Панель Featured Image

Як тільки ви закінчите з даними опціями, перейдіть в самий верх сторінки – там по правому краю ви побачите поле «Publish». Щоб відразу ж опублікувати запис в системі, просто натисніть на кнопку «Publish». Запис тут же з'явиться в системі.

В WordPress також присутній механізм для планування публікацій (рис. А.14). Клацніть по кнопці Edit поруч з текстом «Publish immediately», і ви побачите календар, де ви можете встановити потрібну дату і час публікації запису. Якщо ви задасте дату публікації і клацнете Ok, кнопка Publish буде замінена на Shedule, клацання по якій приведе до планування вашої запису. Це дуже зручна можливість, якщо ви додаєте масу записів і бажаєте, щоб вони рівномірно публікувалися в системі.

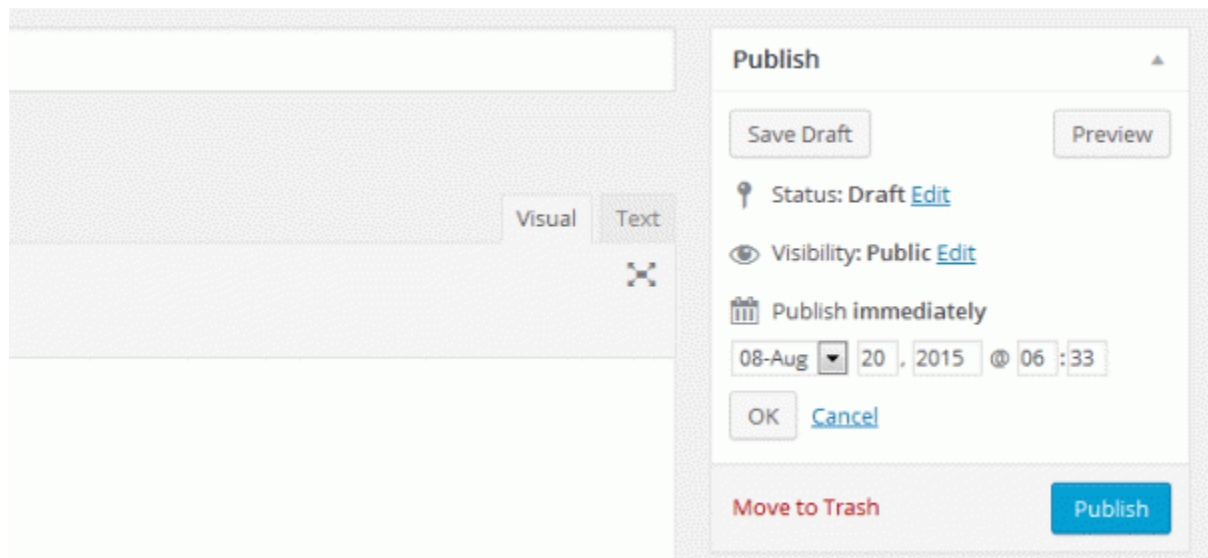


Рисунок А.14 – Механізм планування публікацій WordPress

Тепер ви знаєте, як додавати записи у вашій WordPress-системі. Сторінки додаються тим же самим способом. Відвідайте «Pages» – «Add New», і ви побачите той же самий екран редагування записів. Єдиний виняток: при створенні сторінок у вас не буде можливості вибору міток і рубрик. Сторінки також можуть плануватися [35].