# АНОТАЦІЯ –

Завданням даної бакалаврської роботи є проектування та розробка веб-рішення, яке б дозволяло переглядати та аналізувати прогноз погоди з різних джерел (сайти з погодою).

В роботі здійснено опис предметної області, вибір технології розробки, створено специфікацію вимог програмного засобу, створено документацію, описано основні алгоритми, проведено тестування та виконано економічний аналіз, де пораховано собівартість продукту та ціну одиниці продукції.

В результаті роботи створено модуль автоматичного збору інформації про погоду з різних сайтів та веб-застосування, в якому можна переглядати та аналізувати дані прогнозу погоди.

Розроблена система має гнучку розширювану структуру та архітектуру, що дозволяє легко додавати нові міста та сайти, з яких потрібно брати інформацію не змінюючи будову системи.

# ABSTRACT

The aim of this bachelor work is to design and develop a web solution that allows viewing and analyzing weather from various sources (sites with the weather).

The work carried out the subject area description, the choice of technology development, created the software requirements specification created documentation describes the basic algorithms, tested and performed an economic analysis, where calculated the cost of the product and the price per unit.

As a result of a module created automatically collect weather information from various websites and web applications, allowing you to view and analyze the data of weather forecasting.

The system has a flexible structure and extensible architecture that makes it easy to add new cities and sites that need to take information without changing the structure of the system.

# ЗМІСТ

[АНОТАЦІЯ 3](#_gjdgxs)

[ABSTRACT 4](#_30j0zll)

[ЗМІСТ 5](#_1fob9te)

[ВСТУП 8](#_3znysh7)

[РОЗДІЛ 1. ВИКОРИСТАННЯ ПРОГНОЗУ ПОГОДИ 9](#_2et92p0)

[1.1.](#_tyjcwt) Опис предметної області 9

[1.2.](#_3rdcrjn) Огляд та аналіз існуючих сайтів 12

[1.2.1. Аналіз сайту «opogode.ua» 12](#_26in1rg)

[1.2.2. Аналіз сайту «gismeteo.ua» 13](#_lnxbz9)

[1.2.3. Аналіз сайту «sinoptik.ua» 14](#_35nkun2)

[1.2.4. Аналіз сайту «meteoprog.ua» 16](#_1ksv4uv)

[РОЗДІЛ 2. ПОСТАНОВКА ЗАДАЧІ АНАЛІЗУ ДАНИХ ПРОГНОЗУ ПОГОДИ ТА АНАЛІЗ ВИМОГ НА ПЗ 17](#_44sinio)

[2.1. Постановка задачі 17](#_2jxsxqh)

[2.2. Обґрунтування технології програмної реалізації 17](#_z337ya)

[2.3. Специфікація програмного засобу 27](#_3j2qqm3)

[2.3.1. Вступ 27](#_1y810tw)

[2.3.2. Загальний опис 27](#_4i7ojhp)

[2.3.3. Характеристики та функції системи 28](#_2xcytpi)

[РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА АРХІТЕКТУРА СИСТЕМИ ДЛЯ ПОРІВНЯННЯ ПРОГНОЗУ ПОГОДИ ЗА РІЗНИМИ САЙТАМИ 37](#_1ci93xb)

[3.1. Структура системи 37](#_3whwml4)

[3.2. Архітектура системи 38](#_2bn6wsx)

[3.2.1. Архітектурний шаблон MVC 38](#_qsh70q)

[3.2.2. Опис класів 40](#_3as4poj)

[3.3. Проектування бази даних 41](#_1pxezwc)

[3.4. Проектування графічного інтерфейсу 43](#_49x2ik5)

[РОЗДІЛ 4. РОЗДІЛ РЕАЛІЗАЦІЇ ТА ТЕСТУВАННЯ 46](#_2p2csry)

[4.1. Реалізація системи 46](#_147n2zr)

[4.1.1. Автоматичний збір даних про погоду з різних сайтів та занесення їх у базу даних. 46](#_3o7alnk)

[4.1.2. Перегляд погоди на 7 днів 46](#_23ckvvd)

[4.1.3. Побудова графіків температур 48](#_ihv636)

[4.1.4. Побудова матриці відстаней між графіками температур 48](#_32hioqz)

[4.1.5. Перегляд класифікації даних сайтів прогнозу погоди 50](#_1hmsyys)

[4.1.6. Перегляд порівняльного аналізу даних прогнозу погоди різних сайтів 51](#_41mghml)

[4.2. Звіт про тестування 52](#_vx1227)

[4.2.1. Вступ 52](#_3fwokq0)

[4.2.2. Функціональне тестування 52](#_1v1yuxt)

[4.2.2.1. Функціональні тестові випадки 52](#_4f1mdlm)

[4.2.2.2. Пройдені тести, їх результати 52](#_19c6y18)

[4.2.2.3. Підсумки тестування 54](#_3tbugp1)

[4.2.3. Тестування продуктивності 54](#_28h4qwu)

[4.2.3.1. WAPT 55](#_nmf14n)

[4.2.3.2. Результати тестування 55](#_37m2jsg)

[4.2.4. Критерій успіх/провал проекту 57](#_1mrcu09)

[РОЗДІЛ 5. ЕКОНОМІЧНА ЧАСТИНА 58](#_46r0co2)

[5.1.](#_2lwamvv) Економічна характеристика проектного рішення 58

[5.2.](#_111kx3o) Інформаційне забезпечення та формування гіпотези щодо потреби розроблення товару 58

[5.3.](#_3l18frh) Оцінювання та аналіз факторів зовнішнього та внутрішнього середовищ 59

[5.4.](#_206ipza) Формування стратегічних альтернатив 61

[5.5.](#_4k668n3) Бюджетування 64

[5.6.](#_1egqt2p) Остаточний вибір стратегії 69

[ВИСНОВКИ 70](#_3ygebqi)

[ЛІТЕРАТУРА 71](#_2dlolyb)

[ДОДАТКИ 72](#_sqyw64)

[Додаток А. Діаграма компонентів 72](#_3cqmetx)

[Додаток Б. Діаграма класів 73](#_1rvwp1q)

[Додаток В. Схема бази даних 74](#_4bvk7pj)

[Додаток Д. DDL-скрипт створення бази даних 75](#_2r0uhxc)

[Додаток Ж. Вигляд головної сторінки 77](#_1664s55)

[Додаток З. Графіки температур з усіх джерел 78](#_3q5sasy)

[Додаток И. Інструкція користувача 79](#_25b2l0r)

# ВСТУП

Майже кожен користувач порівнює погоду на кількох джерелах за один сеанс. Кому довіряти, кому вірити? Інтернет насичений сайтами з прогнозами погоди, але майже всі вони користуються різними моделями прогнозування.

Погодні сайти можуть використовувати як свої унікальні методики, так і по-різному інтерпретувати дані глобальних моделей. Через це виникають стабільні відмінності в прогнозах погоди.

Метою бакалаврської кваліфікаційної роботи є аналіз та дослідження проблеми відмінностей у прогнозах погоди та розробка системи для порівняння та аналізу прогнозу погоди з різних сайтів.

Умовно роботу можна поділити на етапи. Першим етапом буде пошук технічного рішення для збору інформації про погоду з різних сайтів. Наступним етапом буде реалізація знайденого рішення та розробка відповідного модуля.

Після того, як розробка модуля для збору інформації про погоду буде завершена, можна приступати до розробки веб-застосування, що демонструватиме використання зібраних даних про погоду для аналізу та порівняння.

**РОЗДІЛ 1. ВИКОРИСТАННЯ ПРОГНОЗУ ПОГОДИ**

**1.1. Опис предметної області**

Погода – стан нижнього шару атмосфери в даній місцевості в даний момент або за якийсь проміжок часу: добу, тиждень, місяць. Погода характеризується елементами і явищами. Елементи погоди: температура повітря, вологість, тиск. До явищ погоди відносяться вітер, хмари, атмосферні опади, іноді прояви погоди носять незвичний і навіть катастрофічний характер: наприклад, урагани, сильні грози, зливи, засухи.

Погода характеризується не окремо взятими елементами і явищами, а їх сукупністю. При однаковій температурі (високій або низькій), але при різній вологості повітря, з опадами або без опадів, з вітром або без нього, погода не буде однаковою. Різним буде і її вплив на рослини та людину. Відомо, що морозна погода без вітру переноситься людьми значно легше, ніж навіть менш морозна, але з сильними вітрами. Рослини по-різному переносять однаково жарку, але вологу або суху погоду.

У помірних широтах закономірні зміни погоди впродовж доби, залежні від надходження сонячного тепла, часто порушуються зміною повітряних мас, проходженням атмосферних вихорів і фронтальними процесами.

Зміна циклональної і антициклональної погоди характерна для помірних широт. У цих широтах добре виражені зміна погоди за сезонами. Влітку погода тепла, безморозна, протягом доби навіть мінімальна температура не опускається нижче 0°. Взимку погода холодна, морозна, максимальна температура нижче 0°. Для весни і осені характерна погода з переходом температури через 0°. Це означає, що при додатній середньодобовій температурі мінімальна температуру від'ємна, а при від'ємній - додатна. Погода з переходом через 0° (з відлигою) буває і взимку. Безморозна, морозна погода, погода з переходом через 0° може бути з вітром і без вітру, хмарна і безхмарна, з опадами і без опадів. У полярних широтах увесь рік переважає морозна погода, особливо сувора вона біля Південного полюса. У екваторіальних і тропічних широтах морозних погод не буває, там немає холодної пори року.

Прогнози погоди важливі для господарського комплексу і всіх областей діяльності людей. Уточнити прогноз погоди для даної території допомагають місцеві ознаки. Необхідно користуватися не однією, а кількома ознаками (зміна температури, хмарності, вигляд хмар, напрям вітру, поведінка тварин і рослин).

Існує Всесвітня служба погоди (ВСП), що об'єднує Національні служби погоди. Вона має три світові центри: Москва, Вашингтон і Мельбурн. Дані про стан атмосфери збирають і передають 10000 метеорологічних станцій, 7000 морських суден, 3000 літаків, метеосупутники, буйкові станції, що дрейфують в океані. Опрацювання величезної кількосгі даних стала можливою тільки після появи сучасних ЕОМ. Для складання прогнозів викреслюються карти погоди (синоптичні) на момент спостережень, моделюються можливі зміни стану атмосфери. Правдивість прогнозів погоди тим більша, чим коротше термін передбачення. Прогнози на термін до трьох діб виправдовуються на 87-89%. Найбільшою точністю відрізняються спеціалізовані прогнози, наприклад авіаційні.

Прогнóз погóди (англ. Weather forecasting) — складання науково-обґрунтованих передбачень про майбутнє стану погоди. Прогноз погоди синоптичним методом ділиться на прогноз синоптичного розташування і, на його основі, власне сам прогноз погоди, тобто, значень або ходу метереологічних елементів або втілення тих чи інших атмосферних процесів (випадання дощу, утворення туману, гроза тощо).

**Класифікація**

Прогноз погоди буває:

* короткостроковий,
* довгостроковий.

Прогнози погоди можуть укладатись як для загального користування, без певної специфіки, так і спеціалізовані, приміром, авіаційні тощо.

Прогноз синоптичного розташування робиться перед власне прогнозом погоди. Він містить інформацію про майбутній розподіл і характеристики повітряних мас, фронтів, атмосферних збурень тощо. Прогнози бувають синоптичні, чисельні, статистичні (фізико-статистичні) та інші.

**Проблема, що вирішується в даній роботі**

Щоб з'ясувати прогноз погоди у своєму місті чи улюбленому курорті, зазвичай ми відвідуємо відразу кілька сайтів. Ми порівнюємо інформацію, шукаємо в інтернеті відгуки про точність того чи іншого прогнозу, складаємо власні рейтинги. Адже саме прогноз погоди деколи визначає перспективи нашої подорожі, настрій, успіх, продуктивність. Саме погода коригує наші плани на відпочинок, розваги, диктує вибір одягу.

Проте зручно було б мати сайт, де можна було б одночасно дізнатися прогноз від різних джерел і порівняти їх.

**1.2. Огляд та аналіз існуючих сайтів**

Незважаючи на величезний вибір інтернет-сайтів про погоду, лише деякі з них використовують власні прогнози. Проте саме такого сайту, який би реалізував поставлену задачу не існує. Схожим за функціональними можливостями є сайти opogode.ua, gismeteo.ua, sinoptik.ua, meteoprog.ua.

**1.2.1. Аналіз сайту «opogode.ua»**

Сервіс порівняння погоди OPOGODE - це аналітичний погодний сервіс, який порівнює погодні дані світових погодних сайтів та формує свій власний прогноз погоди.

Сервіси доступні на порталі opogode.ua

По-перше, ви відразу ж побачите поточну погоду від 5 популярних джерел, прогноз погоди на тиждень, на 14 днів та довгостроковий прогноз. Дані представлені у вигляді зрозумілої та зручної таблички, де ви можете подивитися як на середній показник по всіх джерелах, так і вибрати один з них.

По-друге, ми підготували унікальні порівняльні графіки точності прогнозу у вашому місті на підставі нашої статистики.

Третє, ми формуємо рейтинги довіри користувачів до того чи іншого джерела. Голосуйте на нашому порталі за найдостовірніше джерело. Ваш голос допоможе нам зробити рейтинги якомога точними та актуальними.

Четверте, ви зможете побачити прогноз погоди на мапі, а також скласти уявлення про погоду на дорогах за заданим вами маршрутом у визначений час.

П'яте, ми зробили наш власний погодний інформер, який ви легко зможете встановити на своїй сторінці.

Кожен, для кого має значення погода, точність прогнозу та економія часу знайде в нашій особі незамінного щоденного помічника.

Кожен день ми публікуємо новини, статті та відеосюжети про клімат, стихійні явища, подорожі, екстремальний спорт та розваги, події у світі екології, поради з адаптації до погодних умов та багато чого іншого.

На підставі наших прогнозів погоди та новин ви зможете спланувати подорож, дізнатися про цікаві туристичні місця, розібратися, що брати з собою в дорогу.

Основним недоліком є відсутність графіків та можливості порівняти погоду за джерелами.

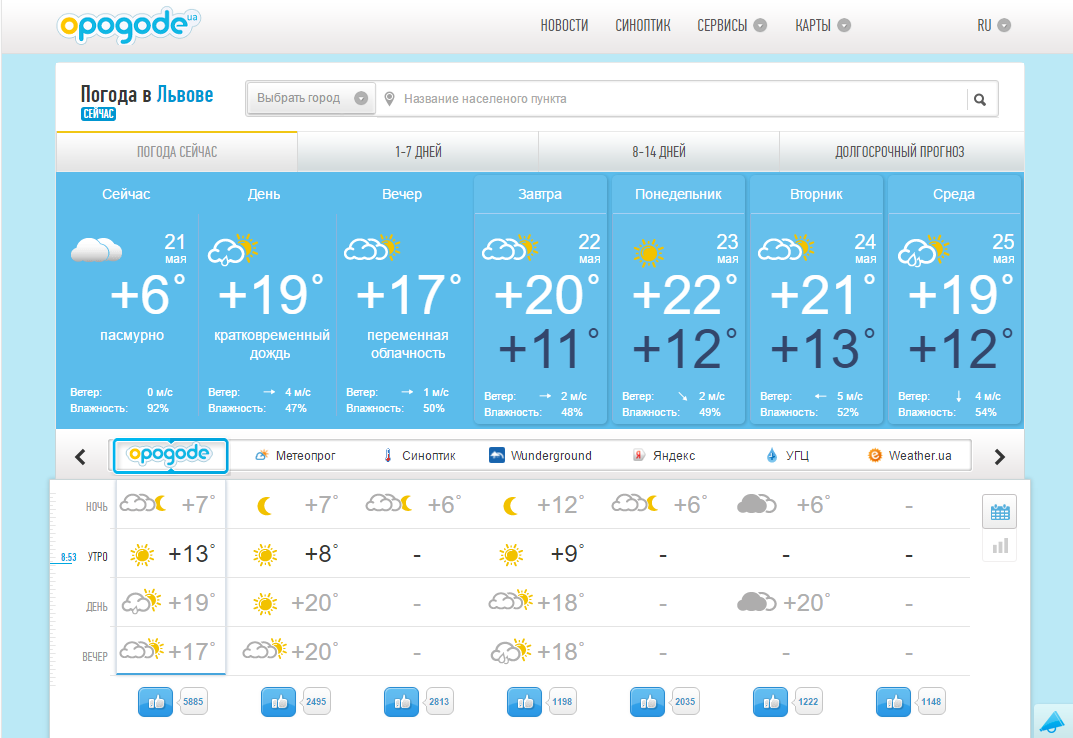


Рис. 1.2.1.1. Вигляд сайту opogode.ua

**1.2.2. Аналіз сайту «gismeteo.ua»**

У базі сайту понад 15 000 міст України й близько 50000 міст світу, в компанії використовують власну унікальну модель прогнозування погоди на основі даних GFS і UkMET.

Gismeteo був заснований в 1998 році компанією «Мэп Мейкер», яка з 90-х років займається створенням власних систем обробки метеорологічної інформації для синоптиків. Головний продукт компанії – геоінформаційна система «Метео», звідки й пішла назва сайту.

Тут є все, що має бути у хорошого погодного ресурсу: різноманітні прогнози, включаючи експериментальний прогноз на місяць, карти погоди й опадів, новини, широкий вибір інформерів для сайтів та мобільних телефонів, повноцінний портал для туристів та багато іншого. Проте, немає даних про погоду з інших сайтів, повноцінних графіків і можливості порівнювати погоду.



Рис. 1.2.2.1. Вигляд сайту gismeteo.ua

**1.2.3. Аналіз сайту «sinoptik.ua»**

Сайт надає інформацію про погоду в Україні у всіх 29815 населених пунктах та для 104 000 міст світу. Прогнози на Sinoptik беруть у Foreca, а інформацію про фактичну погоду - в українського гідрометцентру, що й дозволяє ресурсу впевнено триматися у верхніх рядках рейтингів.

Дешево і сердито: так можна охарактеризувати інтерфейс. Ніяких додаткових розділів що дратують очі, крім, власне, прогнозу погоди. Хоча на Sinoptic можна знайти й комфортні карти погоди по заданих маршрутах, і без додаткового пошуку подивитися інформацію про температуру у водоймах - наприклад, на Чорному та Середземному морях.

Основними недоліками є відсутність даних про погоду з інших сайтів, графіків та можливості порівняння погоди.



Рис. 1.2.3.1. Вигляд сайту sinoptik.ua

**1.2.4. Аналіз сайту «meteoprog.ua»**

Прогноз для 15 000 міст світу і 1209 населених пунктів України складається на власному високопродуктивному комп'ютерному кластері Meteoprog на основі чисельної моделі прогнозу погоди WRF (Weather Research and Forecasting). Це дочірня модифікація GFS, розроблена тим же NOAA, Національним центром атмосферних і океанічних досліджень США.

Відмінна риса Meteoprog - форма виведення інформації про погоду: у вигляді карт, метеограм, таблиць та іншої різноманітної цікавої інфографіки та веселих картинок.

Тут же можна подивитися й почитати новини не тільки про погоду, але й про події у світі, відвідати туристичний розділ, завантажити симпатичні віджети.

Основним недоліком є відсутність даних про погоду з інших сайтів.

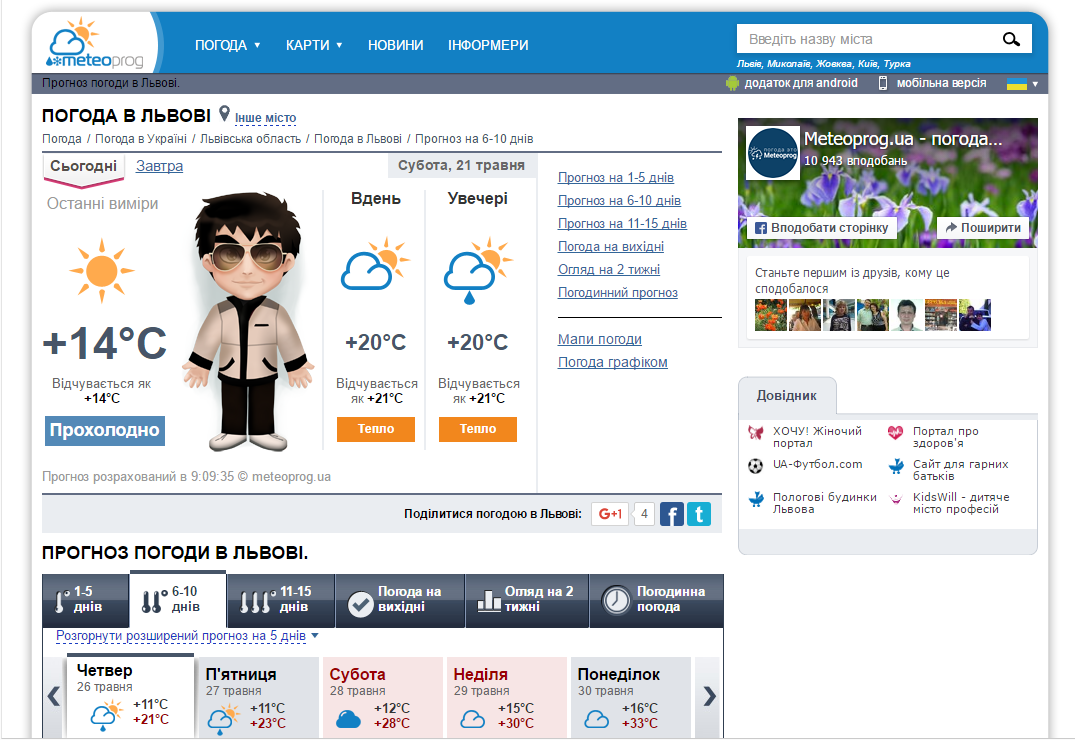


Рис. 1.2.4.1. Вигляд сайту meteoprog.ua

**РОЗДІЛ 2. ПОСТАНОВКА ЗАДАЧІ АНАЛІЗУ ДАНИХ ПРОГНОЗУ ПОГОДИ ТА АНАЛІЗ ВИМОГ НА ПЗ**

**2.1. Постановка задачі**

Завданням бакалаврської роботи є проектування та розробка веб-рішення, яке б дозволяло переглядати та аналізувати прогноз погоди з різних джерел (сайти з погодою).

Результатом виконання роботи має бути сайт на якому можна буде переглянути відповідну інформацію та модуль, який автоматично буде збирати інформацію з різних сайтів з погодою.

**2.2. Обґрунтування технології програмної реалізації**

Середовищем розробки буде **JetBrains** **PhpStorm 10**. Серверна частина – **PHP 5.4**. Клієнтська частина – **JavaScript 5.1**, **HTML 5**, **CSS 3**. База даних – **MySQL 5.5**. Автоматичний запуск скриптів – **Cron**. Сайт буде розміщено на **віртуальному хостингу**, з припаркованим **доменом**.

**JetBrains** **PhpStorm -** комерційне крос-платформове інтегроване середовище розробки для PHP, яке розробляється компанією JetBrains на основі платформи IntelliJ IDEA.

PhpStorm являє собою інтелектуальний редактор для PHP, HTML і JavaScript з можливостями аналізу коду на льоту, запобігання помилок у сирцевому коді і автоматизованими засобами рефакторинга для PHP і JavaScript. Автодоповнення коду в PhpStorm підтримує специфікацію PHP 5.3, 5.4 та 5.5 (сучасні і традиційні проекти), включаючи генератори, співпрограми, простори імен, замикання, типажі і синтаксис коротких масивів. Присутній повноцінний SQL-редактор з можливістю редагування отриманих результатів запитів.

PhpStorm розроблений на основі платформи IntelliJ IDEA, написаної на Java. Користувачі можуть розширити функціональність середовища розробки за рахунок установки плаґінів, розроблених для платформи IntelliJ, або написавши власні плаґіни.

**Переваги PhpStorm**

* Інтелектуальний редактор PHP коду з підсвічуванням синтаксису, автодоповненням коду, розширеними настройками форматування коду, запобігання помилок нальоту.
* Підтримує PHP 7.0, 5.6, 5.5, 5.4 і 5.3, генератори, співпрограми і всі синтаксичні поліпшення.
* PHP рефакторинг, code (re) arranger, детектор дубльованого коду
* Підтримка Vagrant, Composer, вбудований REST клієнт, Command Line Tools, SSH консоль.
* Підтримка фреймворків (MVC view для Symfony2, Yii) і спеціалізовані плагіни для провідних PHP фреймворків (Symfony, Magento, Drupal, Yii, CakePHP і багато інших).
* Візуальний відладчик для PHP додатків, валідація конфігурації відладчика, PHPUnit з покриттям коду (підтримка PHPUnit 5), а також інтеграція з профілювальником.
* HTML, CSS, JavaScript редактор. Налагодження і модульне тестування для JS. Підтримка HTML5, CSS, Sass, SCSS, Less, Stylus, Compass, CoffeeScript, TypeScript, ECMAScript Harmony, Emmet і інших передових технологій веб-розробки.
* Повний набір інструментів для фронтенд-розробки.
* Підтримка стилів коду, вбудовані стилі PSR1 / PSR2, Symfony2, Zend, Drupal та інші.
* Інтеграція з системами управління версіями, включаючи уніфікований інтерфейс.
* Віддалене розгортання додатків і автоматична синхронізація з використанням FTP, SFTP, FTPS і ін.
* Live Edit: зміни в коді можна миттєво переглянути в браузері без перезавантаження сторінки.
* PHP UML.
* Інтеграція з баг-трекера.
* Інструменти роботи з базами даних, SQL редактор.
* Крос-платформенність (Windows, Mac OS X, Linux).

**PHP –** скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважною більшістю хостинг-провайдерів. PHP — проект відкритого програмного забезпечення.

PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, бо браузер отримує готовий html-код. Це є перевага з точки зору безпеки, але погіршує інтерактивність сторінок. Але ніщо не забороняє використовувати PHP для генерування і JavaScript-кодів які виконуються вже на стороні клієнта.

**Переваги PHP**

Головним фактором мови РНР є практичність. РНР повинен надати програмісту кошти для швидкого і ефективного вирішення поставлених завдань. Практичний характер РНР обумовлений п'ятьма важливими характеристиками:

* традиційністю;
* простотою;
* ефективністю;
* безпекою;
* гнучкістю.

Існує ще одна «характеристика», яка робить РНР особливо привабливим: він поширюється безкоштовно. Причому, з відкритими вихідними кодами (Open Source).

**JavaScript (JS)** — динамічна, об'єктно-орієнтованa мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується як частина браузера, що надає можливість коду на стороні клієнта (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки. Мова JavaScript також використовується для програмування на стороні сервера (подібно до таких мов програмування, як Java і C#), розробки ігор, стаціонарних та мобільних додатків, сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite), всередині PDF-документів тощо.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Незважаючи на схожість назв, мови Java та JavaScript є двома різними мовами, що мають відмінну семантику, хоча й мають схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов отриманний «у спадок» від мови С, але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme.

**Можливості JavaScript**

* Створювати нові HTML-теги, видаляти існуючі, змінювати стилі елементів, ховати, показувати елементи і т.п.
* Реагувати на дії відвідувача, обробляти кліки миші, переміщення курсору, натискання на клавіатуру і т.п.
* Посилати запити на сервер і завантажувати дані без перезавантаження сторінки (технологія називається "AJAX").
* Отримувати і встановлювати cookie, запитувати дані, виводити повідомлення.
* Додавати різні ефекти анімації.
* Здійснювати перевірку введення даних в поля форми до відправки на сервер, що в свою чергу знімає додаткове навантаження з сервера.
* Визначати браузер і в залежності від браузера - завантажувати потрібну сторінку, спеціально призначену для цього браузера.

**HTML** (англ. HyperText Markup Language — мова розмітки гіпертекстових документів) — стандартна мова розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.

HTML є похідною мовою від SGML, успадкувавши від неї визначення типу документу та ідеологію структурної розмітки тексту.

Попри те, що HTML — штучна комп'ютерна мова, вона не є мовою програмування.

HTML разом із каскадними таблицями стилів та вбудованими скриптами — це три основні технології побудови веб-сторінок.

HTML впроваджує засоби для:

* створення структурованого документу шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
* отримання інформації із Всесвітньої мережі через гіперпосилання;
* створення інтерактивних форм;
* включення зображень, звуку, відео, та інших об'єктів до тексту.

**Переваги HTML5**

* Можливість застосування більш простого коду, де div замінені удосконаленими елементами.
* Витончені дизайнерські форми, що дозволяють створювати ексклюзивний вигляд веб-сайту. У HTML5 вдалося істотно поліпшити користувальницький інтерфейс, зробити його більш функціональним, простим і інтуїтивно зрозумілим. Валідація форм тепер доступна класичними засобами HTML.
* Можливість використання нових полів введення для різних цілей, включаючи пошук.
* Удосконалена семантика HTML5 дозволяє швидше і простіше розрізняти футер, заголовки, панель навігації. Справа в тому, що для досягнення таких цілей тепер використовуються нові теги для спрощеного призначення основних елементів в розмітці.
* Використання елементів HTML5 дозволяє зробити більш видимим семантичне ядро ​​кожної сторінки. Для реалізації такої задумки відтепер використовуються тільки стандартні коди.
* Покращена зручність. Тепер можливе створення сайтів з HTML5, використовуючи нові елементи, включаючи <canvas>, <audio>, <video>. Виконано поліпшення для інтеграції з вмістом SVG. Таким чином, при використанні нових елементів відпала необхідність в установці таких плагінів, як Flash.
* Музику, графіки, відео та малюнки можна вбудовувати без сторонніх програм, оскільки така функція розроблена в самому браузері. При бажанні можна використовувати стару систему інтеграції аудіо та відеофайлів.
* Покращення структурування документів, для цього використовуються нові теги, наприклад, <header>, <section> і <article>.
* Покращена сприйняття розмітки сторінки, тепер вихідний код буде інтуїтивно зрозумілий.
* За допомогою HTML5 вдалося значно поліпшити клієнтську частину додатків. Для цього розроблені дошки спілкувань, вікі і drag-n-drop.
* Завдяки зменшенню розміру cookies розробники змогли істотно збільшити швидкість відгуку сторінки. У HTML5 цього вдалося домогтися за рахунок використання сховищ даних localStorage і sessionStorage, які стали якісної частковою заміною cookies.
* Geolocation API для визначення місця розташування, отримана інформація може використовуватися в веб-додатках.

**Каскадні таблиці стилів** (англ. Cascading Style Sheets **CSS**) - спеціальна мова, що використовується для опису сторінок, написаних мовами розмітки даних.

Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі.

CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі — сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці веб-сторінок. Головна перевага блочної верстки — розділення змісту сторінки (даних) та їхньої візуальної презентації.

**Нові можливості CSS3**

* background (можливо використовувати фонове зображення і фоновий колір одночасно);
* box-shadow (додавання тіней до блокам);
* text-shadow (додавання тіней до тексту);
* opacity (додавання прозорості блоків);
* gradient (переливання кольору);
* border-radius (округлює кути блоків);
* transition (зміні властивості);
* transition-delay (час затримки зміни властивості);

**MySQL**

MySQL - це система управління реляційними базами даних. У реляційній базі даних дані зберігаються в окремих таблицях, завдяки чому досягається виграш у швидкості й гнучкості. Таблиці зв'язуються між собою за допомогою відносин, завдяки чому забезпечується можливість поєднувати при виконанні запиту дані з декількох таблиць. SQL як частина системи MySQL можна охарактеризувати як мова структурованих запитів, що використовується для доступу до баз даних.

MySQL - це ПЗ з відкритим кодом. Застосовувати його і модифікувати може будь-хто. Таке ПЗ можна отримувати за допомогою Internet і використовувати безкоштовно. При цьому кожен користувач може вивчити вихідний код і змінити його у відповідності зі своїми потребами.

MySQL складається з двох частин: серверної і клієнтської.

Сервер MySQL постійно працює на комп'ютері. Клієнтські програми (наприклад, скрипти PHP) посилають серверу MySQL SQL-запити через механізм сокетів (тобто за допомогою мережевих засобів), сервер їх обробляє і запам'ятовує результат. Тобто скрипт (клієнт) вказує, яку інформацію він хоче отримати від сервера баз даних. Потім сервер баз даних посилає відповідь (результат) клієнтові (скрипту).

Cтруктура MySQL трирівнева: бази даних - таблиці - записи. Бази даних і таблиці MySQL фізично представляються файлами з розширеннями frm, MYD, MYI. Логічно таблиця являє собою сукупність записів. А запису - це сукупність полів різного типу. Ім'я бази даних MySQL унікально в межах системи, а таблиці - в межах бази даних, поля - в межах таблиці. Один сервер MySQL може підтримувати одразу декілька баз даних, доступ до яких може розмежовуватись логіном і паролем.

**Переваги MySQL 5.5**

* Використання за замовчуванням InnoDB.
* Підтримка напівсинхронного (semi-synchronous) механізму реплікації, заснованого на патчах до InnoDB від компанії Google.
* Поліпшення функцій з партіціювання даних. Розширений синтаксис для розбиття великих таблиць на кілька частин, розміщених в різних файлових системах (partitioning). Додані операції RANGE, LIST і метод оптимізації «partition pruning».
* Новий механізм оптимізації вкладених запитів та операцій JOIN.
* Перероблена система внутрішніх блокувань.
* Інтегровані патчі Google з оптимізацією роботи InnoDB на CPU з великою кількістю ядер.

**Віртуальний хостинг** (англ. shared hosting) — вид хостингу, при якому багато веб-сайтів розташовано на одному веб-сервері. Це найекономічніший вид хостингу, який підходить для невеликих проектів. Зазвичай кожен веб-сайт розташований на своєму власному розділі веб-сервера, але вони всі разом користуються одним і тим же програмним забезпеченням.

Віртуальний хостинг як послугу порівнюють і описують, в залежності від кількісних обмежень:

* розмір дискового простору
* кількість місячного трафіку
* кількість сайтів, які можна розмістити в рамках хостингу як однієї послуги
* кількість баз даних, типи і кількість місця під бази даних
* кількість поштових скриньок і FTP-акаунтів
* мови програмування і фреймворки які підтримує хостинг
* вільні ресурси CPU, оперативної пам'яті, які впливають на швидкодію сервера

**cron** — утиліта в операційних системах Unix і Linux, яка дозволяє користувачам виконувати команди або скрипти (групи команд) автоматично в заданий час.

Кожен користувач системи має свій файл завдань «crontab» (cron табулятор), в якому описано, в який час і які команди і програми запускати від імені цього користувача.

Таблиця crontab

Кожний рядок у файлі crontab складається з п'яти колонок, що відокремлюються пробілами чи табуляціями. Колонки задають час виконання (Хвилина, Година, День, Місяць, День тижня), в них може міститися число, список чисел, або символ '\*'. Решта символів в рядку інтерпретуються як виконувана команда та її параметри.

# .---------------- хвилина (0 - 59)

# | .------------- година (0 - 23)

# | | .---------- день місяця (1 - 31)

# | | | .------- місяць (1 - 12) АБО jan,feb,mar,apr ...

# | | | | .---- день тижня (0 - 6) (неділя=0 чи 7)

# | | | | |

\* \* \* \* \* виконувана команда

**2.3. Специфікація програмного засобу**

**2.3.1. Вступ**

*Призначення, мета.*

Ця специфікація вимог до ПЗ описує функціональні та нефункціональні вимоги до системи iWEA . Цей документ призначений для команди, яка реалізовуватиме і перевірятиме коректність роботи системи.

**2.3.2. Загальний опис**

*Перспективи продукту.*

iWEA - система для порівняння прогнозу погоди за різними сайтами у вигляді веб-застосування.

Дана система призначена для того, щоб можна було одночасно дізнатися прогноз від різних джерел і порівняти їх.

**Характеристики продукту**

Повинні бути реалізовані такі функції:

* Автоматичний збір даних про погоду з різних сайтів та занесення їх у базу даних.
* Перегляд погоди на 7 днів.
* Можливість вибору основного джерела та міста.
* Побудова графіків температур.
* Здійснення пошуку по містах.
* Перегляд різниці температур між різними джерелами.
* Побудова матриці відстаней між графіками температур.
* Перегляд класифікації даних сайтів прогнозу погоди.
* Перегляд порівняльного аналізу даних прогнозу погоди різних сайтів.

*Типи користувачів та їх характеристики.*

Доступ до функцій розроблюваної системи не буде поділений на рівні, тому будь-який користувач, який зайде на сайт, матиме доступ до всіх функцій системи.

*Середовище функціонування*

Система буде у вигляді сайту, який розміщено на хостингу з припаркованим доменом(адресою). Він буде доступний через мережу Інтернет.

*Обмеження проектування і реалізації.*

* Час на виконання - 2 місяці.
* Робота без помилок.
* Використання мінімуму ресурсів;

### **2.3.3. Характеристики та функції системи**

* **Автоматичний збір даних про погоду з різних сайтів та занесення їх у базу даних**

Таблиця 2.3.1

*Опис і пріоритет*

|  |  |
| --- | --- |
| Опис | Пріоритет |
| Після налаштувань cron скрипт записуватиме у базу даних інформацію про погоду. | Високий |

Таблиця 2.3.2

*Послідовності дія/відгук*

|  |  |
| --- | --- |
| Дія | Відгук |
| Настання 00:00, 09:00, 15:00, 21:00. | виконується запуск скритпа збору інформації. |

Таблиця 2.3.3

*Функціональні вимоги*

|  |  |
| --- | --- |
| Ідентифікатор | Вимога |
| iWEA\_REQ\_1\_1 | Система повинна брати інформацію з усіх вказаних сайтів. |
| iWEA\_REQ\_1\_2 | Система повинна записувати інформацію для усіх вказаних міст. |
| iWEA\_REQ\_1\_3 | Система повинна записувати прогнози температур на 7 днів. |

* **Перегляд погоди на 7 днів**

Таблиця 2.3.4

*Опис і пріоритет*

|  |  |
| --- | --- |
| Опис | Пріоритет |
| Користувач зайшовши на сайт може переглянути погоду на 7 днів для вибраного міста та з вибраного джерела. | Високий |

Таблиця 2.3.5

*Послідовності дія/відгук*

|  |  |
| --- | --- |
| Дія | Відгук |
| Користувач зайшов на сайт. | Система відобразила головну сторінку. |
| Користувач обрав місто та джерело. | Система відобразила прогноз погоди на 7 днів. |

Таблиця 2.3.6

*Функціональні вимоги*

|  |  |
| --- | --- |
| Ідентифікатор | Вимога |
| iWEA\_REQ\_2\_1 | Система повинна дати можливість вибору джерела. |
| iWEA\_REQ\_2\_2 | Система повинна дати можливість вибору міста. |
| iWEA\_REQ\_2\_3 | Система повинна відображати прогноз для вибраного міста та з обраного джерела. |

* **Можливість вибору основного джерела та міста**

Таблиця 2.3.7

*Опис і пріоритет*

|  |  |
| --- | --- |
| Опис | Пріоритет |
| Користувач зайшовши на сайт може обрати своє місто та джерело з якого беруться дані. | Середній |

Таблиця 2.3.8

*Послідовності дія/відгук*

|  |  |
| --- | --- |
| Дія | Відгук |
| Користувач зайшов на сайт. | Система відобразила головну сторінку. |
| Користувач обрав місто та джерело. | Система відобразила прогноз погоди на 7 днів. |
| Користувач оновив сторінку. | Система відобразила прогноз погоди для обраного міста та джерела вибраних раніше. |

Таблиця 2.3.9

*Функціональні вимоги*

|  |  |
| --- | --- |
| Ідентифікатор | Вимога |
| iWEA\_REQ\_3\_1 | Система повинна зберігати місто та джерело вибрані користувачем при наступних відвідуваннях сайту. |

* **Побудова графіків температур**

Таблиця 2.3.10

*Опис і пріоритет*

|  |  |
| --- | --- |
| Опис | Пріоритет |
| Користувач зайшовши на сайт може переглянути графіки температур для свого міста на 7 днів. | Високий |

Таблиця 2.3.11

*Послідовності дія/відгук*

|  |  |
| --- | --- |
| Дія | Відгук |
| Користувач зайшов на сайт. | Система відобразила головну сторінку. |
| Користувач обрав місто та джерело. | Система відобразила графік температур на 7 днів. |

Таблиця 2.3.12

*Функціональні вимоги*

|  |  |
| --- | --- |
| Ідентифікатор | Вимога |
| iWEA\_REQ\_4\_1 | Графік повинен бути у вигляді ліній. |
| iWEA\_REQ\_4\_2 | Колір ліній повинен відповідати кольору джерела. |
| iWEA\_REQ\_4\_3 | На графіку повинно відображатись лінії максимальної та мінімальної температур. |

* **Здійснення пошуку по містах**

Таблиця 2.3.13

*Опис і пріоритет*

|  |  |
| --- | --- |
| Опис | Пріоритет |
| Користувач зайшовши на сайт може здійснювати пошук по містах, які є на сайті. | Середній |

Таблиця 2.3.14

*Послідовності дія/відгук*

|  |  |
| --- | --- |
| Дія | Відгук |
| Користувач зайшов на сайт. | Система відобразила головну сторінку. |
| Користувач ввів назву міста або його частину у поле «Ваше місцезнаходження» та натиснув кнопку «Шукати». | Система відобразила список міст, які відповідають критеріям пошуку. |

Таблиця 2.3.15

*Функціональні вимоги*

|  |  |
| --- | --- |
| Ідентифікатор | Вимога |
| iWEA\_REQ\_5\_1 | Пошук повинен бути нечутливим до регістру. |
| iWEA\_REQ\_5\_2 | Пошук повинен здійснюватись як по повній назві міста, так і за частиною введеної назви . |
| iWEA\_REQ\_5\_3 | Пошук повинен працювати українською та англійською мовами. |

* **Перегляд різниці температур між різними джерелами**

Таблиця 2.3.16

*Опис і пріоритет*

|  |  |
| --- | --- |
| Опис | Пріоритет |
| Користувач може обрати два джерела і переглянути різницю температур між ними. | Середній |

Таблиця 2.3.17

*Послідовності дія/відгук*

|  |  |
| --- | --- |
| Дія | Відгук |
| Користувач зайшов на сайт. | Система відобразила головну сторінку. |
| Користувач натискає посилання «Аналітика» | Система відобразила сторінку «Аналітика». |
| Користувач обрав два джерела зі списку | Система відобразила різницю температур між ними. |

Таблиця 2.3.18

*Функціональні вимоги*

|  |  |
| --- | --- |
| Ідентифікатор | Вимога |
| iWEA\_REQ\_6\_1 | Система повинна відобразити список всіх джерел. |
| iWEA\_REQ\_6\_2 | Система повинна обмежити вибір користувача тільки двома джерелами. |
| iWEA\_REQ\_6\_3 | Система повинна відображати різницю температур у вигляді графіка. |

* **Побудова матриці відстаней між графіками температур**

Таблиця 2.3.19

*Опис і пріоритет*

|  |  |
| --- | --- |
| Опис | Пріоритет |
| Користувач може переглянути матрицю відстаней між графіками температур | Середній |

Таблиця 2.3.20

*Послідовності дія/відгук*

|  |  |
| --- | --- |
| Дія | Відгук |
| Користувач зайшов на сайт. | Система відобразила головну сторінку. |
| Користувач натискає посилання «Аналітика» | Система відобразила сторінку «Аналітика», на якій можна переглянути матрицю відстаней. |

Таблиця 2.3.21

*Функціональні вимоги*

|  |  |
| --- | --- |
| Ідентифікатор | Вимога |
| iWEA\_REQ\_7\_1 | Система повинна будувати матрицю для всіх джерел. |
| iWEA\_REQ\_7\_2 | Матриця повинна бути симетричною. |
| iWEA\_REQ\_7\_3 | Матриця повинна бути квадратною. |
| iWEA\_REQ\_7\_4 | Матриця повинна містити нулі на головній діагоналі. |

* **Перегляд класифікації даних сайтів прогнозу погоди**

Таблиця 2.3.22

*Опис і пріоритет*

|  |  |
| --- | --- |
| Опис | Пріоритет |
| Користувач може переглянути класифікацію даних сайтів прогнозу погоди та графіки відповідних груп | Високий |

Таблиця 2.3.23

*Послідовності дія/відгук*

|  |  |
| --- | --- |
| Дія | Відгук |
| Користувач зайшов на сайт. | Система відобразила головну сторінку. |
| Користувач натискає посилання «Аналітика» | Система відобразила сторінку «Аналітика», на якій можна переглянути класифікацію та графіки. |

Таблиця 2.3.24

*Функціональні вимоги*

|  |  |
| --- | --- |
| Ідентифікатор | Вимога |
| iWEA\_REQ\_8\_1 | Система повинна класифікувати дані на основі матриці відстаней між графіками температур. |
| iWEA\_REQ\_8\_2 | Система повинна виділити дві групи. |
| iWEA\_REQ\_8\_3 | Система повинна побудувати графіки для цих двох груп. |

* **Перегляд порівняльного аналізу даних прогнозу погоди різних сайтів**

Таблиця 2.3.25

*Опис і пріоритет*

|  |  |
| --- | --- |
| Опис | Пріоритет |
| Користувач може здійснювати порівняльний аналізу даних прогнозу погоди | Високий |

Таблиця 2.3.26

*Послідовності дія/відгук*

|  |  |
| --- | --- |
| Дія | Відгук |
| Користувач зайшов на сайт. | Система відобразила головну сторінку. |
| Користувач натискає посилання «Аналітика» | Система відобразила сторінку «Аналітика». |
| Користувач задає інтервал для аналізу | Система запам’ятовує вказаний інтервал. |
| Користувач натискає кнопку «Ок» | Система аналізує дані прогнозу погоди та показує похибку кожного джерела. |

Таблиця 2.3.27

*Функціональні вимоги*

|  |  |
| --- | --- |
| Ідентифікатор | Вимога |
| iWEA\_REQ\_9\_1 | Система повинна аналізувати всі джерела. |
| iWEA\_REQ\_9\_2 | Інтервал повинен бути від 3 до 7 днів. |
| iWEA\_REQ\_9\_3 | Система повинна обчислювати похибку з точністю до сотих. |

**РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА АРХІТЕКТУРА СИСТЕМИ ДЛЯ ПОРІВНЯННЯ ПРОГНОЗУ ПОГОДИ ЗА РІЗНИМИ САЙТАМИ**

**3.1. Структура системи**

Основна мета роботи – розроблення веб-застосування для порівняння прогнозу погоди за різними сайтами.

На концептуальному рівні систему потрібно розділити на декілька складових – модулів, кожен з яких відповідатиме за свій обсяг робіт, що призведе до покращення розуміння системи.

Отже, система буде складатись з:

* модуль збору інформації;
* веб-сайт.

**Модуль збору інформації**

Даний модуль буде виконувати роботу по збору інформації про погоду з різних сайтів та записуватиме її у базу даних. Це буде окремий php-скрипт, який буде запускатись 4 рази на день по графіку за допомогою утиліти cron.

**Веб-сайт**

Сукупність веб-сторінок, на яких можна буде переглядати та аналізувати прогноз погоди з різних джерел.

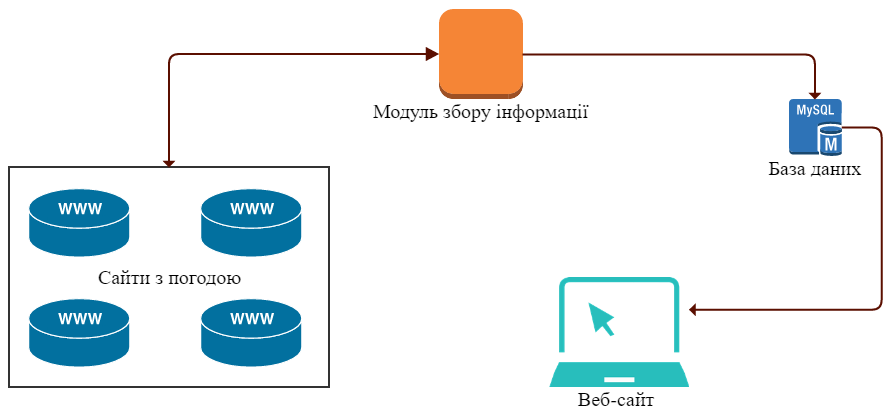


Рис. 3.1.1. Структура системи.

Діаграму компонентів розміщено у додатку А.

**3.2. Архітектура системи**

Для побудови архітектури веб-застосування використовувався шаблон MVC.

**3.2.1. Архітектурний шаблон MVC**

Архітектурний шаблон Модель-Вид-Контролер (MVC) поділяє програму на три частини. У тріаді до обов'язків компоненту Модель (Model) входить зберігання даних і забезпечення інтерфейсу до них. Вигляд (View) відповідальний за представлення цих даних користувачеві. Контролер (Controller) керує компонентами, отримує сигнали у вигляді реакції на дії користувача, і повідомляє про зміни компоненту Модель. Така внутрішня структура в цілому поділяє систему на самостійні частини і розподіляє відповідальність між різними компонентами.

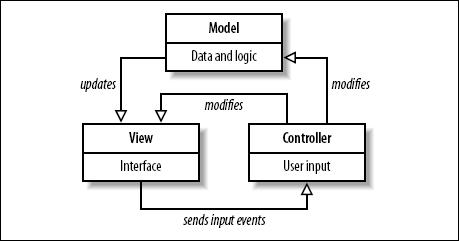


Рис. 3.2.1.1. MVC – взаємодія компонентів.

MVC поділяє цю частину системи на три самостійні частини: введення даних, компонент обробки даних і виведення інформації. Модель, як вже було відмічено, інкапсулює ядро даних і основний функціонал з їх обробки. Також компонент Модель не залежить від процесу введення або виведення даних. Компонент виводу Вигляд може мати декілька взаємопов'язаних областей, наприклад, різні таблиці і поля форм, в яких відображається інформація. У функції Контролера входить моніторинг за подіями, що виникають в результаті дій користувача (зміна положення курсора миші, натиснення кнопки або введення даних в текстове поле).

Зареєстровані події транслюються в різні запити, що спрямовуються компонентам Моделі або об'єктам, відповідальним за відображення даних. Відокремлення моделі від вигляду даних дозволяє незалежно використовувати різні компоненти для відображення інформації. Таким чином, якщо користувач через Контролер внесе зміни до Моделі даних, то інформація, подана одним або декількома візуальними компонентами, буде автоматично відкоригована відповідно до змін, що відбулися.

**Причини використання MVC**

Мабуть, головною причиною використання моделі стала необхідність поділу коду та подання або прийнято ще говорити: "відділення бізнес логіки додатка від дизайну". Дійсно, проблема поділу коду та відображення стає перед кожним розробником. Якщо спочатку про це не задуматися, то через деякий час підтримувати такий код стане просто нереально.

У сфері веб-застосувань, важливим є зовнішній вигляд web-додатків. Тому щоб додаток залишався популярним необхідно підтримувати його зовнішній вигляд в актуальному стані. З використанням шаблону MVC дизайн буде відділений окремим компонентом і його зміна або заміна не потребуватиме великих зусиль і переписування бізнес-логіки.

Важливий плюс шаблону MVC ще в тому, що він відразу адаптований для роботи в команді. Розробник, який відповідає за бізнес-логіку може спокійно працювати над своїми запитами і алгоритмами, а фронтенд-розробник може в цьому час розробляти нові представлення. У підсумку, кожен займається своєю справою, і найголовніше не заважає іншим.

### **3.2.2. Опис класів**

**AutoLoader** – автозагрузчик класів PHP

Більшість розробників об'єктно-орієнтованих додатків використовують таку угоду іменування файлів, в якому кожен клас зберігається в окремо створеному для нього файлі. Однією з найбільш при надокучливих деталей є необхідність писати на початку кожного скрипта довгий список підвантажуваних файлів (include, require).

У PHP 5 це робити не обов'язково. Можна визначити функцію \_\_autoload(), яка буде автоматично викликана при використанні раніше невизначеного класу або інтерфейсу. Виклик цієї функції - останній шанс для інтерпретатора завантажити клас перш, ніж він закінчить виконання скрипта з помилкою.

Саме в цьому класі реалізований метод \_\_autoload() з нього і починається виконання.

**Config** – клас для доступу до налаштувань.

**MMySQLi** – клас для роботи з базою даних MySQL.

**Model** – клас, в якому знаходяться основні методи для роботи з даними.

**Controller** – клас, який в залежності від посилання викликає відповідний метод з моделі та повідомляє, які дані потрібно оновити на представленні.

**Template** – клас для генерації сторінок.

**ISiteHelper** – інтрерфейс, який наслідують всі джерела, містить методи: buildQuery() – будує запит (url) за яким потрібно звернутись, щоб отримати дані про погоду, addWeatherData() – тут здійснюється обробка отриманих даних та занесення їх у базу даних.

**AerisWeather, Interia, Meteoprog, OpenWeatherMap, SinoptikUa, TheDarkSkyCompany, WorldWeatherOnline** – класи, джерел. Кожне джерело представляється окремим класом, назва якого співпадає з назвою джерела. Для того щоб додати нове джерело необхідно додати новий клас, який реалізовуватиме інтерфейс ISiteHelper.

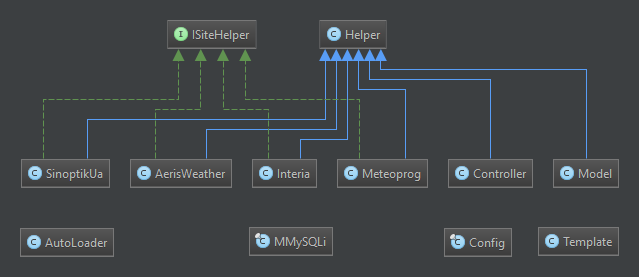


Рис. 3.2.2.1. Спрощена діаграма класів.

Повна діаграма класів наведена у додатку Б.

**3.3. Проектування бази даних**

База даних складатиметься з таких таблиць:

* City
* Site
* Weather
* User
* Setting

Таблиця 3.2.1

Структура таблиці City

|  |  |
| --- | --- |
| Стовпець | Опис |
| id | Ідентифікатор, первинний ключ |
| country | Країна, в якій знаходиться місто |

Продовження таблиці 3.2.1

|  |  |
| --- | --- |
| Стовпець | Опис |
| name | Назва міста |
| lat | Географічна широта |
| lon | Географічна довгота |
| name\_iso | Міжнародна назва міста |

Таблиця 3.2.2

Структура таблиці Site

|  |  |
| --- | --- |
| Стовпець | Опис |
| Id | Ідентифікатор, первинний ключ |
| country | Країна |
| name | Назва джерела |
| url | Web-сайт джерела |
| image\_url | Логотип джерела |
| color | Колір ліній на графіках |

Таблиця 3.2.3

Структура таблиці Weather

|  |  |
| --- | --- |
| Стовпець | Опис |
| Id | Ідентифікатор, первинний ключ |
| site\_id | Ідентифікатор сайту, вторинний ключ |
| city\_id | Ідентифікатор міста, вторинний ключ |
| date | Дата прогнозу |
| date\_write | Поточна дата |
| min\_temp | Мінімальна температура |
| max\_temp | Максимальна температура |

Таблиця 3.2.4

Структура таблиці User

|  |  |
| --- | --- |
| Стовпець | Опис |
| user\_id | Ідентифікатор, первинний ключ |
| pass | Хеш паролю |
| name | Ім’я |
| email | E-mail |
| date | Дата реєстрації |

Таблиця 3.2.5

Структура таблиці Setting

|  |  |
| --- | --- |
| Стовпець | Опис |
| Id | Ідентифікатор, первинний ключ |
| user\_id | Ідентифікатор користувача, вторинний ключ |
| setting | Json-стрічка з налаштуваннями |

Схема бази даних знаходиться у додатку В. DDL-скрипт створення бази даних знаходиться у додатку Д.

**3.4. Проектування графічного інтерфейсу**

Створення графічного інтерфейсу користувача — дуже важливий етап в процесі розроблення програмного продукту. Недооцінювати його значущість не тільки недалекоглядно, але й небезпечно. Створення дизайну і програмна розробка інтерфейсу повинні вестися з урахуванням безлічі нюансів. Нерідко, саме вони зумовлять успіх всього проекту. Сьогодні ринок програмних продуктів буквально переповнений найрізноманітнішими рішеннями. І програмний інтерфейс — саме та складова, за рахунок якої ви можете виділити власний програмний продукт.

При розробці інтерфейсу необхідно керуватися наступними рекомендацій:

* Стандартизація. Рекомендується використовувати стандартними, перевіреними багатьма програмістами і користувачами інтерфейсні рішення. Під рішеннями маються на увазі дизайн форм, розподіл елементів керування у формах, їхнє взаємне розташування, значки на кнопках керування, назви команд меню.
* Зручність і простота роботи. Інтерфейс повинний бути інтуїтивно зрозумілим. Бажано, щоб усі дії легко запам'ятовувалися і не вимагали стомлюючих процедур: виконання додаткових команд, зайвих натискань на кнопки, виклику проміжних діалогових вікон.
* Зовнішній дизайн. Не можливо, щоб інтерфейс стомлював зір. Він повинний бути розрахований на тривалу роботу користувача з додатком протягом дня.
* Неперевантаженість форм. Форми повинні бути оптимально завантажені елементами керування. При необхідності можливо використовувати вкладки чи додаткові сторінки форм.
* Угрупування. Елементи керування у формі необхідно групувати за змістом, використовуючи елементи угрупування: рамки, фрейми.
* Розрідженість об'єктів форм. Елементи керування варто розташовувати на деякій відстані, а не ліпити один на одного; для виділення елементів керування можливо організувати порожні простори у формі.

Три принципи розробки користувальницького інтерфейсу :

* контроль користувачем інтерфейсу;
* зменшення завантаження пам'яті користувача;
* послідовність користувальницького інтерфейсу.

Проектування графічного інтерфейсу користувача здійснювалось з урахуванням специфіки задачі та згідно рекомендацій та принципів.

Головна сторінка (рис. 3.4.1) – перша сторінка, яка відкривається при заході на сайт. У верхній частині розміщено логотип, назву сайту та меню. Далі розміщено пошукове поле для міст, випадаючий список для вибору джерел. В основній частині розміщено таблицю для відображення прогнозу погоди та графіки температур. Внизу розміщено назву автора.

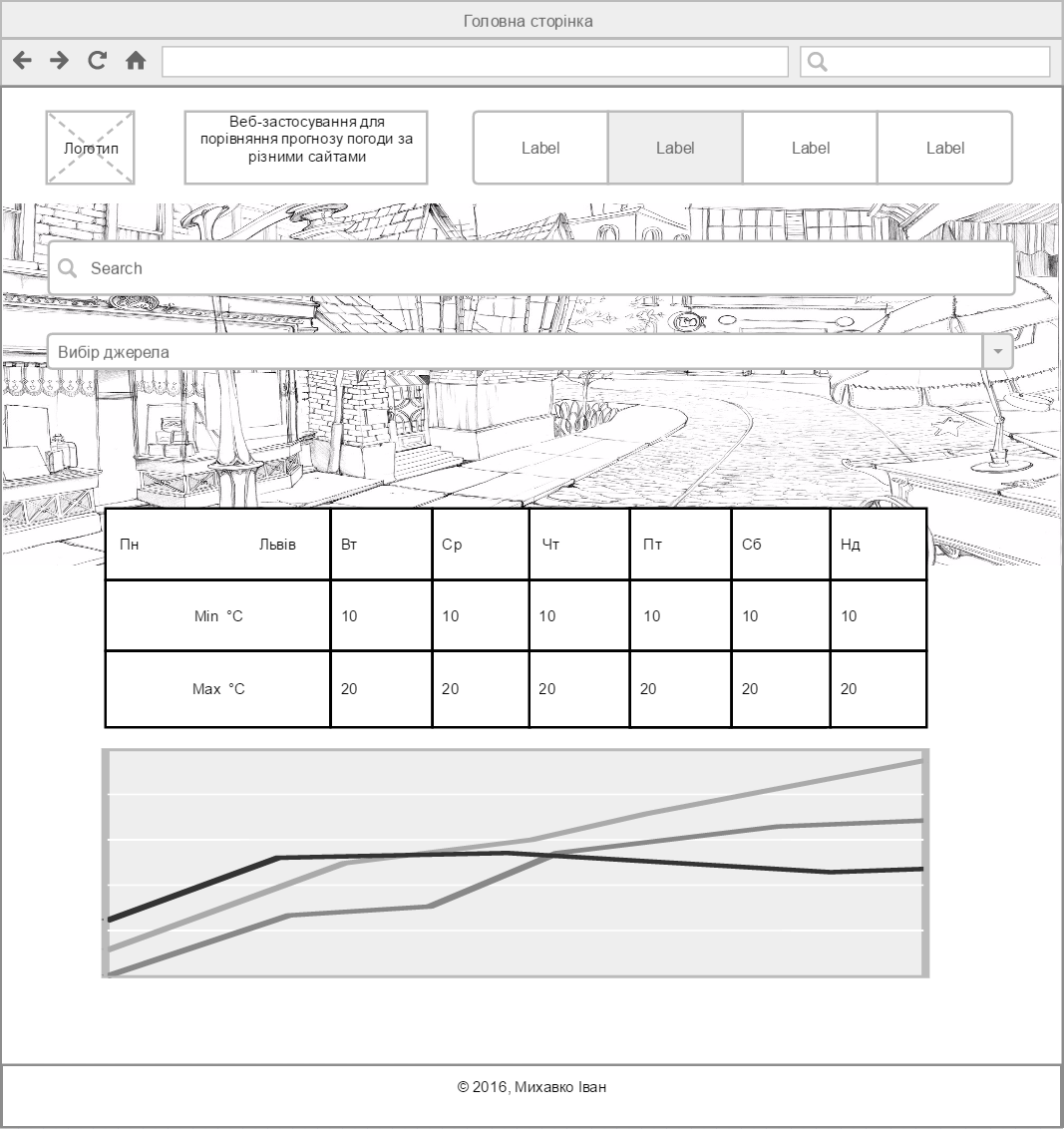


Рис. 3.4.1. Головна сторінка.

**РОЗДІЛ 4. РОЗДІЛ РЕАЛІЗАЦІЇ ТА ТЕСТУВАННЯ**

**4.1. Реалізація системи**

Сайт розміщено на хостингу з установленим сервером Apache 2.3, базою даних MySQL 5.5. До хостингу припарковано домен [**http://iwea.ml**](http://iwea.ml) .

Вигляд головної сторінки наведено у додатку Ж.

**4.1.1. Автоматичний збір даних про погоду з різних сайтів та занесення їх у базу даних.**

Дана функція реалізована через утиліту cron, яка запускає відповідний скрипт 4 рази на день.

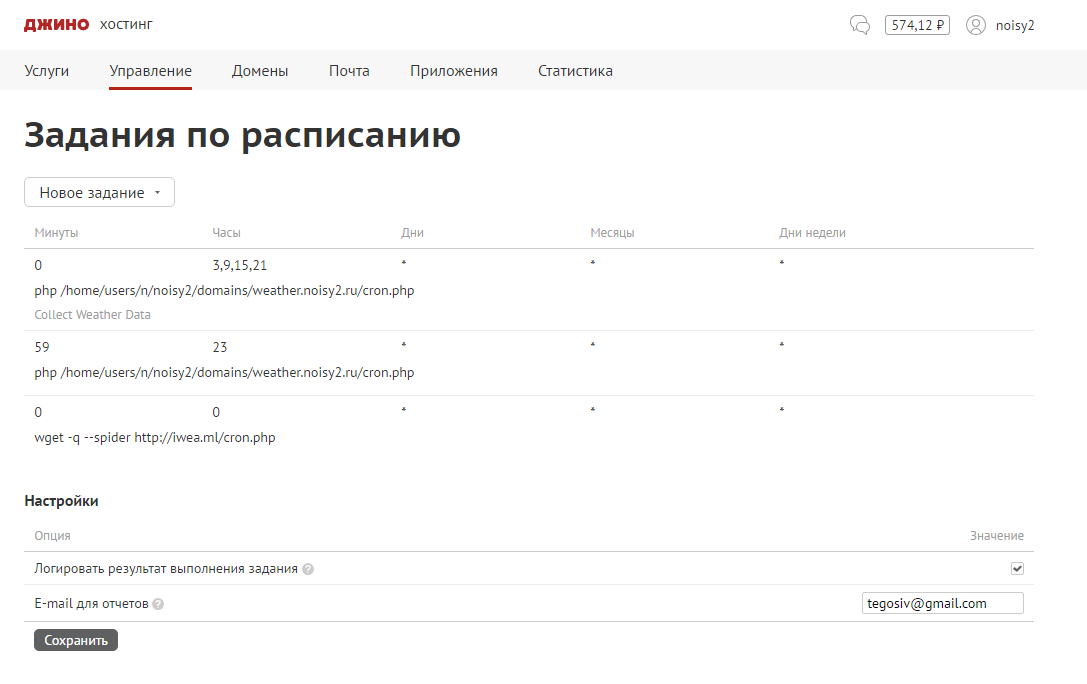


Рис. 4.1.1.1. Налаштування cron на хостигну.

**4.1.2. Перегляд погоди на 7 днів**

Для перегляду погоди необхідно обрати потрібне місто (рис. 4.1.2.1) та джерело (рис. 4.1.2.2) з якого братимуться дані про погоду.

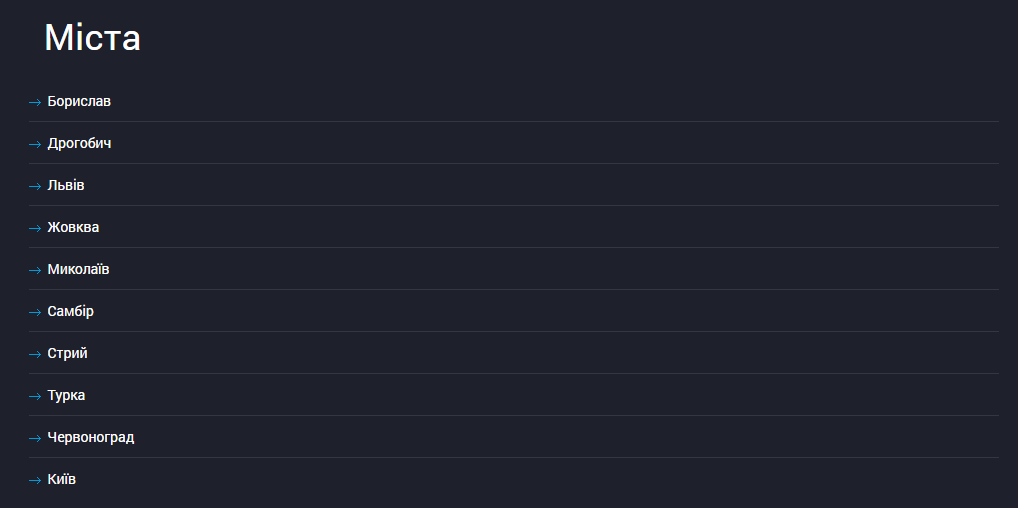


Рис. 4.1.2.1. Вибір міста.

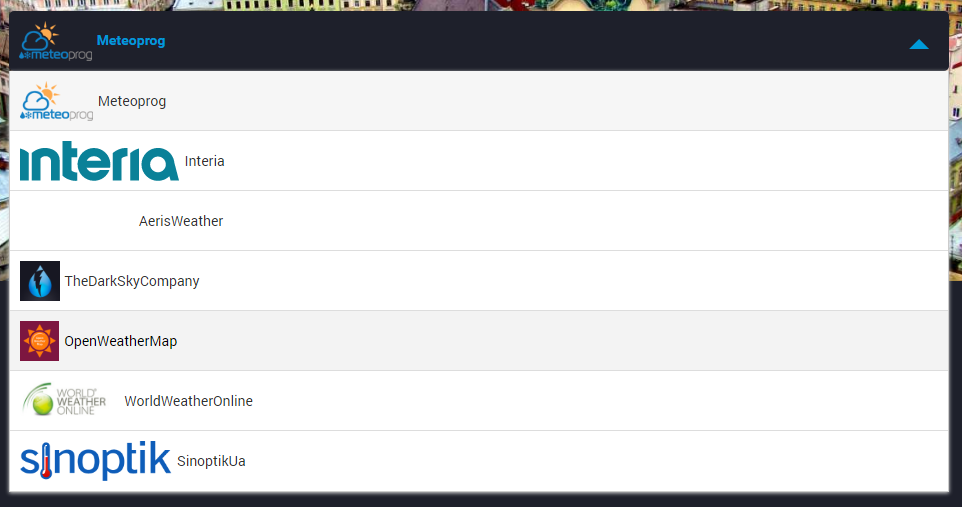


Рис. 4.1.2.2. Вибір джерела.

Після цього з’явиться прогноз погоди на 7 днів (рис. 4.1.2.3).



Рис. 4.1.2.3. Прогноз погоди на 7 днів.

**4.1.3. Побудова графіків температур**

Після вибору міста та джерела можна також подивитись графіки максимальних та мінімальних температур.

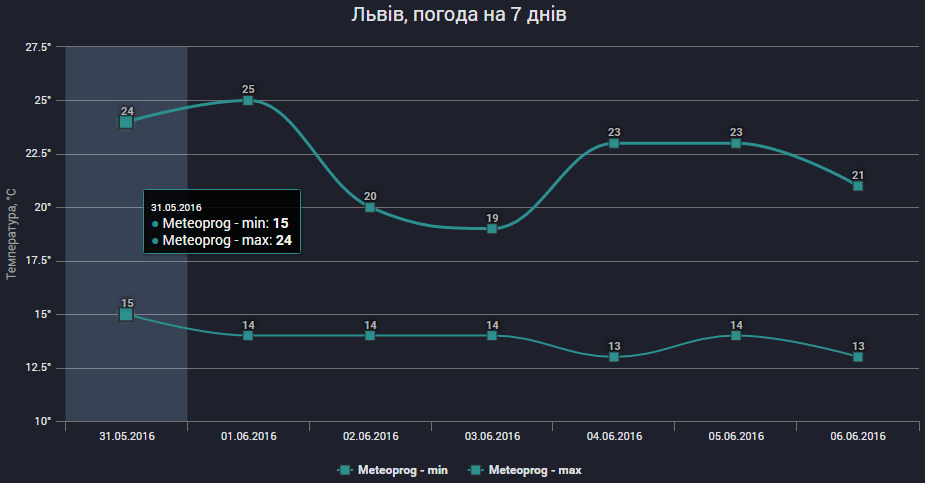


Рис. 4.1.3.1. Графіки мінімальних та максимальних температур.

Графіки температур з усіх джерел наведено у додатку З.

**4.1.4. Побудова матриці відстаней між графіками температур**

Для проведення аналізу та класифікації даних з сайтів про погоду, необхідно побудувати матрицю відстаней між графіками температур.

Матриця відстаней – це квадратна матриця типу «об'єкт-об'єкт», що містить в якості елементів відстані між об'єктами в метричному просторі.

Властивості матриці є відображенням властивостей самих відстаней:

* симетричність щодо діагоналі, тобто dij = dji;
* відображення властивості тотожності відстані dij = 0 ⇔ i = j в матриці відстаней проявляється в наявності 0 по діагоналі матриці, так як відстань об'єкта з самим собою очевидно дорівнює 0, а також в наявності нульових значень для абсолютно подібних об'єктів;
* значення відстаней в матриці завжди невід'ємні dij ⩾ 0
* нерівність трикутника приймає форму dij + djk ⩾ dik для всіх i, j і k.

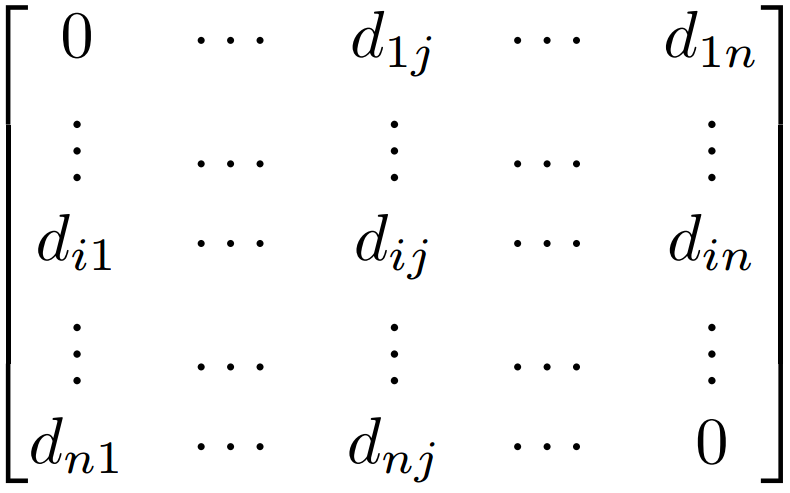


Рис. 4.1.4.1. Загальний вигляд матриці відстаней.

Відстань обчислюється за формулою:

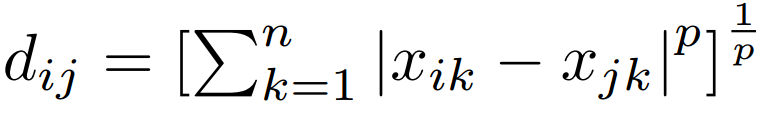




Рис. 4.1.4.2. Побудована матриця відстаней між графіками температур.

**4.1.5. Перегляд класифікації даних сайтів прогнозу погоди**

Класифікація здійснюється на основі матриці відстаней між графіками температур. В даному випадку матрицю відстаней розглянемо як матрицю суміжності повного графа на 7 вершин. Виділимо 2 групи: 1 група – 4 сайти, друга – 3 сайти, тобто два підграфи з 4 і 3 вершинами. Підграфи виділяємо, беручи ті вершини, ребра між якими, мають найменшу вагу.

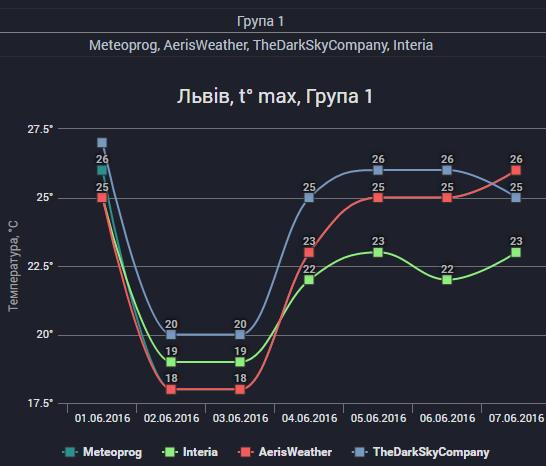


Рис. 4.1.5.1. Перша виділена група.

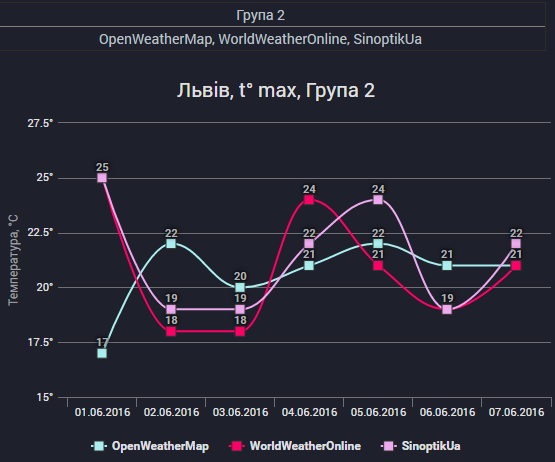


Рис. 4.1.5.2. Друга виділена група.

**4.1.6. Перегляд порівняльного аналізу даних прогнозу погоди різних сайтів**

Для того, щоб побачити, яке саме джерело дає найкращі прогнози, потрібно порівнювати його прогноз в минулому на поточний час.



Рис. 4.1.6.1. Порівняльний аналіз даних прогнозу погоди різних сайтів.

**4.2. Звіт про тестування**

**4.2.1. Вступ**

Документ відображає звіт про розроблені тести, їх результати та дозволяє оцінити якість програмного продукту. У звіті наведені розроблені тести для функціонального тестування та тестування продуктивності, а також результати проходження цих тестів.

**4.2.2. Функціональне тестування**

**4.2.2.1. Функціональні тестові випадки**

Розроблено 5 тестових випадків. Таблиця 4.2.2.1.1. відображає розподіл функціональних тестових випадків і наборів тестових даних для цих випадків за варіантами використання.

Таблиця 4.2.2.1.1

Розподіл варіантів використання для функціонального тестування

|  |  |  |
| --- | --- | --- |
| **Варіанти використання** | **Тестові випадки** | **Тестові дані** |
| Тестування роботи сайту | 1 | 20 |
| Тестування відображення погоди | 2 | 20 |
| Тестування аналітики | 2 | 20 |
| **Загалом** | **5** | **60** |

**4.2.2.2. Пройдені тести, їх результати**

Функціональність програми ми тестуємо перевіряючи чи виконуються тестові випадки.

Таблиця 4.2.2.2.1

*ID Test-1 :* Запуск програми

|  |  |
| --- | --- |
| *Summary:* Перевіряємо чи запускається сайт | |
|  | |
| *Steps:* | *Expected Results:* |
| 1. Перейти за адресою сайту(iwea.ml) | 1. Відображення головної сторінки |

Таблиця 4.2.2.2.2

*ID Test-2 :* Здійснення пошуку

|  |  |
| --- | --- |
| *Summary:* Перевіряємо працює пошук | |
| *Pre-condition:* Виконано тестовий випадок TEST-1. | |
| *Steps:* | *Expected Results:* |
| 1. Перейти за адресою сайту(iwea.ml) 2. Ввести частину назви міста 3. Клік по кнопці «Шукати» | 1. Відображення головної сторінки 2. Назва залишається у полі введення 3. Відображаються міста, що відповідають критеріям пошуку. |

Таблиця 4.2.2.2.3

*ID Test-3 :* Вибір джерела

|  |  |
| --- | --- |
| *Summary:* Перевіряємо чи працює вибір джерела та його збереження | |
| *Pre-condition:* Виконано тестовий випадок TEST-1. | |
| *Steps:* | *Expected Results:* |
| 1. Перейти за адресою сайту(iwea.ml) 2. Вибрати джерело з випадаючого списку 3. Оновити сторінку | 1. Відображення головної сторінки 2. Джерело виділилось 3. Джерело залишилось виділеним. |

Таблиця 4.2.2.2.4

*ID Test-4 :* Різниця температур

|  |  |
| --- | --- |
| *Summary:* Перевіряємо роботу відображення різниці температур | |
| *Pre-condition:* Виконано тестовий випадок TEST-1. | |
| *Steps:* | *Expected Results:* |
| 1. Перейти за адресою сайту(iwea.ml) 2. Клік по меню «Аналітика» 3. Вибір одного джерела 4. Вибір другого джерела | 1. Відображення головної сторінки 2. Відображення сторінки «Аналітика» 3. Нічого не відбувається 4. Будується графік різниці температур для двох обраних джерел. |

Таблиця 4.2.2.2.5

*ID Test-5 :* Порівняльний аналіз даних прогнозу погоди

|  |  |
| --- | --- |
| *Summary:* Перевіряємо чи працює відображення порівняльний аналіз | |
| *Pre-condition:* Виконано тестовий випадок TEST-1. | |
| *Steps:* | *Expected Results:* |
| 1. Перейти за адресою сайту(iwea.ml) 2. Клік по меню «Аналітика» 3. Вибір інтервалу 4. Клік по кнопці «Ок» | 1. Відображення головної сторінки 2. Відображення сторінки «Аналітика» 3. Інтервал зберігається 4. Будується таблиця з температурами та список з похибками. |

**4.2.2.3. Підсумки тестування**

Кількість тестових випадків 5, з них успішних 100%. Усі тестові випадки пройшли успішно, тестування вважається успішним.

**4.2.3. Тестування продуктивності**

Тестування продуктивності в програмній інженерії – це тестування, яке проводиться з ціллю визначення, як швидко працює програма або її частина під деяким навантаженням.

**4.2.3.1. WAPT**

**WAPT** (Web Application Performance Testing) є засобом для тестування продуктивності, навантаження та стресового тестування веб-сайтів і інтернет аплікацій з веб інтерфейсом. Цей засіб можна отримати за адресою http://www.loadtestingtool.com/

Характеристики продуктивності Веб сайту і Інтернет аплікацій напряму залежить від якості сервісу та реалізації завдання. Для гарантування необхідної продуктивності Веб аплікації необхідно тестувати Веб системи при навантаженні перед їх запуском у використання.

Навантажувальне тестування є мірою повноти здатності Веб аплікації обслуговувати одночасно певну кількість користувачів і/чи транзакцій при дотриманні адекватного часу відгуку. Таке тестування дає змогу розробникам знайти і оцінити вузькі місця у Веб інфраструктурі.

**4.2.3.2. Результати тестування**

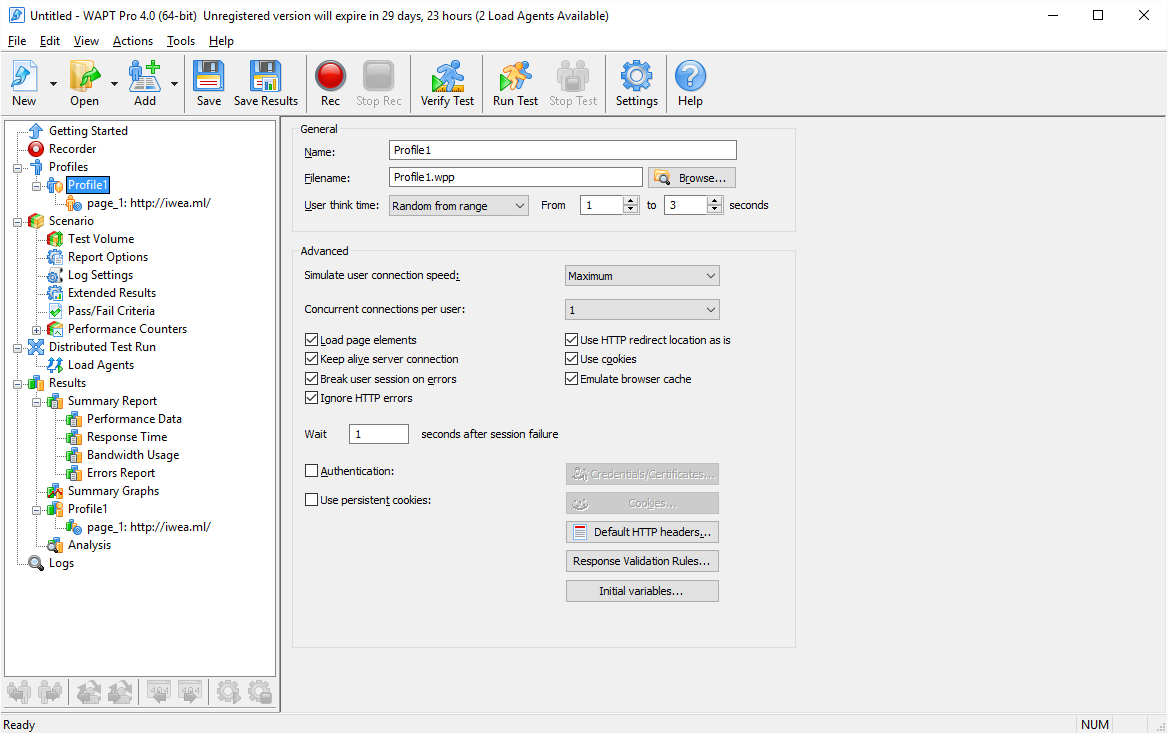


Рис. 4.2.3.2.1. Налаштування профілю.

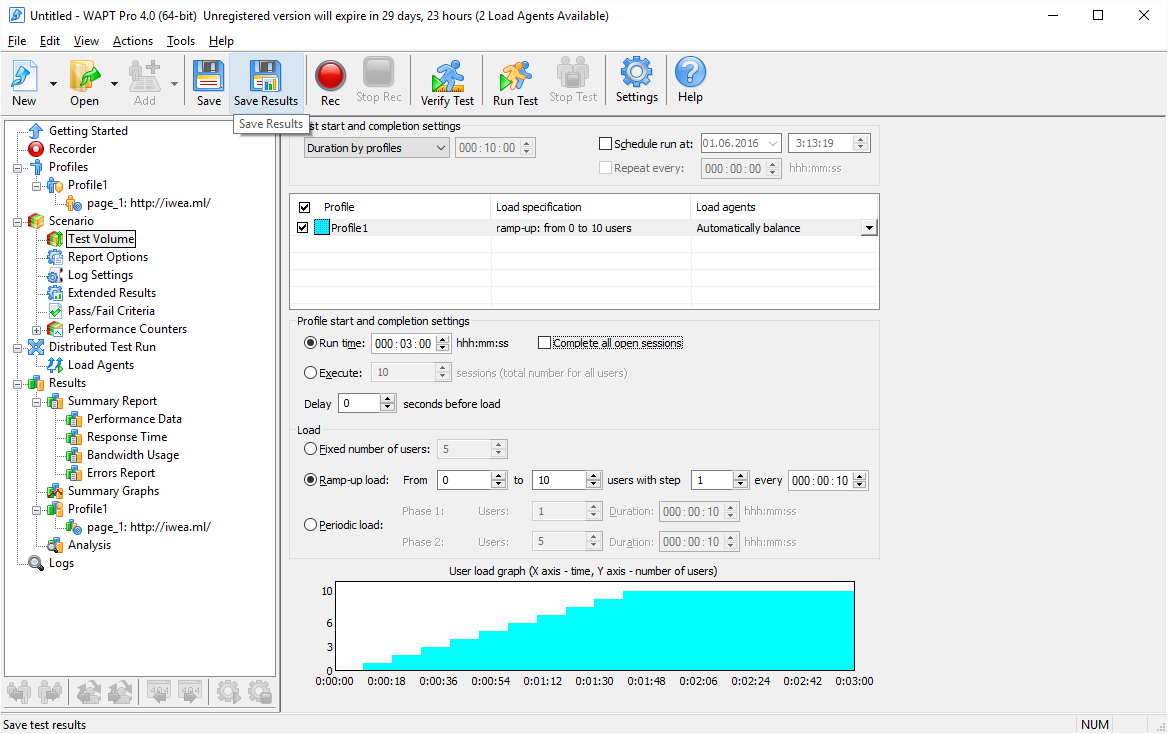


Рис. 4.2.3.2.2. Інформація про рівень навантаження.

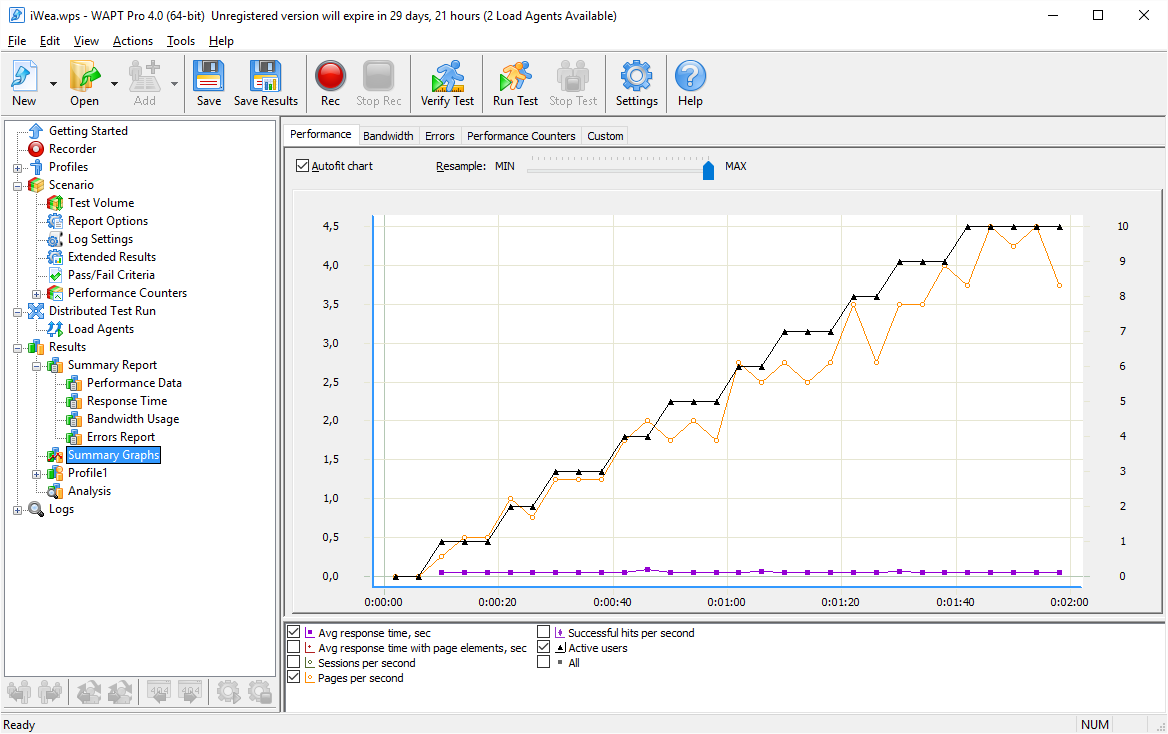


Рис. 4.2.3.2.3. Виконання тестування.

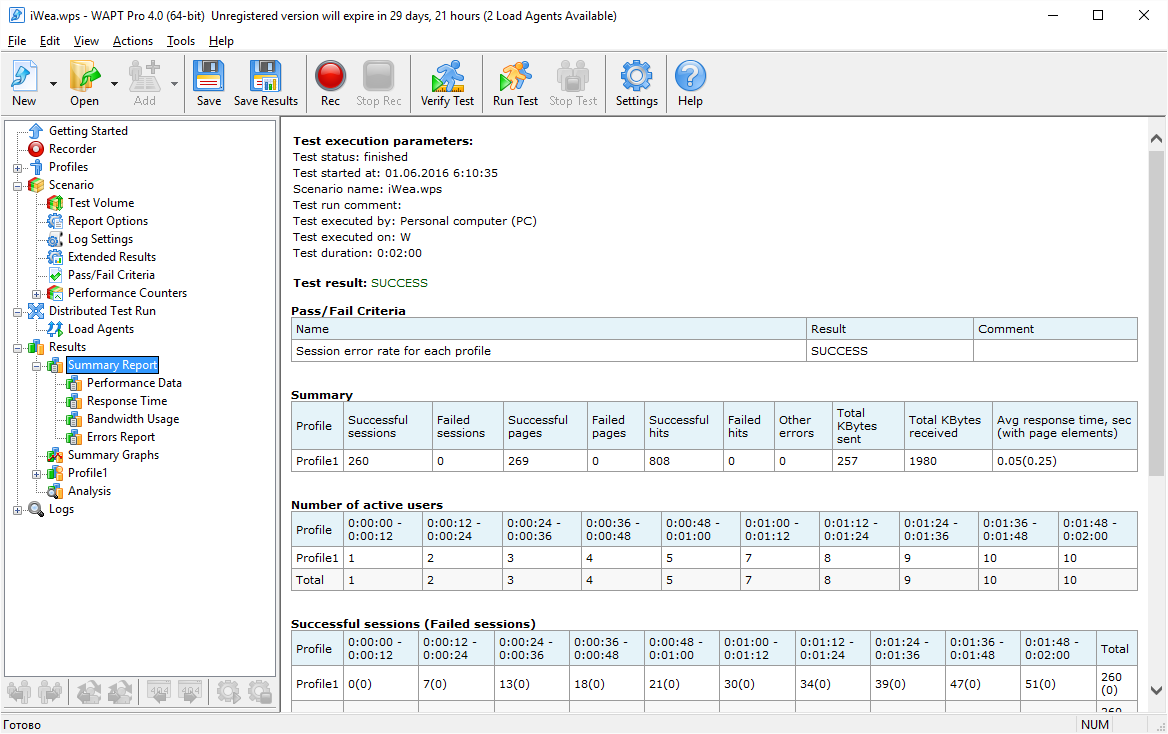


Рис. 4.2.3.2.4. Результат про проходження тесту.

**4.2.4. Критерій успіх/провал проекту**

Проект вважатиметься успішним, якщо покриття тестами становитиме не менше 80%, усі тести дають позитивний результат. Так, як умови виконано, проект вважається успішним.

**РОЗДІЛ 5. ЕКОНОМІЧНА ЧАСТИНА**

**5.1. Економічна характеристика проектного рішення**

Завданням бакалаврської роботи є проектування та розробка веб-рішення, яке б дозволяло переглядати та аналізувати прогноз погоди з різних джерел (сайти з погодою).

Результатом виконання роботи має бути сайт на якому можна буде переглянути відповідну інформацію та модуль, який автоматично буде збирати інформацію з різних сайтів з погодою.

До основних характеристик програмного продукту можна віднести:

* Автоматичний збір даних про погоду з різних сайтів та занесення їх у базу даних.
* Перегляд погоди на 7 днів.
* Побудова графіків температур.
* Здійснення пошуку по містах.
* Перегляд різниці температур між різними джерелами.
* Перегляд класифікації даних сайтів прогнозу погоди.
* Перегляд порівняльного аналізу даних прогнозу погоди різних сайтів.

**5.2. Інформаційне забезпечення та формування гіпотези щодо потреби розроблення товару**

*Стан ринку програмного забезпечення та перспектив його розвитку*

Ринок програмного забезпечення для погоди характеризується великою кількістю програмних продуктів більшістю з яких є мережеві системи, які крім погоди дають ще і інші сервіси.

*Характеристики аналогічного продукту на ринку та процесу його продажу*

Типовий погодний портал пропонує ряд послуг своїм користувачам, серед яких можуть бути:

- перегляд погоди на тиждень та довгостроковий прогноз;

- можливість перегляду погоди у різних поданнях (графіки, діаграми);

- надання погодних інформерів;

- карти погоди й опадів;

- публікація новин, статей та відеосюжетів про погоду.

*За методами монетизації погодні портали можна розділити на три групи:*

- комерційні проект, що пропонують доступ до своїх ресурсів за плату.

- комерційні проект, що надають умовно безкоштовний доступ з показом реклами.

- не комерційні проекти .

*Основні споживачі та їхні потреб;*

Основними споживачами погодного порталу є будь-які користувачі, для яких має значення погода.

*Основні конкуренти*

Схожим за функціональними можливостями є сайти opogode.ua, gismeteo.ua, sinoptik.ua, meteoprog.ua.

Незважаючи на величезний вибір інтернет-сайтів про погоду, лише деякі з них використовують власні прогнози. Проте саме такого сайту, який би реалізував поставлену задачу не існує.

**5.3. Оцінювання та аналіз факторів зовнішнього та внутрішнього середовищ**

Фактори зовнішнього середовища оцінюються за шкалою [-5;5], де межі шкали відображають максимальний негативний та позитивний вплив факторів на організацію, 0 показує, що фактор впливає на організацію нейтрально.

Фактори зовнішнього середовища оцінюються за шкалою [0;5], де 0 демонструє відсутність, нерозвинутість чи катастрофічний стан фактору внутрішнього середовища, оцінка 5 демонструє високий рівень розвитку даного фактору.

Сума ваг всіх факторів становить одиницю, тобто рівень вагомості для кожного фактору визначається за допомогою коефіцієнтів. Зважений рівень впливу факторів розраховується як добуток впливу фактору у балах та рівня вагомості. Результати експертних оцінок впливу факторів зовнішнього та внутрішнього середовища на організацію наведено у табл. 5.1

*Таблиця 5.3.1*

Результати експертного оцінювання впливу факторів зовнішнього та внутрішнього середовищ

|  |  |  |  |
| --- | --- | --- | --- |
| Фактори | Середня експертна оцінка, бали | Середня вагомість факторів | Зважений рівень впливу, бали |
| 1 | 2 | 3 | 4 |
| Фактори зовнішнього середовища | | | |
| Споживачі | 5 | 0,11 | 0,55 |
| Постачальники | 1 | 0,1 | 0,1 |
| Конкуренти | 1 | 0,1 | 0,1 |
| Державні органи влади | 1 | 0,05 | 0,05 |
| Інфраструктура | 4 | 0,06 | 0,24 |
| Законодавчі акти | 2 | 0,1 | 0,2 |
| Профспілки, партії та інші громадські організації | 0 | 0,05 | 0 |
| Система економічних відносин в державі | 1 | 0,06 | 0,06 |
| Організації-сусіди | 0 | 0,01 | 0 |
| Міжнародні події |  | 0,01 | 0 |
| Міжнародне оточення | 0 | 0,03 | 0 |
| Науково-технічний прогрес | 0 | 0,07 | 0 |
| Політичні обставини | -1 | 0,06 | -0,06 |

*Продовження таблиці 5.3.1*

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Соціально-культурні обставини | 0 | 0,05 | 0 |
| Рівень техніки та технологій | 3 | 0,04 | 0,12 |
| Особливості міжнародних економічних відносин | 0 | 0,02 | 0 |
| Стан економіки | 2 | 0,08 | 0,16 |
| Загальна сума | 17 | 1 | 1,52 |
| Фактори внутрішнього середовища | | | |
| Цілі | 3 | 0,11 | 0,33 |
| Структура | 4 | 0,16 | 0,64 |
| Завдання | 3 | 0,07 | 0,21 |
| Технологія | 5 | 0,2 | 1 |
| Працівники | 3 | 0,21 | 0,63 |
| Ресурси | 2 | 0,25 | 0,5 |
| Загальна сума | 20 | 1 | 3,31 |

Серед факторів зовнішнього середовища на проект найбільший позитивний вплив мають споживачі, так як вони є джерелом формування прибутку проекту, рівень техніки та технологій, бо використовуючи нові технології підприємству буде необхідно застосовувати нове програмне забезпечення для забезпечення ефективної роботи підприємства. А серед внутрішніх факторів – структура та технологія.

Негативно впливають політичні та економічні обставини, бо при тяжкій політичній та економічній ситуації у країні прибутки підприємств можуть скоротитись і розробка програмної системи може бути не доцільною.

**5.4. Формування стратегічних альтернатив**

Існує дві групи стратегічних альтернатив. Для першої групу стратегічних альтернатив критеріями поділу альтернативних стратегій розвитку є існуючий продукт (програмне забезпечення) та новий, а також супутні послуги (рис. 5.4.1).



Рис. 5.4.1. Стратегічні альтернативи

Стратегія розроблення нового продукту (проектного рішення) характеризується створенням абсолютно нового програмного забезпечення, яке дає змогу вирішити новоутворені потреби людини, суспільства, економіки тощо.

Стратегія розвитку існуючого продукту (проектного рішення) означає модифікацію програмного забезпечення, його якісних характеристик.

Стратегія розвитку існуючого продукту (проектного рішення) з супутніми послугамиозначає пропонування на ринку модифікованого програмного забезпечення із додатковими послугами (встановлення, супроводження, коригування, адаптування до специфіки конкретного підприємства тощо).

Стратегія нового продукту (проектного рішення) з супутніми послугамиозначає розроблення нового програмного забезпечення та пропонування при його експлуатації додаткових послуг.

Для другої групи стратегічних альтернатив критеріями поділу альтернативних стратегій розвитку є існуючий ринок та продукт, новий ринок та продукт (рис. 5.4.2).



Рис. 5.4.2. Стратегічні альтернативи

Глибше проникнення на ринок полягає в використанні існуючого продукту (проектного рішення) для збільшення частки на існуючому ринку. Якщо фірма володіє достатніми ресурсами та потужностями для виготовлення існуючого продукту, то ця стратегія є найменш ризикованою. Однак, активно зростання на існуючому ринку призведе до зростання конкуренції. Стратегія буде успішною за умови обмежень у ресурсах та потужностях конкурентів або стрімкому розвитку самого ринку. Слід зазначити, що кожен ринок за обсягом має свій ліміт і якщо підприємство прагнутиме розвиватись, то воно повинно використовувати інші запропоновані стратегії.

Стратегія розвитку ринку полягає в використанні існуючого продукту (програмного забезпечення) або незначній його модифікації для виходу на новий сегмент ринку, весь ринок або іноземний ринок. Ця стратегія є з вищим рівнем ризику, оскільки необхідно виходити на новий ринок, де можуть бути інші правила гри, вимоги та смаки споживачів тощо.

Стратегія розвитку продукту полягає у створенні нового продукту (програмного забезпечення) для існуючого сегменту ринку. Ця стратегія є досить ризиковою, оскільки вимагає створення нового продукту (програмного забезпечення) для існуючого сегменту споживачів. Однак, якщо ринок починає зменшувати обсяги та існуючий продукт є на етапі зрілості та падіння, тоді доцільно застосовувати стратегію розвитку продукту.

Стратегія диверсифікації реалізується шляхом виходу на нові сфери бізнесу. Тобто розширення номенклатури товарів, послуг тощо.

З першої групи стратегія розвитку існуючого продукту є найбільш прийнятною для веб-застосування, яке призначене для того, щоб можна було одночасно дізнатися прогноз від різних джерел і порівняти їх, оскільки вже існують продукти-аналоги і даний продукт не потребує надання супутніх послуг.

З другої групи найкраще підходить стратегія розвитку ринку, оскільки продукт не новий, але було розроблено нові методи для класифікації, що підвищить систему на новий рівень.

При виборі цих стратегій необхідно звернути увагу на слабкі сторони конкурентів та врахувати їх при створенні власного продукту. Для системи пов’язаною з погодою дуже важливими характеристиками є простота, швидкодія, привабливість графічного інтерфейсу користувача та використання нових методів подання погоди: карти, графіки, діаграми, інфографіки, чого бракує більшості програмам-аналогам.

**5.5. Бюджетування**

Бюджетування є комплексно обґрунтованою системою розрахунку витрат, пов’язаних з виготовленням та реалізацією продукту, яка дає можливість здійснити аналіз витрат та розробити заходи щодо підвищення рентабельності виробництва. На даному етапі необхідно визначити собівартість продукту, який розробляється та економічно обґрунтувати доцільність вибору однієї із стратегій. Розрахунок витрат поданий у табл. 5.5.1 – 5.5.7.

*Таблиця 5.5.1*

Бюджет витрат матеріалів та комплектуючих виробів

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва матеріалів та комплектуючих | Марка, тип, модель | Фактична кількість, шт. | Ціна за одиницю, грн. | Разом, грн. |
| 1 | 2 | 3 | 4 | 5 |
| Комп'ютер | Asus ET1801IUK-BC023M (90PT0141-M00540) | 2 | 8159 | 11779 |
| Монітор | 23" Asus PA238QR | 2 | 3339 | 6199 |
| Клавіатура | Genius KB-110 PS/2 | 2 | 210 | 420 |
| Комп'ютерна мишка | Logitech Wireless Mouse M185 (910-002238) | 2 | 161 | 322 |
| Принтер | HP Laserjet 1200 | 1 | 1400 | 1400 |
| Стіл | - | 2 | 2000 | 4000 |
| Крісло | - | 4 | 1000 | 4000 |
| Разом: |  | | | 28120 |

*Таблиця 5.5.2*

Бюджет витрат на оплату праці

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Посада, спеціальність | Кількість працівників, осіб | Час роботи, дні | Денна заробітна плата працівників, грн. | Сума витрат на оплату праці, грн. |
| 1 | 2 | 3 | 4 | 5 |
| Основна заробітна плата | | | | |
| Програміст | 1 | 30 | 570 | 17100 |
| Дизайнер | 1 | 10 | 450 | 4500 |
| Разом: | 2 | 40 |  | 21600 |

*Таблиця 5.5.3*

Бюджет обов’язкових відрахувань та податків

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Посада, спеціальність | Сума основної заробітної плати | Сума додаткової заробітної плати | Разом витрат на оплату праці | Сума єдиного внеску на соціальне страхування (22%), грн. | Сума податку з доходів фізичних осіб (18%), грн. | Військовий збір (1.5%), грн |
| Програміст | 17100 | 0 | 17100 | 3762 | 3078 | 256,5 |
| Дизайнер | 4500 | 0 | 4500 | 990 | 810 | 67,5 |
| Разом: | 21600 | 0 | 21600 | 4752 | 3888 | 324 |

*Таблиця 5.5.4*

Бюджет загальновиробничих витрат

|  |  |
| --- | --- |
| Статті витрат | Сума, грн. |
| Змінні загальновиробничі витрати, у т. ч.: | |
| заробітна плата допоміжного персоналу | 1500 |
| витрати на МШП | 270 |
| витрати на електроенергію та технологічні цілі | 250 |
| витрати на ремонт | 0 |
| інші змінні витрати | 210 |
| Разом змінних витрат: | 2230 |
| Постійні загальновиробничі витрати, у т. ч.: | |
| комунальні послуги | 745 |
| витрати на оренду | 5800 |
| Разом постійних витрат: | 6545 |
| Разом загальновиробничих витрат. | 8775 |

*Таблиця 5.5.5*

Бюджет адміністративних витрат на збут

|  |  |
| --- | --- |
| Статті витрат | Сума, грн. |
| Адміністративні витрати, у т. ч.: | |
| заробітна плата адміністративного персоналу | 3300 |
| витрати на МШП | 270 |
| витрати на сплату податків, зборів | 500 |
| знос адміністративного обладнання | 400 |
| Разом адміністративних витрат: | 4470 |
| Витрати на збут, у т. ч.: | |
| заробітна плата менеджерів зі збуту | 3300 |
| витрати на гарантійний ремонт | 500 |
| витрати на гарантійне обслуговування | 500 |
| витрати на налагодження і експлуатацію | 450 |
| витрати на паливно - мастильні матеріали | 1000 |
| витрати на рекламу | 4350 |
| Разом витрат на збут: | 10100 |

*Таблиця 5.5.6*

Зведений кошторис витрат на розробку проектного рішення(продукту)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Статті витрат | Одиниці виміру | Фактична кількість шт. | Ціна одиниці, грн. | Разом, грн. |
| 1 | 2 | 3 | 4 | 5 |
| Комплектуючі вироби | грн | 1 | 28120 | 28120 |
| Паливо та електроенергія на технологічні цілі | кВт | 100 | 1,5 | 150 |
| Основна заробітна плата | грн./день | - | - | 21600 |
| Відрахування на соціальне страхування | % | 22 | - | 4752 |
| Витрати на утримання та експлуатацію устаткування | грн | - | - | 0 |
| Загальновиробничі витрати, у т. ч.: |  | | | |
| змінні; | грн | - | - | 2230 |
| постійні; | грн | - | - | 8775 |
| Адміністративні витрати | грн | - | - | 4470 |
| Витрати на збут | грн | - | - | 10100 |
| Разом виробничих і операційних витрат: |  |  |  | 80197 |

*Таблиця 5.5.7*

Бюджет фінансових результатів

|  |  |
| --- | --- |
| Показники | Сума, грн. |
| 1 | 2 |
| Дохід від реалізації продукції | 120295,5 |
| Податок на додану вартість (20%) | 20049,25 |
| Чистий дохід від реалізації продукції | 100246,25 |
| Собівартість реалізованої продукції | 80197 |
| Валовий прибуток | 40098,5 |
| Операційні витрати: |  |
| - адміністративні витрати: | 4470 |
| - витрати на збут; | 10100 |
| - інші операційні витрати; | 0 |
| Фінансовий результат від операційної діяльності | 25528,5 |
| Податок на прибуток (18%) | 4595,13 |
| Чистий прибуток (збиток) | 20933,37 |

Для визначення фінансових результатів потрібно розрахувати ціну розроблюваного продукту.

*Ц = СБ \* Р* + *СБ*

Ц – ціна продукту, грн

СБ – собівартість продукту, грн

Р – рентабельність 25%

Ц = 80197 \* 1,25 = 100246,25 грн(грн)

Вартість однієї одиниці продукції 100246,25 (грн)

Ц з ПДВ = 100246,25 \* 1,2 = 20049,25 (грн)

Собівартість реалізованої продукції = 80197 (грн)

Валовий прибуток = Чистий дохід від реалізації продукції – Собівартість реалізованої продукції = 120295,5 – 80197 = 40098,5 (грн)

Фінансовий результат від операційної діяльності = Валовий прибуток – Адміністративні витрати – Витрати на збут – Інші операційні витрати = 40098,5 – 4470 – 10100 = 25528,5 (грн)

Податок на прибуток = Фінансовий результат від операційної діяльності \* розмір податку на прибуток = 25528,5 \* 0,18 = 4595,13 (грн.)

Чистий прибуток (збиток) = Фінансовий результат від операційної діяльності – Податок на прибуток = = 25528,5 – 4595,13 = 20933,37 (грн.)

**5.6. Остаточний вибір стратегії**

Основними споживачами погодного порталу є будь-які користувачі, для яких має значення погода.

Собівартість одиниці продукції становить 80197 грн. При рентабельності 25% вартість одиниці продукції становитиме 100246,25 грн. Чистий прибуток від реалізації проекту очікується на рівні 20933,37 грн.

Через наявність на ринку аналогічних продуктів обрано стратегію розвитку існуючого продукту та стратегію стратегія розвитку ринку, оскільки було розроблено нові методи для класифікації, що підвищить систему на новий рівень.

Після проведення економічного аналізу можна зробити висновок, що розроблюваний продукт матиме попит в серед широких кіл користувачів.

**ВИСНОВКИ**

Завданням бакалаврської роботи є проектування та розробка веб-рішення, яке б дозволяло переглядати та аналізувати прогноз погоди з різних джерел (сайти з погодою).

В процесі виконання бакалаврської кваліфікаційної роботи, були поетапно та успішно, з урахуванням специфікації вимог, реалізовані усі частини завдання. Модуль збору інформації про погоду з різних сайтів розроблений, налаштований та успішно працює. Також розроблено та розміщено на хостингу web-сайт, адреса якого: http://iwea.ml.

Дана система зможе зацікавити також інших власників сайтів, які б хотіли розмістити інформацію про погоду у себе на сайті.

Враховуючи розвиток технологій у майбутньому, дана система може бути представлена також на мобільних пристроях.

Собівартість одиниці продукції становить 80197 грн. При рентабельності 25% вартість одиниці продукції становитиме 100246,25 грн. Чистий прибуток від реалізації проекту очікується 20933,37 грн.

**ЛІТЕРАТУРА**

1. OPOGODE [Electronic resource]. - Web page: ru.tsn.ua/ukrayina/ukrainskie-yuzery-teper-sami-mogut-analizirovat-pogodu-v-strane-306182.html?
2. PHP, MySQL и другие веб-технологии [Electronic resource]. – Web page: http://www.php.su/
3. PhpStorm – интегрированная среда разработки на PHP [Electronic resource]. – Web page: http://jetbrains.ru/products/phpstorm/
4. Брюс Баклі, Едвард Дж. Хопкінс, Річард Уайтекер Погода: Енциклопедичний путівник. — К.: Махаон-Україна, 2007, с. 304.
5. Гілецький Й. Р. Економічна і соціально географія: Й. Р. Гілецький, Р. P. Сливка, М. М. Богович; — Львів: ВНТЛ-Класика, 2005.— 400 с.
6. Ким Дж.-О., Мьюллер Ч. У., Клекка У. Р., Олдендер- фер М. С., Блэшфилд Р. К. Факторный, дискриминантный и кластерный анализ. — М.: Финансы и ста- тистика, 1989. — 215 с.
7. Мишель Е. Дэвис и Джон А. Филипс. Изучаем PHP и MySQL, 2008
8. Міхеєва В. С. Математичні методи в економічній географії. Ч. 2. Додаток теорії графів: Курс лекцій - М., 1983.
9. Николай Прохоренок. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера, 2010
10. Шрейдер Ю. А. Что такое расстояние? — М.: Физмат- лит, 1963. — 76 с.

**ДОДАТКИ**

**Додаток А. Діаграма компонентів**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C:\Users\Personal computer\Desktop\Component_diagram.png | | | | | |
| Виконав | Михавко І.В. | Підпис |  | Діаграма компонентів | Сторінка |
| Перевірив | Мельник Р.А. | Підпис |  | 1 з 1 |
| ІКНІ кафедра ПЗ ПІ-42 | | | |

**Додаток Б. Діаграма класів**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C:\Users\Personal computer\Desktop\d_9.png | | | | | |
| Виконав | Михавко І.В. | Підпис |  | Діаграма класів | Сторінка |
| Перевірив | Мельник Р.А. | Підпис |  | 1 з 1 |
| ІКНІ кафедра ПЗ ПІ-42 | | | |

**Додаток В. Схема бази даних**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C:\Users\Personal computer\Desktop\scheme_database.png  jklkjlk | | | | | |
| Виконав | Михавко І.В. | Підпис |  | Схема бази даних | Сторінка |
| Перевірив | Мельник Р.А. | Підпис |  | 1 з 1 |
| ІКНІ кафедра ПЗ ПІ-42 | | | |

**Додаток Д. DDL-скрипт створення бази даних**

--

-- База даних: `noisy2\_weather`

--

-- --------------------------------------------------------

--

-- Структура таблиці `city`

--

CREATE TABLE IF NOT EXISTS `city` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`country` varchar(100) NOT NULL DEFAULT 'Україна',

`name` varchar(100) NOT NULL,

`lat` float NOT NULL,

`lon` float NOT NULL,

`name\_tr` varchar(100) NOT NULL,

`name\_pl` varchar(150) NOT NULL,

`cid\_pl` int(11) NOT NULL,

`name\_iso` varchar(150) NOT NULL,

`name\_sinoptik` varchar(150) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=11 ;

-- --------------------------------------------------------

--

-- Структура таблиці `setting`

--

CREATE TABLE IF NOT EXISTS `setting` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`user\_id` int(11) NOT NULL,

`setting` text NOT NULL,

PRIMARY KEY (`id`),

KEY `setting\_user\_user\_id\_fk` (`user\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=1 ;

-- --------------------------------------------------------

--

-- Структура таблиці `site`

--

CREATE TABLE IF NOT EXISTS `site` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(100) NOT NULL,

`url` varchar(200) NOT NULL,

`image\_url` varchar(200) NOT NULL,

`country` varchar(50) NOT NULL,

`color` varchar(30) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=9 ;

-- --------------------------------------------------------

--

-- Структура таблиці `user`

--

CREATE TABLE IF NOT EXISTS `user` (

`user\_id` int(11) NOT NULL AUTO\_INCREMENT,

`pass` varchar(200) NOT NULL,

`email` varchar(150) NOT NULL,

`name` varchar(100) NOT NULL,

`date` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (`user\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=2 ;

-- --------------------------------------------------------

--

-- Структура таблиці `weather`

--

CREATE TABLE IF NOT EXISTS `weather` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`site\_id` int(11) DEFAULT NULL,

`city\_id` int(11) NOT NULL,

`date` date NOT NULL,

`date\_write` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`min\_temp` float NOT NULL,

`max\_temp` float NOT NULL,

PRIMARY KEY (`id`),

KEY `date` (`date`),

KEY `city\_id` (`city\_id`),

KEY `site\_id` (`site\_id`),

KEY `date\_write` (`date\_write`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=338019 ;

--

-- Обмеження зовнішнього ключа збережених таблиць

--

--

-- Обмеження зовнішнього ключа таблиці `setting`

--

ALTER TABLE `setting`

ADD CONSTRAINT `setting\_user\_user\_id\_fk` FOREIGN KEY (`user\_id`) REFERENCES `user` (`user\_id`);

--

-- Обмеження зовнішнього ключа таблиці `weather`

--

ALTER TABLE `weather`

ADD CONSTRAINT `weather\_city\_id\_fk` FOREIGN KEY (`city\_id`) REFERENCES `city` (`id`),

ADD CONSTRAINT `weather\_site\_id\_fk` FOREIGN KEY (`site\_id`) REFERENCES `site` (`id`);

**Додаток Ж. Вигляд головної сторінки**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C:\Users\Personal computer\Desktop\page_main_iwea.png | | | | | |
| Виконав | Михавко І.В. | Підпис |  | Вигляд головної сторінки | Сторінка |
| Перевірив | Мельник Р.А. | Підпис |  | 1 з 1 |
| ІКНІ кафедра ПЗ ПІ-42 | | | |

**Додаток З. Графіки температур з усіх джерел**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C:\Users\Personal computer\Desktop\all_sites_graph.png | | | | | |
| Виконав | Михавко І.В. | Підпис |  | Графіки температур з усіх джерел | Сторінка |
| Перевірив | Мельник Р.А. | Підпис |  | 1 з 1 |
| ІКНІ кафедра ПЗ ПІ-42 | | | |

**Додаток И. Інструкція користувача**

* **Компоненти ПЗ**
  + Мова програмування:
  + Сервер: PHP
  + Клієнт: JavaScript
  + Середовище розробки – PhpStorm 10.
  + Середовище функціонування – Cross-platform
  + Апаратні вимоги – наявність браузера.
* **Встановлення ПЗ**
  + Для встановлення програмної системи необхідно налаштувати сервер, скопіювати файли сайту та налаштувати роботу з базою даних.
* **Базові функції ПЗ**
  + Веб-сайт надає доступ до основних функцій системи, а саме перегляд погоди на 7 днів, побудова графіків температур, перегляд класифікації даних сайтів прогнозу погоди та порівняльний аналіз даних прогнозу погоди.
* **Аналіз помилок**
  + У випадку збою системи переглянути файл error\_log на хостингу.