

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ

к.т.н, доц. Михелєв І.Л.

" ____ " _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "бакалавр"

на тему:

«Розробка інформаційної системи каталогу підприємств м. Миколаєва»

Виконав: студент групи 4144

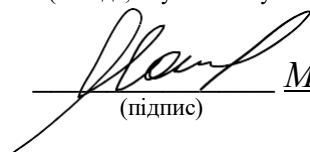
Фоменко Антон Євгенійович

(підпис)

Керівник роботи:

Старший викладач каф. ІУСТ

(посада, науковий ступінь вчене звання)

 Маршак О.І.
(підпис)

м. Миколаїв – 2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій
Спеціальність 126 "Інформаційні системи та технології"
Освітня програма "Інформаційні системи та технології"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

к.т.н, доц. НУК Книрик Н.Р.

" ____ " _____ 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ на здобуття ступеня вищої освіти "бакалавр"

Студенту Фоменко Антону Євгенійовичу

1. Тема роботи: «Розробка інформаційної системи каталогу підприємств м. Миколаєва».

Керівник роботи старший викладач кафедри ІУСТ Маршак О.І.

Затверджені наказом ректора № 269-уч від "18" травня 2022 року.

2. Термін подання роботи: 20 червня 2022 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем.

4. Перелік питань, що належать до розробки (найменування розділів):

1. розробка концепції інформаційної системи каталогу підприємств м. Миколаєва;
2. розробка проектних рішень;
3. реалізація проекту;
4. охорона праці.

5. Перелік презентаційних матеріалів: рисунок 2.1 «Опис взаємодії джерела інформації і отримувача інформації», рисунок 2.2 «Структура ІС, представлена користувачу», рисунок 2.3 «Концептуальна модель бази даних», рисунок 2.4 «Схема комп'ютерної мережі, реалізована на основі Gigabit Ethernet», рисунок 4.1 «Вимоги до системи освітлення робочого місця користувача ПК»; повний огляд таблиць з атрибутами; знімки з користувацького інтерфейсу.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Маршак О.І.	14.02. 2022	21.03.2022
2	Маршак О.І.	21.03.2022	21.04.2022
3	Маршак О.І.	21.04.2022	01.06.2022
4	Гурець Н.В.	01.06.2022	07.06.2022

7. Дата видачі завдання 14 лютого 2022 р.

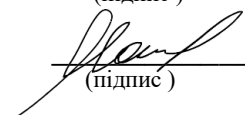
КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Провести аналіз необхідності розробки інформаційної системи каталогу підприємств м. Миколаєва	01.03.2022	Виконано
2	Дослідити сучасні інформаційні технології, що використовуються для розробки інформаційних систем	14.03.2022	Виконано
3	Сформулювати основні вимоги до інформаційної системи каталогу підприємств м. Миколаєва	21.03.2022	Виконано
4	Розробити рішення з інформаційного забезпечення системи	04.04.2022	Виконано
5	Розробити рішення з технічного забезпечення системи	12.04.2022	Виконано
6	Розробити рішення з програмного забезпечення системи	21.04.2022	Виконано
7	Сформулювати перелік та розробити робочу документацію	01.06.2022	Виконано

Студент

Керівник роботи

(підпис)



(підпис)

Фоменко А.Є.
(ПІБ)

Маршак О.І.
(ПІБ)

АННОТАЦІЯ

Дипломна робота присвячена розробці інформаційної системи «Єдина довідкова Миколаєва». У цьому проекті реалізована інформаційна пошукова система, яка була створена шляхом впровадження та використання сучасних інформаційних технологій для надання інформаційних та інших послуг громадянськості міста Миколаїв.

Пояснювальна записка містить 86 сторінок, 4 розділи, 5 рисунків, 3 таблиці, 5 додатків і 20 посилань.

ANNOTATION

Thesis is devoted to creation of an information system «Directory of Mykolaiv city». In this project was realized the information search system which was created through the introduction and use of modern information technology for providing information and other services to the the public of Mykolaiv city.

The explanatory note contains 86 pages, 4 sections, 5 pictures, 3 tables, 5 attachments and 20 links.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

Інформаційна система – ІС

Інформаційні технології – ІТ

Програмне забезпечення – ПЗ

Система управління базою даних – СУБД

Персональний комп'ютер - ПК

Персональна електронно-обчислювальна машина - ПЕОМ

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. РОЗРОБКА КОНЦЕПЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ КАТАЛОГУ ПІДПРИЄМСТВ М. МИКОЛАЄВА.....	11
1.1 Аналіз предметної області.....	11
1.1.1 Характеристики ІС каталогу підприємств м. Миколаєва.....	11
1.1.2 Обґрунтування необхідності розробки ІС каталогу підприємств м. Миколаєва.....	12
1.2 Формування вимог до ІС каталогу підприємств м. Миколаєва.....	13
1.2.1 Аналіз функцій ІС каталогу підприємств м. Миколаєва.....	13
1.2.2 Аналіз системних вимог	15
1.2.3 Вибір архітектури програмного забезпечення.....	15
1.3 Розробка концепції ІС каталогу підприємств м. Миколаєва.....	18
1.3.1 Розробка концепції ІС каталогу підприємств м. Миколаєва.....	18
1.3.2 Технічне обґрунтування обраної концепції та орієнтовний план її реалізації.....	19
1.4 Постановка задачі.....	20
1.4.1 Загальні відомості.....	20
1.4.2 Призначення і мета створення ІС каталогу підприємств м. Миколаєва.....	20
1.4.3 Вимоги до ІС каталогу підприємств м. Миколаєва.....	21
1.4.4 Вимоги до функцій ІС каталогу підприємств м. Миколаєва.....	21
1.4.5 Вимоги до інформаційного забезпечення ІС каталогу підприємств м. Миколаєва.....	22
1.4.6 Вимоги до лінгвістичного забезпечення та програмного забезпечення ІС каталогу підприємств м. Миколаєва.....	22
1.4.7 Вимоги до технічного забезпечення ІС каталогу підприємств м. Миколаєва.....	23

1.4.8 Вимоги до документації ІС каталогу підприємств м. Миколаєва.....	23
1.4.9 Порядок контролю ІС каталогу підприємств м. Миколаєва.....	24
РОЗДІЛ 2. РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ.....	25
2.1 Загальний опис ІС каталогу підприємств м. Миколаєва.....	25
2.2 Рішення з інформаційного забезпечення.....	28
2.2.1 Концептуальна модель даних.....	28
2.2.2 Розробка фізичної моделі ІС каталогу підприємств м. Миколаєва.....	32
2.3 Рішення з технічного забезпечення.....	34
2.4 Рішення з програмного забезпечення.....	35
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЕКТУ.....	36
3.1 Розробка робочої документації.....	36
3.2 Екранні форми розробленого ІС каталогу підприємств м. Миколаєва.....	36
РОЗДІЛ 4. ОХОРОНА ПРАЦІ.....	37
4.1 Правове регулювання охорони праці.....	37
4.2 Значення охорони праці у роботі фахівців з інформаційних технологій.....	38
4.3 Виробничі фактори, які можуть спричинити професійні захворювання в галузі ІТ.....	39
4.4 Основні аспекти забезпечення охорони праці фахівців ІТ галузі.....	41
ВИСНОВКИ.....	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	46
ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ.....	48
ДОДАТОК Б. ЗАГАЛЬНИЙ ОПИС СИСТЕМИ.....	52
ДОДАТОК В. ІНСТРУКЦІЯ КОРИСТУВАЧА.....	54
ДОДАТОК Г. ПРОГРАМА І МЕТОДИКА ВИПРОБУВАНЬ.....	61
ДОДАТОК Д. ПРОГРАМНИЙ КОД.....	65

ВСТУП

Актуальність теми. В умовах глобалізаційних процесів ХХІ століття спостерігається екстремальне збільшення джерел інформації як у світовому масштабі, так і у локальному мережевому просторі. Науковці пояснюють таку інформаційну експансію пропорційним зростанням кількості користувачів мережі Інтернет та обсягом створюваної ними інформації. Таким чином, хаотичне та безсистемне розміщення матеріалів спричинює стрімку втрату значення банку інтернет-даних та створює труднощі з доступом до потрібного джерела.

Як рішення проблеми пропонується метод індивідуалізації засобів інформаційного пошуку, тобто створення профільної пошукової інформаційної системи, яка б дозволила мінімізувати витрачені на пошук ресурси та була б адаптована до індивідуальних особливостей користувачів.

Тому у рамках громадської ініціативи було запропоновано створення інформаційної пошукової системи каталогу підприємств м. Миколаєва.

Метою дипломної роботи є розробка пошукової системи для надання інформаційних та інших послуг громадськості міста Миколаїв.

Виходячи з мети дослідження, в роботі поставлені наступні **завдання**:

- провести аналіз необхідності розробки системи;
- дослідити сучасні інформаційні технології, що використовуються для розробки інформаційних систем;
- сформулювати основні вимоги до ІС;
- розробити рішення з інформаційного, технічного та програмного забезпечення системи;
- сформулювати перелік та розробити робочу документацію.

Об'єктом дослідження є процес розробки та створення ІС каталогу підприємств м. Миколаєва.

Предметом дослідження є послідовність дій необхідних для створення ІС та забезпечення її ефективної роботи.

Структура та обсяг роботи. Дипломна робота складається зі вступу, чотирьох розділів, висновків, списку літератури із 20 найменуваннями та п'яти додатків.

РОЗДІЛ 1. РОЗРОБКА КОНЦЕПЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ КАТАЛОГУ ПІДПРИЄМСТВ М. МИКОЛАЄВА

1.1 Аналіз предметної області

1.1.1 Характеристики ІС каталогу підприємств м. Миколаєва

Інформаційна система «Єдина довідкова Миколаєва» - це унікальна пошукова система, яка була створена шляхом впровадження та використання сучасних інформаційних технологій для надання інформаційних та інших послуг громадськості міста Миколаїв. Архітектура проекту дозволила створити функціональну інформаційну систему, яка містить дані щодо 4683 компаній міста Миколаїв. Профілі компаній наповнені актуальною інформацією з контактними даними, описом товарів та послуг, показниками місця розташування на Google Maps, а також структуровані відносно галузі: органи влади, заклади медицини, освіти, ріелторські послуги, ЗМІ, юридичні послуги та ін.

При розробці ІС «Єдина довідкова Миколаєва» використовувалась «концепція телефонних довідників» - користувач всього за найменуванням або категорією компанії може отримати щодо неї вичерпну інформацію.

Характеристики ІС:

- можливість швидкого пошуку даних щодо обраної компанії;
- можливість інформаційної комунікації між користувачами ІС;
- високий рівень дизайну та оформлення ІС;
- гнучка платформа сайту, можливість подальшого розвитку (модульність);
- зручна навігація, орієнтована як на професійний пошук інформації (журналісти, спеціалісти), так і на пересічного користувача (мешканці, туристи);
- високий рівень захисту інформації;

- швидка реєстрація через поштову скриньку;
- зручність у адмініструванні та наповненні порталу інформацією за допомогою інтуїтивно зрозумілої та простої системи управління;
- відповідність сучасним технічним стандартам;
- можливість додавати компанії та заходи до ІС для користувачів.

1.1.2 Обґрунтування необхідності розробки ІС каталогу підприємств м. Миколаєва

Наразі мережа Інтернет налічує більше 3 мільйонів веб-сторінок за пошуковим запитом «заклади Миколаєва» і жоден ресурс не містить централізованого та структурованого переліку компаній, які наявні у місті. Більш того, задля створення іміджу компанії у медійному просторі кожен підприємець має витрачати значні ресурси, тоді як ІС «Єдина довідкова Миколаєва» дозволить безкоштовно розмістити необхідну інформацію про компанію та залучити нових клієнтів.

Таким чином, видається логічною розробка ІС, яка може поєднувати загалом:

- актуальне оновлення інформації щодо усіх підприємств міста; самозайнятих осіб, що надають послуги населенню; магазинів; закладів харчування, освіти, науки; пам'яток міста та ін.;
- туристичну привабливість, адже кожен, хто відвідує місто вперше, може у повному обсязі ознайомитися з інфраструктурою Миколаєва;
- активну участь громадськості у створенні ІС.

1.2 Формування вимог до ІС каталогу підприємств м. Миколаєва

1.2.1 Аналіз функцій ІС каталогу підприємств м. Миколаєва

До основних функцій ІС належать функції збору та реєстрації інформаційних ресурсів, їхнє збереження, обробка, актуалізація, а так само обробка запитів користувача [1, с. 63].

Збір і реєстрація забезпечують фіксування інформації, яка міститься у ІС. Реалізація цієї функції залежить від джерел інформації, якими виступають електронні та автоматизовані технічні системи. Збір і реєстрація ІС каталогу підприємств м. Миколаєва здійснюється за допомогою обміну даних з іншими автоматизованими системами, а також введення даних у систему у ручному режимі.

Забезпечення контролю над інформацією є необхідним етапом попередньої обробки даних і підготовки їх до завантаження в систему, особливо якщо використовується кілька джерел даних. Він включає процедури фільтрації даних, перевірки, забезпечення логічної цілісності, усунення надмірностей і помилок, заповнення прогалин та інші процедури, призначені для підвищення якості інформації.

У системі керування базами даних (СУБД) адміністратор відповідає за точність даних. Перевірка логічної цілісності даних відбувається на етапі попередньої обробки даних [1, с. 65].

Збереження та нагромадження інформації викликані необхідністю багаторазового використання одних даних під час функціонування ІС. Для цього використовується технологія обробки даних, що передбачає надання користувачеві можливості оперувати інформацією, змінювати її, групувати чи перетворювати. Також користувач може:

- здійснити операцію пошуку в ІС. Для реалізації даної функції потрібно у пошуковому полі ввести необхідну інформацію. Наприклад, знайти

компанію можна за назвою або через відповідну категорію, до якої вона належить;

- знищити інформацію. Здебільшого ця операція не пов'язана з фізичним знищенням даних. Під знищенням розуміється зміна правил доступу до даних і зв'язків з ними інших даних. Так, користувач може видалити свій аккаунт відповідно видаливши особисті дані;

- модифікувати інформацію. Результатом виконання цієї операції є зміна або доповнення даних, які зберігаються в базі. Допустимими є не довільні зміни, а лише ті, які не спричинять істотного збільшення обсягів бази даних понад заздалегідь задане значення. Обмеження на модифікацію накладаються адміністратором через певні зв'язки між окремими даними; наявність формату описання даних та ін.

Представлені опції потрібні для задоволення інформаційних потреб користувачів. Таким чином, в ІС передбачено користувацькі інтерфейси, які представляють собою спосіб взаємодії користувача із ІС. З погляду впливу користувача на послідовність операцій у процесі функціонування ІС використано пакетний інтерфейс [2, с. 53].

Характеристики пакетного режиму виявляються у наступних особливостях:

- алгоритм рішення задачі формалізований, не вимагає втручання людини;
- розрахунок виконується для більшості записів вхідних файлів;
- великий час рішення задачі зумовлений великими обсягами даних [3, с. 185].

ІС – це універсальна пошукова система економічною інфраструктурою міста, яка має надавати виключно актуальну інформацію на момент запиту. Таким чином, необхідно своєчасно і належним чином оновлювати дані. Цей процес полягає в тому, щоб включити та/або видалити рядки в таблицях бази даних, а також відновити значення певних реквізитів.

В ІС інформація оновлюється шляхом введення в систему нових даних, рідше – шляхом видалення існуючих.

1.2.2 Аналіз системних вимог

Системні вимоги - характеристики, яким повинен відповідати цифровий пристрій для коректної роботи інформаційної системи [4, с. 34].

Відрізняють мінімальні та рекомендовані системні вимоги. Перші вказують, яка конфігурація необхідна для початку роботи ІС, другі вказують, яка конфігурація здатна забезпечити максимально комфортні умови роботи ІС [5].

Рекомендовані системні вимоги ІС каталогу підприємств м. Миколаєва:

- жорсткий диск з вільним місцем 10 Гб і більше;
- домен у мережі Інтернет;
- поштовий сервер;
- база даних PostgreSQL;
- система Linux;
- процесор AMD Athlon;
- мережна карта Ethernet (не менше 10 Мбіт/с).

1.2.3 Вибір архітектури програмного забезпечення

Архітектура програмного забезпечення – це система програмного забезпечення, що дає інформацію про компоненти, які складають систему, про взаємозв'язки між цими компонентами і правила, що регламентують такі взаємозв'язки.

Вибір архітектури ПЗ визначає спосіб взаємодії структурних частин програми. Саме архітектура майже повністю визначає такі характеристики ПЗ як надійність, стійкість і зручність супроводу [6, с. 45].

Для створення архітектури ПЗ ІС каталогу підприємств м. Миколаєва застосовані наступні технології.

1. Yii2 – це компонентний PHP фреймворк, призначений для швидкої розробки веб-додатків. Завдяки його компонентній структурі і підтримці кешування, фреймворк особливо підходить для розробки таких великих проектів як портали, форуми, інформаційні системи, інтернет-магазини. Серед переваг платформи:

- висока продуктивність;
- перехоплення та обробка помилок;
- аутентифікація та авторизація;
- можливість підключення сторонніх бібліотек [7, с. 3-4].

2. PHP – це скриптова мова програмування, у код якої вбудовується html-код сторінок.

У PHP вбудовані бібліотеки для роботи з MySQL, SQLite, Oracle, dbm, Hyperware, Informix, InterBase, Sybase та зокрема PostgreSQL. Це також мова, що спрямована на роботу в мережі Інтернет, має універсальний та зрозумілий синтаксис [8].

Аналогами технологій виступають:

1. Laravel – це фреймворк з відкритим кодом, призначений для розробки веб-додатків. Він дозволяє спростити вирішення таких завдань як аутентифікація, маршрутизація, кешування.

Розробка ІС за допомогою Laravel має переваги. По-перше, платформа має варіативний і широкий функціонал, який дає можливість створювати масштабні інтернет-проекти, незалежно від складності і спрямованості.

По-друге, розробник з легкістю може тестувати, додавати оновлення, вносити зміни в роботу сайту за рахунок простої і зрозумілої адмін-панелі [9, с.48].

2. Java – це мова програмування, яка допомагає створювати додатки для цифрових пристроїв. Особливістю програм на Java є те, що вони мають здатність функціонувати на будь-яких комп'ютеризованих пристроях із різними операційними системами [10].

Основні переваги роботи з Java:

- безпека;
- стійкість щодо помилок;
- багатопоточність;
- незалежність від архітектури;
- інтерпретація;
- висока продуктивність;
- розподіленість;
- динамічність [11, с. 96].

3. Spring – це фреймворк з відкритим кодом та підтримкою інверсії управління для платформи Java.

Spring має складові, які надають розробникові можливість використовувати:

- контейнер інверсії управління: конфігурація компонентів додатків і управління життєвим циклом об'єктів Java здійснюється через інверсію управління;
- аспектно-орієнтоване програмування: дозволяє реалізувати наскрізні процедури;
- доступ до даних: робота з реляційною системою управління базами даних на платформі Java;
- управління транзакціями: об'єднує кілька API, управління транзакціями та координує операції для Java-об'єктів;
- програмний каркас, що забезпечує створення веб-додатків;
- аутентифікація і авторизація: налаштовувані процеси безпеки, які підтримують цілий ряд стандартів, протоколів, інструментів і практик;
- тестування: підтримка класів для написання юніт-тестів та інтеграційних тестів [12].

1.3 Розробка концепції ІС каталогу підприємств м. Миколаєва

1.3.1 Розробка концепції

Новітні інформаційні системи розраховані на обробку значної кількості інформації із жорсткими обмеженнями на час видачі результатів. Вони відрізняються складною формалізацією процедур прийняття рішень, високим ступенем інтеграції елементів, що містяться у ІС, великою кількістю зв'язків між елементами.

Мета розробки інформаційних систем – за максимально короткий час створити систему обробки даних, що має визначені параметри [13, с. 3].

Виходячи з цього, метою створення ІС «Єдина довідкова Миколаєва» є створення зручної навігації економічною екосистемою міста, яка зможе задовольнити потреби як пересічних громадян, так і підприємців.

За типом ІС «Єдина довідкова Миколаєва» належатиме до інформаційно–пошукової системи. З її допомогою буде виконуватися введення даних, систематизація, зберігання, видача інформації за запитом користувача без складних перетворень даних.

Створення ІС має ґрунтуватися на трьох концепціях:

1. Об'єктно-орієнтована технологія – сукупність принципів, методів і засобів створення ІС на основі архітектури взаємодії об'єктів. Сьогодні об'єктно-орієнтовані методи є перевіреним способом створення програмних продуктів. Його основними компонентами є об'єктно-орієнтований аналіз, проектування та програмування

2. CASE-технологія – методологія для аналізу, проектування, розробки та підтримки програмного забезпечення ІС, що підтримується взаємопов'язаним набором автоматизованих інструментів. Послідовність етапів розробки така: створення прототипу (макету), специфікація дизайну, контроль проекту, генерація коду, тестування системи, обслуговування.

3. Основана на знаннях (інтелектуальна) технологія передбачає впровадження в ІС елементів штучного інтелекту, зокрема, баз знань і правил виводу для оброблення інформації [6, с. 14].

Однією з основних умов створення ІС каталогу підприємств м. Миколаєва має бути орієнтація на користувача. Більш того, саме людина має виступати активним учасником інформаційного процесу. Тобто користувач здійснюватиме введення інформації в систему і її редагування, а адміністратор зможе виконувати обробку запитів щодо створення нової компанії або заходу та актуалізацію даних «вручну». Тому зазначимо, що ІС має бути інструментом управління, в якому основну роль відіграє людина.

1.3.2 Технічне обґрунтування обраної концепції та орієнтовний план її реалізації

Користувачем ІС може бути кожен, хто має доступ до мережі Інтернет. Щодо виконання управлінських функцій, то необхідний адміністратор системи, який повинен підтримувати її ефективну роботу.

Якщо взаємодія із ІС відбувається вперше, то користувач може зареєструвати особистий кабінет у системі шляхом його прив'язки до поштової скриньки. У разі наявності особистого кабінету користувач може пройти авторизацію у ІС, увівши свої дані (адресу поштової скриньки, ім'я користувача, пароль).

При вході в ІС, система надає можливість скористатися строкою пошукового запиту. Для цього користувачу потрібно увести відповідний запит та натиснути на вкладку «Знайти». Доступна функція пошуку найбільш географічно близьких до місцезнаходження користувача об'єктів, яку можна застосувати, натиснувши на позначення компасу у правому кутку пошукової строки.

Також користувач може знайти компанію шляхом обрання відповідної категорії на головній вкладці ІС. Необхідно натиснути на відповідний розділ категорії, потім обрати спеціалізацію і серед наданого переліку відшукати

компанію. Система надасть актуальну інформацію щодо підприємства, контактні дані, посилання на веб-сторінки та позначить на мапі його місцезнаходження. Для того, щоб додати компанію до ІС користувач має ввести назву та інформацію щодо компанії у спеціальну форму сайту. Адміністратор у свою чергу перевіряє дані та дозволяє показ широкому загалу.

Таким чином, користувач має надіслати HTTP запит на сервер ІС. Наступний крок – це запуск PHP скрипта, що обробляє, перевіряє надану інформацію та формує відповідь, яка відображається на цифровому пристрої користувача.

Інформаційне забезпечення системи складається з бази даних, у якій зберігається вся необхідна інформація.

Технічне забезпечення системи складається з двох серверів. Програмне забезпечення системи складається з бази даних PostgreSQL, веб-серверів PHP 7.4, Apache та операційної системи Ubuntu 20.04.

Значною перевагою ІС є те, що інформація зберігається в єдиній базі даних, вона буде систематизована і придатна для подальшої обробки. Цю інформацію зможуть надалі використовувати інші інформаційні системи.

1.4 Постановка задачі

1.4.1 Загальні відомості

Потрібно розробити інформаційну систему каталогу підприємств міста Миколаєва.

Назва розробки: ІС «Єдина довідкова Миколаєва».

1.4.2 Призначення і мета створення ІС каталогу підприємств м. Миколаєва

Призначенням системи є створення єдиної бази даних із актуальною, централізованою та структурованою інформацією щодо компаній та заходів, міста Миколаїв. У подальшому система має стати платформою для взаємодії

між бізнесом та громадськістю з широким варіативом можливостей. Користувачі матимуть змогу донести до широкого загалу новини щодо економічної інфраструктури та заходів міста, підприємці отримають зручний інструмент для ведення бізнесу, а клієнти зможуть обрати потрібні послуги у декілька кліків.

Метою створення системи є надання можливості ефективного таргетованого пошуку економічною екосистемою міста.

1.4.3 Вимоги до ІС каталогу підприємств м. Миколаєва

Режим роботи ІС має відповідати пакетному режиму, у якому завдання виконуються без необхідності втручання користувача у процес.

Кваліфікація користувачів системи не має значення для застосування можливостей ІС. Рівень комп'ютерної підготовки користувачів повинен відповідати рівню «користувач ПЕОМ». Супроводження інформаційної системи повинно здійснюватися розробниками ІС «Єдина довідкова Миколаєва».

Гнучкість системних компонентів має забезпечувати можливість їх модернізації повністю або частково.

Цільове призначення системи повинне зберігатися протягом підтримки технічного, програмного, інформаційного й організаційного забезпечення.

1.4.4 Вимоги до функцій ІС каталогу підприємств м. Миколаєва

ІС повинна виконувати наступні функції:

- 1) Аутентифікація користувача.
- 2) Ідентифікація користувача.
- 3) Конфіденційність передачі даних по комп'ютерній мережі.
- 4) Ведення бази компаній:
 - додавання нової компанії – створення вкладки;
 - редагування інформації про існуючу компанію;
 - пошук компанії.
- 5) Ведення бази заходів:

- додавання нового заходу – створення вкладки;
- редагування інформації про існуючий захід;
- пошук заходу.

6) Створення запитів та звітів:

- створення запиту та звіту про компанії;
- створення запиту та звіту про заходи.

7) Взаємодія з Google Maps:

- пошук місцезнаходження користувача;
- пошук місцезнаходження компанії;
- пошук місцезнаходження заходу.

1.4.5 Вимоги до інформаційного забезпечення ІС каталогу підприємств м. Миколаєва

Система повинна базуватися на загальноприйнятих стандартах, мати здатність до взаємодії з іншими системами без будь-яких обмежень доступу та реалізації.

Система повинна захищати інформацію за допомогою аутентифікації й ідентифікації користувачів, захисту каналів зв'язку (шифрування даних).

Середовищем розробки має стати система Yii2.

Генерація даних повинна здійснюватися на основі інформації, що зберігається в базі даних ІС. Вхідними даними будуть також поля форм, заповнені користувачем.

1.4.6 Вимоги до лінгвістичного забезпечення та програмного забезпечення ІС каталогу підприємств м. Миколаєва

ІС «Єдина довідкова Миколаєва» розробляється мовою програмування PHP.

До ІС входить сервер бази даних та клієнтські машини.

Вимоги до сервера бази даних:

- операційна система Debian 10;

- веб-сервер PHP 7.4, Apache або Nginx;
- база даних PostgreSQL.

Вимоги до ПЗ клієнтської машини:

- будь-яка операційна система з можливістю доступу до мережі Інтернет і графічним інтерфейсом. Операційна система, що рекомендується – MS Windows 7/XP.

1.4.7 Вимоги до технічного забезпечення ІС каталогу підприємств м. Миколаєва

Вимоги до технічного забезпечення сервера баз даних:

- Процесор AMD Athlon.
- Оперативна пам'ять RAM 2Гб і більше.
- Жорсткий диск з вільним місцем 10 Гб і більше.
- Мережна карта Ethernet (не менше 10 Мбіт/с).
- Вимоги до технічного забезпечення клієнтської машини:
- Мікропроцесор класу Intel i3-1500 і вище.
- Оперативна пам'ять RAM 2Гб і більше.
- Жорсткий диск з вільним місцем 5Гб і більше.
- Мережна карта Ethernet (не менше 10 Мбіт/с).

1.4.8 Вимоги до документації ІС каталогу підприємств м. Миколаєва

Необхідна наявність наступних документів:

- технічне завдання;
- загальний опис системи;
- інструкція користувача;
- програма і методика випробувань;
- програмний код.

1.4.9 Порядок контролю ІС каталогу підприємств м. Миколаєва

Написання документації ІС виконується розробником з метою зафіксувати відповідність чи невідповідність ІС технічному завданню (додаток А).

Тестування ІС виконує розробник протягом створення. Процес тестування являє собою експлуатацію ІС в контрольованих умовах і вивчення отриманих результатів. При цьому перевіряється робота ІС з нормальними і помилковими даними і подіями.

ІС повинна вважатися придатною, якщо вона задовольняє всім пунктам даного технічного завдання (постановці задачі). При виявленні помилок у роботі складається їхній перелік і проводиться їх виправлення. Після цього розробник проводить повторне тестування.

РОЗДІЛ 2. РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ

2.1 Загальний опис ІС каталогу підприємств м. Миколаєва

Мета розробки ІС каталогу підприємств м. Миколаєва – створення зручної пошукової системи економічною інфраструктурою міста для громадськості та бізнесу.

Автоматизовані функції, що може виконувати користувач:

- авторизуватися в системі;
- здійснювати пошук компаній;
- здійснювати пошук заходів;
- створювати інформаційну сторінку нової компанії;
- редагувати інформацію про існуючу компанію;
- створювати інформаційну сторінку нового заходу;
- редагувати інформацію про майбутній захід;
- отримувати інформацію щодо кількості переглядів вкладки доданої компанії;
- отримувати інформацію щодо кількості переглядів вкладки доданого заходу.

Опис ІС

ІС призначена для автоматизованого збирання, реєстрації, збереження, пошуку, оброблення та видачі інформації за запитами користувачів. Вона складається з сукупності взаємозв'язаних компонентів: технічних засобів обробки і передачі даних, методів і алгоритмів обробки, бази даних, об'єднаних за організаційно-структурними, тематичними, технологічними та іншими ознаками для виконання автоматизованої обробки даних з метою задоволення інформаційних потреб споживачів інформації. При цьому визначальним є принцип орієнтації розробленої ІС на потреби конкретних користувачів – жителів міста Миколаїв, підприємців, туристів.

До головних завдань ІС належать:

- виявлення джерел інформації;

– збирання, реєстрація, оброблення та видача інформації про компанії та заходи.

Характерною рисою ІС є те, що людина виступає активним учасником інформаційного процесу. Це виявляється в умовах функціонування системи, де користувач здійснює введення інформації в систему, обробку інформації і її застосування. Таким чином, інформація служить для опису взаємодії джерела інформації і отримувача інформації (рис. 2.1).

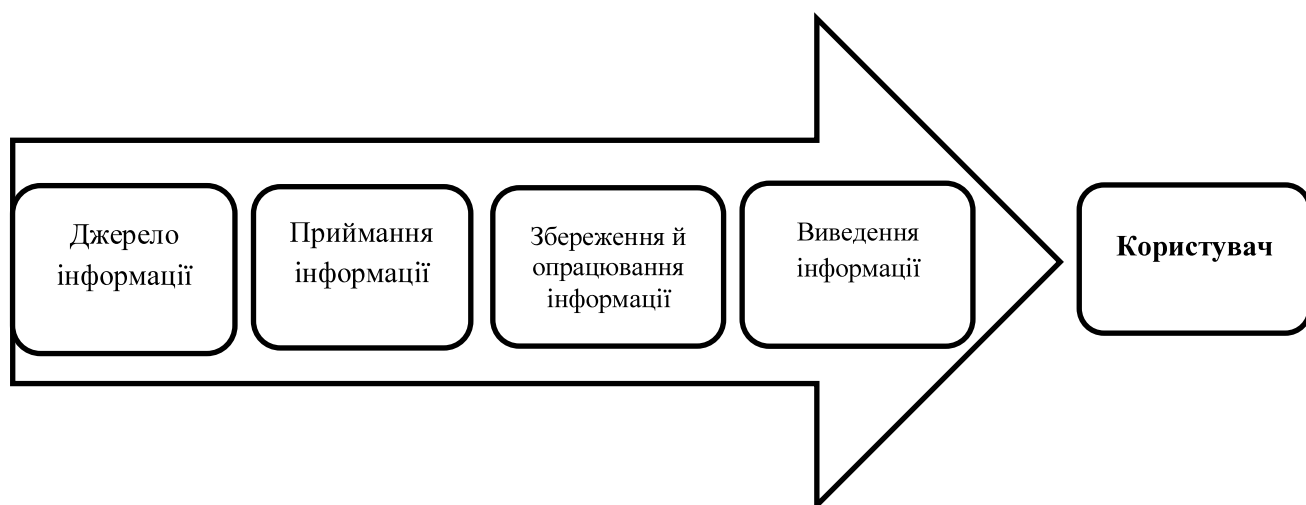


Рис. 2.1- Опис взаємодії джерела інформації і отримувача інформації

Процеси, що забезпечують роботу ІС:

- введення інформації з зовнішніх чи внутрішніх джерел;
- оброблення вхідної інформації (класифікація, розподіл) і представлення її у зручному вигляді;
- виведення інформації для представлення користувачам;
- зберігання інформації може здійснюватися на етапах введення, оброблення і виведення;
- зворотний зв'язок — це інформація, перероблена адміністратором для корекції вхідної інформації.

У вузькому розумінні ІС – це організаційне та управлінське рішення, яке ґрунтується на ІТ, призначене для роботи в певній інформаційній екосистемі. Таким чином, для забезпечення ефективного функціонування ІС необхідний системний адміністратор, який повинен розбиратися у всіх загальних, організаційних та управлінських аспектах таких систем і знати їх можливості в області вирішення виникаючих проблем.

Основними вимогами до адміністратора є:

- інформаційна грамотність (Information Systems Literacy) — загальне уявлення про ІС, що включає в себе знання принципів роботи з ними організацій та окремих користувачів, разом з технічними навичками роботи з комп'ютерами.
- комп'ютерна грамотність (Computer Literacy) — знання ІТ, чітке уявлення про принципи функціонування комп'ютерних систем [14, с. 201].

Однією з основних умов створення ІС каталогу підприємств м. Миколаєва є орієнтація на користувача. Таким чином, видається, що метою розробки ІС стало максимальне задоволення потреб користувача і створення зручного інтерфейсу для пошуку. Отже, на рис. 2.2 зображена структура ІС, представлена користувачу.

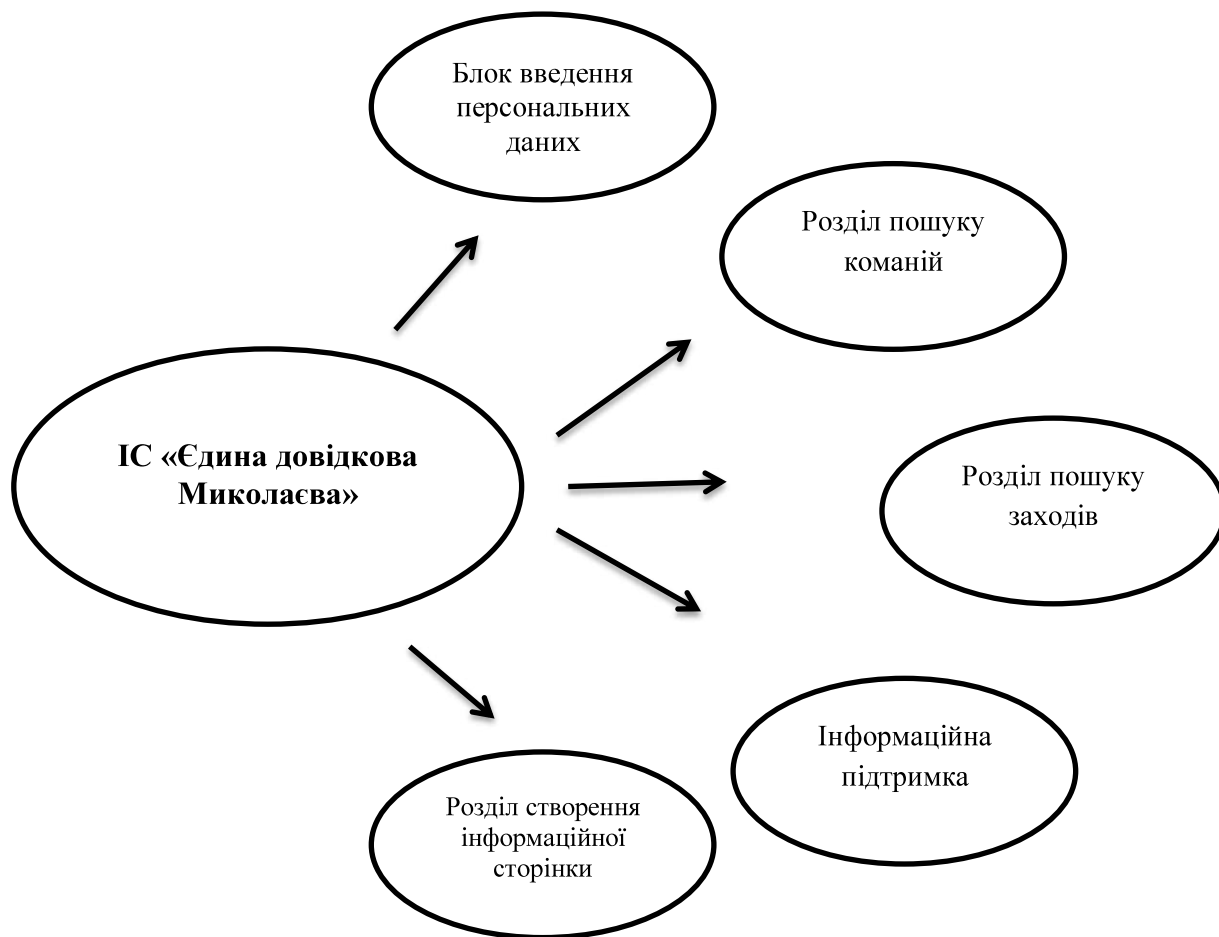


Рис.2.2 - Структура ІС, представлена користувачу

2.2 Рішення з інформаційного забезпечення

Файл бази даних та файл налаштування знаходяться у файловій системі на жорсткому диску сервера даних. Розмір жорсткого диска - не менше 20 Гб.

Наступна інформація повинна бути введена до початку роботи ІС:

- дані про компанії;
- дані про заходи.

2.2.1 Концептуальна модель даних

Концептуальна модель відбиває логічну природу даних, тобто уявлення про них основних користувачів. Модель цього типу має подавати інформацію про ІС у вигляді, незалежному від СУБД. Головне призначення – смисловий опис ІС.

Модель розглядає основні логічні об'єкти (сутності) ІС та зв'язки між ними. Для створення концептуальної моделі даних ІС, потрібно визначити сутності та їх атрибути.

Так, можна виділити основні сутності:

- користувач;
- компанія;
- захід.

Визначимо атрибути сутностей.

Сутність «Користувач»:

- ім'я користувача;
- електронна адреса;
- пароль;
- ключ авторизації;
- ір-адреса.

Сутність «Компанія»:

- місто;
- категорія;
- компанія;
- електронна адреса;
- власник;
- форма власності компанії;
- пароль;
- тариф;
- назва посилання;
- модерація;
- додаткова інформація;
- логотип.

Сутність «Захід»:

- захід;

- категорія;
- місце;
- вартість;
- кількість відвідувачів;
- логін;
- опис;
- вікові обмеження;
- фото;
- статус;
- тариф;
- електронна адреса;
- час.

Концептуальна модель бази даних має наступний вигляд (рис. 2.3):

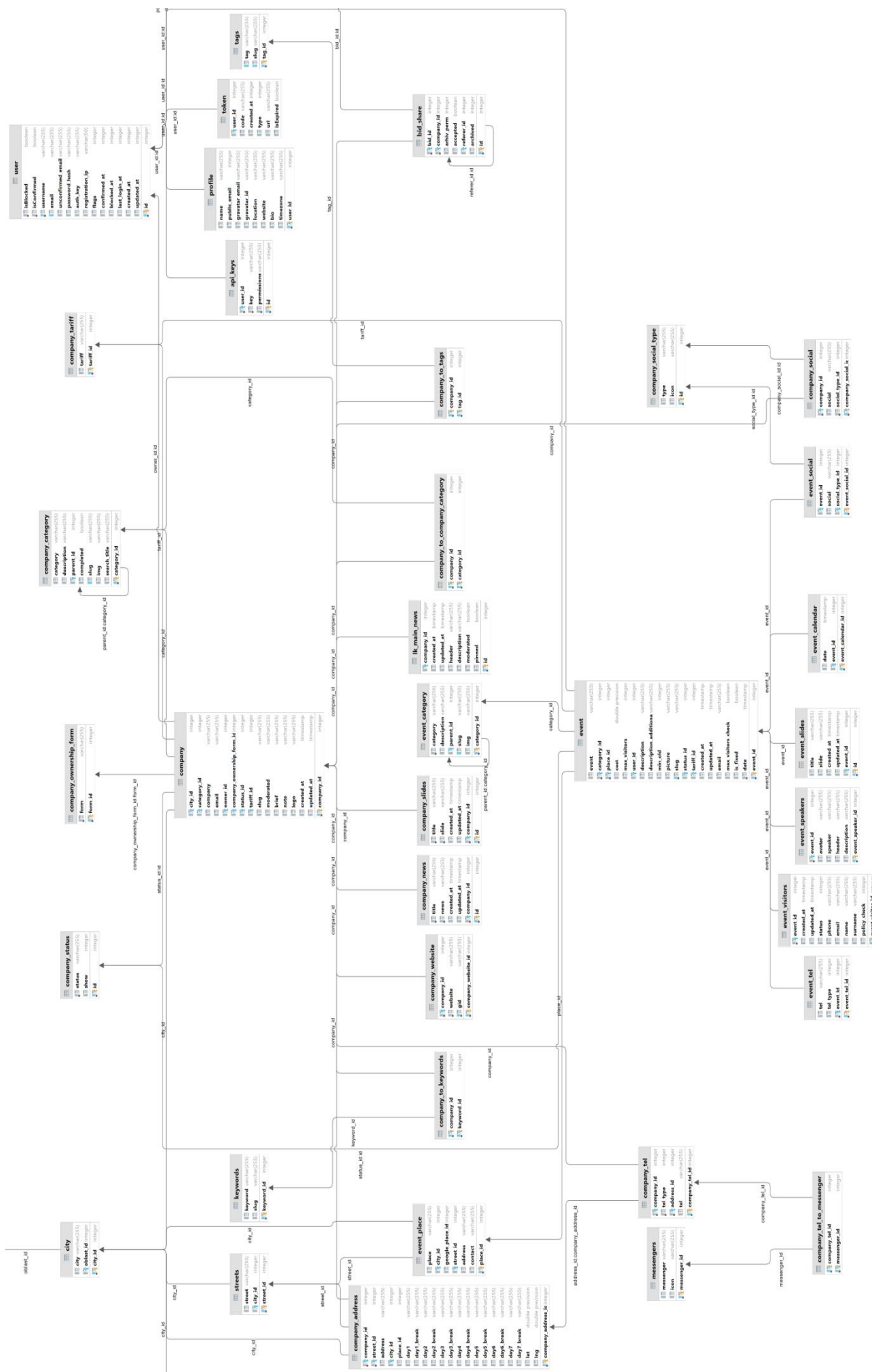


Рис. 2.3 - Концептуальна модель бази даних

2.2.2 Розробка фізичної моделі ІС каталогу підприємств м. Миколаєва

На основі вищевикладеної інформації, розробимо фізичну структуру кожної таблиці, що входить у базу даних.

Таблиця 1 – Структура таблиці «Користувач»

Назва поля	Тип і розмір поля	Опис поля
Id	integer	Первинний ключ
Username	varchar (255)	Ім'я користувача
Email	varchar (255)	Електронна адреса
Password hash	varchar (255)	Пароль
Auth key	varchar (255)	Ключ авторизації
Registration ip	varchar (50)	ІР-адреса

Таблиця 2 – Структура таблиці «Компанія»

Назва поля	Тип і розмір поля	Опис поля
Company id	integer	Первинний ключ
City id	integer	Місто
Category	integer	Категорія
Company	varchar (255)	Компанія
Email	varchar (255)	Електронна адреса
Owner id	integer	Власник
Company ownership form id	integer	Форма власності компанії
Status id	integer	Статус

Tariff id	integer	Тариф
Moderated	varchar (255)	Модерація
Note	varchar (255)	Додаткова інформація
Logo	varchar (255)	Логотип

Таблиця 3 – Структура таблиці «Захід»

Назва поля	Тип і розмір поля	Опис поля
Event id	integer	Первинний ключ
Event	varchar (255)	Захід
Category id	integer	Категорія
Place id	integer	Місце
Cost	double precision	Вартість
Max visitors	integer	Кількість відвідувачів
User id	integer	Логін
Description	varchar (255)	Опис
Min old	integer	Вікові обмеження
Picture	varchar (255)	Фото
Status id	integer	Статус
Tariff id	integer	Тариф
Email	varchar (255)	Електронна адреса
Date	timestamp	Час

Також у базі даних є окремі таблиці «Категорії компаній», «Статус компаній», «Форми власності компаній», «Веб-сайти компаній», «Новини

компаній», «Місцезнаходження компаній», «Відвідувачі заходів», «Спікери заходів», «Календар заходів» та ін.

2.3 Рішення з технічного забезпечення

ІС «Єдина довідкова Миколаєва» призначена для пошуку компаній та заходів міста користувачами, цифрові пристрої яких знаходяться в різних мережах.

Для передачі даних по мережі використовується протокол HTTP.

Програмне забезпечення системи складається з бази даних PostgreSQL, веб-серверів PHP 7.4, Apache та операційної системи Ubuntu 20.04.

Для обслуговування нормального функціонування ІС потрібен середньоскоросний трафік. Таким чином, було обрано технологію Gigabit Ethernet, пропускна здатність якої є достатньою для роботи з ІС.

Фізична передача даних здійснюється на основі кабелю одного типу, а саме 1000BASE-T – «кручена пара» 5-ї категорії. Обрана технологія відповідає стандартам та використовується у більшості випадків роботи з комп'ютерними мережами.

На рис. 2.4 представлена схема комп'ютерної мережі, реалізована на основі Gigabit Ethernet.

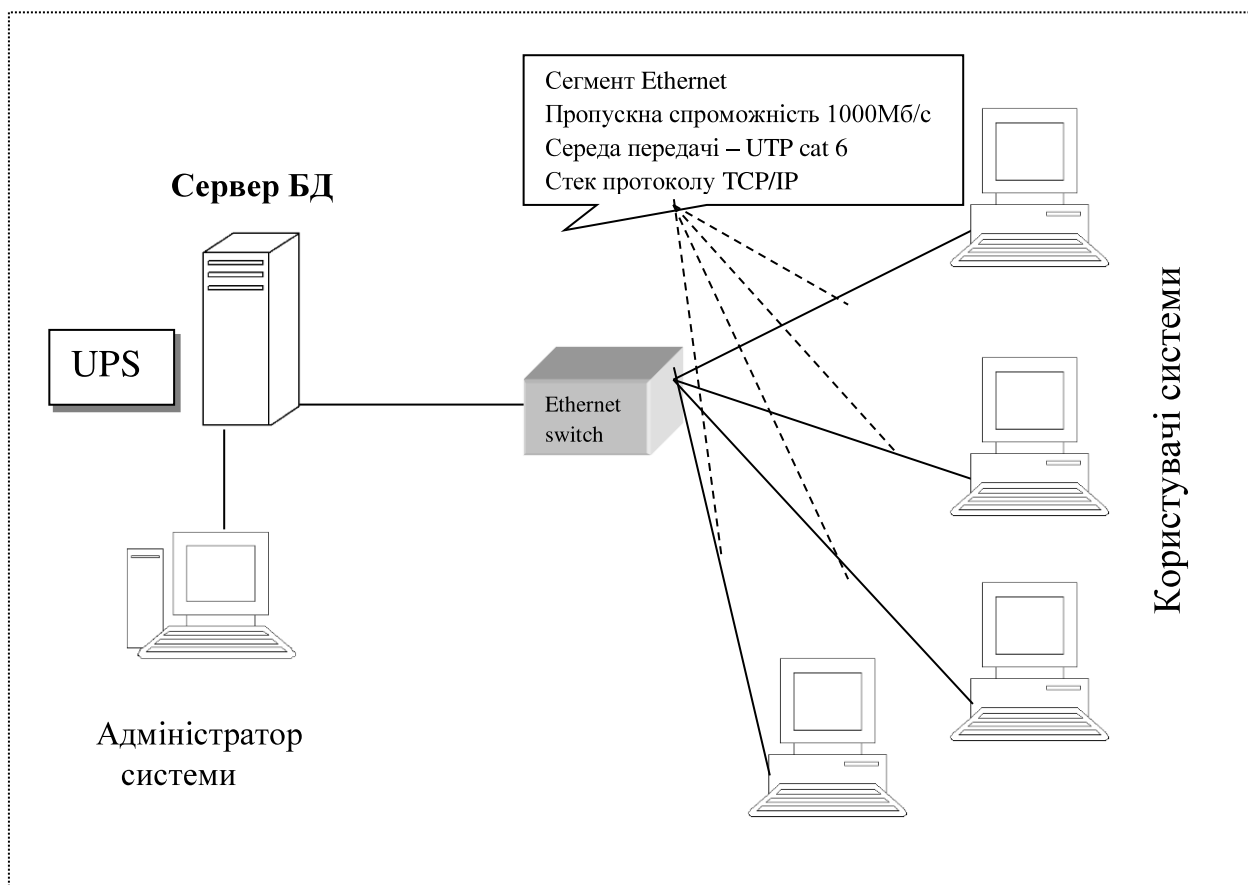


Рис. 2.4 - Схема комп'ютерної мережі, реалізована на основі Gigabit Ethernet

2.4 Рішення з програмного забезпечення

Робоче місце адміністратора облаштовано ноутбуком з такими параметрами:

- процесор Intel Core i5;
- оперативна пам'ять RAM 8Гб;
- жорсткий диск 500 Гб;
- операційна система з графічним інтерфейсом.

Ці показники задовольняють вимогам (п. 1.4.6) до програмного забезпечення.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЕКТУ

3.1 Розробка робочої документації

Вимоги до документації зазначені у п.1.4 «Постановка задачі». Таким чином, були підготовлені такі документи:

- технічне завдання (Додаток А).
- загальний опис системи (Додаток Б).
- інструкція користувача (Додаток В).
- програма і методика випробувань (Додаток Г).
- програмний код (Додаток Д).

Загальні відомості, призначення, мета, основні вимоги до ІС викладені в технічному завданні (додаток А).

Перший розділ дипломної роботи визначає необхідність розробки ІС, містить розробку концепції та обґрунтування архітектури програмного забезпечення ІС.

Другий розділ містить детальний опис системи (додаток Б), рішення з інформаційного, програмного та технічного забезпечення ІС.

Третій розділ визначає робочу документацію до ІС та її опис.

Четвертий розділ містить правові аспекти охорони праці, а також виробничі фактори, які можуть спричинити професійні захворювання в галузі ІТ, та аспекти забезпечення охорони праці фахівців ІТ галузі.

Інструкція користувача розробленої ІС наведена в додатку В. Також були проведені випробування ІС, методика та результати зазначені в додатку Г.

Програмний код інформаційної системи «Єдина довідкова Миколаєва» розміщений у додатку Д.

3.2 Екранні форми розробленого ІС

Екранні форми розробленого ІС приведені у інструкції користувача (додаток В).

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1 Правове регулювання охорони праці

Охорона праці — це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я та працездатності людини у процесі трудової діяльності [15].

Законодавство України про охорону праці включає Конституцію України, спеціальний Закон України «Про охорону праці», Кодекс законів про працю України та інші закони.

У Конституції України проголошені права громадян на: належні, безпечні й здорові умови праці (ст. 43); соціальний захист громадян, забезпечення їх у випадку повної, часткової або тимчасової втрати працездатності, втрати годувальника (ст. 46); охорону здоров'я, медичну допомогу й страхування (ст.49); безпечне для життя й здоров'я навколишнє середовище й відшкодування заподіяної порушенням цього права шкоди (ст. 50) [16].

Основним законодавчим документом у сфері охорони праці є Закон України «Про охорону праці», який поширюється на всі підприємства, установи та організації незалежно від форм власності, на всіх громадян, які працюють і на цих підприємствах. Він визначає основні положення щодо реалізації конституційного права працівників на охорону життя і здоров'я на виробництві, на належні, безпечні та здорові умови праці, регулює відносини між роботодавцем і працівником з питань безпеки, гігієни праці та виробничого середовища та встановлює єдиний порядок організації охорони праці в Україні.

Кодекс законів про працю (КЗпП) України регулює трудові відносини в цілому. У Кодексі питання охорони праці відображені в ряді статей та в спеціальному розділі відповідно до Закону «Про охорону праці». Зокрема, відповідно до ст. 29 Кодексу перед початком роботи власник або уповноважений ним орган зобов'язаний: роз'яснити працівнику його права та обов'язки та повідомити під розписку про умови праці; ознайомити працівника

з правилами внутрішнього трудового розпорядку та колективного договору; визначити місце роботи працівника, забезпечити його необхідними засобами для роботи; провести інструктаж робітника з техніки безпеки, виробничої санітарії, гігієни праці та протипожежної безпеки.

До законів, які регулюють суспільні відносини в сфері безпеки людини в процесі трудової діяльності, також належать: Закон України «Про забезпечення санітарного й епідемічного благополуччя населення»; Кодекс цивільного захисту України.

Таким чином, державна політика України в галузі безпеки праці спрямована на створення безпечних і здорових умов праці, запобігання нещасним випадкам та професійним захворюванням. Вона базується на низці принципів, основними з яких є пріоритет життя і здоров'я працівників, повна відповідальність роботодавця за створення безпечних та належних умов праці, соціальний захист працівників, повне відшкодування шкоди особам, які потерпіли від нещасних випадків на виробництві та професійних захворювань [17, с. 140-143].

4.2 Значення охорони праці у роботі фахівців з інформаційних технологій

Безперечно ІТ технології у ХХІ столітті стали невід'ємною частиною життя кожного, професія фахівця цієї області популярною, а кількість компаній, які створюють цифрові продукти, злічити вже неможливо. Саме тому на перший план виходять безпечні умови праці кожного працівника зазначеної сфери.

Професія програміста та інших фахівців ІТ-технологій пов'язана з колосальним розумовим напруженням. При цьому не варто забувати, що умови праці програмістів також характеризуються можливістю впливу на них наступних небезпечних і шкідливих виробничих факторів:

- шуму;

- тепловиділень, причому шкоди організму можуть завдати не тільки високі, але і низькі температури;
- іонізуючих і неіонізуючих випромінювань: рентгенівське, інфрачервоне, електромагнітне випромінювання;
- статичної електрики;
- недостатнього штучного та природнього освітлення;
- візуальних факторів: яскравість, контрастність, мерехтіння зображення, відблиски тощо.

У цих умовах зростає роль і значення охорони праці, як системи правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних, лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я та працездатності людини на робочому місці. Адже плоди науково-технічного прогресу можуть бути ефективні лише тією мірою, якою вони забезпечують людині безпеку, комфорт та зручність праці [18, с. 127].

4.3 Виробничі фактори, які можуть спричинити професійні захворювання в галузі ІТ

У процесі трудової діяльності на людину впливає комплекс несприятливих чинників виробничого середовища і трудового процесу, у результаті чого можливі виробничі травми і професійні захворювання. Комплексний характер впливу цих чинників визначає необхідність системного підходу під час вирішення питань безпеки праці. Усі етапи технологічного процесу характеризуються інформаційним навантаженням, їх виконання вимагає уваги, зосередженості, що необхідно враховувати під час створення безпечних і нешкідливих умов праці користувачів ІТ. Необхідно розуміти, що комп'ютер як технічний засіб може бути джерелом небезпечних і шкідливих виробничих чинників.

З принципу роботи ПК виходить, що він є джерелом:

- іонізуючих випромінювань, а саме рентгенівського випромінювання;
 - оптичного випромінювання у видимому діапазоні, а також в УФ діапазоні, яке виникає завдяки взаємодії електронів з шаром люмінофора;
 - електромагнітного випромінювання в радіочастотному діапазоні.
- При цьому високочастотні поля виникають під впливом електронного променя і пов'язані з частотою формування елемента зображення й інтенсивністю променя, низькочастотні поля виникають у системі горизонтальної розгортки, а дуже низькочастотні поля пов'язані з генерацією (вертикальною розгорткою);
- магнітного поля, що виникає через наявність відхиляючого пристрою;
 - електростатичного поля, що виникає у зв'язку з високим потенціалом анодів.

Отже, при роботі із ПК можна виділити наступні шкідливі фактори:

- фізичні - електромагнітне випромінювання моніторів, недостатня освітленість робочого місця, підвищена напруженість електричних і магнітних полів;
- психофізіологічні - тривалі навантаження м'язів рук, шиї, спини, зап'ясть, пліч, ніг, постійна напруга зору, нервово-емоційні перевантаження

Професійними захворюваннями вважаються такі захворювання, які виникли в результаті трудової діяльності особи і викликані впливом шкідливих виробничих чинників.

Праця користувача комп'ютера часто відбувається за умови надмірного нервово-емоційного й зорового напруження, обмеженої рухової активності, електростатичних полів, радіочастотного та рентгенівського випромінювання, які призводять до неврозів, відхилень у психіці, погіршення пам'яті, захворювань опорно-рухового апарату, серцево-судинної, імунної та статеві систем, органів зору, шкіри та ін. [19, с. 43].

4.4 Основні аспекти забезпечення охорони праці фахівців ІТ галузі

Облаштування робочого місця повинно забезпечувати:

- належні умови освітлення приміщення і робочого місця, відсутність відблисків;
- оптимальні параметри мікроклімату;
- належні ергономічні характеристики основних елементів робочого місця, а також враховувати небезпечні і шкідливі фактори.

Площа приміщень для роботи з ПК розраховується з урахуванням кількості осіб, які працюють в одному приміщенні, таким чином, щоб площа на одне робоче місце була не менше 6,0 м² [20, с. 16].

Приміщення для роботи з ПК повинно мати природне та штучне освітлення. Штучне освітлення повинно забезпечуватися системою загального освітлення, яка включає суцільні або підвісні лінії приборів освітлення, розташованих на робочій зоні (переважно з лівого боку від робочого місця).

При розміщенні ПК по периметру приміщення лінії світильників штучного освітлення повинні розміщуватися локально над робочими місцями [17, с. 243]. Система освітлення робочого місця користувача ПК має відповідати наступним вимогам (рис. 4.1).

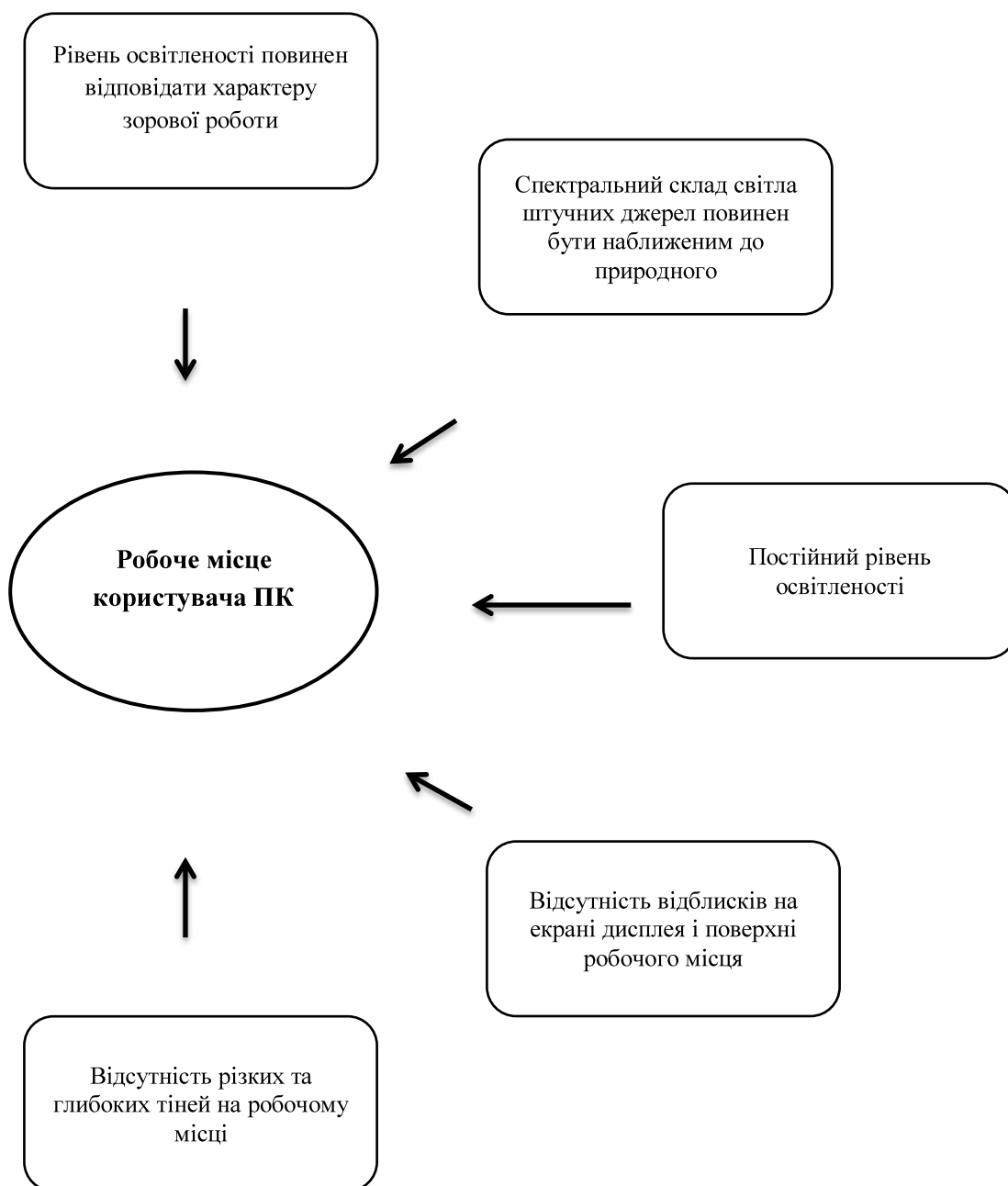


Рис. 4.1 - Вимоги до системи освітлення робочого місця користувача ПК

Організація робочого місця користувача ПК повинна відповідати сучасним вимогам ергономіки, характеру виконуваної роботи і забезпечити оптимальне розміщення на робочій поверхні обладнання ПК.

Монітор на робочому місці встановлюється так, щоб верхній край екрана знаходився на рівні очей. Розташування монітора ПК має забезпечувати:

- 1) безпечність роботи в цілому;
- 2) зручність та ефективність зорової роботи з екраном в вертикальній площині під кутом ± 30 градусів від лінії зору.

Клавіатура розміщується на поверхні столу або висувній полиці на відстані 100-300 мм від краю, ближчого до користувача. Клавiші клавіатури мають бути зручними в роботі (хід всіх клавiш має бути однаковим з мінімальним опором натискання 0,25 Н та максимальним – не більше 1,5 Н).

Конструкція робочого місця користувача ПК має забезпечувати підтримання оптимальної робочої пози. Висота поверхні сидіння робочого крісла має регулюватися в межах 400- 500 мм, а кут нахилу поверхні – від 150 градусів вперед і до 50 градусів назад. При виконанні великого обсягу роботи кращим варіантом буде, якщо сидіння буде трошки нахилене вперед.

Робоче місце користувача ПК необхідно розташовувати відносно світлових прорізів (вікон) так, щоб природне світло падало збоку, переважно зліва [17, с. 245-247].

ВИСНОВКИ

У результаті виконання даної дипломної роботи було розроблено ІС каталогу підприємств м. Миколаєва. Для досягнення цієї мети були вирішені наступні завдання:

- проведений аналіз необхідності розробки системи;
- досліджено сучасні інформаційні технології, що використовуються для розробки інформаційних систем;
- сформовано основні вимоги до ІС;
- розроблено рішення з інформаційного, технічного та програмного забезпечення системи;
- сформовано перелік та розроблено робочу документацію.

Для аналізу необхідності розробки системи були досліджені Інтернет джерела, які містять інформацію щодо закладів та заходів міста Миколаїв. У результаті цього встановлено, що жоден веб-ресурс не має структурованої та повної інформації щодо наявних у місті компаній та запланованих заходів.

Для дослідження сучасних інформаційних технологій, що використовуються для розробки інформаційних систем, були вивчені наукові статті та веб-ресурси відповідного профілю. Таким чином, були наведені технології, які використовувалися для розробки ІС каталогу підприємств м. Миколаєва та їх аналоги.

При формуванні вимог до ІС були визначені призначення, мета, функції, задачі системи та як наслідок сформовані системні вимоги і обрано архітектуру програмного забезпечення.

Під час розробки рішень з інформаційного забезпечення були створені концептуальна і фізична модель даних для бази даних системи.

Було реалізоване необхідне програмне забезпечення системи завдяки базі даних PostgreSQL, веб-серверам PHP 7.4, Apache та операційній системі Ubuntu 20.04.

Визначена та розроблена необхідна документація до системи: технічне завдання; загальний опис системи; інструкція користувача; програма і методика випробувань; програмний код.

Розроблена ІС «Єдина довідкова Миколаєва» відповідає технічному завданню. Інформаційна система наразі доступна для використання у мережі Інтернет за посиланням «<https://otvet.mk.ua/>».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бутко М. П., Бутко І. М., Мащенко В. П. Теорія прийняття рішень: навч. посіб. Київ: Центр учбової літератури, 2015. 360 с.
2. Ушакова І. О., Плеханова Г. О. Інформаційні системи та технології на підприємстві: навч. посіб. Харків: ХНЕУ, 2009. 128 с.
3. Бенько М.М. Інформаційні системи і технології в бухгалтерському обліку: монографія. Київ: КНТЕУ, 2010. 336 с.
4. Bourque P. R., Fairley R. E. Guide to the Software Engineering Body of Knowledge: SWEBOOK. Washington; DC; United States: IEEE Computer Society Press, 2004. 335 p.
5. System Requirements Definition. *The Computer Dictionary*. URL: https://techterms.com/definition/system_requirements (Last accessed: 07.06.2022).
6. Авраменко В.С., Авраменко А.С.Проектування інформаційних систем: навч. посіб. Черкаси: Черкаський національний університет ім. Б. Хмельницького, 2017. 433 с.
7. Полное руководство Yii 2.0. *Yiiframework Guide*. URL: <https://www.yiiframework.com/doc/guide/2.0/ru> (Last accessed: 07.06.2022).
8. Руководство по PHP. URL: <https://www.php.net/manual/ru/index.php> (Last accessed: 07.06.2022).
9. Веллинг Л., Томсон Л. Разработка веб-приложений с помощью PHP и MySQL: пособие. Санкт-Петербург: Диалектика, 2019. 768 с.
10. Мова програмування Java та платформа JavaFx: приклади застосування. URL: <http://www.dut.edu.ua/ua/news-1-626-6031-movaprogramuvannya-java-ta-platforma-javafx-prikladi-zastosuvannya> (Дата звернення: 07.06.2022).
11. Шилдт Г. Java: полное руководство. Москва: ООО «И.Д. Вильямс», 2012. 1104 с.

12. Walls C. Spring in action. URL: <https://www.manning.com/books/spring-in-action-fourth-edition> (Last accessed: 07.06.2022).

13. Недашківський О.Л. Планування та проектування інформаційних систем: навч. посіб. Київ: ДУТ, 2014. 215 с.

14. Палеха Ю.І. Інформаційна культура: навч. посібн. Київ: Ліра-К, 2020. 400 с.

15. Про охорону праці: Закон України від 14.10.1992 р. № 2694-XII. Законодавство України : база даних / Верхов. Рада України. Дата оновлення: 14.08.2021. URL: <https://zakon.rada.gov.ua/laws/show/2694-12#Text> (Дата звернення: 07.06.2022).

16. Конституція України від 28.06.1996 р. № 254к/96-ВР. Законодавство України : база даних / Верхов. Рада України. Дата оновлення: 01.01.2020. URL: <https://zakon.rada.gov.ua/laws/show/254%D0%BA/96-%D0%B2%D1%80#Text> (Дата звернення: 07.06.2022).

17. Голінько В.І., Іконніков М.Ю., Лебедєв Я.Я. Охорона праці в галузі інформаційних технологій : навч. посіб. Дніпро: НГУ, 2015. 246 с.

18. Левон М.С. Система факторов, влияющих на производительность труда в it-фирмах. *Экономика, Статистика и Информатика*. 2015. №3. С.127-128.

19. Яскілка В.Я., Олійник М.З. Охорона праці в галузі: конспект лекцій. Тернопіль: ТНТУ імені Івана Пулюя, 2016. 56 с.

20. Мигидин М.В. Значення знань з охорони праці у сфері інформаційних технологій. *Вісник Вінницького національного технічного університету*. 2016. № 2. С. 14-17.

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ

Загальні відомості

Потрібно розробити інформаційну систему каталогу підприємств міста Миколаєва.

Назва розробки: ІС «Єдина довідкова Миколаєва».

Призначення і мета створення ІС каталогу підприємств м. Миколаєва

Призначенням системи є створення єдиної бази даних із актуальною, централізованою та структурованою інформацією щодо компаній та заходів, міста Миколаїв. У подальшому система має стати платформою для взаємодії між бізнесом та громадськістю з широким варіативом можливостей. Користувачі матимуть змогу донести до широкого загалу новини щодо економічної інфраструктури та заходів міста, підприємці отримають зручний інструмент для ведення бізнесу, а клієнти зможуть обрати потрібні послуги у декілька кліків.

Метою створення системи є надання можливості ефективного таргетованого пошуку економічною екосистемою міста.

Вимоги до ІС каталогу підприємств м. Миколаєва

Режим роботи ІС має відповідати пакетному режиму, у якому завдання виконуються без необхідності втручання користувача у процес.

Кваліфікація користувачів системи не має значення для застосування можливостей ІС. Рівень комп'ютерної підготовки користувачів повинен відповідати рівню «користувач ПЕОМ». Супроводження інформаційної системи повинно здійснюватися розробниками ІС «Єдина довідкова Миколаєва».

Гнучкість системних компонентів має забезпечувати можливість їх модернізації повністю або частково.

Цільове призначення системи повинне зберігатися протягом підтримки технічного, програмного, інформаційного й організаційного забезпечення.

Вимоги до функцій ІС каталогу підприємств м. Миколаєва

ІС повинна виконувати наступні функції:

- 1) Аутентифікація користувача.
- 2) Ідентифікація користувача.
- 3) Конфіденційність передачі даних по комп'ютерній мережі.
- 4) Ведення бази компаній:
 - додавання нової компанії – створення вкладки;
 - редагування інформації про існуючу компанію;
 - пошук компанії.
- 5) Ведення бази заходів:
 - додавання нового заходу – створення вкладки;
 - редагування інформації про існуючий захід;
 - пошук заходу.
- 6) Створення запитів та звітів:
 - створення запиту та звіту про компанії;
 - створення запиту та звіту про заходи.
- 7) Взаємодія з Google Maps:
 - пошук місцезнаходження користувача;
 - пошук місцезнаходження компанії;
 - пошук місцезнаходження заходу.

Вимоги до інформаційного забезпечення ІС каталогу підприємств м. Миколаєва

Система повинна базуватися на загальноприйнятих стандартах, мати здатність до взаємодії з іншими системами без будь-яких обмежень доступу та реалізації.

Система повинна захищати інформацію за допомогою аутентифікації й ідентифікації користувачів, захисту каналів зв'язку (шифрування даних).

Середовищем розробки має стати система Yii2.

Генерація даних повинна здійснюватися на основі інформації, що зберігається в базі даних ІС. Вхідними даними будуть також поля форм, заповнені користувачем.

Вимоги до лінгвістичного забезпечення та програмного забезпечення ІС каталогу підприємств м. Миколаєва

ІС «Єдина довідкова Миколаєва» розробляється мовою програмування PHP.

До ІС входить сервер бази даних та клієнтські машини.

Вимоги до сервера бази даних:

- операційна система Debian 10;
- веб-сервер PHP 7.4, Apache або Nginx;
- база даних Postgresql.

Вимоги до ПЗ клієнтської машини:

- будь-яка операційна система з можливістю доступу до мережі Інтернет і графічним інтерфейсом. Операційна система, що рекомендуються – MS Windows 7/XP.

Вимоги до технічного забезпечення ІС каталогу підприємств м. Миколаєва

Вимоги до технічного забезпечення сервера баз даних:

- Процесор AMD Athlon.
- Оперативна пам'ять RAM 2Гб і більше.
- Жорсткий диск з вільним місцем 10 Гб і більше.

- Мережна карта Ethernet (не менше 10 Мбіт/с).

Вимоги до технічного забезпечення клієнтської машини:

- Мікропроцесор класу Intel i3-1500 і вище.
- Оперативна пам'ять RAM 2Гб і більше.
- Жорсткий диск з вільним місцем 5Гб і більше.
- Мережна карта Ethernet (не менше 10 Мбіт/с).

Вимоги до документації ІС каталогу підприємств м. Миколаєва

Необхідна наявність наступних документів:

- технічне завдання;
- загальний опис системи;
- інструкція користувача;
- програма і методика випробувань;
- програмний код.

Порядок контролю ІС каталогу підприємств м. Миколаєва

Написання документації ІС виконується розробником з метою зафіксувати відповідність чи невідповідність ІС технічному завданню (додаток А).

Тестування ІС виконує розробник протягом створення. Процес тестування являє собою експлуатацію ІС в контрольованих умовах і вивчення отриманих результатів. При цьому перевіряється робота ІС з нормальними і помилковими даними і подіями.

ІС повинна вважатися придатною, якщо вона задовольняє всім пунктам даного технічного завдання (постановці задачі). При виявленні помилок у роботі складається їхній перелік і проводиться їх виправлення. Після цього розробник проводить повторне тестування.

ДОДАТОК Б

ЗАГАЛЬНИЙ ОПИС СИСТЕМИ

Призначенням системи є створення єдиної бази даних із актуальною, централізованою та структурованою інформацією щодо компаній та заходів, міста Миколаїв.

Мета розробки ІС каталогу підприємств м. Миколаєва – створення зручної пошукової системи економічною інфраструктурою міста для громадськості та бізнесу.

Автоматизовані функції, що може виконувати користувач:

- авторизуватися в системі;
- здійснювати пошук компаній;
- здійснювати пошук заходів;
- створювати інформаційну сторінку нової компанії;
- редагувати інформацію про існуючу компанію;
- створювати інформаційну сторінку нового заходу;
- редагувати інформацію про майбутній захід;
- отримувати інформацію щодо кількості переглядів вкладки доданої компанії;
- отримувати інформацію щодо кількості переглядів вкладки доданого заходу.

Користувачем ІС може бути кожен, хто має доступ до мережі Інтернет. Щодо виконання управлінських функцій, то необхідний адміністратор системи, який повинен підтримувати її ефективну роботу.

Інформаційне забезпечення системи складається з:

- бази даних, у якій зберігається вся інформація про компанії та заходи;
- файлів автентифікації, які містять логіни та паролі користувачів системи. Доступ до них має лише адміністратор системи.

Технічне забезпечення системи складається з двох серверів.

Програмне забезпечення системи складається з бази даних PostgreSQL, веб-серверів PHP 7.4, Apache та операційної системи Ubuntu 20.04.

ДОДАТОК В

ІНСТРУКЦІЯ КОРИСТУВАЧА

Вступ

ІС «Єдина довідкова Миколаєва» - це автоматизована система пошуку підприємств міста Миколаєва.

Система автоматизує наступні функції:

- ведення бази компаній;
- ведення бази заходів;
- створення запитів та звітів;
- взаємодія з Google Maps.

Початок роботи

1. Для початку роботи користувач має пройти реєстрацію у системі, натиснувши на вкладку «Реєстрація» у правому верхньому кутку (рис. В.1). Потрібно ввести адресу поштової скриньки, ім'я користувача та пароль та підтвердити реєстрацію, перейшовши за посиланням з електронної пошти.

4683 компанії в каталогі

ВОЙТИ РЕГІСТРАЦІЯ

otvet.mk.ua

Введите что вы ищете

КАТАЛОГ МЕРОПРИЯТИЯ ТУРБИРЖА

Введите запрос, например "рестораны"

ПОИСК

Зарегистрироваться

Email

Имя пользователя

Пароль

☐ Я соглашаюсь с правилами регистрации.

ЗАРЕГИСТРИРОВАТЬСЯ

Уже зарегистрированы? Авторизуйтесь!

Рис В.1 – Вкладка для реєстрації користувача

2. Після реєстрації користувач може авторизуватися в системі, увівши логін та пароль від особистого кабінету (рис. В.2 та В.3)

Рис В.2 – Вкладка для авторизації користувача

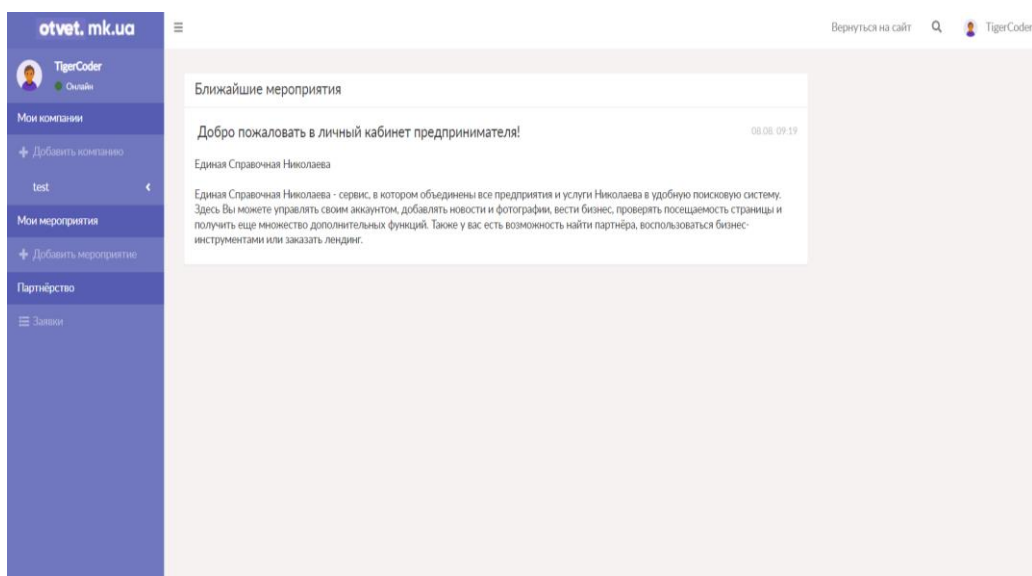


Рис В.3 – Особистий кабінет

Пошук компанії та заходу

1. Для пошуку компанії чи заходу користувач може скористатися пошуковою строкою на головній сторінці системи. Потрібно ввести назву закладу чи заходу або категорію, до якої він належить. Також можна обрати

найближчий об'єкт до свого місцезнаходження, натиснувши на іконку навігації справа у пошуку (рис. В.4).

Для того, щоб ознайомитися з категоріями компаній та обрати з переліку, користувач має натиснути на вкладку «Каталог», а щоб дізнатися більше про найближчі заходи – на вкладку «Заходи». Вони розташовані над пошуковою строкою у нижньому правому кутку.

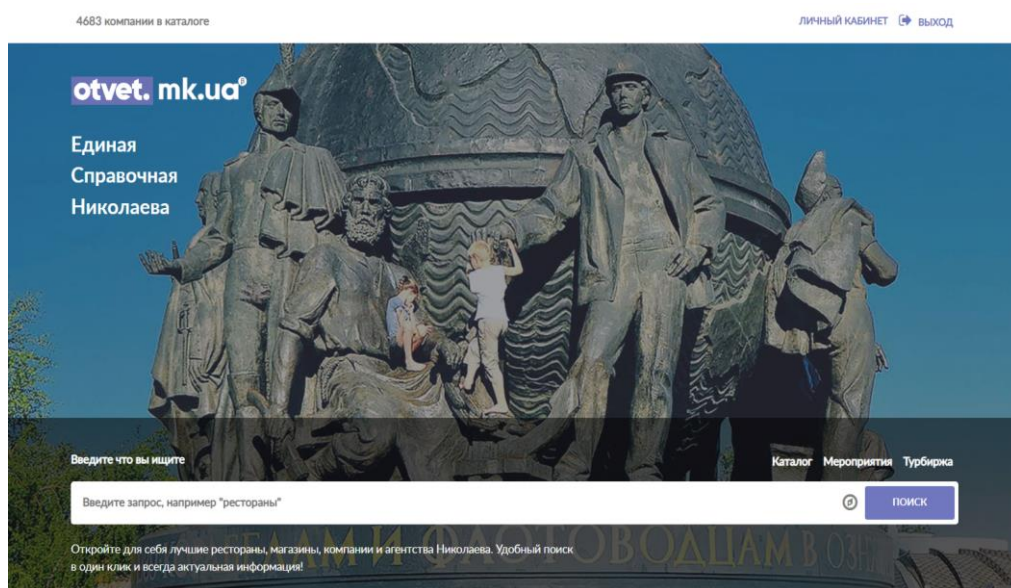


Рис. В.4 – Пошукова строка

2. У каталозі компаній 4683 заклади, які розподілені на 32 головні категорії. Користувач в залежності від зацікавленості може обрати категорію, натиснувши на її вкладку (рис. В.5).

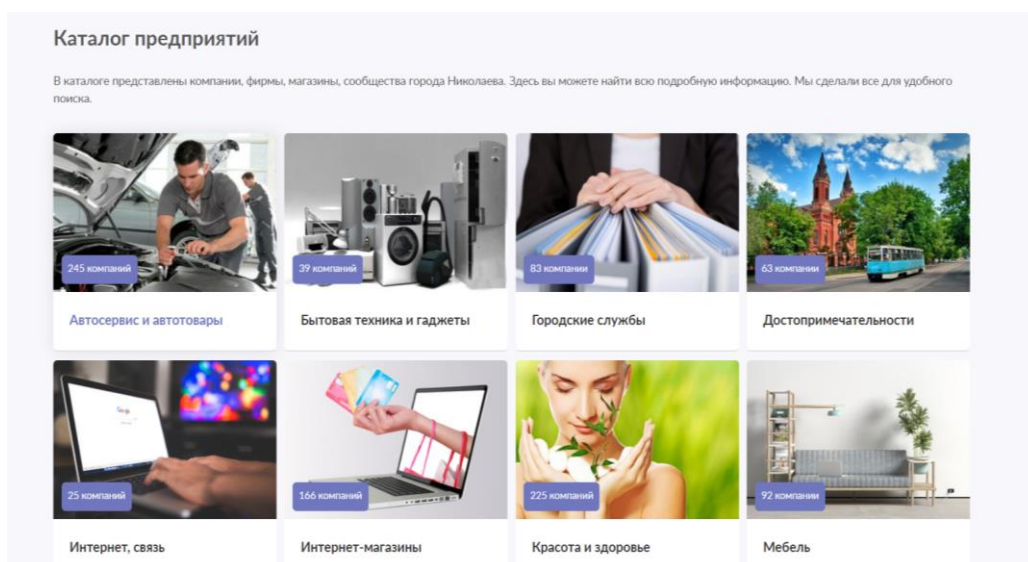


Рис. В.5 – Категорії компаній

3. Обравши головну категорію, користувач може ознайомитися із повним списком компаній та перейти на сторінку компанії, або обрати субкатегорію (рис. В.6)

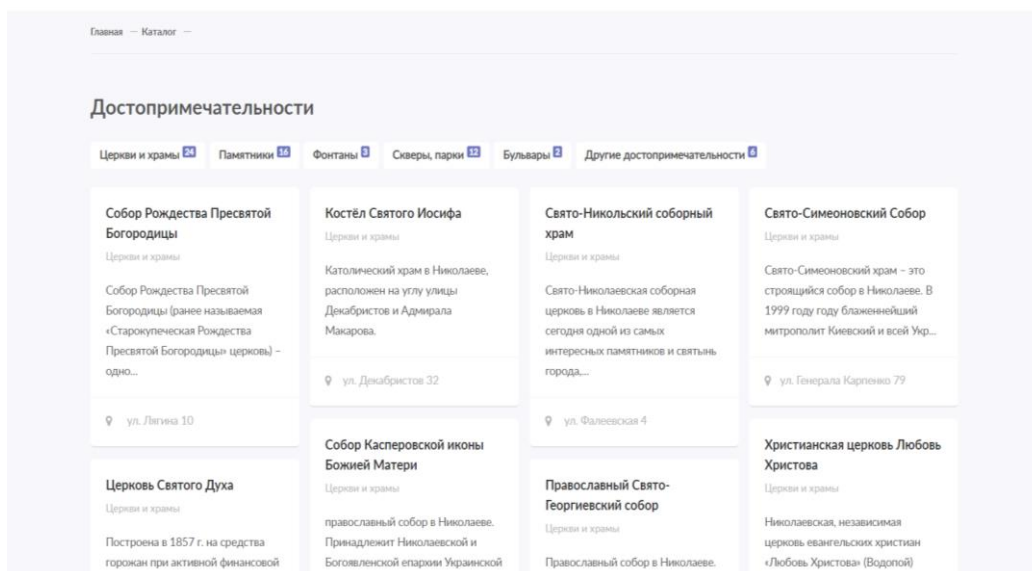


Рис. В.6 – Список компаній за категорією

4. Кожен заклад має назву, адресу, опис діяльності та місцезнаходження на Google Maps (рис. В.7)

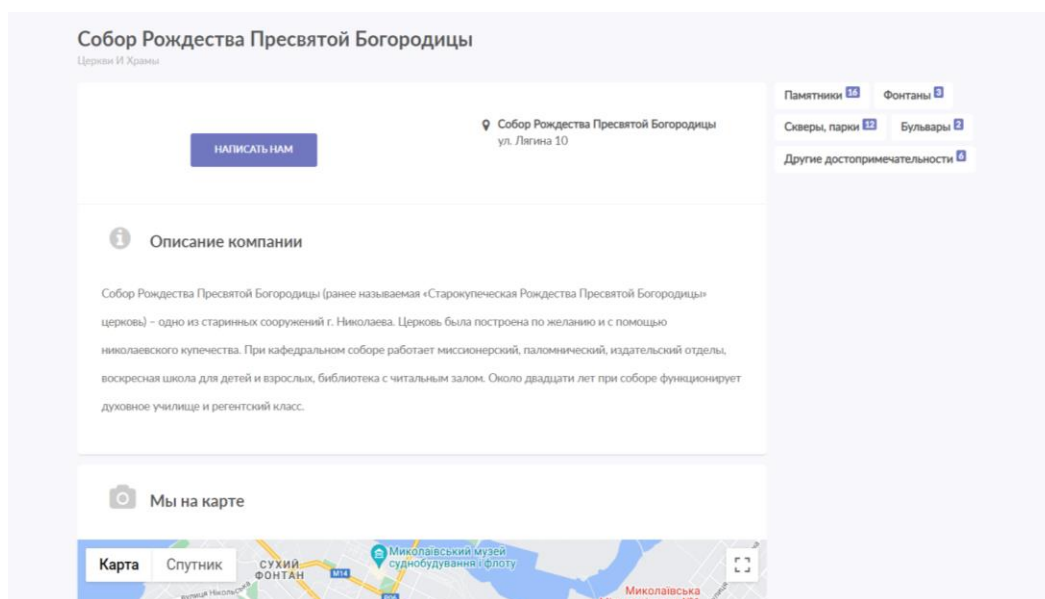


Рис В.7 – Вкладка компанії

Додати компанію та захід

1. Щоб переглянути вже додані компанії або додати нову, користувач повинен натиснути на відповідну іконку зліва у особистому кабінеті. Далі обрати категорію компанії, логотип, додати назву та опис (рис. В.8).

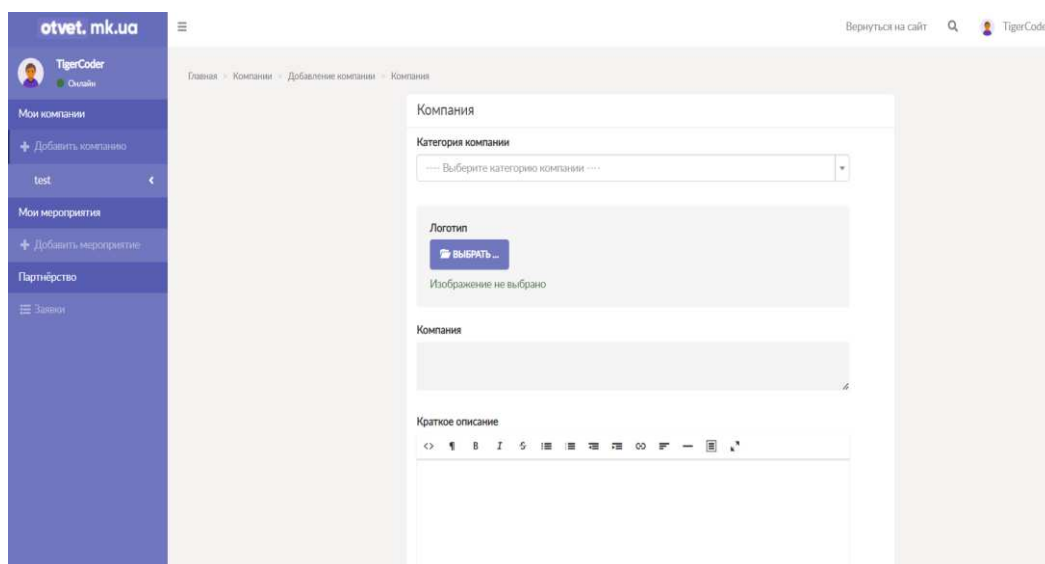


Рис. В.8 – Вкладка «Додати компанію»

2. Щоб переглянути вже додані заходи або додати новий, користувач повинен натиснути на відповідну іконку зліва у особистому кабінеті. Далі

обрати категорію заходу, логотип, додати назву, опис, місце та дату проведення, вартість квитка, кількість місць, вікове обмеження (рис. В.9-В.10).

The screenshot shows the 'Мероприятие' (Event) form in the 'Добавление мероприятия' (Add Event) section. The left sidebar contains the user profile 'TigerCoder' and navigation links: 'Мои компании' (My Companies), 'Мои мероприятия' (My Events), 'Партнёрство' (Partnership), and 'Заявки' (Applications). The main form includes the following fields:

- Категория мероприятия** (Event Category): A dropdown menu with the placeholder text 'Выберите категорию мероприятия'.
- Изображение** (Image): A section with a 'ВЫБРАТЬ...' (SELECT...) button and the text 'Изображение не выбрано' (Image not selected).
- Название мероприятия** (Event Name): A text input field.
- Описание мероприятия** (Event Description): A rich text editor with a toolbar containing various formatting options.

Рис. В.9 – Вкладка «Додати захід»

This screenshot shows the lower portion of the 'Add Event' form. It includes the following fields:

- Email**: A text input field.
- Место проведения** (Event Location): A dropdown menu with the placeholder text 'Выберите место проведения'.
- Даты проведения** (Event Dates): A section with a checkbox, a date selection interface, and a green '+ Добавить' (Add) button.
- Плата за посещение** (Event Fee): A dropdown menu with the placeholder text 'Выберите тип мероприятия'.
- Кол. мест** (Number of Seats): A dropdown menu with the placeholder text 'Выберите тип мероприятия'.
- Возрастное ограничение** (Age Restriction): A dropdown menu with the placeholder text 'Выберите ограничение по возрасту'.
- СОХРАНИТЬ** (SAVE): A blue button at the bottom of the form.

Рис. В.10 – Вкладка «Додати захід»

Завершення роботи

1. Для завершення роботи у системі необхідно натиснути на іконку особистого профілю у правому верхньому кутку та обрати вкладку «Вийти» (рис. В.11).

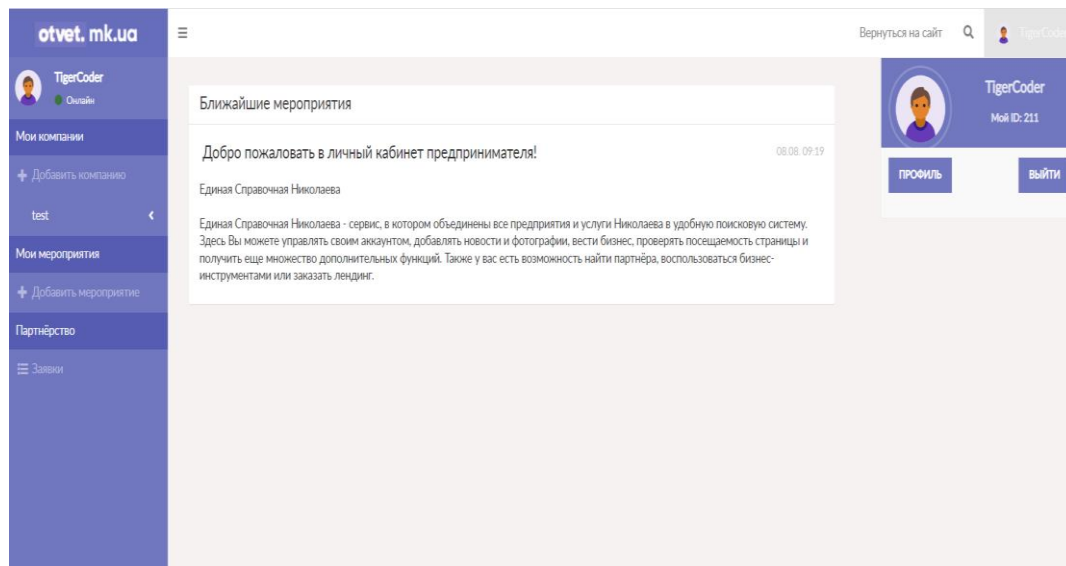


Рис. В.11 – Завершення роботи

ДОДАТОК Г

ПРОГРАМА І МЕТОДИКА ВИПРОБУВАНЬ

Об'єкт випробувань

Об'єктом випробувань є ІС «Єдина довідкова Миколаєва», яка призначена для пошуку та розміщення інформації щодо компаній та заходів міста.

Цілі випробувань

Цілі випробувань: перевірка придатності ІС до використання по призначенню, її відповідність заданим вимогам; перевірка правильності роботи системи.

Вимоги до системи

Повний список вимог до системи приведений у пункті 1.4 пояснювальної записки. Нижче приводяться функціональні вимоги до системи.

Система повинна задовольняти наступним функціональним характеристикам:

- 1) Аутентифікація користувача.
- 2) Ідентифікація користувача.
- 3) Конфіденційність передачі даних по комп'ютерній мережі.
- 4) Ведення бази компаній:
 - додавання нової компанії – створення вкладки;
 - редагування інформації про існуючу компанію;
 - пошук компанії.
- 5) Ведення бази заходів:
 - додавання нового заходу – створення вкладки;
 - редагування інформації про існуючий захід;
 - пошук заходу.
- 6) Створення запитів та звітів:

- створення запиту та звіту про компанії;
- створення запиту та звіту про заходи.

7) Взаємодія з Google Maps:

- пошук місцезнаходження користувача;
- пошук місцезнаходження компанії;
- пошук місцезнаходження заходу.

Вимоги до програмної документації

У комплект програмної документації повинні входити наступні документи:

- технічне завдання;
- загальний опис системи;
- інструкція користувача;
- програма і методика випробувань;
- програмний код.

Засоби і порядок випробувань

Вимоги до технічних і програмних засобів вказані в пункті 1.4. Оскільки ІС працює з використанням веб-технологій, вона не потребує інсталяції. Проте, для використання необхідний веб-браузер, який підтримує актуальні веб-стандарти. Також необхідно мати стабільний доступ до інтернету з швидкістю не менше ніж 10 Мбіт/сек.

Випробування системи будуть проводитися протягом трьох днів. Розробник самостійно проводить тестування усіх функцій системи. При наявності відповідної можливості, у процесі тестування можуть додатково брати участь програмісти, тестувальники, які не брали участь у розробці системи.

Порядок проведення випробувань:

- виконуються контрольні тести в довільному порядку. Результати обов'язково фіксуються.
- проводиться аналіз результатів і встановлюється відповідність системи заданим вимогам і системним документам.

При виявленні помилок у роботі системи складається їхній перелік і визначається термін їхнього виправлення. Після цього розробник проводить повторне тестування в повному обсязі.

Методика випробувань

Процес тестування призначений для виявлення інформації про ІС у середовищі, в якому її мають використовувати. Тестування має виявити відповідність системи вимогам, якими керувалися розробники; правильність відповіді для всіх можливих вхідних даних; виконання функцій за прийнятний час.

Найкраще застосовувати методи статичного та динамічного тестування.

Статичне тестування передбачає перевірку програмних кодів, контроль та перевірку системи без запуску на комп'ютері. Тестова діяльність, що передбачає експлуатацію програмного продукту, називається динамічним тестуванням. Динамічне та статичне тестування доповнюють одне одного.

На етапі статичного тестування перевіряється вся документація, отримана як результат життєвого циклу системи. Це і технічне завдання, і вихідний текст програми на мові програмування. Вся документація аналізується на предмет дотримання стандартів програмування. У результаті статичної перевірки встановлюється, наскільки програма відповідає заданим критеріям та вимогам.

Динамічний метод застосовуються в процесі безпосереднього виконання програми. Коректність системи перевіряється на безлічі тестів та наборів підготовлених вхідних даних. При прогоні кожного тесту збираються та аналізуються дані про відмови та збої в роботі системи.

Програма випробувань

Тести складаються розробником. При цьому перевіряється функціонування системи з нормальними і помилковими даними і подіями.

Наприклад:

- Ідентифікація користувача з вірними даними.
- Ідентифікація користувача з невірними даними.

Неправильні дані можуть бути такими:

- логін і/або пароль користувача, яких немає у файлах аутентифікації (для перших двох тестів).
- неіснуюча дата (для всіх інших тестів).

ІС розроблена так, що користувач не може ввести в систему невірні дані через наданий інтерфейс. Таким чином, будь-які дані, що не є неправильними, потрібно вважати правильними.

Вихідні дані чи результат при правильних даних повинні відповідати виконанню функції, для тестування якої створений тест; при неправильних даних – відмовлення в доступі до системи чи повідомлення про помилку.

ДОДАТОК Д

ПРОГРАМНИЙ КОД

ApiKey

```
<?php
namespace common\models;

use yii\db\ActiveRecord;

/**
 * @property int $id [int(11)]
 * @property int $user_id [int(11)]
 * @property string $key
 * @property string $permissions
 */
class ApiKey extends ActiveRecord
{
    public static function tableName()
    {
        return 'api_keys';
    }

    private $preparedJson = null;

    private function prepare()
    {
        if ($this->preparedJson===null){
            $this->preparedJson = json_decode($this->permissions);
        }
    }
    public function getPermissions()
    {
        $this->prepare();
        return $this->preparedJson;
    }

    public function hasPermission($permission)
    {
        $this->prepare();
```

```

        return in_array($permission,$this->preparedJson);
    }

    public function addPermission($permission)
    {
        $this->prepare();
        $this->preparedJson = array_merge($this->preparedJson,[$permission]);
    }

    public function save($runValidation = true, $attributeNames = null)
    {
        $this->prepare();
        $this->permissions = json_encode($this->preparedJson);
        return parent::save($runValidation, $attributeNames);
    }
}

```

Company

```
<?php
```

```
namespace common\models;
```

```
use common\behaviors\TextPurifier;
use common\models\CompanyWebsite;
use common\models\Tags;
```

```
use cornernote\linkall\LinkAllBehavior;
use leinonen\Yii2Algolia\ActiveRecord\Searchable;
use leinonen\Yii2Algolia\ActiveRecord\SynchronousAutoIndexBehavior;
use leinonen\Yii2Algolia\SearchableInterface;
use Yii;
use yii\behaviors\TimestampBehavior;
use yii\db\Expression;
use yii\helpers\HtmlPurifier;
use yii\helpers\Url;
use yii\helpers\VarDumper;
```

```
/**
```

```
 * This is the model class for table "company".
```

```
 *
```

```
 * @property integer $company_id
```

```

* @property integer $city_id
* @property integer $category_id
* @property string $company
* @property string $email
* @property integer $owner_id
* @property integer $company_ownership_form_id
* @property integer $status_id
* @property integer $tariff_id
* @property integer $created_at
* @property integer $updated_at
* @property string $slug
* @property string $moderated
* @property string $brief
* @property string $note
* @property string $logo
*
* @property City $city
* @property CompanyCategory $category
* @property User $owner
* @property CompanyStatus $status
* @property CompanyTariff $tariff
* @property CompanyAddress[] $companyAddresses
* @property CompanySocial[] $companySocials
* @property CompanyTel[] $companyTels
* @property CompanyToKeywords[] $companyToKeywords
* @property Keywords[] $keywords
* @property Seo $seo
* @property CompanyWebsite[] $companyWebsites
* @property CompanyOwnershipForm[] $companyOwnershipForm
* @property mixed $websites
* @property \yii\db\ActiveQuery $tags
* @property null|string $companyLogo
* @property array $news
* @property array $socials
* @property mixed $algoliaRecord
* @property array $tels
* @property string $categoryName
* @property string $companyUrl
* @property null $companyName
* @property array $slides
* @property array $addresses
* @property string $domain [varchar(255)]
*/

```

```

class Company extends \yii\db\ActiveRecord implements SearchableInterface

```

```

{
    use Searchable;
    public $tag_ids;
    public $keyword_ids;
    public $image;

    CONST STATUS_PUBLISHED = 1;

    CONST SCENARIO_CREATE = 'create';
    CONST SCENARIO_UPDATE = 'update';
    /**
     * @inheritdoc
     */
    public static function tableName()
    {
        return 'company';
    }

    private static function formatMethers($distance)
    {
        // var_dump( $distance);
        switch (floor($distance)){
            case $distance<1000:{
                return round($distance).' м.';
                break;
            }
            // case ( $distance>1000 && $distance<1100 ): {
            //     return '1 км. ' .(round($distance) - 1000).' м.';
            //     break;
            // }
            default:{
                return round($distance/1000).' км.';
            }
        }
    }

    public function scenarios()
    {
        $scenarios = parent::scenarios();

        $scenarios[self::SCENARIO_CREATE] = ['category_id', 'company',
        'brief', 'email', '!owner_id'];
    }
}

```



```

        $scenarios[self::SCENARIO_UPDATE] = ['category_id', 'company',
'brief', 'email'];

        return $scenarios;
    }

    /**
     * @inheritdoc
     */
    public function rules()
    {
        return [
            [['city_id', 'owner_id', 'category_id', 'status_id', 'tariff_id',
'company_ownership_form_id'], 'integer'],
            [['company', 'category_id'], 'required'],
            [['city_id'], 'default', 'value' => 1],
            [['status_id'], 'default', 'value' => CompanyStatus::PUBLISHED],
            [['tariff_id'], 'default', 'value' => CompanyTariff::BASE],
            [['company', 'brief'], 'string'],
            [['email'], 'email'],
            [['owner_id'], 'default', 'value' => null],
            [['company_ownership_form_id'], 'default', 'value' => null],
            //[['moderated'], 'date', 'format' => 'Y-m-d'],
            [['moderated'], 'default', 'value' => null],
            [['tag_ids', 'keyword_ids', 'created_at', 'updated_at'], 'safe'],
            [['slug'], 'string', 'max' => 128],
            [['note'], 'string', 'max' => 255],
            [['city_id', 'exist', 'skipOnError' => true, 'targetClass' =>
City::className(), 'targetAttribute' => ['city_id' => 'city_id']],
            [['owner_id', 'exist', 'skipOnError' => true, 'targetClass' =>
User::className(), 'targetAttribute' => ['owner_id' => 'id']],
            [['status_id'], 'exist', 'skipOnError' => true, 'targetClass' =>
CompanyStatus::className(), 'targetAttribute' => ['status_id' => 'id']],
            [['tariff_id'], 'exist', 'skipOnError' => true, 'targetClass' =>
CompanyTariff::className(), 'targetAttribute' => ['tariff_id' => 'tariff_id']],
            [['category_id'], 'exist', 'skipOnError' => true, 'targetClass' =>
CompanyCategory::className(), 'targetAttribute' => ['category_id' =>
'category_id']],
            [['company_ownership_form_id'], 'exist', 'skipOnError' => true,
'targetClass' => CompanyOwnershipForm::className(), 'targetAttribute' =>
['company_ownership_form_id' => 'form_id']],
            [['logo'], 'string', 'max' => 255],
            [['image'], 'safe'],

```

```

        [['image'], 'file', 'extensions' => ['png', 'jpg', 'jpeg'], 'maxSize' =>
500*500],
        [['domain'],'string']
    ];
}

/**
 * { @inheritdoc }
 */
public function getAlgoliaRecord()
{
    $record = $this->toArray();

    //if($this->isRelationPopulated('companyTels')) {

        $array = array_map(function($e) {
            return is_object($e) ? $e->tel : $e['tel'];
        }, $this->companyTels);

        $record['companyTels'] = $array;
    //}

    //if($this->isRelationPopulated('companySocials')) {

        $array = array_map(function($e) {
            return is_object($e) ? $e->social : $e['social'];
        }, $this->companySocials);

        $record['companySocials'] = $array;
    // }

    //if($this->isRelationPopulated('companyAddresses')) {

        $array = array_map(function($e) {
            return $e->getGeoloc();
        }, CompanyAddress::find()->where(['is not', 'lat', null])->andWhere(['is
not', 'lng', null])->andWhere(['company_id'=>$this->company_id])->all());

        $record['_geoloc'] = $array;
    //}

```

```

$array = array_map(function($e) {
    return is_object($e) ? $e->website : $e['website'];
}, $this->companyWebsites);

$record['companyWebsites'] = $array;

$record['category_id'] = $this->category->category;

if (isset($this->seo)) {
    $record['keywords'] = $this->seo->keywords;
}

//die(var_dump($record));

/*    if($this->isRelationPopulated('companyWebsites')) {
        $record['companyWebsites'] = $this->companyWebsites->toArray();
    }*/

return $record;
}

public function viewDistance()
{
    // $company->companyAddresses[0]->street->street.' '.$company-
    >companyAddresses[0]->address
    // $a = acos(sin(pi()/180*$ILat)* sin(pi()/180*$Lat) +
    cos(pi()/180*$ILat)* cos(pi()/180*$Lat)* cos(pi()/180*($ILng-$Lng)));
    $lat = Yii::$app->session->get('lat',"");
    $lng = Yii::$app->session->get('lng',"");

    if ($lat=="||$lng==" )return $this->companyAddresses[0]->street-
    >street.' '.$this->companyAddresses[0]->address ;

    $adress = CompanyAddress::find()-
    >select(['company_address.*','distance'=>"round(6371000.*acos(sin(PI()*lat/180)*si
    n(PI()*$lat/180)+cos(PI()*lat/180)*cos(PI()*$lat/180)*cos(PI()*(lng-$lng)/180))"])-
    >where(['company_id'=>$this->company_id])->orderBy("distance")->asArray()-
    >one();

```

```

        return Streets::findOne($adress['street_id'])->street.'
        '.$adress['address'].'<br> ~'.self::formatMethers($adress['distance']).";
    }
    /**
     * @inheritdoc
     */
    public function attributeLabels()
    {
        return [
            'company_id' => '№',
            'city_id' => 'Город',
            'company' => 'Компания',
            'email' => 'Email',
            'owner_id' => 'Владелец',
            'created_at' => 'Создано',
            'updated_at' => 'Обновлено',
            'slug' => 'Slug',
            'moderated' => 'Промодерировано',
            'brief' => 'Краткое описание',
            'category_id' => 'Категория компании',
            'categoryName' => 'Категория компании',
            'status_id' => 'Статус компании',
            'statusName' => 'Статус компании',
            'tariff_id' => 'Тариф',
            'company_ownership_form_id' => 'Форма собственности',
            'tariffName' => 'Тариф',
            'note' => 'Заметка',
            'logo' => 'Логотип',
            'image' => 'Логотип',
            'domain' => 'Домен'
        ];
    }
}

```

```

public function behaviors()
{
    return [
        LinkAllBehavior::className(),
        SynchronousAutoIndexBehavior::className(),
        [
            'class' => TimestampBehavior::className(),
            'createdAtAttribute' => 'created_at',

```

```

        'updatedAtAttribute' => 'updated_at',
        'value' => new Expression('NOW()'),
    ],
    'slug' => [
        'class' => 'common\behaviors\Slug',
        'in_attribute' => 'company',
        'out_attribute' => 'slug',
        'translit' => true
    ],
    [
        'class' => TextPurifier::className(),
        'attributes' => ['brief', 'note']
    ]
];
}
function closetags($html) {
    preg_match_all('#<(?!meta|img|br|hr|input\b)\b([a-z]+)(?:\s(?:.*))?(?<![/])>#iU', $html, $result);
    $openedtags = $result[1];
    preg_match_all('#</([a-z]+)>#iU', $html, $result);
    $closedtags = $result[1];
    $len_opened = count($openedtags);
    if (count($closedtags) == $len_opened) {
        return $html;
    }
    $openedtags = array_reverse($openedtags);
    for ($i=0; $i < $len_opened; $i++) {
        if (!in_array($openedtags[$i], $closedtags)) {
            $html .= '</'.$openedtags[$i].>';
        } else {
            unset($closedtags[array_search($openedtags[$i], $closedtags)]);
        }
    }
    return $html;
}
/**
 * @return \yii\db\ActiveQuery
 */
public function getCity()
{
    return $this->hasOne(City::className(), ['city_id' => 'city_id']);
}
public function getBrief(){
    return ($this->brief);
}

```

```

    }
    /**
     * @return \yii\db\ActiveQuery
     */
    public function getCategory()
    {
        return $this->hasOne(CompanyCategory::className(), ['category_id' =>
'category_id']);
    }

    /**
     * @return \yii\db\ActiveQuery
     */
    public function getCompanyOwnershipForm()
    {
        return $this->hasOne(CompanyOwnershipForm::className(), ['form_id'
=> 'company_ownership_form_id']);
    }

    /**
     * @return \yii\db\ActiveQuery
     */
    public function getOwner()
    {
        return $this->hasOne(User::className(), ['id' => 'owner_id']);
    }

    /**
     * @return \yii\db\ActiveQuery
     */
    public function getCompanyAddresses()
    {
        return $this->hasMany(CompanyAddress::className(), ['company_id' =>
'company_id']);
    }

    /**
     * @return \yii\db\ActiveQuery
     */
    public function getCompanySocials()
    {
        return $this->hasMany(CompanySocial::className(), ['company_id' =>
'company_id']);
    }

```

```

    }

    /**
     * @return \yii\db\ActiveQuery
     */
    public function getStatus()
    {
        return $this->hasOne(CompanyStatus::className(), ['id' => 'status_id']);
    }

    /**
     * @return \yii\db\ActiveQuery
     */
    public function getTariff()
    {
        return $this->hasOne(CompanyTariff::className(), ['tariff_id' =>
'tariff_id']);
    }

    /**
     * @return \yii\db\ActiveQuery
     */
    public function getCompanyTels()
    {
        return $this->hasMany(CompanyTel::className(), ['company_id' =>
'company_id']);
    }

    /**
     * @return \yii\db\ActiveQuery
     */
    public function getCompanyToKeywords()
    {
        return $this->hasMany(CompanyToKeywords::className(),
['company_id' => 'company_id']);
    }

    /**
     * @return \yii\db\ActiveQuery
     */
    public function getKeywords()
    {

```

```

        return $this->hasMany(Keywords::className(), ['keyword_id' =>
'keyword_id'])->viaTable('company_to_keywords', ['company_id' => 'company_id']);
    }

    /**
     * @return \yii\db\ActiveQuery
     */
    public function getTags()
    {
        return $this->hasMany(Tags::className(), ['tag_id' => 'tag_id'])->
>viaTable('company_to_tags', ['company_id' => 'company_id']);
    }

    /**
     * @return \yii\db\ActiveQuery
     */
    public function getSeo()
    {
        return $this->hasOne(Seo::className(), ['post_id' => 'company_id'])->
>andOnCondition(['post_type_id' => PostType::COMPANY]);
    }

    /**
     * @return \yii\db\ActiveQuery
     */
    public function getCompanyWebsites()
    {
        return $this->hasMany(CompanyWebsite::className(), ['company_id' =>
'company_id']);
    }

    public function getCompanyName(){
        return (isset($this->category_id)) ? $this->category->category : null;
    }

    public function getCategoryName(){
        return $this->category->parent->category.' -> '.$this->category->category;
    }

    public function afterSave($insert, $changedAttributes)
    {
        $tags = [];
        $keywords = [];
    }

```



```

    if (!empty($this->tag_ids)) {
        foreach ($this->tag_ids as $tag_name) {
            $tag = Tags::getTagByName($tag_name);
            if ($tag) {
                $tags[] = $tag;
            }
        }
    }

    if (!empty($this->keyword_ids)) {
        foreach ($this->keyword_ids as $keyword_name) {
            $keyword = Keywords::getTagByName($keyword_name);
            if ($keyword) {
                $keywords[] = $keyword;
            }
        }
    }

    $this->linkAll('keywords', $keywords);
    $this->linkAll('tags', $tags);
    parent::afterSave($insert, $changedAttributes);
}

public static function getLastAddedCompanies() {
    $companies = self::find()
        ->leftJoin('company_status','company.status_id=company_status.id')
        ->andWhere(['company_status.show'=>1])
        ->andWhere(['!','=','logo',''])
        ->orderBy(['company_id' => SORT_DESC])
        ->limit(8)
        ->all();

    return (count($companies) > 0) ? $companies : null;
}

/**
 * @var $tel CompanyTel
 * @return array
 */
public function getTels(){
    $tels = [];
    $telsQ = CompanyTel::find()
        ->where(['company_id' => $this->company_id])
        ->andWhere(['address_id' => null])

```

```

->joinWith('messengers')
->all();

foreach ($telsQ as $tel) {
    $tels[] = [
        'tel' => $tel->tel,
        'messengers' => $tel->messengers
    ];
}
return $tels;
}

public function getWebsites(){
    $q = CompanyWebsite::find()
        ->where(['company_id' => $this->company_id])
        ->all();

    return $q;
}

public function getSocials(){
    $socials = [];
    $q = CompanySocial::find()
        ->where(['company_id' => $this->company_id])
        ->joinWith('socialType')
        ->all();
    foreach ($q as $social) {

        $socials[] = [
            'social' => $social->social,
            'social_type' => $social->socialType
        ];
    }

    return $socials;
}

public function getAddresses(){
    $info = [];
    $model = self::find()
        ->where(['company.company_id' => $this->company_id])
        ->joinWith('companyTels')
        ->joinWith('companyAddresses')
        ->one();
}

```

```

foreach ($model->companyAddresses as $address) {
    $info['addresses'][] = [
        'address' => $address->address,
        'street' => (isset($address->street)) ? $address->street->street : null,
        'place_id' => $address->place_id,
        'schedule' => $this->getSchedule($address),
        'tels' => CompanyTel::find()
            ->where(['company_id' => $this->company_id])
            ->andWhere(['address_id' => $address->company_address_id])
            ->joinWith('messengers')
            ->asArray()
            ->all()
    ];
}

return $info;
}

public function getSchedule($address) {
    /**
     * @var $address CompanyAddress
     */
    $temp = [];

    $schedule = [];

    $days = [
        'day1' => 'Пн',
        'day2' => 'Вт',
        'day3' => 'Ср',
        'day4' => 'Чт',
        'day5' => 'Пт',
        'day6' => 'Сб',
        'day7' => 'Вс',
    ];
    $empty = true;

    for ($i = 1; $i <= 7; $i++) {
        if (!empty($address->{'day'.$i})) {
            $temp[0] = $days["day$i"];
            $temp[1] = $days["day$i"];
            $temp[2] = $address->{'day'.$i};
        }
    }
}

```

```

        $empty = false;
    }

    break;
}

if ($empty === false) {

    for ($i = 2; $i <= 7; $i++) {
        $z = $i-1;
        if ((($address->{'day'.$i} === $address->{'day'.$z}) && (($address->{'day'.$i} === $address->{'day'.$z}) && ($address->{'day'.$i.'_break'} === $address->{'day'.$z.'_break'}))) && ($i !== 7))) {
            $temp[1] = $days['day'.$i];
        } else {
            if ($i === 7) {

                if (($address->{'day'.$i} !== $address->{'day'.$z}) || ($address->{'day'.$i.'_break'} !== $address->{'day'.$z.'_break'}))) {

                    $daysSchedule = ($temp[0] === $temp[1]) ? $temp[0] :
                    $temp[0].' - '.$temp[1];
                    $schedule[] = $this->getScheduleMini($address, $i,
                    $daysSchedule, $temp, $days);

                } else {

                    $daysSchedule = ($temp[0] === $days['day'.$i]) ? $temp[0] :
                    $temp[0].' - '.$days['day'.$i];
                    $schedule[] = $this->getScheduleMini($address, $i,
                    $daysSchedule, $temp, $days);

                }

                return $schedule;
            }
        }
    }

    $daysSchedule = ($temp[0] === $temp[1]) ? $temp[0] : $temp[0].'
- '.$temp[1];
    $schedule[] = $this->getScheduleMini($address, $i,
    $daysSchedule, $temp, $days);
}

```

```

    }
    }
    return $schedule;

}

public function getScheduleMini ($address, $i, $daysSchedule, &$temp,
$days) {

    $dayBreak    =    (!empty($address->{'day'.$i.'_break'}))    ||    $address->{'day'.$i.'_break'}    !==    '-'    ||    $address->{'day'.$i.'_break'}    !==    null)    ?    ' Перерыв: ' . $address->{'day'.$i.'_break'}    :    '';

    $schedule = [
        'days'=> $daysSchedule,
        'schedule'=> $this->getScheduleText($temp[2], $dayBreak)

    ];

    $temp[0] = $days['day'.$i];
    $temp[1] = $days['day'.$i];
    $temp[2] = $address->{'day'.$i};

    return $schedule;
}

function getScheduleText($temp, $dayBreak) {
    if ($temp.$dayBreak === '- Перерыв: -') {
        return 'выходной';
    }

    if ((trim($dayBreak) === 'Перерыв:') || (trim($dayBreak) === 'Перерыв:
-')) {
        return $temp.' (без перерыва)';
    }

    return $temp.$dayBreak;
}

```

```

public function getNews(){

    $total = CompanyNews::find()
        ->where(['company_id' => $this->company_id])
        ->count();

    $next = ($total > 2) ? true : false;

    $news = CompanyNews::find()
        ->where(['company_id' => $this->company_id])
        ->orderBy(['id' => SORT_DESC])
        ->limit(2)
        ->all();
    return [
        'data' => $news,
        'next' => $next,
        'total' => $total
    ];
}

public function getSlides(){
    $total = CompanySlides::find()
        ->where(['company_id' => $this->company_id])
        ->count();
    $next = ($total > 3) ? true : false;
    $slides = CompanySlides::find()
        ->where(['company_id' => $this->company_id])
        ->orderBy(['id' => SORT_DESC])
        ->limit(3)
        ->all();

    return [
        'data' => $slides,
        'next' => $next,
        'total' => $total
    ];
}

public function getCompanyId(){
    if ($this->tariff_id!=CompanyTariff::BASE&&$this->tariff_id!=0&&$this->domain!=null){
        $domain = $this->domain;
        if (!(substr($domain, 0, 7 ) === "http://")&&!(substr($domain, 0, 7 ) === "https://"))
    }
}

```

```

        $domain = 'http://'.$domain;

        return $domain;
    }else{
        return      Url::to([Yii::$app->params['companies_catalog_base'].$this->category->parent->slug.'/'.$this->category->slug.'/'.$this->slug]);
    }
}

public function getCompanyUrlInConsole()
{
    return      'https://otvet.mk.ua/companies/catalog/'.$this->category->parent->slug.'/'.$this->category->slug.'/'.$this->slug;
}

public function getCompanyLogo(){
    $withoutExt      =      Yii::getAlias('@companies-logos-path')..'/'.$this->company_id;
    $url = Yii::getAlias('@companies-logos-url')..'/'.$this->company_id;
    if (!empty($this->logo) && file_exists(Yii::getAlias('@companies-logos-path')..'/'.$this->logo)) {
        return Yii::getAlias('@companies-logos-url')..'/'.$this->logo;
    } elseif (file_exists($withoutExt.'.png')) {
        return $url.'.png';
    } elseif (file_exists($withoutExt.'.jpg')) {
        return $url.'.jpg';
    } elseif (file_exists($withoutExt.'.jpeg')) {
        return $url.'.jpeg';
    } elseif (file_exists($withoutExt.'.webp')) {
        return $url.'.webp';
    } else {
        return null;
    }
}

public function getCompanyPageViewAnalytics($date_start = null,
$dimensions = 'ga:pagePath'){
    $results = [];

    $data = \Yii::$app->{'google-api'}
        ->analytics
        ->data_ga
        ->get(
            "ga:169966338",

```

```

        $date_start,
        "today",
        "ga:pageviews",
        [
            "dimensions" => $dimensions,
            'filters' => 'ga:pagePath=='.$this->getCompanyId()
        ]
    );

    if ($dimensions === 'ga:pagePath') {
        return (isset($data->rows[0][1]) && $data->rows[0][1] !== null) ?
        $data->rows[0][1] : 0;
    }

    if ($dimensions === 'ga:pagePath,ga:day') {
        $visitors = [];
        if (isset($data->rows )) {
            foreach ($data->rows as $key => $row) {
                $visitors[] = [
                    strtotime(date('Y').'-'.date('m').'-'. $row[1])*1000,
                    (int)$row[2]
                ];
            }

            $results[] = [
                'name' => 'Посетители',
                'data' => $visitors
            ];
        }
    }

    return $results;
}

public function actionAnalyticsById($google_id, $date_start = null,
$dimensions = 'ga:pagePath')
{
    $results = [];
    $additional = [
        "dimensions" => $dimensions,

        'filters'      =>      $google_id===self::GAI?'ga:pagePath=='.$this-
>getCompanyId():'

```



```

];
$data = \Yii::$app->{'google-api'}
->analytics
->data_ga
->get(
    $google_id,
    $date_start,
    "today",
    "ga:pageviews",
    $additional
);
if ($dimensions === 'ga:pagePath') {
    return (isset($data->rows[0][1]) && $data->rows[0][1] !== null) ?
$data->rows[0][1] : 0;
}

if ($dimensions === 'ga:pagePath,ga:day') {
    $visitors = [];
    if (isset($data->rows)) {
        foreach ($data->rows as $key => $row) {
            $visitors[] = [
                strtotime(date('Y').'-'.date('m').'-'. $row[1])*1000,
                (int)$row[2]
            ];
        }

        $results[] = [
            'name' => 'Посетители',
            'data' => $visitors
        ];
    }
}

return $results;
}
/*-----Website Analytic Id---*/
const GAI = "ga:169966338";
/*-----*/
//Websites and company page analytic for 1day 7day and 30days
public function getAllAnalytic()
{
    $results = [];

```

```

        $data = [
            'name' => $this->company,
            'status' => 200,
            'today' => $this->actionAnalyticsById(self::GAI,date('Y-m-d')),
            'days7' => $this->actionAnalyticsById(self::GAI,date('Y-m-d',
strtotime('-7 days'))),
            'days30' => $this->actionAnalyticsById(self::GAI,date('Y-m-d',
strtotime('-30 days'))),
            'chart' => $this->actionAnalyticsById(self::GAI,date('Y-m-d',
strtotime('-30 days')), "ga:pagePath,ga:day")
        ];
        $results[] = $data;

        foreach (CompanyWebsite::find()->where(['is not', 'gid', null])-
>andWhere(['company_id' => $this->company_id])->all() as $website) {
            try {
                $data = [
                    'name' => $website->website,
                    'status' => 200,
                    'today' => $this->actionAnalyticsById($website->gid,date('Y-m-
d')),
                    'days7' => $this->actionAnalyticsById($website->gid,date('Y-m-
d', strtotime('-7 days'))),
                    'days30' => $this->actionAnalyticsById($website->gid,date('Y-
m-d', strtotime('-30 days'))),
                    'chart' => $this->actionAnalyticsById($website->gid,date('Y-m-
d', strtotime('-30 days')), "ga:pagePath,ga:day")

                ];
            } catch (\Exception $e){
                $data = [
                    'name' => $website->website,
                    'status' => 403,
                ];
            }
            $results[] = $data;
        }
        return $results;
    }
}

```