

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: «Розробка он-лайн сервісу із
застосуванням технології веб-програмування, мова PHP»

Виконав(ла): студент(ка) VI курсу, групи СПм-61
спеціальності _____

121 – Інженерія програмного забезпечення

(шифр і назва спеціальності)

	_____	<u>Залізник В.В.</u>
	(підпис)	(прізвище та ініціали)
Керівник	_____	<u>Цуприк Г.Б.</u>
	(підпис)	(прізвище та ініціали)
Нормоконтроль	_____	<u>Бойко І.В.</u>
	(підпис)	(прізвище та ініціали)
Завідувач кафедри	_____	<u>Петрик М.Р.</u>
	(підпис)	(прізвище та ініціали)
Рецензент	_____	<u>Лупенко С.А.</u>
	(підпис)	(прізвище та ініціали)

Тернопіль 2020

АНОТАЦІЯ

Залізник В.В. Розробка он-лайн сервісу із застосуванням технології веб-програмування, мова PHP. – Рукопис.

Кваліфікаційна робота на здобуття освітнього ступеня магістр за спеціальністю 121 – Інженерія програмного забезпечення. – Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра програмної інженерії, група СПм-61 // м.Тернопіль, 2020 // С. , рис. – 19, табл. – 14, додат. – , бібліогр. –

Метою кваліфікаційної роботи є використанні сучасних інформаційних технологій для розробки веб-продукту, з можливістю врахування специфіки діяльності конкретного підприємства з метою його гідного представлення та популяризації серед бізнес-партнерів та підвищення конкурентоспроможності на світовому ринку.

Практичне застосування – розроблено надійний програмний продукт, що дозволить підвищити ефективність та продуктивність роботи та який є новим, унікальним та відповідає цілям свого замовника.

Технічні вимоги – методи розробки базується на технології PHP та фреймворк Laravel, сервер бази даних MySQL.

Ключові слова: ВЕБ-ДИЗАЙН, КЛІЄНТ, СЕРВЕР, PHP, LARAVEL, MVC.

ABSTRACT

Zaliznyak V.V. Development of the online service using web programming technology and PHP programming language. – Manuscript.

The master degree thesis for the qualification level of magistr in the specialty 121 — Software Engineering. – Ternopil Ivan Pul'ui National Technical University, Faculty of Computer Information Systems and Software Engineering, Software Engineering Department, group SPm-61 // Ternopil, 2020 //

Pages. – , pictures. – 19, tables. – 14, supp. – , bibl.ref. –

The purpose of the qualification work is to use modern information technologies for web product development. It will give the ability to take into account the specifics of a particular enterprise and to properly represent and progressing it among business partners. Also it will increase competitiveness in the global market.

Practical application - the reliable software product has been developed. It will allows to increase the efficiency and productivity of work. It is new, unique and meets the goals of its orderer.

Technical requirements - development methods based on PHP technologies and Laravel framework and MySQL database server.

Keywords: WEB DESIGN, CLIENT, SERVER, PHP, LARAVEL, MVC.

ЗМІСТ

ВСТУП	7
1 Аналітичний огляд в області досліджень.	10
1.1 Аналіз та огляд предметної області	12
1.2 Постановка задачі	13
2 Розробка програмної системи	17
2.1 Актанти та варіанти використання	17
2.1.1 Опис варіантів використання	18
2.2 Архітектура системи	20
2.3 Технології розробки	22
2.4 Розробка бази даних	26
3 Реалізація програмної системи	33
3.1 Реалізація архітектурного шаблону	33
3.1.1 Опис реалізації моделей даних	33
3.1.2 Опис реалізації управління запитом користувача	35
3.1.3 Опис реалізації генерації HTML-сторінок на	39
основі	
переданих параметрів	
3.2 Використання програмної системи	41
3.2.1 Розгортання програмної системи та системні	41
3.2.2 Опис типових схем використання розробки	43
3.3 Тестування системи	53
3.3.1 Опис процесу тестування	53
3.3.2 Розробка тестів	55
4 Охорона праці та безпека в надзвичайних ситуаціях	60
4.1 Охорона праці	60
4.2 Безпека в надзвичайних ситуаціях	63

ВСТУП

В загальному про користувачів будь-якого програмного продукту, навіть побутового рівня, вже не говорячи про користувачів, які спеціалізуються в даному напрямку, сьогодні вже можна сказати, що це люди з базовими знаннями та навиками, яких важко здивувати якимось додатковими ефектами при оформленні чи форматуванні тексту чи графічного представлення. Що ж тоді вже говорити про представництва в всесвітній мережі організацій малого чи середнього (не говорячи вже про великий) бізнесу, яке повинне бути не просто формальністю, а ефективним інструментом реалізації підвищення показників успішного ведення діяльності, основним і зрозумілим критерієм оцінки якого є чистий прибуток за оптимальний строк. Варто також зазначити і те, що тенденційно вимоги та очікування суттєво змінюються. Користувачі стають все більш вибагливими і за свої гроші прагнуть отримати все більше та більше можливостей, від найпростіших, так званих базових (зворотній зв'язок, можливість, розсилання, пошук по сайту й багато чого іншого), до спеціалізованих, відповідно до специфіки діяльності. І в принципі це логічно, адже саме для цього і існує веб-програмування в одному з його напрямків – створення веб-сайтів.

Однак, при використанні простого html досягнути цього не просто, оскільки html обмежується, як мова, лише розміткою, вважається інструментом для створення гіперпосилань, вставки зображень, можливо таблиць та подібне. Основним же, і вважаю, що найважливішим завданням веб-програмування є створення та реалізація інтерактивних компонентів, що є ефективно та обґрунтовано вимогами сьогодення.

Для реалізації ідеї, з можливістю врахування специфіки діяльності конкретного підприємства з метою його гідного представлення та популяризації серед бізнес-партнерів та підвищення конкурентоспроможності у всьому світі, використано серверну мову програмування Hypertext

Preprocessor (PHP), як скриптову та призначену безпосередньо для генерування HTML-сторінок на стороні сервера, яка має простий синтаксис, характеризується високою швидкістю та підтримується зі сторони більшості хостингів. І вже тоді PHP інтерпретується веб-сервером в HTML-код, який вже напямку передається у браузер.

Важливим моментом в роботі є й те, що хоча тема не є новою, адже існує достатня кількість продуктів подібного типу, однак, варта наголосити, що кожен з них є новим, унікальним та відповідає цілям свого конкретного замовника, адже всі вони повинні співпадати та враховувати його потреби, вимоги та специфіку і реалізуватись з використанням оптимальних найсучасніших, на час розробки, інформаційних технологій. Також беззаперечним є й той факт, що успішність і затребуваність продукту і є критеріями оцінки його ефективності. Отже, актуальність теми є беззаперечною та на часі.

Об'єктом дослідження обрано актуальні спеціалізовані напрямки конкретної галузі та сучасні способи і види її представлення з використанням сучасних інформаційних технологій. Аналіз грамотності, оригінальності та естетичної привабливості, доступності, ергономічності, а також врахування особливостей та деталей, злагоженості роботи, надійності ефективності та безпеки.

Предметом дослідження визначено програмний веб-продукт у вигляді веб-сайту для конкретного підприємства, розроблений на замовлення та з врахуванням вимог сучасного інформаційного простору та з використанням сучасних технологій. Зокрема, висока швидкість, підтримка зі сторони більшості хостингів. Простота вставки сценарію PHP в код html-сторінки та відсутність необхідності в додаткових розширеннях браузера, а також зручна система керування.

Планується, що буде розроблено актуальний та надійний програмний продукт зі з сучасним та зрозумілим інтерфейсом, що в свою чергу допоможе

розширити коло користувачів. Крім цього враховано потребу забезпечити надійне з'єднання та безперебійність роботи продукту.

Науковою новизною варта вважати використання сучасних Web-технологій для розробки нового та унікального програмного продукту, з врахуванням вимог та специфіки діяльності конкретного підприємства.

Прикладною стороною результатів наукового дослідження може бути специфічна діяльність, при чому продукт можна і потрібно вдосконалювати та підлаштовувати під вимоги сьогодення.

Оформлення дипломної роботи виконуватиметься з врахуванням Державного стандарту України стосовно системи технологічної документації, її термінів та визначення; документації, а саме: звітів в галузі науки та техніки, їх структури та правил оформлення, а також Єдиної системи конструкторської документації та з врахуванням положень інших чинних стандартів.

1 АНАЛІТИЧНИЙ ОГЛЯД В ОБЛАСТІ ДОСЛІДЖЕНЬ.

Спочатку згадаємо, що по суті веб-сторінка – це файл, що складається з HTML-коду. За запитом користувача ці файли завантажуються на комп'ютер, де браузер спочатку обробляє їх, а потім відображає на екрані.



Рисунок 1.1 – Види веб-сторінок [1]

Для того щоб розширити функціональність веб-сторінки використовують веб-програмування. З його допомогою можна внести інтерактивність на сторінку: здатність реагувати на дії користувача. У веб-програмуванні використовують веб-мови, які здатні працювати з веб-технологіями.

У веб-програмуванні використовуються дві технології: клієнтська та серверна. Перша дозволяє вставляти в веб-сторінку код, який буде виконаний на комп'ютері користувача, а серверна технологія має на увазі, що код буде виконаний на сервері, а в браузері буде відображатися результат.

До клієнтських мов відносяться: JavaScript і VBScript. Додаючи до веб-сторінці їх код, можна отримати лише деяку інтерактивність на сторінці. Сторінка тепер буде реагувати не деякі дію користувача. Але клієнтські мови по

збавлені можливості працювати з файловою системою і з базою даних. Їх робота також залежить від типу і версії браузера використовуваного користувачем.

Серверна технологія була розроблена для зняття обмежень присутніх в клієнтській. Для цього код виконується на стороні сервера з правом доступу до файлової системи і баз даних. Принцип роботи цієї технології виглядає так: коли користувач звертається до будь-якої сторінки (натискаючи на посилання або набираючи її в адресному рядку браузера), то спочатку йде виконання програм пов'язаних з цією сторінкою і тільки потім користувачеві відсилається вже оброблена сторінка.

В даний час використовуються наступні серверні мови програмування: PHP, SQL, Perl, JAVA, Python, MySQL [2].

Таким чином веб-програмування - це: спосіб оновлення та поповнення інформації на сайті за допомогою CMS [3]. Крім цього, це: -спосіб та можливість урізноманітнити сайт, додавши на нього динамічні розділи і інтерактивні елементи: голосування, хмара тегів, можливість відправки електронного листи з сайту, гостьові книги, форуми, ігри, чати і багато іншого;

- можливість працювати з захищеною базою даних, в якій можна зберігати: статистичні дані, прайси, опису товарів, відгуки відвідувачів, фотоальбоми;

- інструмент для вирішення будь-яких завдань по розширенню функціональності сайту.

1.1 Аналіз та огляд предметної області

Досліджувана предметна область являє собою ринок нерухомості. Для обраної області досліджень ймовірними «кандидатами» на сутності в майбутній базі даних можуть бути: орендодавець, орендар, квартира, будинок, оголошення, площа. На даний час, обмін інформацією здійснюється за допомогою оголошень в періодичних виданнях, через агенції нерухомості та за допомогою дошок оголошень, розміщених в інтернеті. З цих факторів і впливають основні проблеми, які пов'язані з обміном інформації, а саме: недостатньо оперативний обмін інформацією між зацікавленими особами; неможливість пошуку потрібної інформації; велика кількість недостовірної інформації.

При аналізі предметної області було враховано фактори які безпосередньо стосується інформації, її обміну:

1. як зберігається інформації;
2. як часто оновлюються дані;
3. як відбувається обмін інформацію між суб'єктами ринку обраної об'єкту дослідження;
4. яка інформація важлива для користувача, модератора;
5. яку додаткову інформацію варто внести.

Основна причина проектування системи – забезпечення цілісності та достовірності даних, уникання можливої втрати даних, полегшення роботи суб'єктам ринку нерухомості, збереження даних та ефективний обмін інформацією.

З вищезгаданих проблем впливають наступні завдання, які можуть бути реалізовані:

1. Необхідно якимось чином надавати загальну інформації про нерухомість.
2. Потрібно зібрати інформації в одному місці, для забезпечення ефективного пошуку інформації.

3. Також необхідно забезпечити ефективний доступ, збереження, редагування оголошень про нерухомість.

4. Розпізнавання користувачів.

Для забезпечення коректної та правдивої інформації необхідне забезпечення інструментною модерацією доданих оголошень. Для коректної роботи необхідно додати можливість розпізнавання користувачів, мета – забезпечення додаткового функціоналу.

Для того, щоб визначити які саме переваги матиме кожен користувач, спочатку необхідно встановити якими є ті типові користувачі. З точки зору логіки, і з попереднього аналізу, це користувач і модератор. Крім них, можна згадати і користувача, який зможе поверхнево ознайомлюватись з інформацією про нерухомість, переглядати оголошення. Таким чином, було виявлені наступних користувачів: користувач, модератор, гість.

Користувач – той, хто додає оголошення. Дозволяє полегшити збереження, редагування даних, дозволить не носити з собою журнали, що зменшить до нуля можливість втрати даних, зменшить кількість запитань.

Модератор – той, хто перевіряє всі додані та відредаговані оголошення. Також модератор переглядає скарги на оголошення від інших користувачів. При невідповідності нормам оголошення може бути заблоковане.

Гість – той, хто може ознайомитись з оголошеннями.

1.2 Постановка задачі

Таким чином, після проведення аналізу предметної області в попередньому підрозділі і виявлення основних проблем, варта підсумувати та окреслити завдання, які потребують вирішення. Перше, і найголовніше завдання, – який тип бази даних необхідно обрати. Яким чином будемо з'єднуватися з базою даних, через конектори чи ORM. Також яким чином буде

проходити реєстрації користувачів в систему: самореєстрація чи за допомогою адміністратор через СУБД. Чи необхідні додаткові дані для внесення в базу даних, чи достатньо вже попередньо згаданих. Які процеси можна автоматизувати в даній предметній області. Чи необхідна взагалі інформаційна система даних предметній області. Чи можливо покращити процеси та які. Розглянути необхідність покращення процесів. Яким функціям кінцевий споживач (користувач) надасть перевагу. Чи варта затратити ресурси на ці процеси (чи настільки вони є потрібними кінцевому користувачеві). Чи буде інтерфейс інтуїтивно зрозумілим та зручним. Проаналізувати оптимальність числа сутностей в результаті аналізу предметної області. Чи достатньо повно вивчена предметна область, чи необхідно проводити додатковий аналіз. Цей перелік можна продовжувати довго, однак рамки роботи обмежені, тому, я прийняв рішення визначити, на мою думку, основні, і вже з ними працювати.

Отже, завданням дослідження є :

1. визначити оптимальний необхідний тип бази даних;
2. встановити тип з'єднання з базою даних;
3. визначитися, які дані зберігати;
4. запропонувати процеси для автоматизації та покращення;
5. обґрунтувати необхідні функції для користувача;
6. описати сутності бази даних і зв'язки між ними;
7. встановити чи є вся необхідна інформація;
8. сказати яким чином веб-сайт допоможе користувачам.

Далі необхідно визначити, які дані нам будуть потрібні для бази даних, оскільки перераховано основну, проте її може виявитись недостатньо.

Узагальню, тобто напишу, яка інформація виявиться необхідною в проєктованій базі даних:

1. про користувача;
2. про модератора;
3. про нерухомість;
4. про оголошення;
5. про скарги;
6. про перевірку оголошення;
7. про контактну інформацію користувачів;
8. про закладки користувачів;
9. про паролі користувачів.

З завдань для вирішень виявлено наступні функції системи:

1. вивід списку всіх оголошень та з використанням фільтрів;
2. перегляд оголошення;
3. створення, редагування та видалення оголошень;
4. перегляд списку своїх оголошень;
5. перегляд оголошень доданих до закладок;
6. створення закладок на оголошення;
7. можливість подати скаргу на оголошення;
8. перегляд оголошень на які було подано скаргу;
9. блокування оголошення;
10. вивід списку оголошень для перевірки;
11. можливість обміну повідомленнями між користувачами.

Основні запити які формує користувач будуть

1. здійснюватися через графічний інтерфейс,
2. надсилатись сервер для процесу обробки.

Після опрацювання результату на запит користувача буде надіслана відповідь.

2.1 Актанти та варіанти використання

Актанти (актори) в UML [4] – зовнішні відносно системи сутності, які співпрацюють з системою, або використовують її функціональні можливості. На основі аналізу ПО можна виділити таких акторів: гість (користувач не зареєстрований в системі), користувач, модератор.

Було виявлено три основних актори: гість, користувач, та модератор.

Гість може переглядати оголошення, увійти або зареєструватись в системі. Після реєстрації чи входу гості стають користувачами.

Користувач має можливість створювати, редагувати та видаляти оголошення, надсилати та отримувати повідомлення. Також додавати оголошення до закладок або подати скаргу.

Модератор може переглядати оголошення без перевірки та скарги користувачів, блокувати або верифікувати оголошення.

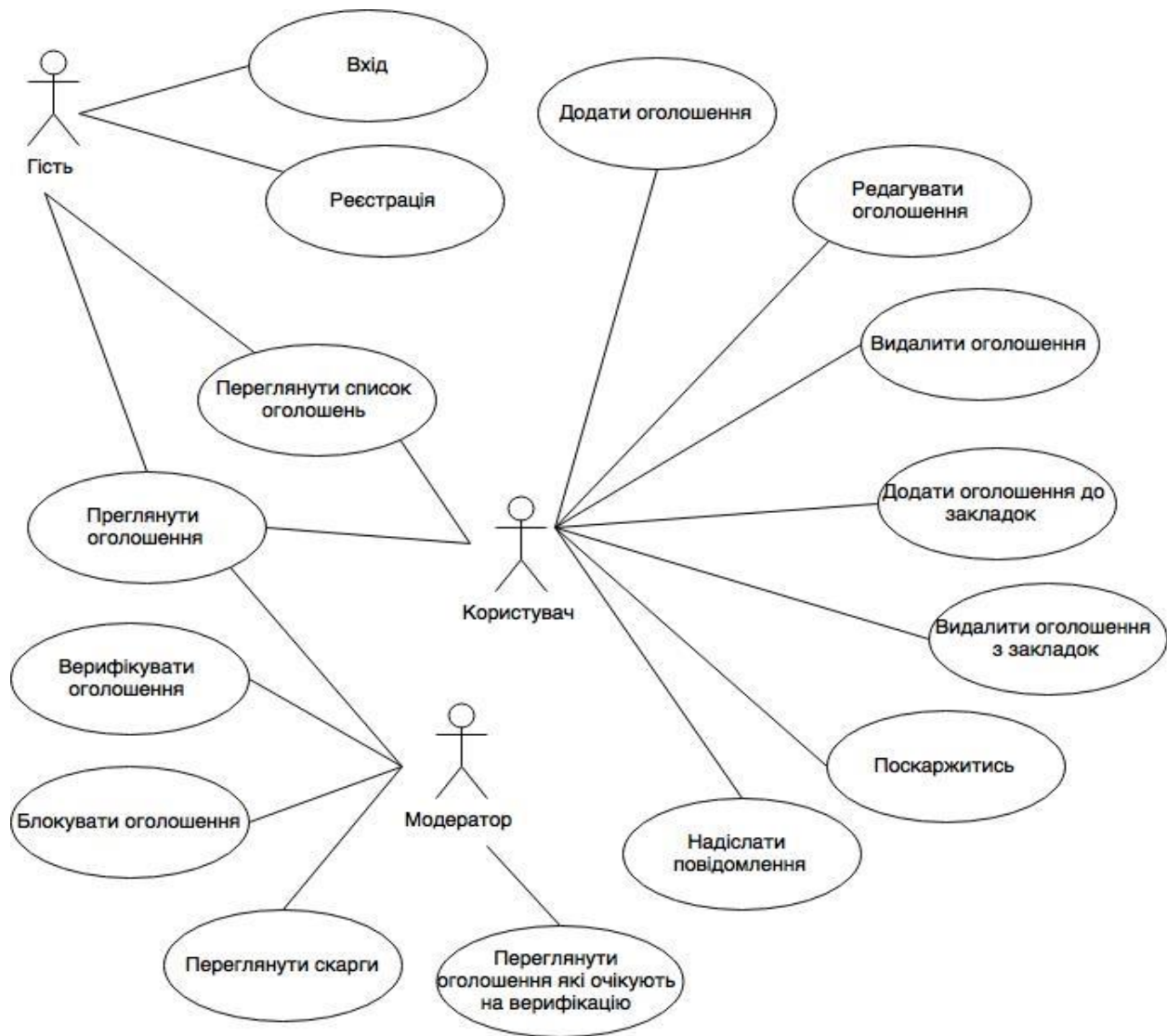


Рисунок 2.1 – Діаграма варіантів використання системи

2.1.1 Опис варіантів використання

Для кращого розуміння в даному підпункті детальніше опишу кожен з попередньо визначених варіантів використання.

Вхід – у випадку успішної автентифікації проводиться авторизація з подальшим допуском користувача до роботи в системі.

Реєстрація – створення нового користувача. У разі успішної реєстрації буде виконаний автоматичний вхід у систему.

Додати контактну інформацію – користувач може додати додаткову контактну інформацію, яка буде відображатись в оголошеннях доданих користувачем.

Переглянути оголошення – даний варіант використання дозволяє переглядати оголошення додані користувачами. Користувачу виводяться оголошення, які пройшли перевірку модераторами. У разі використання фільтру виведеться у відповідності до заданих критерій.

Додати оголошення – створення користувачем нового оголошення. Користувач вказує назву, опис та інформацію про нерухомість.

Редагувати оголошення – редагування даних вказаних в оголошенні користувачем.

Видалити оголошення – даний варіант використання дозволяє користувачу видалити своє оголошення.

Додати оголошення до закладок – створюється закладка на оголошення.

Переглянути закладки – дозволяє користувачу переглянути список оголошень доданих до закладок.

Видалити закладу – видаляється закладка на оголошення.

Переглянути повідомлення – даний варіант використання дозволяє користувачу переглянути список вхідних та надісланих повідомлень.

Видалити повідомлення – видалення повідомлення зі списку повідомлень.

Написати повідомлення – користувач надсилає повідомлення іншому зареєстрованому користувачу.

Переглянути скарги – даний варіант використання дозволяє модератору продивитись список оголошень, на які скаржились користувачі.

Переглянути оголошення, які потребують перевірки – перегляд модератором списку оголошень, які були додані або відредаговані користувачами.

Блокувати оголошення – модератор блокує оголошення через невідповідність нормам. У випадку блокування оголошення не буде відображатись у списку для інших користувачів.

Верифікувати оголошення – модератор дозволяє відображення оголошення у списку всіх оголошень.

2.2 Архітектура системи

Як найбільш вдалий з погляду реалізації для веб-сайту вважаю потрібно обрано архітектурний шаблон MVC [5], який безпосередньо і призначено для створення інтерактивних та динамічних сайтів.

Суть цієї моделі, яка так мені підійшла в наступному. Оскільки особливість статистичної сторінки на HTML відсутність реакції на дії користувачів, то виникає потреба в динамічних веб-сторінках. Саме вони і допоможуть двосторонньо взаємодіяти. Саме MVC – є ключем для розуміння процесу розробки веб-додатків з динамікою. MVC можна розшифрувати як Модель-Уявлення-Контролер (від англ. Model-View-Controller). Це спосіб організації коду, що передбачає виділення блоків, що відповідають за вирішення різних завдань. Один блок відповідає з а дані додатки, інший відповідає за зовнішній вигляд, а третій контролює роботу додатка. Компоненти MVC (рисунок 2.2):

-модель (компонент, який відповідає за дані, визначатиме структуру програмного продукту).

Модель відповідає за дані, які зберігаються і обробляються на сервері. Наприклад, User: - userName- firstName- lastName- friends

-уявлення (компонент, що відповідає за взаємодію з користувачем. Тобто код компонента view визначає зовнішній вигляд програми і способи його використання).

Це HTML-шаблон, який повертає сервер після обробки запиту. Якщо запит коректно обробляється, ви отримуєте веб-сторінку зі списком друзів. Якщо запит некоректний, ви потрапляєте на сторінку помилки 404. Наприклад,

```
<ul>
  <li>Friend 1: {friendList[0].userName}</li>
  <li>Friend 2: {friendList[1].userName}</li>
  <li>Friend 3: {friendList[2].userName}</li>
  ...
</ul>
```

-контролер (компонент, який відповідає за зв'язок між model і view.

Саме код компонента controller визначатиме, яким чином сайт реагуватиме на активність (дії) користувача. Таким чином, це мозок MVC-додатку.

Контролер обробляє вхідні запити. У фреймворку це може полягати у визначенні конкретних URL, на які потрапляє користувач при переході по посиланню або при натисканні кнопки. Наприклад,

```
Перехід за посиланням website (.) Com / profile / ->
повертає profilewebsite (.) com / friends / ->
повертає friendswebsite (.) com / friend = {userName} / ->
повертає профіль конкретного одного
```

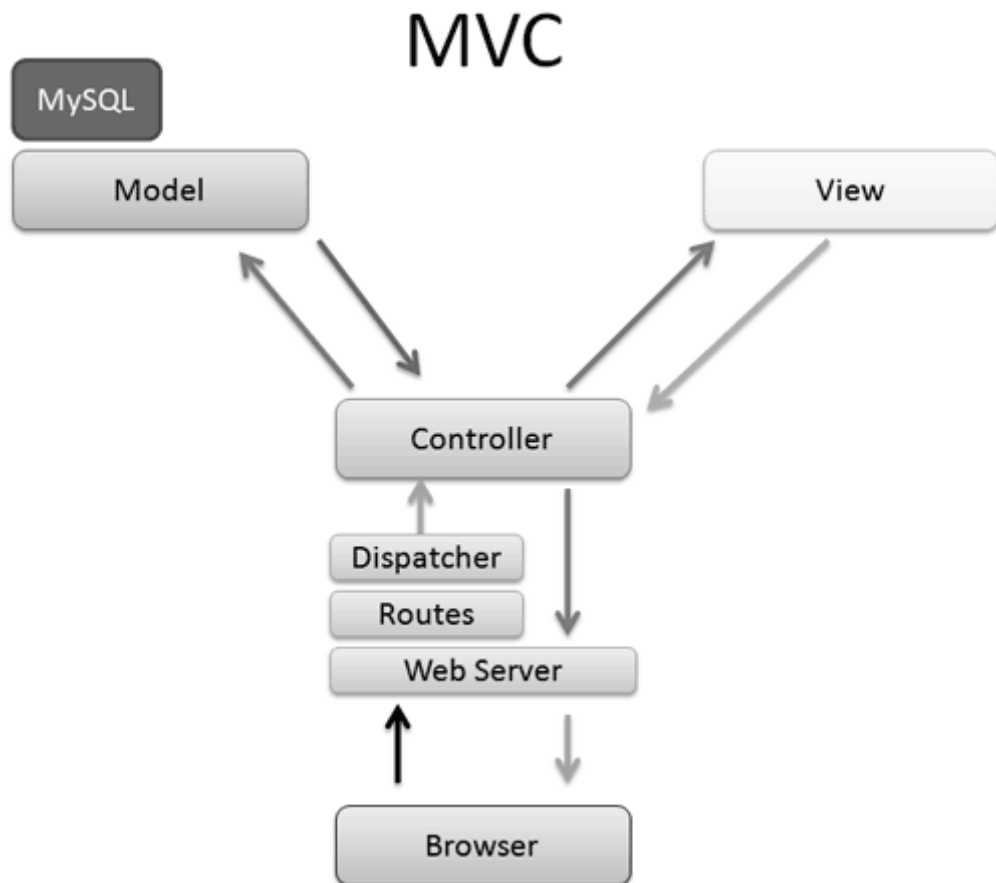


Рисунок 2.2 – Діаграма Модель-Уявлення-Контролер [6]

2.3 Технології розробки

В метюю збереження даних використано реляційну базу даних MSQL.

MySQL [7] – це швидка і проста у використанні СУБД, яка використовується для багатьох підприємств малого та середнього бізнесу. MySQL розробляється, продається і підтримується шведською компанією MySQL AB. Ця СУБД є настільки популярною з цілого ряду вагомих причин, зокрема:

- 1 – MySQL випущена під ліцензією з відкритим вихідним кодом. Таким чином, її використання є безкоштовним.
- 2 – сама по собі MySQL є дуже потужною програмою, яка обробляє великий набір функцій найдорожчих і потужних пакетів баз даних.

3 – MySQL використовує стандартну форму відомого мови даних SQL.

4 – MySQL працює в багатьох операційних системах та з багатьма мовами, включаючи PHP, PERL, C, C ++, JAVA і т. д.

5 – працює дуже швидко і добре навіть з великими об'ємами даних.

6 – добре поєднується з PHP – найціннішою мовою для веб-розробки.

7 – підтримує бази даних великого об'єму, до 50 мільйонів або більше рядків в таблиці. Граничний розмір файлу за замовчуванням для таблиці становить 4 ГБ, але можна його збільшити (якщо операційна система зможе його обробити) до теоретичної межі 8 мільйонів терабайт (ТБ).

8 – MySQL налаштовується. Ліцензія GPL з відкритим вихідним кодом дозволяє модифікувати програмне забезпечення відповідно до своїми специфічних середовищ.

Серверну частину запропоновано реалізувати з використанням мови програмування PHP.

По-перше, що потрібно розуміти, PHP – це просто мова програмування [8]. Тобто, з її допомогою ми можемо виконувати якісь комп'ютерні команди, умови, цикли і так далі (Наприклад, «Якщо то - роби це», «Роби це - поки», т і подібне, по суті якісь умови та цикли). Однак, справа в тому, що ця мова програмування орієнтована на розробку веб-додатків і веб-сайтів. Це так званий серверна мова програмування або backend. Тобто, зворотня частина мови PHP, якщо вона використовується з метою розробки веб-сайтів та якихось скриптів і додатків, працює на віддаленому сервері на якомусь хостингу, який підтримується цим веб-сервером. Ось саме в цьому місці мова php і працює. Це означає, що на комп'ютері клієнтів, які відвідуватимуть сайт ця мова програмування може взагалі бути не встановлена. Тобто їх комп'ютери, смартфони, телефони, ноутбуки та інші гаджети взагалі можуть мову PHP не розуміти і зовсім з нею не бути взагалі «знайомими».

Мова PHP працює на віддаленому сервері, тому він і називається серверний мова програмування.

Це працює за такою схемою: якийсь відвідувач вирішив зайти на запропонований сайт через мережу Інтернет з нутбука. Він звертається до нашого веб-сервера, який ,відповідно, перенаправляє відповідний запит до відповідного PHP скрипта. Далі вже скрипт PHP взаємодіє з різними службами. Тобто і наша мова PHP, і він знаходяться саме в цьому якомусь файлі, на який від клієнта отримано запит через веб-сервер. PHP виконує якийсь певний набір дій, готує якусь веб-сторінку, віддає її веб-серверу, який, в свою чергу, «віддає» назад тому клієнту який запросив цю саму сторінку.

Отже, мова PHP є, як би певною єднальною ланкою, таким собі ядром, центральним процесором, який виконує всі операції автоматизації на віддаленому web-сервері. Перевіряє якісь умови і загалом виконує все що роблять мови web-програмування.

І основна перевага мови PHP полягає у тому, що вона орієнтована саме на роботу з web-серверами. У ній є дуже багато можливостей, які дозволяють взаємодіяти з серверними базами даних, такими як MySQL і так далі, працювати з поштою, файловою системою на веб-сервері.

Тобто всі ці можливості «вбудовані» в мову PHP. Крім цього, вона містить певні функції, які дозволяють це робити швидко і зручно в порівнянні з іншою мовою програмування встановленою на веб-сервері.

Проте потрібно розуміти, що мову PHP можна використовувати не тільки для web-серверів, для створення web-сайтів і web-додатків. Також її можна легко використати в командному рядку. Тобто запустивши термінал, можна за допомогою команди PHP запустити якийсь PHP-файл, який виконає той скрипт, який в ньому знаходиться після чого видасть якийсь результат в командний рядок.

Тобто терміналі, в командному рядку, в операційній системі яка у вас встановлена мову php теж можна використовувати. Також ми за допомогою

мови php можемо створювати так званий графічний юзер інтерфейс або graphical user interface. Т. е. Це звичайна програма для операційної системи. Тобто можливості мови php це не тільки веб-сайти і веб-додатки. Це також і командний рядок, це також і графічні додатки якісь для операційної системи. Також ми за допомогою мови PHP можна створювати т.з. графічного користувача чи GraphicalUserInterface.

Таким чином можливості мови PHP не обмежуються лишу веб-сайтами і веб-додатками. Це також і командний рядок, і якіст графічні додатки для операційної системи.

Але найчастіше, і в більшості випадків, мова PHP використовується саме для створення web-сайтів і web-додатків. Для прикладу, і напевно на слуху в кожного хто цікавиться тематикою web-розробки - такі сайти як Wordpress і Joomla. Так ось ці самі рушії написані на мові програмування PHP. Крім того Facebook написаний на PHP, так само як і Вікіпедія, яка також написана на PHP.

Таким чином з цього випливає достатня кількість можливостей мови PHP, за допомогою якої можна створювати додатки практично будь-якої складності [9].

Laravel – безкоштовний веб-фреймворк з відкритим вихідним кодом, призначений для розробки з використанням архітектурного шаблону Модель-Уявлення-Контролер (від англ. Model-View-Controller) [10, 11].

Основними ключовими можливостями, що є основою обраної архітектури (Laravel) є:

1. EloquentORM – є реалізацією шаблону проектування Active Record на PHP.

Система ORM Eloquent (об'єктно-реляційне відображення) – зручна та гарна, й підкупає простотою реалізації Active Record для Laravel, робота з БД. Тут характерно, що для кожної з таблиці є відповідний клас-модель, який

застосовується при роботі з даною таблицею. Моделі дозволяють робити запит даних з таблиць, а також вставляти в них нові записи [12].

2. Механізм автоматичної загрузки (автозавантаження) класів PHP без необхідності підключати файли.

3. Blade – двигун шаблонів який забезпечує чистий синтаксис для реалізації виглядів даних.

4. REST-контролери – додатковий шар для розділення логіки обробки GET- і POST-запитів HTTP [13].

Для розробки клієнтської частини було використано платформу Materialize. Materialize – це CSS-фреймворк, який суттєво спрощує розробку, орієнтований на користувача та простий у використанні. Даний фреймворк реалізує матеріал дизайн, який був запропонований Google 2014 року.

2.4 Розробка бази даних

Базу даних можна означити в якості сукупності пов'язаних між собою даних, особливістю яких є організація з врахуванням певних правил, якими передбачено узагальнені принципи описування, збереження і маніпулювання, та яка є незалежною від будь-яких програм прикладного програмного забезпечення.

Поняття «бази даних» класифікують по різному. Одна з класифікацій звучить таким чином «за моделлю представлення даних», і сюди відносять: картотеки, ієрархічні та мережеві, багатовимірні та об'єктно-орієнтовані, за дедуктивним принципом та реляційні.

Далі мова піде про реляційної моделі. Ця модель баз даних не була хронологічно першою, але першою отримала математичний опис, і, будучи економною за частиною базових понять (що зробило її неймовірно привабливою для програмістів і користувачів), в основному використовується в web-додатках.

У реляційних базах даних вся інформація міститься в таблиці. Стовпці в такій таблиці мають конкретний тип і розмір; весь вміст комірок стовпчика має відповідати їх типу. Рядки представляють собою набір неструктурованих даних і утворюють математичне відношення. Таким чином, вся база даних – це безліч таблиць-відносин, звідси і назва моделі (від англ. Relation – відношення).

Один з головних принципів організації даних – побудова взаємозв'язків між усіма елементами, що і відрізняє базу даних від простого набору таблиць. Тобто інформація в таблицях реляційної бази даних повинна бути відповідним чином організована. Ще один найважливіший принцип – нормалізація таблиць, яка зводиться до усунення недоліків структури бази даних, що призводять до різних аномалій і порушень цілісності даних. Недоліками структури можна назвати, наприклад, суперечливість даних, а аномалією – виникнення випадкових помилок в процесі експлуатації БД. Простіше кажучи, нормалізація – розбиття таблиці на дві або більше для виключення повторення (надмірності) інформації.

Розробка баз даних – складний тривалий процес, який варта поділити на три етапи:

1 – збору, аналізу і редагування вимог до даних, так зване концептуальне проектування;

2 – логічного проектування. На цьому етапі вимоги до власне даних перетворюють в безпосередньо структури даних;

3 – фізичного проектування, тобто тут визначаються особливості зберігання даних так само як і методів доступу та інше [14, 15].

Далі у таблицях 2.1 – 2.9 наведено опис таблиць розробленої бази даних.

Таблиця 2.1 – Таблиця user

Ідентифікатор	Тип даних	Обмеження	Значення по замовчуванню	Обов'язковість
id	int(10)	NOT NULL	-	Так
login	varchar(255)	NOT NULL	-	Так
email	varchar(255)	NOT NULL	-	Так
password	varchar(60)	NOT NULL	-	Так
created_at	timestamp	NOT NULL	-	Так
user_info_id	int(10)	NOT NULL	-	Так
group	enum (‘user’, ‘moderator’)	NOT NULL	‘user’	Так

Таблиця 2.2 – Таблиця user_info

Ідентифікатор	Тип даних	Обмеження	Значення по замовчуванню	Обов'язковість
id	int(10)	NOT NULL	-	Так
user_id	int(10)	NOT NULL	-	Так
name	varchar(255)	-	NULL	Ні
email	varchar(255)	-	NULL	Ні
phone	varchar(15)	-	NULL	Ні
skype	varchar(60)	-	NULL	Ні

Таблиця 2.3 – Таблиця ads

Ідентифікатор	Тип даних	Обмеження	Значення по замовчуванню	Обов'язковість
id	int(10)	NOT NULL	-	Так
author_id	int(10)	NOT NULL	-	Так
realty_id	int(10)	NOT NULL	-	Так
verefiction_id	int(10)	NOT NULL	-	

Продовження таблиці 2.3

type	enum (‘rent’, ‘sale’)	NOT NULL	-	Так
title	varchar(60)	NOT NULL	-	Так
description	varchar(500)	NOT NULL	-	Так
price	decimal	-	NULL	Ні
created_at	timeshtamp	NOT NULL	-	Так
updaterd_at	timeshtamp	NOT NULL	-	Так

Таблиця 2.4 – Таблиця realty

Ідентифікатор	Тип даних	Обмеження	Значення по замовчуванню	Обов’язковість
id	int(10)	NOT NULL	-	Так
ads_id	int(10)	NOT NULL	-	Так
type	enum(‘apartament’, ‘room’, ‘house’)	NOT NULL	-	Так
area	int(10)	-	NULL	Ні
room_count	int(10)	-	NULL	Ні
city_id	int(10)	NOT NULL	-	Так
address	varchar(10)	-	NULL	Ні
photo	varchar(20)	-	NULL	Ні
created_at	timeshtamp	NOT NULL	-	Так
updaterd_at	timeshtamp	NOT NULL	-	Так

Таблиця 2.5 – Таблиця city

Ідентифікатор	Тип даних	Обмеження	Значення по замовчуванню	Обов’язковість
id	int(10)	NOT NULL	-	Так
name	varchar(25)	NOT NULL	-	Так

Таблиця 2.6 – Таблиц verification

Ідентифікатор	Тип даних	Обмеження	Значення по замовчуванню	Обов'язковість
id	int(10)	NOT NULL	-	Так
ads_id	int(10)	NOT NULL	-	Так
status	enum('waiting', 'checked', block)	NOT NULL	'waiting'	Так
reviwer_id	int(10)	-	NULL	Ні

Таблиця 2.7 – Таблиц favorite

Ідентифікатор	Тип даних	Обмеження	Значення по замовчуванню	Обов'язковість
id	int(10)	NOT NULL	-	Так
ads_id	int(10)	NOT NULL	-	Так
user_id	int(10)	NOT NULL	-	Так

Таблиця 2.8 – Таблиц report

Ідентифікатор	Тип даних	Обмеження	Значення по замовчуванню	Обов'язковість
id	int(10)	NOT NULL	-	Так
ads_id	int(10)	NOT NULL	-	Так
reporter_id	int(10)	NOT NULL	-	Так

Таблиця 2.9 – Таблиця message

Ідентифікатор	Тип даних	Обмеження	Значення по замовчуванню	Обов'язковість
id	int(10)	NOTNULL	-	Так
sender_id	varchar(255)	NOTNULL	-	Так
reciver_id	varchar(255)	NOT NULL	-	Так

Продовження таблиці 2.9

Ідентифікатор	Тип даних	Обмеження	Значення по замовчуванню	Обов'язковість
topic	varchar(60)	NOT NULL	-	Так
text	timestamp	NOT NULL	-	Так
sender_status	enum ('sendet', 'deleted')	NOT NULL	'sendet'	Так
reciver_status	enum('read', 'unread', 'deleted')	NOT NULL	'unread'	Так

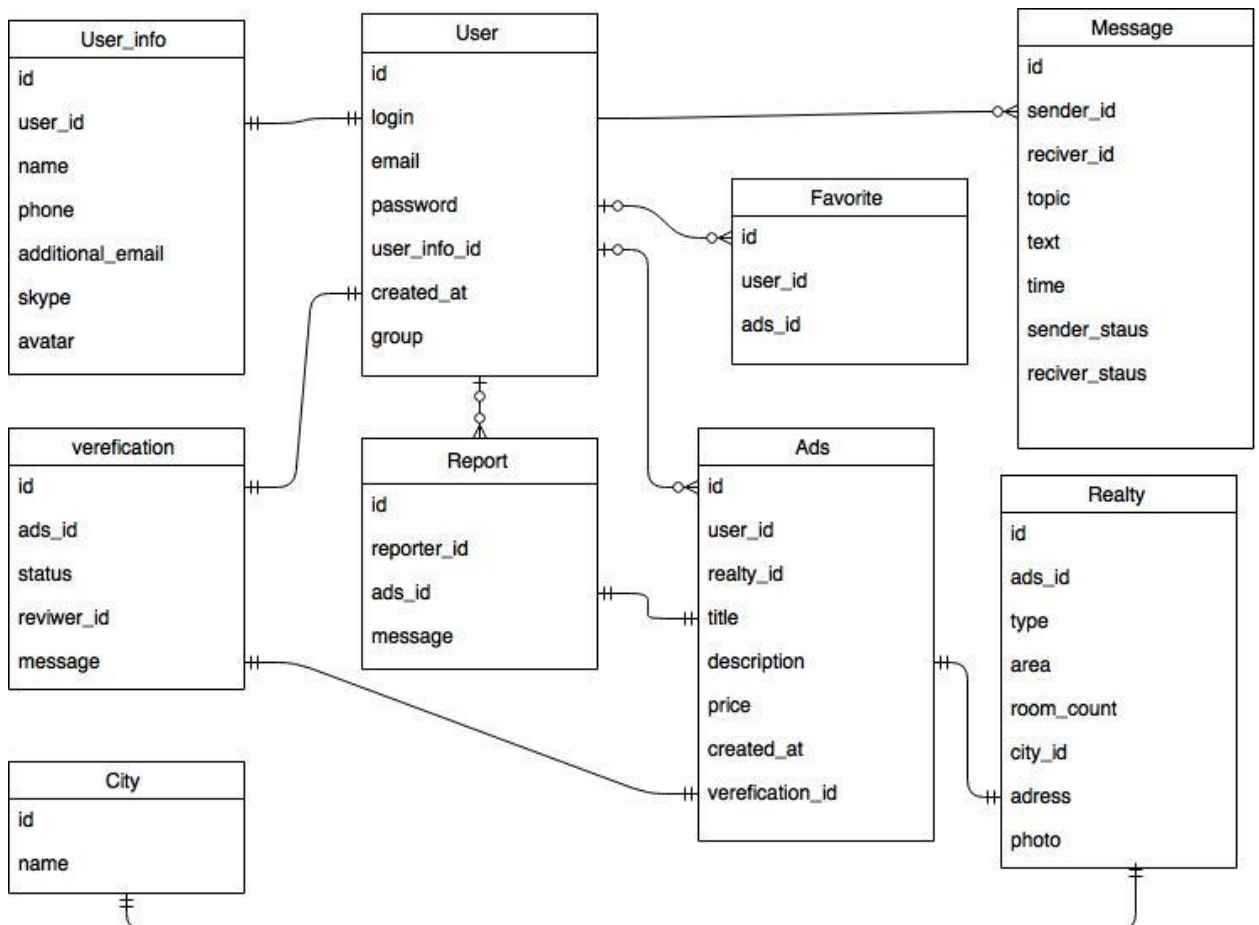


Рисунок 2.3 – Діаграма розробленої бази даних

На рисунку 2.2 представлений кінцевий варіант діаграми розробленої бази даних. Розроблена база даних відповідає всім вимогам і забезпечить зберігання даних необхідних для функціонування веб-сайту.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОЇ СИСТЕМИ

3.1 Реалізація архітектурного шаблону

3.1.1 Опис реалізації моделей даних

Модель даних (Data model) – в шаблоні проектування MVC відповідає за збереження даних та забезпечення доступу до них. Фреймворк Laravel використовує реалізацію ActiveRecord – Eloquent ORM в якому всі класи моделей даних повинні наслідуватись від базового класу Model.

Для роботи з даними необхідно реалізувати модель для кожної з таблиць створених в базі даних. Повний перелік реалізованих класів:

1. User;
2. UserInfo;
3. Ads;
4. Realty;
5. City;
6. Verification;
7. Favorite;
8. Report;
9. Message.

Лістинг 3.1 – Реалізація моделі User

```
<?php namespace App;

use Illuminate\Auth\Authenticatable;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Auth\Passwords\CanResetPassword;
use Illuminate\Contracts\Auth\Authenticatable as AuthenticatableContract;
use Illuminate\Contracts\Auth\CanResetPassword as CanResetPasswordContract;

class User extends Model implements AuthenticatableContract, CanResetPasswordContract
{

    use Authenticatable, CanResetPassword;
    protected $table = 'users';

    protected $fillable = ['login', 'email', 'password'];

    protected $hidden = ['password', 'remember_token'];

    public function is Moderator()
    {
        if ($this->group == "moderator") {
            return true;
        }
        return false;
    }
}
```

В лістингу 3.1 представлено реалізацію моделі User. Змінній \$table присвоюється назва таблиці до якої ми зможемо отримувати доступ через об'єкт

даного класу. Масив значень `$fillable` визначає для яких атрибутів таблиці дозволено масивне присвоєння.

3.1.2 Опис реалізації управління запитами користувача

Контролер(Controller) - управляє запитами користувача. Його основна функція викликати та координувати дії необхідних ресурсів і об'єктів, необхідних для виконання дій, заданих користувачем.

В Laravel фреймворку всі класи контролерів наслідуються від базового класу `Controller`. Для визначення методу контролера для обробки конкретного запиту користувача використовується файл `routes.php`. В лістингу 2.2 відображений вміст файлу маршрутизації.

Лістинг 3.2 – Вміст файлу маршрутизації

```

<?php

Route::get('/', 'MainController@index');

Route::get('/profile/{id}', 'ProfileController@get_profile')
->where('id', '[0-9]+');

Route::get('/profile/ads', array('middleware' => 'auth', 'uses' =>
'AdsController@profile_ads'));

Route::get('/profile', function (){
    return redirect('/profile/ads');});

Route::get('/profile/favorite', array('middleware' => 'auth', 'uses' =>
'FavoriteController@profile_list'));

Route::post('/profile/set-info', 'ProfileController@set_info');
Route::get('/profile/settings', 'ProfileController@profile_settings');
Route::get('/profile/make-user-moderator', 'ProfileController@add_moderator');
Route::post('/change-password', 'ProfileController@change_password');

Route::get('/profile/verification', array('middleware' => 'auth', 'uses' =>
'VerificationController@waiting_for_verification'));

Route::get('/profile/reports', array('middleware' => 'auth', 'uses' =>
'ReportController@reports'));

```

```

Route::get('/show-ads/{id}', 'AdsController@show_ads')
->where('id', '[0-9]+');
Route::get('/new-ads', array ('middleware' =>'auth','uses' => 'AdsController@new_ads'));
Route::post('/add-ads', array ('middleware' =>'auth','uses' => 'AdsController@add'));
Route::post('/edit-ads', array ('middleware' =>'auth','uses' => 'AdsController@edit'));
Route::get('/delete-ads', array ('middleware' =>'auth','uses' => 'AdsController@remove'));
Route::get('/ads', 'AdsController@get_ads');

Route::get('/add-to-favorite',array ('middleware'=>'auth','uses'=> 'FavoriteController@add'));
Route::get('/remove-from-favorite',array('middleware'=>'auth','uses'=>
'FavoriteController@remove'));

Route::post('/send-message',array('middleware'=>'auth','uses'=>
'MessageController@send'));
Route::get('/delete-message',array('middleware'=>'auth','uses'=>
'MessageController@delete'));
Route::get('/messages/new-message',array('middleware'=>'auth','uses'=>
'MessageController@new_message'));
Route::get('/messages/read/',array('middleware'=>'auth','uses'=>
'MessageController@read'));
Route::get('/messages/inbox',array('middleware'=>'auth','uses'=>
'MessageController@inbox'));
Route::get('/messages/outbox',array('middleware'=>'auth','uses'=>
'MessageController@outbox'));

Route::get('/report', array('middleware' => 'auth', 'uses' => 'ReportController@report'));
Route::get('/remove-report', array ('middleware'=>'auth','uses'=>
'ReportController@remove'));

Route::get('/block', array('middleware' =>'auth','uses' => 'VereficationController@block'));
Route::get('/check', array('middleware' =>'auth','uses' => 'VereficationController@check'));

```

```

Route::get('/home', function(){
    return redirect('/');
});

Route::get('/login', function(){
    return redirect('/auth/login');
});

Route::controllers ([
    'auth'=>'Auth\AuthController',
    'password'=>'Auth>PasswordController',
]);

```

Реалізіція контролера Verefication відображена в лістингу 3.3

Лістинг 3.3 – Реалізація контролера VereficationController

```

<?php namespaceApp\Http\Controllers;
useApp \Ads;
useApp \Verefication;
useApp \Report;
useIlluminate\Http\Request;
useIlluminate\Support\Facades\Redirect;

classVereficationController extendsController {

    publicfunction __construct()
    {
    }

    public function block(Request$request)
    {

```

```

        If ($request ->user()->isModerator){
            $verification = Verification::where ("ads_id", "=", $request->input('id'))-
>first();

            $verification->status = "block";
            $verification->save();
            Report::where("ads_id", "=", $request->input('id'))->delete();
            return Redirect::to("/profile/verification");}
        else return Redirect::to("/profile/ads");
    }

    public function check(Request $request)
    {
        $verification = Verification::where("ads_id", "=", $request->input('id'))-
>first();

        $verification->status = "checked";
        $verification->save();
        Report::where("ads_id", "=", $request->input('id'))->delete();
        return Redirect::to("/profile/verification");
    }

    public function waiting_for_verification(Request $request)
    {
        $ads_mass = array();
        $verifications = Verification::where("status", "=", "waiting")->get();
        foreach($verifications as $verification){
            $ads_mass[] = Ads::find($verification->ads_id);}
        return view('profile.waiting_for_approve', ["ads_mass"=>$ads_mass]);
    }
}

```

Функція `waiting_for_verification` в якості аргумента отримує об'єкт класу `Request` в якому містить повну інформація про запит користувача. Як результат функція повертає згенеровану HTML-сторінку. Для генерації HTML використовується вигляд даних, якому передається масив об'єктів моделі `Ads`.

3.1.3 Опис реалізації генерації HTML-сторінок на основі переданих параметрів

Вигляд даних(View) – використовується для генерації HTML-сторінок на основі переданих параметрів. В фреймворці Laravel для реалізації виглядів даних використовується синтаксис Blade.

Лістинг 3.4 – Реалізація вигляду даних `inbox`

```

@extends('message_base')
@section('title')
    <title> Вхідні повідомлення</title>
@endsection
@section('login')
    {{ Auth::user()->name }}
@endsection
@section('message_content')
    <div class="col s9">
        <ul class="collection with-header z-depth-1">
            <li class="collection-header"><h5>Непрочитані повідомлення:</h5></li>
            @forelse($unreadet as $message)
                <li class="collection-item">
                    <div>
                        <a href="{{ url('/messages/read?id=' . $message['id']) }}" ><b>{{
$message['login'] }}</b>&nbsp;{{ $message['topic'] }}</a>
                        <a href="{{ url('/delete-message?id=' . $message['id']) }}" class="right"><i
class="mdi-action-delete"></i></a>
                    </div>
                </li>
            @empty
                <li class="collection-item">
                    <div>
                        Ви не отримали жодного повідомлення.
                    </div>
                </li>
            @endforelse
        </ul>
    <div class="row container">&nbsp;</div>

```



```

<ul class="collection with-header z-depth-1">
  <li class="collection-header"><h5>Прочитані повідомлення:</h5></li>
    @forelse($readet as $message)
      <li class="collection-item">
        <div>
          <a href="{{ url('/messages/read?id=' . $message['id']) }}" ><b>{{
$message['login'] }}</b>&nbsp;{{ $message['topic'] }}</a>
          <a href="{{ url('/delete-message?id=' . $message['id']) }}" class="right"><i
class="mdi-action-delete"></i></a>
        </div>
      </li>
    @empty
      <li class="collection-item">
        <div>
          Ви не прочитали жодного повідомлення.
        </div>
      </li>
    @endforelse
  </ul>
</div>

<div class="fixed-action-btn" style="bottom: 24px; right: 24px;">
  <a class="btn-floating btn-large red" href="{{ url('messages/new-message') }}">
    <i class="large mdi-content-mail"></i>
  </a>
</div>
@endsection

```

Реалізація вигляду даних inbox – генерує HTML – сторінку з вхідними повідомленнями. Даний вигляд даних розширює базовий вигляд даних message_base.

3.2 Використання програмної системи

3.2.1 Розгортання програмної системи та системні вимоги

Для використання веб-сайту користувачу користувачу необхідний комп'ютер із будь-якою операційною системою. Обов'язковою вимогою для використання є наявність сучасного веб-браузера.

Для серверної частини програмної системи потрібен виділений веб-сервер з операційною системою Linux. Також необхідно встановити певний перелік програмного забезпечення:

1. веб-сервер Nginx або Apache;
2. PHP версії не нижче 5.4;
3. MySQL;
4. Composer.

Мінімальні вимоги до комп'ютера, який буде виконувати роль веб-сервера:

1. Багатоядерний процесор з тактовою частотою ядра 1.6 GHz і більше. Бажано процесор з вищою тактовою частотою та двома і більше ядрами для зручності використання;
2. Мінімальний обсяг оперативної пам'яті – 1024 мегабайт;
3. Мінімальний обсяг вільного дискового простору – 1 гігібайт.

Для початку роботи встановити Laravel за допомогою Composer виконавши відповідну команду в консолі (`composer global require "laravel/installer=~1.1"`). Після цього перемістити папку з реалізованим проектом в папку із встановленим фреймворком Laravel. Далі необхідно виконати

базову конфігурацію веб-сайту. Всі конфігураційні файли знаходяться в директорії “config”.

При використанні веб-серверу Apache необхідно переконатись що модуль “mod_rewrite” є увімкненим.

3.2.2 Опис типових схем використання розробки

В даній частині будуть описані типові схеми використання розробленого веб-сайту.

Головна сторінка містить перелік посилань на оголошення про нерухомість в кожному з доступних міст. При кліку на будь-яке з них користувач перейде на сторінку із списком усіх оголошень з вибраним фільтром по місту. Приклад

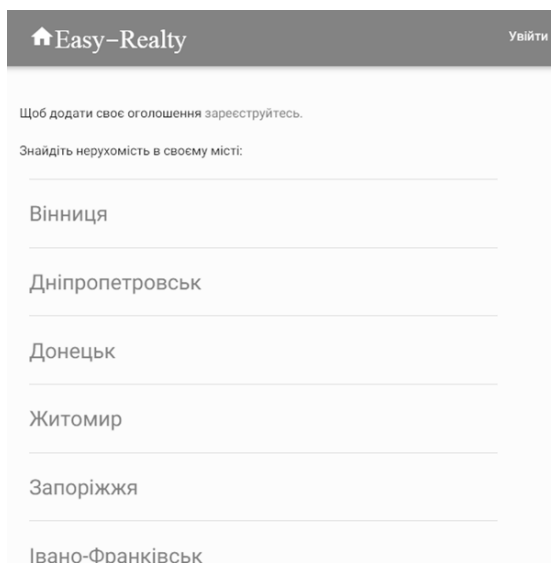


Рисунок 3.1 – Головна сторінка веб-сайту

На сторінці з оголошеннями відображаються форма для вибору фільтру за містом, типом оголошення (оренда, продаж) та типом нерухомості(квартира, кімната або будинок). При кліку на кнопку

“Фільтрувати” користувачу повернеться згенерована сторінка із оголошеннями, як відповідають обраним фільтрам.

Для входу на веб-сайт користувачу необхідно перейти на сторінку входу. В поля необхідно ввести значення емейлу та перолю користувача. І натиснути кнопку “Увійти”. При правильній комбінації емейлу та паролю користувач авторизується і перейде на стрінку свого профілю. Якщо вибрана опція “Запам’ятати мене” тоді користувач буде авторизованим при повторному заході на веб-сайт.

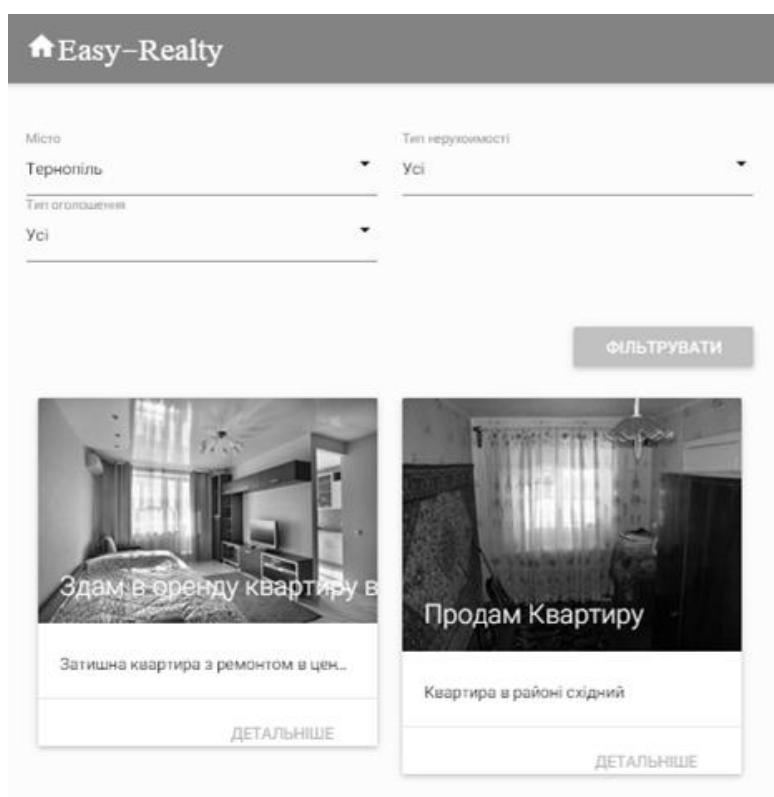
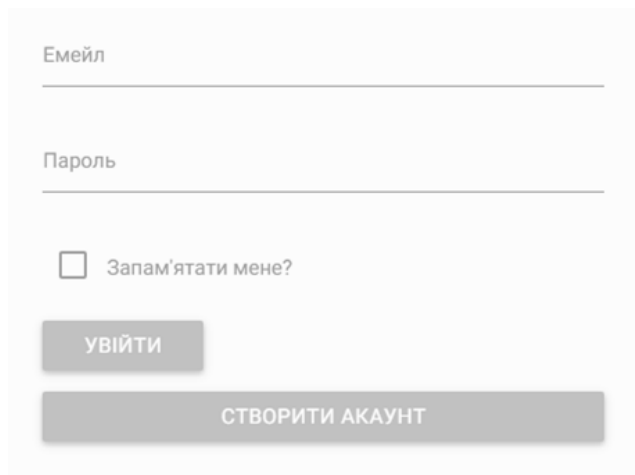


Рисунок 3.2 – Сторінка із списком оголошень



Емейл

Пароль

Запам'ятати мене?

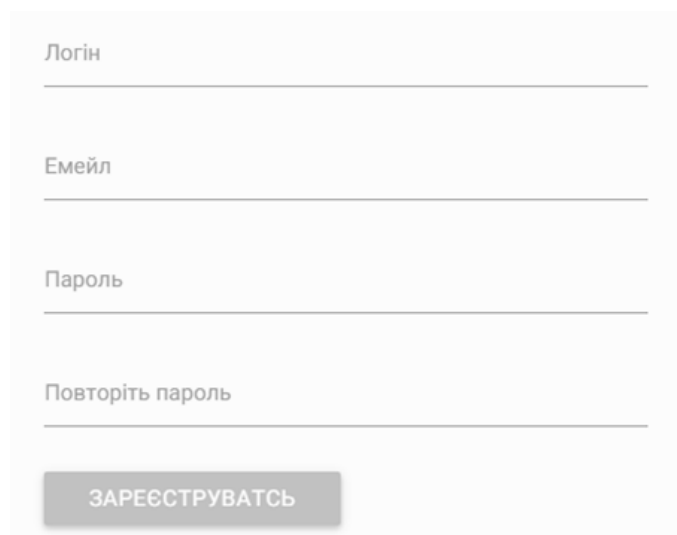
УВІЙТИ

СТВОРИТИ АКАУНТ

Рисунок 3.3 – Сторінка входу користувача

У разі невідповідності емелу чи паролю існуючому користувачу користувач побачить повідомлення.

Якщо користувач не є зареєстрованим він може перейти на сторінку реєстрації. Йому буде необхідно ввести унікальний логін та емейл, які не використовуються вже існуючими користувачами. Також необхідно ввести пароль та його підтвердження. Всі поля для вводу є обов'язковими. При вводі невідповідних даних, або не повного введення даних користувач буде бачити повідомлення.



Логін

Емейл

Пароль

Повторіть пароль

ЗАРЕЄСТРУВАТЬСЯ

Рисунок 3.4 – Сторінка реєстрації нового користувача

Після успішної реєстрації користувач переходить на сторінку свого профілю де йому доступні: оголошення, закладки,повідомлення, та контакти. У вкладці відображається список оголошень, які додав користувач. Також можна перейти на сторінку створення нового оголошення.

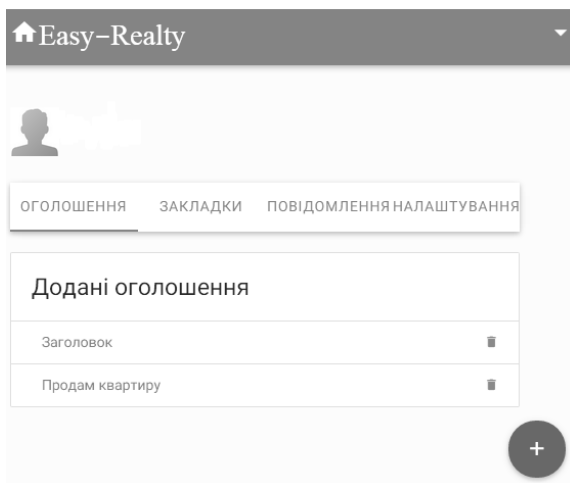


Рисунок 3.5 – Сторінка із списком оголошень доданих користувачем

У вкладці “Закладки” користувач може переглянути список оголошень, які він додав до закладок. При кліку на назву оголошення користувача буде перенаправлено на сторінку оголошення. Є можливість видалити закладку.

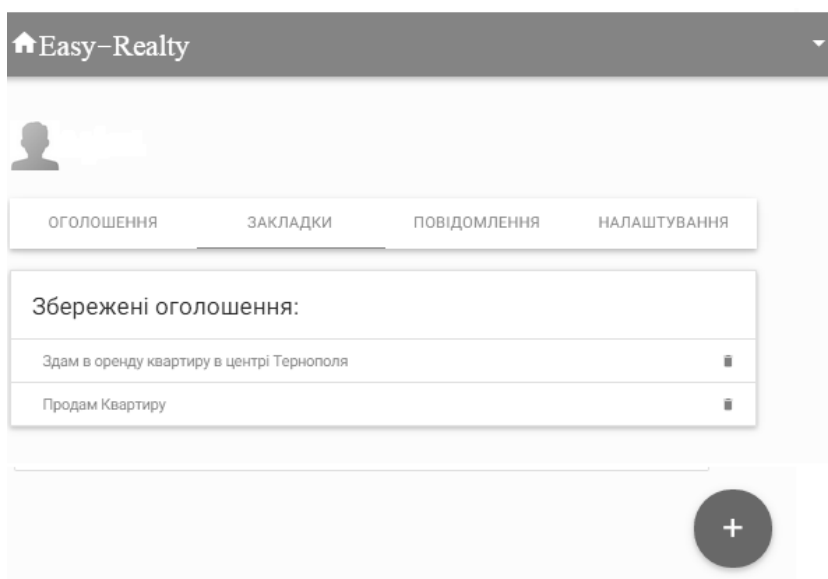


Рисунок 3.6 – Сторінка із списком оголошень доданих до закладок

Вкладка “Повідомлення” містить перелік вхідних та надісланих повідомлень. Повідомлення можуть бути видалені і більше не будуть відображатись у переліку повідомлень. При кліку на кнопку із значком конверта користувач перейде на сторінку для написання повідомлення.

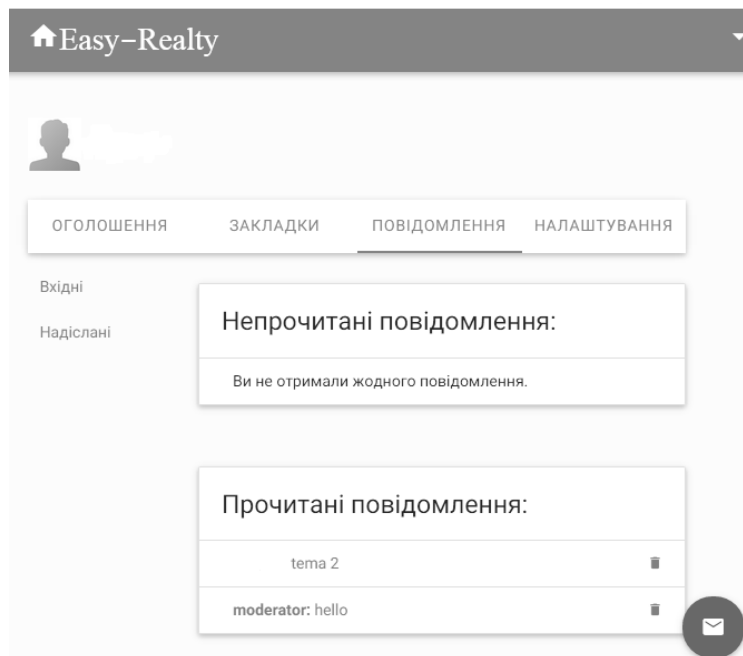


Рисунок 3.7 – Вкладка із списком повідомлень користувача

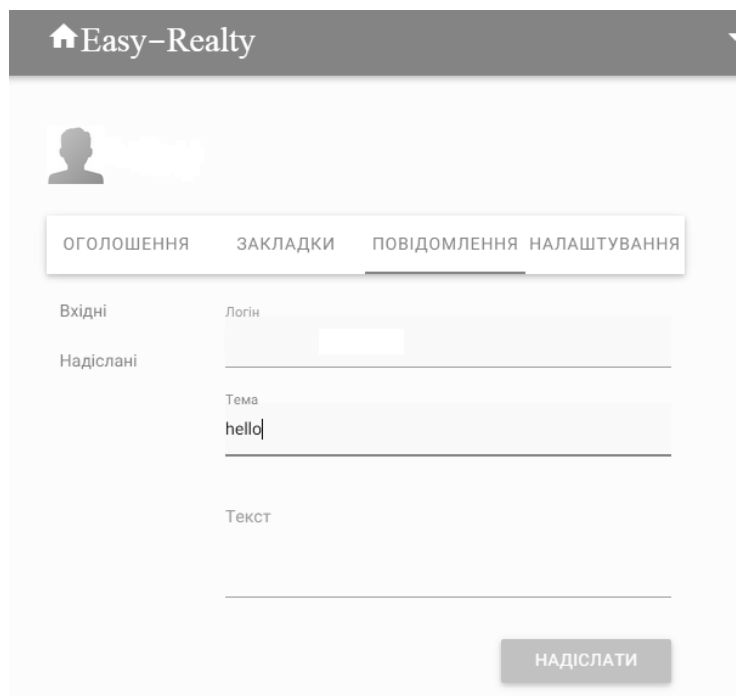


Рисунок 3.8 – Форма для надсилання приватного повідомлення

Під вкладкою “Налаштування” знаходиться форма (рисунок 3.11) для додавання та редагування контактної інформації.



Рисунок 3.9 – Сторінка з приватним оповідженням

Для модератора сторінка профілю дещо відрізняється. Замість вкладок “Оголошення” та “Закладки” йому доступні вкладки “Верифікація” і “Скарги”. У вкладці “Верифікація” відображаються оголошення, додані користувачами і очікують на перевірку модератором. Вміст вкладки “Скарги” складається із списку оголошень на які подали скарги.



Рисунок 3.10 – Список оголошень, які потребують перевірки

Оголошення буде відображатись для всіх тільки після перевірки оголошення модератором.

Заголовок

Оренда Продаж

Тип нерухомості
Виберіть свій варіант

Опис

Місто
Виберіть місто

Адреса

К-сть кімнат

Площа

Ціна

ФОТО

ДОДАТИ ОГолошення

Рисунок 3.11 – Стрінка для додавання нового оголошення

Форма для додавання нового оголошення містить поля для введення всіх необхідних даних (заголовок, опис, тип оголошення, тип нерухомості, місто, адреса, к-сть кімнат та площа) при коректному заповненні всіх полів і надсиланні форми буде створено нове оголошення і користувач буде перенаправлений на сторінку профілю.

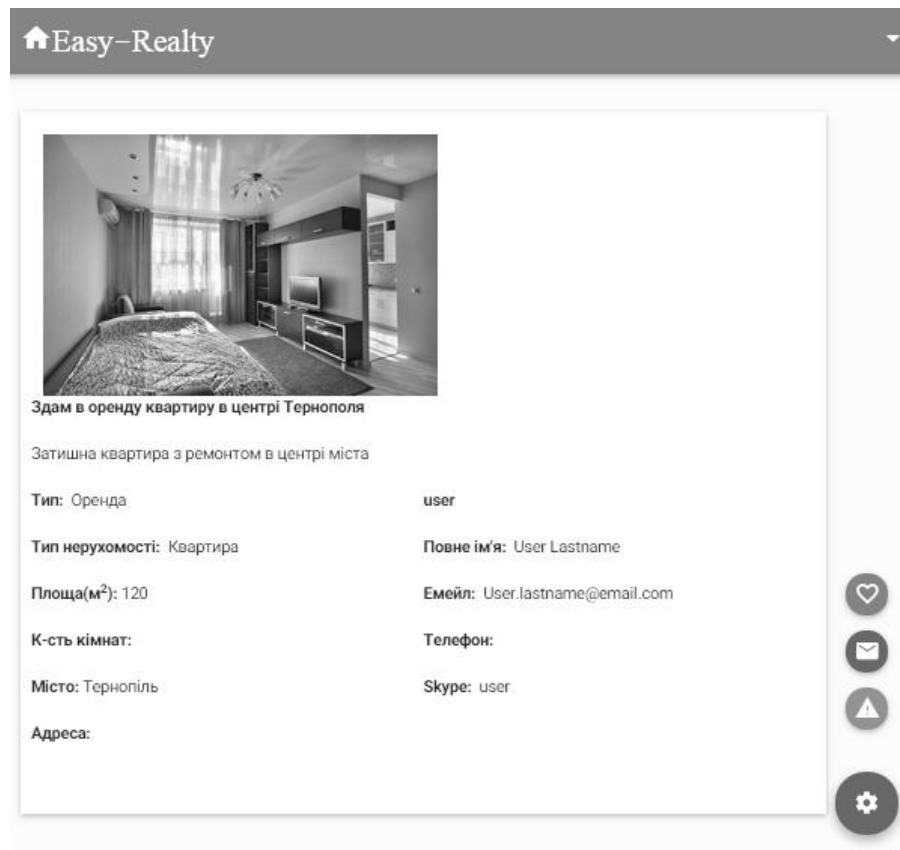



Рисунок 3.12 – Вигляд сторінки з оголошенням для зареєстрованого користувача

Сторінка з оголошенням має декілька варіантів відображення. Не зареєстрований користувач буде бачити оголошення, але не буде мати можливості виконувати будь-які дії з цим оголошенням. Автор оголошення зможе видалити оголошення або перейти на сторінку редагування оголошення.

Модератор може заблокувати повідомлення або поставити статус перевіреного.

Easy-Realty moderator



Здам в оренду квартиру в центрі Тернополя


Затишна квартира з ремонтом в центрі міста

Тип: Оренда	user
Тип нерухомості: Квартира	Повне ім'я: User Lastname
Площа(м ²): 120	Емейл: User.lastname@email.com
К-сть кімнат:	Телефон:
Місто: Тернопіль	Skype: user
Адреса:	

🔒
✓
⚙️

Рисунок 3.13 – Вигляд сторінки з оголошенням для модератора

Easy-Realty user



Здам в оренду квартиру в центрі Тернополя

Затишна квартира з ремонтом в центрі міста

Тип: Оренда	user
Тип нерухомості: Квартира	Повне ім'я: User Lastname
Площа(м ²): 120	Емейл: User.lastname@email.com
К-сть кімнат:	Телефон:
Місто: Тернопіль	Skype: user
Адреса:	

🗑️
✎
⚙️

Рисунок 3.14 – Вигляд сторінки з оголошенням для власника оголошення

На сторінці з оголошенням відображається повна інформація про нерухомість та контакти користувача який додав оголошення.

3.3 Тестування програмної системи

3.3.1 Опис процесу тестування

Ціль даного плану тестування являється опис процесу тестування веб-сайту для роботи з нерухомістю.

Тестування – процес направлений на виявлення дефектів та помилок в програмному продукті шляхом пошуку невідповідності між очікуваним результатом і отриманим [16, 17].

Планується три етапи тестування. Перший етап заключається в аналізі ТЗ і складання тест плану. Другий етап буде присвячений функціональному тестуванню. Другий етап буде включати в себе:

1. перевірка роботи всіх обов'язкових функцій сайту;
2. тестування працездатності користувацьких форм на сайті;
3. перевірка посилань.

На третьому етапі відбудеться тестування верстки. Даний етап необхідний для перевірки розташування елементів, у відповідності до їх визначених позицій. Даний етап тестування полягає у порівнянні відображення в різних веб-браузерах. Перелік веб-браузерів, які будуть використанні для тестування верстки:

1. Google Chrome;
2. Mozilla Firefox;
3. Safari;
4. Internet Explorer 11;
5. Opera.

3.3.2 Розробка тестів

Для даного проекту в основному будемо проводити функціональне тестування, методом чорної скриньки [18] (рисунок 3.15) - це стратегія або метод тестування, який базується лише на тестуванні по функціональній специфікації і вимогам, не зважаючи на внутрішню структуру коду і без доступу до бази даних. Фактично ми я розумію який повинен бути результат при певному наборі даних, які подаються на вхід. Результат перевіряємо з юзер інтерфейсу на рівні простого користувача. На сьогодні, саме ця стратегія є однією з найпопулярніших в ІТ компаніях.

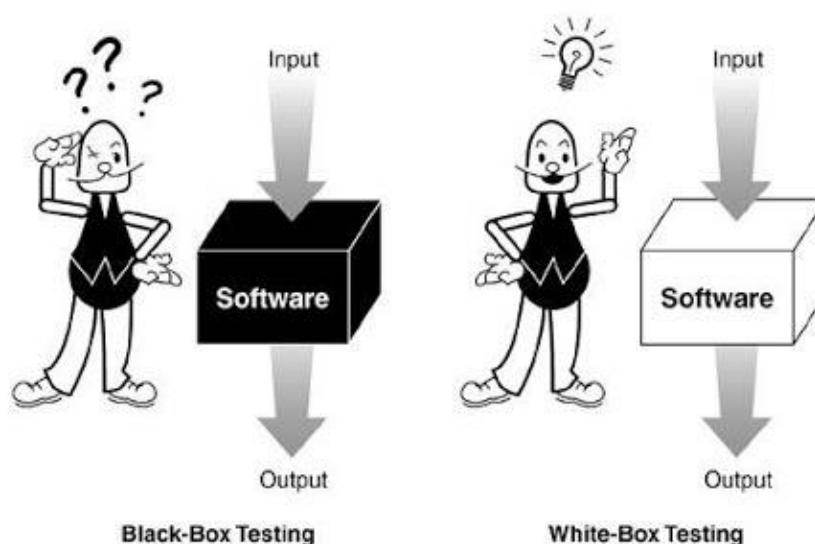


Рисунок 3.15 – Black box testing

Набір тесткейсів для перевірки реєстрації приведені в таблиці 3.1.

Таблиця 3.1 – Тест кейси для тестування реєстрації

Послідовність кроків	Очікуваний результат	Кінцевий результат
1) Відкрити сторінку для реєстрації користувача;	Реєстрація пройшла успішно і в браузері відкривається сторінка	Успішно

2) Заповнити поля для введення коректними даними;	профілю створеного користувача	
3) Підтвердити форму реєстрації.		

1) Відкрити сторінку для реєстрації користувача; 2) В полі логіна ввести значення, яке відповідає значенню логіна вже існуючого користувача; 3) Підтвердити форму реєстрації.	Користувач побачить повідомлення про те що користувач з таким логіном вже існує	Успішно
1) Відкрити сторінку для реєстрації користувача; 2) В полях для введення пвролю та підтвердження паролю ввести різні значення; 3) Підтвердити форму реєстрації.	Користувач побачить повідомлення про те що паролі не співпадають	Успішно
1) Відкрити сторінку для реєстрації користувача; 2) Підтвердити надсилання форми незаповненими полями	Користувач побачить повідомлення про те що всі поля є обов'язковими для заповнення	Успішно

Даний набір тесткейсів перевіряє коректність реєстрації нових користувачів.

Він включає в себе основні випадки, серед яких:

1. всі введені дані є коректними і валідними;
2. введено логін вже існуючого користувача;
3. введено різні паролі;
4. надсилання форми з незаповненими полями.

Як видно з результатів проведеного тестування, реєстрація користувачів працює коректно.

Набір тесткейсів для перевірки надсилання повідомлення приведені в таблиці 3.2.

Таблиця 3.2 – Тест кейси для тестування реєстрації

Послідовність кроків	Очікуваний результат	Кінцевий результат
1) Відкрити сторінку з формою для надсилання нового повідомлення 2) Заповнити поля для введення коректними даними; 3) Надіслати повідомлення.	Користувач буде перенаправлений на сторінку із списком надісланих повідомлень і надіслане повідомлення також буде відображатись	Успішно
1) Відкрити сторінку з формою для надсилання нового повідомлення; 2) Вполе з логіном ввести значення неіснуючого користувача; 3) Надіслати повідомлення.	Користувач побачить повідомлення про те що користувач з таким логіном не існує	Успішно
1) Відкрити сторінку з формою для надсилання нового повідомлення; 2) Надіслати повідомлення з порожніми полями для даних.	Користувач побачить повідомлення про те що всі поля є обов'язковими	Успішно

Даний набір тесткейсів перевіряє коректність надсилання нового повідомлення. Він включає в себе основні випадки:

1. всі введені дані є коректними і валідними;
2. введено логін вже неіснуючого користувача;
3. надсилання повідомлення з незаповненими полями.

Як видно з результатів проведеного тестування, надсилання повідомлень працює коректно.

Набір тесткейсів для перевірки роботи закладок таблиці 3.3.

Таблиця 3.3 – Тест кейси для тестування реєстрації

Послідовність кроків	Очікуваний результат	Кінцевий результат
1) Відкрити сторінку з оголошенням 2) Додати оголошення до закладок; 3) Перейти на сторінку профілю і відкрити вкладу із закладками.	Оголошення додане до закладок відображається у списку	Успішно
1) Перейти на сторінку профілю і відкрити вкладу із закладками; 2) Клікнути на оголошення в списку закладок.	Сторінка з оголошенням відкриється	Успішно
1) Перейти на сторінку профілю і відкрити вкладу із закладками; 2) Видалити закладку на оголошення.	Видалене оголошення більше не відображається у списку закладок	Успішно

Даний набір тесткейсів перевіряє коректність роботи із закладками. Він включає в себе основні випадки:

1. додання оголошення до закладок;
2. відкриття оголошення доданого до закладок;
3. видалення закладки.

Набір тесткейсів для перевірки модерування оголошень таблиці 3.3.

Таблиця 3.4 – Тест кейси для тестування реєстрації

Послідовність кроків	Очікуваний результат	Кінцевий результат
1) Додати оголошення; 2) Увійти в систему як модератор; 3) Перейти на сторінку профілю .	Додане оголошення відображається в списку оголошень, які очікують на перевірку	Успішно

1) Увійти в систему як модератор; 2) Перейти на сторінку профілю; 3) Верифікувати оголошення;	Верифіковане оголошення більше не відображатиметься у списку оголошень, які очікують на перевірку	Успішно
---	---	---------

Даний набір тесткейсів перевіряє коректність роботи модерування. Він включає в себе основні випадки:

1. перегляд оголошень, які очікують на перевірку;
2. верифікація оголошення.

Для перевірки правильності відображення веб-сайту визначимо перелік сторінок, які будуть використані даного етапу тестування:

1. головна сторінка;
2. сторінка входу;
3. сторінка реєстрації;
4. сторінка профілю;
5. сторінка з оголошеннями;
6. сторінка для додавання нового оголошення;
7. сторінка з повідомленнями.

Результати виконання тестування верстки відображені в таблиці 2.5

Таблиця 3.5 – Результати тестування верстки

	1	2	3	4	5	6	7
Google Chrome	+	+	+	+	+	+	+
Mozilla Firefox	+	+	+	+	+	+	+
Safari	+	+	+	+	+	+	+
Internet Explorer	+	+	+	+	+	+	+

Opera	+	+	+	+	+	+	+
-------	---	---	---	---	---	---	---

За результатами проведеного тестування розроблений веб-сайт перевірено базовий функціонал, користувацькі форми та відображення в різних браузерях.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Темою кваліфікаційної роботи є розробка сучасного on-line сервісу із застосуванням технологій веб-програмування. Дотримання основних вимог з охорони праці [19] при роботі з ПК дозволяє безпечно використовувати програмні продукти. Тому необхідно розглянути санітарні норми та правила що стосуються приміщень та робочих місць де використовується ПК.

При розробці системи потрібно запобігти негативному впливу виробничих факторів, а саме дотримуватися правил та норм щодо приміщень де знаходиться робоче місце, відповідно до НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин» та ДСанПіН 3.3.2.007-98 «Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».

Відповідно до ДСанПіН 3.3.2.007-98 робоче місце працівника не можна розміщувати у підвалах та цокольних поверхах. Площа робочого місця для виконання розробки повинна становити не менше 6 м², а об'єм 20 м³. Приміщення, в якому працюють на ПК повинно бути забезпечене природнім та штучним освітлення згідно з нормами ДБН В.2.5-28-2006. Природне освітлення здійснюється крізь світлові отвори, направлені на північний схід і вони забезпечують коефіцієнт природньої освітленості (КПО) не менший ніж 1,5%. Розраховується КПО за методикою, що міститься у ДБН В.2.5-28-2006. В

приміщеннях де проводиться розробка та міститься ПК, штучне освітлення здійснюється системою загального рівномірного освітлення. Джерелом штучного освітлення слугують люмінесцентні лампи. Віконні отвори приміщення обладнані вертикальними жалюзями та зовнішніми козирками. У даному приміщенні функціонує система опалення. Відповідно до санітарних норм та правил у даному приміщенні знаходиться аптечка першої медичної допомоги. Також, у даному приміщенні щоденно проводиться вологе прибирання.

Згідно правил НПАОП 0.00-1.28-10 обладнання та організація робочого місця працюючого з ПК має відповідати ергономічним вимогам ГОСТ 12.2.032-78, ГОСТ 22.269-76, ГОСТ 21.889-76, а саме:

- оптимальна робоча поза користувача ПК забезпечується конструкцією робочого місця;
- робоче місце розташоване так відносно світлових отворів, що природне світло падає збоку зліва;
- відповідно вимогам, робочий стілець є підйомно-поворотним з регульованою висотою;
- будова робочого стола відповідає вимогам ергономіки та дозволяє оптимально розміщувати на робочій поверхні ПК, допоміжне обладнання та документи.

Нормовані параметри мікроклімату, іонного складу повітря, вмісту шкідливих речовин мають відповідати вимогам СН 4088-86, СН 2152-80, ГОСТ 12.1.005-88, ГОСТ 12.1.007-76;

Мікрокліматичні умови приміщення повинні відповідати нормальним значенням таких показників:

- температура повітря;
- відносна вологість повітря;
- швидкість руху повітря;
- інтенсивність теплового (інфрачервоного) опромінення.

За ступенем впливу на тепловий стан людини мікрокліматичної умови поділяють на оптимальні та допустимі.

Для робочої зони приміщення встановлюються оптимальні та допустимі мікрокліматичні умови з урахуванням складності виконуваної роботи та періоду року. При виконанні роботи на ПК, що пов'язано з нервово-емоційним напруженням, у приміщенні потрібно підтримувати температура повітря $+22^{\circ}\text{C}$ – $+24^{\circ}\text{C}$, відносну вологість 60-40%. Дані вимоги до параметрів мікроклімату містяться у санітарних нормах ДСН 3.3.6.042-99.

Оскільки розробка системи комплексної обробки подій здійснюється за ПК, необхідно проводити перерви по 15 хвилин через кожну годину. Якщо, виробничі обставини не дозволяють здійснювати часті перерви, тривалість роботи з ПК не повинна бути більшою чотирьох годин.

Відповідно до основних правил електробезпеки під час використання ПК потрібно дотримуватися таких вимог:

- при використанні ліній електромережі необхідно запобігти виникненню електричного джерела займання внаслідок короткого замикання чи перевантаження мережі;
- до електромережі ПК підключається тільки з а допомогою штепсельних з'єднань і електророзеток;
- спеціальні контакти для підключення нульового захисного провідника повинні міститися у штепсельних з'єднаннях та електророзетках. При цьому, під'єднання нульового захисного провідника відбувається швидше, ніж під'єднання фазового та нульового робочого провідника;
- підключення живлення ПК до двопровідної електромережі є недопустимим, навіть з використанням перехідних пристроїв. Тому здійснюється підключення по трьохпровідній мережі.

Згідно основних вимог до пожежної безпеки приміщення, у яких знаходиться ПК, мають бути оснащені переносними вуглекислотними або

аерозольно-водопінними вогнегасниками. Підходи до засобів пожежогасіння повинні бути вільними. Також, згідно вимог НАПБ Б.06.004-2005 «Перелік однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежогасіння та пожежної сигналізації», у приміщенні де здійснюється робота з ПК, робочі місця повинні бути обладнанні системою автоматичної пожежної сигналізації з димовим пожежним сповіщувачем.

Таким чином, в даному підрозділі виконано огляд основних законодавчих актів та нормативів при роботі з ПК. Так як, дипломна робота магістра спрямована на розробку системи комплексної обробки подій та її реалізація для сфери алгоритмічної торгівлі, то було наведено основні стандарти і правила щодо влаштування робочих місць де використовують комп'ютерну техніку. Також було наведено вимоги, які стосуються приміщення де знаходиться робоче місце працівника. Крім того, розглянуті санітарні норми та правила щодо мікроклімату у даному приміщенні. Наведенні правила електробезпеки під час роботи з ПК. Також поданні основні вимоги до пожежної безпеки приміщення. При дотриманні даних правил гарантуються безпечні умови праці на ПК та запобігання шкідливих виробничих факторів.

4.2 Безпека в надзвичайних ситуаціях

4.2.1 Запобігання виникненню надзвичайних ситуацій

Запобігання виникненню надзвичайних ситуацій [20] – це підготовка та реалізація комплексу правових, соціально-економічних, політичних, організаційно-технічних, санітарно-гігієнічних та інших заходів, спрямованих на регулювання безпеки, проведення оцінки рівнів ризику, завчасне реагування на загрозу виникнення надзвичайної ситуації на основі даних моніторингу (спостережень), експертизи, досліджень та прогнозів щодо можливого перебігу

подій з метою недопущення їх переростання у надзвичайну ситуацію або пом'якшення її можливих наслідків [20].

Зазначені функції запобігання надзвичайним ситуаціям техногенного та природного характеру в нашій країні виконує Єдина державна система запобігання надзвичайним ситуаціям техногенного і природного характеру і реагування на них, затверджена Постановою Кабінету Міністрів України від 3 серпня 1998 р. № 1198.

Єдина державна система запобігання надзвичайним ситуаціям техногенного і природного характеру і реагування на них (ЄДСЗР) включає в себе центральні та місцеві органи виконавчої влади, виконавчі органи рад, державні підприємства, установи та організації з відповідними силами і засобами, які здійснюють нагляд за забезпеченням техногенної та природної безпеки, організують проведення роботи із запобігання надзвичайним ситуаціям техногенного та природного походження і реагування у разі їх виникнення з метою захисту населення і довкілля, зменшення матеріальних втрат.

Основною метою створення ЄДСЗР є забезпечення реалізації державної політики у сфері запобігання і реагування на надзвичайні ситуації, забезпечення цивільного захисту населення.

ЄДСЗР складається з постійно діючих функціональних і територіальних підсистем і має чотири рівні: загальнодержавний, регіональний, місцевий та об'єктовий.

Функціональні підсистеми створюються міністерствами та іншими центральними органами виконавчої влади для організації роботи, пов'язаної із запобіганням надзвичайним ситуаціям та захистом населення і територій від їх наслідків.

Кожний рівень ЄДСЗР має координуючі та постійні органи управління щодо розв'язання завдань у сфері запобігання надзвичайним ситуаціям, захисту населення і територій від їх наслідків, систему повсякденного управління, сили і

засоби, резерви матеріальних та фінансових ресурсів, системи зв'язку та інформаційного забезпечення.

Координуючі органи ЄДСЗР:

- державна комісія з питань техногенно-екологічної безпеки та надзвичайних ситуацій;
- національна рада з питань безпечної життєдіяльності населення;
- комісії Ради міністрів Автономної Республіки Крим, обласних, Київської та Севастопольської міських державних адміністрацій з питань техногенно-екологічної безпеки та надзвичайних ситуацій;
- комісії районних державних адміністрацій і виконавчих органів рад з питань техногенно-екологічної безпеки та надзвичайних ситуацій;
- комісії з питань надзвичайних ситуацій об'єкта.

До складу ЄДСЗР входять відповідні сили і засоби функціональних і територіальних підсистем, а також недержавні (добровільні) рятувальні формування, які залучаються для виконання відповідних робіт.

У виняткових випадках, коли стихійне лихо, епідемія, епізоотія, аварія чи катастрофа ставить під загрозу життя і здоров'я населення і потребує термінового проведення великих обсягів аварійно-рятувальних і відновлювальних робіт, Президент України може залучати до виконання цих робіт у порядку, визначеному Законом України «Про надзвичайний стан», спеціально підготовлені сили і засоби Міноборони.

На базі існуючих спеціалізованих служб і підрозділів (будівельних, медичних, хімічних, ремонтних та інших) в областях, районах, населених пунктах, підприємствах, установах та організаціях утворюються позаштатні спеціалізовані формування, призначені для проведення конкретних видів невідкладних робіт у процесі реагування на надзвичайні ситуації. Ці формування проходять спеціальне навчання, періодично залучаються до участі у практичному відпрацюванні дій з ліквідації надзвичайних ситуацій.

У виконанні робіт, пов'язаних із запобіганням надзвичайним ситуаціям і реагуванням на них, можуть брати участь також добровільні громадські об'єднання за наявності у представників цих об'єднань відповідного рівня підготовки, підтвердженого в атестаційному порядку. Свої дії вони повинні узгоджувати з територіальними органами та уповноваженими з питань надзвичайних ситуацій та цивільного захисту населення, а роботи виконувати під їх керівництвом.

Надзвичайний стан по всій території України або в окремих її місцевостях вводиться постановою Верховної Ради України з негайним повідомленням Президента України або Указом Президента України, який підлягає затвердженню Верховною Радою України.

Під час надзвичайного стану держава може вживати заходів, передбачених Законом «Про надзвичайний стан», відступаючи від своїх зобов'язань за Конституцією лише настільки, наскільки це вимагається гостротою стану, за умови, що такі заходи не є несумісними з іншими зобов'язаннями за міжнародним правом і не тягнуть за собою дискримінації на основі національності, мови, статі, релігії чи соціального походження.

З метою ліквідації наслідків надзвичайної ситуації у мирний час може проводитись цільова мобілізація. У виняткових випадках допускається залучення працездатного населення і транспортних засобів громадян для виконання невідкладних аварійно-рятувальних робіт за умови обов'язкового забезпечення безпеки праці. При цьому забороняється залучення неповнолітніх, а також вагітних жінок до робіт, які можуть негативно вплинути на стан їхнього здоров'я.

4.2.2 Аварії з викидом шкідливих речовин

Аварії з викидом сильнодіючих отруйних речовин (СДОР) [21] і зараженням навколишнього середовища виникають на підприємствах хімічної, нафтопереробної, целюлозно-паперової, м'ясо-молочної та харчової

промисловості (що мають холодильні установки і які використовують у них в якості холодоагенту речовини типу аміак), водопровідних і очисних спорудах (що використовують хлор), а також при транспортуванні СДОР по залізниці і автомобільних дорогах.

Безпосередніми причинами викиду СДОР є порушення правил зберігання і транспортування, недотримання техніки безпеки, вихід з ладу агрегатів, механізмів, трубопроводів, ушкодження ємностей тощо. При цьому можливе ураження робітників, службовець і інших категорій населення, що знаходяться в районах викиду СДОР.

Сильнодіючими отруйними речовинами називаються хімічні сполуки, які в певних кількостях, що перевищують ГДК, шкідливо впливають на людей, сільськогосподарських тварин, рослини, викликаючи у них ураження різного ступеня [21].

СДОР можуть бути елементами технологічного процесу (аміак, хлор, сірчана й азотна кислоти, фтористий водень) і можуть утворюватись при пожежах на об'єктах господарства (оксид вуглецю, оксид азоту, хлористий водень, сірчистий газ).

У народному господарстві великого поширення набуло використання таких СДОР, як хлор (для знезараження води; при виробництві целюлози на виробництво 1 т целюлози потрібно 40 кг хлору), аміак (при виробництві добрив; як холодоагент у холодильних установках), сірководень, сірковуглець та ін.

Об'єкти, які виробляють СДОР, використовують їх у процесі виробництва, здійснюють їх зберігання, поділяють на 3 ступені хімічної небезпеки. Ступінь хімічної небезпеки визначається видом СДОР та його сумарною кількістю.

Ступінь хімічної небезпеки дає змогу оцінити його з точки зору хімічної небезпеки для населення, сільськогосподарських тварин і навколишнього середовища та розробити засоби їх захисту. На території України у 140 містах та

46 населених пунктах розташовано 877 хімічно небезпечних об'єктів та 287 000 об'єктів використовують у своєму виробництві СДОР чи їх похідні.

Шкідлива дія СДОР на організм людини може проявлятися як в результаті потрапляння таких речовин у краплиннорідкому вигляді на шкіру, так і в результаті вдихання їх парів. За токсичним властивостям СДОР в основному відносяться до групи речовин загальноотруйної та задушливої дії. Симптомами отруєння ними в більшості випадків є головний біль, запаморочення, потемніння в очах, шум у вухах, наростаюча слабкість, задишка, нудота, блювота, а при сильних отруєннях - втрата свідомості, судоми і навіть смерть.

Внаслідок аварій із СДОР утворюється зона хімічного — зараження та осередок хімічного зараження.

Зона хімічного зараження СДОР включає територію, на яку поширюється хмара СДОР. Площі хімічного зараження СДОР визначаються напрямком і швидкістю вітру та іншими параметрами.

Осередок хімічного ураження включає територію, на якій відбулися масові ураження людей, тварин та рослин.

У населених пунктах, лісах, виробничих приміщеннях, підвалах і комунікаційних тунелях стійкість зараження СДОР буде вище, ніж на відкритій місцевості, оскільки вплив вітру проявляється у меншій мірі.

При цьому характер токсичної дії деяких СДОР може змінюватися, наприклад, пари хлору в концентрації 0,1-0,2 мг / л смерть людини можуть викликати при вдиханні їх протягом не менше як 1 години, а пари в концентрації 10-15 мг / л викличуть рефлекторну зупинку дихання при 1-2 вдихах. Деякі з СДОР при високих концентраціях їх парів здатні викликати шкірні ураження людини (з утворенням пухирів).

В першу чергу необхідно захистити органи дихання від подальшої дії сильнодіючих отруйних речовин. На потерпілого необхідно надіти протигаз або ватяну марлеву пов'язку, попередньо змочивши її при отруєнні хлором водою або

5 % розчином питної соди, а при отруєнні аміаком – водою або 2% розчином лимонної кислоти, і винести (вивести) його із зони ураження. У випадках отруєння сильнодіючими отруйними речовинами потерпілому забезпечити спокій і тепло. При виході із зони зараження вимийте очі і відкриті ділянки тіла і зверніться за допомогою до медичного працівника або до медичного закладу.

При отруєнні аміаком винесіть потерпілого з зони зараження, шкіряні покрови, очі, ніс, рот вимийте водою. В очі закапайте дві три краплі 30% розчину альбуциду, в ніс оливкове масло.

При отруєнні хлором винесіть потерпілого із зони зараження. При зупинці дихання зробіть штучне дихання. Шкіряні покрови, рот, ніс вимийте 5 % розчином питної соди або водою.

При отруєнні метаном винесіть потерпілого із зони зараження. При зупинці дихання зробіть потерпілому штучне дихання.

При отруєнні чадним газом винесіть потерпілого із зони зараження, розстебніть комірць одягу. При необхідності зробіть штучне дихання. При необхідності зверніться за допомогою до медичного працівника або відправте потерпілого в медичний заклад.

При отруєнні СДОР дійте у відповідності з отриманими розпорядженнями управління (відділу) з питань надзвичайних ситуацій та цивільного захисту населення області (міста обласного підпорядкування, району).

У системі цивільної оборони розроблена «Методика прогнозування масштабів зараження СДОР при аваріях». Методика дає змогу розрахувати можливу площу хімічного зараження та оцінити можливі втрати людей.

4.2.3 Дії населення в надзвичайних ситуаціях при ураганах

Дії населення при ураганах. (смерч, гроза, злива). З отриманням штормового попередження слід негайно вдатися до проведення запобіжних робіт: зміцнити слабкі конструкції, зачинити двері, вентиляційні отвори. Великі вікна і вітрини

необхідно оббити дошками. Шибки заклеїти смужками паперу або тканини. Двері і вікна з підвітряної сторони залишити відчиненими, щоб урівноважити внутрішній тиск у будівлі. З дахів, балконів, лоджій прибрати предмети, які при падінні можуть заподіяти людям травмувань.

Якщо є можливість і необхідність, треба вимкнути комунальні енергетичні мережі, відкрити допоміжні люки для пропускання води. З легких споруд людей перевести у міцніші будівлі або укрити в захисних спорудах.

Необхідно припинити зовнішні роботи, заpastися електричними ліхтарями, газовими лампами, свічками. Доцільно створити запаси води на 2-3 доби, підготувати похідні плити, примуси, не забути заpastися продуктами харчування і медикаментами, особливо перев'язочними матеріалами; радіоприймачі і телевізори тримати постійно ввімкненими.

Перебуваючи у будинку, слід остерігатися поранень уламками скла, що розлітається. Для цього треба відійти від вікон і встати впритул до простінку. Можна використовувати також міцні меблі. Найбезпечнішим місцем є сховища, підвали або внутрішні приміщення перших поверхів цегляних і кам'яних будинків. Не можна виходити на вулицю одразу ж після послаблення вітру тому що через кілька хвилин порив може повторитися. Якщо все-таки необхідно, треба триматися подалі від будівель і споруд високих парканів, стовпів, дерев, щогл, опор, проводів.

Заборонено знаходитися на шляхопроводах, наближатися, місць зберігання легкозаймистих або сильнодіючих отруйних речовин.

Слід пам'ятати, що найчастіше в таких умовах люди зазнають травмувань від уламків скла, шиферу, черепиці, покрівельного заліза, зірваних шляхових знаків, від деталей оздоблень фасадів і карнизів від предметів, що зберігають на балконах і лоджіях.

Якщо ураган (смерч) застав вас на відкритій місцевості, ліпше за все сховатися у канапі, ямі, яру, будь-якій виїмці: лягти дно заглиблення і щільно

притулитися до землі. Перебувати пошкодженій будівлі небезпечно - вона може обвалитися і новим натиском вітру.

Особливо слід остерігатися розірваних електропроводів виключена ймовірність того, що вони під напругою.

Ураган (смерч) може супроводжуватися грозою, ухиляйтесь від ситуацій, при яких збільшується ймовірність ураження блискавкою: не укривайтеся під деревами, які стоять окремо, не підходьте до ліній електропередач і т. п.

Головна умова – це не піддаватися паніці. Діяти грамотні свідомо, утримуватися від нерозумних вчинків, надавати допомогу, потерпілим.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було розроблено якісний продукт - он-лайн сервіс з використанням сучасних веб-технологій.

В результаті аналізу предметної області було виявлено та виокремлено основні проблеми, підсумувавши які окреслено завдання, які було вирішено, зокрема:

1. визначено оптимально необхідний тип бази даних;
2. встановлено тип з'єднання з базою даних;
3. визначено, які дані зберігати;
4. запропоновано процеси для автоматизації та покращення;
5. обґрунтовано необхідні функції для користувача;
6. описано сутності бази даних і зв'язки між ними;
7. встановлено чи є вся необхідна інформація;
8. сформульовано чим розроблений веб-сайт допоможе користувачам.

Кваліфікаційна робота складається з вступу, чотирьох розділів, висновку, списку літератури та додатків.

Оформлення кваліфікаційної роботи відповідає вимогам та положенням діючих стандартів: ДСТУ 2391-94. «Система технологічної документації. Терміни та визначення»; ДСТУ 3008-95. «Документація. Звіти у сфері науки і техніки. Структура і правила оформлення», а також ЕСКД та іншим чинним стандартам.

Роботу апробовано в рамках VIII науково-технічної конференції «Інформаційні моделі, системи та технології» м.Тернопіль, 2020 р.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Веб-сторінка [Електронний ресурс] – Режим доступу: URL: <https://sites.google.com/site/informatikaforeve/internet/1>
2. Веб-програмування. Серверні скрипти [Електронний ресурс] – Режим доступу: URL: <https://webstudio2u.net/ua/programming/144-server-scripts.html>
3. Система керування контентом [Електронний ресурс] – Режим доступу: URL: <https://hostiq.ua/wiki/ukr/cms-ukr/>
4. Unified Modeling Language Guide User Guide, The (2 ed.). Аддісон-Веслі. 2005. с. 496. ISBN 0321267974.
5. Модель-вид-контролер [Електронний ресурс] – Режим доступу: URL : <https://uk.wikipedia.org/wiki/Модель-вид-контролер>.
6. MVC- -MVP [Електронний ресурс] – Режим доступу: URL: <https://coderoad.ru/-asp-net-MVC--MVP>
7. Система керування базами даних MySQL [Електронний ресурс] – Режим доступу: URL : <http://www.znannya.org/?view=mysql>.
8. Переваги PHP [Електронний ресурс] – Режим доступу: URL: <https://webstudio2u.net/ua/programming/127-programming.html>
9. Знайомство з PHP [Електронний ресурс] – Режим доступу: URL : <http://wiki.kspu.kr.ua/index.php>).
10. Laravel [Електронний ресурс] – Режим доступу: URL : <http://laravel.com/docs/master>
11. Laravel [Електронний ресурс] – Режим доступу: URL : <https://laravel.su/docs/5.0/eloquent>
12. Laravel [Електронний ресурс] – Режим доступу: URL : <https://laravel.su/docs/5.4/eloquent>

13. Why Laravel is Taking the PHP Community by Storm [Електронний ресурс] – Режим доступа : URL : <http://code.tutsplus.com/tutorials/why-laravel-is-taking-the-php-community-by-storm--pre-52639>.

14. Сутність баз даних і джерела їх формування [Електронний ресурс] – Режим доступа : URL : <http://library.if.ua/book/119/7983.html>.

15. Проектування баз даних [Електронний ресурс] – Режим доступа : URL : http://pidruchniki.com/11718/bankivska_sprava/proektuvannya_baz_danih.

16. Лайза Криспин, Джанет Грегори. Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд = Agile Testing: A Practical Guide for Testers and Agile Teams. — М. : «Вильямс», 2010. — 464 с. — (Addison-Wesley Signature Series) — 1000 прим. — [ISBN 978-5-8459-1625-9](https://www.isbn-international.org/view/title/978-5-8459-1625-9).

17. Калбертсон Роберт, Браун Крис, Кобб Гэри. Быстрое тестирование. — М. : «Вильямс», 2002. — 374 с. — [ISBN 5-8459-0336-X](https://www.isbn-international.org/view/title/978-5-8459-0336-X).

18. Бейзер Б. Тестирование чёрного ящика. Технологии функционального тестирования программного обеспечения и систем. — СПб. : Питер, 2004. — 320 с. — [ISBN 5-94723-698-2](https://www.isbn-international.org/view/title/978-5-94723-698-2).

19. Охорона праці : Навчальний посібник [Текст] Геврик Є.О. - Ельга: Ніка-Центр. - 2003. – 279 с.

20. Запобігання виникненню надзвичайних ситуацій [Електронний ресурс] / Wikipedia – URL: https://uk.wikipedia.org/wiki/Запобігання_виникненню_надзвичайних_ситуацій_характеру – 02.12.2015 р.

21. Ураження сильнодіючими отруйними речовинами [Електронний ресурс] / URL: <http://www.wikidocs.ru/preview/44941> – 03.12.2015 р.

22. Методичні вказівки до виконання атестаційної роботи магістра за спеціальністю 121 – Інженерія програмного забезпечення (Освітньо-професійна програма - «Програмне забезпечення систем», Освітньо-наукова програма - «Інженерія програмного забезпечення») для студентів усіх форм навчання / Упор.:

М.Р. Петрик, Д.М. Михалик, О.Ю. Петрик, Г.Б. Цуприк - Тернопіль: ТНТУ, 2020-51с..

23. Дистанційний курс «Основи охорони праці» сайту дистанційного навчання ТНТУ [Електронний ресурс]. – Режим доступу: URL: <http://dl.tntu.edu.ua/index.php>

ДОДАТКИ

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
Факультет комп'ютерно-інформаційних систем і програмної інженерії
Кафедра програмної інженерії

ЗАТВЕРДЖУЮ
Завідувач кафедру
програмної інженерії
Петрик Михайло Романович
“ ___ “ _____ 2020 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання кваліфікаційної роботи
на здобуття освітнього ступеня «магістр»

на тему: «Розробка он-лайн сервісу із застосуванням технології веб-
програмування, мова PHP»
«СПМ-19-226»

Керівник роботи:
к.т.н., доцент
Цуприк Галина Богданівна
“ ___ “ _____ 2020 р.

Виконавець: студент
групи СПМ-61
Залізняк Віктор Володимирович
“ ___ “ _____ 2020 р.

м. Тернопіль – 2020

ЗМІСТ

Вступ

1. Підстави до розробки
2. Призначення до розробки
3. Вимоги до програмного продукту
 - 3.1 Функціональні характеристики
 - 3.2 Склад та параметри технічних засобів
 - 3.3 Інформаційна та програмна сполучність
4. Стадії розробки
5. Програмна документація
6. Порядок контролю та приймання

1 ПІДСТАВИ ДО РОЗРОБКИ

Розробка проводиться у відповідності до графіку навчального процесу на 2020-2021 навчальний рік та згідно наказу на виконання кваліфікаційної роботи на здобуття освітнього ступеня «магістр».

Тема проекту: «Розробка он-лайн сервісу із застосуванням технології веб-програмування, мова PHP».

2 ПРИЗНАЧЕННЯ РОЗРОБКИ

Кваліфікаційна робота присвячена створенню програмного забезпечення – он-лайн сервісу із застосуванням технології веб-програмування з використанням мови програмування PHP.

Метою дослідження є використання сучасних Web-технологій для розробки нового та унікального програмного продукту, з врахуванням вимог та специфіки діяльності конкретного підприємства.

Об'єкт дослідження: спеціалізовані напрямки конкретної галузі та сучасні способи і види її представлення з використанням сучасних інформаційних технологій. Аналіз грамотності, оригінальності та естетичної привабливості, доступності, ергономічності, а також врахування особливостей та деталей, злагодженості роботи, надійності ефективності та безпеки.

Предмет дослідження: програмний веб-продукт у вигляді веб-сайту для конкретного підприємства, розроблений на замовлення та з врахуванням вимог сучасного інформаційного простору та з використанням сучасних технологій. Зокрема, висока швидкодія, підтримка зі сторони більшості хостингів. Простота вставки сценарію PHP в код html-сторінки та відсутність необхідності в додаткових розширеннях браузера, а також зручна система керування.

За результатами виконаної роботи необхідно розробити програмний продукт, який би відповідав завданню.

3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

3.1 Функціональні характеристики

Програмне забезпечення має реалізувати наступні можливості:

- вивід списку всіх оголошень та з використанням фільтрів;
- перегляд оголошення;
- створення, редагування та видалення оголошень;
- перегляд списку своїх оголошень;
- перегляд оголошень доданих до закладок;
- створення закладок на оголошення;
- можливість подати скаргу на оголошення;
- перегляд оголошень на які було подано скаргу;
- блокування оголошення;
- вивід списку оголошень для перевірки;
- можливість обміну повідомленнями між користувачами.

Основні запити які формує користувач будуть

- здійснюватися через графічний інтерфейс,
- надсилатись сервер для процесу обробки.

Після опрацювання результату на запит користувача буде надіслана відповідь.

3.2 Склад та параметри технічних засобів

Функціонування програмної системи забезпечується: ПК x64, з 2 Гб оперативної пам'яті, встановленою системою Microsoft Windows Server 2008/2008 R2//2012/2012 R2/2016/2019, та з не менш, ніж 5 Гб вільного місця на жорсткому диску.

3.3 Інформаційна та програмна сполучність

Програмний продукт повинен коректно функціонувати під керуванням

ОС Microsoft Windows Server 2008/2008 R2//2012/2012 R2/2016/2019.

Програмний продукт повинен бути написаний з використанням мови PHP.

4. СТАДІЇ РОЗРОБКИ

В ході реалізації роботи проєкт повинен пройти крізь наступні стадії розробки:

- сформулювати та погодити з керівником тему роботи;
- розробити та затвердити технічне завдання;
- проаналізувати технічне завдання, підібрати та проаналізувати бібліографічні матеріали необхідні для виконання дипломної роботи;
- провести аналітичний огляд області досліджень;
- викласти загальні підходи та основні методи досліджень, основні методи проектування, розробки архітектури, підсистем і класів, обґрунтування платформ і засобів програмної реалізації;
- розробити допоміжну документацію;
- проаналізувати роботу щодо питань з дотримання положень про охорону праці та безпеку в надзвичайних ситуаціях;
- зробити відповідні висновки за результатами виконаної роботи.

5. ПРОГРАМНА ДОКУМЕНТАЦІЯ

Для програмного продукту повинні бути розроблені наступні документи:

- Пояснювальна записка;
- Технічне завдання;
- Презентаційний матеріал;
- Додатки.

6. ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Розроблений програмний продукт має виконувати всі вимоги, що складаються з перерахованих у п. 3.1 характеристик.

Під час виконання дипломної роботи в дане технічне завдання можуть вноситися зміни та доповнення.

Приймання проводиться спеціально створеною

екзаменаційною комісією №42 в термін до 22 грудня 2020 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

МАТЕРІАЛИ

VIII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



9–10 грудня 2020 року

ТЕРНОПІЛЬ
2020

УДК 004.41

В.В. Залізняк, Г.Б. Цуприк, канд. техн. наук
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

**РОЗРОБКА СУЧАСНОГО ON-LINE СЕРВІСУ
ІЗ ЗАСТОСУВАННЯМ ТЕХНОЛОГІЙ ВЕБ-ПРОГРАМУВАННЯ**

UDC 004.41

V.V. Zalisnyak, H.B. Tsupryk, Ph.D.

**DEVELOPMENT OF MODERN ON-LINE SERVICE
WITH USE OF THE WEB-TECHNOLOGIES**

В загальному про користувачів будь-якого програмного продукту, навіть побутового рівня, яке не говорячи про користувачів, які спеціалізуються в даному напрямку, сьогодні вже можна сказати, що це люди з базовими знаннями та навиками, яких важко здивувати якоюсь додатковими ефектами при оформленні, форматуванні тексту чи графічного представлення. Що ж тоді вже говорити про представництва в всевітній мережі організацій малого чи середнього (не говорячи вже про великий) бізнесу, яке повинне бути не просто формальністю, а ефективним інструментом реалізації підвищення показників успішного ведення діяльності, основним і зрозумілим критерієм оцінки якого є чистий прибуток за мінімальний строк. Варто також зазначити і те, що тенденційно вимоги та очікування суттєво змінюються. Користувачі стають все більш вимогливими і за свої гроші прагнуть отримати все більше та більше можливостей, від найпростіших, так званих базових (зворотній зв'язок, можливість розсилання, пошук по сайту й багато чого іншого), до спеціалізованих, відповідно до специфіки діяльності. І в принципі це логічно, адже саме для цього і існує веб-програмування в одному з його напрямків – створення веб-сайтів.

Однак, при використанні простого html досягнути цього не просто, оскільки html обмежується, як мова, лише розміткою, вважається інструментом для створення гіперосилань, вставки зображень, можливо таблиць та подібне. Основним же, і вважаю, що найважливішим завданням веб-програмування є створення та реалізація інтерактивних компонентів, що є ефективно та обгрунтовано вимогами сьогодення.

Для реалізації ідеї, з можливістю врахування специфіки діяльності конкретного підприємства з метою його гідного представлення та популяризації серед бізнес-партнерів та підвищення конкурентоспроможності у всьому світі, використовуємо серверну мову програмування Нуретекст Преросессор (PHP), як скриптову та призначену безпосередньо для генерування HTML-сторінок на стороні сервера, яка має простий синтаксис, характеризується високою швидкістю та підтримується зі сторони більшості хостингів. І вже тоді PHP інтерпретується веб-сервером в HTML-код, який вже напряму передається у браузер.

Здавалося б – тема не нова, адже існує велика кількість такого типу продуктів. Однак, слід зазначити, що кожен з них є новим, унікальним та відповідає цілям свого замовника, адже всі вони враховують його потреби, вимоги та специфіку і реалізуються з використанням найсучасніших, на час розробки, інформаційних технологій. Також беззаперечним є й той факт, що успішність і затребуваність продукту і є критеріями оцінки його ефективності. Отже, актуальність теми є беззаперечною та на часі.

Література.

1. Переваги PHP [Електронний ресурс] – Режим доступу: URL: <https://php.ru/php/oppoat/>
2. Новини та ревізи PHP [Електронний ресурс] – Режим доступу: URL: <https://habr.com/ru/post/522042/>.

