

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1.ЗАГАЛЬНІ ВІДОМОСТІ ТА ВИМОГИ ДО СИСТЕМИ.....	6
1.1. Вибір інструментарію для розробки проекту	6
1.2. Технічне завдання.....	10
1.3. Висновки до розділу 1	13
РОЗДІЛ 2.ПРОЕКТУВАННЯ WEB-САЙТУ	14
2.1. Опис програмного продукту.....	14
2.2. Опис графічної схеми.....	16
2.3. Проектування бази даних.....	17
2.4. Висновки до розділу 2	22
РОЗДІЛ 3.РОЗРОБКА WEB-САЙТУ ОХОРОННОЇ ФІРМИ.....	23
3.1. Розробка міграції бази даних	23
3.2. Висновки до розділу 3	29
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	30
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	31
ДОДАТКИ	

					<i>КНТЕУ 6.050103 10-18.БР</i>		
Змн.	Арк.	№ докум.	Підпис	Дата			
Зав. Каф.		<i>Криворучко О.В</i>			<i>Розробка web-сайту охоронної фірми «Лабораторія нестандартних рішень» м. Біла Церква</i>	Стадія	Аркуш
Керівник		<i>Палагута К.О.,</i>				Зміст	2
Гарант		<i>Цензура М.О.</i>				<i>Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група</i>	
Розробив		<i>Шабельник І.Я,</i>					
					<i>Зміст</i>		<i>32</i>

ВСТУП

Актуальність теми: на сьогоднішній день створення web-сайтів за допомогою новітніх технологій дозволяє не тільки рекламувати свій бізнес в інтернеті, а також ефективно просувати його через всесвітню павутину, знаходити нових клієнтів і вести співпрацю по всьому земному шару. Важливо відмітити, що створення будь-якого сайту досить кропіткий та важкий процес, який потребує структурованого підходу, чіткого планування цілей, систематизації даних, а також багато іншого. Враховуючи те що наше життя плавно переноситься в всесвітню павутину, проблема створення досить гарного і функціонального веб-сайту є досить значною.

Не менш важливу роль займає і серверна частина web-сайту, логічна частина дозволяє керувати користувачами, надаючи їм інформацію, яка є допустима для розгляду, тобто на сайті можливо розміщувати багато інформації, але доступ до неї можливо обмежити застосовуючи сервер, який фільтрує інформацію, яка повинна показуватися користувачу. Також можлива реалізація реєстрування користувачів у системі з деякими правами. Користувачі як правило заносяться у базу даних і зберігаються там, поки їх не видалить сам адміністратор сайту.

Основні ідеї сучасної інформаційної технології базуються на концепції, згідно з якою дані повинні бути організовані в бази даних з метою адекватного відображення мінливого реального світу і задоволення інформаційних потреб користувачів. Ці бази даних створюються і функціонують під управлінням спеціальних програмних комплексів, званих системами управління базами даних (СКБД).

					КНТЕУ 6.050103 10-18.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка web-сайту охоронної фірми «Лабораторія нестандартних рішень» м. Біла Церква	Стадія	Аркуш	Аркушів
Зав. Каф.		Криворучко О.В				В	3	32
Керівник		Палагуца К.О.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група		
Гарант		Цензура М.О.						
Розробив		Шабельник І.Я,			Вступ			

Сайт дозволяє:

- Збільшити імідж компанії;
- Збільшити довіру до компанії;
- Збільшити потік клієнтів;
- Збільшити продажі;
- Зменшити кількість запитань, які часто задаються по телефону або при зустрічі;

З розвитком інформаційних технологій і підприємництва, актуальність використання баз даних значно збільшилася. Успішні і великі компанії не можуть уявити свій бізнес без чітко побудованої інформаційної системи. Бази даних, побудовані на SQL Server, відповідають високим вимогам продуктивності і безпеки.

Об'єкт дослідження – діяльність охоронної фірми.

Предмет дослідження – редактор коду WebStorm, засоби проектування БД, PhpStorm для реалізації серверної частини сайту, Adobe Illustrator CC 2018 для створення svg картинок.

Мета та завдання проекту – створення веб-сайту для охоронної фірми, розміщення інформації на сайті здійснюється за допомогою мови розмітки HTML, стилізування сайту буде зроблено за допомогою каскадних таблиць стилів CSS, анімація, плавні переміщення будуть реалізовані за допомогою мови програмування JavaScript, а також серверна частина буде реалізована за допомогою мови програмування PHP. По закінченню цього проекту будуть закріплені навички по створенню веб-сайтів за допомогою HTML, CSS, JS, а також написання логіки за допомогою PHP.

Практична значущість – полягає в тому, що цей проект можна буде використовувати для просування фірми на ринку, а також на сайті люди зможуть дізнатися все, що їм потрібно і подати заявку на отримання послуг,

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		4

після подачі заявки люди будуть отримувати рекомендації від фахівця в телефонному режимі.

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		5

РОЗДІЛ 1

ЗАГАЛЬНІ ВІДОМОСТІ ТА ВИМОГИ ДО СИСТЕМИ

1.1. Вибір інструментарію для розробки проекту

Для розробки web-сайту було вирішено обрати технологію «клієнт-сервер». Цей вид технології включає в себе «клієнт» частину, яка розробляється за допомогою мови гіпертекстової розмітки HTML, стилізація виконується за допомогою каскадних таблиць стилів CSS, а також для реалізації деяких завдань спілкування «сайту з клієнтом» використовується мова програмування JavaScript.

Для написання HTML, CSS, JavaScript існують такі інструменти, як редактори коду їх найменування досить різне і функціонал, які в них пропонується досить не відрізняється. Приклади редакторів коду:

1. NotePad++, Sublime Text, WordPad, Brackets, jEdit і багато інших – більшість з них при встановленні не мають плагінів і тому вони досить швидкі, але не завжди досить зручні у використанні. Для комфортного написання коду потрібно встановлювати плагіни, які допомагатимуть в роботі.
2. CodePen, JSFiddle, Liveweave, Plunker, JS Bin, CSS Deck, kodtest – онлайн редактори коду, вони досить зручні у використанні, мають стандартні плагіни, але потрібно мати підключення до інтернету.
3. WebStorm IDE, Visual Studio Code – більш професійні редактори коду, які мають велику кількість переваг, встановлені плагіни, підсвічений синтаксис коду, можливість відновлення великих частин коду, які були видалені при роботі, але при умові, що редактор не був закритий після видалення коду. Для користування ними потрібні сучасні

					КНТЕУ 6.050103 10-18.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка web-сайту охоронної фірми «Лабораторія нестандартних рішень» м. Біла Церква	Стадія	Аркуш	Акрюшів
Зав. Каф.		Криворучко О.В				РІ	6	32
Керівник		Палагута К.О.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група		
Гарант		Цензура М.О.						
Розробив		Шабельник І.Я..			Розділ 1			

потужні комп'ютери, тому що при розробці сайту відкрито не тільки редактор коду, але і браузер, і фотошоп, і т.д.

Для написання клієнтської частини було обрано WebStorm IDE, тому що ресурси комп'ютера дозволяють безболісно використовувати редактор і отримувати максимальну користь від потужності даного редактору.

Для реалізації серверної частини потрібно вибрати мову програмування, відповідний редактор коду, при необхідності використовувати базу даних, але зазвичай БД постійно використовується для збереження даних, які користувач відправляє на сервер. Також локальний сервер, який допоможе реалізувати спілкування клієнтської частини сайту з серверною. Приклади мов для програмування:

- Мови сценаріїв: Python, Ruby, PHP, Node.js, TypeScript.
- Функціональні мови: Elixir, Scala, Erlang, Clojure, Haskell
- Мультипарадигмальності мови: Golang, Rust
- Корпоративні рішення: Java, .NET.

Для реалізації серверної частини була обрана PHP — мова, яка може бути вбудована безпосередньо в html-код сторінок, які, в свою чергу коректно будуть оброблені PHP -інтерпретатором. Механізм PHP просто починає виконувати код після першої екрануючої послідовності (<?) і продовжує виконання до того моменту, коли він зустріне парну екрануючу послідовність (?>).

Також для написання коду був обраний редактор PhpStorm - це інтегроване середовище розробки на PHP з інтелектуальним редактором, яка глибоко розуміє код, підтримує PHP 5.3-7.4 для сучасних і класичних проектів, забезпечує краще в індустрії автодоповнення коду, рефакторинг, запобігання помилок нальоту і підтримує змішування мов.

Наступний крок для початку розробки потрібно було обрати базу

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		7

даних та сервер. MySQL - вільна СУБД. MySQL є власністю компанії Oracle Corporation, що отримала її разом з поглиненою Sun Microsystems, яка здійснює розробку і підтримку додатку. Розповсюджується під GNU General Public License і під власною комерційною ліцензією, на вибір. Крім цього розробники створюють функціональність на замовлення ліцензійних користувачів, саме завдяки такому замовленню майже в нарізноманітніших версіях з'явився механізм реплікації.

FileZilla Server - це сервер який підтримує FTP і FTP через SSL / TLS і забезпечує безпечне шифрування підключення до сервера.

Tomcat - контейнер сервлетів з відкритим вихідним кодом, що розробляється Apache Software Foundation. Реалізує специфікацію сервлетів, специфікацію JavaServer Pages (JSP) і JavaServer Faces (JSF).

Сервер Apache налаштований для роботи з файлами конфігурації, в які додаються директиви для управління його поведінкою. У неактивному стані Apache прослуховує IP-адреси, зазначені в його файлі конфігурації (HTTPd.conf). Всякий раз, коли він отримує запит, він аналізує заголовки, застосовує правила, зазначені для нього в файлі конфігурації, і вживає заходів.

Але на одному сервері може розміщуватися безліч веб-сайтів, а не тільки один, хоча зовнішнього світу вони здаються відокремленими один від одного. Для цього кожному з цих веб-сайтів має бути присвоєно своє ім'я, навіть якщо всі вони в кінцевому підсумку відображаються на одному і тому ж комп'ютері. Це досягається за допомогою так званих віртуальних хостів.

Оскільки IP-адреси важко запам'ятати і відвідувачі певних сайтів, зазвичай вводять відповідні доменні імена в поле URL-адреси в наших браузерах. Потім браузер підключається до DNS-сервера, який переводить

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

доменні імена в їх IP-адреси. Потім браузер бере повернутий IP-адреса і підключається до нього. Браузер також відправляє заголовок Host із запитом, щоб, якщо на сервері було розміщено кілька сайтів, він знав, який з них слід повернути.

Наприклад, ввівши `www.google.com` в поле адреси вашого браузера, можна відправити наступний запит на сервер з цього IP-адресою:

```
01. GET / HTTP/1.1
02. Host: www.google.com
```

Перший рядок містить кілька частин інформації. По-перше, є метод (в даному випадку це GET), URI, який вказує, яку сторінку потрібно витягти або яку програму запустити (в даному випадку це кореневий каталог, позначений /), і, нарешті, є Версія HTTP (в даному випадку це HTTP 1.1).

HTTP є протоколом без запиту стану / відповіді. Це набір правил, які керують зв'язком між клієнтом і сервером. Клієнт (зазвичай, але не обов'язково, веб-браузер) робить запит, сервер відправляє відповідь, і зв'язок припиняється. Сервер не очікує подальшого обміну даними, як у випадку з іншими протоколами, які залишаються в стані очікування після завершення запиту.

Якщо запит виконаний успішно, сервер повертає код стану 200 (що означає, що сторінка знайдена), заголовки відповіді разом із запитаними даними. Тема відповіді сервера Apache може виглядати приблизно так:

```
01. HTTP/1.1 200 OK
02. Date: Sun, 10 Jun 2012 19:19:21 GMT
03. Server: Apache
04. Expires: Wed, 11 Jan 1984 05:00:00 GMT
05. Cache-Control: no-cache, must-revalidate, max-age=0
06. Pragma: no-cache
07. Last-Modified: Sun, 10 Jun 2012 19:19:21 GMT
08. Vary: Accept-Encoding, User-Agent
09. Content-Type: text/html; charset=UTF-8
10. Content-Length: 7560
```

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		9

Перший рядок у заголовку відповіді - це рядок стану. Він містить версію HTTP і код стану. Далі слід дата, а потім деяка інформація про хост-сервері і витягнутих даних. Content-Type Тема дозволяє клієнту знати тип даних, отриманих таким чином, що знає, як впоратися з цим. Content-Length дозволяє клієнту дізнатися розмір тіла відповіді. Якщо клопотання не буде оброблений, клієнт отримає код помилки і повідомлення, наприклад, наступний заголовок відповіді, якщо сторінка не знайдена:

1. HTTP/1.1 404 Not Found

1.2. Технічне завдання

Розробити клієнтську (Front-End) та серверну (Back-End) частину сайту «Laboratory of non-standard solutions».

Клієнтська частина:

1. Необхідний Landing Page, який розповість всім користувачам, потенційним клієнтам, що компанія може запропонувати послуги, за яку ціни і як буде зроблена робота, і в які строки.
2. Шапка сайту – логотип, назва фірми, 4 кнопки «Головна», «Послуги», «Контакти», «Подача заяви». Для незареєстрованих користувачів плюс 2 кнопки: «Вхід», «Реєстрація». Для авторизованих кнопка з «ім'ям» користувача при натисненні по ній відкривається список такими пунктами «Особистий кабінет» та «Вихід»(з облікового запису).
3. Футер сайту – дубльоване меню за шапки сайту по середині, наступна строчка соціальними мережами та месенджерами «Viber», «Facebook», «Instagram». Робочі номери телефонів співробітників фірми, а також робоча пошта та копірайт.
4. Форми для автентифікації користувача: «Реєстрація» та «Авторизація»:

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		10

- а) Реєстрація повинна містити такі поля: «Ім'я», «Пошта (E-mail)» – використовується, як логін для авторизації, «Пароль», а також «Повторити пароль». Повинна бути кнопка «Головна» або іконка з можливістю повернення на головну сторінку сайту, а також кнопка «Увійти» для переходу на сторінку «Авторизації» у разі, якщо користувач вже зареєстрований.
 - б) Авторизація повинна містити такі поля: «Введіть e-mail», «Введіть пароль». Повинен бути чекбокс або кнопка для запам'ятовування користувача у системі. Кнопка «Головна» або іконка. Рядок «Забув пароль» з посиланням на сторінку з формою для скидання паролю.
5. Сторінка з формою для скидання паролю, щоб відновити доступ до облікового запису користувачу у разі втрати паролю.
 6. Сторінка для подачі заяв на отримання послуг. Заяви можуть відправляти лише зареєстровані та авторизовані у системі користувачі, але інформація повинна бути висвітлена для всіх відвідувачів сторінки.
 7. Для зареєстрованих користувачів повинен бути «Особистий кабінет», який повинен містити:
 - а) Вкладка «Профіль» - в ній повинна бути форма, яка буде містити дані про користувача, які були введені при реєстрації.
 - б) Вкладка «Заяви» - в ній повинні відображатися заяви, які користувач подавав для отримання послуг.
 - в) Вкладка «Технічне завдання» - в ній повинні відображатися технічні завдання, які були сформульовані у процесі розмови спеціаліста з замовником послуг. А також над текстом заяви повинна відображатися послуга, яка була вибрана.

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		11

d) Вкладка «Звіт(и)» - в ній повинен відображатися звіт, які був сформульований у після проведення робіт.

8. Створити сторінку «Адмін панель», має містити:

- a) Меню зліва «sidebar» з вкладками – «Головна», «Заяви» (Вхідні, Прочитані), «Технічні завдання», «Користувачі», «Звіти»
- b) «Головна» - містить невелику, але зрозумілу інструкцію, як користуватися адмін панеллю
- c) «Заяви» - Вхідні – містить заяви, які подані користувачами, але вони ще не були розглянуті. Прочитані – містить заяви, які були розглянуті спеціалістом.
- d) «Технічні завдання» - містить форму с 2-ма полями: «Email» користувача, «Розпишіть сформульоване технічне завдання» а також кнопку «Додати тех. завдання користувачеві на сторінку».
- e) «Користувачі» - вивести таблицю з усіма користувачами, на сторінці максимум, щоб було відображено 25 користувачів, наступні 25 на наступній сторінці, реалізувати пагінацію.
- f) «Звіти» - містить форму для заповнення даними для формуванню звіту для користувача.

Спроектувати реляційну базу даних для вирішення конкретних задач, а саме:

- 1. Зберігання користувачів
- 2. Зберігання коментарів залишені користувачами
- 3. Зберігання даних для відновлення паролів користувачів
- 4. Зберігання заяв відправленні користувачами.
- 5. Зберігання технічних завдань
- 6. Зберігання Excel файлів

Серверна частина:

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		12

1. Система автентифікації:

- а) Реєстрація – отримання, валідування та поміщення даних після валідації у таблицю.
- б) Авторизація – перевірка даних введені користувачем, якщо дані правильно, то авторизувати користувача та перекинути його на головну сторінку «Особистого кабінету». Дані введені не правильно – сповістити користувача, що «дані невірні».

2. Система коментарів відправлені користувачами – їх валідація, поміщення у таблицю та публікування на головній сторінці.

3. Скидання паролю – отримання e-mail користувача, який він ввів у форму, звірити з таблицею «користувачі», якщо email співпадають, то відправити на пошту листа з можливістю скидання паролю.

Система заяв – прийняття даних з форми, валідування, поміщення у таблицю та висвітлення цієї заяви у конкретного користувача в «особистому кабінеті», який подав заяву. А також виведення заяви на сторінці адміністратора з додатковими даними «Дата подачі заяви», «Email» користувача.

1.3. Висновки до розділу 1

В першому розділі було описано інструментарій для вирішення поставлених завдань. Була дана характеристика кожному інструменту, а також обрано конкретно, що потрібно для роботи.

Також було поставлене технічне завдання у якому було перераховано основний та побічний функціонал проекту та вимоги до технічної складової проекту.

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		13

РОЗДІЛ 2

ПРОЕКТУВАННЯ WEB-САЙТУ

2.1. Опис програмного продукту

Ідеєю проекту є створення web-сайту для охоронної фірми задля збільшення клієнтської бази та розширення можливостей самої фірми. Сайт створюється для того щоб налаштувати та полегшити взаємодію між потенційним клієнтом та посередником, який після отримання побажань, перенаправляє всю інформацію отриману від клієнта до спеціаліста для вирішення всіх подальших питань, які виникнуть у спеціаліста при розмові з клієнтом.

Клієнт при ознайомленні з сайтом буде мати змогу дізнатися про саму діяльність охоронної фірми. Діяльність буде описана на головній сторінці, яка буде висвітлена у блоці «Про охоронну фірму», наступний блок «Наші послуги» розповість клієнту, які саме послуги надає фірма. Якщо людина зацікавилась однією із послуг, вона зможе перейти до сторінки «Послуги» на якій більш детально висвітлено інформацію про дану послугу і при бажанні отримати цю послугу людина зможе натиснути на кнопку «Подати заяву». Але повернемося до головної сторінки, на якій зможемо побачити, які переваги у даної охоронної фірми описані в блоці «Наші переваги».

Перейдемо до веб-сторінки «Послуги» на якій клієнт зможе прочитати інформацію про послуги більш детально, для кожної послуги відведена своя картка, на цій картці буде назва послуги, картинка, текст який описує дану послугу у повному обсязі та буде присутня кнопка «Подати заяву».

Перейдемо до веб-сторінки «Контакти». На даній сторінці висвітлено інформацію про те як зв'язатися з представниками фірми. Буде

					КНТЕУ 6.050103 10-18.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка web-сайту охоронної фірми «Лабораторія нестандартних рішень» м. Біла Церква	Стадія	Аркуш	Акрюшів
Зав. Каф.		Криворучко О.В				P2	14	32
Керівник		Палагута К.О.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група		
Гарант		Цензура М.О.						
Розробив		Шабельник І.Я.						
					Розділ 2			

представлено таку інформацію «E-mail» фірми та номери телефонів. Також ця інформація доступна у футері даного сайту, але ж хто буде пролистувати сайт до кінця, для того щоб дізнатися просто контакти. Тому вирішено зробити окрему сторінку для того щоб людина змогла знайти контактні дані фірми швидко.

Перейдемо до веб-сторінки «Подача заяви». На даній сторінці буде висвітлена інформація про заяву. Людина пролистуючи дану сторінку зможе дізнатися про процедуру подачі самої заяви, а також після прочитання всіх правил та рекомендацій приступити до написання заяви та її відправлення.

Після відправлення заява потрапляє на сторінку модератора або адміністратора. Після того як заява буде прочитана, дану інформацію відправляють до спеціаліста разом із контактними даними відправника заяви, тобто, клієнта. Спеціаліст після того як ознайомиться з заявою, зв'язується з клієнтом та пропонує вирішити деякі питання, після вирішення, формується технічне завдання, яке передається модератору або адміністратору в письмовому або електронному вигляді.

Сформоване технічне завдання буде завантажено у базу даних. Після цього людина, якщо зайде в свій «особистий кабінет» на сайті, то в розділі «Технічне завдання» буде висвітлене технічне завдання, яке було присвоєно користувачу. В процесі виконання робіт фірмою, на сайті будуть висвітлюватися фотозвіти на сторінці користувача. Людина зможе бачити як проходить процес виконання технічного завдання на об'єкті.

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		15

2.2. Опис графічної схеми

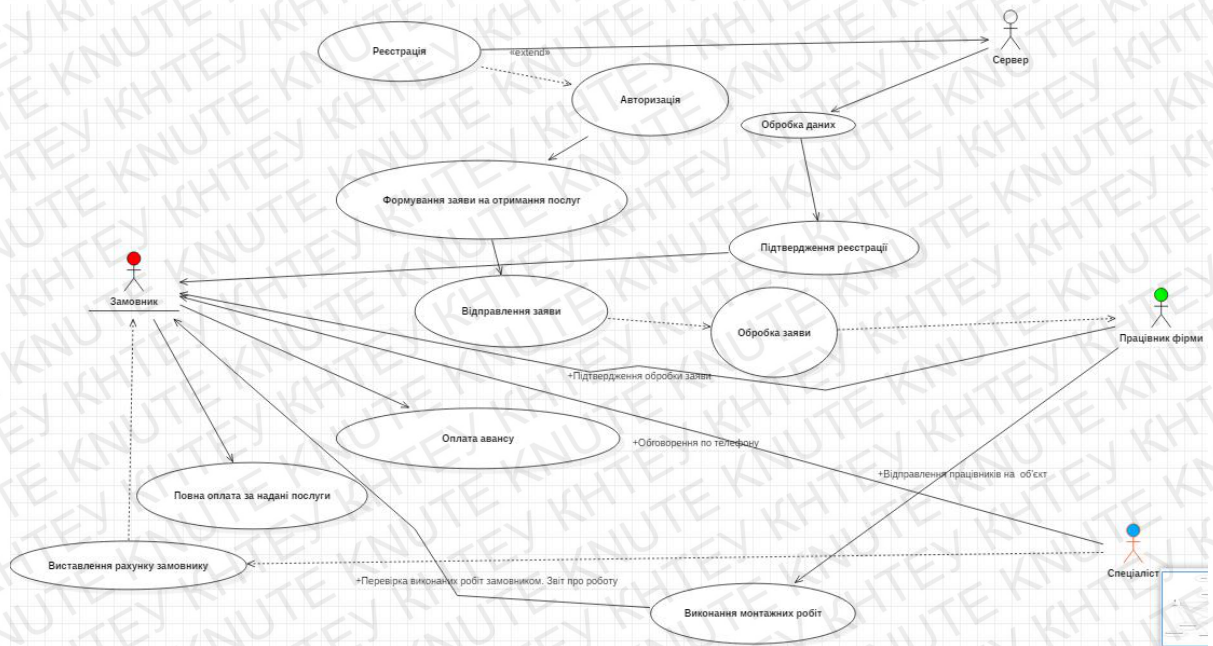


Рис.2.1. Графічна схема спілкування «замовник – працівник фірми»

На схемі зображено весь процес, який буде проходити клієнт. Клієнт потрапляє на сайт і далі відбуватимуться такі кроки:

- 1) Реєстрація людини – клієнт повинен зареєструватися, щоб отримати змогу відправити заяву «на отримання послуг».
- 2) Після реєстрації, як зображено на схемі, дані відправляються на сервер, сервер їх обробляє, та підтверджує реєстрацію людини і перенаправляє на сторінку авторизації.
- 3) Після авторизації, клієнт має змогу написати заяву на отримання послуг.
- 4) Клієнт формулює заяву, вводячи дані про те, що потрібно зробити на об'єкті.
- 5) Заява відправляється на розгляд працівникам фірми. Після обробки заяви працівниками, відправляється підтвердження того, що заява прийнята.
- 6) Всі аспекти та деталі, обговорюються по телефону зі спеціалістом.

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		16

- 7) Виставляється попередній рахунок клієнту. Клієнт оплачує аванс.
- 8) Виконуються монтажні роботи, проводиться налаштування апаратури. Після виконання робіт формується інструкція для користувачів цієї апаратури в подальшому.
- 9) Формується звіт про виконану роботу і надається замовнику, для сплати послуг надані фірмою.

2.3. Проектування бази даних

В ході аналізу технічного завдання, була визначена необхідність у створенні бази даних для зберігання необхідної інформації. У Laravel можна надзвичайно просто взаємодіяти з БД на різних "двигунах", будь то сирові SQL, гнучкий будівник запитів або Eloquent ORM. На даний момент Laravel підтримує чотири системи баз даних:

- MySQL
- Postgres
- SQLite
- SQL Server

Кожна таблиця бази даних має відповідну «Модель», яка використовується для взаємодії з цією таблицею. Моделі дозволяють запитувати дані у таблиці, а також вставляти нові записи в таблицю. Налаштування бази даних для програми зберігається в *config / database.php*. Тут можливо вказати всі використовувані з'єднання до БД, а також задати з'єднання за замовчуванням. Приклади налаштування більшості підтримуваних систем БД знаходяться в цьому ж файлі.

Також всі запити на сервері відбуваються через Eloquent ORM – забезпечує красиву, просту реалізацію ActiveRecord для роботи з базою даних. ActiveRecord надає нам кілька механізмів, найбільш важливими з яких є здібності для:

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		17

- Уявлення моделей і їх даних.
- Уявлення зв'язків між цими моделями.
- Уявлення ієрархій успадкування за допомогою пов'язаних моделей.
- Валідації моделей до того, як вони стануть персистентного в базі даних.
- Виконання операцій з базою даних в об'єктно-орієнтованому стилі.

В ході роботи над технічним завданням, було визначено, що потрібні сформулювати сутності, для подальшого проектування:

1. **Користувачі** – сутність, яка включає в себе інформацію про користувача, який буде реєструватися.
2. **Коментарі** – сутність, яка включає в себе інформацію про коментарі, які будуть залишатися користувачем.
3. **Заяви** – сутність, яка включає в себе інформацію про заяви, які будуть подані зареєстрованими користувачами для отримання послуг.
4. **Файли** – сутність, яка включає в себе інформацію про файли, які будуть завантажуватися адміністратором в бд.
5. **Звіти** – сутність, яка включає в себе інформацію про зроблену роботу.
6. **Технічні завдання** – сутність, яка включає в себе інформацію про технічні завдання, які були сформовані в процесі розмови спеціаліста з замовником.

Після розробки даних сутностей та їх детального аналізу, було визначено, що потрібно створити такі таблиці, як:

- *users* – таблиця для зберігання даних про користувачів

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		18

- *password_resets* – таблиця для зберігання інформації про відновлення паролів для користувачів
- *comments* – таблиця для зберігання коментарів
- *requests* – таблиця для зберігання заяв
- *files* – таблиця для зберігання файлів
- *reports* – таблиця для зберігання звітів
- *technical_tasks* – таблиця для зберігання технічних завдань

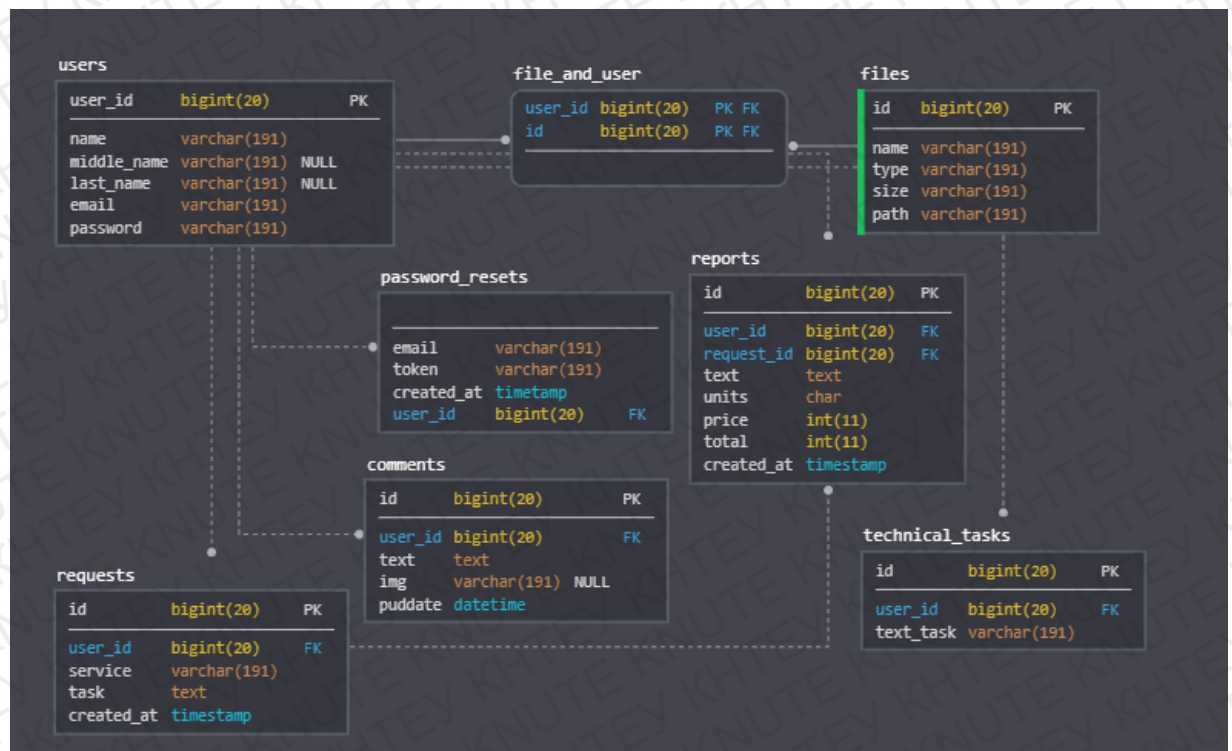


Рис.2.2. Фізична модель бази даних «app_db»

В даній моделі представлені всі таблиці, які були перелічені вище.

Таблиця 2.1.

Користувачі

Атрибути	Тип	Ключ	Опис
user_id	bigint(20)	Primary Key	Номер користувача
name	varchar(191)	-	Ім'я користувача
middle_name	varchar(191)	-	Прізвище

Продовження таблиці «Користувачі»

Атрибути	Тип	Ключ	Опис
last_name	varchar(191)	-	По-батькові
Email	varchar(191)	-	Е-mail користувача
password	varchar(191)	-	Пароль

Таблиця 2.2.

Скидання паролю

Атрибути	Тип	Ключ	Опис
user_id	bigint(20)	Foreign Key	Номер користувача
email	varchar(191)	-	Е-mail користувача
token	varchar(191)	-	Токен користувача
created_at	timestamp	-	Дата скидання

Таблиця 2.3.

Коментарі

Атрибути	Тип	Ключ	Опис
Id	bigint(20)	Primary Key	Номер коментаря
user_id	bigint(20)	Foreign Key	Номер користувача
text	text	-	Коментарі
img	varchar(191)	-	Графічний матеріал
pubdate	datetime	-	Дата публікації

Таблиця 2.4.

Файли

Атрибути	Тип	Ключ	Опис
Id	bigint(20)	Primary Key	Номер файлу
name	varchar(191)	-	Назва файлу
type	varchar(191)	-	Тип файлу
size	varchar(191)	-	Розмір файлу
path	varchar(191)	-	розташування файлу

					<i>КНТЕУ 6.050103 10-18.БР</i>	Аркуш
						20
Зм.	Аркуш	№ докум	Підпис	Дата		

Таблиця 2.5.

Заяви

Атрибути	Тип	Ключ	Опис
Id	bigint(20)	Primary Key	Номер заявки
user_id	bigint(20)	Foreign Key	Номер користувача
request	varchar(191)	-	Заява

Таблиця 2.6.

Користувачі і файли

Атрибути	Тип	Ключ	Опис
Id	bigint(20)	Primary Key	Номер файлу
user_id	bigint(20)	Foreign Key	Номер користувача

Таблиця 2.7.

Звіти

Атрибути	Тип	Ключ	Опис
Id	bigint(20)	Primary Key	Номер звіту
user_id	bigint(20)	Foreign Key	Номер користувача
text	text	-	Текст
units	char	-	Одиниці виміру (шт.,м.)
price	int(11)	-	Ціна
total	int(11)	-	Всього
created_at	timestamp	-	Дата створення звіту

Таблиця 2.8.

Технічне завдання

Атрибути	Тип	Ключ	Опис
Id	bigint(20)	Primary Key	Номер файлу
user_id	bigint(20)	Foreign Key	Номер користувача
text_task	varchar(191)		Технічне завдання

					КНТЕУ 6.050103 10-18.БР	Аркуш
						21
Зм.	Аркуш	№ докум	Підпис	Дата		

2.4. Висновки до розділу 2

В даному розділі були:

1. Описана ідея написання цього проекту, чому саме і для чого це робиться.
2. Продемонстрована схема спілкування «Замовник – працівник фірми» і її детальний опис.
3. Обрана та спроектована база даних MySQL, з описом усіх таблиць, що містить база. Та були описані типи зв'язків, які встановлені між таблицями.

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		22

РОЗДІЛ 3

РОЗРОБКА WEB-САЙТУ ОХОРОННОЇ ФІРМИ

3.1. Розробка міграції бази даних

В Laravel є інструмент який допомагає створювати таблиці в базі даних разом із залежностями – міграції. Міграції забезпечують контроль над базою даних, дозволяючи змінювати структуру бази даних, в той же час залишаючись у курсі змін інших учасників. Міграції зазвичай полегшують роботу з базою даних тому, що взаємодіють з будівником структур для більш простого поводження з архітектурою бази.

Декілька слів про Artisan, який присутній у фреймворці:

Це інтерфейс командної строки (CLI), який входить в склад Laravel. Він представляє ряд команд, які будуть корисні при розробці додатків. Доступних Artisan-команд існує велика кількість, щоб подивитись ці команди потрібно ввести в командну строку PHPstorm: `php artisan list` >> ми отримаємо такий результат:

					КНТЕУ 6.050103 10-18.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка web-сайту охоронної фірми «Лабораторія нестандартних рішень» м. Біла Церква	Стадія	Аркуш	Акрюшів
Зав. Каф.		Криворучко О.В				РЗ	23	32
Керівник		Палагута К.О.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група		
Гарант		Цензура М.О.						
Розробив		Шабельник І.Я.			Розділ 3			

```

C:\xampp\htdocs\www\nonSLS>php artisan list
Laravel Framework 5.4.13

Usage:
  command [options] [arguments]

Options:
  -h, --help            Display this help message
  -q, --quiet            Do not output any message
  -v, --verbose          Display this application version
  --ansi                Force ANSI output
  --no-ansi             Disable ANSI output
  -n, --no-interaction  Do not ask any interactive question
  --env[=ENV]           The environment the command should run under
  -vvv, --verbose       Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  clear-compiled  Remove the compiled class file
  down           Put the application into maintenance mode
  dump-serve     Start the dump server to collect dump information.
  env            Display the current framework environment
  help           Displays help for a command
  inspire        Display an inspiring quote
  list           Lists commands

```

Рис.3.1. Приклад команд, які виконує php artisan

Для того, щоб створити нову міграцію в Laravel потрібно використати Artisan-команду *make:migrate*. Прописуємо: *php artisan make:migrate (назва міграції) create_users_table*

В результаті ми отримали файл під назвою: *2014_10_12_000000_create_users_table.php*

Він має дві функції (*public function up()*, *public function down()*) , які виконують:

1. Створення таблиці
2. Створення полів в таблиці
3. Створення зовнішніх зв'язків.

Тому ми заповнюємо даними ці функції.

					КНТЕУ 6.050103 10-18.БР	Аркуш
						24
Зм.	Аркуш	№ докум	Підпис	Дата		


```

public function up()
{
    Schema::create( table: 'users', function (Blueprint $table) {
        $table->bigIncrements( column: 'id');
        $table->string( column: 'name');
        $table->string( column: 'middle_name')->nullable();
        $table->string( column: 'last_name')->nullable();
        $table->string( column: 'email')->unique();
        $table->timestamp( column: 'email_verified_at')->nullable();
        $table->string( column: 'password', length: 60);
        $table->boolean( column: 'isAdmin')->default( value: false);
        $table->softDeletes();
        $table->rememberToken();
        $table->timestamps();
    });
}

```

Рис.3.2. Функція up() заповнена даними

```

public function down()
{
    Schema::dropIfExists( table: 'users');
}

```

Рис.3.3. Функція down() заповнена даними

Після заповнення даними цих функцій, потрібно виконати команду:
php artisan make:migrate

В результаті ми отримаємо таблицю *users*, яка містить такі поля:

- 1) *Id* – має тип даних *biginteger*, також це поле автозаповнюється (властивість *autoincrement*) при додаванні нового запису у таблицю.
- 2) *Name* – має тип даних *string*, це поле обов'язкове для заповнення.
- 3) *Middle_name* – має тип даних *string*, це поле необов'язкове для заповнення, тому що має властивість *null*.

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		25

- 4) *Last_name* – має тип даних *string*, це поле необов'язкове для заповнення, тому що має властивість *null*.
- 5) *Email* – має тип даних *string*, це поле має властивість *unique*, тобто, в цьому полі неможливо додати два ідентичних e-mail.
- 6) *Email_verified_at* – має тип даних *timestamp*, це поле необов'язкове для заповнення, тому що має властивість *null*.
- 7) *Password* – має тип даних *string*, це поле обов'язкове для заповнення, максимальна кількість символів цьому полі дорівнює 60.
- 8) *Is_admin* – має тип даних *boolean*, це поле за замовчуванням має значення *false* або 0.

За допомогою міграцій було створено ще декілька таблиць *comments*, *password_resets*.

Таблиця тепер готова до використання, в неї можливо записувати дані та висвітлювати їх на сторінці. Запис даних здійснюється за допомогою форм на сторінці.

Рис.3.4. Реєстрація

За допомогою форми «Реєстрація» заповнюється таблиця *users*. Для висвітлення даних використовується друга таблиця *comments*. Заповнення таблиці відбувається за допомогою цієї форми:

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		26

Оставьте свой комментарий

Ваше имя

Simon

Напишите отзыв

Ваш комментарий

Отправить

Рис.3.5. Форма для коментарів

Коментарии

Антонин

Et veritatis aperiam ullam exercitationem aliquam excepturi. Recusandae quaerat provident voluptatem repudiandae. Voluptas porro velit iusto dicta quis expedita est. Non omnis blanditiis est temporibus iure in dolore.

Рис.3.6. Приклад опублікованого коментаря

Заповнення даними таблиць для контрольного прикладу використовується Artisan-команда: `php artisan db:seed`. Але перед тим, як виконати цю потрібно створити файл `factory`. В даному файлі ми вказуємо дані, які потрібно згенерувати та вставити в поля таблиць, наприклад:

```
$factory->define(User::class, function (Faker $faker) {  
    return [  
        'name' => $faker->firstName,  
        'email' => $faker->unique()->safeEmail,  
        'email_verified_at'=>now(),  
        'password'=>'$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/a  
t2.uheWG/igi',      'remember_token' => Str::random(10),  
    ];  
});
```

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		27

});

В даному прикладі вказано, що поля:

- *'name'* – за допомогою *\$faker* потрібно заповнити даними із *first_name* (в даному випадку *first_name* - це просто імена людей).
- *'email'* – за допомогою *\$faker* потрібно заповнити даними із *safeEmail*, а при вказана властивість *unique()* – це означає, що e-mail адреса, якими заповнюються поля, будуть унікальні, тобто, різні.
- *'email_verified_at'* – вказана властивість *now()* – це означає, що при генеруванні даних, ця колонка буде заповнюватися датою та часом, які вказані на сервері на даний момент, час вказується з точністю до секунди.
- *'password'* – поле заповнюється вказаним рядком (*'\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi'*) для всіх тестових користувачів
- *'remember_token'* – заповнюється символами довжиною 10 симлолів.

Після того як дані вказані, потрібно цей *factory* викликати за допомогою функції *run()* у файлі під назвою *UsersTableSeeder.php*:

```
public function run()
{
    factory(App\User::class, 50)->create();
}
```

В цій функції вказано, що ми викликаємо *factory*, який відноситься до моделі *User* та виконуємо цей *factory* 50 разів, тобто, 1 виклик *factory* дорівнює тому, що поля в таблиці *users* 1 раз заповняться даними і відповідно 50 викликів призведе до того, що таблиця заповниться 50-ма різними даними. Приклад *factory* та *seeder* для таблички коментарі

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		28

(comments).

3.2. Висновки до розділу 3

В даному розділі було:

1. Описано практичну частину проектної роботи.
2. Були створені файли міграцій, які були використані для створення таблиць в базі даних. Заповнення цих таблиць контрольним прикладом відбувалось за допомогою *db:seed*, яка описана вище в цьому розділі.
3. Розроблено адаптивні сторінки сайту, форми реєстрації, авторизації, коментарів і заяв.

					КНТЕУ 6.050103 10-18.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		29

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Розробка web-сайту повинна відбуватися поетапно, визначили предметну область, визначено інформацію, яку необхідно буде висвітлювати на сайті для контрольного прикладу.

Був проведений аналіз діяльності охоронної фірми «Лабораторія нестандартних рішень» її особливості функціонування, сформульовані вимоги до оформлення дизайну web-сайту. В ході аналізу технічного завдання, було вирішено використати для реалізації даного завдання такі інструменти, як мова гіпертекстової розмітки HTML, каскадні таблиці стилів CSS, точніше був використаний framework з готовими стилями bootstrap 4, але в ході стилізації web-сторінок було вирішено, використовувати тільки систему сітки, яка реалізована в bootstrap, а кольори, картини використовувати відповідно до поставленої задачі. Для реалізації деякого функціоналу на стороні клієнта було використано мову програмування Javascript, а для реалізації динамічності сайту було вирішено використовувати скриптову мову програмування PHP, для генерації HTML-сторінок, точніше був використаний шаблонізатор Blade.

Дана проектна робота виконана в повній відповідності до поставленого завдання.

В результаті роботи ми отримали працездатний web-сайт, перевагами якого є презентабельність, інформація, яка описує діяльність охоронної фірми. Сам сайт можливо доробити для розширення функціоналу і зручності користування цим сайтом в подальшому.

					<i>КНТЕУ 6.050103 10-18.БР</i>		
Змн.	Арк.	№ докум.	Підпис	Дата			
Зав. Каф.		Криворучко О.В			Розробка web-сайту охоронної фірми «Лабораторія нестандартних рішень» м. Біла Церква	Стадія	Аркуш
Керівник		Палагута К.О.				ВП	30
Гарант		Цензура М.О.				Акрюшів	
Розробив		Шабельник І.Я.				32	
					Висновки та пропозиції	Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група	

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. O'Brien J.A. Management Information Systems: managing information technology in the internetworked enterprise. / J.A. O'Brien– Glencoe, McGraw-Hill, 1999. – 795 p
2. Gary E. Programmer: PHP Today / E. Gary, Ph.D. Clayton. – Glencoe, McGraw-Hill, 2008. – 1050 p
3. Денис Колисниченко. PHP и MySQL. Розробка Web-додатків
4. PHP and MySQL Web Development
5. High Performance MySQL
6. PHP і MySQL. Вичерпне керівництво

Інтернет ресурси

1. Режим доступу «Теорія розробки веб-сайту»: https://pidruchniki.com/2015082665983/informatika/teoriya_rozrobki_veb-saytu
2. Режим доступу «Аналіз діяльності охоронних фірм» : <https://pravdop.com.ua/publications/kommentarii-zakonodatelstva/normativnoe-regulirovanie-ohrannoy-deyatelnosti/>
3. Режим доступу «Діяльність охоронної фірми» : https://www.sheriff.com.ua/uk/items_ua/liz22/
4. Режим доступу «Проектування веб-сайту» : <https://www.bibliofond.ru/view.aspx?id=701071>
5. Режим доступу «Робота з bootstrap» : <https://bootstrap-4.ru/docs/4.2.1/getting-started/introduction/>
6. Режим доступу «Основні вимоги до створення веб-сайтів» : <https://recommerce.com.ua/vimogi-do-stvorenniya-saitu>

					КНТЕУ 6.050103 10-18.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка веб-сайту охоронної фірми «Лабораторія нестандартних рішень» м. Біла Церква	Стадія	Аркуш	Акрюшів
Зав. Каф.		Криворучко О.В				СД	31	32
Керівник		Палагута К.О.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 10 група		
Гарант		Цензура М.О.						
Розробив		Шабельник І.Я.						
					Список використаних джерел			

7. Режим доступу «Офіційна документація PHP»:

<https://www.php.net/>

8. Режим доступу «Робота з Composer (документація)»:

<https://getcomposer.org/doc/>

					<i>КНТЕУ 6.050103 10-18.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		32

Лістніг коду: «create_comments_table»

```
<?php

use Illuminate\Support\Facades\Schema;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Database\Migrations\Migration;

class CreateCommentsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */

    public function up()
    {
        Schema::create('comments', function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->unsignedBigInteger('user_id');
            $table->text('comment');
            $table->timestamps();

            $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');

            $table->index('user_id');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */

    public function down()
    {
        Schema::dropIfExists('comments');
    }
}
```


Лістинг коду: «create_password_resets_table»

```
<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreatePasswordResetsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('password_resets', function (Blueprint $table) {
            $table->string('email')->index();
            $table->string('token');
            $table->timestamp('created_at')->nullable();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('password_resets');
    }
}
```

Лістнінг коду: «CommentFactory»

```
<?php

/* @var $factory \Illuminate\Database\Eloquent\Factory */

use App\Comment;

use Faker\Generator as Faker;

$factory->define(Comment::class, function (Faker $faker) {
    return [
        'user_id' => mt_rand(1, 50),
        'comment' => $faker->text(300),
    ];
});
```

