

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

факультет інформаційних технологій

(факультет)

Кафедра\_\_ інформаційних технологій та комп'ютерної інженерії \_\_\_\_  
(повна назва)

**ПОЯСНЮВАЛЬНА ЗАПИСКА**  
кваліфікаційної роботи ступеня \_\_\_\_ бакалавра \_\_\_\_  
(бакалавра, спеціаліста, магістра)

Студента \_\_\_\_\_ Юхимчука Дениса Петровича \_\_\_\_\_  
(ПІБ)

академічної групи \_\_\_\_\_ 126-17-1 \_\_\_\_\_  
(шифр)

спеціальності \_\_\_\_\_ 126 « Інформаційні системи та технології» \_\_\_\_\_  
(код і назва спеціальності)

за освітньо-професійною програмою \_\_\_\_\_  
«Інформаційні системи та технології»  
(офіційна назва)

на тему Комплексна кваліфікаційна робота: Проектування інформаційного веб-сайту  
установи "Стоматологічна клініка VlaDent"  
(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	проф. Коротенко Г.М.			
розділів:				

Рецензент	доц. Коротенко Л.М.			
-----------	---------------------	--	--	--

Нормоконтролер	проф. Коротенко Г.М.			
----------------	----------------------	--	--	--

Дніпро  
2021

**ЗАТВЕРДЖЕНО:**

завідувач кафедри

інформаційних технологій

та комп'ютерної інженерії

(повна назва)

Гнатушенко В.В.

(підпис)

(прізвище, ініціали)

« \_\_\_\_\_ » \_\_\_\_\_ 2021 року

**ЗАВДАННЯ**  
**на кваліфікаційну роботу**  
**ступеня бакалавра**  
(бакалавра, спеціаліста, магістра)

студенту Юхимчуку Д. П. академічної групи 126-17-1  
(прізвище та ініціали) (шифр)

спеціальності 126«Інформаційні системи та технології»

за освітньою-професійною програмою \_\_\_\_\_

«Інформаційні системи та технології»

на тему Комплексна кваліфікаційна робота: Проектування інформаційного веб-сайту установи "Стоматологічна клініка VlaDent"

затверджену наказом ректора НТУ «Дніпровська політехніка» від 27.04.2021 № 234-с

Розділ	Зміст	Термін виконання
Розділ 1	АНАЛІЗ СТАНУ ОБЛАСТІ РІШЕННЯ ЗАДАЧІ	10.02.2021 – 10.03.2021
Розділ 2	ПРОЕКТНІ РІШЕННЯ	11.03.2021 – 15.04.2021
Розділ 3	РЕАЛІЗАЦІЯ ПРОЕКТНИХ РІШЕНЬ	16.04.2021 – 14.05.2021

Завдання видано \_\_\_\_\_ Коротенко Г.М.  
(підпис керівника) (прізвище, ініціали)

Дата видачі 01.02.2021 р.

Дата подання до екзаменаційної комісії 24.06.2021 р.

Прийнято до виконання \_\_\_\_\_ Юхимчук Д.П.  
(підпис студента) (прізвище, ініціали)

## РЕФЕРАТ

**Пояснювальна записка:** 122 стор., 48 мал. 5 додатків, 21 джерел, 8 таб.

**Об'єкт розроблення:** Проектування інформаційного веб-сайту установи 'стоматологічна клініка VlaDent'.

**Мета кваліфікаційної роботи:** Створення цифрового сервісу для надання інформаційних послуг з допомогою фреймворка Bootstrap та Python фреймворку Django.

У вступі подані причини створення веб-сайту для підприємства, короткий опис поняття "веб-сайт", значення FrontEnd`у та BackEnd`у та їх інструментів для розробки.

У першому розділі наведені технології, які будуть використані для виконання цілей кваліфікаційної роботи, визначені основні поняття та означення, які використовуються в даній роботі. У роботі використовуються сучасні підходи та технології для роботи над інформаційним сервісом.

У другому розділі наведені технічні завдання для роботи. Вимоги до функціональних характеристик, підстави для розробки.

У третьому розділі представлена реалізація проектних рішень, а саме: Створення сховища в GitHub та розподіл ролей та завдань для розробки. Створення таблиць в БД, приклади таблиць та схеми. Архітектурна схема сайту та більш детальний опис виконаної кваліфікаційної роботи.

Практичне значення кваліфікаційної роботи полягає у тому що, створений інформаційний веб-сайт допомагає підприємству у наданні інформаційних послуг, детального опису послуг, немає необхідності користуватись друкарнями, доступ до сайту можливо отримати 24/7.

**Ключові слова:** ВЕБ-САЙТ, BOOTSTRAP, DJANGO, СУБД, GIT, FRONTEND, BACKEND, ФРЕЙМВОРК, БД

## ABSTRACT

**Explanatory note:** 122 pages, 48 fig. 5 appendices, 21 sources, 8 tab.

**Object of development:** Design of the information website of the institution 'VlaDent Dental Clinic'.

**Purpose of the qualification work:** Creation of a digital service for providing information services using the Bootstrap framework and the Python framework Django.

The first section presents the technologies that will be used to fulfill the objectives of the qualification work, defines the basic concepts and definitions used in this work. The work uses modern approaches and technologies to work on the information service.

The second section provides specifications for the work. Requirements for functional characteristics, grounds for development.

The third section presents the implementation of design solutions, namely: Creating a repository in GitHub and the distribution of roles and tasks for development. Creating tables in the database, examples of tables and diagrams. Architectural scheme of the site and a more detailed description of the qualification work.

The practical significance of the qualification work is that the created information website helps the company to provide information services, a detailed description of services, there is no need to use printing houses, access to the site can be obtained 24/7.

**Keywords:** WEBSITE, BOOTSTRAP, DJANGO, DBMS, GIT, FRONTEND, BACKEND, FRAMEWORK, DB

## ЗМІСТ

ВСТУП .....	8
РОЗДІЛ 1. АНАЛІЗ СТАНУ ОБЛАСТІ РІШЕННЯ ЗАДАЧІ.....	10
1.1 Вибір засобу розробки для FrontEnd`у.....	10
1.1.1 Поняття FrontEnd.....	10
1.1.2 Компоненти для розробки FrontEnd`у.....	10
1.1.3 Поняття фреймворка.....	11
1.1.4 Вибір css фреймворку. Bootstrap. Foundation. TailwindCSS.....	12
1.1.5 Фреймворк Bootstrap.....	14
1.1.6 Переваги фреймворка Bootstrap.....	16
1.2 Вибір засобу розробки для BackEnd`у.....	18
1.2.1 Поняття BackEnd.....	18
1.2.2 Поняття BackEnd-фреймворку.....	18
1.2.3 Опис фреймворка Django.....	19
1.3 Вибір СУБД.....	21
1.3.1 Поняття СУБД.....	21
1.3.2 Вибір СУБД. MYSQL. MongoDB. PostgreSQL.....	21
1.4 Вибір хостингу.....	22
1.5 Вибір системи управління версіями.....	23
1.5.1 Поняття Системи Управління Версіями GIT.....	23
1.5.2 Вибір Системи Управління Версіями GitHub.....	24
РОЗДІЛ 2. ПРОЕКТНІ РІШЕННЯ.....	25
2.1. Технічне завдання.....	25
2.1.1. Найменування і область застосування.....	25
2.1.2. Підстави для розробки.....	25
2.1.3. Призначення розробки.....	25

2.1.4. Вимоги до функціональних характеристик.....	25
2.1.5. Вимоги до надійності.....	25
2.1.6. Вимоги до складу і параметрів технічних засобів.....	26
2.1.7. Вимоги до інформаційної та програмної сумісності.....	26
2.1.8. Вимоги до програмної документації.....	26
<b>РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЕКТНИХ РІШЕНЬ.....</b>	<b>27</b>
3.1 Створення сховища в GitHub.....	27
3.1.1 Розподіл ролей.....	27
3.1.2 GitFlow.....	27
3.2 Створення таблиць в БД.....	29
3.2.1 Приклад створення таблиці в БД.....	29
3.2.2 Схема БД.....	30
3.2.3 Список таблиць.....	31
3.3 Архітектурна схема сайту.....	37
3.3.1 Алгоритм створення бази даних.....	37
3.3.2 Відображення сторінки.....	37
3.3.3 Опис розробленого сайту і його інтерфейсу.....	40
<b>ВИСНОВКИ .....</b>	<b>50</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>51</b>
Додаток А. Відомість матеріалів кваліфікаційної роботи.....	54
Додаток Б. Приклади коду програми.....	55
Додаток В. Відгук керівника кваліфікаційної роботи.....	118
Додаток Г. Рецензія.....	120
Додаток Д. Довідка.....	122

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

XHTML - extensible hypertext markup language (розширювана мова гіпертекстової розмітки)

БД - баз даних

СУБД - системи управління базами даних

WEB & WWW – World Wide Web (Всесвітня павутина)

SSL - Secure Sockets Layer (рівень захищених сокетів)

GIT - Global Information Tracker (система управління версіями)

PaaS - Platform as a Service (платформа як послуга)

ASGI - Asynchronous Server Gateway Interface (клиент-серверний протокол взаємодіяння веб-сервера и приложения)

XML - XML User Interface Language (мова розмітки для створення динамічних призначених для користувача інтерфейсів)

Веб-сервер - сервер, що зв'язують вхідні http запити з програмною частиною сайту.

Фронтенд - публічна частина веб сайту.

Бекенд - програмно-апаратна частина проекту.

## ВСТУП

Згідно освітньо-професійною програмою (ОПП) і освітньо-кваліфікаційної характеристики (ОКХ) спеціаліста по напрямом підготовки «126 - Інформаційні системи та технології», в число функціональних завдань і, відповідно, компетенцій фахівця входять множинні елементи проектування, організації і реалізації функцій доступу до БД. В даному випадку, отримані навички будуть розкриті на прикладі розробки веб-проекту з використанням системи управління базами даних (СУБД).

### Навіщо підприємству веб-сайт?

1. Веб-сайт - це місце, де можна опублікувати детальну інформацію про своє підприємство, послуги, умови замовлення та виконання послуг, контактах і реквізити. Сайт інформує клієнтів круглий рік і вночі і вдень. На сайті легко публікувати інформацію, яку неможливо розмістити ні в жодному іншому вигляді традиційної реклами. Змінювати інформацію на сайті дуже просто. І при цьому не потрібно робити новий буклет, віддавати його в друкарню і розсилати клієнтам.

2. Веб-сайт - це інструмент, за допомогою якого можна залучити потенційних клієнтів і спростити початковий процес замовлення ваших послуг. Потенційні клієнти за допомогою каталогів і пошукових систем за ключовими словами знайдуть сайт, ознайомляться з послугами та умовами і можуть захотіти стати вашими клієнтами. Відвідувачам буде нескладно заповнити на сайті спеціальну форму-заявку і вказати в ній свої реквізити і побажання щодо послуги.

3. Веб-сайт - працює на імідж підприємства. Наявність веб-сайту свідчить про технічну просунутість компанії та відповідність вимогам часу.

Таким чином, метою виконання даного дипломного проекту є розробка веб-сайту для підприємства - стоматологічної клініки “VlaDent”.



Backend сайту розроблений на Python з використанням фреймворку Django, котрий являє собою набір інструментів для розробки, тестування та деплою. За допомогою сайту можливо додавати, змінювати та видаляти новини, лікарів, ціни на послуги стоматології та контакти клініки.

Frontend сайту розроблений за допомогою Bootstrap, який представляє собою набір інструментів для верстки та використовує найсучасніші напрацювання в області HTML, CSS і JS. Також за допомогою Bootstrap можлива адаптивна багатоколонкова верстка.

## РОЗДІЛ 1

### АНАЛІЗ СТАНУ ОБЛАСТІ РІШЕННЯ ЗАВДАННЯ.

#### 1.1 ВИБІР ЗАСОБУ РОЗРОБКИ ДЛЯ FRONTEND`У

##### 1.1.1 Поняття FrontEnd

Frontend - це публічна частина web-додатків, з якої користувач може взаємодіяти і контактувати напряму. У Frontend входить відображення функціональних завдань, призначеного для користувача інтерфейсу, що виконуються на стороні клієнта, а також обробка запитів користувачів. По суті, фронтенд - це все те, що бачить користувач при відкритті web-сторінки.

У свою чергу, web-додаток - клієнт-серверний додаток, в якому клієнтом виступає в основному браузер, а сервером - web-сервер. Логіка web-додатки розподілена між сервером і клієнтом, зберігання даних здійснюється переважно на сервері, обмін інформацією відбувається по мережі. Простіше кажучи, це те, що бачить користувач і які дії виконує кожен раз, коли підключається до мережі інтернет і відкриває будь-який браузер.

Frontend-розробка - це робота зі створення публічної частини web-додатки, з якої безпосередньо контактує користувач, і функціоналу, який зазвичай виконується на стороні клієнта. Тобто, фронтенд розробник працює над тим, щоб на сайті кожна кнопка, іконка, текст і вікно не тільки стояли на своєму місці, не перекривали один одного і виглядали цілісно, але і щоб вони виконували своє пряме призначення - виробляли якісь дію (наприклад, щоб кнопка "купити" відкривала кошик, а "play" - запускала відтворення фільму або музики).[1]

##### 1.1.2 Компоненти для розробки FrontEnd`у

На поточний момент є багато інструментів та фреймворків для роботи з фронт-ендом але існують три основних інструмента без яких не обійдеться ні жоден сайт.

- **HTML (HyperText Markup Language)** - мова, на якому пишуться самі веб-сторінки. Саме з його допомогою текст розбивається на абзаци і заголовки, створюються списки і таблиці, на сторінки містяться графічні зображення, аудіота відеоролики. [2]
- **CSS (Cascading Style Sheets)** - мова, на якому описується оформлення веб-сторінок: яким шрифтом потрібно вивести текст абзців, а яким - заголовків, яким фоном слід залити панель навігації, який рамкою потрібно її оточити і як зробити так, щоб при наведенні курсору миші на гіперпосилання вона плавно змінювала колір (тобто створити анімацію). [2]
- **JavaScript** - мультіпарадігменна мова програмування. Підтримує об'єктно-орієнтована, імперативний і функціональний стилі. Є реалізацією специфікації ECMAScript. [4]

JavaScript зазвичай використовується як вбудований мова для програмного доступу до об'єктів додатків. Найбільш широке застосування знаходить в браузерах як мова сценаріїв для додання інтерактивності веб-сторінок. [4]

### 1.1.3 Поняття фреймворка

**Фреймворк** - це платформа, яка надає розробникам основу, для створення програмного забезпечення. Він містить заздалегідь визначені і реалізовані класи або функції. Також, для конкретних завдань можна додавати власний код до того, що вже міститься у фреймворку. [5]

Основні плюси фреймворків:

- **Продуктивність** - значно покращують продуктивність, що максимально оптимізує процес. Тому що, простіше використовувати фреймворк з оптимізацією і добре структурованими шаблонами, ніж писати сотні рядки коду.
- **Рентабельність** - безкоштовні і мають відкритий вихідний код для використання. Оскільки, це значно прискорює створення додатка, то, відповідно і зменшує ціну самого веб-дodatки.

- **Безпека** - Трендові JavaScript-фреймворки можуть похвалитися не тільки підтримкою великої спільноти на GitHub, а й залізної системою безпеки.

#### 1.1.4 Вибір css фреймворку. Bootstrap. Foundation. Tailwind CSS.

Створення красивих стилів вбиває багато часу. CSS фреймворки виконують ту ж задачу на льоту. Верстальники отримують продумані вирішення найнагальніших проблем - розмітку, інтерфейси, адаптивність.[6]

Класичний CSS фреймворк - набір базових стилів для верстки веб-сторінки:

- сітка;
- іконки;
- таблиці;
- елементи форм і кнопок;
- типографіка;
- інтерфейсні патерни, наприклад, картки і модальні вікна;
- допоміжні класи оформлення елементів: відступи, кольору і т. д.

Розглянемо популярні css фреймворки для розробки FrontEnd`у:

**1) Bootstrap** - вільний набір інструментів для створення сайтів і веб-додатків. Включає в себе HTML- і CSS-шаблони оформлення для типографіки, веб-форм, кнопок, міток, блоків навігації та інших компонентів веб-інтерфейсу, включаючи JavaScript-розширення. [7] Приклад логотипу наведений на рис.1.



Рис. 1. Логотип Bootstrap

Популярна бібліотека компонентів. У 2021 році вийшов Bootstrap 5. У ньому ще більше зручних утиліт і модифікаторів. П'ята версія написана на SASS, так що можна використовувати в розробці всі переваги препроцесорів.

Головні плюси Bootstrap:

- Опрацьована адаптивна сітка, заснована на Flex-моделі, пройшла випробування часом на всіх можливих дозволах. Вирішує більшість завдань компоновання контенту.
  - На Bootstrap створені тисячі шаблонів, тим і готових компонентів.
  - Можна зібрати сайт, не написавши жодного рядка CSS-коду.
  - Простота навчання. Очевидні концепції, детальна документація з прикладами. Маса посібників для розробників з різним ступенем занурення в тему.
  - Готовий JavaScript для більшості інтерактивних компонентів. •
- Акордеони і каруселі вже написані за вас.
- Препроцесори SASS і LESS.

**2) Foundation** - це чуйний інтерфейс. Foundation надає адаптивну сітку і HTML і CSS компоненти користувальницького інтерфейсу, шаблони і фрагменти коду, включаючи типографіку, форми, кнопки, навігацію та інші елементи інтерфейсу, а також додаткові функції, що надаються розширеннями JavaScript. [8] Приклад логотипу наведений на рис. 2.



Рис. 2. Логотип Foundation

Витончений і гнучкий інструмент підійде для дуже великих проектів. Ним користуються Facebook, eBay, Mozilla, Adobe, HP, Cisco і Disney.

Головні плюси Foundation:

- Адаптивна система сіток. У створенні чуйного дизайну не поступається Bootstrap.
- Потужний email-фреймворк. Responsive-дизайн не тільки в веб-додатках, але і в листах. Ніяких громіздких табличок.
- Проста кастомизація. Foundation гнучкий. Ви повністю контролюєте інтерфейс проекту.
- JavaScript-компоненти.
- Легке створення анімацій.
- Вертикальний контроль розмітки.
- Інструменти для розробника.
- Препроцесор SASS.

**3) Tailwind css** - самоописується як утиліта першого фреймворка CSS. Замість того, щоб зосередити увагу на функціональності елемента, що стилізується, Tailwind зосереджується навколо того, як його слід відобразити. Це полегшує розробнику тестувати нові стилі та змінювати макет. [9] Приклад логотипу наведений на рис. 3.



Рис. 3 Логотип Tailwind CSS

Низькорівневий CSS фреймворк, відкриті всі можливості кастомізації.  
Tailwind CSS підходить для нестандартних дизайн-рішень.

Головні плюси Tailwind css:

- Безліч утилітарних класів.
- Проста кастомізація елементів.
- Набір адаптивних опцій.

### **1.1.5 Фреймворк Bootstrap**

Bootstrap включає в себе безліч різних компонентів для веб-сайтів: типографіку, веб-форми, кнопки, блоки навігації і інше.

Реліз відбувся в 2011 році, потім в 2012 вийшла друга версія, в 2013 - третя і, нарешті, в 2018 - четверта. Кожна з версій має істотні зміни.

Спочатку Bootstrap носив назву Twitter Blueprint, так як створювався компанією Twitter. Це відмінний приклад того, як розробляється для внутрішнього використання інструмент стає в результаті затребуваним по всьому світу.

Після того, як проект був викладений у відкритий доступ, його стали використовувати багато розробників.[10]

#### **Основні інструменти Bootstrap:**

- Сітки - заздалегідь задані розміри колонок, які можна відразу ж використовувати.
- Шаблони - фіксований або гумовий шаблон документа.
- Типографіка - опису шрифтів, визначення деяких класів для шрифтів, таких як код, цитати і т. П.
- Медіа - надає деяке упорядкування зображень та відео.

- Таблиці - кошти оформлення таблиць, аж до додавання функціональності сортування.
- Форми - класи для оформлення форм і деяких подій, що відбуваються з ними.
- Навігація - класи оформлення для панелей, вкладок, переходу по сторінках, меню і панелі інструментів.
- Алерт - оформлення діалогових вікон, підказок і спливаючих вікон.

Приклад структури файлів наведений на рис. 4.

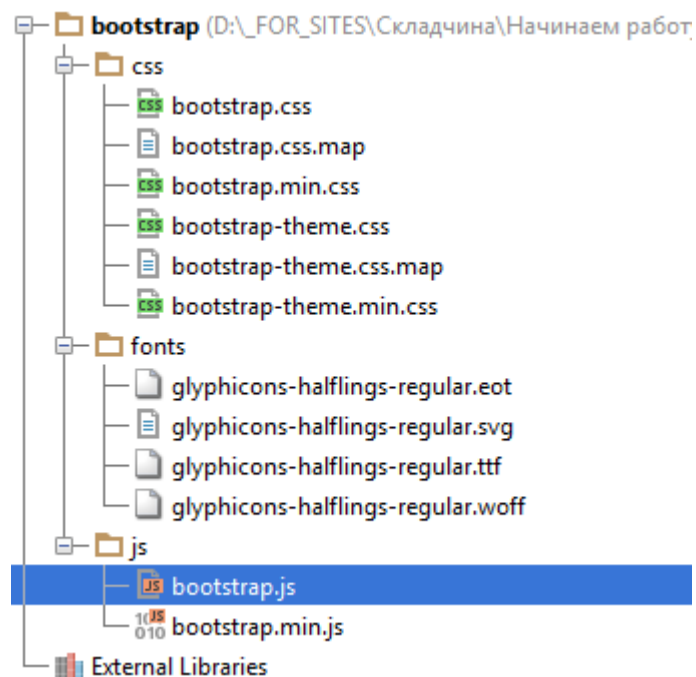


Рис. 4. Приклад структури файлів Bootstrap [11]

### 1.1.6 Переваги фреймворка Bootstrap

Використання фреймворків і бібліотек значно полегшує роботу розробникам і дозволяє розробляти проекти швидше. Переваги Bootstrap у порівнянні з другими фреймворками:

- **Адаптивність:**



Bootstrap дозволяє створювати адаптивні сайти. Дизайн сайту буде коректно відображатися на екранах пристроїв різних розмірів незалежно від їх діагоналі.

- **Кросбраузерність:**

Сайти, зроблені з використанням Bootstrap, будуть однаково відображатися у всіх сучасних браузерах.

- **Легкість у використанні і швидкість в освоєнні:**

Bootstrap легко використовувати в розробці, в ньому легко розібратися.

А ті, хто ще не знайомий з Bootstrap, зможуть швидко навчитися працювати з цим інструментом. І справа не тільки в його простоті, але і у великій кількості уроків та інструкцій, які можна знайти на просторах мережі.

- **Зрозумілий код:**

Bootstrap дозволяє писати якісний і зрозумілий код, який легко зрозуміє інший розробник. Це значно спрощує розробку в команді.

- **Єдність стилів:**

Елементи Bootstrap виглядають гармонійно між собою і дозволяють створювати сторінки і сайти в єдиному стилі.

Таким чином - Bootstrap дозволяє швидко створити сайт з стандартних блоків. У цьому полягає і його перевага, і його недолік - можна швидко отримати якісний сайт, при цьому втративши в оригінальності. Втім, якщо розробник добре знає CSS і Bootstrap, то він зможе модифікувати стандартні блоки і зробити дизайн відрізняється від інших сайтів.

Bootstrap відмінно підходить для маленьких проектів, де потрібно швидко зібрати верстку, але в цілому він використовується на сайтах різного масштабу

## 1.2 ВИБІР ЗАСОБУ РОЗРОБКИ ДЛЯ BACKEND'У

На поточний момент існує багато різних засобів для створення сайту. Можливо розробити сайт з нуля, використовувати CMS системи, такі як WordPress, Joomla, Drupal, vBulletin та інші, або використовуючи бекенд-фреймворки, такі як NodeJS, Django, Laravel і інші.

У даній дипломній роботі потрібно розробити інформаційний сайт з маленьким набором функцій, безпечний та швидкий в розробці і масштабованості для подальших змін і внесення нових функцій. Таким чином ми зупинимося на веб-фреймворку для бекенд розробки - Django.

### 1.2.1 ПОНЯТТЯ BACKEND

На відміну від frontend, backend це програмне забезпечення яке, як правило, розташоване на серверній стороні багатокomпонентної веб системи. Backend відповідає за дані проекту, забезпечує доступ до них, модифікацію, та видає їх на клієнтську сторону. Backend ще також називають «двигуном» веб-сайту. [17]

### 1.2.1 ПОНЯТТЯ BACKEND-ФРЕЙМВОРКУ

Фреймворк (англ. Framework, каркас, платформа, структура, інфраструктура) — інфраструктура програмних рішень, що полегшує розробку складних систем. [16] Спрощено дану інфраструктуру можна вважати своєрідною комплексною бібліотекою, але при цьому вона має ряд обмежень, що задають правила створення структури проекту та написання коду.

Фреймворки стали популярні з появою графічних інтерфейсів користувача. З їх використанням стало набагато простіше створювати засоби для створення веб-сервісів, так як архітектура реалізації коду програми стала відома заздалегідь. Для забезпечення роботи фреймворків використовуються техніки ООП (об'єктно-орієнтованого програмування).

Фреймворк відрізняється від бібліотеки тим, що бібліотека може бути використана як спів програми і вона не накладає ніяких обмежень тим самим не впливаючи на архітектуру програми. У той час як фреймворк встановлює правила написання коду, визначає архітектуру програми, заданої на початковому етапі розробки за замовчуванням. Так само фреймворк може включати в себе різні допоміжні для роботи бібліотеки.

### 1.2.3 ОПИС ФРЕЙМВОРКА DJANGO



Django (Джанго рис.5) — високорівневий відкритий Python-фреймворк (програмний каркас) для розробки веб-систем. [12] Названо його було на честь джазмена Джанго Рейнхардта (відповідно до музичних смаків одного зі засновників проекту).

Рис. 5 Логотип Django

Початкова розробка Django, як засобу для роботи новинних ресурсів, досить сильно позначилася на його архітектурі: він надає ряд засобів, які допомагають у швидкій розробці веб-сайтів інформаційного характеру. Так, наприклад, розробнику не потрібно створювати контролери та сторінки для адміністративної частини сайту, в Django є вбудований модуль для керування вмістом, який можна включити в будь-який сайт, зроблений на Django, і який може керувати відразу декількома сайтами на одному сервері. Адміністративний модуль дозволяє створювати, змінювати і вилучати будь-які об'єкти наповнення сайту, протоколюючи всі дії, а також надає інтерфейс для управління користувачами і групами (з призначенням прав).

Основні інструменти Django:

- **Форми:** HTML-форми використовуються для збору даних користувача для обробки на сервері. Django спрощує створення, перевірку та обробку форми.

- **Аутентифікація користувача і дозволу:** Django включає надійну систему аутентифікації і авторизації користувачів, яка була побудована з урахуванням безпеки.

- **Кешування:** Створення динамічного контенту набагато більш інтенсивною (і повільніше), ніж обслуговування статичного вмісту. Django забезпечує гнучке кешування, щоб зберігати всю або частину інформації, що відображається сторінки, для того, щоб вона не викликала повторно, за винятком випадків, коли це необхідно.

- **Адмін:** Адміністративна панель в Django включена за замовчуванням при створенні програми з використанням основного каркасу. Це спрощує управління адмінкою адміністраторам сайту для створення, редагування та перегляду будь-яких даних на сайті.

- **Серіалізація даних (перетворення в послідовну форму):** Django спрощує серіалізацію і обслуговування даних у таких форматах як XML або JSON. Це може бути корисно при створенні веб-сервісів (веб-сайтів, які виключно служать для використання даних іншими додатками або сайтами і самі нічого не відображають) або при створенні веб-сайту, на якому клієнтський код обробляє весь рендерінг даних.

Файлова структура Django на момент створення проекту приведена на рис. 6:

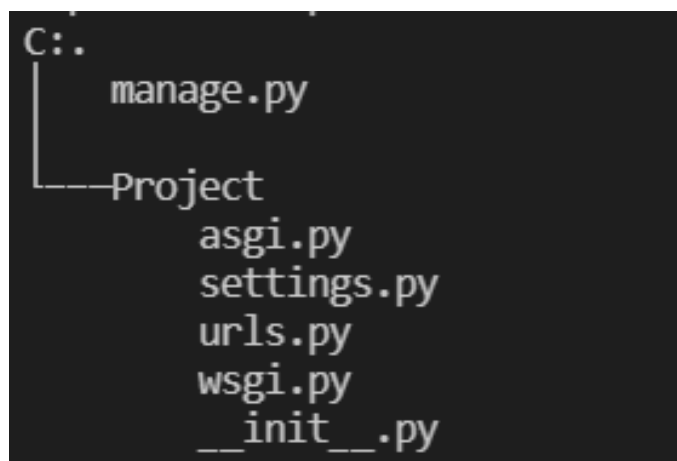


Рис. 6 Файлова структура проекту

Це основні виконавчі файли Django, в яких зберігаються налаштування програми, настройки WSGI і ASGI сумісних серверів, файл для основної url-адресації і головний файл для запуску сервера Django.

## 1.3 ВИБІР СУБД

### 1.3.1 ПОНЯТТЯ СУБД

Система управління базами данихх (СУБД, СКБД англ. Database Management System, DBMS) — набір взаємопов'язаних даних (база даних) і програм для доступу до цих даних [13]. Надає можливості створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до даних.

### 1.3.2 ВИБІР СУБД. MYSQL. MongoDB. PostgreSQL

Вибір системи управління баз даних це важлива частина при розробці архітектури додатку. Система управління баз даних повинна відповідати вимогам додатка до швидкості, безпеки зберігання, доступу по мережі інтернет, набору певних функцій, фінансових витрат на розробку і придбання необхідного обладнання та програмного забезпечення, а так само на навчання персоналу.

Розглянемо популярні СУБД для розробки додатків на Django:

1) **СУБД MySQL** — вільна система керування реляційними базами даних, яка була розроблена компанією «ТсХ» для підвищення швидкодії обробки великих баз даних [14]. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все

розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

2) СУБД **MongoDB** — документо-орієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів [15].

3) СУБД **PostgreSQL** (вимовляється «Пост-грес-К'ю-ель», або «постгрес») — об'єктно-реляційна система керування базами даних (СКБД). Є альтернативою як комерційним СКБД (Oracle Database, Microsoft SQL Server, IBM DB2 та інші), так і СКБД з відкритим кодом (MySQL, Firebird, SQLite).

Порівняно з іншими проєктами з відкритим кодом, такими як Apache, FreeBSD або MySQL, PostgreSQL не контролюється якоюсь однією компанією, її розробка можлива завдяки співпраці багатьох людей та компаній, які хочуть використовувати цю СКБД та впроваджувати у неї найновіші досягнення.

Для тестування програми буде використана локальна база даних, яка стоїть за умовчанням в настройках Django - SQLite3.

Для деплоя на сервер буде використана СУБД PostgreSQL, тому що вона відповідає всім вимогам додатка, а так само це популярне рішення серед розробників на Django

## 1.4 ВИБІР ХОСТИНГУ

Так як даний дипломний проєкт не фінансується, буде використаний безкоштовний хостинг - Heroku



Heroku (рис. 7) — хмарна PaaS-платформа, що підтримує ряд мов програмування. Компанією Heroku володіє Salesforce.com [18]. Heroku, одна з перших хмарних платформ, з'явилась в червні 2007 року і спочатку підтримувала тільки мову програмування Ruby, але зараз список підтримуваних мов також містить в

Рис. 7 Логотип Heroku

собі Java, Node.js, Scala, Clojure, Python і PHP. На серверах Heroku використовуються операційні системи Debian або Ubuntu (яка також заснована на Debian).

Heroku має ряд корисних функцій, такі як:

- 1) Deployment method - деплой з систем управління версіями (GIT). Є підтримка GitHub.
- 2) Automatic deploys - автоматичне розгортання певної обраної гілки сховища GitHub.
- 3) Має можливість установки власних доменів і SSL сертифікатів.
- 4) 500 мегабайт безкоштовного місця на хостингу.
- 5) Підтримка WSGI сумісних серверів.
- 6) Зберігання логів і помилок.
- 7) Зберігання та захист СУБД.

Посилання на сайт - <https://vladent.herokuapp.com/>

## 1.5 ВИБІР СИСТЕМИ УПРАВЛІННЯ ВЕРСІЯМИ

### 1.5.1 ПОНЯТТЯ СИСТЕМИ УПРАВЛІННЯ ВЕРСІЯМИ GIT

Git — розподілена система керування версіями файлів та спільної роботи. Проєкт створив Лінус Торвальдс для керування розробкою ядра Linux, а сьогодні підтримується Джуніо Хамано (англ. Junio C. Hamano) [19].

Git є однією з найефективніших, надійних і високопродуктивних систем керування версіями, що надає гнучкі засоби нелінійної розробки, що базуються на відгалуженні і злитті гілок. Для забезпечення цілісності історії та стійкості до змін заднім числом використовуються криптографічні методи, також можлива прив'язка цифрових підписів розробників до тегів і комітів.

### 1.5.2 ВИБІР СИСТЕМИ УПРАВЛІННЯ ВЕРСІЯМИ GITHUB.



GitHub (рис. 8) — один з найбільших веб-сервісів для спільної розробки програмного забезпечення. Існують безкоштовні та платні тарифні плани користування сайтом. Базується на системі керування версіями Git і розроблений на Ruby on Rails і Erlang компанією GitHub, Inc (раніше Logical Awesome).

Рис. 8 Логотип GitHub

Сервіс безкоштовний для проєктів з відкритим вихідним кодом, з наданням користувачам усіх своїх можливостей (включаючи SSL), а для окремих індивідуальних проєктів пропонуються різні платні тарифні плани.



## **РОЗДІЛ 2.**

### **ПРОЕКТНІ РІШЕННЯ**

#### **2.1 ТЕХНІЧНЕ ЗАВДАННЯ.**

##### **2.1.1. НАЙМЕНУВАННЯ І ОБЛАСТЬ ЗАСТОСУВАННЯ**

Інформаційний веб сайт стоматологічної клініки VlaDent.

##### **2.1.2 ПІДСТАВИ ДЛЯ РОЗРОБКИ**

Завдання на виконання дипломного проекту за наказом ректора НТУ «Дніпровська політехніка» від 27.04.2021 № НАКАЗ 234-с

##### **2.1.3 ПРИЗНАЧЕННЯ РОЗРОБКИ**

Веб-сайт призначений для підвищення ефективності ведення бізнесу підприємства

##### **2.1.4 ВИМОГИ ДО ФУНКЦІОНАЛЬНИХ ХАРАКТЕРИСТИК**

Функціональні характеристики веб-сайту повинні забезпечувати:

- Адміністративна панель для управління сайтом
- Можливість додавання, зміни і видалення тарифів послуг

підприємства

- Можливість додавання, зміни і видалення працівників підприємства
- Можливість додавання, зміни і видалення статей і їх банерів
- Можливість додавання, зміни і видалення загальної інформації про

підприємство

- Можливість підписки на розсилку підприємства
- Можливість онлайн запису на прийом / консультацію

##### **2.1.5. ВИМОГИ ДО НАДІЙНОСТІ**

Надійність і безпека сайту залежить від:

- Справної роботи сервера
- Справної роботи операційної системи
- Наявності необхідного програмного забезпечення
- Справної роботи інтернету

#### **2.1.6. ВИМОГИ ДО СКЛАДУ І ПАРАМЕТРІВ ТЕХНІЧНИХ ЗАСОБІВ**

Для функціонування сервера потрібно:

- процесор з частотою 1 ГГц і вище
- ОЗУ не менше 1024 МБ
- не менше 200 МБ вільного місця на жорсткому диску

#### **2.1.7. ВИМОГИ ДО ІНФОРМАЦІЙНОЇ ТА ПРОГРАМНОЇ СУМІСНОСТІ**

Програма призначена для роботи на Windows і Unix / Linux сумісних системах.

#### **2.1.8. ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ**

Для розробленого сайту повинна бути представлена така технічна документація:

- таблиці атрибутивної інформації
- текст скриптів
- опис проекту

## РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 СТВОРЕННЯ СХОВИЩА В GITHUB

#### 3.1.1 РОЗПОДІЛ РОЛЕЙ

Приклад розподілу ролей наведений на рис. 9.

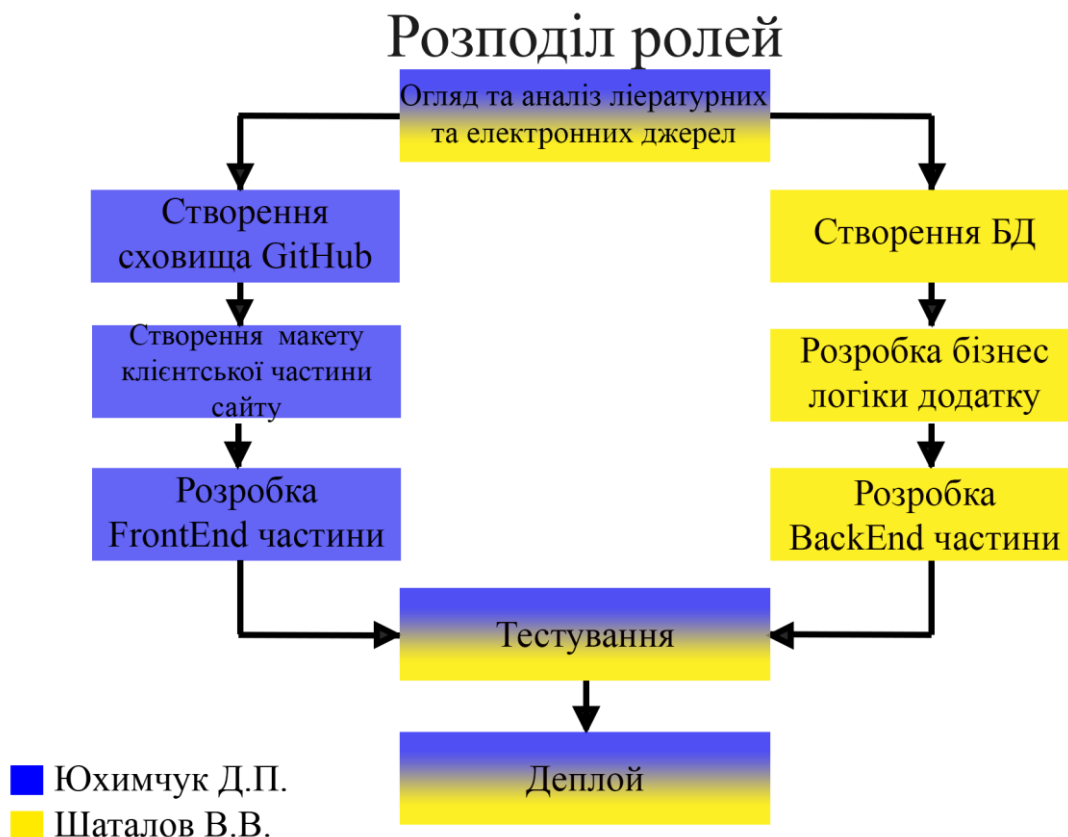


Рис. 9 Розподіл ролей.

#### 3.1.2 GITFLOW

**Gitflow** - це робочий процес, що допомагає вести безперервну розробку програмного забезпечення і впроваджувати методики DevOps. Ця модель була вперше опублікована і популяризована Вінсентом Дріссеном, відомим під псевдонімом nvie. Gitflow передбачає вибудовування суворої моделі розгалуження з урахуванням випуску проекту. Така модель забезпечує надійну основу для управління великими проектами.

Gitflow ідеально підходить для проектів, в яких є запланований цикл релізів і потрібна реалізація методики DevOps, відомої як безперервна поставка. У цьому робочому процесі використовуються поняття і команди, які були запропоновані в рамках робочого процесу з функціональними гілками. Однак Gitflow привносить нові специфічні ролі для різних гілок і визначає характер і частоту взаємодії між ними. Крім функціональних гілок в рамках цього робочого процесу використовуються окремі гілки для підготовки, підтримки та реєстрації релізів. При цьому ви як і раніше можете користуватися перевагами робочого процесу з функціональними гілками, такими як запити pull, ізольовані експерименти і ефективне командну взаємодію. [20]

Схема ведення сховища на GitHub для цього проекту наведена на рис. 10.

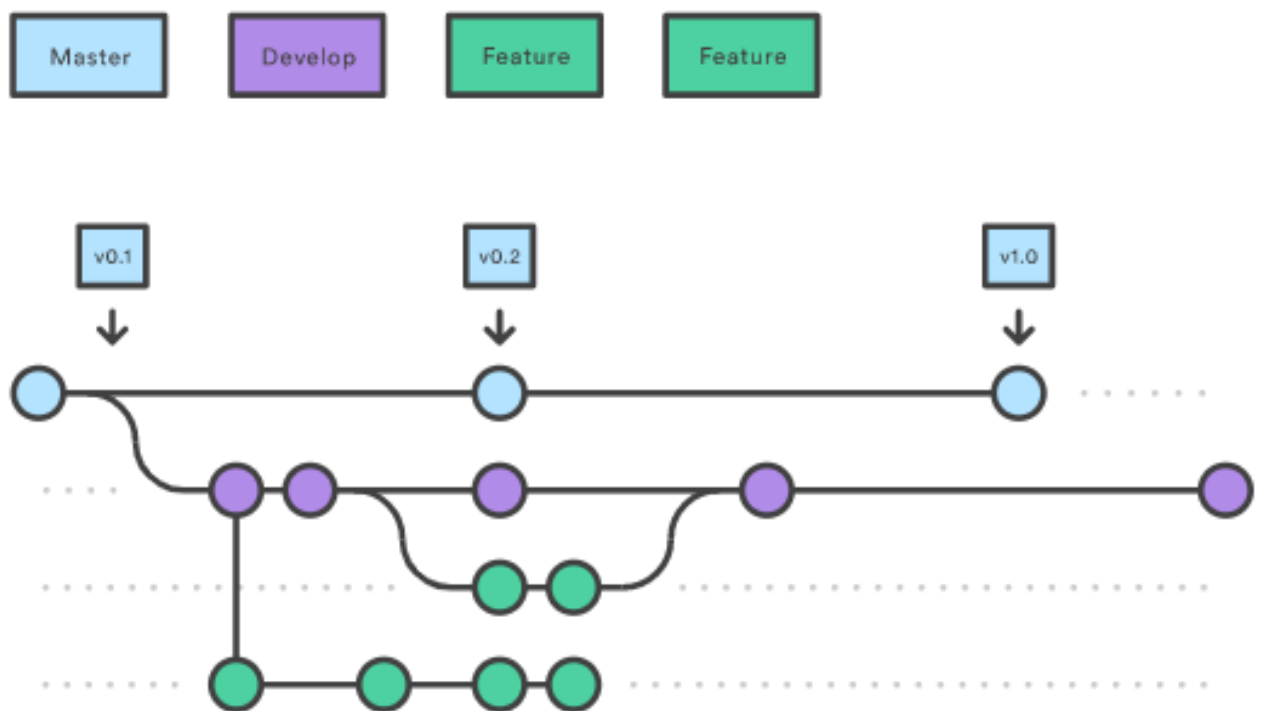


Рис. 10 GitFlow на сховищі GitHub

Гілка 'master' - це гілка для деплоя на бойовий сервер. Гілка 'master' пов'язана з хостингом Heroku.

Гілка 'develop' - це гілка використовується для основної розробки.

Гілка 'feature' - це гілка використовується для розробки нових функцій. Після тестування вона зливається з гілкою 'develop'.

При написанні нової функції розробник повинен створити для своєї роботи нову гілку з гілки 'develop' і назвати її відповідно назві завдання. Після тестування розробником нової функції, йде запит на злиття в гілку 'develop'. Якщо всі інші розробники перевірили код і підтвердили злиття, гілка з новою функцією переходить в гілку 'develop'.

Коли має сенс переходити на наступну версію продукту, гілка 'develop' зливається в гілку 'master', а сервер автоматично завантажує і встановлює оновлення.

Таким чином, розробка ведеться в окремих від деплоя гілках, це дозволяє нам працювати разом над одним проектом, створювати нові функції і тестувати їх безпечно, не заважаючи іншим розробникам.

## 3.2 СТВОРЕННЯ ТАБЛИЦЬ В БД

### 3.2.1 ПРИКЛАД СТВОРЕННЯ ТАБЛИЦІ В БД

**Django ORM** (Object Relational Mapping) є однією з найпотужніших особливостей Django. Це дозволяє нам взаємодіяти з базою даних, використовуючи код Python, а не SQL. [21]

Приклад створення таблиці докторів наведений на рис. 11.

```

class Medic(models.Model):
    surname = models.CharField(verbose_name="Прізвище", max_length=25)
    name = models.CharField(verbose_name="Ім'я", max_length=25)
    patronymic = models.CharField(verbose_name="По батькові", max_length=25, blank=True)
    # photo = models.ImageField(verbose_name='Фото', blank=True)
    photo = models.CharField(verbose_name='Фото', blank=True, max_length=300)
    # about_me = models.TextField(verbose_name="Про мене", blank=True)
    about_me = models.CharField(verbose_name="Про мене", blank=True, max_length=360)
    specialization = models.CharField(verbose_name='Спеціалізація', blank=True, max_length=25)
    facebook = models.CharField(verbose_name='Фейсбук', blank=True, max_length=100)
    instagram = models.CharField(verbose_name='Інстаграм', blank=True, max_length=100)

    def __str__(self):
        | return self.name

    class Meta():
        | verbose_name = "Персонал"
        | verbose_name_plural = "Персонал"
        | ordering = ('id',)

```

Рис. 11 Приклад створення таблиці докторів

Приклад створення запиту на всіх докторів з таблиці наведений на рис. 12.

```
medics = Medic.objects.all()
```

Рис. 12 Приклад створення запиту на всіх докторів з таблиці

### 3.2.2 СХЕМА БД

Схема БД у проєкті наведена на рис. 13.

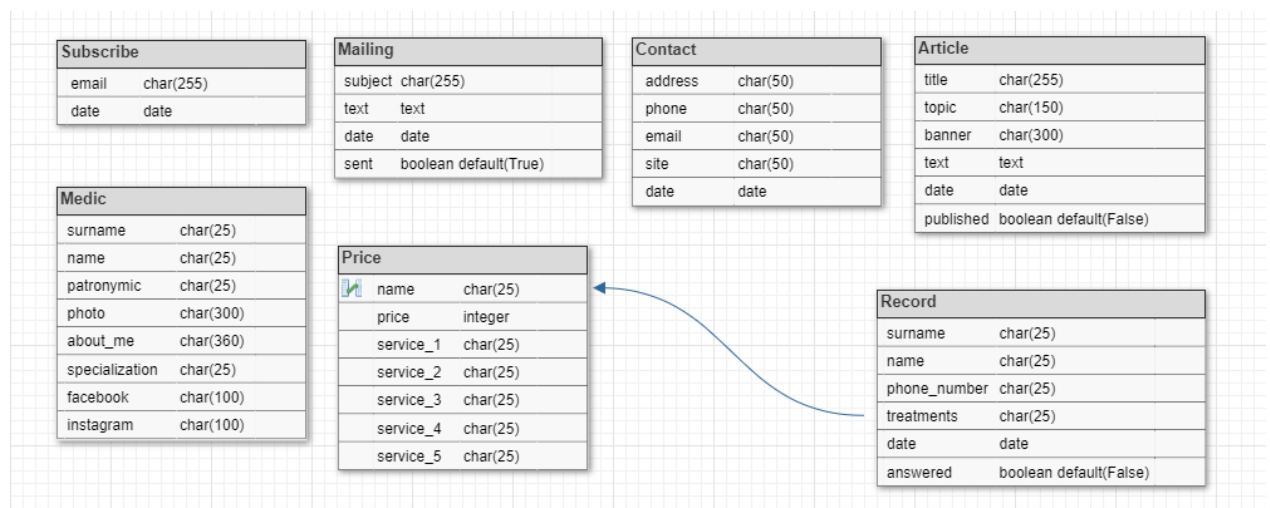


Рис. 13 Схема БД

### 3.2.3 СПИСОК ТАБЛИЦЬ

Таблиця 3.2.3 - 1

#### Список таблиць БД

№	Ім'я	Опис
1	Subscribe	Зберігає в собі список користувачів, які підписалися на розсилки
2	Mailing	Зберігає в собі розсилки по групі користувачів з таблиці Subscribe
3	Contact	Зберігає в собі контактну інформацію стоматологічної клініки
4	Article	Зберігає в собі статті блогу
5	Medic	Зберігає в собі список персоналу (лікарів)
6	Price	Зберігає в собі список послуг, їх опису та цін
7	Record	Зберігає в собі список користувачів, які замовили послуги

Таблиця 3.2.3 - 2

#### Subscribe

№	Ім'я поля	Тип	Розмір	Ключ	Примітка
1	email	Char	255	-	Email передплатника
2	date	Date	-	-	Дата підписки

```
class Subscribe(models.Model):
    email = models.EmailField(verbose_name='Маїл', max_length=255)
    date = models.DateTimeField(verbose_name='Дата створення', auto_now_add=True)

    def __str__(self):
        return self.email

    class Meta():
        verbose_name = 'Підписка'
        verbose_name_plural = 'Підписки'
        ordering = ("-date",)
```

Рис. 14 Створення таблиці Subscribe

Таблиця 3.2.3 - 3

Mailing

№	Ім'я поля	Тип	Розмір	Ключ	Примітка
1	subject	Char	255	-	Тема
2	text	Text	-	-	Повідомлення
3	date	Date	-	-	Дата відправки
4	sent	Bool	-	-	Відправлено

```
class Mailing(models.Model):
    subject = models.CharField(verbose_name='Тема', max_length=255)
    text = models.TextField(verbose_name='Повідомлення')
    date = models.DateField(verbose_name='Дата відправки', auto_now_add=True)
    sent = models.BooleanField(verbose_name='Відправлено', default=True)

    def __str__(self):
        return self.subject

    class Meta():
        verbose_name = 'Розсилку'
        verbose_name_plural = 'Розсилки'
        ordering = ("-date", "-id")
```

Рис. 15 Створення таблиці Mailing

Таблиця 3.2.3 - 4

Contact

№	Ім'я поля	Тип	Розмір	Ключ	Примітка
1	address	Char	50	-	Адреса
2	phone	Char	50	-	Телефон
3	email	Char	50	-	Електронна пошта
4	site	Char	50	-	Веб-сайт
5	date	Date	-	-	Дата оновлення



```

class Contact(models.Model):
    address = models.CharField(verbose_name='Адреса', max_length=50)
    phone = models.CharField(verbose_name='Телефон', max_length=50)
    email = models.CharField(verbose_name='Електронна пошта', max_length=50)
    site = models.CharField(verbose_name='Веб-сайт', max_length=50)
    date = models.DateField(verbose_name='Дата оновлення', auto_now_add=True)

    def __str__(self):
        return self.address

    class Meta():
        verbose_name = 'Контакт'
        verbose_name_plural = 'Контакти'
        ordering = ("-date",)

```

Рис. 16 Створення таблиці Contact

Таблиця 3.2.3 - 5

Article

№	Ім'я поля	Тип	Розмір	Ключ	Примітка
1	title	Char	255	-	Назва
2	topic	Char	150	-	Тема
3	banner	Char	300	-	Баннер
4	text	Text	-	-	Текст
5	date	Date	-	-	Дата оновлення
6	published	Bool	-	-	Опубліковано

```

class Article(models.Model):
    title = models.CharField(verbose_name='Назва', max_length=255)
    topic = models.CharField(verbose_name='Тема', max_length=150)
    # banner = models.ImageField(verbose_name='Баннер', blank=True, upload_to="blog/")
    banner = models.CharField(verbose_name='Баннер', blank=True, max_length=300)
    text = RichTextUploadingField(verbose_name='Текст')
    date = models.DateField(verbose_name='Дата створення', auto_now=True)
    published = models.BooleanField(verbose_name='Опубліковано', default=False)

    def __str__(self):
        | return self.title

    class Meta():
        | verbose_name = 'Статтю'
        | verbose_name_plural = 'Статті'
        | ordering = ("-date", "-id")

```

Рис. 17 Створення таблиці Article

Таблиця 3.2.3 - 6

Medic

№	Ім'я поля	Тип	Розмір	Ключ	Примітка
1	surname	Char	25	-	Прізвище
2	name	Char	25	-	Ім'я
3	patronymic	Char	25	-	По батькові
4	photo	Char	300	-	Фото
5	about_me	Char	360	-	Про мене
6	specialization	Char	25	-	Спеціалізація
7	facebook	Char	100	-	Фейсбук
8	instagram	Char	100	-	Інстаграм

```

class Medic(models.Model):
    surname = models.CharField(verbose_name="Прізвище", max_length=25)
    name = models.CharField(verbose_name="Ім'я", max_length=25)
    patronymic = models.CharField(verbose_name="По батькові", max_length=25, blank=True)
    # photo = models.ImageField(verbose_name='Фото', blank=True)
    photo = models.CharField(verbose_name='Фото', blank=True, max_length=300)
    # about_me = models.TextField(verbose_name="Про мене", blank=True)
    about_me = models.CharField(verbose_name="Про мене", blank=True, max_length=360)
    specialization = models.CharField(verbose_name='Спеціалізація', blank=True, max_length=25)
    facebook = models.CharField(verbose_name='Фейсбук', blank=True, max_length=100)
    instagram = models.CharField(verbose_name='Інстаграм', blank=True, max_length=100)

    def __str__(self):
        return self.name

    class Meta():
        verbose_name = "Персонал"
        verbose_name_plural = "Персонал"
        ordering = ('id',)

```

Рис. 18 Створення таблиці Medic

Таблиця 3.2.3 - 7

Price

№	Ім'я поля	Тип	Розмір	Ключ	Примітка
1	name	Char	25	-	Назва
2	price	Integer	-	-	Ціна
3	service_1	Char	25	-	Перша послуга
4	service_2	Char	25	-	Друга послуга
5	service_3	Char	25	-	Третя послуга
6	service_4	Char	25	-	Четверта послуга
7	service_5	Char	25	-	П'ята послуга

```

class Price(models.Model):
    name = models.CharField(verbose_name='Назва', max_length=25, unique=True)
    price = models.PositiveIntegerField(verbose_name='Ціна')
    service_1 = models.CharField(verbose_name='Перша послуга', max_length=25, blank=True)
    service_2 = models.CharField(verbose_name='Друга послуга', max_length=25, blank=True)
    service_3 = models.CharField(verbose_name='Третя послуга', max_length=25, blank=True)
    service_4 = models.CharField(verbose_name='Четверта послуга', max_length=25, blank=True)
    service_5 = models.CharField(verbose_name="П'ята послуга", max_length=25, blank=True)

    def __str__(self):
        | return self.name

    class Meta():
        | verbose_name = "Ціна"
        | verbose_name_plural = "Ціни"
        | ordering = ('-price',)

```

Рис. 19 Створення таблиці Price

Таблиця 3.2.3 - 8

Record

№	Ім'я поля	Тип	Розмір	Ключ	Примітка
1	surname	Char	25	-	Прізвище
2	name	Char	25	-	Ім'я
3	phone_number	Char	25	-	Телефонний номер
4	treatments	FK	-	Price	Послуга
5	date	Date	-	-	Дата оновлення
6	answered	Bool	-	-	Відповіли?

```

class Record(models.Model):
    surname = models.CharField(verbose_name="Прізвище", max_length=25)
    name = models.CharField(verbose_name="Ім'я", max_length=25)
    phone_number = models.CharField(verbose_name="Телефонний номер", max_length=25)
    treatments = models.ForeignKey(Price, verbose_name='Послуга', blank=True, null=True, on_delete=models.SET_NULL)
    date = models.DateTimeField(verbose_name='Дата створення', auto_now_add=True)
    answered = models.BooleanField(verbose_name='Відповіли?', default=False)

    def __str__(self):
        | return self.name

    class Meta():
        | verbose_name = 'Запис'
        | verbose_name_plural = 'Записи'
        | ordering = ("answered", "date")

```

Рис. 20 Створення таблиці Record

### 3.3 АРХІТЕКТУРНА СХЕМА САЙТУ

#### 3.3.1 АЛГОРИТМ СТВОРЕННЯ БАЗИ ДАНИХ

Django надає власну бібліотеку для роботи з базою даних заснованої на ORM архітектурі. Ми можемо повністю відмовити від написання SQL коду на користь об'єктно-орієнтованого коду Python. Кожна таблиця є класом, а кожен елемент таблиці це атрибут класу.

Звернення до таблиць БД проводиться за допомогою створення об'єкта класу і виклику певного методу.

Приклад методів:

- `create()` - Створює поле в таблиці
- `save()` - Зберігає зміни
- `all()` - Повертає всі записи в таблиці
- `get()` - Повертає єдину запис в таблиці
- `filter(args*)` - Повертає відфільтровану за запитом інформацію
- `order_by(args*)` - Повертає відсортовану по запиту інформацію
- `select_related()` та `prefetch_related()` - Повертають всі записи в таблиці (для пов'язаних таблиць)
- та інші

#### 3.3.2 ВІДОБРАЖЕННЯ СТОРІНКИ

Коли надходить запит (Request), url-адресація перенаправляє його на потрібну функцію (View) за адресою. Функція (View) починає обробляти запит, звертається до бази даних і робить певну потрібну вибірку інформації, після за допомогою шаблонізатора Jinja2 вставляє інформацію з бази даних в сторінку (Templates) та повертає її у відповідь (Response) клієнту.

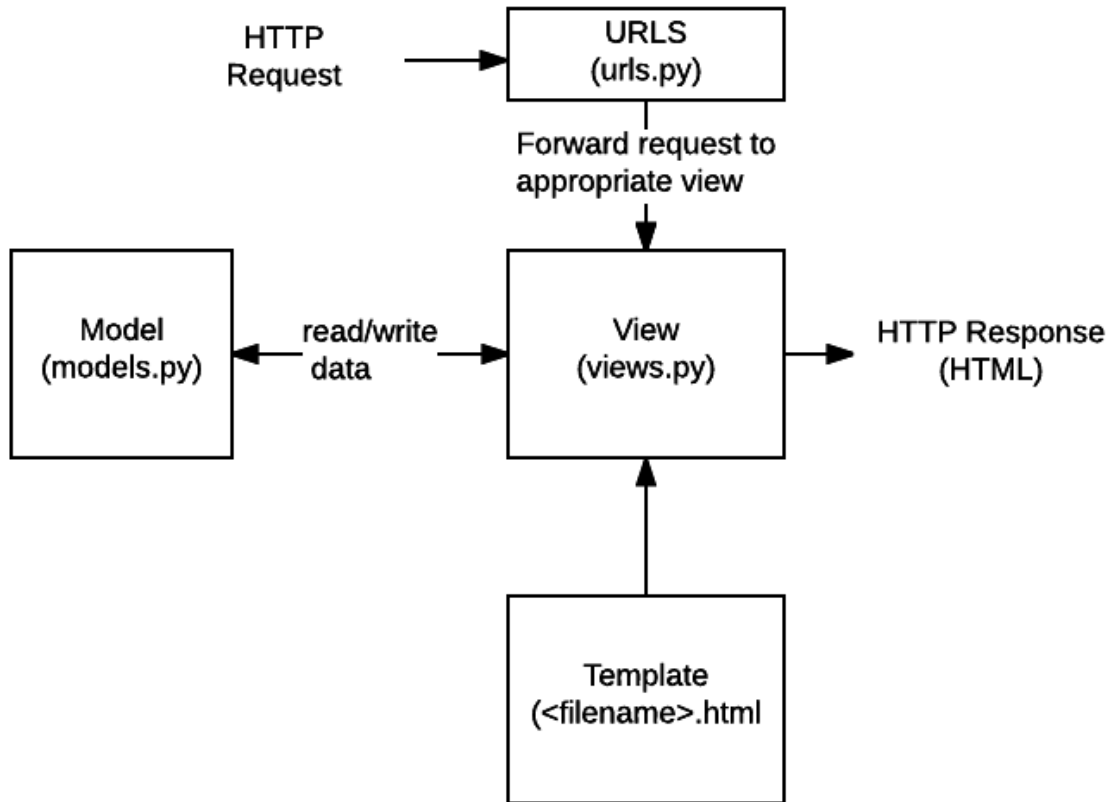


Рис. 21 Процес відображення сторінки

Приклад роботи функції відображення головної сторінки Main(View) наведений на рис. 22.

```

class Main(View):
    def get(self, request, alert=False, *args, **kwargs):
        articles = Article.objects.filter(published=True)[:3]
        medics = Medic.objects.all()[:4]
        prices = list(Price.objects.all()[:4])

        return render(request, 'main/index.html', {'index': True, 'alert': alert, 'contact': ContactView.get_contact(),
                                                    'articles': articles, 'medics': medics, 'prices': prices})

    def post(self, request, *args, **kwargs):
        if 'name' in request.POST:
            treatments = request.POST['service'] if 'service' in request.POST else 1
            Record.objects.create(name = request.POST['name'],
                                  surname = request.POST['surname'],
                                  phone_number = request.POST['phone_number'],
                                  treatments = Price.objects.get(pk = treatments))

        if 'send-mail' in request.POST:
            Subscribe.objects.create(email = request.POST['send-mail'])

        return self.get(request, alert = True)
  
```

Рис. 22 Приклад роботи функції відображення сторінки Main

Функція `get()` відповідає за GET-запити. При такому запиті, функція звертається до таблиці `Article` з фільтром `'published=True'` та робить вибірку з

останніх 3 записів. Потім функція звертається до таблиці Medic, бере усі записи та робить вибірку з останніх записів. Потім функція звертається до таблиці Price, де теж робить вибірку з останніх записів. Після запитів до БД, викликається функція яка буде рендерить сторінку, та їй передається: стан запиту, html-документ та уся зібрана інформація з БД.

Результат GET-запиту до головної сторінки сайту наведений на рис. 23:

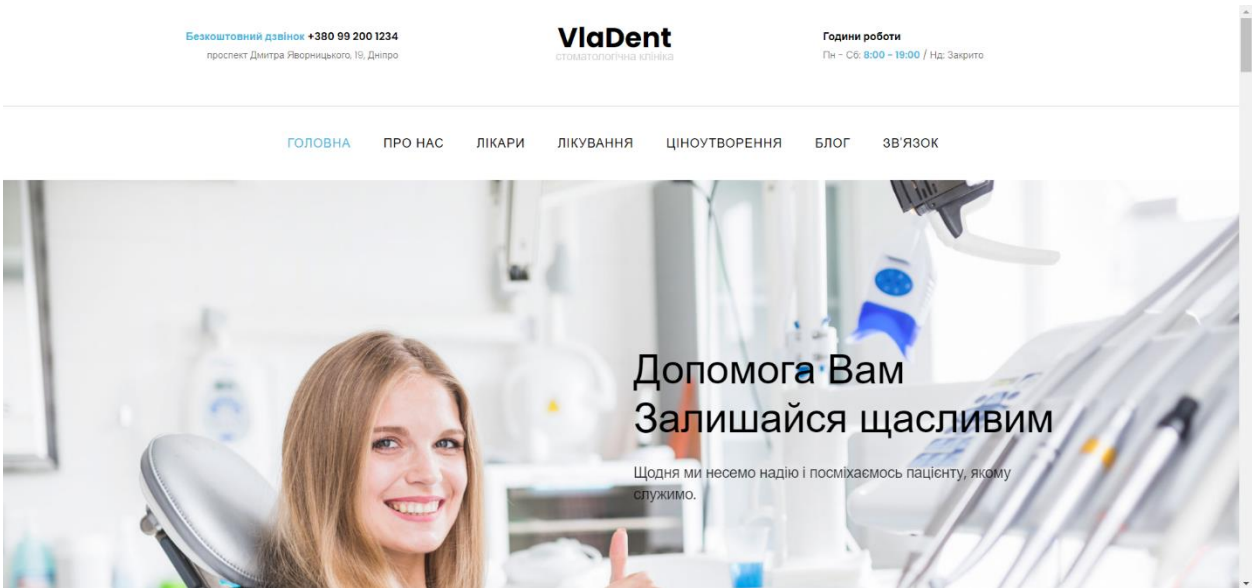


Рис. 23 Результат Get-запиту до головної сторінки сайту



Рис. 24 Результат Get-запиту до головної сторінки сайту (2)

### 3.3.3 ОПИС РОЗРОБЛЕНОГО САЙТУ І ЙОГО ІНТЕРФЕЙСУ

Приклад відображення головної сторінки наведений на рис. 25.

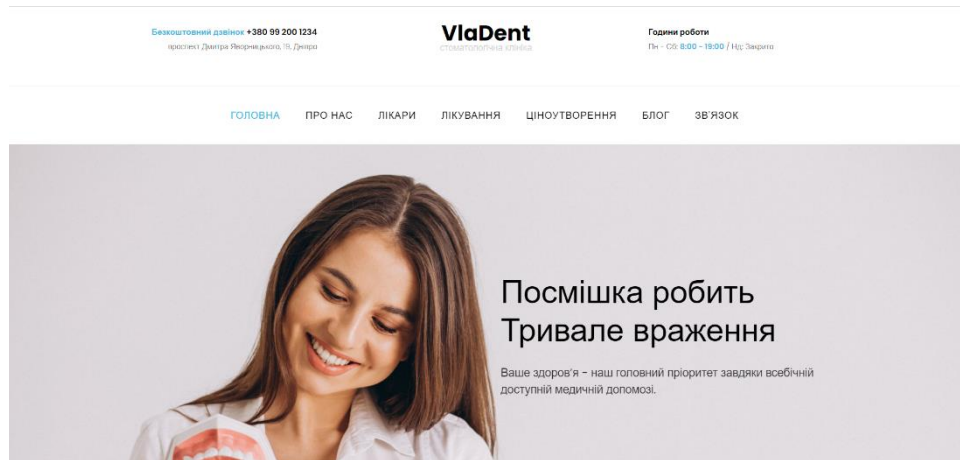


Рис. 25 Головна сторінка сайту

На головній сторінці відображається меню навігації котра знаходиться у верхній частині сайту по центру, контакти стоматології а також графік роботи.

Нижче знаходиться банер з тематикою стоматології а також інформація про клініку і її засновника. Приклад інформації клініки наведений на рис. 26.

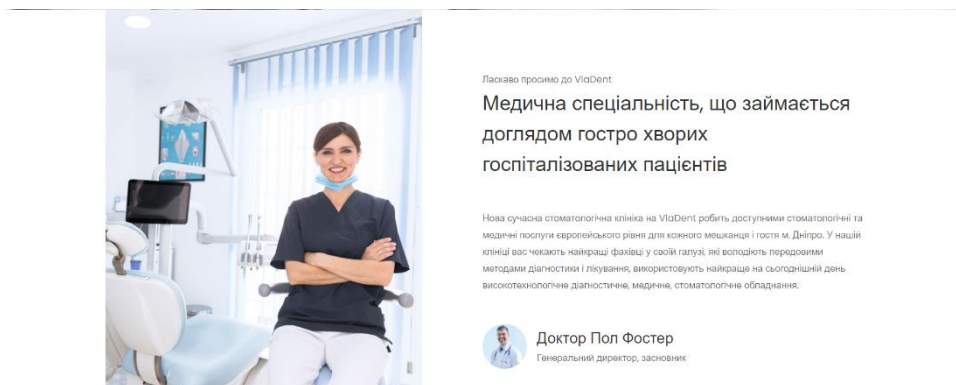


Рис. 26 Інформація про стоматологію

Далі розташовується банер із записом на безкоштовну консультацію. Приклад банеру з записом на консультацію наведений на рис.27.



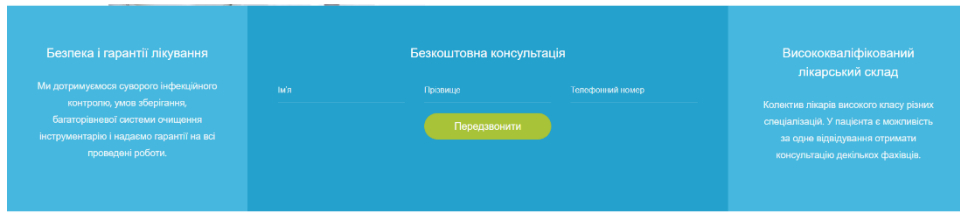


Рис. 27 Запис на консультацію

Також є блок з послугами та головними лікарями стоматології з посиланням на їх інстаграм та фейсбук. Приклад блоку з послугами та лікарями наведені на рис. 28, 29.

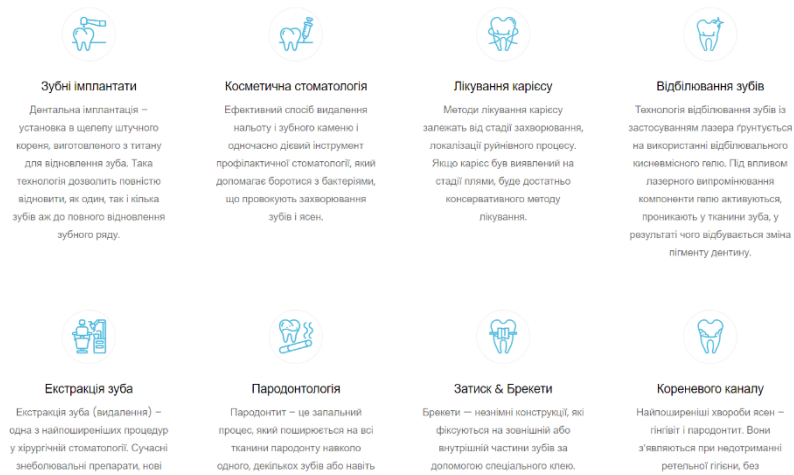


Рис. 28 Послуги стоматології

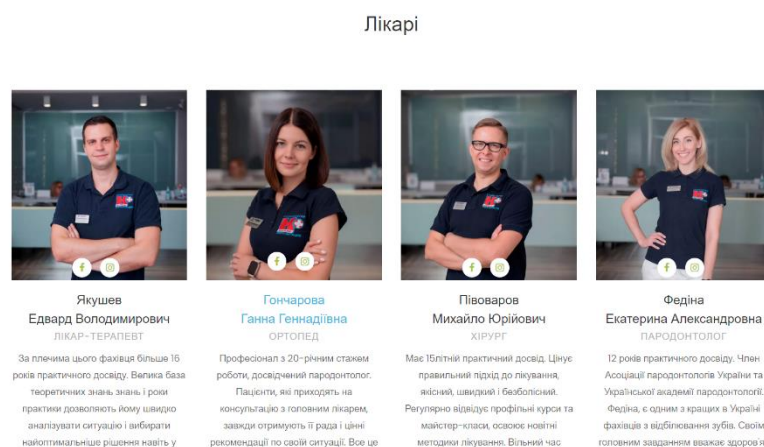


Рис. 29 Лікарі та їх опис

На головній сторінці виводяться карусель з відгуків клієнтів та ціноутворення стоматології. Приклад каруселі з відгуками та ціноутворенням наведені на рис. 30, 31.

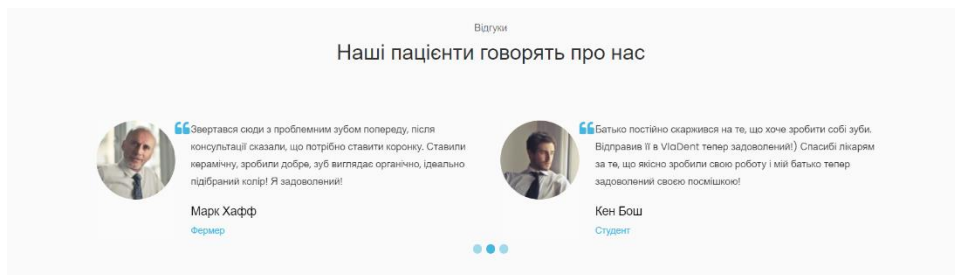


Рис. 30 Відгуки клієнтів

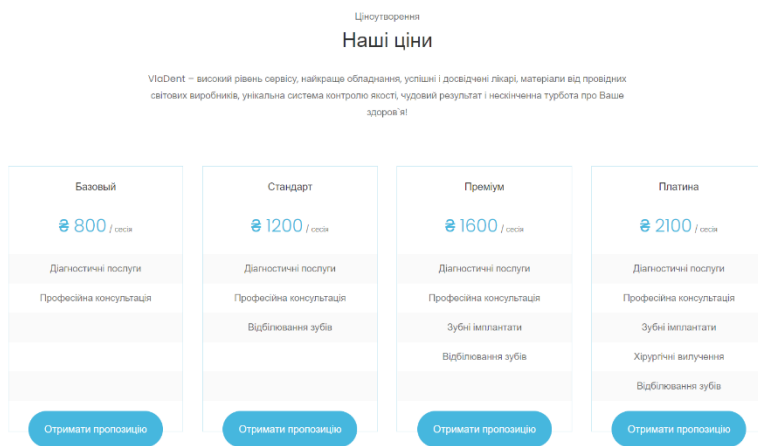


Рис. 31 Ціноутворення послуг

При натисканні на одну з послуг відкривається вікно із записом на прийом. Приклад вікна з записом на прийом наведений на рис. 32.

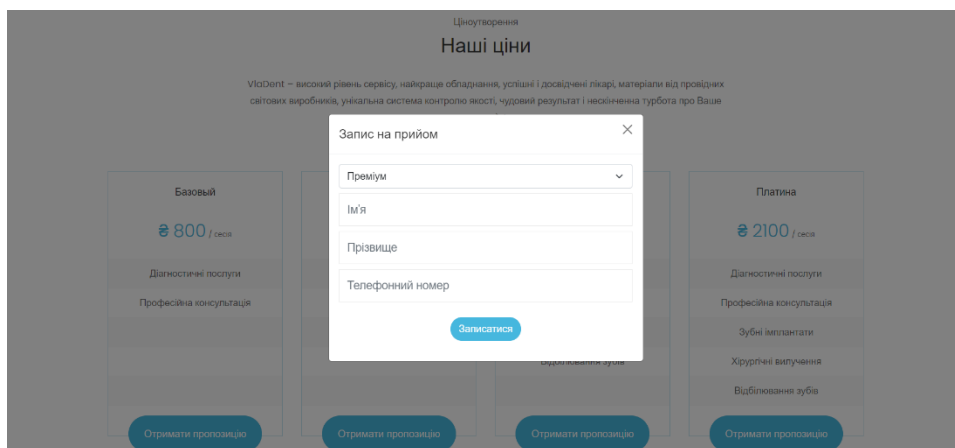


Рис. 32 Вікно з записом на прийом

Блок з останніми блогами клініки. Приклад з блогами клініки наведений на рис. 33.

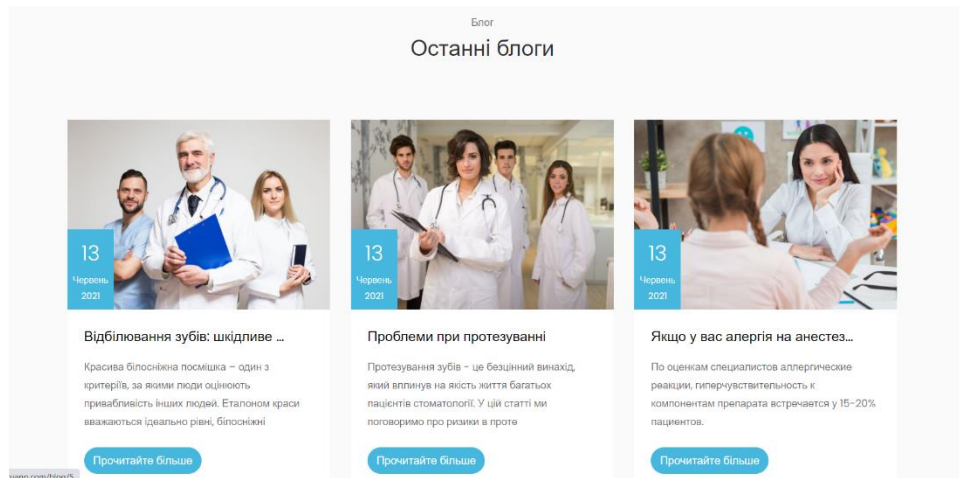


Рис. 33 Останні блоги

При натисканні на кнопку “Прочитати більше” нас перенаправляє на сторінку обраної статті. Приклад статті наведений на рис. 34.

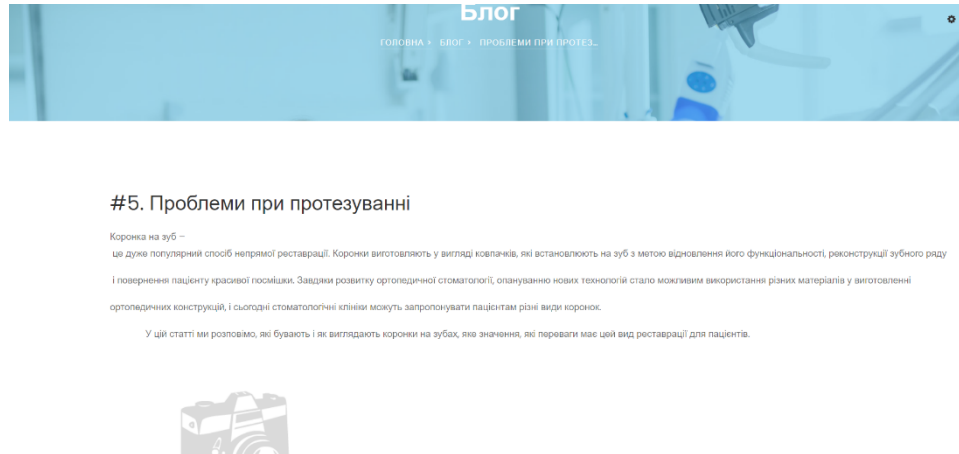


Рис. 34 Приклад статті

В самому кінці сайту розташовується футер в якому є контакти, пошта для зв'язку, адреса стоматології, час роботи і підписка на розсилку новин і спеціальних пропозицій. При натисканні на номер телефону відкриється вікно з заповненим номером телефону для здійснення дзвінків (якщо ви увійшли з телефону). Приклад футеру наведений на рис. 35.

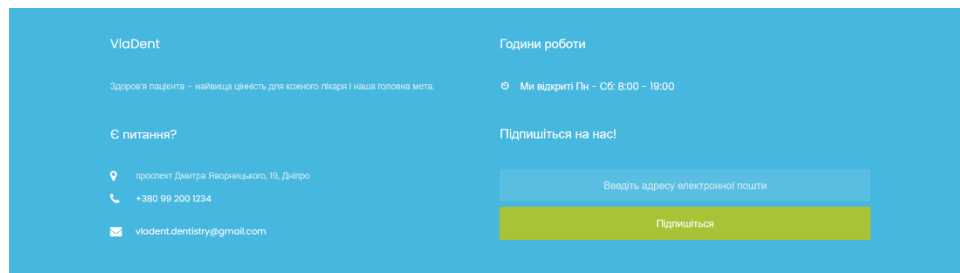


Рис. 35 Футер сайту з усієї інформацією

При переході на сторінку “Лікування” нам доступні всі послуги клініки та їх опис. Приклад усіх послуг та їх опис наведений на рис. 36.

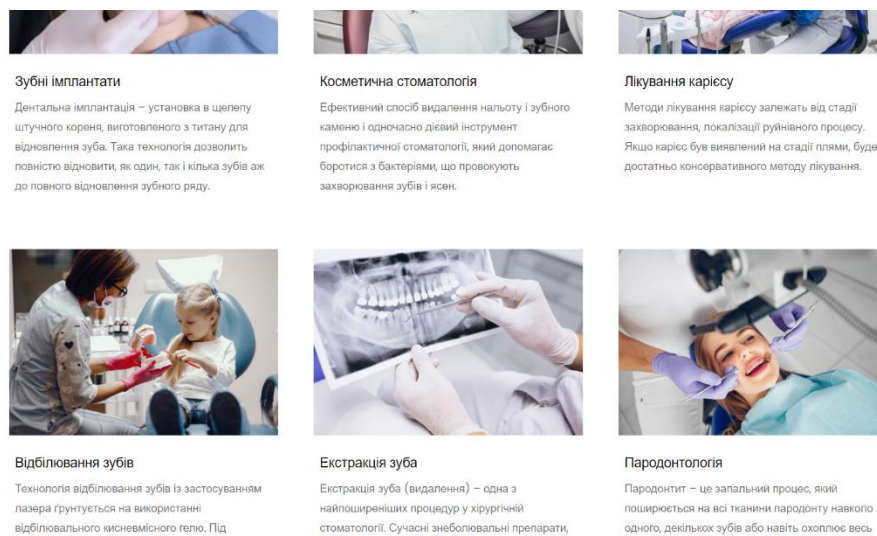


Рис. 36 Усі послуги клініки з описом

Сторінка “Зв’язок” відображає детальну інформацію з контактною інформацією та місцезнаходженням клініки. Приклад сторінки зв’язок наведений на рис. 37.

## Контактна інформація

Адреса: проспект Дмитра Яворницького, 19, Дніпро	Телефон: +380 99 200 1234	Електронна пошта: vladent.dentistry@gmail.com	Веб-сайт: vladenta.herokuapp.com
--	---------------------------	---	----------------------------------

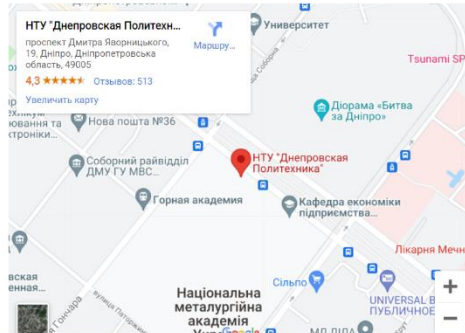


Рис. 37 Контактна інформація

Для управління інформацією на сайті, треба перейти в адміністративну панель за посиланням - <https://vladent.herokuapp.com/admin/> і увійти під своїм логіном і паролем.

## Grappelli

**Увійти**

**Ім'я користувача**

**Пароль**

Рис. 38 Вхід в адмін-панель

Після авторизації з'явиться список таблиць з бази даних наведених на рис. 39.

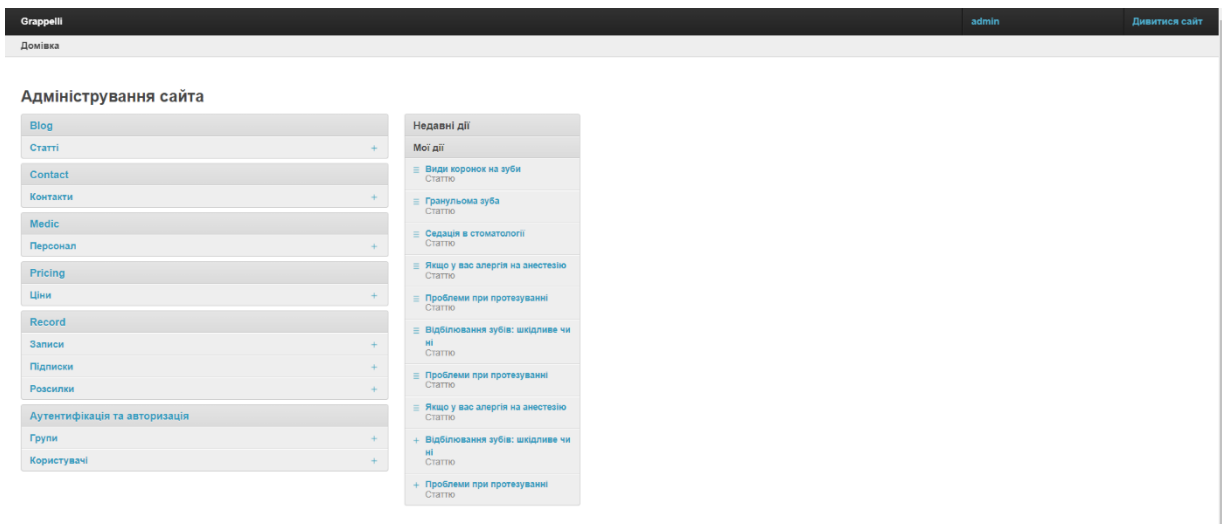


Рис. 39 Адмін-панель

Так само, після авторизації, адміністратору не треба буде постійно вводити /admin/ в адресний рядок для переходу в адмін-панель, з'явиться кнопка для швидко переходу на кожній сторінці сайту.

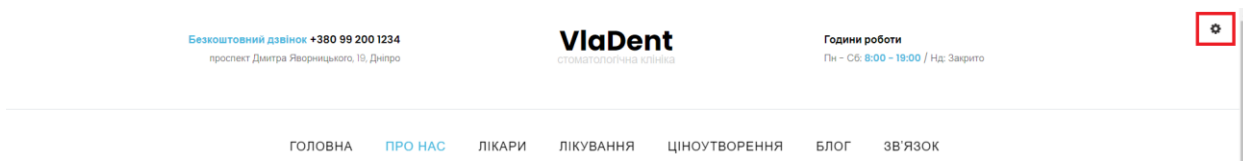


Рис. 40 Кнопка для швидкого переходу в адмін-панель

В адмін-панелі можна переглядати інформацію з БД.

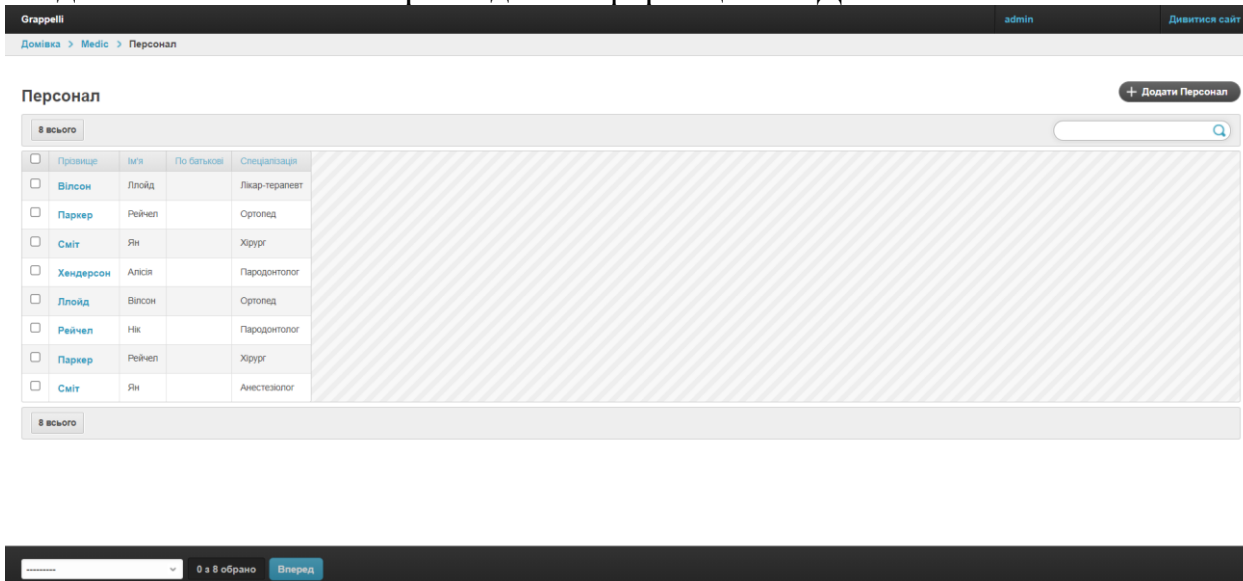


Рис. 41 Приклад інформації з таблиці "Персонал"

Так само можна додавати нову інформацію.

Додати Персонал

Прізвище	<input type="text"/>
Ім'я	<input type="text"/>
По батькові	<input type="text"/>
Фото	<input type="text"/>
Про мене	<input type="text"/>
Спеціалізація	<input type="text"/>
Фейсбук	<input type="text"/>
Інстаграм	<input type="text"/>

Зберегти і продовжити редагування

Зберегти і додати інше

Зберегти

Рис. 42 Приклад додавання нового співробітника

Змінити або видалити інформацію можна натиснувши на пункт в таблиці.

Змінити Персонал

Історія

Прізвище	Вілсон
Ім'я	Ллойд
По батькові	<input type="text"/>
Фото	<a href="https://cdn1.savepice.ru/uploads/2021/4/6/2b69ea9bed749af03bceb5d594ec54b9-full.jpg">https://cdn1.savepice.ru/uploads/2021/4/6/2b69ea9bed749af03bceb5d594ec54b9-full.jpg</a>
Про мене	За плечима цього фахівця більше 16 років практичного досвіду. Велика база теоретичних знань знань і роки практики дозво
Спеціалізація	Лікар-терапевт
Фейсбук	#
Інстаграм	#

Видалити

Зберегти і продовжити редагування

Зберегти і додати інше

Зберегти

Рис. 43 Приклад редагування інформації співробітника

Щоб зробити розсилку по користувачам треба перейти в розділ "Розсилання" і створити новий запис. При збереженні цієї записи, користувачам автоматично прийде повідомлення на пошту.

Grappelli

Домівка > Record > Розсилки > Додати Розсилку

### Додати Розсилку

Тема	Тестування розсилок
Повідомлення	Добрий день, ми проводимо тестування розсилок.

Рис. 44 Приклад додавання розсилки

Результат проведеної розсилки:

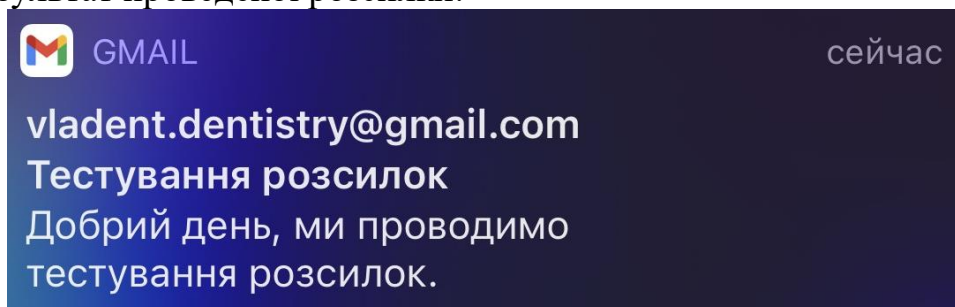


Рис. 45 Приклад результату проведеної розсилки

В адмін-панелі можна додавати адміністраторів сайту і надавати їм певні права та привілеї. Переходимо в розділ “Користувачі” та додаємо нового користувача. У цьому розділі можна додавати особисті інформацію, статус супер користувача, групу користувача і певні права на редагування БД.

Grappelli

admin

Домівка > Аутентифікація та авторизація > Користувачі > admin1

Дивитися сайт

### Змінити користувач

Історія

Ім'я користувача	admin1
Пароль	алгоритм: pbkdf2_sha256 ітерації: 216000 сім্ব: DCJOGa***** хеш: pxVSDa*****
Персональна інформація	
Ім'я	
Прізвище	
Емейл адреса	
Дозволи	
<input checked="" type="checkbox"/> Активний	Визначас, чи можна цього користувача звикати доним. Заберть палочку, замість видалення запіску користувача.
<input type="checkbox"/> Статус персоналу	Визначас, чи може користувач увійти до цього сайту адміністрування.
<input type="checkbox"/> Статус суперкористувача	Визначас, що цей користувач має всі дозволи без їх точного зазначення.
Групи	В наявності групи: Обрано групи +

Фільтр

Рис. 46 Приклад додавання нового користувача



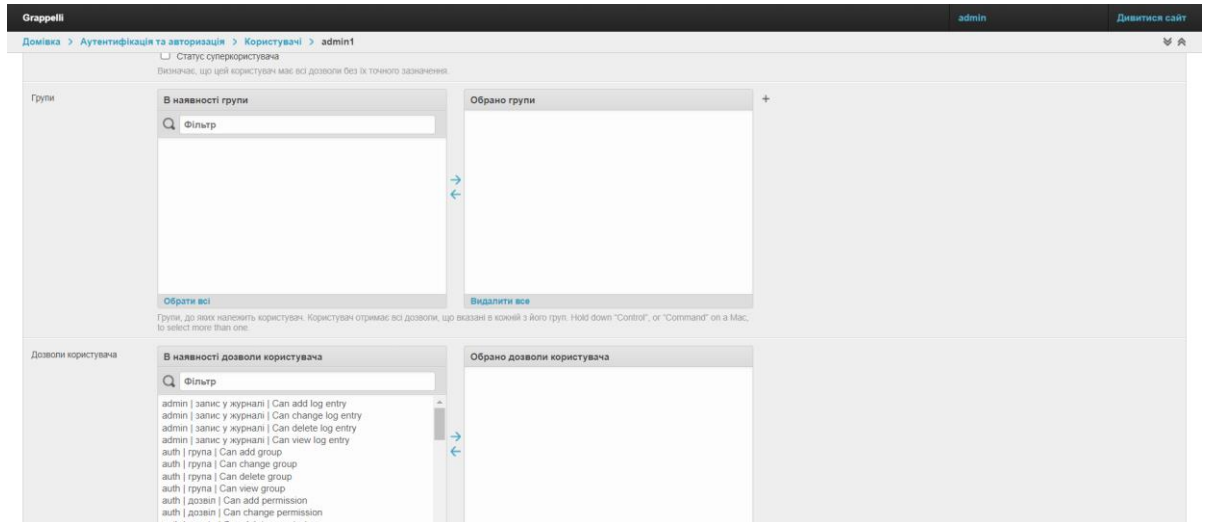


Рис. 47 Приклад редагування прав нового користувача

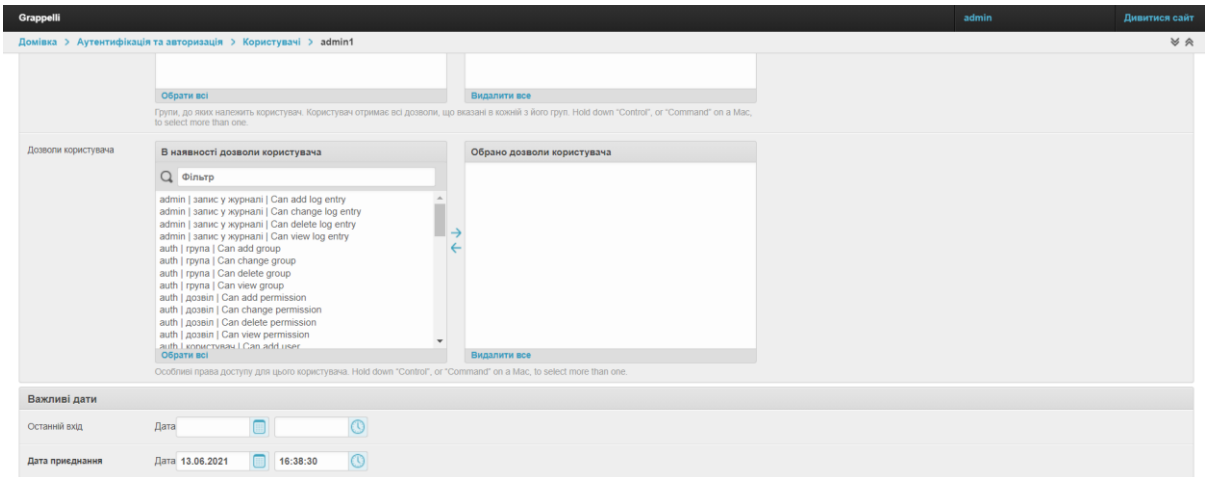


Рис. 48 Приклад редагування прав нового користувача (2)

## ВИСНОВКИ

В процесі роботи над дипломним проектом були здобуті наступні результати:

- Компоненти для розробки FrontEnd`у;
  - Виконаний огляд популярних CSS фреймворків;
  - визначені головні переваги і функціонал фреймворку Bootstrap;
  
- Вибір засобу розробки для BackEnd`у;
  - Вибір СУБД.;
  - Вибір хостингу;
  - Вибір системи управління версіями;

Практичним значенням виконаної роботи полягає у створенні сайту для надання інформаційних послуг клієнтам стоматологічної клініки “VlaDent”

Використання фреймворку Django значно спрощує розробку сайту, надає багато можливостей по створенню проекту. Django допомагає писати програмне забезпечення, яке буде: повним, різнобічним, безпечним, масштабованим, зручним в супроводі, переносним.

Використання фреймворку Bootstrap полегшує верстку сайту, надає інструменти для адаптивної верстки також кросбраузерність.

Також у Bootstrap велике російськомовне співтовариство і наявність хорошої документації.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Разработка со стороны Front end — что это такое и чем отличается от Back end? URL:<https://dan-it.com.ua/razrabotka-so-storony-front-end-chto-jeto-takoe-i-chem-otlichaetsja-ot-back-end/>  
(дата звернення: 17.03.2021)
- [2] HTML и CSS: 25 уроков для начинающих. Владимир Александрович Дронов -2020- 12с. URL:  
[https://books.google.com.ua/books/about/HTML\\_%D0%B8\\_CSS\\_25\\_%D1%83%D1%80%D0%BE%D0%BA%D0%BE%D0%B2\\_%D0%B4%D0%BB%D1%8F\\_%D0%BD%D0%B0.html?id=m9UHEAAAQBAJ&source=kp\\_book\\_description&redir\\_esc=y](https://books.google.com.ua/books/about/HTML_%D0%B8_CSS_25_%D1%83%D1%80%D0%BE%D0%BA%D0%BE%D0%B2_%D0%B4%D0%BB%D1%8F_%D0%BD%D0%B0.html?id=m9UHEAAAQBAJ&source=kp_book_description&redir_esc=y)
- [4] JavaScript. URL: <https://ru.wikipedia.org/wiki/JavaScript>  
(дата звернення 24.03.2021)
- [5] ТОП 10 лучших фреймворков для Front-end Dev.  
URL: <https://web-academy.com.ua/stati/336-10-front-end-dev-2018>  
(дата звернення: 25.03.2021)
- [6] Верстать быстро и красиво: 15 популярных CSS фреймворков. URL:  
<https://proglib.io/p/verstat-bystro-i-krasivo-15-populyarnyh-css-freymvorkov-2020-01-16>  
(дата звернення 26.03.2021)
- [7] Bootstrap(фреймворк). URL:  
[https://ru.wikipedia.org/wiki/Bootstrap\\_\(%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA\)](https://ru.wikipedia.org/wiki/Bootstrap_(%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA))  
(дата звернення: 29.03.2021)
- [8] Foundation (framework). URL:  
[https://en.wikipedia.org/wiki/Foundation\\_\(framework\)](https://en.wikipedia.org/wiki/Foundation_(framework))  
(дата звернення: 01.04.2021)
- [9] What is Tailwind CSS.

URL: <https://channel9.msdn.com/Shows/Web-Wednesday/What-is-Tailwind-CSS#:~:text=Tailwind%20CSS%20is%20self%2Ddescribed,styles%20and%20change%20the%20layout.>

(дата звернення 05.04.2021)

[10] Плюсы и минусы Bootstrap.

URL: <https://timeweb.com/ru/community/articles/plyusy-i-minusy-bootstrap-1>

(дата звернення: 10.04.2021)

[11] Изучаем файловую структуру Bootstrap. зображення. URL:

[https://www.google.com/search?q=%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0+bootstrap&rlz=1C1OKWM\\_ruUA906UA906&sxsrf=ALeKk02ZjWlgCM6-](https://www.google.com/search?q=%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0+bootstrap&rlz=1C1OKWM_ruUA906UA906&sxsrf=ALeKk02ZjWlgCM6-)

[0Uhs6qBORS9U68bmgA:1623587806974&source=lnms&tbm=isch&sa=X&ved=2ahUKEwj95fvzz5TxAhXJlosKHdvNApYQ\\_AUoAXoECAEQAw&biw=1536&bih=722#imgrc=pYohEulHc30foM](https://www.google.com/search?q=%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0+bootstrap&rlz=1C1OKWM_ruUA906UA906&sxsrf=ALeKk02ZjWlgCM6-0Uhs6qBORS9U68bmgA:1623587806974&source=lnms&tbm=isch&sa=X&ved=2ahUKEwj95fvzz5TxAhXJlosKHdvNApYQ_AUoAXoECAEQAw&biw=1536&bih=722#imgrc=pYohEulHc30foM) (дата звернення: 22.03.2021)

[12] Django (фреймворк)

<https://uk.wikipedia.org/wiki/Django>

(дата звернення 22.03.2021)

[13] Система управління базами даних

[https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0\\_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%96%D0%BD%D0%BD%D1%8F\\_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8\\_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85](https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%96%D0%BD%D0%BD%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)

(дата звернення 24.03.2021)

[14] MySQL (СУБД)

<https://uk.wikipedia.org/wiki/MySQL>

(дата звернення 24.03.2021)

[15] MongoDB (СУБД)

<https://uk.wikipedia.org/wiki/MongoDB>

(дата звернення 24.03.2021)

[16] Поняття фреймворку (програмний каркас)

[https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B8%D0%B9\\_%D0%BA%D0%B0%D1%80%D0%BA%D0%B0%D1%81](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B8%D0%B9_%D0%BA%D0%B0%D1%80%D0%BA%D0%B0%D1%81)

(дата звернення 25.03.2021)

[17] Який backend потрібний вашому веб-сайту?

<https://goldwebsolutions.com/uk/blog/yakij-backend-potribnij-vashomu-veb-sajtu/>

(дата звернення 25.03.2021)

[18] Heroku (хмарна PaaS-платформа)

<https://uk.wikipedia.org/wiki/Heroku>

(дата звернення 25.03.2021)

[19] GIT (система керування версіями)

<https://uk.wikipedia.org/wiki/Git>

(дата звернення 26.03.2021)

[20] Рабочий процесс Gitflow Workflow

<https://www.atlassian.com/ru/git/tutorials/comparing-workflows/gitflow-workflow>

(дата звернення 26.03.2021)

[21] Django ORM для начинающих | Оптимизируем запросы

<https://habr.com/ru/post/503526/>

(дата звернення 27.03.2021)

## ДОДАТОК А

## Відомість матеріалів кваліфікаційної роботи

		Позначення	Найменування	Кільк. аркушів	Примітка
			Документація		
		<b>ІТКІ.КР 19.03.ДА.ПЗ</b>	Пояснювальна записка	122	
			Презентація	17	
			Диск CD с презентацією	2	
			<b>ІТКІ.КР 19.03.ДА.ПЗ</b>		
Зм.	Ар-куш	№ докум	Підпис	Дата	
Розроб.		Д.П. Юхимчук			<b>Матеріали кваліфікаційної роботи</b>  <b>НТУ «ДП», 126-17-1</b>
Керівник		Г.М. Коротенко			
Рецензент		Л.М. Коротенко			
Н.контр.		Г.М. Коротенко			
Зав.каф.		В.В.Гнатушенко			
				Літ.	Аркуш
					1

## Приклади коду програми

## Settings.py

```
from pathlib import Path
import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '$k$y$om(ppz@u+v!ow+49c!#tcwc=u@+gfn(&3un#8hc0#w3e('

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ['*']

# Application definition

INSTALLED_APPS = [
    'ckeditor', # Text for blog
    'ckeditor_uploader', # Text for blog
    'grappelli', # Frontend for admin

    'Django_cleanup.apps.CleanupConfig',
    'Django.contrib.admin',
    'Django.contrib.auth',
    'Django.contrib.contenttypes',
    'Django.contrib.sessions',
    'Django.contrib.messages',
    'Django.contrib.staticfiles',

    'apps.main',
    'apps.aboutUs',
    'apps.medic',
    'apps.treatments',
    'apps.pricing',
    'apps.blog',
    'apps.contact',
    'apps.record'
```

```
]
```

```
MIDDLEWARE = [
    'Django.middleware.security.SecurityMiddleware',
    'whitenoise.middleware.WhiteNoiseMiddleware',
    'Django.contrib.sessions.middleware.SessionMiddleware',
    'Django.middleware.common.CommonMiddleware',
    'Django.middleware.csrf.CsrfViewMiddleware',
    'Django.contrib.auth.middleware.AuthenticationMiddleware',
    'Django.contrib.messages.middleware.MessageMiddleware',
    'Django.middleware.clickjacking.XFrameOptionsMiddleware',
```

```
]
```

```
STATICFILES_STORAGE = 'whitenoise.storage.CompressedManifestStaticFilesStorage'
```

```
ROOT_URLCONF = 'VlaDent.urls'
```

```
TEMPLATES = [
    {
        'BACKEND': 'Django.template.backends.Django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'Django.template.context_processors.debug',
                'Django.template.context_processors.request',
                'Django.contrib.auth.context_processors.auth',
                'Django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

```
WSGI_APPLICATION = 'VlaDent.wsgi.application'
```

```
# Database
```

```
# https://docs.djangoproject.com/en/3.1/ref/settings/#databases
```

```
DATABASES = {
    'default': {
        'ENGINE': 'Django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

```
# Password validation
```

```
# https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validators
```



```
AUTH_PASSWORD_VALIDATORS = [  
    {  
        'NAME': 'Django.contrib.auth.password_validation.UserAttributeSimilarityValidator',  
    },  
    {  
        'NAME': 'Django.contrib.auth.password_validation.MinimumLengthValidator',  
    },  
    {  
        'NAME': 'Django.contrib.auth.password_validation.CommonPasswordValidator',  
    },  
    {  
        'NAME': 'Django.contrib.auth.password_validation.NumericPasswordValidator',  
    },  
]
```

```
# Internationalization
```

```
# https://docs.djangoproject.com/en/3.1/topics/i18n/
```

```
LANGUAGE_CODE = 'uk'
```

```
TIME_ZONE = 'UTC'
```

```
USE_I18N = True
```

```
USE_L10N = True
```

```
USE_TZ = True
```

```
# Ckeditor config
```

```
CKEDITOR_CONFIGS = {  
    'default': {  
        'toolbar': 'None'  
    },  
}
```

```
CKEDITOR_UPLOAD_PATH = 'static/ckeditor/'
```

```
# For mailinig
```

```
EMAIL_BACKEND = 'Django.core.mail.backends.smtp.EmailBackend'
```

```
EMAIL_USE_TLS = True
```

```
EMAIL_HOST = 'smtp.gmail.com'
```

```
EMAIL_PORT = 587
```

```
EMAIL_HOST_USER = 'vladent.dentistry@gmail.com'
```

```
EMAIL_HOST_PASSWORD = 'nvuonizzzpvkrnry'
```

```
# Static files (CSS, JavaScript, Images)
```

```
# https://docs.djangoproject.com/en/3.1/howto/static-files/

STATIC_URL = '/static/'
# STATIC_ROOT = os.path.join(BASE_DIR, 'static')
STATICFILES_DIRS = (os.path.join(BASE_DIR, 'static'),)
STATICFILES_STORAGE = 'whitenoise.Django.GzipManifestStaticFilesStorage'

MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')

# For Heroku
import dj_database_url
import Django_heroku

Django_heroku.settings(locals())
STATICFILES_STORAGE = 'whitenoise.storage.CompressedManifestStaticFilesStorage'
db_from_env = dj_database_url.config(conn_max_age=600, ssl_require=True)
DATABASES['default'].update(db_from_env)
```

## Url.py

```
from Django.contrib import admin
from Django.urls import path, include
from Django.conf import settings
from Django.conf.urls.static import static

urlpatterns = [
    path('ckeditor/', include('ckeditor_uploader.urls')),
    path('grappelli/', include('grappelli.urls')),
    path('admin/', admin.site.urls),
    path("", include('apps.main.urls')),
    path('about-us', include('apps.aboutUs.urls')),
    path('medic', include('apps.medic.urls')),
    path('treatments', include('apps.treatments.urls')),
    path('pricing', include('apps.pricing.urls')),
    path('blog/', include('apps.blog.urls')),
    path('contact', include('apps.contact.urls'))
]

urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

## Apps/main/views.py

```
from Django.shortcuts import render
from Django.views.generic import View
from apps.record.models import Record, Subscribe
from apps.blog.models import Article
from apps.medic.models import Medic
```

```

from apps.pricing.models import Price
from apps.contact.views import ContactView
# Create your views here.

class Main(View):
    def get(self, request, alert=False, *args, **kwargs):
        articles = Article.objects.filter(published=True)[:3]
        medics = Medic.objects.all()[:4]
        prices = list(Price.objects.all()[:4])

        return render(request, 'main/index.html', {'index': True, 'alert': alert, 'contact': ContactView.get_contact(),
'articles': articles, 'medics': medics, 'prices': prices})

    def post(self, request, *args, **kwargs):
        if 'name' in request.POST:
            treatments = request.POST['service'] if 'service' in request.POST else 1
            Record.objects.create(name = request.POST['name'],
                surname = request.POST['surname'],
                phone_number = request.POST['phone_number'],
                treatments = Price.objects.get(pk = treatments))

        if 'send-mail' in request.POST:
            Subscribe.objects.create(email = request.POST['send-mail'])

        return self.get(request, alert = True)

```

## Apps/aboutUs/views.py

```

from Django.shortcuts import render
from Django.views.generic import View
from apps.record.models import Record, Subscribe
from apps.pricing.models import Price
from apps.contact.views import ContactView
# Create your views here.

class AboutUs(View):
    def get(self, request, alert=False, *args, **kwargs):
        return render(request, 'aboutUs/aboutUs.html', {'aboutus': True, 'alert':
alert, 'contact': ContactView.get_contact()})

    def post(self, request, *args, **kwargs):
        if 'name' in request.POST:

```

```

        treatments = request.POST['service'] if 'service' in request.POST
else 1
    Record.objects.create(name = request.POST['name'],
                           surname = request.POST['surname'],
                           phone_number = request.POST['phone_number'],
                           treatments = Price.objects.get(pk =
treatments))

    if 'send-mail' in request.POST:
        Subscribe.objects.create(email = request.POST['send-mail'])

    return self.get(request, alert = True)

```

## Apps/medic/views.py

```

from Django.shortcuts import render
from Django.views.generic import View
from apps.record.models import Subscribe
from apps.contact.views import ContactView
from .models import Medic
# Create your views here.

class MedicView(View):
    def get(self, request, alert=False, *args, **kwargs):
        medics = Medic.objects.all()
        return render(request, 'medic/medic.html', {'medic': True, 'alert': alert, 'medics': medics, 'contact':
ContactView.get_contact()})

    def post(self, request, *args, **kwargs):
        if 'send-mail' in request.POST:
            Subscribe.objects.create(email = request.POST['send-mail'])

        return self.get(request, alert=True)

```

## Apps/treatments/views.py

```

from Django.shortcuts import render
from Django.views.generic import View
from apps.record.models import Record, Subscribe
from apps.contact.views import ContactView
# Create your views here.

class Treatments(View):
    def get(self, request, alert=False, *args, **kwargs):

```

```
return render(request, 'treatments/treatments.html', {'treatments': True, 'alert': alert, 'contact': ContactView.get_contact()})
```

```
def post(self, request, *args, **kwargs):
    if 'send-mail' in request.POST:
        Subscribe.objects.create(email = request.POST['send-mail'])

    return self.get(request, alert=True)
```

## Apps/pricing/views.py

```
from Django.shortcuts import render
from Django.views.generic import View
from apps.record.models import Subscribe, Record
from apps.contact.views import ContactView
from .models import Price
# Create your views here.
```

```
class Pricing(View):
    def get(self, request, alert=False, *args, **kwargs):
        prices = list(Price.objects.all()[:4])
        return render(request, 'pricing/pricing.html', {'pricing': True, 'alert': alert, 'contact': ContactView.get_contact(), 'prices':
prices})
```

```
def post(self, request, *args, **kwargs):
    if 'name' in request.POST:
        treatments = request.POST['service'] if 'service' in request.POST else 1
        Record.objects.create(name = request.POST['name'],
                               surname = request.POST['surname'],
                               phone_number = request.POST['phone_number'],
                               treatments = Price.objects.get(pk = treatments))

    if 'send-mail' in request.POST:
        Subscribe.objects.create(email = request.POST['send-mail'])

    return self.get(request, alert = True)
```

## Apps/blog/views.py

```
from Django.shortcuts import render, Http404
from Django.views.generic import View
from apps.record.models import Subscribe
from apps.contact.views import ContactView
from .models import Article
# Create your views here.
```

```

class Blog(View):
    def get(self, request, alert=False, *args, **kwargs):
        articles = Article.objects.filter(published=True)[:6]
        return render(request, 'blog/blog.html', {'blog': True, 'alert': alert, 'articles': articles, 'contact': ContactView.get_contact()})

    def post(self, request, *args, **kwargs):
        if 'send-mail' in request.POST:
            Subscribe.objects.create(email = request.POST['send-mail'])

        return self.get(request, alert=True)

```

```

class ArticleView(View):
    def get(self, request, article_id, alert=False, *args, **kwargs):
        try:
            article = Article.objects.get(id=article_id)
        except:
            raise Http404()
        return render(request, 'blog/article.html', {'blog': True, 'alert': alert, 'article': article, 'contact': ContactView.get_contact()})

    def post(self, request, article_id, *args, **kwargs):
        if 'send-mail' in request.POST:
            Subscribe.objects.create(email = request.POST['send-mail'])

        return self.get(request, article_id, alert=True)

```

## Apps/contact/views.py

```

from Django.shortcuts import render
from Django.views.generic import View
from apps.record.models import Subscribe
from .models import Contact
# Create your views here.

```

```

class ContactView(View):
    @staticmethod
    def get_contact():
        return Contact.objects.first()

    def get(self, request, alert=False, *args, **kwargs):
        return render(request, 'contact/contact.html', {'contactview': True, 'alert': alert, 'contact': ContactView.get_contact()})

```

```
def post(self, request, *args, **kwargs):
    if 'send-mail' in request.POST:
        Subscribe.objects.create(email = request.POST['send-mail'])

    return self.get(request, alert=True)
```

## Index.html

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Клініка VlaDent</title>

    <link sizes="32x32" rel="icon" href='static/images/decay.png' type='image/png'>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuuqIj61tLrc4wSX0szH/Ev+nYRRuWlolfffl"
crossorigin="anonymous">

    <link href="https://fonts.googleapis.com/css?family=Poppins:300,400,500,600,700"
rel="stylesheet">

    <link rel="stylesheet" type="text/css" href="static/fonts/flaticon/font/flaticon.css">
    <link rel="stylesheet" href="static/stylesheets/icomoon.css">
    <link rel='stylesheet' href='static/stylesheets/style.css'>

</head>

<body>
    <div class="pb-md-5 pt-md-4 py-4 border-bottom">
        <div class="container">
            <div class="row no-gutters d-flex align-items-start align-items-center px-3 px-md-0">
                <div class="col-md-4 order-md-2 mb-2 mb-md-0 align-items-center text-center">
                    <a class="navbar-brand" href="index.html">VlaDent <span>стоматологічна
клініка</span></a>
                </div>
```

```

right">
    <div class="col-md-4 order-md-1 d-flex toppler mb-md-0 mb-2 align-items-center text-md-
    <div class="pr-md-4 pl-md-0 pl-3 text text">
        <p class="con"><span>Безкоштовний дзвінок</span> <span><a href='tel://+380 99 200
1234' style='color:black;'>+380 99 200 1234</a></span></p>
        <p class="con">проспект Дмитра Яворницького, 19, Дніпро</p>
    </div>
</div>
<div class="col-md-4 order-md-2 d-flex toppler mb-md-0 align-items-center">
    <div class="text pl-3 pl-md-5">
        <p class="hr"><span>Години роботи</span></p>
        <p class="time"><span>Пн - Сб:</span> <span>8:00 - 19:00</span> / Нд: Закрито</p>
    </div>
</div>
</div>
</div>
</div>
</div>
<!-- button for admin -->
<div class='admin-btn'>
    <a href='#' class='icon-gear'></a>
</div>
<!-- alert -->
<div class="alert alert-success" role="alert">
    Ваш запит оброблено!
</div>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark ftco-navbar-light" id="ftco-navbar">
    <div class="container-fluid d-flex align-items-center">
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#ftco-
nav" aria-controls="ftco-nav" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon pl-4 pt-1">Меню</span>
        </button>
        <div class="collapse navbar-collapse" id="ftco-nav">
            <ul class="navbar-nav m-auto">
                <li class="nav-item active"><a href="index.html" class="nav-link pl-0">Головна</a></li>
                <li class="nav-item"><a href="aboutUs.html" class="nav-link">Про нас</a></li>
                <li class="nav-item"><a href="doctors.html" class="nav-link">Лікарі</a></li>
                <li class="nav-item"><a href="treatments.html" class="nav-link">Лікування</a></li>
                <li class="nav-item"><a href="pricing.html" class="nav-link">Ціноутворення</a></li>
                <li class="nav-item"><a href="blog.html" class="nav-link">Блог</a></li>
                <li class="nav-item"><a href="contact.html" class="nav-link">Зв'язок</a></li>
            </ul>
        </div>
    </div>
</nav>
<!-- END nav -->

```



```

<section class="home-slider carousel slide carousel-fade" data-bs-ride="carousel" data-bs-
interval="4000">
  <div class="carousel-inner">
    <div class="carousel-item active slider-item" style="background-
image:url(static/images/bg_1.jpg);">
      <div class='container'>
        <div class="row slider-text">
          <div class="carousel-caption col-12 text text-carousel pb-5">
            <h1 class="mb-4">Допомога Вам <span>Залишайся щасливим</span></h1>
            <h3 class="subheading">Щодня ми несемо надію і посміхаємось пацієнту, якому
служимо.</h3>
          </div>
        </div>
      </div>
    <div class="carousel-item slider-item" style="background-image:url(static/images/bg_2.jpg);">
      <div class='container'>
        <div class="row slider-text">
          <div class="carousel-caption col-12 text text-carousel pb-5">
            <h1 class="mb-4">Посмішка робить <br>Тривале враження</h1>
            <h3 class="subheading">Ваше здоров'я - наш головний пріоритет завдяки всебічній
доступній медичній допомозі.</h3>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>

<section class="ftco-section ftco-no-pt ftco-no-pb">
  <div class="container">
    <div class="row no-gutters">
      <div class="col-md-5 p-md-5 img img-2 mt-5 mt-md-0" style="background-image:
url(static/images/about.jpg);">
      </div>
      <div class="col-md-7 wrap-about py-4 py-md-5 ">
        <div class="heading-section mb-5">
          <div class="pl-md-5 ml-md-5 pt-md-5">
            <span class="subheading mb-2">Ласкаво просимо до VlaDent</span>
            <h2 class="mb-2" style="font-size: 32px;">Медична спеціальність, що займається
доглядом гостро хворих госпіталізованих пацієнтів</h2>
          </div>
        </div>
      <div class="pl-md-5 ml-md-5 mb-5">

```



```

        <input type="tel" name='phone_number' class="form-control"
placeholder="Телефонний номер" required>
    </div>
</div>
<div class="col-md-12 col-lg-6 col-xl-4">
    <div class="form-group">
        <input type="submit" value="Передзвонити" class="btn btn-secondary py-2 px-4">
    </div>
</div>
</div>
</form>
</div>
</div>
<div class="col-md-3 d-flex align-items-stretch">
    <div class="consultation w-100 text-center px-4 px-md-5">
        <h3 class="mb-4">Висококваліфікований лікарський склад</h3>
        <p>Колектив лікарів високого класу різних спеціалізацій. У пацієнта є можливість за
одне відвідування отримати консультацію декількох фахівців.</p>
    </div>
</div>
</div>
</div>
</div>
</section>

```

```

<section class="ftco-section ftco-services">
    <div class="container">
        <div class="row justify-content-center mb-5 pb-2">
            <div class="col-md-8 text-center heading-section ">
                <span class="subheading">Послуги</span>
                <h2 class="mb-4">Наші клінічні послуги</h2>
            </div>
        </div>
        <div class="row">
            <div class="col-md-3 d-flex services align-self-stretch p-4 ">
                <div class="media block-6 d-block text-center">
                    <div class="icon d-flex justify-content-center align-items-center">
                        <span class="flaticon-drilling"></span>
                    </div>
                    <div class="media-body p-2 mt-3">
                        <h3 class="heading">Зубні імплантати</h3>
                        <p>Дентальна імплантація – установка в щелепу штучного кореня, виготовленого з титану для відновлення зуба. Така технологія дозволить повністю відновити, як один, так і кілька зубів аж до повного відновлення зубного ряду.</p>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

<div class="media block-6 d-block text-center">
  <div class="icon d-flex justify-content-center align-items-center">
    <span class="flaticon-tooth"></span>
  </div>
  <div class="media-body p-2 mt-3">
    <h3 class="heading">Косметична стоматологія</h3>
    <p>Ефективний спосіб видалення нальоту і зубного каменю і одночасно дієвий інструмент профілактичної стоматології, який допомагає боротися з бактеріями, що провокують захворювання зубів і ясен.</p>
  </div>
</div>
</div>
<div class="col-md-3 d-flex services align-self-stretch p-4 ">
  <div class="media block-6 d-block text-center">
    <div class="icon d-flex justify-content-center align-items-center">
      <span class="flaticon-dental-floss"></span>
    </div>
    <div class="media-body p-2 mt-3">
      <h3 class="heading">Лікування карієсу</h3>
      <p>Методи лікування карієсу залежать від стадії захворювання, локалізації руйнівного процесу. Якщо карієс був виявлений на стадії плями, буде достатньо консервативного методу лікування.</p>
    </div>
  </div>
</div>
</div>
<div class="col-md-3 d-flex services align-self-stretch p-4 ">
  <div class="media block-6 d-block text-center">
    <div class="icon d-flex justify-content-center align-items-center">
      <span class="flaticon-shiny-tooth"></span>
    </div>
    <div class="media-body p-2 mt-3">
      <h3 class="heading">Відбілювання зубів</h3>
      <p>Технологія відбілювання зубів із застосуванням лазера ґрунтується на використанні відбілювального кисневмісного гелю. Під впливом лазерного випромінювання компоненти гелю активуються, проникають у тканини зуба, у результаті чого відбувається зміна пігменту дентину.</p>
    </div>
  </div>
</div>
</div>
<div class="col-md-3 d-flex services align-self-stretch p-4 ">
  <div class="media block-6 d-block text-center">
    <div class="icon d-flex justify-content-center align-items-center">
      <span class="flaticon-dentist-chair"></span>
    </div>
    <div class="media-body p-2 mt-3">
      <h3 class="heading">Екстракція зуба</h3>

```

<p>Екстракція зуба (видалення) – одна з найпоширеніших процедур у хірургічній стоматології. Сучасні знеболювальні препарати, нові технології дозволяють провести операцію швидко, без болю і ускладнень.</p>

</div>

</div>

</div>

<div class="col-md-3 d-flex services align-self-stretch p-4 ">

<div class="media block-6 d-block text-center">

<div class="icon d-flex justify-content-center align-items-center">

<span class="flaticon-tooth-1"></span>

</div>

<div class="media-body p-2 mt-3">

<h3 class="heading">Пародонтологія</h3>

<p>Пародонтит – це запальний процес, який поширюється на всі тканини пародонту навколо одного, декількох зубів або навіть охоплює весь зубний ряд.</p>

</div>

</div>

</div>

<div class="col-md-3 d-flex services align-self-stretch p-4 ">

<div class="media block-6 d-block text-center">

<div class="icon d-flex justify-content-center align-items-center">

<span class="flaticon-tooth-with-braces"></span>

</div>

<div class="media-body p-2 mt-3">

<h3 class="heading">Затиск & Брекети</h3>

<p>Брекети — незнімні конструкції, які фіксуються на зовнішній або внутрішній частини зубів за допомогою спеціального клею.</p>

</div>

</div>

</div>

<div class="col-md-3 d-flex services align-self-stretch p-4 ">

<div class="media block-6 d-block text-center">

<div class="icon d-flex justify-content-center align-items-center">

<span class="flaticon-decayed-tooth"></span>

</div>

<div class="media-body p-2 mt-3">

<h3 class="heading">Кореневого каналу</h3>

<p>Найпоширеніші хвороби ясен – гінгівіт і пародонтит. Вони з'являються при недотриманні ретельної гігієни, без використання високоефективних засобів для чищення зубів.</p>

</div>

</div>

</div>

</div>

</div>

</section>

<section class="ftco-section intro attachment" style="background-image: url(static/images/bg\_3.jpg);">

```

<div class="container">
  <div class="row">
    <div class="col-md-6">
      <h3 class="mb-4">Ми пообіцяли подбати про наших пацієнтів і доставили.</h3>
      <p>Цифрове обладнання кращих світових брендів і ультрасучасні комп'ютерні 3D-
технології.</p>
    </div>
  </div>
</div>
</section>

<section class="ftco-section">
  <div class="container">
    <div class="row justify-content-center mb-5 pb-2">
      <div class="col-md-8 text-center heading-section ">
        <h2 class="mb-4">Лікарі</h2>
      </div>
    </div>
    <div class="row">
      <div class="col-md-6 col-lg-3 ">
        <div class="staff">
          <div class="img-wrap d-flex align-items-stretch">
            <div class="img align-self-stretch" style="background-image: url(static/images/doc-
1.jpg);"></div>
          </div>
          <div class="text pt-3 text-center">
            <h3>Доктор Ллойд Вілсон</h3>
            <span class="position mb-2">Лікар-терапевт</span>
            <div class="faded">
              <p>За плечима цього фахівця більше 16 років практичного досвіду. Велика база
теоретичних знань і роки практики дозволяють йому швидко аналізувати ситуацію і вибрати
найоптимальніше рішення навіть у найскладніших випадках, що дуже цінують пацієнти.</p>
            <ul class="ftco-social text-center">
              <li class=""><a href="#" class="d-flex align-items-center justify-content-center"><span
class="icon-facebook"></span></a></li>
              <li class=""><a href="#" class="d-flex align-items-center justify-content-center"><span
class="icon-instagram"></span></a></li>
            </ul>
          </div>
        </div>
      </div>
      <div class="col-md-6 col-lg-3 ">
        <div class="staff">
          <div class="img-wrap d-flex align-items-stretch">
            <div class="img align-self-stretch" style="background-image: url(static/images/doc-
2.jpg);"></div>
          </div>
          <div class="text pt-3 text-center">

```

```

<h3>Доктор Рейчел Паркер</h3>
<span class="position mb-2">Ортопед</span>
<div class="faded">
  <p>Професіонал з 20-річним стажем роботи, досвідчений пародонтолог. Пацієнти,
які приходять на консультацію з головним лікарем, завжди отримують її рада і цінні рекомендації по
своїй ситуації. Все це допомагає зробити процес лікування максимально ефективним.</p>
  <ul class="ftco-social text-center">
    <li class=""><a href="#" class="d-flex align-items-center justify-content-center"><span
class="icon-facebook"></span></a></li>
    <li class=""><a href="#" class="d-flex align-items-center justify-content-center"><span
class="icon-instagram"></span></a></li>
  </ul>
</div>
</div>
</div>
</div>
<div class="col-md-6 col-lg-3 ">
  <div class="staff">
    <div class="img-wrap d-flex align-items-stretch">
      <div class="img align-self-stretch" style="background-image: url(static/images/doc-
3.jpg);"></div>
      </div>
      <div class="text pt-3 text-center">
        <h3>Доктор Ян Сміт</h3>
        <span class="position mb-2">Хірург</span>
        <div class="faded">
          <p>Має 15літній практичний досвід. Цінує правильний підхід до лікування, якісний,
швидкий і безболісний. Регулярно відвідує профільні курси та майстер-класи, освоює новітні методики
лікування. Вільний час присвячує туризму і велопогулянкам.</p>
          <ul class="ftco-social text-center">
            <li class=""><a href="#" class="d-flex align-items-center justify-content-center"><span
class="icon-facebook"></span></a></li>
            <li class=""><a href="#" class="d-flex align-items-center justify-content-center"><span
class="icon-instagram"></span></a></li>
          </ul>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="col-md-6 col-lg-3 ">
  <div class="staff">
    <div class="img-wrap d-flex align-items-stretch">
      <div class="img align-self-stretch" style="background-image: url(static/images/doc-
4.jpg);"></div>
      </div>
      <div class="text pt-3 text-center">
        <h3>Доктор Алісія Хендерсон</h3>
        <span class="position mb-2">Пародонтолог</span>
        <div class="faded">

```





```

    <p class="name">Раки Хендерсон</p>
    <span class="position">Керуючий справами</span>
  </div>
</div>
<div class="col-md-6 item">
  <div class="single-box testimony-wrap d-flex mb-5">
    
    <div class="text pl-4 bg-light">
      <span class="quote d-flex align-items-center justify-content-center">
        <i class="icon-quote-left"></i>
      </span>
      <p>Робив свої зуби в VlaDent і можу сказати що це явно одна з кращих
стоматологий Дніпра, ввічливо і доброзичливо зустріли, всюди чистота, новітнє обладнання найвищої
якості, лікарі просто супер.</p>
      <p class="name">Генри Ди</p>
      <span class="position">Бізнесмен</span>
    </div>
  </div>
</div>
<div class="carousel-item">
  <div class="row">
    <div class="col-md-6 item">
      <div class="single-box testimony-wrap d-flex mb-5">
        
        <div class="text pl-4 bg-light">
          <span class="quote d-flex align-items-center justify-content-center">
            <i class="icon-quote-left"></i>
          </span>
          <p>Звертався сюди з проблемним зубом попереду, після консультації сказали, що
потрібно ставити коронку. Ставили керамічну, зробили добре, зуб виглядає органічно, ідеально
підібраний колір! Я задоволений!</p>
          <p class="name">Марк Хафф</p>
          <span class="position">Фермер</span>
        </div>
      </div>
    </div>
  </div>
<div class="col-md-6">
  <div class="single-box testimony-wrap d-flex mb-5">
    
    <div class="text pl-4 bg-light">
      <span class="quote d-flex align-items-center justify-content-center">
        <i class="icon-quote-left"></i>
      </span>
      <p>Батько постійно скаржився на те, що хоче зробити собі зуби. Відправив її в
VlaDent тепер задоволений!) Спасибі лікарям за те, що якісно зробили свою роботу і мій батько тепер
задоволений своєю посмішкою!</p>

```



```

<div class="row">
  <div class="col-md-9">
    <h2>Надаємо безкоштовну стоматологічну консультацію</h2>
    <p class="mb-0">Ваше здоров'я - наш головний пріоритет з комплексним доступним
медичним обслуговуванням ..</p>
    <p></p>
  </div>
  <div class="col-md-3 d-flex align-items-center">
    <p class="mb-0"><button class="btn btn-secondary px-4 py-3" style='padding-bottom: 0rem
!important; id='btn-anim'>Безкоштовна Консультація</p>
  </div>
</div>
</div>
</div>
</section>

<section class="ftco-section">
  <div class="container">
    <div class="row justify-content-center mb-5 pb-2">
      <div class="col-md-8 text-center heading-section ">
        <span class="subheading">Ціноутворення</span>
        <h2 class="mb-4">Наші ціни</h2>
        <p>VlaDent – високий рівень сервісу, найкраще обладнання, успішні і досвідчені лікарі,
матеріали від провідних світових виробників, унікальна система контролю якості, чудовий результат і
нескінченна турбота про Ваше здоров'я!</p>
      </div>
    </div>
    <div class="row">
      <div class="col-md-3 ">
        <div class="pricing-entry pb-5 text-center">
          <div>
            <h3 class="mb-4">Базовий</h3>
            <p><span class="price">#8372 800</span> <span class="per">/ сесія</span></p>
          </div>
          <ul>
            <li>Діагностичні послуги</li>
            <li>Професійна консультація</li>
            <li>&nbsp;</li>
            <li>&nbsp;</li>
            <li>&nbsp;</li>
          </ul>
          <p class="button text-center"><button type='button' class="btn btn-primary px-4 py-3" data-
bs-toggle="modal" data-bs-target="#exampleModal" onclick='selectedService(2)'>Отримати
пропозицію</button></p>
        </div>
      </div>
      <div class="col-md-3 ">
        <div class="pricing-entry pb-5 text-center">
          <div>
            <h3 class="mb-4">Базовий</h3>
            <p><span class="price">#8372 800</span> <span class="per">/ сесія</span></p>
          </div>
          <ul>
            <li>Діагностичні послуги</li>
            <li>Професійна консультація</li>
            <li>&nbsp;</li>
            <li>&nbsp;</li>
            <li>&nbsp;</li>
          </ul>
          <p class="button text-center"><button type='button' class="btn btn-primary px-4 py-3" data-
bs-toggle="modal" data-bs-target="#exampleModal" onclick='selectedService(2)'>Отримати
пропозицію</button></p>
        </div>
      </div>
    </div>
  </div>

```

```

<h3 class="mb-4">Стандарт</h3>
<p><span class="price">&#8372 1200</span> <span class="per"/>/ сесія</span></p>
</div>
<ul>
<li>Діагностичні послуги</li>
<li>Професійна консультація</li>
<li>Відбілювання зубів</li>
<li>&ensp;</li>
<li>&ensp;</li>
</ul>
<p class="button text-center"><button type='button' class="btn btn-primary px-4 py-3" data-
bs-toggle="modal" data-bs-target="#exampleModal" onclick='selectedService(3)'+>Отримати
пропозицію</button></p> </div>
</div>
<div class="col-md-3 ">
<div class="pricing-entry active pb-5 text-center">
<div>
<h3 class="mb-4">Преміум</h3>
<p><span class="price">&#8372 1600</span> <span class="per"/>/ сесія</span></p>
</div>
<ul>
<li>Діагностичні послуги</li>
<li>Професійна консультація</li>
<li>Зубні імплантати</li>
<li>Відбілювання зубів</li>
<li>&ensp;</li>
</ul>
<p class="button text-center"><button type='button' class="btn btn-primary px-4 py-3" data-
bs-toggle="modal" data-bs-target="#exampleModal" onclick='selectedService(4)'+>Отримати
пропозицію</button></p> </div>
</div>
<div class="col-md-3 ">
<div class="pricing-entry pb-5 text-center">
<div>
<h3 class="mb-4">Платина</h3>
<p><span class="price">&#8372 2100</span> <span class="per"/>/ сесія</span></p>
</div>
<ul>
<li>Діагностичні послуги</li>
<li>Професійна консультація</li>
<li>Зубні імплантати</li>
<li>Хірургічні вилучення</li>
<li>Відбілювання зубів</li>
</ul>
<p class="button text-center"><button type='button' class="btn btn-primary px-4 py-3" data-
bs-toggle="modal" data-bs-target="#exampleModal" onclick='selectedService(5)'+>Отримати
пропозицію</button></p> </div>
</div>
</div>

```

```

</div>
</section>

<!-- Modal for button-->
<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel"
aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Запис на прийом</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
      </div>
      <div class="modal-body">
        <form action="#" method='POST'>
          <div class='row justify-content-center'>
            <div class='col-12 pb-2'>
              <select class="form-select" aria-label="Default select example" name='service'>
                <option value="1">Консультація</option>
                <option value="2" id='option-selected'>Базовий</option>
                <option value="3" id='option-selected'>Стандарт</option>
                <option value="4" id='option-selected'>Преміум</option>
                <option value="5" id='option-selected'>Платина</option>
              </select>
            </div>
            <div class='col-12 pb-2'>
              <input type="text" name='name' class="form-control" placeholder="Ім'я" required>
            </div>
            <div class='col-12 pb-4'>
              <input type="tel" name='phone_number' class="form-control" placeholder="Телефонний
номер" required>
            </div>
            <div class='col-auto pb-3'>
              <button type="submit" class="btn btn-primary" >Записатися</button>
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>
</div>

<section class="ftco-section bg-light">
  <div class="container">
    <div class="row justify-content-center mb-5 pb-2">
      <div class="col-md-8 text-center heading-section ">

```

```

    <span class="subheading">Блог</span>
    <h2 class="mb-4">Останні блоги</h2>
  </div>
</div>
<div class="row">
  <div class="col-md-4 ">
    <div class="blog-entry">
      <a href="exampleBlog.html" class="block-20 d-flex align-items-end " style="background-
image: url('static/images/image_1.jpg');">
        <div class="meta-date text-center p-2">
          <span class="day">10</span>
          <span class="mos">січня</span>
          <span class="yr">2021</span>
        </div>
      </a>
      <div class="text bg-white p-4">
        <h3 class="heading"><a href="exampleBlog.html">Види коронок на зуби</a></h3>
        <p>Коронка на зуб - це дуже популярний спосіб непрямой реставрації. У цій статті ми
розповімо, які бувають і як виглядають коронки на зубах, яке значення, які переваги має цей вид
реставрації для пацієнтів.</p>
        <div class="d-flex align-items-center mt-4">
          <p class="mb-0"><a href="exampleBlog.html" class="btn btn-primary">Прочитати
більше</a></p>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="col-md-4 ">
  <div class="blog-entry">
    <a href="#" class="block-20 d-flex align-items-end " style="background-image:
url('static/images/image_2.jpg');">
      <div class="meta-date text-center p-2">
        <span class="day">22</span>
        <span class="mos">вересня</span>
        <span class="yr">2021</span>
      </div>
    </a>
    <div class="text bg-white p-4">
      <h3 class="heading"><a href="#">Гранульома зуба</a></h3>
      <p>По суті, гранульома являє собою капсулу з гнійним вмістом. Її формування -
результат імунної реакції організму на інфекцію. Седативні препарати стабілізують психоемоційний
стан пацієнта.</p>
      <div class="d-flex align-items-center mt-4">
        <p class="mb-0"><a href="#" class="btn btn-primary">Прочитати більше</a></p>
      </div>
    </div>
  </div>
</div>
</div>
<div class="col-md-4 ">

```







```
// selected service

let select = document.querySelectorAll('#option-selected');

function selectedService(num) {
  for(let i of select) {
    let value = i.getAttribute('value');
    if(num == value) {
      i.selected = true;
    }
  }
}

//scroll to form

let scrollTo = document.querySelector('#consultFrom')
let btn = document.querySelector('#btn-anim');

function btnScroll() {
  scrollTo.scrollIntoView({
    behavior: 'smooth'
  })
};

try {
  btn.addEventListener('click', btnScroll);
}
catch {}

//alerts animation

let alertNode = document.querySelector('.alert');

function animateBtn() {
  alertNode.style = 'visibility: visible; opacity: 1;';
  setTimeout(() => {
    alertNode.style = 'visibility: hidden;';
  });
}
```

```
    }, 2500)
  };
```

## Style.scss

```
@import 'bootstrap/bootstrap';
@import 'bootstrap/variables';
```

```
$font-primary: 'Poppins', Arial, sans-serif;
```

```
$white: #fff;
$black: #000000;
```

```
$primary: #46b7de;
$secondary: #a8c337;
```

```
@mixin border-radius($radius) {
  -webkit-border-radius: $radius;
  -moz-border-radius: $radius;
  -ms-border-radius: $radius;
  border-radius: $radius;
}
```

```
@mixin transition($transition) {
  -moz-transition: all $transition ease;
  -o-transition: all $transition ease;
  -webkit-transition: all $transition ease;
  -ms-transition: all $transition ease;
  transition: all $transition ease;
}
```

```
@mixin gradient-background(){
  background: rgba(70,184,222,1);
  background: -moz-linear-gradient(45deg, rgba(70,184,222,1) 0%, rgba(70,184,222,1) 42%, rgba(255,255,255,1)
100%);
  background: -webkit-gradient(left bottom, right top, color-stop(0%, rgba(70,184,222,1)), color-stop(42%,
rgba(70,184,222,1)), color-stop(100%, rgba(255,255,255,1)));
  background: -webkit-linear-gradient(45deg, rgba(70,184,222,1) 0%, rgba(70,184,222,1) 42%, rgba(255,255,255,1)
100%);
  background: -o-linear-gradient(45deg, rgba(70,184,222,1) 0%, rgba(70,184,222,1) 42%, rgba(255,255,255,1) 100%);
```

```

background: -ms-linear-gradient(45deg, rgba(70,184,222,1) 0%, rgba(70,184,222,1) 42%, rgba(255,255,255,1)
100%);
background: linear-gradient(45deg, rgba(70,184,222,1) 0%, rgba(70,184,222,1) 42%, rgba(255,255,255,1) 100%);
filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#46b8de', endColorstr='#ffffff', GradientType=1
);
}

```

```

body {
font-family: $font-primary;
background: $white;
font-size: 15px;
line-height: 1.8;
font-weight: 400;
color: rgba(0,0,0,.6);
&.menu-show {
overflow: hidden;
position: fixed;
height: 100%;
width: 100%;
}
}
a {
color: $primary;
&:hover, &:focus {
text-decoration: none;
color: $primary;
outline: none !important;
}
}
h1, h2, h3, h4, h5,
.h1, .h2, .h3, .h4, .h5 {
line-height: 1.5;
color: rgba(0,0,0,.8);
font-weight: 400;
}

```

```

.text-primary {
color: $primary!important;
}

```

```
//admin button
```

```
.admin-btn {
```

```

position: fixed;
top: 2vh;
right: 1.5vw;
z-index: 35;
.icon-gear {
  color: #333030 !important;
}
}

// alert

.alert {
  width: 20vw !important;
  text-align: center;
  z-index: 100;
  top: 1vh;
  position: fixed !important;
  visibility: hidden;
  opacity: 0;
  transition: .55s opacity, .55s visibility;
  @include media-breakpoint-down(md){
    width: 100vw !important;
  }
  @include media-breakpoint-up(lg){
    left: 40%;
    right: 40%;
  }
}

.topper{
  font-size: 15px;
  width: 100%;
  .icon{
    width: 50px;
    height: 50px;
    background: $white;
    border: 1px solid rgba(0,0,0,.05);
    @include border-radius(50%);
    span{
      color: $primary;
    }
  }
  .text{
    width: calc(100% - 50px);

```

```

p{
  margin-bottom: 0;
  font-size: 12px;
}
.con{
  margin-bottom: 0;
  font-size: 12px;
  font-weight: 300;
  span{
    font-weight: 600;
    font-size: 13px;
    color: $black;
    &:first-child{
      color: $primary;
    }
  }
}
.hr{
  span{
    color: $black;
    font-weight: 600;
    font-size: 13px;
  }
}
.time{
  span{
    &:last-child{
      font-weight: 600;
      color: $primary;
    }
  }
}
}
.ftco-navbar-light {
  background: transparent !important;
  z-index: 3;
  padding: 0;
  @include media-breakpoint-down(md) {
    background: $black!important;
    position: relative;
    top: 0;
    padding: 10px 15px;
  }
}
.navbar-brand {
  color: $black;
  &:hover, &:focus{
    color: $black;
  }
}

```

```

@include media-breakpoint-down(md){
  color: $white;
}
}

.navbar-nav {
  @include media-breakpoint-down(md){
    padding-bottom: 10px;
  }
  > .nav-item {
    > .nav-link {
      font-size: 15px;
      padding-top: 2rem;
      padding-bottom: 2rem;
      padding-left: 20px;
      padding-right: 20px;
      font-weight: 400;
      color: $black;
      letter-spacing: 1px;
      text-transform: uppercase;
      &:hover {
        color: $black;
      }
    }
    opacity: 1!important;
    @include media-breakpoint-down(md){
      padding-left: 0;
      padding-right: 0;
      padding-top: .9rem;
      padding-bottom: .9rem;
      color: rgba(255,255,255,.7);
      &:hover{
        color: $white;
      }
    }
  }
}

.dropdown-menu{
  border: none;
  background: $white;
  -webkit-box-shadow: 0px 10px 34px -20px rgba(0,0,0,0.41);
  -moz-box-shadow: 0px 10px 34px -20px rgba(0,0,0,0.41);
  box-shadow: 0px 10px 34px -20px rgba(0,0,0,0.41);
  .dropdown-item{
    font-size: 14px;
    &:hover, &:focus{
      background: transparent;
      color: $black;
    }
  }
}

```

```

    }
  }
}

&.ftco-seperator {
  position: relative;
  margin-left: 20px;
  padding-left: 20px;
  @include media-breakpoint-down(md) {
    padding-left: 0;
    margin-left: 0;
  }
  &:before {
    position: absolute;
    content: "";
    top: 10px;
    bottom: 10px;
    left: 0;
    width: 2px;
    background: rgba($white, .05);
    @include media-breakpoint-down(md) {
      display: none;
    }
  }
}

&.cta {
  > a {
    color: $black;
    @include media-breakpoint-down(sm){
      padding-left: 15px;
      padding-right: 15px;
    }
    @include media-breakpoint-down(md){
      color: $white;
      background: $primary;
    }
  }
}

&.active {
  > a {
    color: $primary;
    @include media-breakpoint-down(md){
      color: $white;
    }
  }
}
}

```

```

}

.navbar-toggler {
  border: none;
  color: rgba(255,255,255,.5)!important;
  cursor: pointer;
  padding-right: 0;
  text-transform: uppercase;
  font-size: 16px;
  letter-spacing: .1em;
  &:focus{
    outline: none !important;
  }
}

&.scrolled {
  position: fixed;
  right: 0;
  left: 0;
  top: 0;
  margin-top: -130px;
  background: $white!important;
  box-shadow: 0 0 10px 0 rgba(0,0,0,.1);
  .nav-item {
    &.active {
      > a {
        color: $primary!important;
      }
    }
  }
  &.cta {
    > a {
      color: $white !important;
      background: $primary;
      border: none !important;

      span {
        display: inline-block;
        color: $white !important;
      }
    }
  }
  &.cta-colored {
    span {
      border-color: $primary;
    }
  }
}
}

```



```
.navbar-nav {  
  @include media-breakpoint-down(md) {  
    background: none;  
    border-radius: 0px;  
    padding-left: 0rem!important;  
    padding-right: 0rem!important;  
  }  
}
```

```
.navbar-nav {  
  @include media-breakpoint-down(sm) {  
    background: none;  
    padding-left: 0!important;  
    padding-right: 0!important;  
  }  
}
```

```
.navbar-toggler {  
  border: none;  
  color: rgba(0,0,0,.5)!important;  
  border-color: rgba(0,0,0,.5)!important;  
  cursor: pointer;  
  padding-right: 0;  
  text-transform: uppercase;  
  font-size: 16px;  
  letter-spacing: .1em;  
  
}
```

```
.nav-link {  
  padding-top: .9rem!important;  
  padding-bottom: .9rem!important;  
  color: $black!important;  
  &.active {  
    color: $primary!important;  
  }  
}  
  
&.awake {  
  margin-top: 0px;  
  transition: .3s all ease-out;  
}  
  
&.sleep {  
  transition: .3s all ease-out;  
}
```

```
.navbar-brand {
```

```

    color: $black;
  }
}
}

```

```

.navbar-brand {
  font-weight: 700;
  font-size: 34px;
  padding: 0;
  line-height: 1;
  color: $black;
  span{
    font-size: 14px;
    font-weight: 500;
    display: block;
    color: rgba(0,0,0,.2);
    text-align: left;
  }
}

```

```

.hero-wrap{
  width: 100%;
  height: 100%;
  position: inherit;
  background-size: cover;
  background-repeat: no-repeat;
  background-position: top center;
  position: relative;
  @include media-breakpoint-down(md){
    background-position: top center !important;
  }
}
.overlay{
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  content: "";
  opacity: .5;
  background: $primary;
}
&.hero-wrap-2{
  height: 300px;
  .slider-text{
    height: 300px;
  }
}

```



```

    position: absolute;
  }
h1 {
  font-size: 50px;
  color: $black;
  line-height: 1.2;
  font-weight: 300;
  span{
    display: block;
    color: $black;
  }
  @include media-breakpoint-down(md) {
    font-size: 30px;
  }
}
.subheading{
  font-size: 18px;
  font-weight: 400;
  color: rgba(0,0,0,.8);
}
p {
  font-size: 18px;
  line-height: 1.5;
  font-weight: 300;
  color: rgba(0,0,0,.8);
  &.sub-p{
    font-weight: 400;
  }
  strong{
    font-weight: 700;
    a{
      color: $white;
    }
  }
}
//BREADCRUMBS
.breadcrumbs{
  text-transform: uppercase;
  font-size: 13px;
  letter-spacing: 1px;
  span{
    border-bottom: 2px solid rgba(255,255,255,.1);
    a{
      color: $white;
    }
  }
}
}
}

```

```

}

@include media-breakpoint-up(md) {
  .text-carousel {
    padding-right: 15%;
    padding-left: 50%;
    text-align: start;
  }
}

```

```

.container{
  max-width: 1230px;
}

```

```

.bg-light {
  background: #fafafa !important;
}

```

```

.bg-white{
  background: $white !important;
}

```

```

.bg-primary{
  background: $primary;
}

```

```

.ftc-no-pb{
  padding-bottom: 0 !important;
}

```

//BUTTON

```

.btn {
  cursor: pointer;
  @include border-radius(40px);
  box-shadow: none !important;
  &:hover, &:active, &:focus {
    outline: none;
  }
  &.btn-primary {
    background: $primary;
  }
}

```

```

border: 1px solid $primary;
color: $white;
&:hover {
  border: 1px solid $primary;
  background: transparent;
  color :$primary;
}
&.btn-outline-primary {
  border: 1px solid $primary;
  background: transparent;
  color :$primary;
  &:hover {
    border: 1px solid transparent;
    background: $primary;
    color :$white;
  }
}
}
}
&.btn-white {
  background: $white;
  border: 1px solid $white;
  color: $primary;
  &:hover, &:focus, &:active {
    border: 1px solid $white;
    color :$primary;
  }
}
&.btn-outline-white {
  border-color: rgba($white, .8);
  background: none;
  @include border-radius(30px);
  border-width: 1px;
  color: $white;
  &:hover, &:focus, &:active {
    background: $white;
    border-color: $white;
    color: $primary;
  }
}
}
}
&.btn-secondary {
  background: $secondary;
  border: 1px solid $secondary;
  color: $white;
  &:hover, &:focus, &:active {
    border: 1px solid $secondary;
    color :$white;
  }
}
}
}

```

```
.btn-custom{
  font-weight: 500;
  color: darken($primary,10%);
  border-bottom: 4px solid lighten($primary,4%);
}
```

## //ABOUT

```
.img-2{
  background-position: top center !important;
  @include media-breakpoint-down(sm){
    height: 300px;
    margin-bottom: 40px;
  }
}

.wrap-about{
  position: relative;
  @include media-breakpoint-up(md){
    padding-left: 6em;
  }
}

.founder{
  .img{
    width: 70px;
    height: 70px;
    @include border-radius(50%);
  }
  .text{
    width: calc(100%-70px);
  }
  h3{
    font-size: 24px;
  }
}
}
```

## //SERVICES

```
.ftco-services{
  .services{
    display: block;
    width: 100%;
    position: relative;
    .img{
      height: 250px;
    }
  }
}
```

```

.icon{
  line-height: 1.3;
  position: relative;
  width: 80px;
  height: 80px;
  border: 1px solid rgba(0,0,0,.05);
  margin: 0 auto;
  @include border-radius(50%);
  span{
    font-size:50px;
    color: $primary;
  }
}
.media-body{
  h3{
    font-size: 18px;
    color: $black;
  }
}
}
}

.intro{
  position: relative;
  background-size: cover;
  background-repeat: no-repeat;
  background-position: top center;
  @include media-breakpoint-down(md){
    background-position: top center !important;
  }
}

.ftco-intro{
  padding: 4em 0;
  width: 100%;
  height: 100%;
  position: inherit;
  background-size: cover;
  background-repeat: no-repeat;
  background-position: top center;
  position: relative;
  color:rgba(255,255,255,.8);
  @include media-breakpoint-down(md){
    background-position: center center !important;
  }
}
h2{
  color: $white;
  font-size: 36px;
}
}

```



```
.overlay{
  position: absolute;
  // width: 50%;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  content: "";
  opacity: .8;
  background: $primary;
}
}

//STAFF
.staff{
  overflow: hidden;
  margin-bottom: 30px;
  .img-wrap{
    height: 300px;
  }
  .img{
    // height: 300px;
    width: 100%;
    display: block;
    background-position: top center;
  }
  .text{
    position: relative;
    background: $white;
  }
  h3{
    font-size: 20px;
    font-weight: 400;
    margin-bottom: 0;
  }
  .position{
    text-transform: uppercase;
    color: lighten($black,70%);
    display: block;
    letter-spacing: 1px;
  }
  .faded{
    opacity: 1;
  }
}
```

```

}
.ftco-social{
  position: absolute;
  top: -50px;
  left: 0;
  right: 0;
  padding: 15px 10px;
  background: transparent;
  opacity: 1;

  li{
    width: 30px;
    height: 30px;
    position: relative;
    a{
      color: $secondary;
      width: 30px;
      height: 30px;
      background: $white;
      @include border-radius(50%);
    }
  }
}
}
&:hover, &:focus{
  .ftco-social{
    opacity: 1;
    top: -60px;
  }
  .img{
    margin-top: -10px;
  }
  .text{
    h3{
      color: $primary;
    }
  }
}
}
}

```

```

.ftco-social{
  padding: 0;
  li{
    list-style: none;
    margin-right: 10px;
    display: inline-block;
  }
}

```

```
//PRICING
.pricing-entry{
  border: 1px solid lighten($primary,30%);
  position: relative;

  @include media-breakpoint-down(sm){
    margin-bottom: 4em;
  }
  >div{
    padding: 20px;
    p{
      margin-bottom: 0;
    }
  }
  h3{
    font-size: 16px;
    font-weight: 400;
  }
  p{
    .price{
      font-weight: 300;
      font-size: 28px;
      color: $primary;
    }

    .per{
      font-size: 12px;
    }
  }
  ul{
    margin: 0;
    padding: 0;
    li{
      list-style: none;
      padding: 10px 20px;
      &:nth-child(odd){
        background: lighten($black,98%);
      }
    }
  }
  .button{
    position: absolute;
    bottom: -26px;
    left: 0;
    right: 0;
    margin-bottom: 0;
  }
}
```

```

.btn{
  @include border-radius(30px);

}
}
&:hover, &:focus{
  .btn{
    background: $secondary !important;
    border: 1px solid $secondary !important;
    color: $white !important;
  }
  .price{
    color: $secondary;
  }
}
}
}

```

```

#exampleModal {
  margin-top: 20vh;
}

```

```

//BLOG ENTRY

```

```

.block-20 {
  overflow: hidden;
  background-size: cover;
  background-repeat: no-repeat;
  background-position: center center;
  height: 275px;
  position: relative;
  display: block;
}
.blog-entry{
  @include media-breakpoint-up(md){
    margin-bottom: 30px;
  }
  @include media-breakpoint-down(sm){
    margin-bottom: 30px;
  }
  .text {
    position: relative;
    border-top: 0;
    border-radius: 2px;
    .heading {
      font-size: 20px;
      margin-bottom: 16px;
    }
  }
}

```



```

//PAGINATION
.block-27 {
  ul {
    padding: 0;
    margin: 0;
    li {
      display: inline-block;
      margin-bottom: 4px;
      font-weight: 400;
      a,span {
        border: 1px solid $primary;
        text-align: center;
        display: inline-block;
        width: 40px;
        height: 40px;
        line-height: 40px;
        border-radius: 50%;
      }
      &.active {
        a, span {
          background: $primary;
          color: $white;
          border: 1px solid transparent;
        }
      }
    }
  }
}

```

```

// USEFUL CODE
.aside-stretch{
  background: $white;
  &:after{
    position: absolute;
    top: 0;
    left: 100%;
    bottom: 0;
    content: "";
    width: 360%;
    background: $white;
    // background: #333644;
  }
}

```

```
@include media-breakpoint-down(sm){
  background: transparent;
  &:after{
    background: transparent;
    display: none;
  }
}
}
```

```
.form-control {
  height: 52px!important;
  background: $white!important;
  color: $black!important;
  font-size: 18px;
  border-radius: 0px;
  box-shadow: none!important;
  border: 1px solid rgba(0,0,0,.1);
  &:focus, &:active {
    border-color: $black;
  }
}
textarea.form-control {
  height: inherit!important;
}
```

```
.bg-primary {
  background: $primary!important;
}
.bg-secondary {
  background: $secondary!important;
}
```

```
.about-author{
  img{
  }
  .desc{
    h3{
      font-size: 24px;
    }
  }
  .bio{
  }
}
```

```
}

.ftco-section {
  padding: 7em 0;
  position: relative;
}

.ftco-no-pt{
  padding-top: 0;
}
.ftco-no-pb{
  padding-bottom: 0;
}

.ftco-bg-dark {
  background: #3c312e;
}

.ftco-footer {
  font-size: 14px;
  background: $primary;
  padding: 3em 0 0 0;
  .ftco-footer-widget {
    width: 100%;
    display: block;
    h2 {
      font-weight: normal;
      color: $white;
      margin-bottom: 40px;
      font-size: 20px;
      font-weight: 400;
    }
  }
  .logo{
    color: $white;
  }
  ul{
    li{
      a{
        span{
          color: $white;
        }
      }
    }
  }
}
```



```

}
.btn-primary{
  background: $white !important;
  border: 2px solid $white !important;
  &:hover{
    background: $white;
    border: 2px solid $white !important;
  }
}
}
}
p {
  color: rgba($white, .7);
}
a {
  color: rgba($white, .7);
  &:hover {
    color: $white;
  }
}
.ftco-heading-2 {
  font-size: 17px;
  font-weight: 400;
  color: $black;
}

.categories{
  li{
    border-bottom: 1px solid rgba(255,255,255,.1);
    a{
      span{
        color: rgba(255,255,255,.5) !important;
      }
    }
  }
}
.subscribe-form{
  .form-group {
    position: relative;
    margin-bottom: 0;
    @include border-radius(0);
    input {
      @include border-radius(2px);
      width: 100%;
      background: rgba(255,255,255,.1) !important;
      border: none;
      color: rgba(255,255,255,.7) !important;
      font-size: 16px;
      &::-webkit-input-placeholder { /* Chrome/Opera/Safari */

```

```

    color: rgba(255,255,255,.7) !important;
  }
  &::-moz-placeholder { /* Firefox 19+ */
    color: rgba(255,255,255,.7) !important;
  }
  &:-ms-input-placeholder { /* IE 0+ */
    color: rgba(255,255,255,.7) !important;
  }
  &::-moz-placeholder { /* Firefox 18- */
    color: rgba(255,255,255,.7) !important;
  }
}
.submit{
  color: $white !important;
  font-size: 16px;
  background: $secondary !important;
  border: none !important;
  @include border-radius(2px);

  &:hover{
    cursor: pointer;
    background: $secondary !important;
  }
}
}
.icon {
  position: absolute;
  top: 50%;
  right: 20px;
  transform: translateY(-50%);
  color: rgba(255,255,255,.8);
}
}
.block-21 {
  .text {
    .heading {
      font-weight: 400;
      font-size: 16px;
      a {
        color: rgba(255,255,255,.9);
        &:hover, &:active, &:focus {
          color: $white;
        }
      }
    }
  }
  .meta {
    > div {
      display: inline-block;
      font-size: 12px;

```



```
    position: relative;
  }
  .text{
    width: calc(100% - 120px);
    position: relative;
  }
  .quote{
    position: absolute;
    top: 0px;
    left: 0;
    i{
      font-size: 24px;
      color: $primary;
    }
  }
  .name{
    font-weight: 400;
    font-size: 18px;
    margin-bottom: 0;
    color: $black;
  }
  .position{
    font-size: 14px;
    color: $primary;
  }
}

.about-image{
  @include media-breakpoint-down(sm){
    height: 400px;
    margin-bottom: 30px;
  }
}

.carousel-indicators button {
  background-color: $primary !important;
  width: 14px !important;
  height: 14px !important;
  border-radius: 50%;
}

#section-counter{
  position: relative;
```

```

z-index: 0;
&:after{
  position: absolute;
  top: 0;
  left: 0;
  bottom: 0;
  right: 0;
  content: "";
  z-index: -1;
  opacity: 1;
  background: $primary;
}
}
.ftco-counter {
  overflow-x: hidden;
  position: relative;
  @include media-breakpoint-down(lg){
    background-position: center center !important;
  }
  .container{
  }
  .bg-counter{
    position: relative;
    background: $white;
    h3.vr{
      writing-mode: vertical-lr;
      transform: rotate(180deg);
      text-orientation: sideways;
      text-align:center;
      text-orientation: sideways;
      margin: 0rem;
      position: absolute;
      top: 0;
      right: -20px;
      height: 100%;
      text-transform: uppercase;
      font-weight: 500;
      letter-spacing: 4px;
      font-size: 16px;
      z-index: 1;
      @include media-breakpoint-down(lg){
        opacity: 0;
      }
    }
  }
}
}
.icon{
  span{
  }
}
}

```

```

.block-18{
  display: block;
  width: 100%;
  text-align: center;
  margin-bottom: 30px;
}
.text{
  strong.number{
    font-weight: 800;
    font-size: 44px;
    color: $primary;
    display: block;
  }
  span{
    display: block;
    font-size: 16px;
    color: rgba(0,0,0,.8);
  }
}
.counter-wrap{
  @include media-breakpoint-down(sm){
    margin-bottom: 20px;
  }
}
.ftco-number {
  display: block;
  font-size: 72px;
  font-weight: bold;
  color: $primary;
}
.ftco-label {
  font-size: 12px;
  text-transform: uppercase;
  letter-spacing: .1em;
}
}

.consultation{
  background: $primary;
  color: rgba(255,255,255,.9);
  padding-top: 4em;
  padding-bottom: 4em;
  &.consul{
    background: darken($primary,10%);
  }
  h3{
    font-size: 20px;
    color: $white;
  }
}

```

```

    font-weight: 500;
  }
  .btn-custom{
    font-size: 12px;
    text-transform: uppercase;
    color: $white;
    font-weight: 600;
  }
}

//APPOINTMENT
.appointment-form{
  .form-group{
    position: relative;
    width: 100%;
    display: block;
    .icon {
      position: absolute;
      top: 50% !important;
      right: 0;
      font-size: 16px;
      transform: translateY(-50%);
      span{
        color: rgba(255,255,255,1) !important;
      }
    }
  }
  .select-wrap, .input-wrap {
    position: relative;
    select {
      appearance: none;
    }
  }
  .btn{
    width: 100%;
    display: block !important;
  }
}
.form-control {
  border: transparent !important;
  border-bottom: 1px solid rgba(255,255,255,.2) !important;
  height: 40px!important;
  padding-left: 0;
  padding-right: 0;
  background: transparent!important;
  color: rgba(255,255,255,.9)!important;
  font-size: 13px;
}

```

```

border-radius: 0px;
box-shadow: none!important;
&::-webkit-input-placeholder { /* Chrome/Opera/Safari */
  color: rgba(255,255,255,.9);
}
&::-moz-placeholder { /* Firefox 19+ */
  color: rgba(255,255,255,.9);
}
&:-ms-input-placeholder { /* IE 10+ */
  color: rgba(255,255,255,.9);
}
&:-moz-placeholder { /* Firefox 18- */
  color: rgba(255,255,255,.9);
}
&:focus, &:active {
  border-color: $white;
}
}
}
textarea.form-control {
  height: inherit!important;
}
}

```

```

.block-23 {
  ul {
    padding: 0;
    li {

      &, > a {
        display: table;
        line-height: 1.5;
        margin-bottom: 15px;
      }
      span{
        color: rgba($white, .7);
      }
      .icon, .text {
        display: table-cell;
        vertical-align: top;
      }
      .icon {
        width: 40px;
        font-size: 18px;
        padding-top: 2px;
        color: rgba($white, 1);
      }
    }
  }
}

```



```

    }
  }
}

#### .block-4
.contact-section {
  .contact-info{
    p{
      a{
        color: lighten($black,10%);
      }
      span{}
    }
  }
  .box{
    width: 100%;
    display: block;
  }
}
.block-9 {
  .form-control {
    outline: none!important;
    box-shadow: none!important;
    font-size: 15px;
  }
  #map {
  }
}

```

```

#### .block-21
.block-21 {
  .blog-img{
    display: block;
    height: 80px;
    width: 80px;
  }
  .text {
    width: calc(100% - 100px);
    .heading {
      font-size: 18px;
      font-weight: 400;
      a {
        color: $black;
        &:hover, &:active, &:focus {
          color: $primary;
        }
      }
    }
  }
}

```

```

    }
  }
}
.meta {
  > div {
    display: inline-block;
    font-size: 12px;
    margin-right: 5px;
    a {
      color: lighten($black, 50%);
    }
  }
}
}
}
}

.subscribe-form{
  .form-group {
    position: relative;
    margin-bottom: 0;
    @include border-radius(0);
    input {
      background: transparent !important;
      border: 1px solid transparent;
      color: rgba(255,255,255,.7) !important;
      font-size: 16px;
      @include border-radius(0);
      &::-webkit-input-placeholder { /* Chrome/Opera/Safari */
        color: rgba(255,255,255,.7) !important;
      }
      &::-moz-placeholder { /* Firefox 19+ */
        color: rgba(255,255,255,.7) !important;
      }
      &:-ms-input-placeholder { /* IE 10+ */
        color: rgba(255,255,255,.7) !important;
      }
      &:-moz-placeholder { /* Firefox 18- */
        color: rgba(255,255,255,.7) !important;
      }
    }
  }
  .form-control{
    border: 1px solid rgba(255,255,255,.4);
  }
  .submit{
    color: $black !important;
    display: block;
  }
}

```

```

width: 100%;
height: 52px;
font-size: 16px;
background: $white !important;
@include border-radius(0);
&:hover{
  cursor: pointer;
}
}
}
.icon {
  position: absolute;
  top: 50%;
  right: 20px;
  transform: translateY(-50%);
  color: rgba(255,255,255,.8);
}
}

.categories, .sidelink {
padding: 0;
li {
  position: relative;
  margin-bottom: 10px;
  padding-bottom: 10px;
  border-bottom: 1px solid gray('300');
  list-style: none;
  &:last-child {
    margin-bottom: 0;
    border-bottom: none;
    padding-bottom: 0;
  }
  a {
    display: block;
    span {
      position: absolute;
      right: 0;
      top: 0;
      color: #ccc;
    }
  }
  &.active {
    a {
      color: $black;
      font-style: italic;
    }
  }
}

```

```

    }
  }
}

// Loader

#ftco-loader {
  position: fixed;
  width: 96px;
  height: 96px;
  left: 50%;
  top: 50%;
  transform: translate(-50%, -50%);
  background-color: rgba(255,255,255,0.9);
  box-shadow: 0px 24px 64px rgba(0,0,0,0.24);
  border-radius:16px;
  opacity: 0;
  visibility: hidden;
  transition: opacity .2s ease-out, visibility 0s linear .2s;
  z-index:1000;
}

#ftco-loader.fullscreen {
  padding: 0;
  left: 0;
  top: 0;
  width: 100%;
  height: 100%;
  transform: none;
  background-color: #fff;
  border-radius: 0;
  box-shadow: none;
}

#ftco-loader.show {
  transition: opacity .4s ease-out, visibility 0s linear 0s;
  visibility: visible;
  opacity: 1;
}

#ftco-loader.circular {
  animation: loader-rotate 2s linear infinite;
  position: absolute;
  left: calc(50% - 24px);

```

```
top: calc(50% - 24px);
display: block;
transform: rotate(0deg);
}

#ftco-loader .path {
stroke-dasharray: 1, 200;
stroke-dashoffset: 0;
animation: loader-dash 1.5s ease-in-out infinite;
stroke-linecap: round;
}

@keyframes loader-rotate {
100% {
transform: rotate(360deg);
}
}

@keyframes loader-dash {
0% {
stroke-dasharray: 1, 200;
stroke-dashoffset: 0;
}
50% {
stroke-dasharray: 89, 200;
stroke-dashoffset: -35px;
}
100% {
stroke-dasharray: 89, 200;
stroke-dashoffset: -136px;
}
}
```

## ДОДАТОК В

## ВІДГУК

на комплексну кваліфікаційну роботу бакалавра на тему:

**«Проектування інформаційного веб-сайту установи**

**«Стоматологічна клініка «VlaDent»**

**студента групи 126-17-1 Юхимчука Дениса Петровича.**

Метою даної кваліфікаційної роботи є проектування та розробка інформаційного веб-сайту лікувальної установи «Стоматологічна клініка «VlaDent».

Дана тема актуальна тому, що в умовах швидкого розвитку цифрових технологій, розповсюдження точок доступу до мережі Інтернет, а також мобільних технологій та сервісів, надзвичайно розширюються можливості інформування потенційних та поточних пацієнтів про сучасні лікувальні напрями і функціональні спроможності стоматологічних закладів на прикладі установи «Стоматологічна клініка «VlaDent» .

Тема кваліфікаційної роботи безпосередньо пов'язана з об'єктом діяльності фахівця галузі знань 12 «Інформаційні технології» спеціальності 126 «Інформаційні системи та технології» – розробка та експлуатація різнопланових інформаційних систем і пов'язаних з ними програмних засобів.

Завдання кваліфікаційної роботи (проектування та розробка інформаційного веб-сайту лікувальної установи «Стоматологічна клініка «VlaDent») віднесені в освітньо-професійній програмі підготовки випускника відповідної спеціальності до класу евристичних, рішення яких заснована на знаково-розумових уміннях фахівця.

Оригінальність технологічних рішень полягає в використанні потужних та багатофункціональних інструментальних технологій: фреймворків Django

та Bootstrap, мови Python, а також можливостей безкоштовного хмарного хостингу Heroku для розгортання компонентів інформаційного веб-сайту.

Практичне значення результатів кваліфікаційної роботи полягає в створенні компоненту інформаційної системи – цифрового сервісу, що дозволяє розширити можливості надання стоматологічних послуг населенню, а також забезпечення користувачів якісною інформацією для виконання замовлень и планування можливих профілактичних заходів.

Оформлення графічних та текстових матеріалів пояснювальної записки кваліфікаційної роботи виконано на досить високому рівні і без відхилень від стандарту.

Ступінь самостійності виконання кваліфікаційної роботи повною мірою відповідає першому освітньо-кваліфікаційному рівню вищої освіти, тобто ступеню бакалавра.

В роботі досить повно виконано огляд сучасного стану ринку програмних додатків в сфері створення бізнес-рішень відповідного рівня.

Деякі дискусійні положення та несуттєві недоліки пов'язані з не досить чітким викладенням та описом структури роботи та дещо невисокою якістю ілюстративних матеріалів.

Незважаючи на вищевказані зауваження, кваліфікаційна робота в цілому заслуговує оцінки «відмінно», а її виконавець, студент Юхимчук Д.П., присвоєння йому відповідної кваліфікації.

**Керівник кваліфікаційної роботи,  
професор кафедри ІТКІ, д-р техн. наук**

**Г.М.  
Коротенко**

### Рецензія

на комплексну кваліфікаційну роботу бакалавра на тему:  
«Проектування інформаційного веб-сайту установи  
«Стоматологічна клініка «VlaDent»  
студента групи 126-17-1 Юхимчука Дениса Петровича.

Стрімке зростання обсягу інформаційних потоків змушує розробників інформаційних цифрових сервісів підвищувати культуру організації та подання відповідних описових матеріалів. Це особливо стосується питань, що відносяться до лікувальної галузі, а особливо – стоматологічної її складової.

В рецензованій кваліфікаційній роботі спроектовано та розроблено інформаційний веб-сайт лікувальної установи «Стоматологічна клініка «VlaDent».

Використовувані технології розробки даних програмних засобів обліку та контролю різноманітних процесів є евристичною компонентою і безпосередньо пов'язане з об'єктом діяльності фахівця спеціальності 126 «Інформаційні системи та технології» галузі знань 12 «Інформаційні технології».

Студент Д.П. Юхимчук досить добре розібрався в специфіці застосування різноманітних компонентів інструментальних технологій: фреймворків Django та Bootstrap для побудови відповідного веб-сайту.

До недоліків роботи слід віднести:

– недостатньо повно виконано огляд сучасного стану ринку програмних додатків в сфері створення сучасних бізнес-рішень;



– дещо спрощено підхід щодо реалізації функцій збору, аналізу та візуалізації результатів формування та обробки необхідного для обробки контенту.

Однак, розроблений інформаційний веб-сайт можна вважати закінченим етапом виконаної кваліфікаційної роботи.

На підставі вищевикладеного, можна зробити висновок, що представлені матеріали цілком відповідають вимогам, що пред'являються до кваліфікаційних робіт першого рівня вищої освіти, тобто ступеню бакалавра.

З огляду на весь спектр створених компонентів, що забезпечують формування даної кваліфікаційної роботи, в цілому вона заслуговує на оцінку «відмінно», а її виконавець, студент Юхимчук Д.П., присвоєння йому відповідної кваліфікації.

**Рецензент, доцент**  
**кафедри ПЗКС, канд. техн.**  
**наук**

**Л.М. Коротенко**

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

(посада, звання)

(підпис)

(прізвище,  
ініціали)

