

Київський національний торговельно-економічний університет

Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка веб-сайту для клінінгової фірми з використанням патерну MVC»

Студентки 4 курсу, 13 групи, спеціальності 122 «Комп'ютерні науки»	<i>підпис студента</i>	Алекперов Рустам Ельчин Огли
Науковий керівник Доктор технічних наук	<i>підпис керівника</i>	Краскевич Валерій Євгенович
Гарант освітньої програми кандидат технічних наук, доцент	<i>підпис керівника</i>	Демідов Павло Георгійович

Київ 2021

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра комп'ютерних наук та інформаційних систем

Спеціальність 122 «Комп'ютерні науки»

Зав. кафедри _____

Затверджую

Краскевич В.Є

« » грудня 2020р.

Завдання на випускню кваліфікаційну роботу (проект) студентці

Алекперов Рустам Ельчин Огли

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи (проекту)

«Розробка веб-сайту для клінінгової фірми з використанням патерну MVC»

Затверджена наказом ректора від « » грудня 2020 р. № 3780

2. Строк здачі студентом закінченої роботи 29 травня 2021 року

3. Цільова установка та вихідні дані до роботи

Мета роботи: обґрунтування та розробка web-сайту для клінінгової фірми на основі патерну MVC

Об'єкт дослідження: web-сайт клінінгової фірми.

Предмет дослідження: засоби розробки web-ресурсів

4. Перелік графічного матеріалу _____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Краскевич В.Є.	22.12.2020 р.	22.12.2020 р.
2	Краскевич В.Є.	22.12.2020 р.	22.12.2020 р.
3	Краскевич В.Є.	22.12.2020 р.	22.12.2020 р.

6. Зміст випускної кваліфікаційної роботи (проекту) (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ

1.1 Аналіз доступних патернів веб-розробки

РОЗДІЛ 2. Організація розробки web-сайту

2.1 Загальна концепція

РОЗДІЛ 3. Програмна реалізація

3.1 Реалізація бази даних

3.2. Реалізація проекту

ВИСНОВКИ

ДОДАТКИ

7. Календарний план виконання роботи

№ Пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	Вибір теми випускної кваліфікаційної роботи	01.10.2020	01.10.2020
2	Розробка та затвердження завдання на випускну кваліфікаційну роботу	15.12.2020	15.12.2020
3	Вступ	03.02.2021	
4	РОЗДІЛ 1. Обґрунтування доцільності розробки	28.02.2021	
5	РОЗДІЛ 2. Розробка програмного забезпечення моніторингу відвідування web-ресурсів	06.04.2021	
6	РОЗДІЛ 3. Програмна реалізація	12.05.2021	
7	Висновки	15.05.2021	
8	Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику	20.05.2021	
9	Попередній захист випускної кваліфікаційної роботи	26.05.2021	
11	Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи	27.05.2021	

12	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедру</i>	29.05.2021	
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	<i>За розкладом роботи ЕК</i>	

8. Дата видачі завдання «15» грудня 2020 р.

9. Керівник випускної кваліфікаційної роботи (проекту)

Краскевич В.Є.

(прізвище, ініціали, підпис)

10. Гарант освітньої програми

Демідов П.Г.

(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент-дипломник

Алекперов Р.Є.

(прізвище, ініціали, підпис)

12. Відгук керівника випускної кваліфікаційної роботи (проекту)

Керівник випускної кваліфікаційної роботи (проекту)

_____р.

(підпис, дата)

13. Висновок про випускну кваліфікаційну роботу (проект)

Випускна кваліфікаційна робота (проект) студента _____

(прізвище, ініціали)

може бути допущена до захисту в екзаменаційній комісії.

Гарант освітньої програми _____

(підпис, прізвище, ініціали)

Демідов П.Г.

Завідувач кафедри _____

(підпис, прізвище, ініціали)

Пурський О.І.

« _____ » _____ 2021 р.

Анотація

У випускній кваліфікаційній роботі здійснено розробку сайту з метою поліпшення сервісу, розширення доступності спектру послуг та збільшення прибутку клінінгової фірми. Представлена робота є прикладом роботи сайту-візитівки для підприємства. Теоретично обґрунтовано використання патерну розробки веб-ресурсів MVC Framework.

Ключові слова: MVC, Framework, веб-сайт.

Anotation

In the final qualifying work, the site was developed in order to improve the service, expand the availability of services and increase the profits of the cleaning company. The presented work is an example of a business card site for an enterprise. The use of the MVC Framework web resource development pattern is theoretically substantiated.

Keywords: MVC, Framework, website.

ЗМІСТ

ВСТУП

РОЗДІЛ 1. ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ	10
1.1 Аналіз доступних патернів веб-розробки	10
РОЗДІЛ 2. Організація розробки програмного забезпечення web-ресурсів	16
2.1 Загальна концепція	16
РОЗДІЛ 3. Програмна реалізація	20
3.1 Реалізація бази даних	20
3.2. Реалізація проекту	24
ВИСНОВКИ	29
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	31

ВСТУП

Сайт-візитка — невеликий сайт, що зазвичай складається з однієї (або декількох) веб-сторінок та містить основну інформацію про організацію, приватну особу, компанію, товари чи послуги, прайс-листи, контактні дані

Актуальність теми полягає в тому, що робота таких сайтів не потребує значних ресурсів для підтримки подібних рішень і дозволяє досить швидко додавати, оновлювати або видаляти необхідні дані з веб-сайту.

Практичне значення роботи: за допомогою веб-сайту підприємство охоплює значну частину потенційних клієнтів, дозволяє частково автоматизувати процес сервісних та консультаційних послуг, а також надавати всю необхідну інформацію користувачу в зручній формі.

Метою роботи є обґрунтування та розробка веб-сайту на основі патерну MVC. Для досягнення поставленої мети необхідно було вирішити наступні **завдання:**

- провести аналітичне дослідження доступних патернів для розробки;
- дослідити можливості фреймворків;
- розробити модель, контролер і вью;
- розробити веб-сайт для підприємства.

Об'єктом дослідження є розробка веб-сайту на основі патерну MVC.

Предметом дослідження являють собою засоби створення веб-сайту.

Методами дослідження є загальнонаукові підходи до наукового пізнання (системний, структурний, модельний, порівняльний). Застосовано аналіз, результати якого, дали можливість зробити висновки щодо специфіки та розвитку вебсайтів, як інформаційних систем та окреслити комплекс актуальних питань формування та використання ресурсів веб-сайту.

РОЗДІЛ 1. ОБґРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ

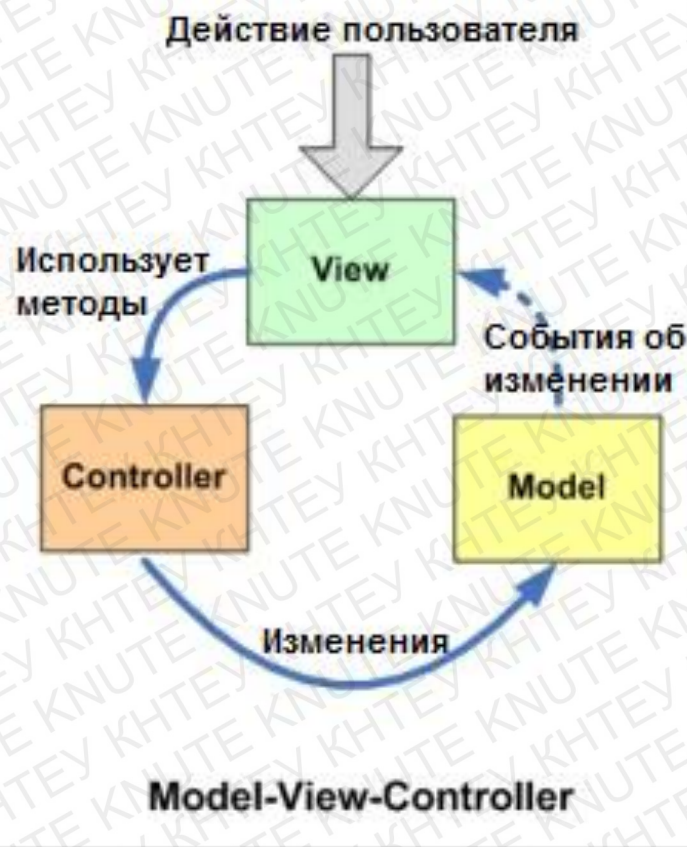
1.1 Аналіз доступних патернів веб-розробки

Фреймворки - це програмні продукти, які спрощують створення і підтримку технічно складних або навантажених проєктів. Фреймворк, як правило, містить тільки базові програмні модулі, а все специфічне для

проекту компоненти реалізуються розробником на їх основі. Тим самим досягається не тільки висока швидкість розробки, а й велика продуктивність і надійність рішень.

Наразі є декілька основних патернів, а саме: MVC, MVP и MVVM. Розглянемо кожен з них по черзі.

Model-View-Controller. MVC - це фундаментальний патерн, який знайшов застосування в багатьох технологіях, дав розвиток нових технологій і кожен день полегшує життя розробникам.



Вперше патерн MVC з'явився в мові SmallTalk. Розробники повинні були придумати архітектурне рішення, яке дозволяло б відокремити графічний інтерфейс від бізнес логіки, а бізнес логіку від даних. Таким чином, в класичному варіанті, MVC складається з трьох частин, які і дали йому назву. Розглянемо їх:

- **Модель**

Під Моделлю, зазвичай розуміється частина містить в собі функціональну бізнес-логіку програми. Модель повинна бути повністю незалежна від інших частин продукту. Модельний шар нічого не повинен знати про елементи дизайну, і яким чином він буде відображатися. Досягається результат, що дозволяє змінювати подання даних, то як вони відображаються, не чіпаючи саму Модель.

1. Модель володіє наступними ознаками:
2. Модель - це бізнес-логіка програми;
3. Модель володіє знаннями про себе саму і не знає про контролерах і уявленнях;
4. Для деяких проектів модель - це просто шар даних (DAO, база даних, XML-файл);
5. Для інших проектів модель - це менеджер бази даних, набір об'єктів або просто логіка додатка;

- Подання (View)

В обов'язки Уявлення входить відображення даних отриманих від Моделі. Однак, уявлення не може безпосередньо впливати на модель. Можна говорити, що уявлення отримує доступ «тільки на читання» до даних.

Подання володіє наступними ознаками:

1. У поданні реалізується відображення даних, які виходять від моделі будь-яким способом;
2. У деяких випадках, уявлення може мати код, який реалізує деяку бізнес-логіку.

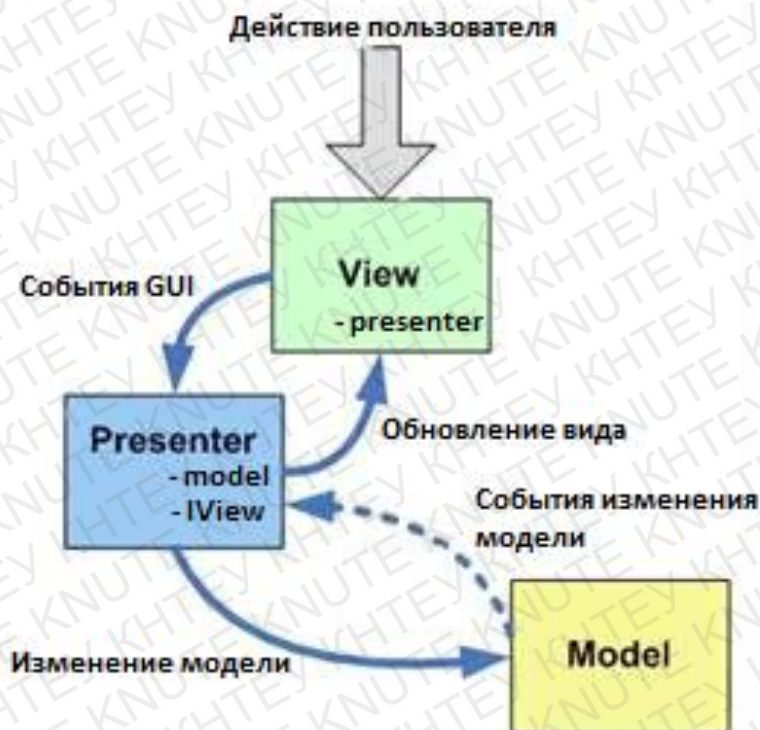
Приклади подання: HTML-сторінка, WPF форма, Windows Form.

- Контролер

Контролер - це компонент, який відповідає за зв'язок між model і view. Код компонента controller визначає, як сайт реагує на дії користувача. По суті, це мозок MVC-додатки.

Model-View-Presenter (MVP) - шаблон проектування, похідний від MVC, який використовується в основному для побудови призначеного для користувача інтерфейсу.

Елемент **Presenter** в даному шаблоні бере на себе функціональність посередника (аналогічно контролера в MVC) і відповідає за управління подіями призначеного для користувача інтерфейсу (наприклад, використання миші) так само, як в інших шаблонах зазвичай відповідає уявлення.



Model-View-Presenter

Даний підхід дозволяє створювати абстракцію уявлення. Для цього необхідно виділити інтерфейс уявлення з певним набором властивостей і методів. Презентер, в свою чергу, отримує посилання на реалізацію інтерфейсу, підписується на події вистави і за запитом змінює модель.

Ознаки презентера:

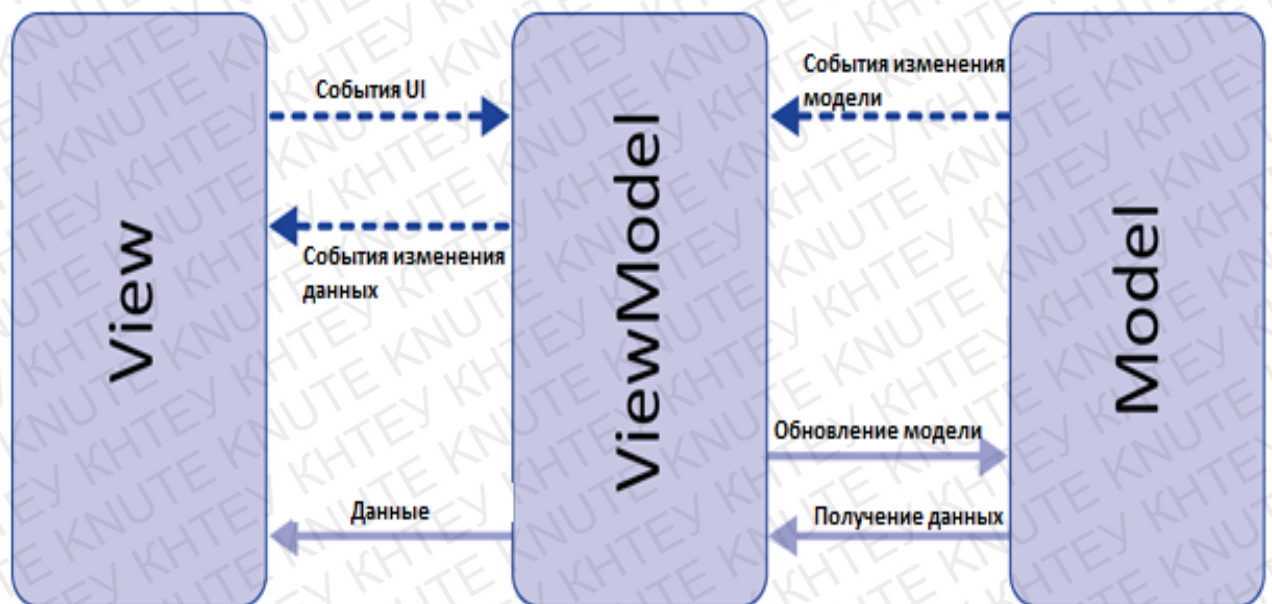
- Двостороння комунікація з поданням;
- Подання взаємодіє безпосередньо з презентером, шляхом виклику відповідних функцій або подій примірника презентера;
- Презентер взаємодіє з View шляхом використання спеціального інтерфейсу, реалізованого поданням;
- Один екземпляр презентера пов'язаний з одним відображенням.

Кожна вистава має реалізовувати відповідний інтерфейс. Інтерфейс подання визначає набір функцій і подій, необхідних для взаємодії з користувачем (наприклад, `IView.ShowErrorMessage (string msg)`). Презентер повинен мати посилання на реалізацію відповідного інтерфейсу, яку зазвичай передають в конструкторі.

Логіка уявлення повинна мати посилання на екземпляр презентера. Всі події вистави передаються для обробки в презентер і практично ніколи не обробляються логікою представлення (в т.ч. створення інших уявлень).

Приклад використання: Windows Forms.

Model-View-View Model



Даний підхід дозволяє пов'язувати елементи уявлення з властивостями і подіями View-моделі. Можна стверджувати, що кожен шар цього патерну не знає про існування іншого шару.

Ознаки View-моделі:

- Двостороння комунікація з поданням;

- View-модель - це абстракція уявлення. Зазвичай означає, що властивості уявлення збігаються з властивостями View-моделі / моделі
- View-модель не має посилання на інтерфейс уявлення (IView). Зміна стану View-моделі автоматично змінює уявлення і навпаки, оскільки використовується механізм скріплення даних (Bindings)
- Один екземпляр View-моделі пов'язаний з одним відображенням.

Реалізація:

При використанні цього патерну, уявлення не реалізує відповідний інтерфейс (IView). Подання повинно мати посилання на джерело даних (DataContext), яким в даному випадку є View-модель. Елементи уявлення пов'язані (Bind) з відповідними властивостями і подіями View-моделі. У свою чергу, View-модель реалізує спеціальний інтерфейс, який використовується для автоматичного оновлення елементів уявлення. Прикладом такого інтерфейсу в WPF може бути INotifyPropertyChanged.

Приклад використання: WPF

Висновки

Реалізація MVVM і MVP-патернів, на перший погляд, виглядає досить простий схожою. Однак, для MVVM зв'язування уявлення з View-моделлю здійснюється автоматично, а для MVP - необхідно програмувати. MVC має більше можливостей по управлінню поданням. Загальні правила вибору паттерна:

MVVM

Використовується в ситуації, коли можливо зв'язування даних без необхідності введення спеціальних інтерфейсів уявлення (тобто відсутня необхідність реалізовувати IView);

MVP

Використовується в ситуації, коли неможливо зв'язування даних (можна використовувати Binding);

MVC

Використовується в ситуації, коли зв'язок між уявлення і іншими частинами програми неможлива (і Ви не можете використовувати MVVM або MVP);

Частим прикладом використання може служити ASP.NET MVC.

Хотів би відзначити, що строго дотримуватися тільки одному паттерну - не завжди найкращий вибір. Наприклад, уявіть, що Ви хотіли б використовувати MVVM для розробки додатків з використанням Windows Forms через властивість контролів Bindings. Ваша мета - це відокремити подання від бізнес логіки і логіки, яка їх пов'язує. Додаток має бути легко тестується і підтримуваним, а для аналітиків - зрозумілим (адже на питання «в чому вимірюється робота жорсткого диска» існує єдиний правильний відповідь - в Джоулях (абстрактний приклад Моделі -> Уявлення)).

РОЗДІЛ 2. Організація розробки web-сайту

2.1 Загальна концепція

Процес розробки складається з безлічі підпроцесів, або дисциплін, деякі з яких показані нижче. У каскадній моделі вони йдуть одна за одною, в інших аналогічних процесах їх порядок або склад змінюється.

1. Аналіз вимог - Специфікація програмного забезпечення
2. Проєктування програмного забезпечення
3. Програмування
4. Тестування програмного забезпечення
5. Системна інтеграція
6. Впровадження програмного забезпечення (або Установка програмного забезпечення)
7. Супровід програмного забезпечення

Специфікація вимог до програмного забезпечення - специфікація вимог для програмної системи - це повний опис поведінки системи що розробляється. Вона включає множину прецедентів які описують всі взаємодії, які користувачі мають з програмним забезпеченням. Прецеденти також відомі як функціональні вимоги. На додачу до прецедентів SRS також включає нефункціональні (чи додаткові) вимоги. Нефункціональні вимоги є вимогами які накладають обмеження на проєкт, чи

реалізацію (такі як вимоги інженерії продуктивності, стандарти якості, чи обмеження проектування) [6].

Аналіз вимог полягає в визначенні потреб та умов, які висуваються щодо нового, чи зміненого продукту, враховуючи можливо конфліктні вимоги різних замовників, таких як користувачі чи бенефіціари. Аналіз вимог є критичним для успішної розробки проекту. Вимоги мають бути задокументованими, вимірними, тестовними, пов'язаними з бізнес-потребами, і описаними з рівнем деталізації достатнім для конструювання системи. Вимоги можуть бути архітектурними, структурними, поведінковими, функціональними, та не функціональними.

Проектування програмного забезпечення - це процес вирішення задач та планування для створення програмного рішення. Після того як мета і специфікація програми описані, розробник створить дизайн проекту, або найме дизайнера для розробки плану рішення. В дизайн включаються як описи низькорівневих компонентів, алгоритмів, так і огляд архітектури.

Проектуванню зазвичай підлягають:

1. Архітектура програмного забезпечення
2. Компоненти ПЗ
3. Користувацькі інтерфейси

В процесі проектування ПЗ застосовують різні моделі - блок-схеми, ER-діаграми, DFD тощо.

Програмування - процес проектування, написання, тестування, зневадження і підтримки комп'ютерних програм. Програмування поєднує в собі елементи інженерії, фундаментальних наук (перш за все математики) і мистецтва.

У вузькому значенні програмування розглядається як кодування - реалізація у вигляді програми одного чи кількох взаємопов'язаних алгоритмів (у сучасних умовах це здійснюється з застосуванням мов програмування). У ширшому сенсі

процес програмування охоплює і створення, тобто розробку, алгоритмів, і аналіз потреб майбутніх користувачів програмного забезпечення.

Тестування програмного забезпечення - це процес технічного дослідження, призначений для виявлення інформації про якість продукту відносно контексту, в якому він має використовуватись. Техніка тестування також включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою оцінки.

Супроводження програмного забезпечення — процес покращення, оптимізації та виправлення дефектів у програмному забезпеченні після його вводу до експлуатації[7].

Цей процес стандартизовано організацією ISO — ISO/IEC 14764.

У зв'язку з вирішенням так званої проблеми 2000 року (пов'язаної з кодуванням дат у новому тисячолітті, зокрема, у двохсимвольному форматі) супроводження почало розглядатися, як важливіший процес, що здійснюють розробники. Після змін система має вирішувати ті самі задачі, а також мати план перенесення інформації в інші БД. Супровід відповідно до стандартів ISO/IEC 12207 і ISO/IEC 14764 проводиться з метою виконання і модифікації програмного продукту в процесі експлуатації за умови збереження його цілісності [8].

Висновки

Отже, веб-сайт являє собою сервіс, який дає можливість розширити функціонал підприємства в цілому. Для того щоб організувати роботу ресурса, необхідно розробити його власноруч або скористатись послугами компаній, які надають послуги по розробці та підтримці веб-сайту. В нашому випадку я скористаюсь послугами компанії WIX, яка пропонує швидко и якісно створит сайт-візитку. В 3 розділі я на прикладі продукту компанії, покажу и розповім детальніше, яким чином MVC патерн працює на практиці.

РОЗДІЛ 3. Програмна реалізація

3.1 Реалізація бази даних

В нашому прикладі (використання послуг компанії) мені не вдасться в повній мірі візуалізувати роботу баз даних, але можу пояснити як саме бази даних підключаються до проекту.

Приклад я наводитиму з веб-додатку, який написаний на мові програмування RUBY, з використанням фреймворку Ruby On Rails (далі я буду використовувати аббревіатуру RoR)

В цьому фреймворку база даних автоматично підключена “з коробки”, у нас лише існує необхідність в заповненні даними нашого пустого сховища. Але часто буває, що щось йде не так і нам необхідно робити все вручну, пропоную розглянути цей процес.

Використовую PostgreSQL:

1. Створення бази даних

Для створення бази даних використовується команда CREATE DATABASE, після якої вказується назва бази даних.

Для виконання запитів будемо використовувати графічний клієнт pgAdmin, хоча також можна використовувати консольний клієнт psql.

Щоб створити нову базу, даних відкриємо pgAdmin. У лівій частині програми виберемо якусь базу даних, наприклад, стандартну бд postgres, і натиснемо на неї правою кнопкою миші.(Рис.3.1.)

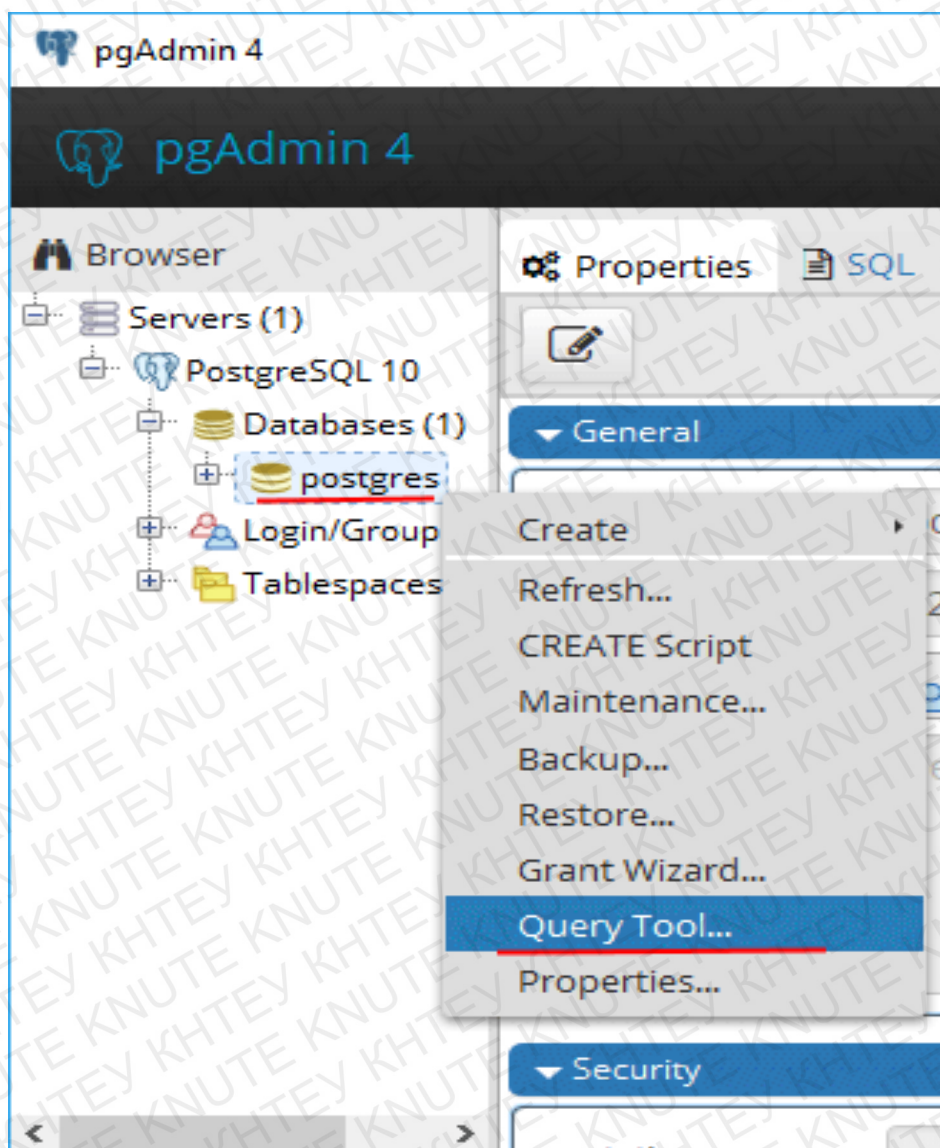
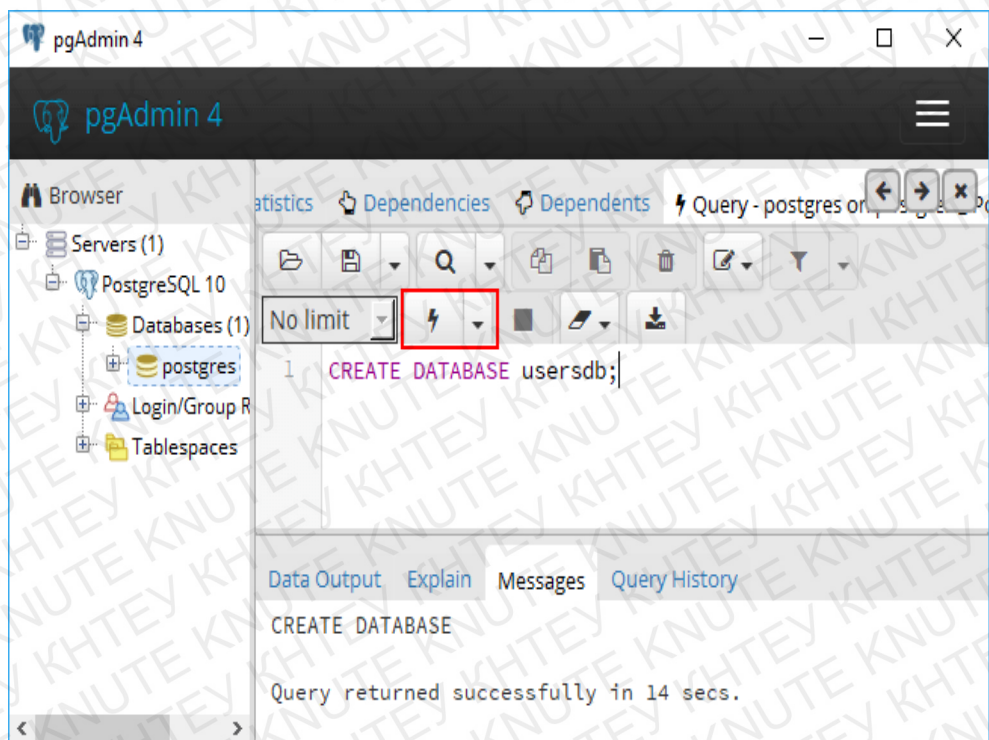
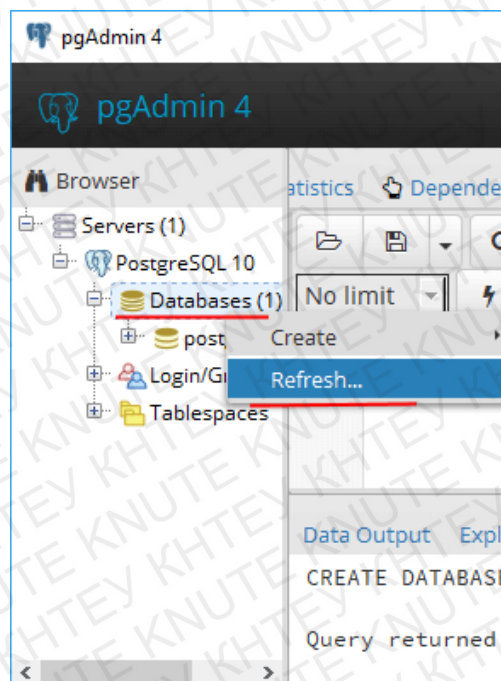


Рис.3.1. База даних

У меню виберемо пункт Query Tool ..., і в центральній частині програми відкриється поле для введення коду SQL. У це поле введемо наступний код:



Для виконання коду натиснемо на значок блискавки, і після цього буде створена база даних usersdb. Щоб побачити нашу базу даних, натиснемо в лівій частині на вузол Databases правою кнопкою миші і в контекстному меню виберемо Refresh ...:



Відбудеться оновлення, і ми побачимо створену базу даних. (Рис.3.2.)

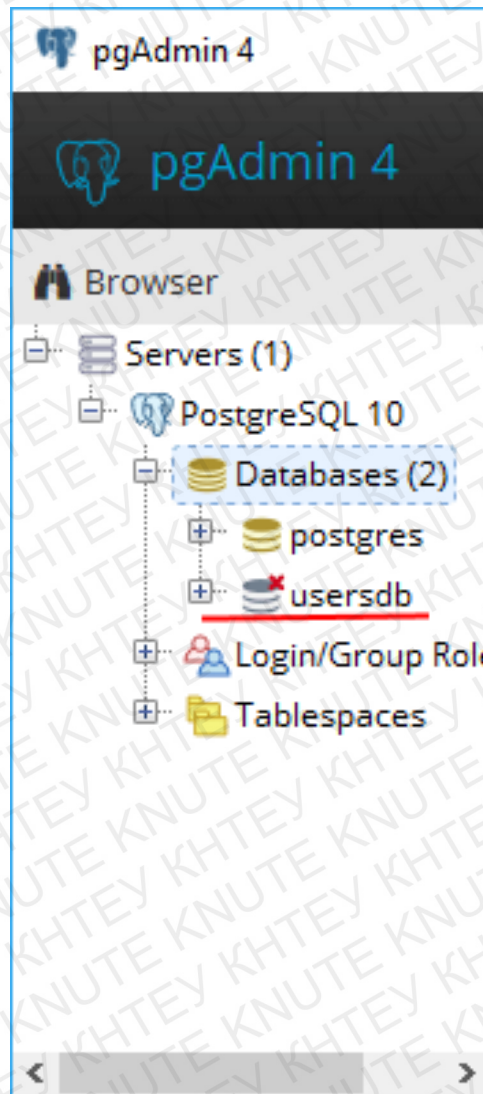


Рис.3.2 Готова База Даних

За замовчуванням база є неактивною, тому її значок має сірий колір. Але щоб до неї підключитися, досить натиснути на неї і розкрити її вузол.

3.2. Реалізація проекту

Для детального пояснення як саме реалізується патерн MVC я перейшов на сторінку wix.com і створив за 1 клік сайт-візитку, який має наступний вигляд.(Рис.3.3)

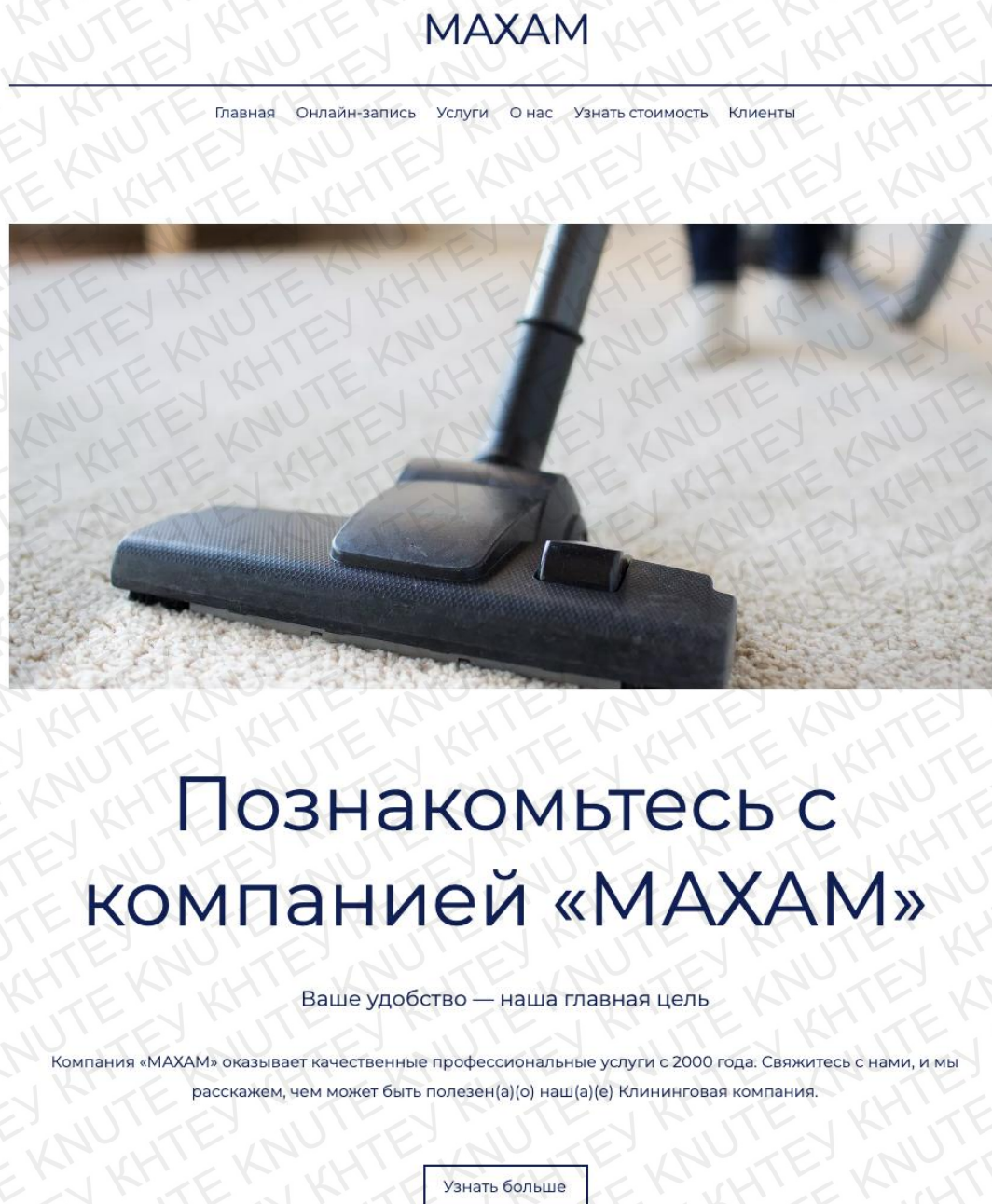


Рис.3.3 Головна сторінка

Вгорі представлений бар, на якому знаходиться основна навігація по веб-сайту. Перейшовши на сторінку “Послуги”, я відправив запит у Контролер(Рис.3.4)

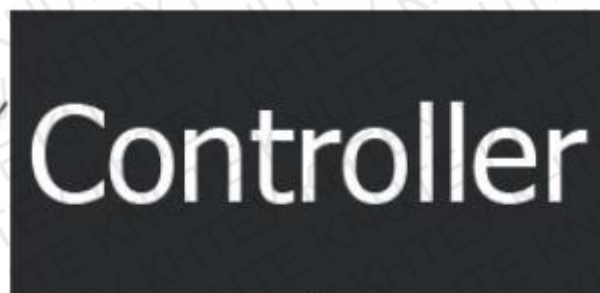


Рис.3.4 Взаємодія Юзера з MVC

Після обробки запиту, контроллер передає його далі в модель, в якій власне реалізована основна логіка веб-додатку. (Request Information) (Рис.3.5)

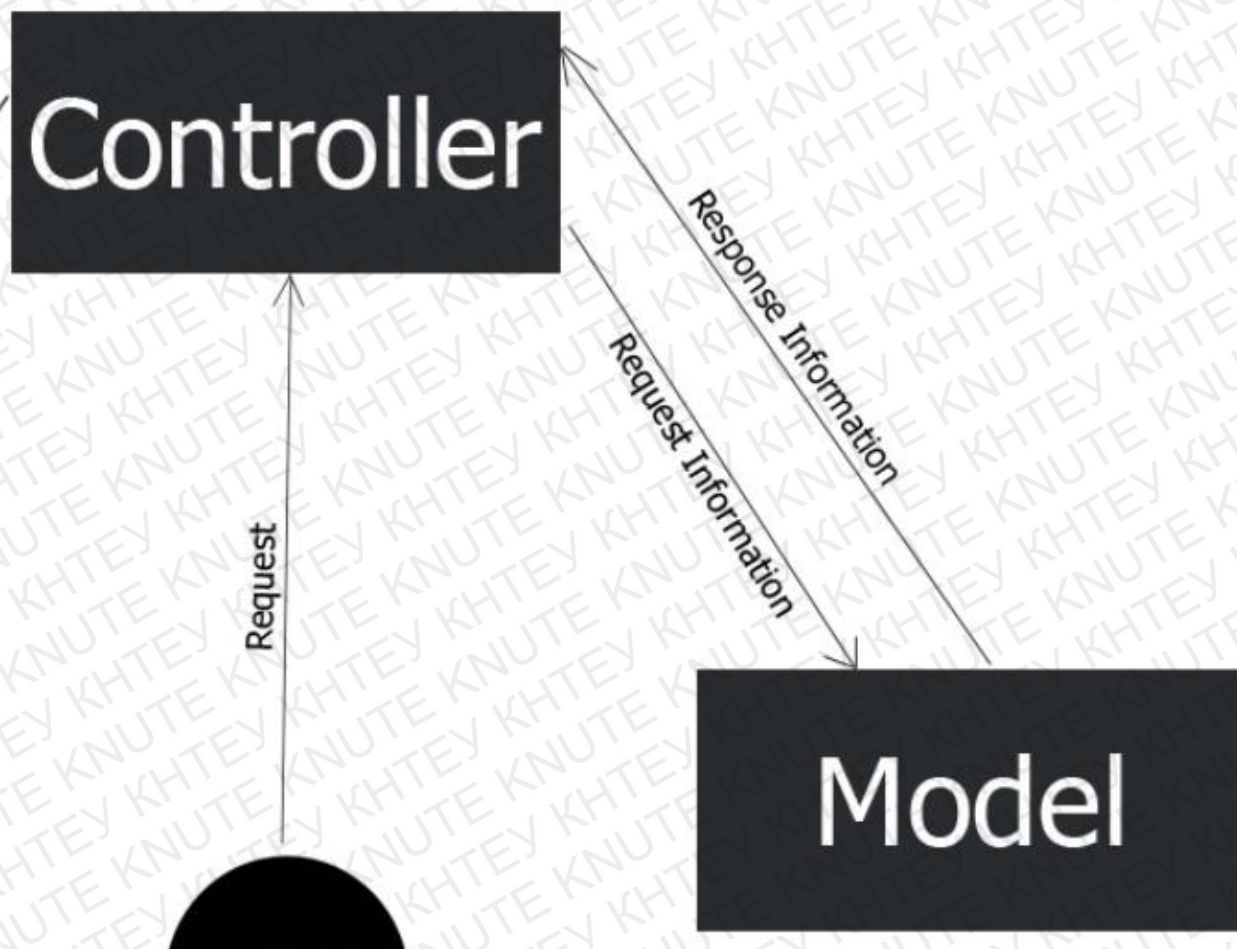


Рис.3.5 Взаємодія Юзера з MVC

Модель в свою чергу реалізує логіку, робить запит в базу даних, “дістає” звітти необхідну інформацію і передає назад відповідь (Response Info) у Контроллер.

Наступним кроком в реалізації патерну MVC є обробка Контролером отриманої відповіді і передача інформації у View. Власне, це є фінальним етапом в реалізації патерну MVC. Користувач отримує необхідні дані і використовує наш веб-сайт далі. (Рис.3.6)

Model-View-Controller

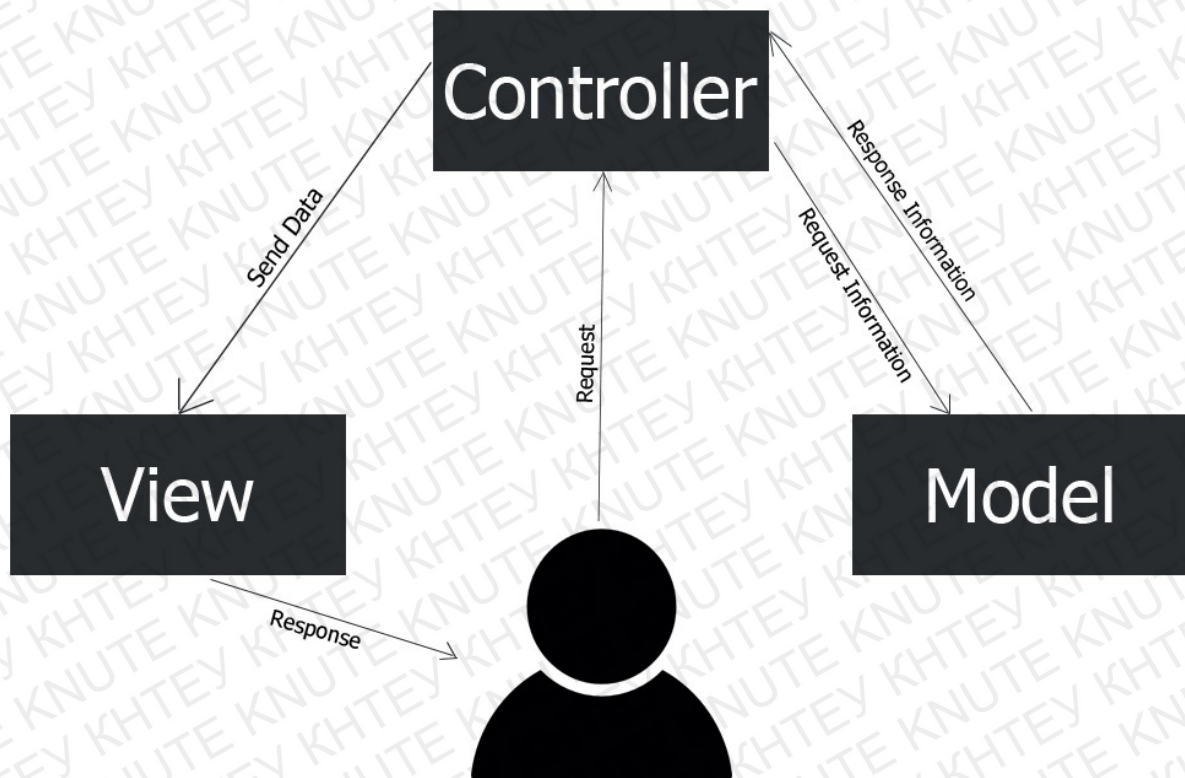


Рис.3.6 Повне коло взаємодії Юзера з MVC

Результат цього ми можемо бачити наступним чином. Ми отримали відповідь від веб-сервера і успішно отримали доступ до бажаної вкладки на веб-сайті.(Рис.3.7)

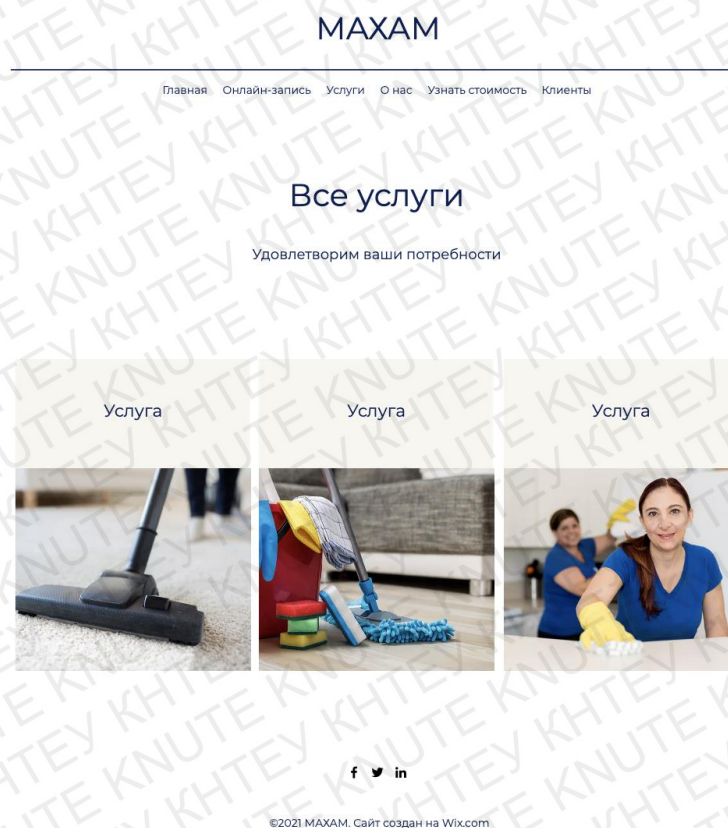


Рис.3.7 Сторінка послуг

ВИСНОВКИ

У випускній кваліфікаційній роботі представлено результати теоретичних і практичних досліджень, що полягають у розробці веб-сайту. Розроблен

Готовий веб-сайт відповідає всім вимогам і має зрозумілий призначений для користувача інтерфейс. В результаті проведених досліджень були отримані такі

ВИСНОВКИ:

1. Патерн MVC наразі є найкращим в своєму роді, але на мою думку, в недалекому майбутньому він буде доповнений або замінений в цілому.
2. Для того щоб організувати роботу веб-сервісу для свого підприємства, необхідно розробити його власноруч або скористатися послугами профільних компаній.
3. У результаті на основі теоритичної інформації був розроблений веб-сайт для клінінгової фарми. Веб-сайт підтримує онлайн-консультації, запис для надання послугу, всю необхідну інформацію, зручний та досконалий на мою думку інтерфейс.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Веб-аналитика : От систем статистики посещаемости до визуализации поведения посетителей сайтов [Электронный ресурс]. – Режим доступа : <http://altblog.ru/veb-analitika-ot-sistem-statistiki-poseshhaemosti-do-vizualizacii-povedeniya-posetitelej-sajtov/>.
2. Забарна Е. М. Система та критерії маркетингового аналізу сайтів вищих навчальних закладів [Електронний ресурс] / Е. М. Забарна, К. Ю. Соловйова // Ефективна економіка : електронне наукове фахове видання. – 2013. – №1. – Режим доступу : <http://www.economy.nayka.com.ua/?op=1&z=1748>.
3. Мелихов Д. С. Веб-аналитика : шаг к совершенству / Д. С. Мелихов, И.И. Сарматов. – К. : Аналитик Интеллект Сервис, 2010. – 112 с.
4. Розкручування і просування сайта. Як визначити цільову аудиторію сайта? [Електронний ресурс]. – Режим доступу : <http://webstudio2u.net/ua/web-promotion/197-target-audience.html>.
5. Розкручування сайта в рейтингах [Електронний ресурс] // Buildsait : створення, просування, оптимізація сайта. – Режим доступу : <http://buildsait.pp.ua/index.php/promotion-web-site/49-web-site-promotion-in-ratings>.
6. Юрчак І. Ю. Конспект лекцій. Веб-проекування [Електронний ресурс] / І. Ю. Юрчак. – Режим доступу : <http://victoria.lviv.ua/html/gim/document.html>.