


**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики  
Кафедра математичної інформатики

**Кваліфікаційна робота  
на здобуття ступеня бакалавра**  
за спеціальністю 122 Комп'ютерні науки


на тему:  
**Розробка веб-додатку проведення опитувань з використанням  
фреймворку Django**

Виконав студент 4-го курсу  
Корніленко Даніель Ярославович


  
\_\_\_\_\_  
(підпис)

Науковий керівник:  
Професор, кандидат фіз.-мат. наук

Вергунова Ірина Миколаївна

  
\_\_\_\_\_  
(підпис)

Засвідчую, що в цій  
роботі немає запозичень з  
праць інших авторів без  
відповідних посилань

Студент   
\_\_\_\_\_  
(підпис)

## РЕФЕРАТ

Обсяг роботи 65 сторінок, 14 ілюстрацій, 1 додаток, 16 джерел посилань.

ІНТЕРНЕТ ОПИТУВАННЯ, PYTHON, DJANGO, ВЕБСАЙТ, SQLITE

Об'єкт роботи – проведення опитувань в мережі Інтернет.

Предмет роботи – веб-додаток для проведення опитувань.

Мета роботи – створити веб додаток для проведення опитувань з використанням фреймворку Django, який можна використовувати як складову частину інших веб проектів та веб-сайтів.

Методи та інструменти дослідження: аналіз існуючих систем, аналіз існуючих шаблонів програмування, аналіз існуючих моделей для вирішення задачі, аналіз існуючих баз даних для зберігання файлів.

Засоби розробки – мова програмування Python, фреймворк Django, HTML, CSS, система керування базами даних SQLite, інтегроване середовище розробки Visual Studio Code.

Результати розробки: виконано огляд інформаційного забезпечення, що використовується для проведення опитувань, розроблено веб додаток для проведення опитувань з використанням фреймворку Django.

Робота має практичне значення та може бути використана в інтернет магазинах та інших веб сайтах, що потребують опитування серед своєї цільової аудиторії.

## ЗМІСТ

РЕФЕРАТ .....	1
ВСТУП .....	4
1 ЗАГАЛЬНИЙ РОЗДІЛ.....	7
1.1 Опис предметної області роботи.....	7
1.1.1 Вивчення об'єкту роботи .....	7
1.1.2 Опис структурних і функціональних особливостей об'єкту роботи.....	9
1.1.3 Проведення детального аналізу проблеми .....	10
1.2 Огляд і аналіз літератури за темою роботи.....	12
1.2.1 Аналіз ступеню сучасного стану розв'язання задач у даній предметній галузі .....	12
1.2.2 Аналіз аналогів програмних продуктів .....	14
1.2.3 Порівняльна оцінка відомих у літературі методів і підходів ....	15
1.3 Обґрунтування актуальності дослідження.....	18
1.4 Постановка задачі .....	20
Висновки за загальним розділом.....	20
2 СПЕЦІАЛЬНИЙ РОЗДІЛ .....	22
2.1 Проектування інформаційної системи.....	22
2.1.1 Загальні положення .....	22
2.1.2 Інформаційна модель системи.....	23
2.2 Інформаційне забезпечення .....	25
2.2.1 Структура і схеми інформаційних об'єктів і ресурсів .....	25
2.2.2 Схеми інформаційних потоків.....	26
2.2.3 Схеми бази даних.....	26
2.3 Алгоритмічне забезпечення.....	29
Висновки за спеціальним розділом.....	30
3 РОЗРАХУНКОВИЙ РОЗДІЛ.....	31
3.1 Програмне забезпечення .....	31
3.1.1 Системне програмне забезпечення .....	32
3.1.2 Прикладне програмне забезпечення .....	33
3.2 Технічне забезпечення.....	33

3.3	Процес розробки системи .....	34
3.4	Опис програми .....	38
3.5	Інструкції з використання програми.....	40
3.5.1	Інструкція системного адміністратора .....	40
3.5.2	Інструкція користувача .....	41
3.5.3	Інструкція користувача із правами адміністратора.....	44
	Висновки за розрахунковим розділом .....	49
	ВИСНОВКИ.....	50
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	52
	ДОДАТОК А.....	54

## ВСТУП

Опитування можна розглядати як один із найпоширеніших методів отримання інформації про суб'єктів — респондентів. Опитування полягає в задаванні людям спеціальних запитань, відповіді на які дозволяють дослідникам отримати необхідні відомості стосовно завдань дослідження. До особливостей опитування можна зарахувати його масовість, що викликано специфікою вирішуваних завдань. Опитування широко застосовуються у найрізноманітніших сферах життя: опитування клієнтів компанії з метою покращення якості товару, послуг або сервісу, опитування як засіб визначення рівня підготовки учнів або студентів, широкі опитування з метою визначення суспільної думки тощо.

В наш час всесвітня мережа Інтернет активно використовується у комерції, освіті, доступі до демократії. Таким чином, зростає й потреба у аналізі зворотної реакції від користувачів. Компанії, що займаються соціологічними опитуваннями, також починають активніше користуватись опитуваннями через інтернет, бо цей спосіб є не менш ефективним, аніж телефонне опитування і дозволяє охопити деякі категорії людей, щодо яких неможливо провести опитування в реальному житті – зокрема, люди з інвалідністю або люди, які вимушені проводити більше часу вдома з метою самоізоляції під час пандемії. Також через пандемію зростає необхідність у доступі до тестування рівня підготовки учнів через мережу Інтернет. В той же час, частка продажів через мережу Інтернет стрімко зросла в 2020 році, а отже зросла й необхідність в опитуванні клієнтів від комерційних установ [1].

Зважаючи на актуальність розробки інструментів для проведення опитувань, які б дотримувалися сучасних вимог законодавства щодо роботи з особистими даними, темою представленої роботи є розробка веб-додатку проведення опитувань з використанням фреймворку Django.

Об'єкт роботи – проведення опитувань в мережі Інтернет.

Предмет роботи – веб-додаток для проведення опитувань.

Мета роботи – створити веб додаток для проведення опитувань з використанням фреймворку Django, який можна було б використовувати як складову частину інших веб проектів та веб-сайтів.

Для досягнення мети роботи поставлено такі завдання:

- проаналізувати існуючі веб додатки для проведення опитувань;
- встановити вимоги до веб додатку;
- обрати IDE для роботи з програмним кодом;
- розробити веб-додаток;
- розробити та реалізувати автоматизовані тести.

Методи та інструменти дослідження: аналіз існуючих систем, аналіз існуючих шаблонів програмування, аналіз існуючих моделей для вирішення задачі, аналіз існуючих баз даних для зберігання файлів.

У роботі використано такі методи дослідження:

- Емпіричні:
  - спостереження – перегляд існуючих рішень у сфері веб опитувачів,
  - порівняння опитувачів, пошук спільних рис та відмінностей [2];
- Теоретичні:
  - ідеалізація – формулювання вимог до опитувача, що буде ідеальним для рішення задачі;
- Загальнонаукові:
  - аналіз поняття веб-додатку для проведення опитувань, виокремлення притаманних для нього рис,
  - метод логічного аналізу – для конструювання на основі отриманих з допомогою попередніх методів відомостей моделі веб опитувача, який необхідно розробити.

Засоби розробки – мова програмування Python, фреймворк Django, HTML, CSS, система керування базами даних SQLite, інтегроване середовище розробки Visual Studio Code.

В ході виконання роботи було визначено доцільність використання фреймворку Django для розробки веб-додатку для проведення опитувань. Отриманий програмний продукт має зручний інтерфейс адміністратора, що дає можливість додавати, видаляти та коригувати запитання та варіанти відповідей на них. Даний програмний продукт може стати частиною інтернет-магазину або веб сервісу з будь-якої іншої галузі, де є необхідним використання опитувань.

# 1 ЗАГАЛЬНИЙ РОЗДІЛ

## 1.1 Опис предметної області роботи

### 1.1.1 Вивчення об'єкту роботи

Веб-додаток – це будь-яка комп'ютерна програма, яка виконує певну функцію, використовуючи веб-браузер у якості свого клієнта, в незалежності від пристрою та платформи, на яких відкрито браузер [1-3].

Додаток може бути таким же простим, як дошка оголошень або контактна форма на веб-сайті, або настільки ж складний, як текстовий процесор або багатокористувацький мобільний ігровий додаток, який ми завантажуюмо на свій телефон.

Також особливістю веб-додатків є інтерактивність. На відміну від переважно статичних веб-сайтів, веб додатки містять можливості для взаємодії із користувачем.

До переваг веб-додатків варто віднести:

- Кросплатформність. На відміну від нативних додатків, веб-додаток може використовуватись на усіх пристроях. Він запрограмований для роботи в будь-якій операційній системі. Він може належним чином адаптуватися як до ОС для персональних комп'ютерів, так і до iOS, Android, Windows Phone та інших систем.

- Немає проблем з підтримкою старих версій програм й з сумісністю. Коли з'являється нова версія нативного додатку, користувачам нерідко доводиться вирішувати проблеми, зв'язаних з оновленням вже встановленої на їх пристроях копії. У випадку з браузерним додатком таких проблем не буває – існує лише одна версія, в якій працюють усі користувачі і у випадку виходу нової, всі без виключень переходять на неї, іноді навіть не помічаючи цього.

- Веб-додатки не вимогливі до ресурсів і не пред'являють ніяких вимог до апаратної платформи.

– Веб-додаток не потребує встановлення на пристрій. Їх не потрібно завантажувати та встановлювати з магазинів додатків, таких як Google Play або Apple Store App.

– Web-додатки дозволяють своїм користувачам бути насправді мобільними. Можна зберігати результат своєї роботи на сервері і у випадку необхідності мати доступ до нього з будь якої точки планети, де є вихід у мережу Інтернет.

Також важливою складовою об'єкта роботи є фреймворк, тобто інфраструктура програмних рішень, що полегшує розробку складних систем. Спрощено дану інфраструктуру можна вважати своєрідною комплексною бібліотекою, але при цьому вона має ряд обмежень, що задають правила створення структури проекту та написання коду.

Django є одним з найбільш популярних фреймворків серед python-програмістів, нарівні з Flask та іншими [4]. Так само слід зазначити, що розробка на цьому фреймворку відносна дешева, так як на ринку праці є багато фахівців. Django дає можливість зосередитися на творчій частині розробки, полегшуючи її не такі цікаві частини, такі як роутинг, інтерфейс для AMQ протоколу, інтерфейс для інтернет протоколів і багато низькорівневих модулів. Розробник має в своєму розпорядженні високорівневі шаблони, а так само й повну і детальну офіційну документацію. В цей же час Django не обмежує свободи в процесі розробки, дозволяючи працювати з безліччю сторонніх сервісів, наприклад Amazon, Kafka та інші. Фреймворк надає вже готову інфраструктуру додатку, що дозволяє зосередитися на написанні бізнес логіки, без необхідності щоразу витратити час на написання низькорівневих функцій.

Коли ведеться розмова про середовищі розробки, існує три важливих факти, які необхідно мати на увазі: ізоляція, зумовленість і схожість. Кожен з них важливий і всі вони узгоджено взаємодіють один з одним.

Ізоляція означає, що не можна випадково скористатися інструментами або пакетами встановленими поза наявного оточення. Це особливо важливо, коли

подібне відбувається з чимось, схожим на пакети Python з розширеннями написаними на C: якщо ми використовуємо щось встановлене на системному рівні і не знаємо про це, то при розгортанні або розповсюдженні свого коду можемо виявити, що він працює не так як передбачалося. Інструменти на зразок `virtualenv` можуть допомогти створити щось схоже на ізольоване середовище.

І, нарешті, схожість з виробничим середовищем або середовищем сервера розробки означає, що всюди встановлена одна і та ж операційна система (можливо навіть однаковий випуск) і використовуємо однакові інструменти як для конфігурації середовища розробки, так і для конфігурації виробничого середовища. Якщо будується велике і складне програмне забезпечення, схожість буде корисною для впевненості в тому, що всі проблеми, які можемо побачити на «продакшн» сервері, відтворювані в тому середовищі, де ведемо розробку.

У Django також є вже готовий інтерфейс для адміністраторів сайту Django Admin з можливістю налаштування за власними потребами. Django Admin буде використовуватися в цьому додатку в якості інтерфейсу для адміністраторів інформаційної системи.

Для додатку використаємо версію Django 3.2.3.

### **1.1.2 Опис структурних і функціональних особливостей об'єкту роботи**

Структура розроблюваного проекту складається з таких елементів:

- користувацька частина;
- адміністративна частина.

Користувацька частина містить стартову сторінку, на якій користувач опиняється, перейшовши за посиланням, сторінку з переліком запитань та сторінку для відповіді на питання. Також у вільному доступі є сторінка зі

статистикою відповідей на питання. Всі ці сторінки не потребують авторизації, що зроблено із метою анонімності збору відповідей та прозорості опитувача.

Адміністративна частина складається зі сторінки авторизації, головної сторінки, сторінок редагування інформації про користувачів, групи користувачів та опитувань.

Функціонально користувацька частина дозволяє відповідати як на всі питання як по черзі, так і обираючи питання для відповідей самостійно та переглядати статистику відповідей на запитання.

Функціонал адміністративної частини полягає в можливості редагування інформації користувачів та питань. Зокрема, адміністратор має можливість додавати нові питання, видаляти старі, планувати появу нових запитань у доступі користувача. Також разом із питаннями є можливість редагувати або додавати й варіанти відповідей. Є можливість пошуку запитань через текстове поле та фільтрації за часом публікації питання.

### **1.1.3 Проведення детального аналізу проблеми**

З 60-х років, з моменту зародження, Інтернет стрімко еволюціонував. Якщо в 60-70-і роки він був просто засобом обміну інформацією між науковими центрами, то з початку 90-х років набув широкої популярності. Безумовно, поштовхом до цього послужила розробка в 1992 р. World Wide Web. Всесвітня павутина заснована на принципово новій концепції представлення інформації і навігації, що зрозуміла не тільки професійним програмістам, але і звичайним користувачам. У міру того як Інтернет отримувал все більшу популярність у користувачів, до нього почали проявляти інтерес і маркетологи.

В даний час все більш актуальною темою обговорення є інтеграція глобальної мережі Інтернет з реальним економічним світом. У зв'язку з цим темою моєї дипломної роботи є розробка вебдодатку проведення опитувань з використанням фреймворку Django. Я вважаю, що маркетингові дослідження,

що проводяться в Мережі, є одним з основоположних ланок у вивченні теми впровадження Інтернет технологій в сучасний бізнес.

Як показує досвід промислово розвинених країн, використання Інтернету з метою опитування приносить колосальний ефект. У Інтернету є величезні переваги, які окупають будь-які складнощі роботи з Інтернет-опитуваннями. До них відносяться:

- низька вартість опитувань (в офлайн доводиться платити зарплату, оплачувати оренду офісу та інші витрати, в Інтернеті ж всі витрати - це разова розробка, хостинг, сервери та їх адміністрування);

- доступність цільових груп (дослідження показують що, аудиторія Інтернету починає поступово відмовлятися від інших джерел отримання інформації);

- швидкість отримання відгуку (швидкість комунікації через Інтернет дуже висока, відповіді з опитування можна отримати вже через кілька хвилин після надходження повідомлення на електронну пошту користувача).

Інтернет для сучасної організації, яка будує всі свої бізнес-процеси на основі маркетингового підходу, служить не тільки джерелом вторинної інформації про тенденції зміни ринку, переваги споживачів, маркетингові зусилля конкурентів. Інтернет стає також можливим джерелом отримання первинної маркетингової інформації при проведенні опитувань в режимі online, організації електронних фокус-груп, вивчення думки відвідувачів сайтів або споживачів про пропонований в електронному магазині товар.

Саме на вивченні думки відвідувачів сайтів я і зосереджу увагу в даній роботі. Опитування користувачів, які вже замовили товар через інтернет магазин або принаймні зареєструвались на ньому, дає можливість ідеально влучити у цільову аудиторію та проаналізувати досвід користувача у взаємодії з торгівельним майданчиком. Аналіз того, що людина відчуває при користуванні продуктом, системою чи сервісом є вкрай важливим для планування змін та покращень до продукту. Основними об'єктами дослідження є враження, емоції

та користь, отримані від взаємодії з продуктом. Також досвід користування включає практичність, простоту використання та швидкодію системи. Досвід користування має суб'єктивний характер (бо пов'язаний з індивідуальними відчуттями і думкою) і може змінюватись з часом при зміні обставин. Через суб'єктивність досвіду можуть виникати ситуації, коли замовник розробки та сам розробник абсолютно задоволені створеним продуктом, але на справі він виявляється занадто складним для цільової аудиторії.

## **1.2 Огляд і аналіз літератури за темою роботи**

### **1.2.1 Аналіз ступеню сучасного стану розв'язання задач у даній предметній галузі**

Онлайн опитування існують вже доволі давно. Точних відомостей про створення першого інтернет-опитування немає, проте дослідники приписують їх появу до періоду бульбашки дот-комів в США, який відбувся в період приблизно з 1995 по 2001 рік. Бульбашка утворилася в кінці ХХ століття в результаті зльоту акцій інтернет-компаній (переважно американських), появи великої кількості нових інтернет-компаній та переорієнтування старих компаній на інтернет-бізнес. Акції компаній, що пропонували використовувати Інтернет для отримання доходу, нечувано злетіли в ціні. Такі високі ціни виправдовували численні коментатори і економісти, які стверджували, що настала «нова економіка», насправді ж ці нові бізнес-моделі виявилися неефективними, а значні кошти, витрачені, в основному, на рекламу і великі кредити, призвели до хвилі банкрутств, сильного падіння індексу NASDAQ, а також обвалу цін на серверні комп'ютери. Остання частина періоду доткомів була зміною різких підйомів і обвалів, хоча сам інтернет-бум прийнято відносити до стабільного комерційному росту інтернет-компаній, викликаного настанням епохи всесвітньої павутини, що почалася з першого релізу веб-браузера Mosaic в 1993 і тривала всі 1990-ті роки. Однак, не зважаючи на економічні наслідки, які

спричинила ця бульбашка, вона також зумовила ріст цікавості до інтернет технологій, браузерів та веб-сайтів. Великі інвестиції у сектор зумовили появу великої кількості високооплачуваних програмістів, без яких, можливо, не з'явилось би Інтернету в тому вигляді, до якого ми звикли.

На даний момент існує незліченна кількість сайтів, які спеціалізуються на розміщенні контенту у вигляді опитувань. Зокрема, це такі сайти, як <https://onlinetestpad.com/> [5] та <https://psyttests.org/> [6]. Ці сайти отримують перегляди, які монетизуються завдяки тому, що багато людей схильні до того, щоб спробувати дізнатись про себе більше, давши відповіді на десяток питань онлайн.

Також часто опитування є складовими частинами інтернет магазинів. Наприклад, <https://tohome.com.ua/> має окрему сторінку із вбудованою формою опитування від GoogleФорм для отримання зворотного зв'язку від клієнтів. Чимало інтернет магазинів також інтегрують опитування для вирішення скарг клієнтів: наприклад, китайський інтернет-магазин Aliexpress, український інтернет-магазин Епіцентру.

Перехід до онлайн форми навчання також зумовив багато навчальних закладів запровадити опитування з метою перевірки рівня знань студентів.

Гапоненко Д.І. у роботі «Інтернет-опитування як метод збору соціологічної інформації» [7] аналізує стан розвитку даних технологій в Україні. Він зауважує, що поточний стан переходу інтернет опитувань в онлайн формат в Україні помітно відстає від стану у розвинених країнах, але ми маємо високі темпи переходу, а отже – маємо шанси досягнути їх рівня. Серед оптимістичних чинників дослідник зауважує зменшення вартості доступу людей в Інтернет та збільшення інтересу громадян до Інтернету. Також віртуальне середовище є надзвичайно ефективним оскільки не вимагає переміщення людей, використання паперових носіїв інформації і може бути досяжним для користувача у будь-який зручний час.

Однак існують і суттєві недоліки в проведенні інтернет опитувань. Соціологічна мета опитувань часто вимагає репрезентативної вибірки

респондентів, досягнути чого майже неможливо при використанні інтернет опитувань. Це зумовлено різною мірою залучення до інтернет технологій людей різного віку та соціальних категорій.

Автор приходить до висновку, що, незважаючи на інфраструктурні обмеження, розвиток телекомунікацій в Україні відбувається в оптимальному напрямку, що безперечно сприятиме подальшому зростанню вітчизняної Інтернет-аудиторії. Крім того, віртуальне середовище є надзвичайно ефективним, оскільки не вимагає переміщення людей, використання паперових носіїв інформації і може бути досяжним для користувача у будь-який зручний час. Разом з тим не слід забувати і про використання мережі українськими підприємствами, які працюють як на внутрішньому, так і на зовнішньому ринках. Кожен з цих факторів має суттєвий вплив на подальше проникнення Інтернету в країні. Відповідно й Інтернет-опитування як метод збору первинної соціологічної інформації, незважаючи на відносну молодість, буде поступово завойовувати собі вагоме місце серед інших соціологічних методів, а також здобувати популярність у наукових колах.

### **1.2.2 Аналіз аналогів програмних продуктів**

Веб-продукти для опитувань можна розділити на такі групи [8]:

- веб-опитувачі, що є складовою частиною сайтів соціологічних організацій;
- веб-опитувачі навчальних закладів;
- веб-опитувачі комерційних веб-сайтів;
- веб-опитувачі у формі чат-ботів.

Кожен із цих опитувачів має свою як програмну, так і користувацьку специфіку. Веб-опитувачі, що є складовою частиною сайтів соціологічних організацій, вимагають збір і зберігання великого обсягу приватної інформації:

вік, стать, освіта, професія, рівень доходу для отримання репрезентативної вибірки. Зазвичай такі веб-додатки створюються і підтримуються штатними програмістами соціологічної організації з метою безпеки зберігання даних та оптимальної форми й кількості питань.

Веб-опитувачі навчальних закладів досі знаходяться у стані становлення. На розвиток цієї групи програмних продуктів значно вплинув локдаун, який спричинив масовий перехід навчальних закладів на віддалену форму навчання.

Веб-опитувачі комерційних веб-сайтів можна розділити на дві групи: опитувачі сайтів, що мають дохід від реклами та сайтів, що отримують дохід від продажу товарів. Специфіка сайту визначає також і предмет опитування: перша група зосереджує увагу на зручності користування сайтом, друга частіше містить питання про конкретні товари. Найчастіше користувачі стикаються з опитуванням, яке має на меті отримання відгуку про товар (наприклад, <https://www.otzyvua.net>).

Веб-додатки з опитуванням у формі чат-боту мають декілька визначних відмінностей: по-перше, відповідь на одне питання впливає на задавання наступного питання. Такі опитувачі можуть міститись як у чат-ботах месенджерів, так і чат-ботах, інтегрованих у веб-сайти. Яскравим прикладом останнього підходу є чат-бот у чаті підтримки Приват-банку, що за ключовими словами пропонує у формі опитування уточнити предмет питання та конкретне питання.

### **1.2.3 Порівняльна оцінка відомих у літературі методів і підходів**

Серед найбільш поширених різновидів методу Інтернет-опитування науковці виділяють наступні: розсилка анкет електронною поштою (email-анкетування); розміщення текстових анкет у групах новин (newsgroups); Інтернет-форуми (Bulletin Boards); Web-сторінка (анкета у форматі HTML); Web-додаток; електронні фокус-групи.

Перші соціологічні дослідження проводилися за допомогою електронної пошти. Цей метод і сьогодні залишається одним з найпростіших і універсальних. Його техніка мало чим відрізняється від традиційного опитування, коли невеликі за обсягом анкети (20 – 25 питань) розсилаються респондентам. Різниця лише в тому, що для цього використовується e-mail. Процедура контролю якості заповнення і підвищення кількості повернення анкет така сама, як і у традиційному поштовому опитуванні. Найголовніша перевага e-mail опитування – простота, низька собівартість і висока швидкість збору даних. Дослідник, який не володіє спеціальною комп'ютерною підготовкою, може легко організувати і провести таке дослідження. E-mail-опитування ідеально підходять для локальних досліджень, наприклад, у межах одного підприємства.

Для розміщення текстів анкет також використовують групи новин (newsgroups). Вони створюються для обговорення певних тем і охоплюють значні аудиторії користувачів. Кожна група має свій список учасників, який можна використати для побудови вибірки. Анкета відправляється у вигляді текстового повідомлення на сайт newsgroup, а опитування відбувається інтерактивно чи автономно. Дані від респондентів надходять до дослідника у вигляді текстів і потребують самостійної обробки. Для залучення нових респондентів необхідно регулярно обновляти розсилку анкети (приблизно раз на тиждень). Опитування newsgroup рідко бувають репрезентативними, хоча необхідність їх проведення викликана об'єктивними причинами – існує чимало цільових аудиторій, які складно дослідити у реальності. Учасники групи новин, як правило, мають певні непоширені в суспільстві уподобання. Взагалі середовище Інтернет включає в себе величезну кількість співтовариств. Воно представляє собою певний конгломерат, у межах якого відбуваються різні форми взаємодії. Загалом їх можна умовно поділити на дві великі групи: професіоналів і непрофесіоналів. До першої належать програмісти, ІТ спеціалісти, професіонали ЗМІ, представники бізнес-структур. До другої – об'єднання людей за певними вподобаннями: музика, кіно, література тощо. Значний недолік

використання даного методу – неможливість відстежити користувачів, які читали опитувач, проте не заповнили його.

Іншим різновидом онлайн-досліджень є технологія проведення опитувань на Інтернет-форумах. Вона відносно проста і не потребує значних витрат, хоча інформація збирається впродовж відносно тривалого проміжку часу. Для цього необхідно зареєструватися на форумі з потрібною досліднику аудиторію і певний час спостерігати за дискусіями, а потім прийняти в них активну участь. Після цього можна розмістити кілька питань, на які дослідник має намір отримати відповіді. Бажано публікувати не всі питання анкети, а тільки ключові. Повний додаток краще розміщати на окремому Web-сайті, на який подається посилання. Респонденти, як правило, активно приймають участь у таких опитуваннях, хоча окремі проблеми все ж таки можливі: викривлення інформації, неадекватне реагування, неповне заповнення анкети тощо. Зазначена технологія особливо ефективна, коли для опитування залучається група експертів.

Web-сторінки – це звичайна текстова анкета, тільки у HTML-форматі, яка розміщається на сайті. Усі питання анкети зосереджуються на одній великій сторінці, яка може містити необмежену кількість питань. Респондент відповідає за допомогою миші (відмічаючи відповіді в закритих питаннях) і клавіатури (для введення тексту у відкритих питаннях). Використання Web-сторінки дозволяє одразу надіслати результати до бази даних і автоматично їх обробити. Недоліки такого опитування пов'язані з необхідністю оволодіння технічними навичками для програмування Web-сторінок, складністю контролю за тривалістю заповнення анкети і кількістю виправлень у відповідях. Для короткотривалих і пошукових досліджень Web-сторінки надзвичайно ефективні. Приблизно 80 % соціологічної інформації з Інтернету збирається саме за допомогою цього методу. Проте можливість сформувати якісну вибірку і цілеспрямовано залучити респондентів буде мінімальною. Досліднику доведеться просто чекати на дані, які будуть з'являтися в ході випадкового і добровільного заповнення анкети

відвідувачами сайту. Ця методика використовується для реєстрації користувачів, неглибоких опитувань і маркетингових досліджень.

Наступний різновид опитування має назву Web-додаток, або веб-опитувач – збір даних відбувається за допомогою спеціальної програми, яка розміщується на сайті і містить кілька сторінок. Під час запуску програми з'являється сторінка з анотацією дослідження і інструкцією до анкети. Потім респондент відповідає на питання, які розділені на блоки. Використання HTML-формату, як і у попередньому випадку, дозволяє отримати чимало переваг, особливо це стосується оформлення анкети, для якої можна застосувати красиві шрифти, графіку і навести чимало пояснень для респондентів[9]. Разом з тим робота за цією технологією потребує більше часу для того, щоб розпочати опитування і доволі складна з точки зору технічних навичок дослідника. Зазвичай розробкою таких програм займаються окремі компанії, що значно збільшує собівартість дослідження.

Якщо ж досліднику необхідно отримати якісну інформацію, то йому слід звернутися до методу онлайн-фокус-групи. Респонденти через Інтернет підключаються до сеансу дискусії, знаходять на екранах моніторів запропоновані модератором питання і потім вводять власні відповіді. Як і у традиційних фокус-групах керівну функцію виконує модератор, вимоги до якого в даному випадку суттєво зростають. Робота онлайн-фокус-групи часто ускладнюється тим, що в дискусії може приймати участь користувач зі слабкими навичками комп'ютерного набору, що потребує більших часових витрат. Ще один недолік полягає у тому, що учасники не можуть у повній мірі відчувати якість групового обговорення. Фактично, дослідник отримує інформацію з кількох глибинних інтерв'ю одночасно.

### **1.3 Обґрунтування актуальності дослідження**

Практична частина моєї дипломної роботи полягає у створенні веб-додатку для проведення інтернет опитувань. Я обрав спеціалізацію на опитуванні, що може бути інтегрованим у будь-який сайт інтернет-магазину.

Наразі існує декілька підходів до втілення таких опитувань. Це:

- створення опитування для конкретного сайту;
- інтеграція у веб-сторінку сайту опитувача від стороннього розробника;
- використання окремо розміщеної сторінки із опитуванням.

Обидва підходи містять як суттєві переваги, так і недоліки. Створення опитування для свого сайту є доволі дорогим процесом, оскільки кожен сайт повинен оплачувати роботу програміста від початку і до кінця. Це об'єднує в собі верстку веб-сторінки, роботи на фронтенді та бекенді, створення й підтримку бази даних. Проте з іншого боку цей підхід дозволяє зробити чудову інтеграцію з веб-сайтом, стилізація може допомогти опитуванню сприйматись як органічна частина веб-сайту, а не сторонній елемент. Через це такий підхід можуть собі дозволити лише великі й ресурсоємні інтернет магазини, для яких переваги від якісно зробленого опитування перевершують недолік у вигляді ціни.

Другий підхід більш характерний для невеликих інтернет-магазинів. Часто використовується інтеграція на сайт опитувача від GoogleФорм [10]. Такий підхід є більш компромісним: опитування все ще здійснюється прямо на сайті, однак вибір оформлення є значно вужчим, через що таке опитування виглядає як чужорідний елемент на сторінці. До того ж, присутня помітна кількість згадок GoogleФорм на сторінці, що візуально засмічує веб-сторінку. Однак створення таких опитувань взагалі не вимагає роботи спеціаліста в сфері айті, від нього потрібна лише інтеграція елемента на веб-сторінку.

Третій підхід є підходом з мінімальними витратами. Як створення, так і розміщення опитування за певним URL бере на себе компанія-провайдер послуг (часто ті самі GoogleФорми). Таким чином проведення, зберігання та аналіз результатів опитування взагалі не вимагає спеціальних знань у сфері

інформаційних технологій. Однак цей варіант має й ряд недоліків: сторінка із опитуванням взагалі не є частиною веб-сайту, а отже – має інший інтерфейс, оформлення та слабко асоціюється із інтернет-магазином. Часто такий підхід до опитування клієнтів обирають інтернет-магазини, які не мають власного сайту й розміщують свій контент та інформацію про товари в соціальних мережах.

Аналіз наданої вище інформації дозволяє зробити висновок, що наразі є затребуваним програмний продукт, який одночасно мінімізує потребу спеціальних знань у сфері інформаційних технологій та матиме широкі можливості з адаптації стилю до стилю сайту, полегшуючи тим самим інтеграцію веб-додатку на веб-сайт компанії або інтернет магазину.

#### **1.4 Постановка задачі**

Для виконання дипломної роботи необхідно:

- провести дослідження літератури та аналіз існуючих веб-додатків зі схожою тематикою;
- спроектувати веб-додаток для проведення опитувань;
- розробити веб-додаток для проведення опитувань мовою Python із використання фреймворку Django;
- забезпечити можливість візуальної інтеграції до існуючих сайтів з допомогою стилів;
- розробити автоматизовані тести для даного додатку;
- розробити адміністративну частину для керування питаннями та варіантами відповідей.

#### **Висновки за загальним розділом**

В розділі описаний об'єкт дослідження, а саме термін веб-додаток, особливості використання фреймворку Django, описано структурні та функціональні особливості додатку для розробки.

Проаналізована література за темою онлайн опитувань допомогла визначити точний напрямок розробки, що допоможе їй закрити собою брак пропозиції у вигляді веб опитувачів, що можуть бути повністю інтегровані до сайту без залучення значних витрат людей зі спеціальними знаннями у сфері комп'ютерних наук.

Окрім того, завдяки аналізу джерел було визначено напрям, в якому використання інтернет опитувань є найбільш виправданим – опитування користувачів інтернет ресурсів, оскільки вони точно є цільовою аудиторією для даних сервісів і результати таких опитувань будуть достатньо репрезентативними.

Також у загальній частині була повністю обґрунтована доцільність та актуальність даного дослідження.

## 2 СПЕЦІАЛЬНИЙ РОЗДІЛ

### 2.1 Проектування інформаційної системи

#### 2.1.1 Загальні положення

Проектування інформаційної системи є одним з головних етапів при розробці проекту в сфері інформаційних технологій. Проектування інформаційних систем - це впорядкована сукупність методологій і засобів створення або модернізації інформаційних систем. В свою чергу, інформаційна система - сукупність інформації, що міститься в базах даних та інформаційних технологій і технічних засобів, що забезпечують її обробку.

За позиціями деяких науковців, етап проектування передує етапу моделювання інформаційних систем, проте інші визначають моделювання як складову частину проектування.

Процес проектування інформаційної системи для мене складався із таких етапів

- аналіз завдання;
- аналіз отриманих теоретичних та практичних даних, що стосувались систем онлайн опитування;
- виокремлення характеристик інформаційної системи, яку має бути створено;
- створення інформаційної моделі системи із використанням UML засобів;
- проектування операцій, які будуть виконуватись із даними;
- проектування програмного забезпечення.

В ході аналізу було сформовано такі вимоги до програмного продукту:

- наявність бази даних;
- доступ до програми через браузер;
- можливість інтеграції до інших сайтів;
- наявність користувацького та адміністративного веб інтерфейсів;

- можливість керувати переліком питань використовуючи веб інтерфейс.

### 2.1.2 Інформаційна модель системи

Є декілька підходів до сприйняття поняття інформаційної моделі. За одним, інформаційна модель — модель об'єкта, представлена у вигляді інформації, що зображує істотні для даного розгляду параметри та змінні величини об'єкта, зв'язки між ними, входи і виходи об'єкта і дозволяє шляхом подачі на модель вхідних величин моделювати можливі стани об'єкта [11]. Інший підхід описує інформаційну модель як сукупність інформації, що характеризує істотні властивості і стани об'єкта, процесу, явища, а також взаємозв'язок із зовнішнім світом.

Базуючись на цих визначеннях, створено декілька інформаційних моделей, які в сукупності дають повне уявлення про зміст дипломної роботи.

В першу чергу, необхідно було змоделювати представлення вебдодатку з точки зору користувача. Принципово можна виділити два основних складника такого представлення. Це:

- користувацька частина – така, що доступна кожному користувачу вебдодатку;
- адміністративна частина – ця частина доступна лише обмеженому колу людей, які мають логін та пароль для входу.

Графічно модель зображено на рис. 2.1.

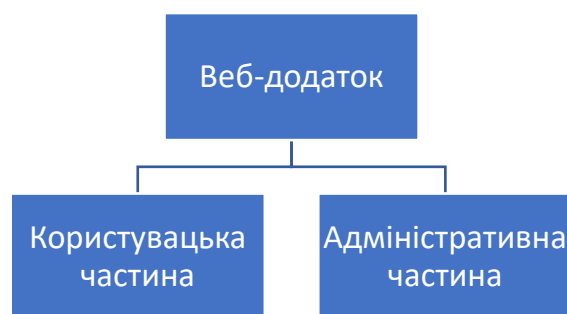


Рисунок 2.1 – Модель загального користувацького представлення веб додатку

В свою чергу, користувацька частина складається із таких типів сторінок:

- перелік питань;
- сторінка-опитувач;
- сторінка з результатами опитувань.

Графічне зображення моделі користувацької частини зображено на рис. 2.2.

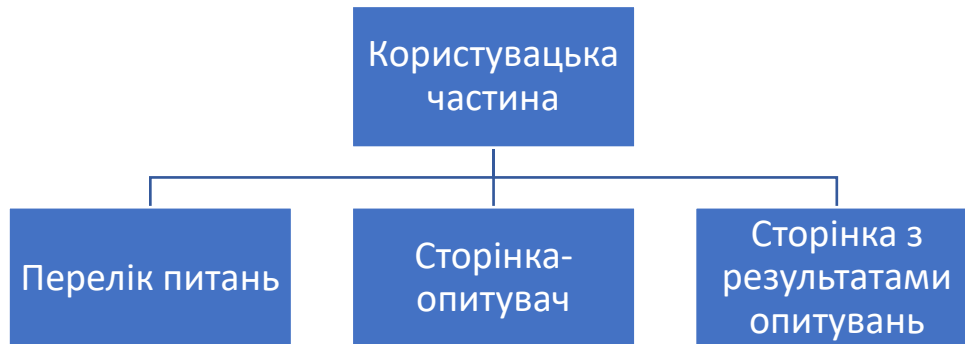


Рисунок 2.2 – Склад користувацької частини вебдодатку

Перелік питань має містити заголовок, перелік питань із гіперпосиланнями для можливості дати відповідь на конкретне питання та гіперпосилання для переходу до сторінки із результатами опитувань.

Сторінка-опитувач має містити текст питання, варіанти відповіді, кнопку для надсилання відповіді, можливість переходу до переліку питань та сторінки із результатами опитувань.

Сторінка з результатами опитувань повинна містити заголовок, перелік питань із варіантами відповідей та інформацію про кількість голосів, віддану за кожен із варіантів.

Адміністративна частина складається із:

- сторінки авторизації;
- головної сторінки;
- сторінки із переліком питань;
- сторінки для редагування питань та варіантів відповідей;
- сторінки для створення питань із варіантами відповідей;
- сторінки для зміни паролю адміністратора.

Структура адміністративної частини вебдодатку зображена на рисунку 2.3.

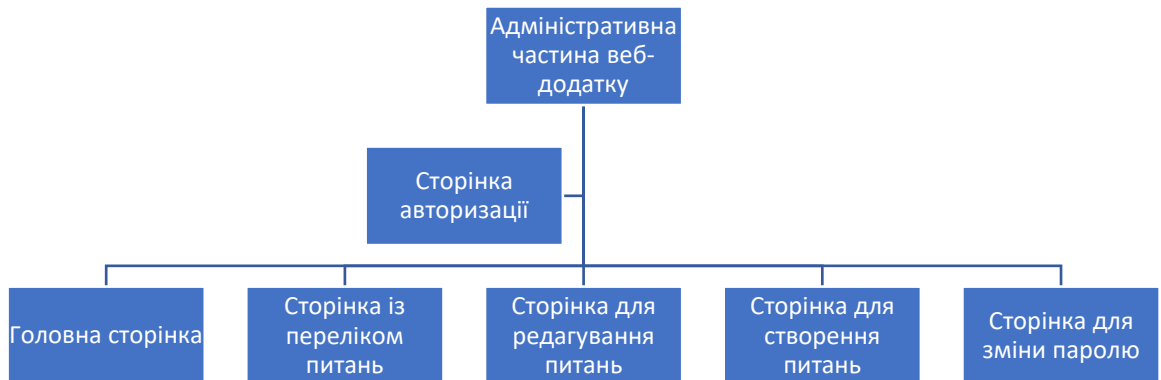


Рисунок 2.3 – Склад адміністративної частини вебдодатку.

## 2.2 Інформаційне забезпечення

Інформаційне забезпечення – сукупність даних та правил їх отримання, організації структури та зміст інформаційних сукупностей, зберігання та оновлення даних, методи класифікації і кодування інформації.

### 2.2.1 Структура і схеми інформаційних об'єктів і ресурсів

Інформаційний об'єкт – сформовані за певними правилами дані, які тлумачаться інформаційною системою як єдине ціле. Сукупність інформаційних об'єктів утворює змістове наповнення інформаційної системи.

В даній роботі ключовими інформаційними об'єктами є:

- база даних;
- сторінки веб додатку.

Системою керування базами даних в даному проекті виступила SQLite – полегшена реляційна система управління реляційними базами даних, що міститься в бібліотеці C. SQLite зазвичай дотримується синтаксису PostgreSQL.

Зі сторінок вебдодатку користувач як отримує дані про питання, варіанти відповіді на них та статистику відповідей, так і передає дані про свій варіант відповіді. Адміністратор може створювати, видаляти й редагувати ці сутності.

### **2.2.2 Схеми інформаційних потоків**

Інформаційним потоком називають рух інформації, спрямований від джерела інформації до отримувача, визначений функціональними зв'язками між ними.

В дипломному проекті можна спостерігати такі інформаційні потоки:

- на сторінці із переліком питань:
  - читання даних про тексти питань;
- на сторінці питання:
  - читання даних про текст та варіанти відповіді;
  - запис даних про надану відповідь;
- на сторінці статистики відповідей:
  - читання даних про питання, варіанти відповідей та кількість голосів за кожно з відповідей.

### **2.2.3 Схеми бази даних**

Схема баз даних – це структура системи баз даних описана формальною мовою, яка підтримується системою керування баз даних і відноситься до

організації даних для створення плану побудови бази даних з розподілом на таблиці.

В даному випадку була відсутня необхідність моделювання багатьох сутностей і я обмежився створенням однієї колекції, що мала таку структуру:

- ідентифікатор запису;
- текст питання;
- дата й час публікації;
- масив варіантів відповідей:
  - текст варіанту відповіді;
  - кількість голосів, віддана за даний варіант відповіді.

Для візуалізації структури даних в базі було створено UML діаграму в онлайн редакторі [app.diagrams.net](http://app.diagrams.net) [12] (рис. 2.4).

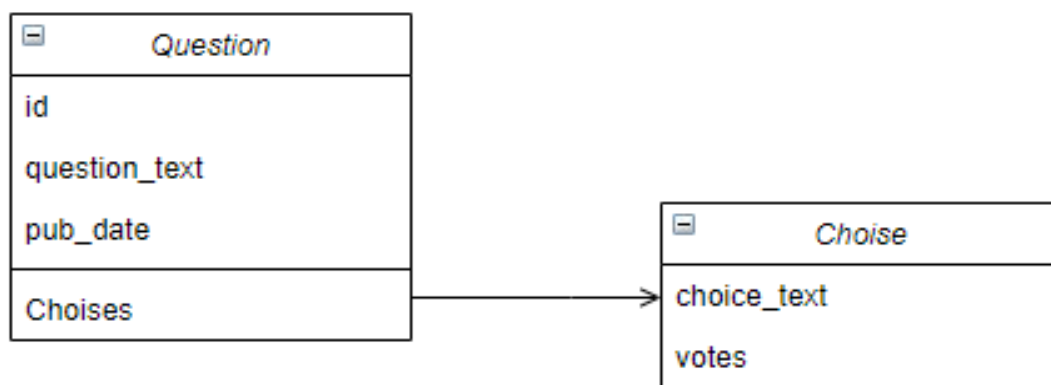


Рисунок 2.4 – UML діаграма структури БД

До даних у базі даних виставлено такі вимоги:

- текст запитання – символічне поле, обмеження у 200 символів;
- дата публікації – поле типу DateTimeField;
- текст варіанту відповіді – символічне поле, обмеження у 200 символів;
- кількість голосів за варіант відповіді – цілочисельне поле, за замовчуванням встановлюється значення 0.

#### 2.2.4 Опис бази даних

База даних – це організований набір даних, які, як правило, зберігаються та мають електронний доступ з комп'ютерної системи. Там, де бази даних є більш складними, вони часто розробляються з використанням формальних методів проектування та моделювання.

Система управління базами даних (СУБД) – це програмне забезпечення, яке взаємодіє з кінцевими користувачами, програмами та самою базою даних для збору та аналізу даних. Програмне забезпечення СУБД додатково охоплює основні засоби, що надаються для адміністрування бази даних. Загальну суму бази даних, СУБД та пов'язаних із нею додатків можна назвати «системою баз даних». Часто термін «база даних» також використовується для вільного посилання на будь-яку СУБД, систему баз даних або додаток, пов'язаний з базою даних.

В цьому проекті було використано систему управління базами даних SQLite. Такий вибір було зроблено через легкість та швидкість роботи з такою системою управління баз даних. Більше того, вона не потребує установки та адміністрування, а отже є чудовим вибором для використання у проекті, що вимагає мінімуму спеціальних знань про інформаційні технології.

Мова програмування Python має вбудовану бібліотеку для взаємодії з базами даних SQLite. Тому робота у проекті мовою Python з використанням фреймворку Django з SQLite є автоматизованою та нативною, що також зменшило витрати людиногодина на виконання проекту. Django підтримує SQLite 3.9.0 і новіших версій.

Дані в SQLite зберігаються в таблицях. Для даного проекту біло прийняте рішення використовувати дві пов'язані між собою таблиці: Question та доповнюючу їй – Choice.

## 2.3 Алгоритмічне забезпечення

Алгоритм – набір інструкцій, які описують порядок дій виконавця, щоб досягти результату розв'язання задачі за скінченну кількість дій; система правил виконання дискретного процесу, яка досягає поставленої мети за скінченний час.

Завдання дипломної роботи полягало у створенні вебдодатку для створення опитувань, тому вважаю доцільним зосередитись у цій частині звіту на ключовому для користувача алгоритмі – проходженні опитування.

Потрапляючи на сторінку із переліком питань, користувач має вибір: обрати конкретне питання для надання відповіді, або пройти голосування повністю. Залежно від обраного варіанту, на сервер надсилається запит із номером питання. Першого питання, якщо обрано проходження опитування повністю та конкретного питання у випадку обрання питання вручну.

На основі цього Django, використовуючи шаблон сторінки для голосування, читає з бази даних текст запитання та варіанти відповіді.

Потрапляючи на сторінку із певним питанням, користувач має обрати варіант відповіді та натиснути на «проголосувати». У випадку, якщо варіант відповіді не обрано, на сторінці з'являється сповіщення про це.

Також зі сторінки із питанням користувач може перейти до статистики відповідей або повернутись до списку питань.

Після обрання відповіді на запитання, Django виконує запис інформації про надану відповідь в базу даних та перенаправляє користувача на сторінку із наступним питанням. Після обрання відповіді на останнє питання з бази даних користувач потрапляє на сторінку із результатами опитування.

Окремо існує алгоритм додавання питань та варіантів відповідей. Для цього адміністратору необхідно авторизуватись за адресою <http://127.0.0.1:8000/admin>, використовуючи логін та пароль адміністратора (суперкористувача за термінологією Django). Після цього перейти за гіперпосиланням на сторінку додавання/редагування питань. На цій сторінці доступні варіанти дій – це:

- видалення існуючого питання;
- редагування існуючого питання, редагування, додавання або видалення варіантів відповідей та статистики відповідей;
- додавання нового питання із варіантами відповідей на нього.

Таке алгоритмічне забезпечення дає необхідний і достатній функціонал для проведення опитувань за допомогою веб додатку.

### **Висновки за спеціальним розділом**

В даному розділі було розкрито тему інформаційної складової проекту, описано хід та результат проектування інформаційної системи дипломного проекту, описано та зображено інформаційну модель системи, описано інформаційне забезпечення. Також було розкрито тему алгоритмічного забезпечення системи.

Розкрито теми структури та схем інформаційних об'єктів і ресурсів. Зображено та описано схеми інформаційних потоків, опис і схеми баз даних.

Після роботи над даним розділом отримано чітке бачення, з яких елементів має складатись дипломний проект, які алгоритми повинен у собі мати, обрано засоби для реалізації проекту та якою має бути структура даних і елементів користувацького інтерфейсу.

## 3 РОЗРАХУНКОВИЙ РОЗДІЛ

### 3.1 Програмне забезпечення

Програмне забезпечення – набір даних та інструкцій, які повідомляють обчислювальній машині, які операції виконувати. На противагу апаратному забезпеченню, з допомогою котрого система фактично працює. В комп'ютерних науках та інформатиці програмне забезпечення у широкому розумінні – це весь обсяг інформації, який передається на обробку комп'ютерним системам, це і програми, і дані. Отже, програмне забезпечення включає програми, фреймворки, бібліотеки та пов'язані з ними дані котрі не виконуються, наприклад такі як документація на онлайн або фізичних носіях. Апаратне та програмне забезпечення потребують одне одного для роботи, і жодне з них не може бути застосоване окремо.

На найнижчому рівні програмування виконуваний код складається з інструкцій машинної мови, що підтримуються окремим процесором - зазвичай це виконується центральним процесором (CPU) чи графічним процесором (GPU). Машинна мова є набором двійкових значень, що містять інструкції для процесора, які змінюють стан обчислюваної машини від попереднього стану. Зокрема, інструкція може змінити значення, яке зберігається в певному місці пам'яті комп'ютера – це є ефектом, який найчастіше не спостерігається користувачем безпосередньо. Інструкція може також викликати певну операцію вводу або виводу, наприклад, відображення певного тексту або зображення на екрані комп'ютера. Таким чином викликаються зміни стану, видимі для користувача. Процесор виконує інструкції в тому порядку, в якому вони надані, за виключенням випадків, коли йому наказано «перейти» до іншої інструкції або операційна система не перериває його. Станом на 2021 рік абсолютна більшість персональних комп'ютерів, смартфонів, серверів та інших цифрових гаджетів мають процесори з декількома ядрами або декількома процесорами, що виконують обчислення одночасно і обчислення стають набагато більш швидкою дією, аніж у минулому.

Більшу частину програмного забезпечення написано високорівневими мовами програмування. Вони простіші та зручніші для програмістів, оскільки вони ближчі до людських мов, аніж ніж машинні мови програмування. Високорівневі мови програмування переводяться у машинну мову з допомогою компілятора, інтерпретатора чи їх поєднання.

Найчастіше програмне забезпечення поділяють на три групи:

- системне програмне забезпечення;
- прикладне програмне забезпечення.

Далі за текстом буде розглянуто основні риси даних видів програмного забезпечення та описано ПО, яке було використано в даному проекті.

### **3.1.1 Системне програмне забезпечення**

Системне програмне забезпечення є основним ПО, невід'ємною частиною комп'ютера. Без нього неможливо взаємодіяти ні з одним пристроєм електронної обчислювальної машини. Саме системне ПО керує злагодженою роботою всіх елементів комп'ютерної системи, як на апаратному рівні, так і на програмному.

До системного програмного забезпечення ПК включають операційні системи, системи програмування (що складається з мови програмування, компілятора/інтерпретатора програм), сервісні програми. Основне призначення цих програм – забезпечення можливості спілкування користувача з комп'ютером, надання програмістам інструментарію для створення прикладних програм.

Дипломний проект створено з використанням операційної системи Windows 10, однак програма готова до хостингу не лише у Windows, а й в Linux системах та Mac OS. Веб додатком після розміщення на хостингу можна користуватись ще і на мобільних операційних системах, оскільки додаток виконується у браузері і є сумісним із мобільними версіями браузерів.

Окрім операційної системи, в проєкті використовувались мова програмування та інтерпретатор Python, фреймворк Django.

### **3.1.2 Прикладне програмне забезпечення**

До прикладного програмного забезпечення відносять програми, що безпосередньо забезпечують виконання конкретних завдань по обробці даних.

Розходження між окремими класами програмного забезпечення є дуже умовними, особливо в даний час, коли переважає прагнення до створення універсальних інтегрованих пакетів програм.

Для виконання дипломної роботи мною було використано таке прикладне програмне забезпечення:

- текстовий процесор Microsoft Word;
- IDE Visual Studio Code;
- веббраузери Google Chrome, Mozilla Firefox;
- вебдодаток для створення UML діаграм [app.diagrams.net/](http://app.diagrams.net/);
- переглядач та редактор PDF файлів ABBYY FineReader;
- програма для створення презентацій Microsoft PowerPoint;
- вебдодаток - перекладач [translate.google.com](http://translate.google.com);
- програма для перевірки унікальності тексту AntiPlagiarism.NET;
- текстовий редактор Блокнот;
- командний рядок Windows.

### **3.2 Технічне забезпечення**

Технічне забезпечення – це комплекс технічних засобів збирання, накопичення, обробки, передачі, ведення та подання інформації, пристроїв

управління ними. Сюди ж належать засоби оргтехніки, призначені для організації тривалого збереження (накопичення) інформації і здійснення інформаційного обміну між різними технічними засобами.

Даний дипломний проект має такі вимоги до технічного забезпечення:

- для хостингу:
  - операційна система Windows/Linux/MacOS;
  - наявність встановленого інтерпретатора Python;
  - наявність встановленого фреймворку Django;
  - доступ до Інтернету;
- для користування веб-додатком:
  - операційна система Windows/Linux/MacOS/Android/IOS;
  - наявність веб браузеру.

### **3.3 Процес розробки системи**

Процес розробки програмного забезпечення – сукупність ряду послідовних дій, спрямованих на розробку програмного забезпечення (ПЗ). Процес втілення результатів моделювання складався із описаних нижче ключових етапів.

#### **3.3.1 Встановлення необхідного для розробки програмного забезпечення**

Програмне забезпечення, необхідне для розробки веб-додатку – IDE та веббраузер. В якості IDE було обрано Visual Studio Code – засіб для створення, редагування та зневадження сучасних вебзастосунків і програм для хмарних систем. Visual Studio Code розповсюджується безкоштовно і доступний у версіях для платформ Windows, Linux і OS X.

В якості браузера було обрано Google Chrome. Це безкоштовний веббраузер, розроблений компанією Google на основі браузера з відкритим кодом Chromium та іншого відкритого програмного забезпечення.

### **3.3.2 Встановлення програмного середовища Python**

Python – інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. Розроблена в 1990 році Гвідо ван Россумом. Встановлення Python на Windows відбувається з допомогою програми-візарда, а на ОС Linux – з магазину застосунків відповідного дистрибутиву [13].

### **3.3.3 Встановлення фреймворку Django**

Встановлення фреймворку Django відбувається в командному рядку або терміналі [14]. Для встановлення актуальної версії Django достатньо ввести код

```
python -m pip install Django
```

### **3.3.4 Створення проекту Django**

Створення проекту також відбувається у командному рядку. Для цього потрібно ввести команду

```
django-admin startproject <Назва проекту>
```

В результаті утвориться така структура папок та файлів:

```
diploma/  
  manage.py  
diploma/  
  __init__.py  
  settings.py
```

urls.py

asgi.py

wsgi.py

### **3.3.5 Запуск та тестування серверу для розробки**

Для запуску серверу в Django необхідно ввести в термінал команду

```
python manage.py runserver
```

В результаті запуститься локальний сервер, за замовчуванням – за адресою `http://127.0.0.1:8000/`. Однак номер порту можна замінити з допомогою параметру, що додається до попередньої команди.

### **3.3.6 Створення веб-додатку**

Для створення веб додатку потрібно ввести в терміналі команду

```
python manage.py startapp <назва застосунку>
```

Це створить такий набір каталогів та файлів:

polls/

    \_\_init\_\_.py

    admin.py

    apps.py

    migrations/

        \_\_init\_\_.py

    models.py

    tests.py

    views.py

### **3.3.7 Налаштування бази даних**

Для створення та оновлення баз даних в Django використовується команда `python manage.py migrate`

Команда `migrate` переглядає настройку `INSTALLED_APPS` і створює всі необхідні таблиці бази даних відповідно до настройками бази даних в вашому файлі `diploma / settings.py` і міграціями бази даних, що поставляються з додатком. Повідомлення виводиться для кожної застосовної міграції.

### **3.3.8 Створення представлень (views)**

Для створення представлень вносяться зміни до файлу `polls/views.py`. Файл вже створено за замовчуванням, однак для налаштування роботи його потрібно редагувати.

### **3.3.9 Роутинг**

Роутинг описується у файлі `polls/urls.py`. В цьому файлі створюються класи для представлень, що використовуються в проекті.

### **3.3.10 Редагування можливостей адміністратора**

Для створення адміністратора необхідно виконати команду `python manage.py createsuperuser`

Після цього діалог пропонує ввести ім'я, адресу електронної пошти та двічі – пароль адміністратора.

### **3.3.11 Створення шаблонів**

Для того, щоб створити шаблон, який буде використовуватись веб застосунком, необхідно створити каталог `templates` в каталозі із застосунком (в даному випадку – `polls`). В цій папці необхідно створити ще один катало із назвою аналогічною назві застосунку. Тобто, шаблони мають знаходитись за адресою `polls/templates/polls/`.

Самі шаблони пишуться мовою шаблонів Django. Шаблон Django – це текстовий документ або рядок Python, розмічений із використанням мови шаблонів Django. Деякі конструкції розпізнаються і інтерпретуються механізмом шаблонів. Основні з них - змінні і теги.

### **3.3.12 Створення автоматизованих тестів**

Фреймворк Django має вбудований функціонал для зручної роботи з автоматизованими тестами. За замовчуванням, кожен застосунок містить файл `tests.py`, звідки виконуються тести. Запуск тестів виконується командою

```
python manage.py test polls
```

### **3.3.13 Застосування до сторінок стилів**

Для розміщення файлів зі стилями, зображеннями та іншим статичним вмістом необхідно в каталозі застосунку створити каталог `static`. В ньому створюємо каталог із назвою застосунку (в даному випадку – `polls`), всередині якого можемо розмістити файл зі стилями. Використання фреймворку Django не зумовлює необхідність будь-якого специфічного оформлення CSS файлів [15].

## **3.4 Опис програми**

Програма являє собою веб додаток. Вона займає на носії 187 кілобайт пам'яті. Код основних файлів програми наведено у додатку А.

На рисунку 3.1 наведено структуру файлів програми.

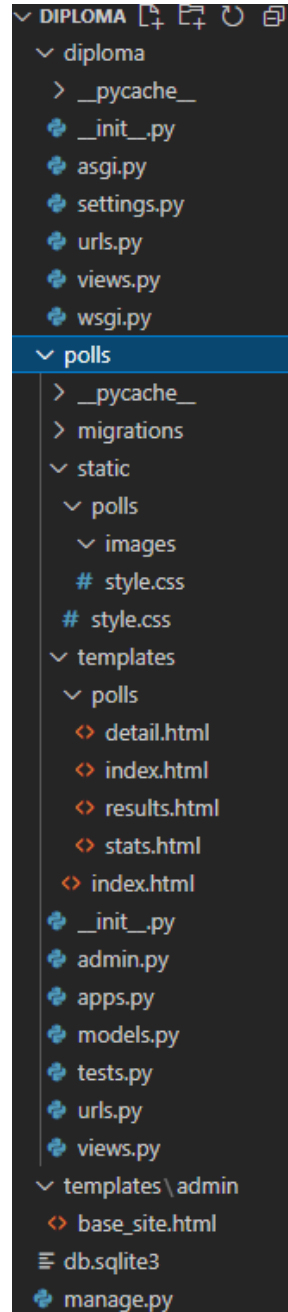


Рисунок 3.1 – Структура файлів програми у вікні IDE

Зовнішній кореневий каталог `diploma/` – це контейнер для проекту.  
`manage.py` – утиліта, що дозволяє взаємодіяти з проектом.

Внутрішній каталог `diploma/` – це Python-модуль проекту. Його назва використовується для імпорту об'єктів з цього модуля (наприклад, `diploma.urls`).

`diploma/__init__.py` – порожній файл, який повідомляє Python, що цей каталог повинен розглядатися як пакет Python'a.

`diploma/settings.py` – конфігурація і налаштування проекту Django.

`diploma/urls.py` – вказівка URL проекту на Django, можна сказати, що це «зміст» проекту, або файл роутингу.

`diploma/asgi.py` – точка входу для ASGI-сумісних веб-серверів для обслуговування проекту.

`diploma/wsgi.py` – точка входу для WSGI сумісних веб-серверів для роботи з проектом.

Каталог `diploma/polls` є каталогом вебдодатку, який є складовою частиною проекту. Файли, назви яких дублюють назви файлів у попередньому каталозі, мають таке саме призначення, як і там, однак виконують вказані функції не для цілого проекту, а лише для веб додатку.

`diploma/polls/admin.py` дозволяє керувати адміністративною частиною проекту.

`diploma/polls/apps.py` – файл для конфігурації Django застосунків.

`diploma/polls/models.py` є файлом для опису моделей бази даних.

`diploma/polls/tests.py` – файл для написання автоматизованих тестів додатку.

Каталог `diploma/polls/static` містить статичні файли веб додатку, такі, як CSS файли, зображення, тощо.

Каталог `diploma/polls/templates` містить HTML файли, що є Python шаблонами сторінок вебдодатку.

### **3.5 Інструкції з використання програми**

#### **3.5.1 Інструкція системного адміністратора**

Для запуску веб додатку на пристрої, що виконуватиме роль сервера, необхідно виконати таку послідовність дій:

- встановити актуальну версію Python;
- встановити актуальну версію Django;
- запустити сервер командою `python manage.py runserver <номер порту>`.

### 3.5.2 Інструкція користувача

Для роботи з веб додатком необхідно відкрити через веб браузер адресу `http://127.0.0.1/` (з-за умови запуску програми локально) або за обраним доменним ім'ям чи IP адресою у випадку розміщення програми в мережі Інтернет. Після цього користувач потрапить на стартову сторінку проекту, яка містить інформацію про автора роботи (рисунк 3.2).

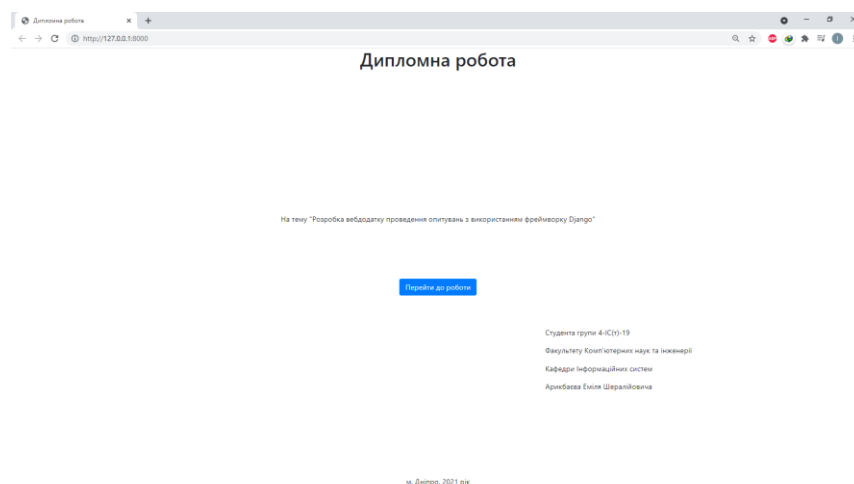


Рисунок 3.2 – Титульна сторінка веб проекту

На цій сторінці міститься інформація про розробника та про те, що даний проект є дипломною роботою на тему «Розробка вебдодатку проведення опитувань з використанням фреймворку Django».

Для переходу до переліку питань користувач має натиснути на стилізоване під кнопку гіперпосилання з текстом «Перейти до роботи». Після цього відкриється сторінка з переліком питань, зображена на рисунку 3.3.

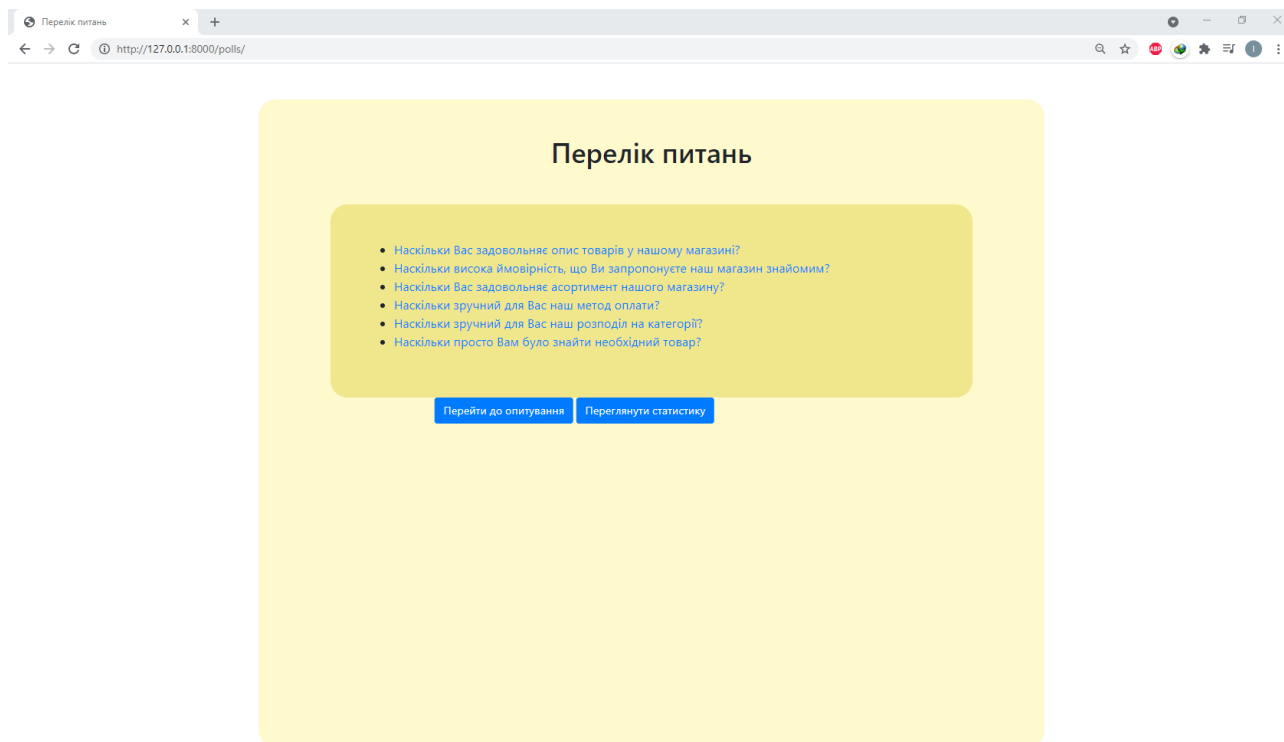


Рисунок 3.3 – Сторінка вебдодатку із переліком питань

На цій сторінці користувач має можливість на вибір:

– натиснути «перейти до опитування», після чого він потрапить до першого питання. Надалі, натискаючи на «проголосувати» після обрання варіанту відповіді користувача буде перенаправляти на наступне питання аж до останнього в списку, після чого користувача буде направлено до сторінки зі статистикою відповідей;

– натиснути на текст конкретного питання, відповідь на яке користувач хоче надати;

– перейти за посиланням «переглянути статистику».

Нижче можна ознайомитись із зовнішнім виглядом сторінок із питаннями (рисунок 3.4) та сторінки зі статистикою відповідей (рисунок 3.5).

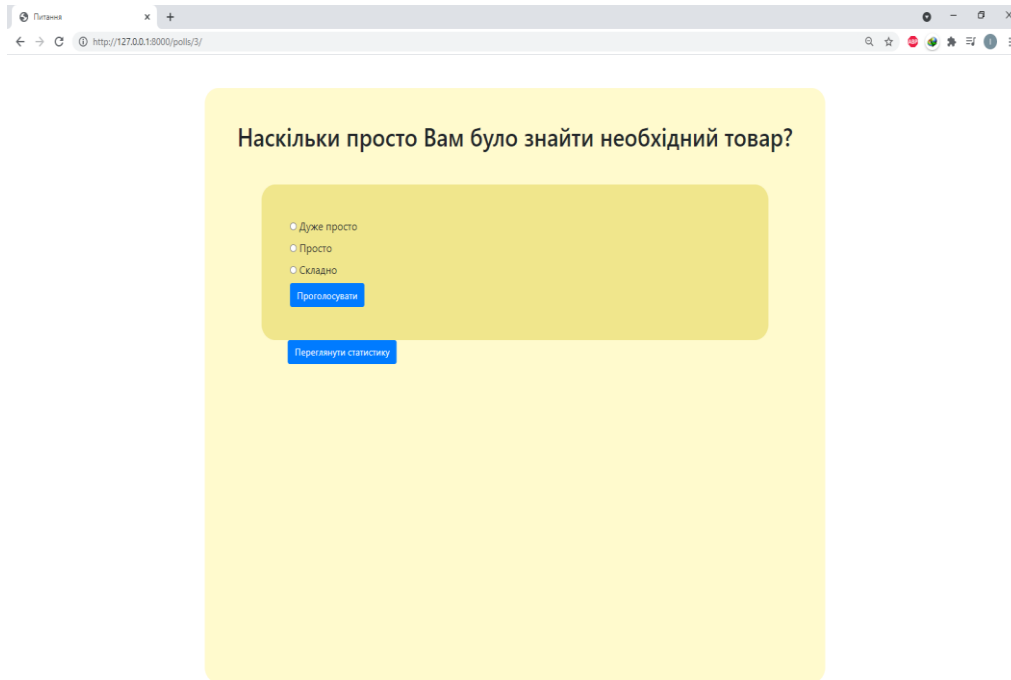


Рисунок 3.4 – Сторінка із питанням та варіантами відповідей

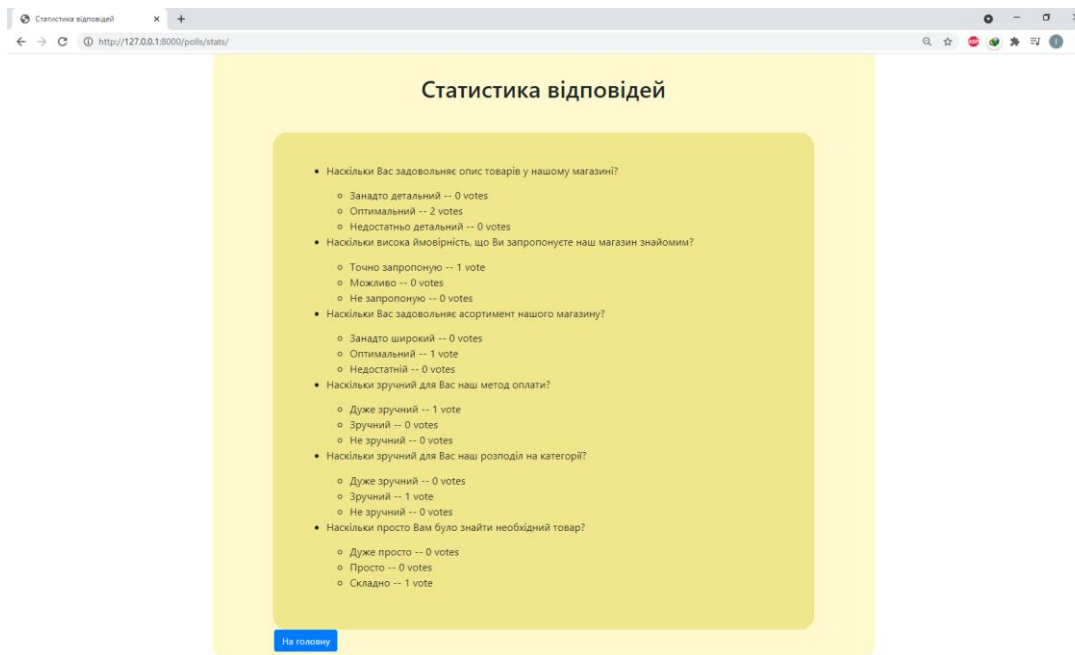


Рисунок 3.5 – Сторінка зі статистикою відповідей

Також варто зазначити, що при спробі натиснути на кнопку «Проголосувати», не обираючи варіант відповіді для голосування, користувач

буде сповіщений про це і додаток не переходитиме на сторінку із наступним питанням (або статистикою відповідей). Сповіщення зображено на рисунку 3.6.

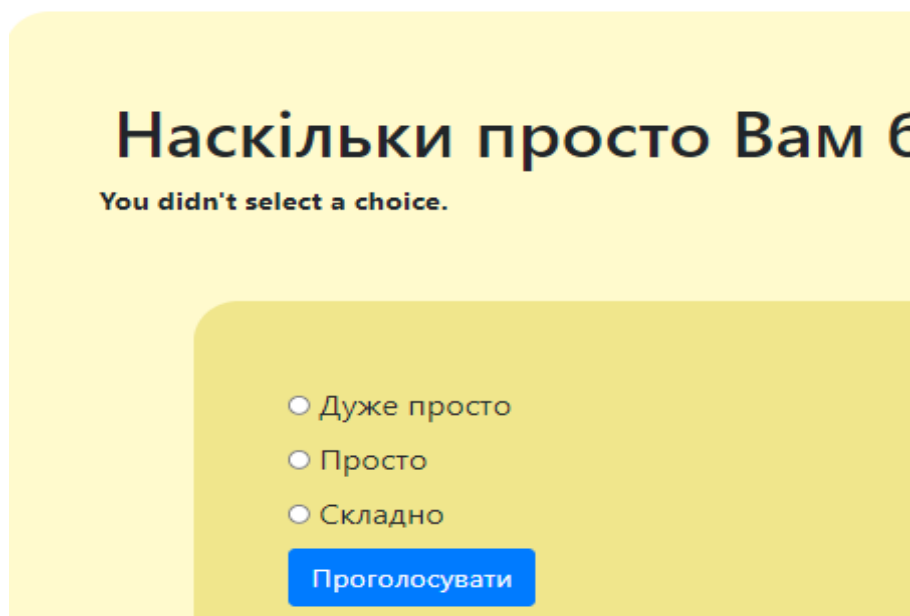


Рисунок 3.6 – Сповіщення про те, що не обрано варіанту відповіді.

### 3.5.3 Інструкція користувача із правами адміністратора

Оскільки при розробці даної програми завданням було зробити додаток для проведення веб опитувань, який не вимагав би спеціальних знань у сфері інформаційних технологій від адміністратора, даний застосунок має окрему адміністративну панель з можливістю додавати, видаляти, редагувати чи відкладати публікацію питань і варіантів відповіді до них.

Для входу в систему в режимі адміністратора необхідно перейти за адресою <http://127.0.0.1:<номер порту>/admin/>.

Після цього користувач, якщо він не був авторизований раніше, потрапить на сторінку авторизації (рис. 3.7). Для входу необхідно заповнити поля Username та Password.

Після введення імені користувача та пароля, адміністратор потрапить на початкову сторінку панелі адміністратора (рисунок 3.8).

Рисунок 3.7 – Форма авторизації адміністратора

**Polls Administration Panel**

Site administration

**AUTHENTICATION AND AUTHORIZATION**

Groups	+ Add	Change
Users	+ Add	Change

**POLLS**

Questions	+ Add	Change
-----------	-------	--------

**Recent actions**

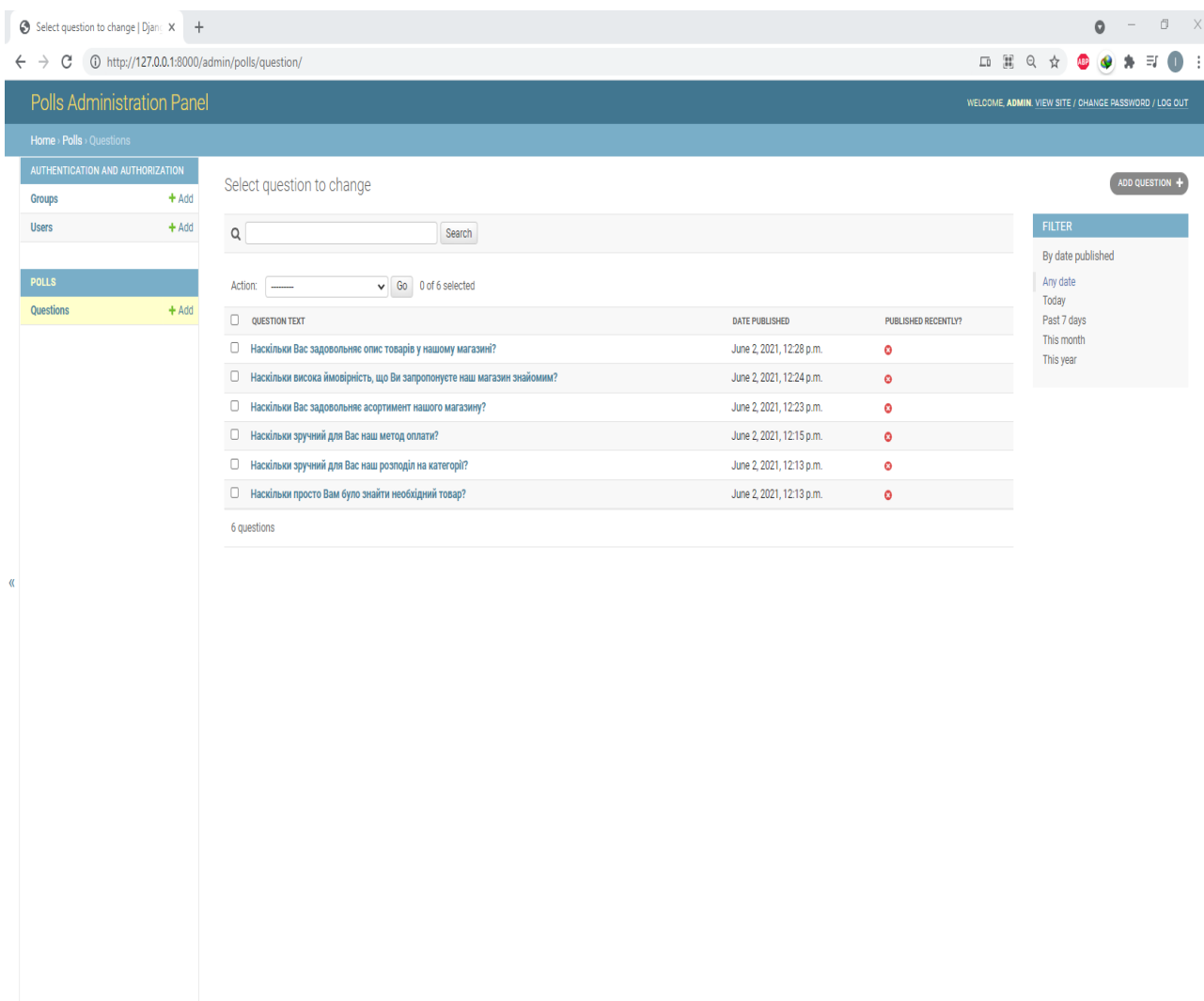
**My actions**

- Наскільки просто Вам було знайти необхідний товар? Question
- Наскільки Вас задовольняє опис товарів у нашому магазині? Question
- Наскільки висока ймовірність, що Ви запропонуєте наш магазин знайомим? Question
- Наскільки зручний для Вас наш розподіл на категорії? Question
- Наскільки зручний для Вас наш метод оплати? Question
- Наскільки Вас задовольняє асортимент нашого магазину? Question
- Наскільки зручний для вас наш метод оплати? Question
- Наскільки зручний для вас наш розподіл на категорії? Question
- Наскільки просто Вам було знайти необхідний товар? Question
- What's new? Question

### Рисунок 3.8 – Головна сторінка панелі адміністратора

Праворуч на панелі відображено перелік останніх дій адміністратора. Для операцій із питаннями та варіантами відповідей (видалення, додавання, редагування) необхідно натиснути на гіперпосилання «Questions» з лівої сторони.

Натиснувши на це посилання, адміністратор переходить до сторінки <http://127.0.0.1:<номер порту>/admin/polls/question/> (рисунок 3.9).



### Рисунок 3.9 – Сторінка для операцій із питаннями

Це – сторінка веб додатку, що має найширший функціонал. Так, праворуч розташовано блок, у якому можна застосувати до питань фільтр за датою

публікації. Вище цього блока знаходиться кнопка, натиснувши на яку адміністратор опиниться на сторінці з формою для додавання питання та варіантів відповідей (рисунок 3.10).

Над переліком питань розташовано поле для пошуку за текстом запитання та випадаючий список для обрання варіантів дій з обраними з допомогою checkbox питань. Натиснувши на текст питання, адміністратор перейде до сторінки редагування вже існуючого питання (рисунок 3.11).

The screenshot shows a web browser window with the URL `http://127.0.0.1:8000/admin/polls/question/add/`. The page title is "Add question | Django site admin". The main content area is titled "Polls Administration Panel" and contains the following elements:

- A breadcrumb trail: Home > Polls > Questions > Add question.
- A sidebar menu with sections: "AUTHENTICATION AND AUTHORIZATION" (Groups, Users) and "POLLS" (Questions).
- A form titled "Add question" with a "Question text:" input field.
- A "Date information (Show)" section.
- A table titled "CHOICES" with columns: CHOICE TEXT, VOTES, and DELETE?.
- Three rows in the table, each with an input field for choice text, a "0" in the votes column, and a delete icon.
- A link "+ Add another Choice" below the table.
- Three buttons at the bottom: "Save and add another", "Save and continue editing", and "SAVE".

Рисунок 3.10 – Сторінка для додавання питань та варіантів відповідей

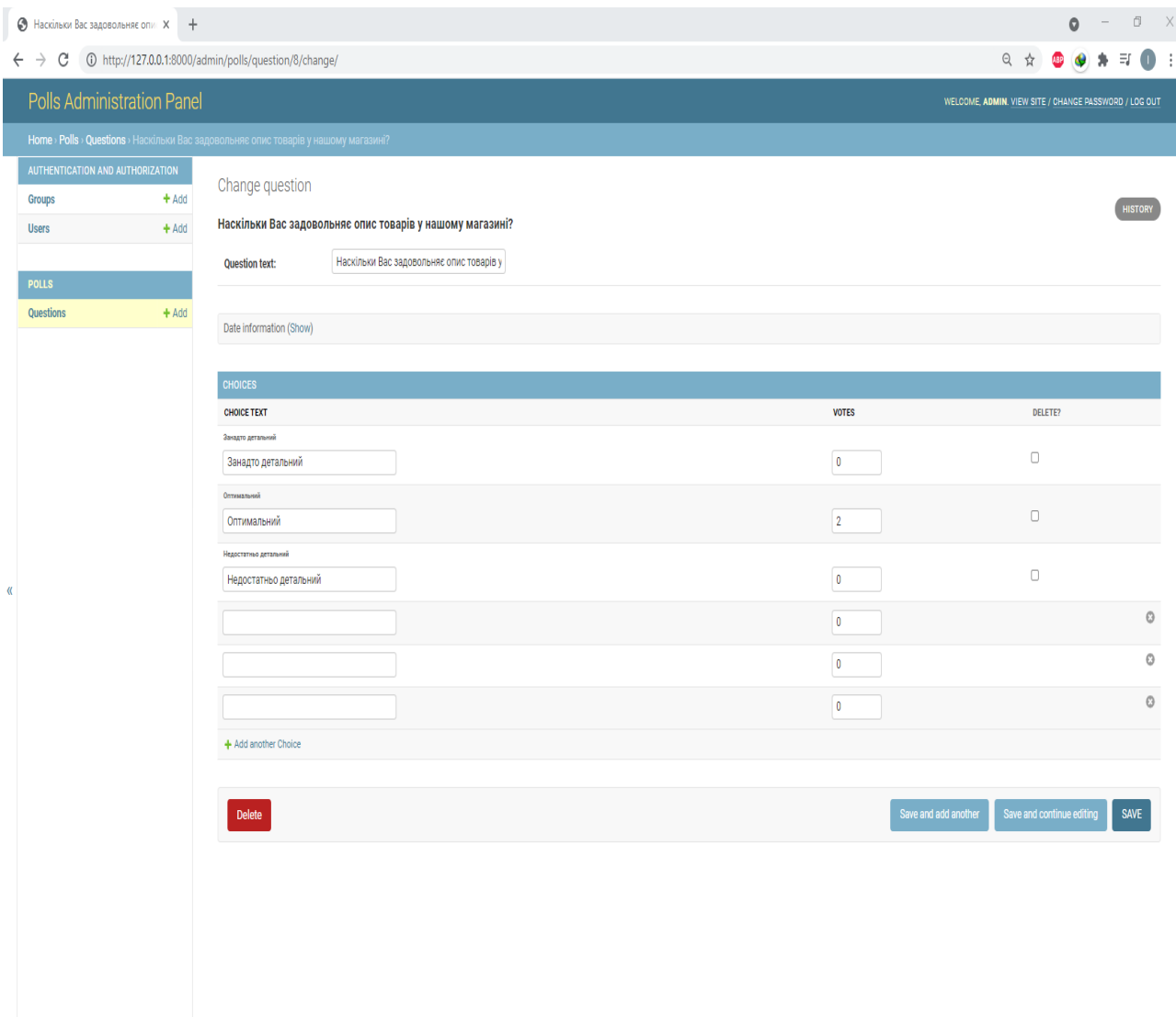


Рисунок 3.11 – Сторінка для редагування питання

Особливістю також є те, що адміністратор може змінювати значення кількості голосів за будь-який з варіантів питання – як при створенні, так і при редагуванні питання.

Таким чином, користувач із правами адміністратора має можливість здійснювати із питаннями та відповідями на них будь-які операції: створення, читання, редагування, видалення. До того ж, при створенні питання є можливість додати дату й час публікації в майбутньому, що дозволить завчасно планувати зміни до опитування (наприклад, після запланованої зміни інтерфейсу або акції в інтернет магазині).

## **Висновки за розрахунковим розділом**

В даному розділі було розглянуто програмне забезпечення, зокрема системне, прикладне та технічне забезпечення, що використовувалось для розробки даного проекту та яке необхідне для запуску програми. Було висвітлено процес розробки системи. Надано опис програми та інструкції з використання програми системним адміністратором, користувачем та користувачем із правами адміністратора.

Показано процес реалізації веб додатку для проведення опитувань, написаного мовою програмування Python із використанням фреймворку Django.

## ВИСНОВКИ

В результаті виконання дипломної роботи було:

- проведено аналіз літературних джерел та існуючих веб-додатків для проведення опитувань. Зроблено висновок про перспективність напрямку розробки та про відсутність на ринку веб-опитувачів, що могли б бути інтегровані до будь-якого веб-сайту із використанням спеціаліста лише для візуального оформлення опитувача;
- спроектовано веб-додаток для проведення опитувань;
- розроблено веб-додаток для проведення опитувань мовою Python із використанням фреймворку Django з можливістю візуальної інтеграції до існуючих сайтів з допомогою стилів;
- розроблено адміністративну частину для керування питаннями та варіантами відповідей;
- створено файл CSS для демонстрації можливості візуальної адаптації опитувача до існуючого сайту.

В загальному розділі був описаний об'єкт дослідження, а саме термін веб-додаток, особливості використання фреймворку Django, описано структурні та функціональні особливості додатку для розробки.

Проаналізована література за темою онлайн опитувань допомогла визначити точний напрямок розробки, який допоможе закрити собою брак пропозиції у вигляді веб опитувачів, що можуть бути повністю інтегровані до сайту без залучення значних витрат людей зі спеціальними знаннями у сфері комп'ютерних наук.

Окрім того, завдяки літературі було визначено напрям, в якому використання інтернет опитувань є найбільш виправданим – опитування користувачів інтернет ресурсів, оскільки вони точно є цільовою аудиторією для даних сервісів і результати таких опитувань будуть достатньо репрезентативними.

Також у загальній частині повністю обґрунтована доцільність та актуальність даного дослідження. Таким чином вважаю, що мною було досягнуто виконання мети дипломного проекту в повному обсязі. Мною було створено програмний продукт, веб-додаток із опитуванням, який може бути інтегрованим на веб сайт інтернет магазину, який можна адмініструвати без спеціальних знань в сфері інформаційних технологій та візуально адаптувати до стилю інтернет-магазину з допомогою CSS.

У спеціальному розділі розглянуто інформаційну складову проекту, описано хід та результат проектування інформаційної системи дипломного проекту, описано та зображено інформаційну модель системи, описано інформаційне забезпечення; вирішене питання алгоритмічного забезпечення системи. Досліджено структури та схеми інформаційних об'єктів і ресурсів, описано схеми інформаційних потоків, схеми баз даних.

У розрахунковому розділі розглянуто програмне забезпечення, зокрема системне, прикладне та технічне забезпечення, використане для розробки даного проекту та необхідне для запуску програми; висвітлено процес розробки системи; описані програми та інструкції з використання програми системним адміністратором, користувачем та користувачем із правами адміністратора.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Седухіна І. Продається все: про тренди в сфері ритейлу і як пандемія змінила формат торгівлі. // Mind.ua, 2020. Ел ресурс. Режим доступу: <https://mind.ua/openmind/20220034-prodaetsya-vse-pro-trendi-v-sferi-ritejlu>.
2. Грабченко А.І., Федорович В.О., Гаращенко Я.М. Методи наукових досліджень: навчальний посібник. – Харків: НТУ ХПІ, 2009. – 142 с.
3. Blakea В.Ф., Neuendorfb К.А., Valdiserri С.М. Tailoring new websites to appeal to those most likely to shop online. // Technovation. – V. 25. – Is. 10. – p. 1205-1214.
4. Документація Django (dev) на русском языке. Версия 3.2. Ел ресурс. Режим доступу: <https://django.fun/docs/django/ru/3.2/>.
5. Online Test Pad. Ел ресурс. Режим доступу: <https://onlinetestpad.com/>.
6. Psychological tests. Online assessment. Ел ресурс. Режим доступу: <https://psyttests.org/>.
7. Гапоненко Д.І. Інтернет-опитування як метод збору соціологічної інформації // Вісник Чернігівського національного педагогічного університету. – 2012. – №103. Ел ресурс. Режим доступу: [http://erpub.chnpu.edu.ua:8080/jspui/bitstream/123456789/321/1/Gaponenko\\_Internet%20surveys%20as%20a%20method%20of%20collecting%20sociological%20information.pdf](http://erpub.chnpu.edu.ua:8080/jspui/bitstream/123456789/321/1/Gaponenko_Internet%20surveys%20as%20a%20method%20of%20collecting%20sociological%20information.pdf)
8. Andrews D., Nonnecke B., Preece J. Conducting Research on the Internet: Online Survey Design, Development and Implementation Guidelines. // International Journal of Human-Computer Interaction. – 2003. – V. 16. – Is. 2. – p. 185-210.
9. HTML: HyperText Markup Language. Resources for Developers, by Developer. Ел ресурс. Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/HTML>.
10. Google Форми. Ел ресурс. Режим доступу: [https://www.google.com/intl/uk\\_ua/forms/about/](https://www.google.com/intl/uk_ua/forms/about/).

11. Mysities. Інформаційна модель. Ел ресурс. Режим доступу:  
<https://sites.google.com/site/mysities02/informacijna-model/>.

12. Веб додаток для створення UML діаграм. Ел ресурс. Режим доступу:  
<https://app.diagrams.net>.

13. Офіційний сайт Python. Ел ресурс. Режим доступу:  
<https://www.python.org/>.

14. Django Web Framework (Python). Ел ресурс. Режим доступу:  
<https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django>.

15. Django Packages. Admin styling. Ел ресурс. Режим доступу:  
<https://djangopackages.org/grids/g/admin-styling/>.

## ДОДАТОК А

### Лістинг програмного коду

#### Модуль manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys

def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'diploma.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()
```

#### Модуль diploma/urls.py

```
from django.contrib import admin
from django.urls import include, path
from . import views
# from . import models

urlpatterns = [
    path('', views.index, name='index'),
    path('polls/', include('polls.urls')),
    path('admin/', admin.site.urls),
]
```

## Продовження додатку А

### Модуль diploma/polls/admin.py

```
from django.contrib import admin

from .models import Choice, Question

class ChoiceInline(admin.TabularInline):
    model = Choice
    extra = 3

class QuestionAdmin(admin.ModelAdmin):
    fieldsets = [
        (None, {'fields': ['question_text']}),
        ('Date information', {'fields': ['pub_date'], 'classes': ['collapse']}),
    ]
    inlines = [ChoiceInline]
    list_display = ('question_text', 'pub_date', 'was_published_recently')
    list_filter = ['pub_date']
    search_fields = ['question_text']

admin.site.register(Question, QuestionAdmin)
```

### Модуль diploma/polls/models.py

```
from django.db import models
import datetime
from django.utils import timezone
from django.contrib import admin

class Question(models.Model):
    question_text = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')
    def __str__(self):
        return self.question_text
    @admin.display(
        boolean=True,
        ordering='pub_date',
        description='Published recently?',
    )
```

## Продовження додатку А

```
def was_published_recently(self):
    now = timezone.now()
    return now - datetime.timedelta(days=1) <= self.pub_date <= now
```

```
class Choice(models.Model):
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
    choice_text = models.CharField(max_length=200)
    votes = models.IntegerField(default=0)
    def __str__(self):
        return self.choice_text
```

## Модуль diploma/polls/tests.py

```
import datetime
from django.urls import reverse
from django.test import TestCase
from django.utils import timezone

from .models import Question

class QuestionModelTests(TestCase):

    def test_was_published_recently_with_future_question(self):
        """
        was_published_recently() returns False for questions whose pub_date
        is in the future.
        """
        time = timezone.now() + datetime.timedelta(days=30)
        future_question = Question(pub_date=time)
        self.assertIs(future_question.was_published_recently(), False)

    def test_was_published_recently_with_old_question(self):
        """
        was_published_recently() returns False for questions whose pub_date
        is older than 1 day.
        """
        time = timezone.now() - datetime.timedelta(days=1, seconds=1)
```

## Продовження додатку А

```

old_question = Question(pub_date=time)
self.assertIs(old_question.was_published_recently(), False)

def test_was_published_recently_with_recent_question(self):
    """
    was_published_recently() returns True for questions whose pub_date
    is within the last day.
    """
    time = timezone.now() - datetime.timedelta(hours=23, minutes=59, seconds=59
)
    recent_question = Question(pub_date=time)
    self.assertIs(recent_question.was_published_recently(), True)

def create_question(question_text, days):
    """
    Create a question with the given `question_text` and published the
    given number of `days` offset to now (negative for questions published
    in the past, positive for questions that have yet to be published).
    """
    time = timezone.now() + datetime.timedelta(days=days)
    return Question.objects.create(question_text=question_text, pub_date=time)

class QuestionIndexViewTests(TestCase):

    def test_no_questions(self):
        """
        If no questions exist, an appropriate message is displayed.
        """
        response = self.client.get(reverse('polls:index'))
        self.assertEqual(response.status_code, 200)
        self.assertContains(response, "No polls are available.")
        self.assertQuerysetEqual(response.context['latest_question_list'], [])

    def test_past_question(self):
        """
        Questions with a pub_date in the past are displayed on the
        index page.
        """
        question = create_question(question_text="Past question.", days=-30)
        response = self.client.get(reverse('polls:index'))
        self.assertQuerysetEqual(
            response.context['latest_question_list'],
            [question],
        )

```

Продовження додатку А

```

def test_future_question(self):
    """
    Questions with a pub_date in the future aren't displayed on
    the index page.
    """
    create_question(question_text="Future question.", days=30)
    response = self.client.get(reverse('polls:index'))
    self.assertContains(response, "No polls are available.")
    self.assertQuerysetEqual(response.context['latest_question_list'], [])

def test_future_question_and_past_question(self):
    """
    Even if both past and future questions exist, only past questions
    are displayed.
    """
    question = create_question(question_text="Past question.", days=-30)
    create_question(question_text="Future question.", days=30)
    response = self.client.get(reverse('polls:index'))
    self.assertQuerysetEqual(
        response.context['latest_question_list'],
        [question],
    )

def test_two_past_questions(self):
    """
    The questions index page may display multiple questions.
    """
    question1 = create_question(question_text="Past question 1.", days=-30)
    question2 = create_question(question_text="Past question 2.", days=-5)
    response = self.client.get(reverse('polls:index'))
    self.assertQuerysetEqual(
        response.context['latest_question_list'],
        [question2, question1],
    )

class QuestionDetailViewTests(TestCase):
    def test_future_question(self):
        """
        The detail view of a question with a pub_date in the future
        returns a 404 not found.
        """
        future_question = create_question(question_text='Future question.', days=5)
        url = reverse('polls:detail', args=(future_question.id,))
        response = self.client.get(url)

```

```

        self.assertEqual(response.status_code, 404)

    def test_past_question(self):
        """
        The detail view of a question with a pub_date in the past
        displays the question's text.
        """
        past_question = create_question(question_text='Past Question.', days=-5)
        url = reverse('polls:detail', args=(past_question.id,))
        response = self.client.get(url)
        self.assertContains(response, past_question.question_text)

```

### Модуль diploma/polls/static/style.css

```

.container {
    margin-left: auto;
    margin-right: auto;
    width: 6em;
    height: 100%;
}
h1 {
    text-align: center;
}
.theme {
    text-align: center;
    margin-top: 30vh;
}
.maker {
    float: right;
    margin-top: 20vh;
}
footer {
    text-align: center;
    margin-top: calc(70vh - 180px);
}
.button-container {
    margin: 0;
    position: absolute;
    top: 50%;
    left: 50%;
    -ms-transform: translate(-50%, -50%);
    transform: translate(-50%, -50%);
}

```

## Продовження додатку А

### Модуль diploma/polls/static/polls/style.css

```
.container {  
  background-color: lemonchiffon;  
  min-height: 90vh;  
  margin-top: 5vh;  
  border-radius: 25px;  
  padding: 5vh;  
}
```

```
.questions-container {  
  margin-top: 5vh;  
  margin-left: 5vh;  
  margin-right: 5vh;  
  background-color: khaki;  
  font-size: large;  
  
  padding: 5vh;  
  border-radius: 25px;  
}
```

```
h1 {  
  text-align: center;  
}
```

```
.buttons-container {  
  margin: 0;  
  position: absolute;  
  transform: translate(50%, 0);  
  vertical-align: bottom;  
}
```

```
form {  
  margin-top: 5vh;  
  margin-left: 5vh;  
  margin-right: 5vh;  
  background-color: khaki;  
  font-size: large;  
  padding: 5vh;  
  border-radius: 25px;  
}
```

```
label {  
  font-size: large;  
}
```

## Продовження додатку А

### Модуль diploma/polls/templates/index.html

```
<!DOCTYPE html>
<html lang="uk-ua">
<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>Дипломна робота</title>
  {% load static %}
  <link rel="stylesheet" type="text/css" href="{% static 'style.css' %}">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/
css/bootstrap.min.css">
</head>
<body>
  <div class="container">
    <h1>Дипломна робота</h1>
    <div class="theme">
      <p>На тему "Розробка вебдодатку проведення опитувань з використанням фр
еймворку Django"</p>
    </div>
    <div class="maker">
      <p>Студента групи 4-ІС(τ)-19</p>
      <p>Факультету Комп'ютерних наук та інженерії</p>
      <p>Кафедри Інформаційних систем</p>
      <p>Арикбаєва Еміля Шералійовича</p>
    </div>
    <div class="button-container"><a href="polls" class="btn btn-
primary">Перейти до роботи</a></div>
  <footer>
    <p>м. Дніпро, 2021 рік</p>
  </footer>
</div>
</body>
</html>
```

## Продовження додатку А

## Модуль diploma/polls/templates/polls/index.html

```
<!DOCTYPE html>
<html lang="en">
<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">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/
css/bootstrap.min.css">
  {% load static %}
  <link rel="stylesheet" type="text/css" href="{% static 'polls/style.css' %}">
  <title>Перелік питань</title>
</head>
<body>
  <div class="container">
    <div class="h"><h1>Перелік питань</h1></div>
    <div class="questions-container">{% if latest_question_list %}
    <ul>
      {% for question in latest_question_list %}
      <li><a href="{% url 'polls:detail' question.id %}">{{ question.question
_text }}</a></li>
      {% endfor %}
    </ul>
    {% else %}
    <p>Немає доступних питань.</p>
    {% endif %}
    </div>
    <div class="buttons-container"><a href="3" class="btn btn-
primary">Перейти до опитування</a>
    <a href="stats" class="btn btn-primary">Переглянути статистику</a></div>
  </div>
</body>
</html>
```

Модуль diploma/polls/templates/polls/detail.html

```
<!DOCTYPE html>
<html lang="en">
<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">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/
css/bootstrap.min.css">
    {% load static %}
    <link rel="stylesheet" type="text/css" href="{% static 'polls/style.css' %}
">
  <title>Питання</title>
</head>
<body>
  <div class="container">
    <h1>{{ question.question_text }}</h1>

    {% if error_message %}<p><strong>{{ error_message }}</strong></p>{% endif %}
  }

  <form action="{% url 'polls:vote' question.id %}" method="post">
    {% csrf_token %}
    {% for choice in question.choice_set.all %}
    <input type="radio" name="choice" id="choice{{ forloop.counter }}" valu
e="{{ choice.id }}">
    <label for="choice{{ forloop.counter }}">{{ choice.choice_text }}</labe
l><br>
    {% endfor %}
    <input type="submit" class="btn btn-primary" value="Проголосувати">
  </form>
  <div class="buttons-container">
    <a href=" ../stats" class="btn btn-
primary">Переглянути статистику</a>
  </div>
</div>

</body>
</html>
```

## Продовження додатку А

### Модуль diploma/polls/templates/polls/stats.html

```
<!DOCTYPE html>
<html lang="en">
<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">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/
css/bootstrap.min.css">
  {% load static %}
  <link rel="stylesheet" type="text/css" href="{% static 'polls/style.css' %}">
  <title>Статистика відповідей</title>
</head>
<body>
  <div class="container">
    <div class="h"><h1>Статистика відповідей</h1></div>
    <div class="questions-container">{% if latest_question_list %}
    <ul>
      {% for question in latest_question_list %}
      <li>
        <p>{{ question.question_text }}</p>
        <ul>
          {% for choice in question.choice_set.all %}
          <li>{{ choice.choice_text }} -
- {{ choice.votes }} vote{{ choice.votes|pluralize }}</li>
          {% endfor %}
        </ul>
      </li>
      {% endfor %}
    </ul>
    {% else %}
    <p>Немає доступних питань.</p>
    {% endif %}
  </div>
  <div class="buttons-container">
    <a href=".." class="btn btn-primary">На головну</a>
  </div>
</div>
</body>
</html>
```