

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет Комп'ютерних наук та кібернетики  
Кафедра Теорії та технології програмування

**Кваліфікаційна робота  
на здобуття ступеня бакалавра**

За спеціальністю 122 Комп'ютерні науки  
на тему:

**ВЕБЗАСТОСУНОК ДЛЯ ПРОВЕДЕННЯ СОЦІАЛЬНИХ ДОСЛІДЖЕНЬ**

Виконала студентка 4-го курсу  
Валерія КАНДИБА

Підпис: В. КАНДИБА

Науковий керівник:  
доцент, кандидат фіз. - мат. наук  
Людмила ОМЕЛЬЧУК

Підпис: Л. ОМЕЛЬЧУК

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

Студент

Підпис: В. КАНДИБА

Роботу розглянуто й допущено до захисту  
на засіданні кафедри теорії та технології  
програмування

«05» червня 2023 р.,

протокол № 18

Завідувач кафедри

Микола НІКІТЧЕНКО

Підпис: М. НІКІТЧЕНКО

Київ – 2023 рік

## РЕФЕРАТ

Загальний обсяг роботи 58 сторінок, основний текст викладено на 55 сторінках, 22 ілюстрації, 20 використаних джерел, 13 таблиць, 3 додатки.

JAVA, SPRING BOOT, POSTGRESQL, HIBERNATE, TYPESCRIPT, POSTMAN, ВЕБЗАСТОСУНОК, ANGULAR.

Об'єктом дослідження даної роботи є процес створення опитувальників. Предметом роботи є вебзастосунок для автоматизації процесу проведення опитувань.

Метою роботи є розробка функціонального інструменту для створення опитувальників. Ця робота спрямована на те, щоб користувачі могли легко створювати, налаштовувати та збирати дані опитування, що дозволить ефективно проводити дослідження, збирати відгуки та отримувати важливу інформацію. Однією з потенційних цілей є полегшення та автоматизація процесу створення опитувальників для різних потреб, забезпечуючи гнучкість і простоту у використанні.

Методи розробки: розробка клієнт-серверного застосунку. Інструменти розробки: Java 8, Spring Framework, Hibernate, JUnit, OAuth2, PostgreSQL, Angular Framework, TypeScript, IntelliJ IDEA 2022.3.2.

Результат роботи: виконано загальний огляд технологій, які використовуються при розробці вебсайтів, розроблено вебсайт для створення опитувальників, впроваджено систему збереження та обробки відповідей користувачів. Створений сайт відрізняється від найближчих популярних аналогів наявністю діаграм статистики відповідей, які дають краще візуальне розуміння питань, за якими були створені опитувальники. Такий сайт може бути корисним у різних сферах, таких як академічна сфера, бізнес та маркетинг, освітні заходи, компанії та організації, соціальні дослідження.

## ЗМІСТ

СКРОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ.....	5
ВСТУП.....	6
РОЗДІЛ 1. ОГЛЯД НАЯВНИХ НА РИНКУ СЕРВІСІВ ДЛЯ СТВОРЕННЯ ОНЛАЙН-ОПИТУВАНЬ.....	9
1.1 Конструктор «Google Forms» .....	9
1.2 Конструктор «Jotform» .....	10
1.3 Конструктор «SurveyMonkey» .....	11
1.4 Порівняння розглянутих рішень .....	12
РОЗДІЛ 2. ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ.....	14
2.1 Вибір мови програмування серверної частини .....	14
2.2 Вибір технології взаємодії з базою даних .....	15
2.3 Вибір мови програмування клієнтської частини .....	15
РОЗДІЛ 3. ОСНОВНІ МЕТОДИ ТА ТЕХНОЛОГІЇ ДЛЯ РОЗРОБКИ ВЕБСАЙТІВ .....	17
3.1 Мова програмування Java.....	17
3.2 Фреймворк Spring Boot Framework .....	18
3.3 Фреймворк Hibernate .....	18
3.4 Фреймворк JUnit .....	19
3.5 Протокол OAuth 2 .....	19
3.6 Система керування базами даних PostgreSQL .....	20
3.7 Фреймворк Angular Framework .....	21
3.8 Мова програмування TypeScript .....	21
РОЗДІЛ 4. ПРИЗНАЧЕННЯ І ЦІЛІ РОЗРОБКИ ЗАСТОСУНКУ .....	23
4.1 Призначення застосунку.....	23
4.2 Цілі розробки застосунку .....	24
4.3 Вимоги до застосунку.....	24
4.3.1 Вимоги до застосунку в цілому .....	24
4.3.2 Вимоги до функцій, які виконуються програмою.....	25
4.3.3 Технічні вимоги до застосунку .....	26
РОЗДІЛ 5. РЕАЛІЗАЦІЯ ВЕБСАЙТУ .....	28

5.1	Опис організації інформаційної бази.....	28
5.1.1	Логічна структура бази даних .....	28
5.1.2	Опис таблиць бази даних програми .....	29
5.2	Розробка вебзастосунку.....	34
5.3	Постановка завдання.....	35
5.4	Структура програми.....	36
5.3.1	Серверна частина .....	36
5.3.2	Клієнтська частина.....	39
5.3.3	Інтеграція зі стороннім сервісом .....	39
5.4	Тестування сайту.....	39
5.4.1	Мануальне тестування .....	40
5.4.2	Тестування API .....	40
5.4.3	Модульне тестування.....	41
РОЗДІЛ 6. ІНСТРУКЦІЯ КОРИСТУВАЧА .....		42
ВИСНОВКИ .....		53
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....		54
ДОДАТКИ .....		56
Додаток А. Use-Case діаграма для застосунку .....		56
Додаток Б. Діаграма бази даних застосунку .....		57
Додаток В. Структура програми застосунку .....		58

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

- API – Application Programming Interface (інтерфейс програмування додатків);
- CSS – Cascading Style Sheets (каскадні таблиці стилів);
- CSV – Comma-Separated Values (значення, розділені комами);
- DOM – Document Object Model (модель об'єктів документа);
- HTTP – HyperText Transfer Protocol (протокол передачі гіпертексту);
- IDE – Integrated Development Environment (інтегроване середовище розробки);
- JDBC – Java Database Connectivity (підключення до бази даних Java);
- JVM – Java Virtual Machine (віртуальна машина Java);
- ORM – Object Relational Mapping (об'єктно-реляційне відображення);
- PHP – Hypertext Preprocessor (обробник гіпертексту);
- SQL – Structured Query Language (мова структурованих запитів);
- SPA – Single Page Application (односторінковий застосунок);
- БД – База Даних;
- МБ – Мегабайт;
- ОРСБД – об'єктно-реляційна система керування базами даних;
- СКБД – система керування базами даних.

## ВСТУП

**Оцінка сучасного стану об'єкта розробки.** У сучасному інформаційному суспільстві важливість збору даних і отримання зворотного зв'язку для прийняття обґрунтованих рішень у всіх сферах життя зростає з кожним днем. Опитування є одним із найпоширеніших методів, які використовуються в цьому процесі, але потреба в спеціалізованих інструментах для створення та проведення опитувань стає все більш важливою.

Опитувальники дозволяють зібрати велику кількість даних від широкого кола людей. Вони допомагають отримати систематичну інформацію про переконання, уподобання та ставлення людей до різних явищ та думок. Також їх можна використовувати для проведення контролю.

Створення опитувальників може бути корисним для соціологічних досліджень та для вивчення громадської думки. При проведенні таких досліджень, опитування можуть допомогти збирати дані про соціальні та демографічні характеристики населення, політичні настрої, соціальні проблеми тощо. Наприклад, дослідник може провести опитування, щоб вивчити ставлення молоді до змін клімату або дослідити вплив соціального статусу на сприйняття політики, або визначити ставлення громадськості до змін в законодавстві щодо зброї, міжнародних подій. Також опитувальники можуть бути корисними для того, аби визначити уподобання або потреби різних груп людей. Такий опитувальник можна зробити для співробітників та клієнтів компанії. Наприклад, дізнатись які товари користуються попитом серед покупців, чи задоволені клієнти ставленням співробітників по відношенню до них або як працівники відчують себе у колективі. Ще популярний приклад, де можуть застосовуватись опитувальники – навчальні заклади. Це може бути корисним інструментом для оцінки знань, розуміння та прогресу студентів та учнів у навчання. Опитувальники можуть використовуватись як частина оцінювальної системи або як інструмент самооцінки студентами.

**Актуальність роботи та підстави для її виконання.** Опитувальники – шлях до комунікації [1]. Вони можуть використовуватись у будь-якій сфері. Будь-то продуктова компанія чи школа, опитування мають важливу ціль – покращення умов та прислухання до думок людей.

Будь-яка людина може легко створити опитувальник, де буде зручний та наповнений функціонал, інтуїтивно зрозумілий кожному. Відсутність зайвих деталей та зосередження на основному функціоналі – це суттєва перевага, висунутого у кваліфікаційній роботі, вебсайту. Тому, тема кваліфікаційної роботи є актуальною.

**Мета й завдання роботи.** Метою кваліфікаційної роботи є розробка вебсайту «SURVEY<X>», який може допомогти у створенні опитувальників для різних сфер діяльності людини чи компанії. Для досягнення цієї мети поставлено такі завдання:

- проаналізувати наявні інструменти для створення опитувальників на ринку;
- дослідити застосування різних технологій для проектування та реалізації вебсайту;
- розробити технічне завдання до програмного продукту «SURVEY<X>»;
- розробити інтерфейс та дизайн вебсайту «SURVEY<X>»;
- реалізувати вебсайт для проведення опитувань.

**Об'єкт, методи й засоби розроблення.** Об'єктом розробки програми «SURVEY<X>» є створення опитувальників користувачем, який бажає дізнатись статистику в будь-якій сфері діяльності.

Предметом роботи є вебсайт для забезпечення можливості створення опитувальників різного типу та поширення різним способом з іншими людьми.

Для розробки програмного засобу було вибрано IntelliJ IDEA, що є інтегрованим середовищем розробки Java, розробленим компанією JetBrains [2].

**Можливі сфери застосування.** Вебсайт «SURVEY<X>» може використовуватись у різних сферах: медична, економічна, соціальна, політична, психологічна, освітня, маркетингова, громадська, спортивна, туристична [3].

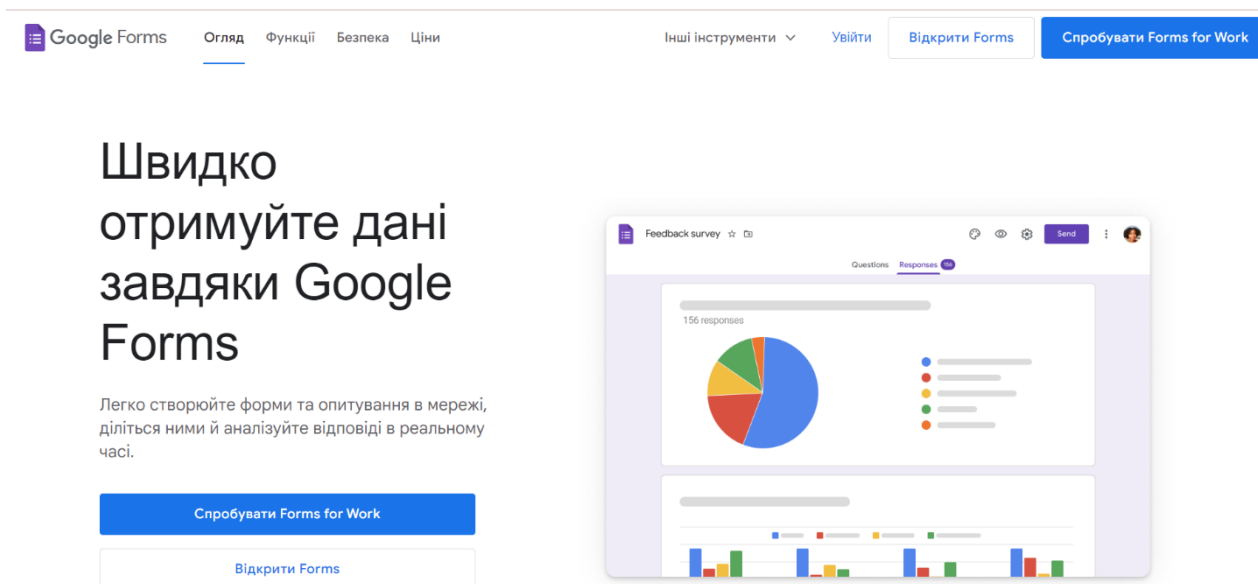
## **РОЗДІЛ 1. ОГЛЯД НАЯВНИХ НА РИНКУ СЕРВІСІВ ДЛЯ СТВОРЕННЯ ОНЛАЙН-ОПИТУВАНЬ.**

Застосунки стали невід'ємною частиною нашого повсякденного життя, спрощуючи багато аспектів нашої роботи, навчання, спілкування та розваг. Їх розмаїття та розвиток в останні роки перетворили їх на потужні інструменти, які змінюють спосіб, яким ми взаємодіємо з технологією та оточуючим нас світом.

### **1.1 Конструктор «Google Forms»**

Це безкоштовний інструмент для створення опитувальників та форм збору даних. З його допомогою можна створити опитування та форми. Їх можна поширювати, аналізувати за допомогою автоматичних зведень статистики [4]. Також розробники стверджують, що створювані опитування можна проходити з будь-якого пристрою, тобто з планшету, з телефону або з комп'ютеру. У цьому конструкторі є можливість придбати пакет зі спеціальними можливостями із взаємодією з іншими сервісами від Google.

Приклад користувацького інтерфейсу «Google Forms» наведено на рис. 1.



**Рисунок 1** – Головна сторінка інструменту «Google Forms»

## 1.2 Конструктор «Jotform»

Конструктор «Jotform» – це безкоштовний конструктор опитувальників який має безліч безкоштовних шаблонів для створення опитувань. Також тут присутня безкоштовна інструмент онлайн опитувань для студентів. У цьому конструкторі є і платні можливості для користувачів: «Bronze», «Silver», «Gold», «Enterprise». Наприклад, Bronze надає ліміт у 25 форм, Silver – 50 форм, Gold – 100 форм та Enterprise – необмежену кількість форм, у той час, коли у безкоштовному тарифі всього лише 5 форм. Також у безкоштовному тарифі існують ліміти у 100 відповідей у місяць, 100 МБ доступного місця та інші можливості, які є доволі малими для людини, яка активно проводить статистику [5].

Згідно з дослідженнями компанії Jotform, користувачі, які активно користуються застосунками, зауважують, що сайт JotForm має привабливий та зручний інтерфейс. Однак, для більш зручного користування, багато з них рекомендують оформити підписку на один з платних тарифів.

Приклад користувацького інтерфейсу сайту «Jotform» наведено на рис. 2.

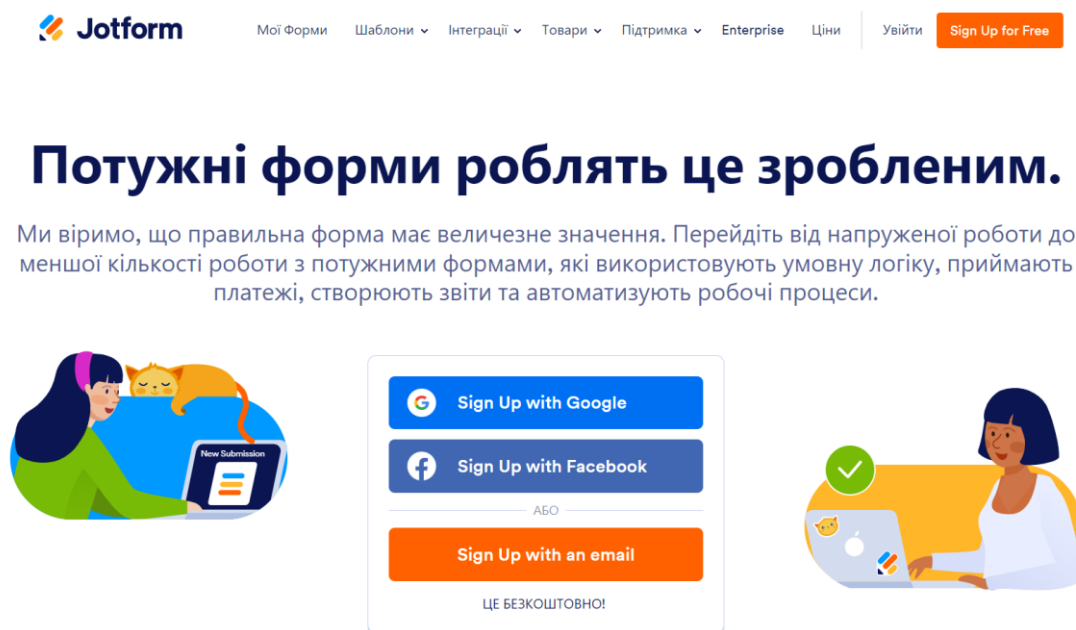
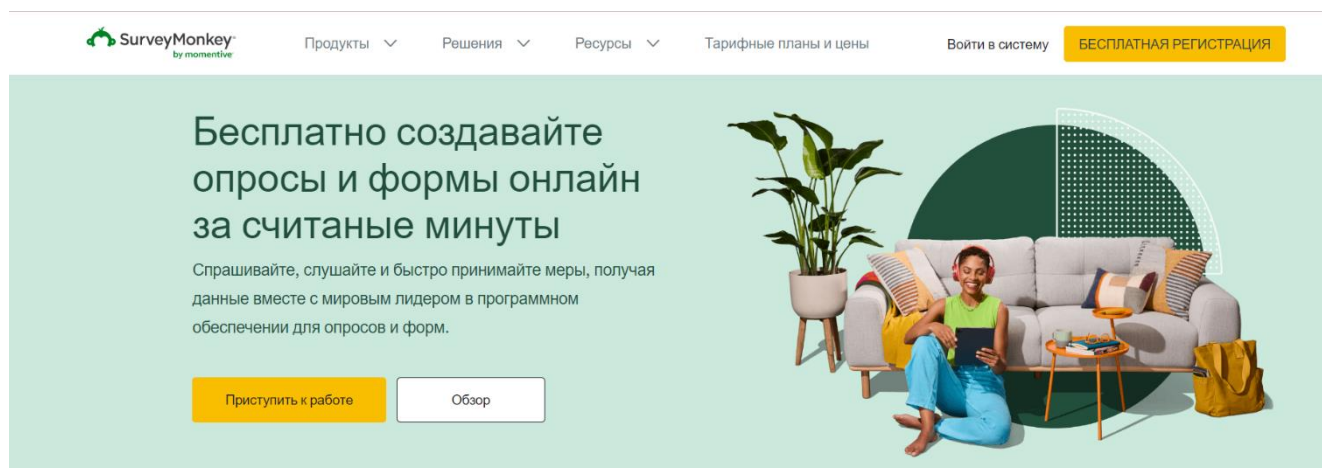


Рисунок 2 – Головна сторінка інструменту «Jotform»

### 1.3 Конструктор «SurveyMonkey»

Конструктор SurveyMonkey, як заявляє виробник, є найпопулярнішим в світі безкоштовним сервісом для створення опитувальників. Тут також присутні платні тарифи: «Advantage (індивідуальний)», «Advantage (груповий)». Цей конструктор має свої продукти, призначені для різних сфер діяльності користувача, наприклад, для компаній, для персоналу, для маркетингу, тощо. Завдяки цьому інструменту можна також проводити аналітику відповідей, створювати опитувальники за шаблонами та взаємодіяти з іншими продуктами від SurveyMonkey.

Загалом цей конструктор є зручним у використанні, але є обмеження у вигляді тарифів, для більш зручного користування. Приклад користувацького інтерфейсу сайту «SurveyMonkey» наведено на рис. 3.



Лидер отрасли с более чем 20-летним опытом и всемирным доверием

**Рисунок 3** – Головна сторінка інструменту «SurveyMonkey»

#### 1.4 Порівняння розглянутих рішень

Розглянуті застосунки було обрано за рейтингом з найбільшою активністю користувачів серед онлайн-сервісів для проведення соціальних досліджень. З цього можна зробити висновок, що дані сайти є найбільш використовуваними в Україні. Через те, архітектура згаданих сайтів є конфіденційною інформацією та захищена авторськими правами, їх можна порівнювати за такими характеристиками, як зручність інтерфейсу, підтримка основних програм тощо (див. табл. 1).

**Таблиця 1** – Порівняння сервісів проведення опитувань

Назва	Google Forms	Jotform	SurveyMonkey
Зручність інтерфейсу	+	+/-	+/-
Функціонал	+	+	+

Простота у використанні	+/-	+/-	+/-
Безкоштовне користування	+/-	+/-	+/-
Каталог збережених поштових адрес	-	-	-

Отже, відповідно до порівняльної характеристики сервісів, яке наведено в табл. 1 можна зробити висновок, що дані конструктори мають як плюси, так і мінуси, але сервіс «Google Forms», за оцінкою даних характеристик, є найкращим серед найпопулярніших інструментів створення опитувальників.

Шляхом проведення аналізу вебсайтів для створення опитувальників, які були розглянуті вище, можна стверджувати, що розробка сайту, який поєднує найкращі характеристики кожного з розглянутих варіантів і виправляє їхні недоліки, є вкрай доцільною.

## РОЗДІЛ 2. ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ

### 2.1 Вибір мови програмування серверної частини

Проектування архітектури є найважливішим етапом розробки будь-якої системи. Добре розроблена гнучка архітектура полегшує підтримку, розширення та вдосконалення системи в майбутньому. Якщо помилки, допущені на етапі проектування, будуть виявлені на пізнішому етапі, їх виправлення буде дорогим.

Тому особливу увагу слід приділити вибору інструментів, мов програмування, технік і бібліотек. При виборі технології розробки важливо забезпечити вимоги до функціональності системи, оскільки широкий набір стандартних функцій бібліотек і технологій може позбавити вас від написання власного коду за рахунок використання готових рішень. Крім того, широкі можливості вибраних бібліотек і технологій полегшують модернізацію системи та її зміни.

Для того, аби використовувати певні технології для створення проєкту потрібно притримуватись таких вимог:

- щоб забезпечити клієнт-серверну архітектуру в застосунку, необхідно вибрати мову програмування, яка забезпечує максимально спрощену взаємодію між сервером і клієнтом шляхом обміну запитами;
- оскільки програми мають справу з великими обсягами даних у базах даних, важливо створити гнучкий і прозорий механізм взаємодії з базами даних. Це зменшить обсяг програмного коду та спростить обслуговування майбутнього застосунку.

Було обрано мову програмування Java, адже вона має здатність масштабуватись на великі проєкти, оскільки вона пропонує потужні інструменти для розробки масштабованих застосунків. Також Java є кросплатформеною мовою програмування, тому код, написаний на Java, можна

використовувати на різних операційних системах. Також важливим аспектом є безпека. Java має вбудовані механізми безпеки, які допомагають уникнути багатьох вразливостей та атак. Вона дозволяє розробнику керувати більшим спектром можливостей для створення сервісів за допомогою різноманітних технологій та бібліотек, які надають інші розробники.

Щоб забезпечити взаємодію між клієнтом програми та сервером, потрібні запити та відповіді у вигляді HTTP. Для цього Java Servlet API використовує клас `HttpServlet`. Цей клас призначений для обробки різних типів запитів HTTP, таких як GET, POST, PUT і DELETE. Він надає можливість обробляти ці запити на сервері та повертати відповідь клієнту у відповідному форматі.

## **2.2 Вибір технології взаємодії з базою даних**

Для ефективної роботи з базою даних і забезпечення гнучкості та розширюваності проєкту, необхідно розробити архітектуру серверної частини таким чином, щоб зменшити складність взаємодії з базою даних і залишити можливість для майбутнього вдосконалення.

Фреймворк `Hibernate` надає зручний спосіб зберігання та отримання об'єктів даних з бази даних. У нашому випадку це `PostgreSQL`. Використання `Hibernate` на проєкті дозволяє спростити роботу з базою даних, забезпечуючи рівень абстракції між об'єктною моделлю застосунку і схемою бази даних

## **2.3 Вибір мови програмування клієнтської частини**

Клієнтська частина веб застосунку представляє собою графічний інтерфейс, який користувач бачить на веб-сторінці. Цей інтерфейс

відображається у веб-браузері, і користувач взаємодіє з вебзастосунком, натискаючи на посилання та кнопки.

Веб-інтерфейс надає користувачеві зручний спосіб взаємодії з функціоналом застосунку. Він може включати різні елементи, такі як кнопки, форми, меню, списки тощо, які дозволяють користувачеві виконувати дії, такі як відправка запитів на сервер, отримання і відображення даних, перехід на інші сторінки або виконання різних операцій у вебзастосунку.

Цей графічний інтерфейс забезпечує зручний та інтуїтивно зрозумілий спосіб взаємодії користувача з вебзастосунком, що робить його більш доступним та привабливим для користувачів.

Було обрано мову TypeScript, яка є мовою програмування, яка розширює JavaScript, додаючи статичну типізацію та деякі додаткові можливості розробки. Angular, з іншого боку, є фреймворком для створення розширених вебзастосунків з використанням TypeScript.

Angular надає фреймворк, який допомагає організувати структуру вебзастосунку. Він надає шаблони, компоненти, модулі, сервіси та інші інструменти для розробки різноманітних функціональностей.

TypeScript і Angular співпрацюють та надають сильну типізацію, структуру, модульність та розширюваність для ефективної розробки вебзастосунків, саме тому, було обрано мову програмування TypeScript та відповідний фреймворк Angular.

## **РОЗДІЛ 3. ОСНОВНІ МЕТОДИ ТА ТЕХНОЛОГІЇ ДЛЯ РОЗРОБКИ ВЕБСАЙТІВ**

### **3.1 Мова програмування Java**

Java – строго типізована об'єктно-орієнтована мова програмування загального призначення [6]. Загалом, це потужна та універсальна мова, яка була розроблена компанією Sun Microsystems (пізніше придбаною Oracle). Вона широко застосовується у розробці програмного забезпечення, починаючи від вебзастосунків до мобільних застосунків та вбудованих систем.

Що відрізняє Java від інших мов програмування, так це її незалежність від платформи. Програми, написані на Java, компілюються в проміжний байт-код, який можна виконати на будь-якій платформі, на якій встановлено віртуальну машину Java (JVM). Це дозволяє розробникам писати код один раз і запускати його в різних операційних системах, зменшуючи витрати на розробку та підтримку.

Однією з головних особливостей Java є її безпека. Він використовує механізми безпеки, такі як керування пам'яттю через збирання сміття, контроль доступу та винятки для запобігання помилкам і забезпечення надійної роботи програми.

Java також має велику кількість розширень (бібліотек і фреймворків), щоб полегшити розробку програм і надати різні функції. Наприклад, фреймворк Spring дозволяє створювати потужні вебзастосунки, тоді як бібліотека JavaFX дозволяє розробляти графічні інтерфейси користувача.

### 3.2 Фреймворк Spring Boot Framework

Spring Boot – це фреймворк розробки програмного забезпечення на основі мови Java [7]. Він побудований на основі Spring Framework [8], що спрощує процес створення стабільних, масштабованих і простих в обслуговуванні програм.

Однією з ключових особливостей Spring Boot є його здатність автоматично налаштовувати багато аспектів програми, що дозволяє розробникам зосередитися на бізнес-логіці, а не витратити час на повсякденні завдання. Він також забезпечує вбудовану підтримку стандартних технологій, таких як бази даних, веб-сервери та кешування, що робить їх простими у використанні та інтеграції.

Spring Boot надає «контейнер сервлетів за замовчуванням», який дозволяє програмам працювати як автономний вбудований веб-сервер без встановлення окремого контейнера сервлетів. Це полегшує розгортання та розробку програми.

Загалом, Spring Boot є потужним фреймворком, який спрощує розробку Java-застосунків шляхом автоматичної конфігурації та інших особливостей.

### 3.3 Фреймворк Hibernate

Hibernate — потужна бібліотека об'єктно-реляційного відображення для Java [9]. Він забезпечує простий спосіб зберігання та отримання об'єктів із бази даних, уникаючи складних запитів SQL.

Hibernate використовує анотації та конфігураційні файли для забезпечення відповідності між класами Java і таблицями бази даних. Він автоматично створює підключення до бази даних і керує ними, а також забезпечує кешування

та оптимізацію запитів. Hibernate дозволяє розробникам зосередитися на логіці застосунків, одночасно спрощуючи керування базами даних.

### 3.4 Фреймворк JUnit

JUnit – це фреймворк для автоматизованого тестування програм на мові Java. Він надає набір класів та анотацій, що допомагають розробникам створювати тести, які перевіряють правильність функціонування їх програмного коду [10].

JUnit використовує концепцію "тестових методів", які містять перевірки очікуваного поведінки коду. Розробники можуть використовувати анотації, такі як `@Test`, для позначення методів як тестових. Крім того, JUnit надає багато інших анотацій для налаштування тестів та роботи з фікстурами.

JUnit забезпечує підтримку запуску тестів із мережевого середовища розробки (IDE), з командного рядка або збирачів проєктів. Він може групувати тести в сюїти, виконувати тести паралельно, відслідковувати стан виконання тестів та генерувати звіти.

JUnit забезпечує механізми перевірки очікуваних результатів, обробки винятків, підготовки тестових умов та управління ресурсами.

### 3.5 Протокол OAuth 2

OAuth2 є протоколом авторизації, який дозволяє користувачам надавати обмежений доступ до своїх ресурсів третім сторонам, без необхідності передавати свої облікові дані [11].

OAuth2 надає різні методи авторизації [12], такі як авторизація за допомогою пароля, авторизація за допомогою коду підтвердження та імпліцитна

авторизація. Протокол також визначає різні типи клієнтських програм, включаючи вебзастосунки, мобільні застосунки та сервер-до-сервер взаємодію.

Він дозволяє користувачам зручно та безпечно давати дозвіл третім сторонам на доступ до своїх даних без необхідності передавати свої облікові дані.

### **3.6 Система керування базами даних PostgreSQL**

PostgreSQL є потужною об'єктно-реляційною системою керування базами даних (ОРСБД), яка надає розширені можливості зберігання, керування та обробки даних. Вона базується на моделі заснованій на таблицях і використовує мову запитів SQL для взаємодії з даними [13].

PostgreSQL підтримує багато функціональних можливостей, включаючи транзакційну надійність, реплікацію, масштабованість та розширення. Він має широку підтримку для різних типів даних, включаючи текстові, числові, географічні, графові та багатовимірні дані.

PostgreSQL підтримує індексацію для оптимізації швидкості пошуку даних і запитів. Він також має механізми для виконання складних запитів, управління правами доступу, забезпечення безпеки даних та резервного копіювання.

Ця база даних є вільним та відкритим програмним забезпеченням, що означає, що її вихідний код доступний для перегляду, змін та розповсюдження.

Завдяки своїм можливостям, надійності та гнучкості, PostgreSQL використовується в різних проєктах та застосунках, включаючи вебзастосунки, аналітичні системи, геопросторові застосунки та багато іншого.

### 3.7 Фреймворк Angular Framework

Angular – це потужний фреймворк для розробки вебзастосунків, побудований на мові програмування TypeScript. Він надає інструменти для створення складних односторінкових застосунків (SPA) з багатофункціональним інтерфейсом [14].

Angular використовує компонентний підхід до розробки, де застосунок розбивається на невеликі, незалежні компоненти, які можуть бути повторно використані. Це сприяє покращенню структури коду, підтримці розширюваності та полегшує тестування.

Фреймворк має потужний інженерний стек, що включає модулі для маршрутизації, обробки форм, керування станом, взаємодії з сервером і багато іншого. Він також надає можливості для розробки анімацій, стилізації і локалізації застосунків.

Angular підтримує двостороннє зв'язування даних, що дозволяє автоматично оновлювати дані в інтерфейсі при зміні моделі даних. Він також надає розширені можливості для обробки подій, валідації даних та маніпуляції DOM-елементами.

### 3.8 Мова програмування TypeScript

TypeScript – це суперсет мови JavaScript, що надає розширену типізацію, об'єктно-орієнтовані можливості та додаткові функціональності для розробки вебзастосунків. Він компілюється до чистого JavaScript, що дозволяє використовувати його в будь-якому середовищі, де підтримується JavaScript [15].

TypeScript надає переваги типізації, що допомагає виявляти помилки в ранній стадії розробки, полегшує рефакторинг коду та підвищує його

читабельність. Він має вбудовану підтримку для статичного аналізу типів, модульної системи і розширених функціональних можливостей, таких як інтерфейси, класи, перелічення, дженеріки та декоратори.

TypeScript інтегрується з редакторами коду та інструментами розробки, що надають автодоповнення, перевірку синтаксису, підказки та інші корисні функції під час написання коду. Він також підтримує нові функції JavaScript, які ще не підтримуються всіма браузерами, за допомогою компіляції до попередніх стандартів JavaScript.

TypeScript підтримує широкий спектр фреймворків та бібліотек JavaScript [16], таких як Angular, React, Vue.js та Node.js. Він широко використовується в розробці вебзастосунків, особливо великих та складних проєктах, де потрібна висока масштабованість, надійність та підтримка.

## РОЗДІЛ 4. ПРИЗНАЧЕННЯ І ЦІЛІ РОЗРОБКИ ЗАСТОСУНКУ

### 4.1 Призначення застосунку

Метою цієї кваліфікаційної роботи є реалізація сервісу зі створення опитувальників, для того, аби користувачі могли проводити соціальні дослідження, проводити контроль опитувань, використовувати результати для покращення бізнесу, тощо. Розробка має задовольняти наступні вимоги:

- надавати можливість зручного користування;
- надавати можливість реєстрації та автентифікації;
- надавати можливість створення опитувальників;
- надавати можливість налаштовувати опитувальники;
- надавати можливість поширення опитувальників;
- надавати можливість брати участь в опитувальниках;
- надавати можливість перегляду статистики;
- надавати можливість керування доступом;
- надавати можливість безпеки даних.

Проаналізувавши наявні рішення та їх функціональні можливості, можна зробити висновок, що актуально розробити систему зі зручним інтерфейсом, багатим функціоналом та простим користуванням. Це дасть змогу легкого та зрозумілого доступу до програми, оскільки такі системи популярні та корисні.

Отже, очікуваний результат має мати вигляд вебсайту з можливістю легко та швидко створювати опитувальники, де можна створювати та налаштовувати опитувальники, поширювати їх та переглядати статистику.

## 4.2 Цілі розробки застосунку

Вебсайт «SURVEY<X>» розробляється з метою:

- забезпечення створення опитувальників;
- надання зручного інтерфейсу;
- забезпечення збору даних для різних сфер;
- можливості вивчення ринку в різних сферах.

Цей сервіс має значний потенціал в різних сферах застосування. Важливість збору та аналізу даних стає все більш вагомим в сучасному світі, і цей сайт допомагає здійснити цей процес ефективно та зручно.

Він надає можливість створювати та проводити опитування онлайн, що робить його зручним та доступним, у ньому є можливість налаштування опитувальників під потреби користувачів – можна вибирати різні типи відповідей. Також він забезпечує аналітичні інструменти для обробки та аналізу зібраних даних.

Повністю весь функціонал вебсайту можна переглянути на use-case діаграмі (див. додаток А).

## 4.3 Вимоги до застосунку

### 4.3.1 Вимоги до застосунку в цілому

Застосунок «SURVEY<X>» повинен мати інтуїтивно зрозумілий інтерфейс з можливістю створення опитувальників. Під час створення опитувальників користувач має змогу вибирати тип відповіді та кількість питань. Після створення опитувальника користувач може редагувати його, поширити людям за згенерованим посиланням або обрати пошти, на які прийде опитування.

Користувач спочатку має пройти авторизацію та автентифікацію перед тим, як розпочати роботу зі створення опитувальників. Користувач може зареєструватись звичайним шляхом, та отримати посилання із підтвердженням особистості або обрати шлях через Google акаунт.

Дослідник може створити опитувальники, додавати питання, обирати тип відповідей, редагувати опитувальники та видаляти їх. На основі відповідей на цей опитувальник, можна також подивитись на статистику.

#### 4.3.2 Вимоги до функцій, які виконуються програмою

Перелік основних функцій користувача наведено в табл. 2.

**Таблиця 2** – Перелік функцій та задач користувача

Функція	Задача
Реєстрація	Введення короткої інформації про себе(логін та пароль).
	Повторне введення паролю
	Вихід із акаунту
Реєстрація Google	Обирання пошти користувачем
Створення опитувальника	Надання назви опитування
	Обирання типу опитувальника
	Обирання фотографії для опитувальника
	Додавання питання
	Обирання чи питання є обов'язковим
	Видалення питання
	Надання назви питання

	Обирання типу відповіді
	Збереження опитування
Керування опитувальником	Перегляд контактів, на які було надіслано опитування
	Редагування опитувальника
	Перейменування опитувальника
	Додавання контактів
	Перегляд статистики
	Видалення опитувальника
Фільтри та пагінація	Фільтрування опитувальників за датою
	Фільтрування опитувальників за назвою
	Розбиття контенту на невеликі блоки
Каталог збережених адрес електронних пошт	Додавання імені та пошти
	Сортування контактів
	Редагування контактів
	Видалення контактів
	Завантаження файлу з контактами
	Вивантаження контактів із сайту в файл

### 4.3.3 Технічні вимоги до застосунку

Для того, аби створити такий застосунок потрібно визначитись, що він має виконувати та яку мати функціональність:

- У розробці має використовуватись SQL Database

- Створення класів на основі сутностей предметної області і структурування їх по пакетам з відповідними назвами, які відображають їх функціональність.
- Використання БД для зберігання інформації про предметну область та доступу до даних за допомогою JDBC API з використанням пулу з'єднань.
- Підтримка кирилиці в застосунку, включаючи зберігання інформації в БД.
- Реалізація Unit-тестів для перевірки функціональності застосунку.
- Логування подій в системі за допомогою Log4j.
- Реалізація системи авторизації і аутентифікації.
- Використання сесій і фільтрів для розробки бізнес-логіки.
- Реалізація пагінації та транзакцій.
- Розділення клієнтської частини від бек-енда та комунікація між ними за допомогою HTTP протоколу.
- Виведення англійських повідомлень про помилки на сайті при некоректних діях користувача, таких як невірні дані або незаповнені обов'язкові поля.
- Підтримка веб-браузера Google Chrome.

## РОЗДІЛ 5. РЕАЛІЗАЦІЯ ВЕБСАЙТУ

### 5.1 Опис організації інформаційної бази

#### 5.1.1 Логічна структура бази даних

Щоб ефективно використовувати базу даних у цій програмі, потрібно мати систему керування базами даних (СКБД) [17], яка підтримує реляційну модель даних. Реляційна модель передбачає існування зв'язків між об'єктами, які зберігаються в різних таблицях. Для зберігання та обробки даних у таких СКБД потрібні таблиці з належною структурою. У таблиці кожен стовпець може містити різні типи даних. Кожен запис складається з набору атрибутів (стовпців) з унікальним ключем, що зберігається в одній таблиці. Усі ці дані взаємопов'язані відповідно до принципів реляційної моделі. Також важливо, щоб обрана СКБД допускала безкоштовне розповсюдження.

Враховуючи вищезазначену ситуацію, вирішено використовувати базу даних PostgreSQL. Для забезпечення зв'язку між об'єктами системи необхідно використовувати зовнішні ключі. Об'єктно-реляційна система управління базами даних PostgreSQL надає зручні можливості для роботи з даними, дозволяючи розробникам зосередитися на програмуванні та уникнути проблем, пов'язаних із зберіганням даних.

Наведено опис кожної таблиці бази даних у таблиці 3.

**Таблиця 3** – Таблиці бази даних «SURVEY<X>»

Номер	Таблиця	Опис
1	users	Таблиця для збереження короткої інформації про користувачів.
2	surveys	Таблиця для збереження інформації опитувальника

3	survey_questions	Таблиця для збереження інформації про питання в опитувальниках
4	survey_answers	Таблиця для збереження інформації про відповіді в опитувальниках
5	survey_contacts	Таблиця для збереження інформації про контакти опитувальника
6	respondent	Таблиця для збереження інформації про учасників опитувальників
7	anonym	Таблиця для збереження інформації про анонімних учасників
8	verificationtoken	Таблиця для збереження інформації про токен верифікації

### 5.1.2 Опис таблиць бази даних програми

Таблиці є основним компонентом бази даних, оскільки вони забезпечують зберігання всіх даних. Перед початком роботи з базою даних необхідно спроектувати структуру кожної таблиці. Ця структура повинна відповідати потребам вихідних форм, запитів та звітів, які мають бути згенеровані при взаємодії з базою даних.

У таблиці “users” присутні наступні атрибути: id (bigint), email (varchar), password (varchar), role (varchar), creationdate (timestamp) та active (boolean). Детальніша інформація про ці атрибути може бути знайдена у таблиці 5. Ці атрибути використовуються для збереження інформації про користувачів системи. Наприклад, вони містять дані про пошту та пароль користувача, його роль в системі, дату реєстрації та статус підтвердження облікового запису.

Таблиця 5 – Опис таблиці users

Атрибут	Тип	Опис
id	bigint	Ідентифікатор (primary key)
email	varchar	Пошта
password	varchar	Пароль
role	varchar	Ідентифікатор клієнта (foreign key)
creationdate	timestamp	Дата реєстрації
active	boolean	Підтверджений користувач

У таблиці “surveys” присутні наступні атрибути: id (bigint), active (boolean), creationdate (timestamp), imageurl (varchar), status (integer), title (varchar), type (varchar) та user\_id (bigint). Детальніша інформація про ці атрибути може бути знайдена у таблиці 6. Ці атрибути використовуються для збереження інформації про опитувальник. Наприклад, вони містять дані про статус та підтвердження опитувальника, дату його створення, шлях до фотографії, назву та тип опитувальника, а також зв'язок з користувачем, який створив опитувальник.

Таблиця 6 – Опис таблиці surveys

Атрибут	Тип	Опис
id	bigint	Ідентифікатор (primary key)
active	boolean	Підтверджений користувач
creationdate	timestamp	Дата створення
imageurl	varchar	Шлях до фотографії опитувальника
status	integer	Статус опитувальника
title	varchar	Назва опитувальника
type	varchar	Тип опитувальника

user_id	bigint	Ідентифікатор користувача(foreign key)
---------	--------	--

У таблиці “surveys\_questions” присутні наступні атрибути: id (bigint), choiceanswers (varchar), index (integer), question (varchar), required (boolean), type (varchar) та survey\_id (bigint). Детальніша інформація про ці атрибути може бути знайдена у таблиці 7. Ці атрибути використовуються для збереження інформації про питання в опитувальнику, зокрема текст питання, варіанти відповідей, порядок питань, обов'язковість відповіді та тип питання. Також використовується зв'язок з таблицею опитувальника за допомогою атрибуту survey\_id.

**Таблиця 7** – Опис таблиці surveys\_questions

Атрибут	Тип	Опис
id	bigint	Ідентифікатор (primary key)
choiceanswers	varchar	Текстові варіанти можливих відповідей
index	integer	Порядок питань
question	varchar	Питання опитувальника
required	boolean	Чи є питання обов'язковим
type	varchar	Тип питань
survey_id	bigint	Ідентифікатор опитувальника(foreign key)

У таблиці “surveys\_answers” присутні наступні атрибути: id (bigint), value (varchar), required (boolean), question\_id (bigint) та respondent\_id (bigint). Детальніша інформація про ці атрибути може бути знайдена у таблиці 8. Ці атрибути дозволяють зберігати інформацію про відповіді на питання опитування, зазначати, чи є питання обов'язковими та встановлювати зв'язки з відповідними питаннями та респондентами.

**Таблиця 8** – Опис таблиці surveys\_answers

Атрибут	Тип	Опис
id	bigint	Ідентифікатор (primary key)
value	varchar	Відповідь на питання
required	boolean	Чи є питання обов'язковим
question_id	bigint	Ідентифікатор питання(foreign key)
respondent_id	bigint	Ідентифікатор респондента(foreign key)

У таблиці присутні наступні атрибути: id (bigint), anonym\_id (bigint) та contact\_id (bigint). Детальніша інформація про ці атрибути може бути знайдена у таблиці 9. Ці атрибути дозволяють встановлювати зв'язки з анонімними та контактними користувачами, які можуть бути пов'язані з даним записом.

**Таблиця 9** – Опис таблиці respondent

Атрибут	Тип	Опис
id	bigint	Ідентифікатор (primary key)
anonym_id	bigint	Ідентифікатор аноніма(foreign key)
contact_id	bigint	Ідентифікатор контакта(foreign key)

У таблиці містяться такі атрибути: id (bigint), can\_pass (boolean), contact\_id (bigint) та survey\_id (bigint). Таблиця 10 відображає деталізовану інформацію про ці атрибути. Ці атрибути використовуються для збереження інформації про можливість проходження опитування користувачем з певним контактом та опитуванням.

**Таблиця 10** – Опис таблиці surveys\_contacts

Атрибут	Тип	Опис
---------	-----	------

id	bigint	Ідентифікатор (primary key)
can_pass	boolean	Допуск до опитування
contact_id	bigint	Ідентифікатор питання(foreign key)
survey_id	bigint	Ідентифікатор респондента(foreign key)

У таблиці містяться такі атрибути: id (bigint), email (varchar), name (varchar) та user\_id (bigint). Таблиця 11 відображає деталізовану інформацію про ці атрибути. Ці атрибути використовуються для збереження інформації про контакти, зокрема їх пошту, ім'я та пов'язаного респондента.

**Таблиця 11** – Опис таблиці contacts

Атрибут	Тип	Опис
id	bigint	Ідентифікатор (primary key)
email	varchar	Пошта контакту
name	varchar	Ім'я контакту
user_id	bigint	Ідентифікатор респондента(foreign key)

У таблиці присутні такі атрибути: id (bigint) та description (varchar). Таблиця 12 містить детальніше опис кожного атрибуту. Ці атрибути використовуються для збереження інформації про анонімів, зокрема їх опису.

**Таблиця 12** – Опис таблиці анонум

Атрибут	Тип	Опис
id	bigint	Ідентифікатор (primary key)
description	varchar	Опис аноніма

У таблиці містяться наступні атрибути: id (bigint), expiredate (timestamp), token (varchar) та user\_id (bigint). Опис кожного атрибуту

наведено у таблиці 13. Ці атрибути використовуються для збереження даних, пов'язаних з верифікацією нового користувача. Наприклад, `expiredate` визначає дату, до якої токен верифікації є дійсним. Токен (`token`) представляє собою унікальний рядок, який використовується для перевірки та підтвердження нового користувача. `user_id` вказує на респондента (користувача), пов'язаного з цим токеном.

**Таблиця 13** – Опис таблиці `verificationtoken`

Атрибут	Тип	Опис
<code>id</code>	<code>bigint</code>	Ідентифікатор (primary key)
<code>expiredate</code>	<code>timestamp</code>	Дата експірації
<code>token</code>	<code>varchar</code>	Токен верифікації нового користувача
<code>user_id</code>	<code>bigint</code>	Ідентифікатор респондента (foreign key)

## 5.2 Розробка вебзастосунку

Розробка вебзастосунку «SURVEY<X>» пройшла кілька етапів, кожен з яких описано в наступних розділах. Під час аналізу предметної області було встановлено, що створення опитувальників є актуальною темою та допомагає людям у багатьох сферах життя.

Однією з ключових вимог до бази даних є наявність інформації про опитувальники, їхні питання, відповіді та користувачів. Реляційна база даних PostgreSQL була обрана через її високу продуктивність, надійність і масштабованість. Тому структурна схема бази даних може змінюватися з часом.

Таким чином, інформаційна система «SURVEY<X>» повинна містити повну інформацію про користувачів, опитувальники та його компоненти, які

необхідні для збору статистичних даних. Застосунок також має забезпечувати зручну навігацію та функції пошуку.

### 5.3 Постановка завдання

Після первинного аналізу предметної області виявилось, що для успішної реалізації проєкту необхідно виконати наступні завдання:

- авторизація та автентифікація користувачів;
- створення комфортного інтерфейсу для дослідників у написанні опитувальників;
- редагування, перейменовування опитувальників;
- можливість додавати контакти для опитувальника закритого типу;
- можливість ділитись опитувальником за посиланням для відкритого типу;
- збереження до каталогу часто використовуваних контактів;
- можливість редагування контактів у каталозі;
- можливість імпортувати та експортувати файли зі списком контактів;
- можливість переглядати статистику опитувальника;
- можливість фільтрування опитувальників;
- можливість пагінації опитувальників;
- можливість фільтрувати контакти;

Після узагальнення вимог можна зробити висновок, що вебзастосунок матиме інтуїтивно зрозумілий і зручний інтерфейс, доступ до якого можна отримати з будь-якого комп'ютера, який має доступ до Інтернету.

## 5.4 Структура програми

Вебзастосунок структурно ділиться на дві частини: клієнт і сервер. Кожен із цих розділів виконує власну функцію та відповідає за різні аспекти програми з різними рівнями доступу.

Клієнтська частина відповідає за передню частину програми, тобто за відображення і взаємодію з інтерфейсом користувача. Він виконується на стороні клієнта (наприклад, у веббраузері) і має обмежений доступ до ресурсів сервера. Клієнтська частина відповідає за відображення даних, обробку дій користувача та взаємодію з сервером для отримання та надсилання даних.

Серверна частина відповідає за обробку логіки програми, доступ до бази даних і обробку запитів від клієнтської частини. Він працює на сервері з повним доступом до ресурсів інфраструктури. Серверна частина відповідає за обробку бізнес-логіки, зберігання й обробки даних і надання відповідей клієнтській частині.

### 5.3.1 Серверна частина

Серверна частина вебзастосунку розділена на три рівні відповідно до принципу трирівневої архітектури Three Layer Architecture: рівень презентації, рівень бізнес-логіки та рівень доступу до даних [18]. Архітектура Розділяє функціональні можливості програми на логічні групи, зменшуючи залежність між компонентами та полегшуючи розробка та супровід програмного забезпечення. У додатку В наведений рисунок структури проєкту.

Контролери, які взаємодіють з клієнтами та обробляють запити користувачів, знаходяться на рівні презентації. Вони приймають вхідні дані, перевіряють їх і викликають відповідні компоненти бізнес-логіки для обробки запиту.

Логіка бізнес-процесу знаходиться на рівні бізнес-логіки. Нижче наведено класи, які реалізують бізнес-правила та логіку програми. Вони отримують дані від контролерів, обробляють їх, взаємодіють з базами даних та іншими службами для забезпечення необхідної функціональності.

Рівень доступу до даних відповідає за зберігання та отримання даних із баз даних або інших джерел. Він включає в себе класи і методи для взаємодії з даними, забезпечення роботи з ORM (Object Relational Mapping) і виконання запитів до баз даних.

Ця структура проекту, поділена на рівні відповідно до трирівневої архітектури, допомагає покращити модульність, масштабованість і зручність обслуговування коду. Кожен рівень виконує власну функцію та взаємодіє з іншими рівнями через чітко визначені інтерфейси, що полегшує розробку та тестування програмного забезпечення.

Програмний продукт було реалізовано з використанням мови програмування Java та фреймворку Spring Boot. Використання Spring Boot дозволяє побудувати застосунок з чіткою розширеною архітектурою, де кожна складова може бути легко змінена або модернізована незалежно від інших.

Для управління проектом використовувався інструмент залежностей та збирання Gradle. Він забезпечує зручну організацію структури директорій проекту та автоматичне вирішення залежностей між компонентами. За допомогою Gradle можна ефективно керувати залежностями, компілювати та збирати програмний код, а також виконувати інші рутинні завдання розробки.

Папка "src" поділена на "main" та "test", а папка "main" додатково також має поділ на "java", та "resources". Модуль "src" містить усю логіку застосунку.

У модулі "target" розміщено скомпільовану версію проекту, яка включає в себе файли з розширенням ".class" та веб-архів з розширенням ".jar". Цей

скомпільований проєкт передається на керування веб-серверу Apache Tomcat. Веб-сервер Tomcat відповідає за розгортання та запуск програмного продукту.

Папка "photo" – виступає у ролі storage для фото, які завантажують користувачі при створенні опитувальників.

Папка "util" – класи, які відповідають за роботу з email, парсингом CSV файлів та інші допоміжні класи.

Папка "config" – місце, де налаштовуються клієнти та створюються beans.

Основний файл проєкту, використовуваний Gradle – це pom.xml. В цьому файлі міститься основна інформація про проєкт, включаючи підключені бібліотеки, плагіни та налаштування компіляції модулів.

Згідно обраної архітектури, були реалізовані моделі, які виступають посередниками між програмою та базою даних. Класи моделей розміщені у папці "entity".

Файл .gitignore є плагіном, який розробник додає до проєкту. Він використовується як допоміжний механізм для підтримки системи контролю версій. Цей файл дозволяє розробнику вказати, які файли або директорії в проєкті необхідно ігнорувати під час розробки. Це може бути використано, наприклад, для виключення файлів, стан яких не змінюється протягом розробки або зміни яких не впливають на роботу програми.

Як зазначено раніше, в модулі "src" міститься модуль "main", який в свою чергу розділяється на два підмодулі:

- папка "java" містить серверну частину програми, яка реалізована на мові програмування Java. Згідно з шаблоном проєктування, програма включає контролери, сервіси та репозиторії, які знаходяться у відповідних папках. Контролер приймає HTTP запити, які надходять з клієнтської частини. У сервісах знаходиться бізнес-логіка і в репозиторіях робляться запити до бази даних.

- папка "resources" містить основні конфігураційні файли проєкту, такі як налаштування бази даних, конфігурація системи логуювання та інші важливі параметри системи.

### **5.3.2 Клієнтська частина**

Основна логіка програми на сервері відокремлена від клієнтської частини, що дозволяє розробнику уникнути зайвих обчислень або перевірок, і зосередитись на зовнішньому вигляді та інтерфейсі для користувача. Для досягнення інформативного, лаконічного та інтерактивного вигляду кінцевого продукту, були використані графічні бібліотеки мови TypeScript та фреймворк Angular. Крім того, були використані наперед задані каскадні CSS стилі з фреймворку Bootstrap.

### **5.3.3 Інтеграція зі стороннім сервісом**

Інтеграція з Google OAuth 2.0 надає можливість автентифікації та авторизації користувачів через їх облікові записи Google [19]. Цей протокол авторизації забезпечує безпеку та захист інформації, а також спрощує процес входу для користувачів.

У даній роботі було проведено інтеграцію з Google OAuth 2, саме тому користувачі, при автентифікації та авторизації, можуть скористатись своїми обліковими записами Google, що є зручно та безпечно.

## **5.4 Тестування сайту**

У процесі розробки вебсайту було проведено тестування як мануальним шляхом, так і за допомогою інструменту Postman для забезпечення якості та

надійності програмного продукту. Також було проведене модульне тестування програми.

Окрім цього було також проведене тестування із сумісністю програми з різними браузерями, наприклад Google Chrome, Microsoft Edge або Opera.

#### **5.4.1 Мануальне тестування**

Мануальне тестування включало в себе перевірку різних аспектів функціональності та коректності вебсайту. Ручне тестування проводилося шляхом взаємодії з сайтом, де проводилась перевірка різних функціональних можливостей сайту, такі як реєстрація, вхід у систему, навігація по сторінкам, заповнення форм, фільтрування даних тощо. Цей підхід дозволив виявити потенційні проблеми та покращити зручність вебсайту для користувачів.

#### **5.4.2 Тестування API**

Крім того, було використано інструмент Postman для автоматизації тестування API вебсайту. Postman надає можливість створювати запити до сервера і перевіряти відповіді, що дозволяє перевірити правильність роботи API, обробку даних та взаємодію з базою даних. Тестові сценарії були створені для різних функціональних частин API, що дозволило виявити та виправити можливі помилки і недоліки.

### 5.4.3 Модульне тестування

Модульний тест є невеликим фрагментом коду [20], спрямованим на перевірку конкретної, добре визначеної функціональної частини коду. Він зазвичай тестує певний метод в певному контексті.

Оскільки модульне тестування є необхідним, під час розробки системи було використано переваги вбудованої бібліотеки JUnit для виконання таких тестів. JUnit – це інтегроване середовище для тестування клієнтських Java-застосунків, яке надає зручні засоби для модульного тестування.

## РОЗДІЛ 6. ІНСТРУКЦІЯ КОРИСТУВАЧА

Коли користувач вперше потрапляє на голову сторінку вебсайту (див. рис. 4), він має змогу зареєструватись або увійти в систему, як попередньо зареєстрований користувач сервісу.



“Everything is surveyed. Few things are surveyed well.” - John Doe

**Рисунок 4** – Головна сторінка «SURVEY<X>»

Під час процесу авторизації та реєстрації користувача (зображеного на рисунку 5 та на рисунку 6 відповідно), всі дані перевіряються на їх правильність. Ця перевірка відбувається як на стороні користувача, так і на стороні сервера. Зокрема, проводиться перевірка на валідність введених даних, а також перевірка на унікальність електронної адреси при реєстрації. Якщо виявляються неправильні або недопустимі дані, користувачу надсилається повідомлення про помилку. Користувач може зареєструватись або авторизуватись за допомогою облікового запису Google.

The screenshot shows the 'Registration' page of the Survey application. At the top left, the word 'Survey' is displayed. At the top right, there are links for 'Sign up' and 'Log in'. The main content area is titled 'Registration' and contains three input fields: 'Email', 'Password', and 'Confirm Password'. Below these fields are two buttons: 'Register' (in blue) and 'Cancel'. A horizontal line separates this section from the next, which is headed 'Or' and features a red 'Google' login button. At the bottom of the page, a black footer contains the text '© Survey X - All rights reserved'.

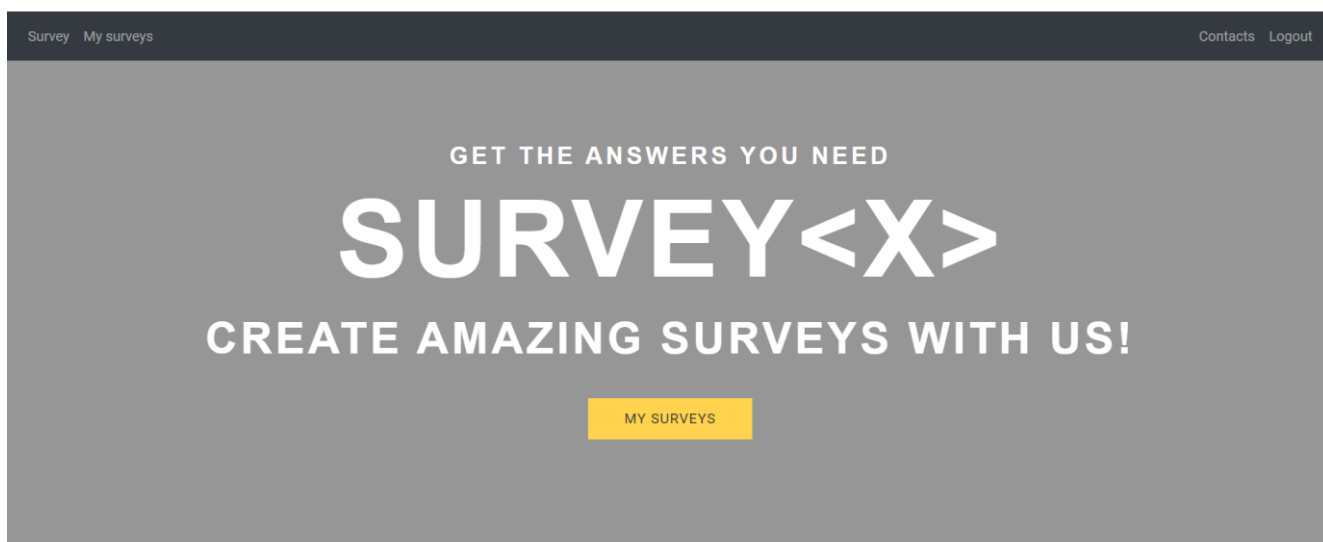
**Рисунок 5 –** Сторінка авторизації

Сторінка реєстрації виглядає наступним чином (див. рис. 6). Під час реєстрації користувач може обрати два шляхи реєстрації: пошта та пароль або обліковий запис Google.

The screenshot shows the 'Login' page of the Survey application. At the top left, the word 'Survey' is displayed. At the top right, there are links for 'Sign up' and 'Log in'. The main content area is titled 'Login' and contains two input fields: 'Email' and 'Password'. Below these fields are two buttons: 'Login' (in blue) and 'Cancel'. A horizontal line separates this section from the next, which is headed 'Or' and features a red 'Google' login button. At the bottom of the page, a black footer contains the text '© Survey X - All rights reserved'.

**Рисунок 6 –** Сторінка реєстрації

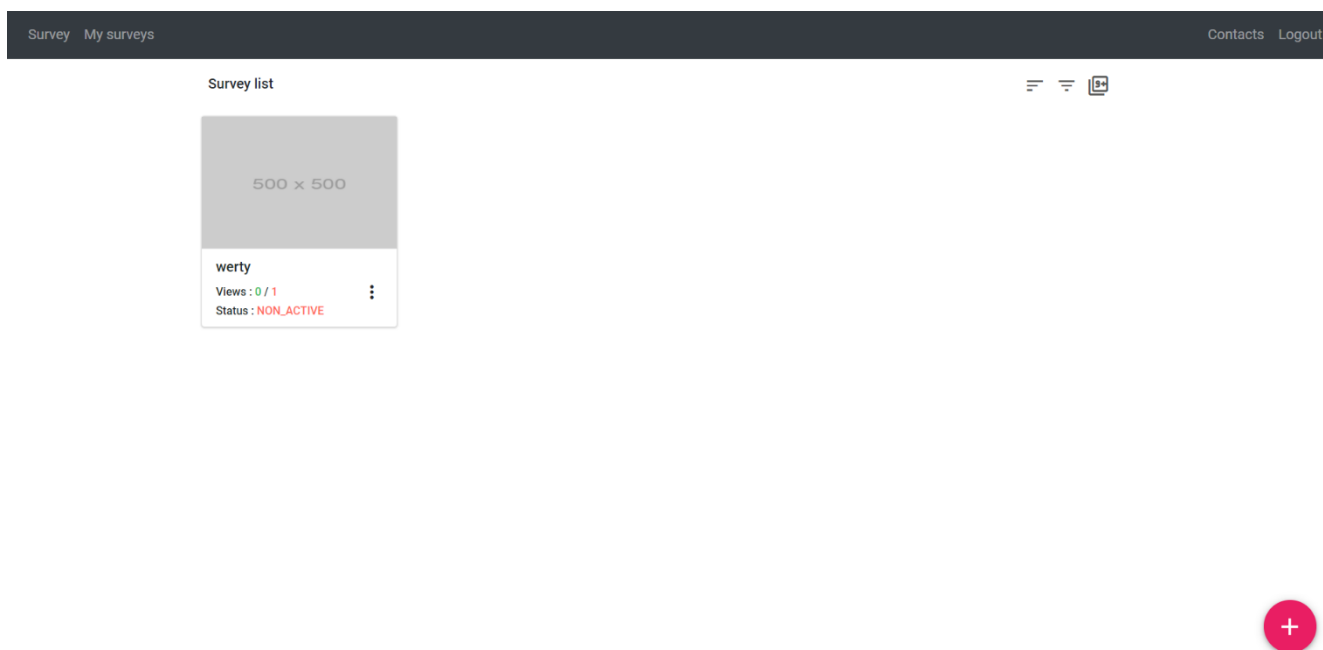
Після вдалої реєстрації користувач бачить головну сторінку, на якій може обрати функцію створити опитувальник (див. рис. 7).



"Everything is surveyed. Few things are surveyed well." - John Doe

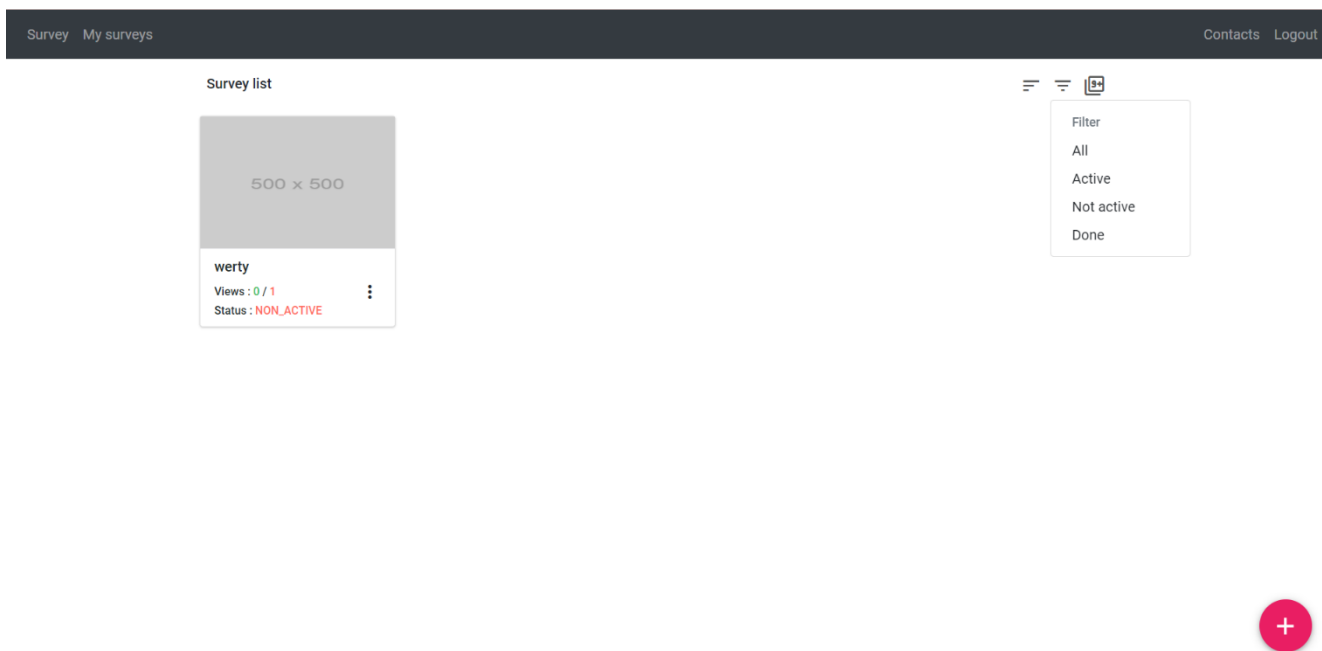
### Рисунок 7 – Головна сторінка після реєстрації

На даній сторінці (див. рис. 8) користувач може створити опитувальник та переглянути створені раніше опитувальники.



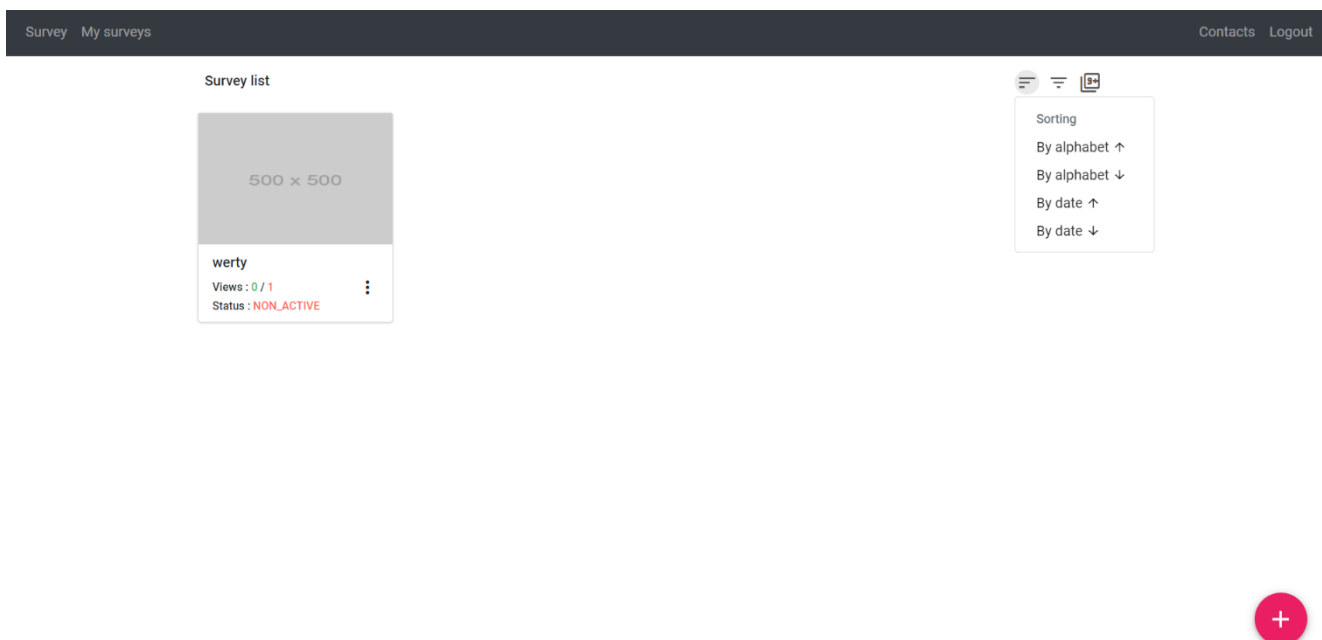
### Рисунок 8 – Сторінка для створення та перегляду опитувальників

На цій сторінці користувач може відфільтрувати створені опитувальники за їхнім станом (див. рис. 9).



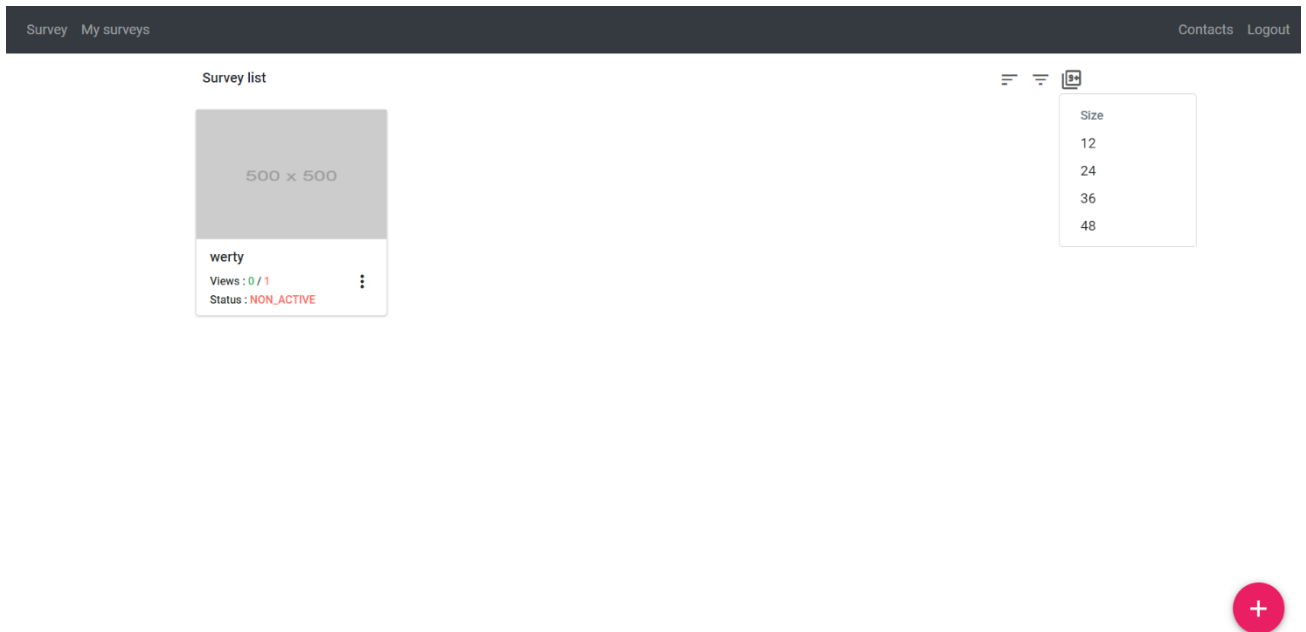
**Рисунок 9 – Фільтрування опитувальників**

На цій же ж сторінці можна відсортувати опитувальники за алфавітом або датою у зростаючому та спадаючому порядку (див. рис. 10).



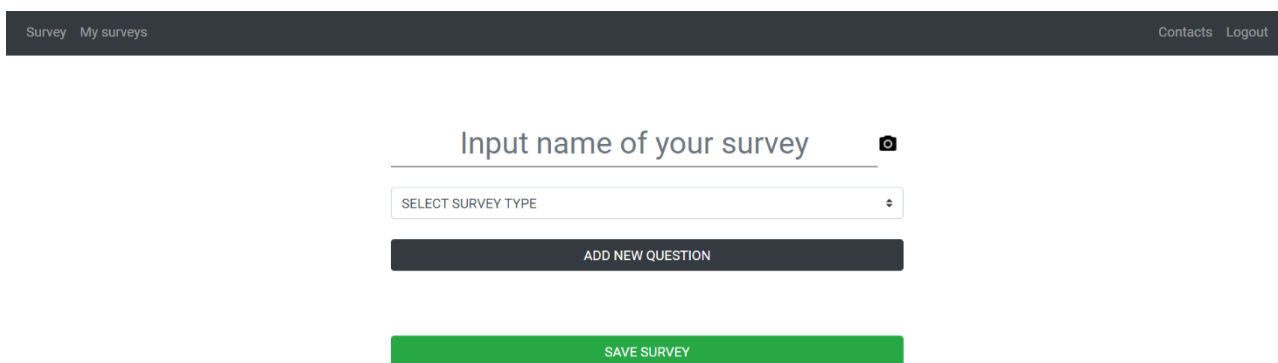
**Рисунок 10 – Сортування опитувальників**

Також можна провести пагінацію опитувальників, тобто зробити їхнє розбиття в розмірі від 12 до 48 (див. рис. 11).



**Рисунок 11** – Пагінація опитувальників

Створення опитувальників було розроблено так, щоб користувачу було інтуїтивно зрозуміло, які кроки для цього потрібно зробити. Початок створення опитувальника виглядає таким чином (див. рис. 12). Користувач дає ім'я опитувальнику.



Survey My surveys Contacts Logout

Input name of your survey

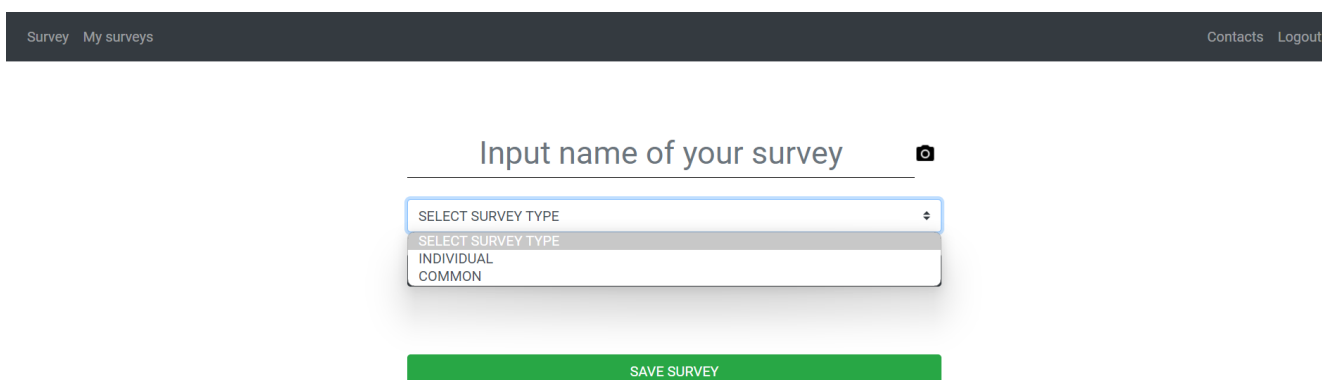
SELECT SURVEY TYPE

ADD NEW QUESTION

SAVE SURVEY

**Рисунок 12** – Сторінка створення опитувальника

На даній сторінці (див. рис. 13) користувач обирає тип опитувальника, а також за бажанням може завантажити фотографію для нього. Користувач може обрати Individual або Common тип опитувальника. Перший тип відповідає за те, щоб він був доступний лише для обраних творцем опитувальника людей.



Survey My surveys Contacts Logout

Input name of your survey

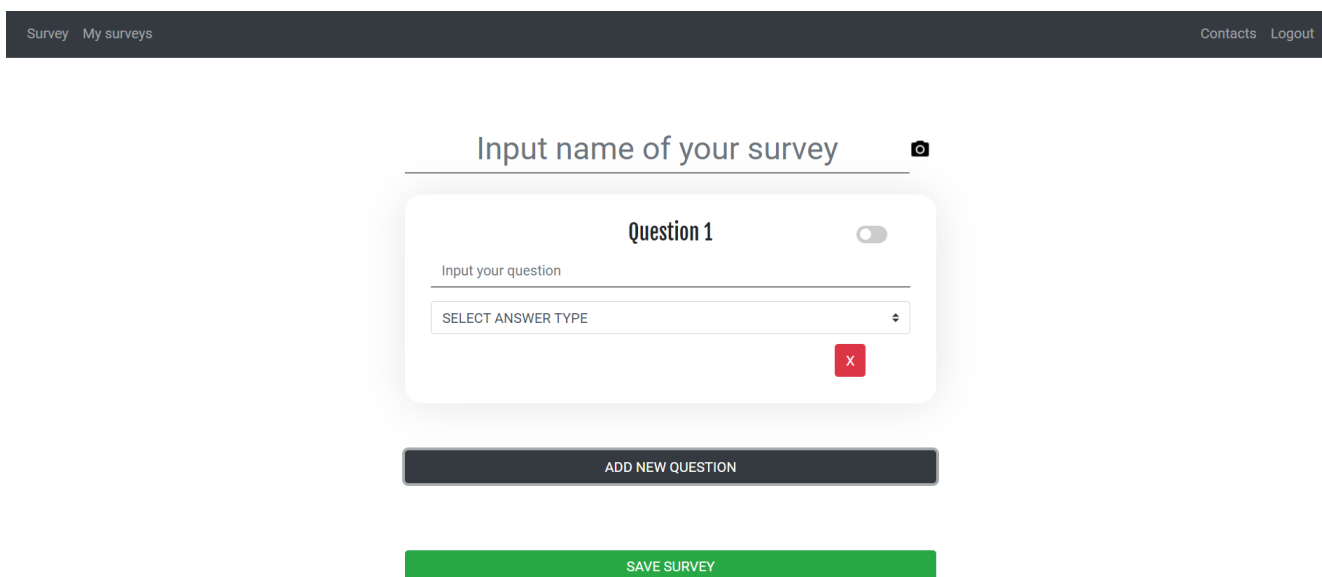
SELECT SURVEY TYPE

- SELECT SURVEY TYPE
- INDIVIDUAL
- COMMON

SAVE SURVEY

**Рисунок 13** – Обирання типу опитувальника

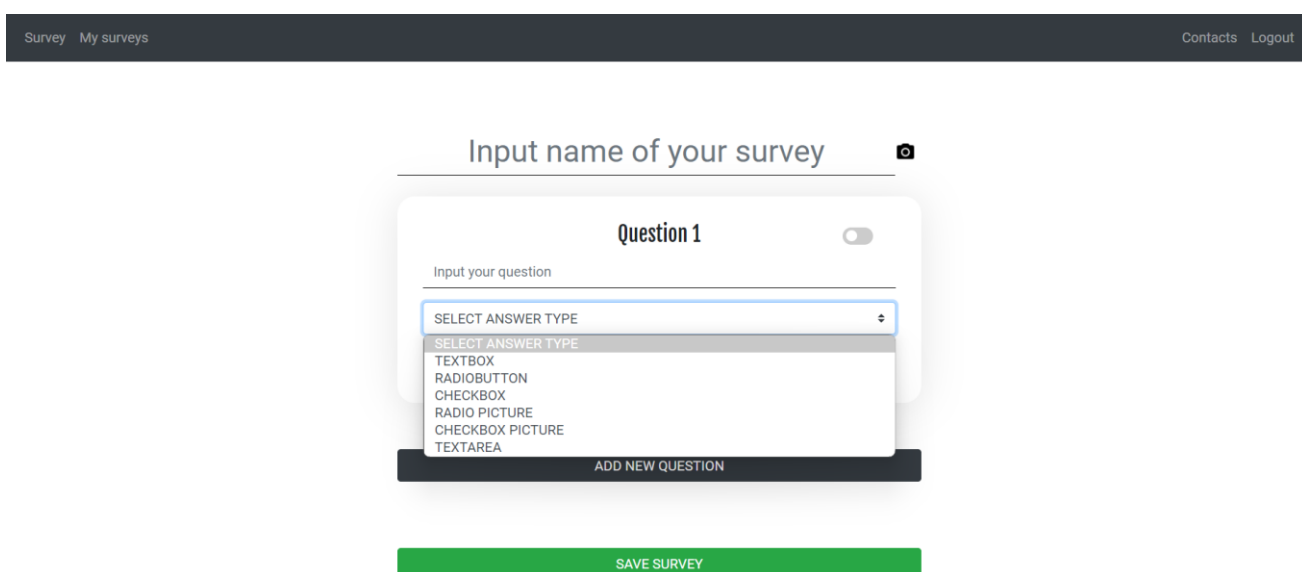
На етапі створення тіла опитувальника можна обрати чи питання має бути обов'язковим (див. рис. 14). Користувач має змогу видалити питання.



The screenshot shows a web interface for creating a survey. At the top, there is a dark navigation bar with "Survey My surveys" on the left and "Contacts Logout" on the right. Below the navigation bar, the main heading is "Input name of your survey" with a camera icon to its right. The central part of the interface is a white card titled "Question 1" with a toggle switch on the right. Inside the card, there is a text input field labeled "Input your question" and a dropdown menu labeled "SELECT ANSWER TYPE" with a red 'X' button to its right. Below the card, there are two buttons: a dark grey "ADD NEW QUESTION" button and a green "SAVE SURVEY" button.

**Рисунок 14** – Написання тіла опитувальника

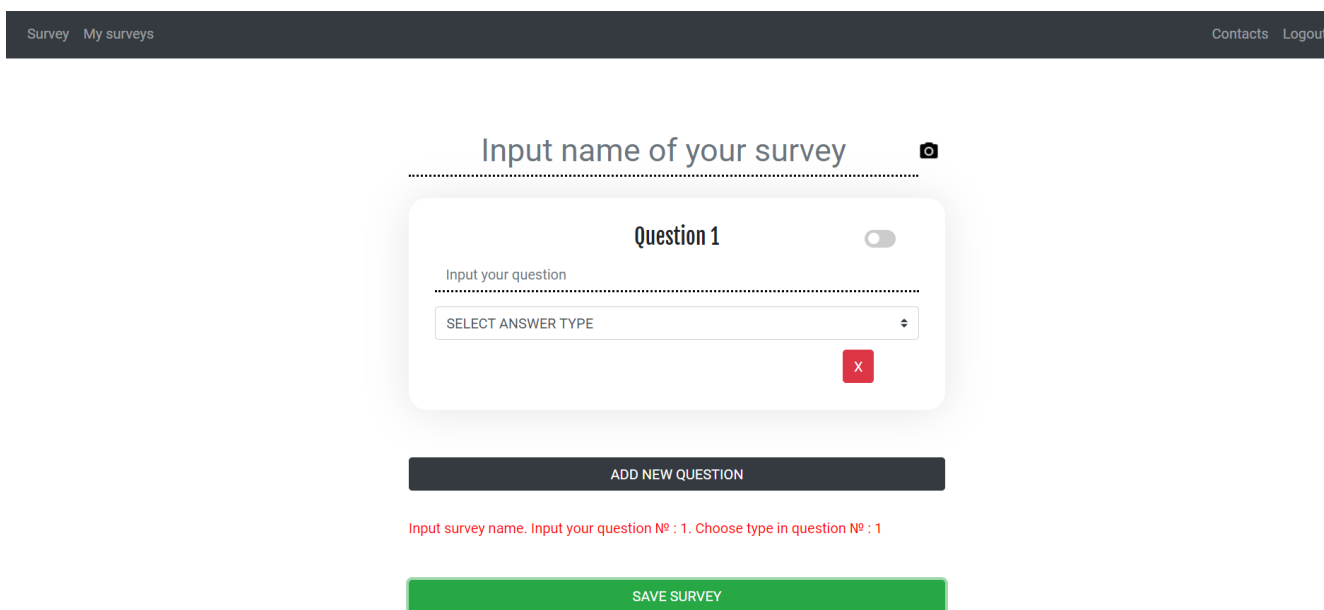
На етапі створення завдань, користувач має змогу обрати тип відповідей (див. рис. 15).



The screenshot shows the same survey creation interface as in Figure 14, but with the "SELECT ANSWER TYPE" dropdown menu open. The dropdown menu lists several options: "SELECT ANSWER TYPE", "TEXTBOX", "RADIOBUTTON", "CHECKBOX", "RADIO PICTURE", "CHECKBOX PICTURE", and "TEXTAREA". The "ADD NEW QUESTION" button is now dark grey, and the "SAVE SURVEY" button is green.

**Рисунок 15** – Обирання типу відповідей

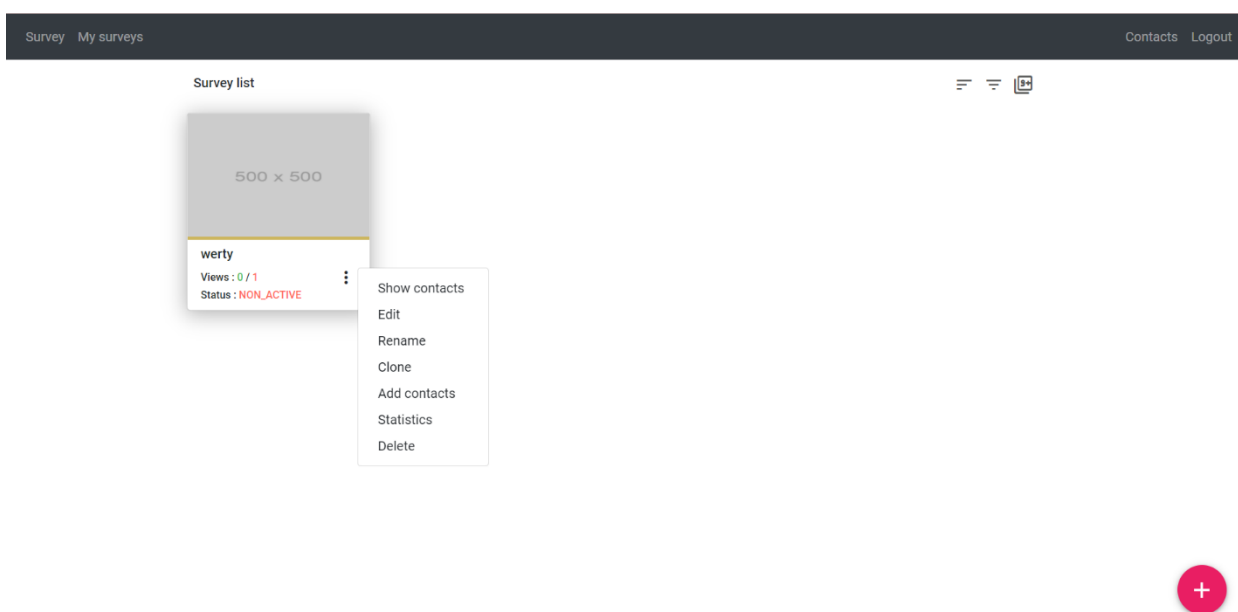
Якщо користувач хоче зберегти опитувальник, не заповнивши всі поля, йому буде висвітлено повідомлення про помилку (див. рис. 16).



The screenshot shows a web interface for creating a survey. At the top, there is a navigation bar with "Survey My surveys" on the left and "Contacts Logout" on the right. The main heading is "Input name of your survey". Below this is a form for "Question 1" with a toggle switch. The form contains an input field for the question text and a dropdown menu for "SELECT ANSWER TYPE". A red "X" icon is visible in the bottom right corner of the question form. Below the form is a dark button labeled "ADD NEW QUESTION". A red error message is displayed: "Input survey name. Input your question № : 1. Choose type in question № : 1". At the bottom is a green button labeled "SAVE SURVEY".

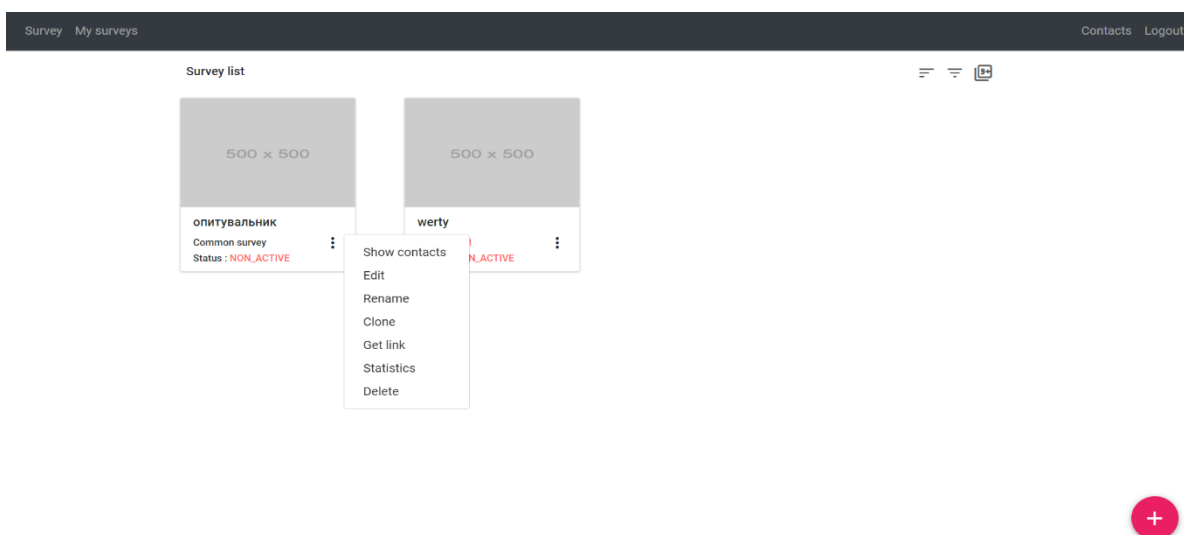
**Рисунок 16** – Повідомлення про помилку

Після створення опитувальника користувач може проводити з ним деякі операції (див. рис. 17).



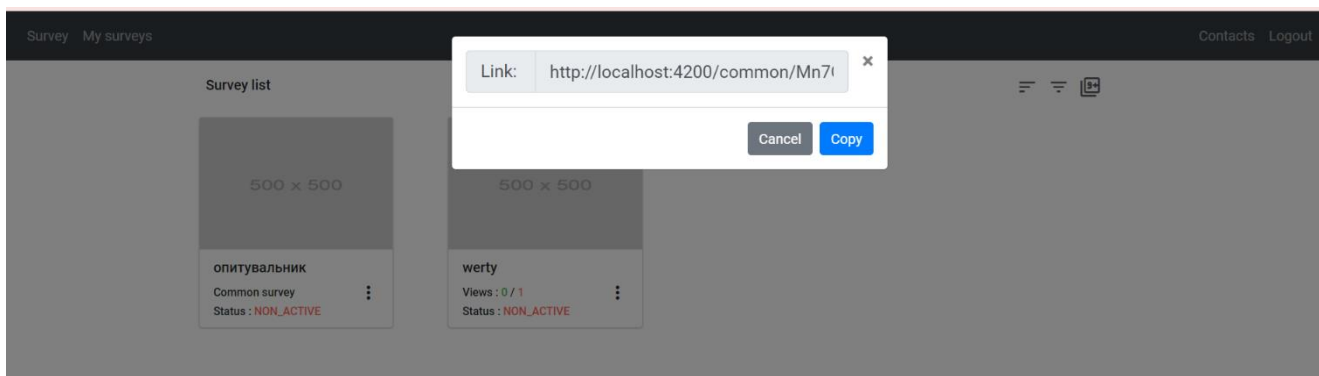
**Рисунок 17** – Функціонал після створення опитувальника. Режим Individual

Користувач може редагувати, перейменувати, видалити опитувальник. У режимі Individual користувач може додати та переглянути контакти, які мають доступ до створеного опитувальника (див. рис. 17). А у Common режимі користувач може переглянути контакти, яким було надіслано посилання опитувальника (див. рис. 18).



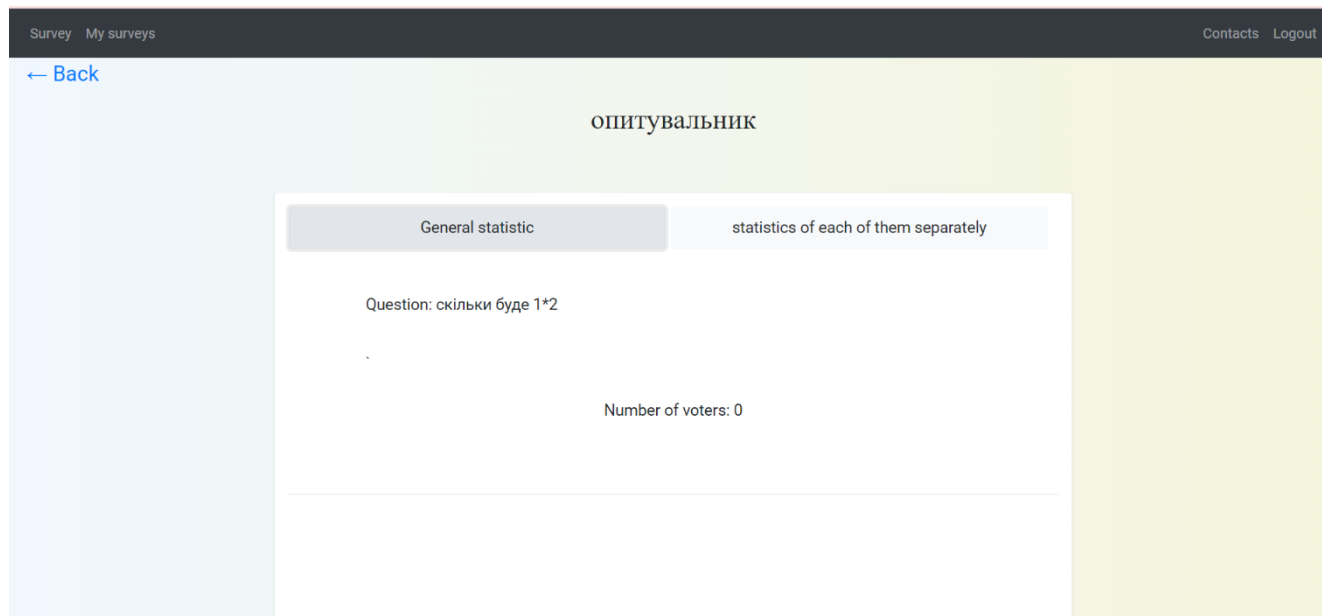
**Рисунок 18** – Функціонал після створення опитувальника. Режим Common

На даному рисунку (див. рис. 19) видно як користувач у режимі Common може скопіювати посилання опитувальника для подальшого його поширення.



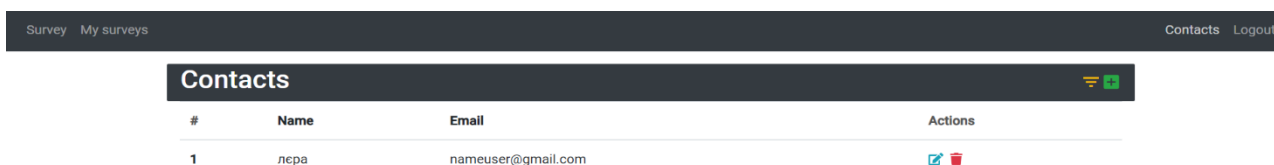
**Рисунок 19** – Отримання посилання для поширення опитувальника

Також користувач може переглянути статистику опитувальника (див. рис. 20).



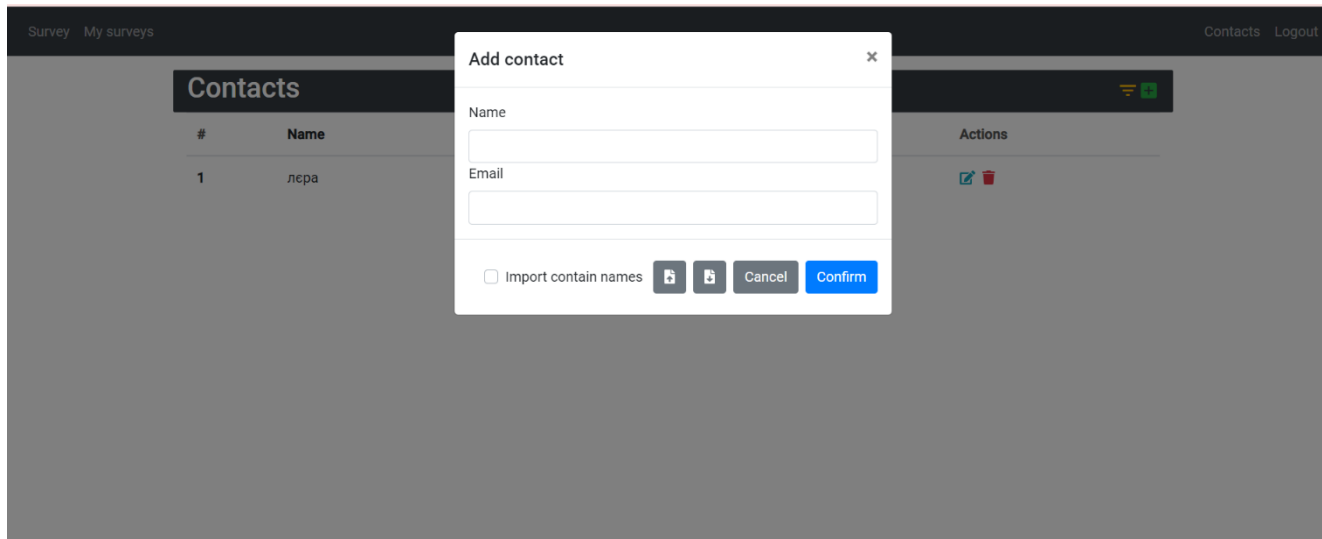
**Рисунок 20** – Сторінка статистики

Даний сайт дає можливість користувачу зберігати часто використовувані поштові адреси (див. рис. 21)



**Рисунок 21** – Каталог збережених контактів

У такому каталозі можна додавати контактів, видаляти їх, редагувати та фільтрувати. Також у користувача є можливість завантажувати списки контактів у вигляді CSV файлів до себе у каталог та вивантажувати відповідно у вигляді CSV файлу(див. рис. 22).



**Рисунок 22** – Додавання нових контактів

## ВИСНОВКИ

У даній кваліфікаційній було розглянуто основні принципи та інструменти для розробки вебзастосунків загалом. Було проведено аналіз використання різних технологій для проєктування та реалізації вебсайту.

У роботі було досліджено наявні системи для створення соціальних досліджень та проведено їхнє порівняння. У ході розробки даного сервісу було розроблено технічне завдання до продукту.

Протягом виконання кваліфікаційної роботи бакалавра було успішно виконані заплановані завдання, зокрема здійснено аналіз застосунків, які призначені для створення опитувальників, досліджено застосування різноманітних технологій для проєктування та реалізації вебзастосунків, розроблено технічне завдання, дизайн та інтерфейс «SURVEY<X>», було розроблено вебзастосунок для створення опитувальників у вигляді клієнт-серверного сервісу. Для серверної частини було обрано мову програмування Java. Для клієнтської частини – TypeScript.

У подальшому рекомендується акцентувати увагу на розширенні та вдосконаленні функціональності даного вебсервісу. Крім того, можна розглянути можливість впровадження цього рішення в існуючий сервіс з метою розширення його можливостей.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Lesley A. Designing and Doing Survey Research / Andres Lesley., 2012. – 208 с. – (1).
2. JetBrains [Електронний ресурс] – Режим доступу до ресурсу: <https://www.jetbrains.com/>.
3. Abramson J. Survey Methods in Community Medicine: Epidemiological Research, Programme Evaluation, Clinical Trials / J. Abramson, Z. Abramson., 2008. – 432 с.
4. Google Forms [Електронний ресурс] – Режим доступу до ресурсу: <https://www.google.com/intl/uk/forms/about/>.
5. Jotform [Електронний ресурс] – Режим доступу до ресурсу: <https://www.jotform.com/>.
6. Eckel B. Thinking in Java / Bruce Eckel. – Prentice Hall PTR, 2006. – (4).
7. Professional Java Development with the Spring Framework / [R. Johnson, J. Höller, A. Arendsen та ін.], 2005. – 672 с. – (1).
8. Spring Boot Framework [Електронний ресурс] – Режим доступу до ресурсу: <https://spring.io/>.
9. Hibernate [Електронний ресурс] – Режим доступу до ресурсу: <https://hibernate.org/>.
10. JUnit [Електронний ресурс] – Режим доступу до ресурсу: <https://foxminded.ua/unit-testuvannya-v-java/>.
11. Whitson G. Understanding OAuth: What Happens When You Log Into a Site with Google, Twitter, or Facebook / Gordon Whitson., 2012.
12. OAuth [Електронний ресурс] – Режим доступу до ресурсу: <https://www.loginradius.com/blog/identity/what-is-oauth/>.
13. PostgreSQL [Електронний ресурс] – Режим доступу до ресурсу: <https://aws.amazon.com/ru/rds/postgresql/what-is-postgresql/>.
14. Angular Framework [Електронний ресурс] – Режим доступу до ресурсу: <https://brander.ua/technologies/angularjs>.

15. TypeScript [Електронний ресурс] – Режим доступу до ресурсу:  
<https://codeguida.com/post/475>.
16. Eich B. An Introduction to JavaScript / Brendan Eich., 2007.
17. Reese G. Database Programming with Jdbc & Java / George Reese., 2000. – 348 с. – (2).
18. Three layer architecture [Електронний ресурс] – Режим доступу до ресурсу:  
<https://www.ibm.com/topics/three-tier-architecture>.
19. Google OAuth [Електронний ресурс] – Режим доступу до ресурсу:  
<https://developers.google.com/identity/protocols/oauth2?hl=ua>.
20. Модульне тестування [Електронний ресурс] – Режим доступу до ресурсу:  
<https://geosoft.no/unitesting.html>

## ДОДАТКИ

## Додаток А. Use-Case діаграма для застосунку

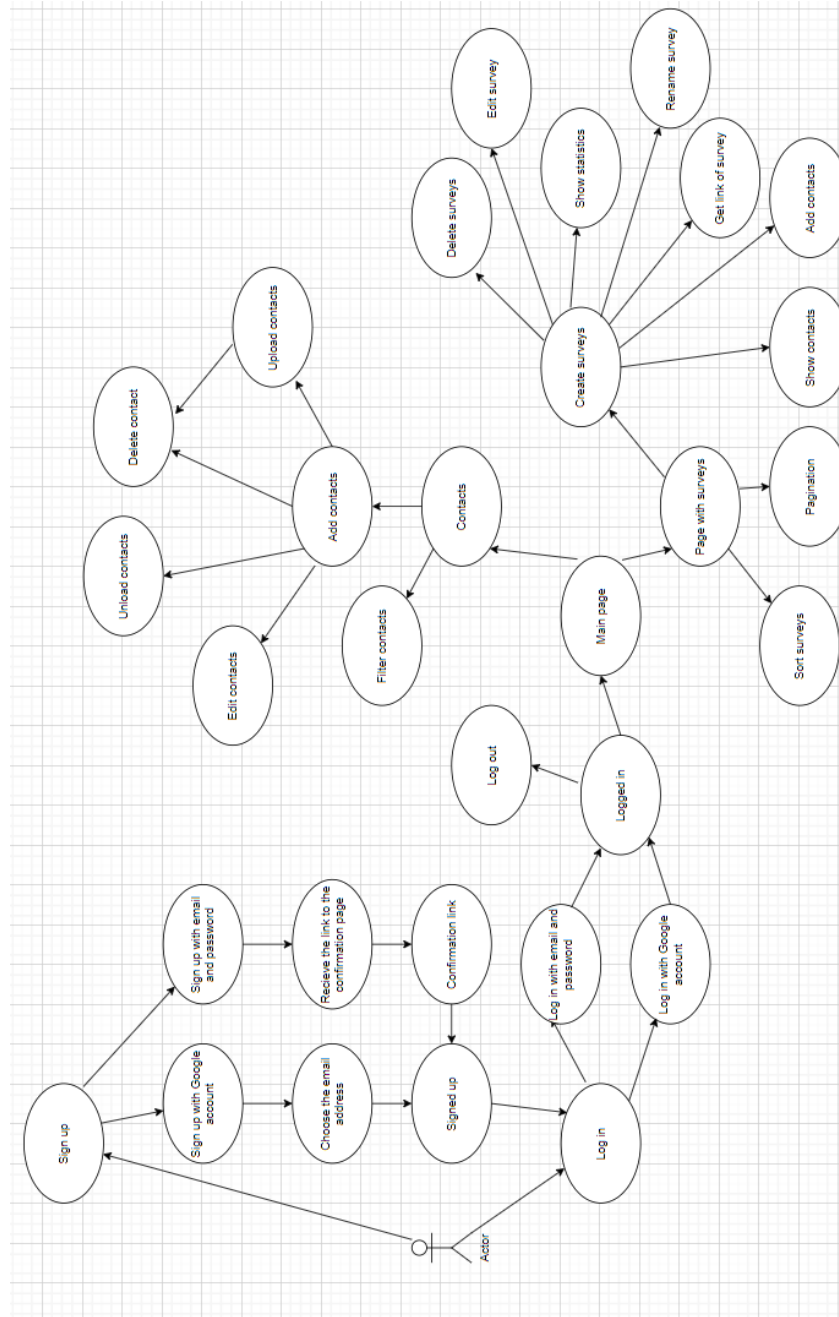


Рисунок А.1 – Use-Case діаграма сайту «SURVEY&lt;X&gt;»

## Додаток Б. Діаграма бази даних застосунку

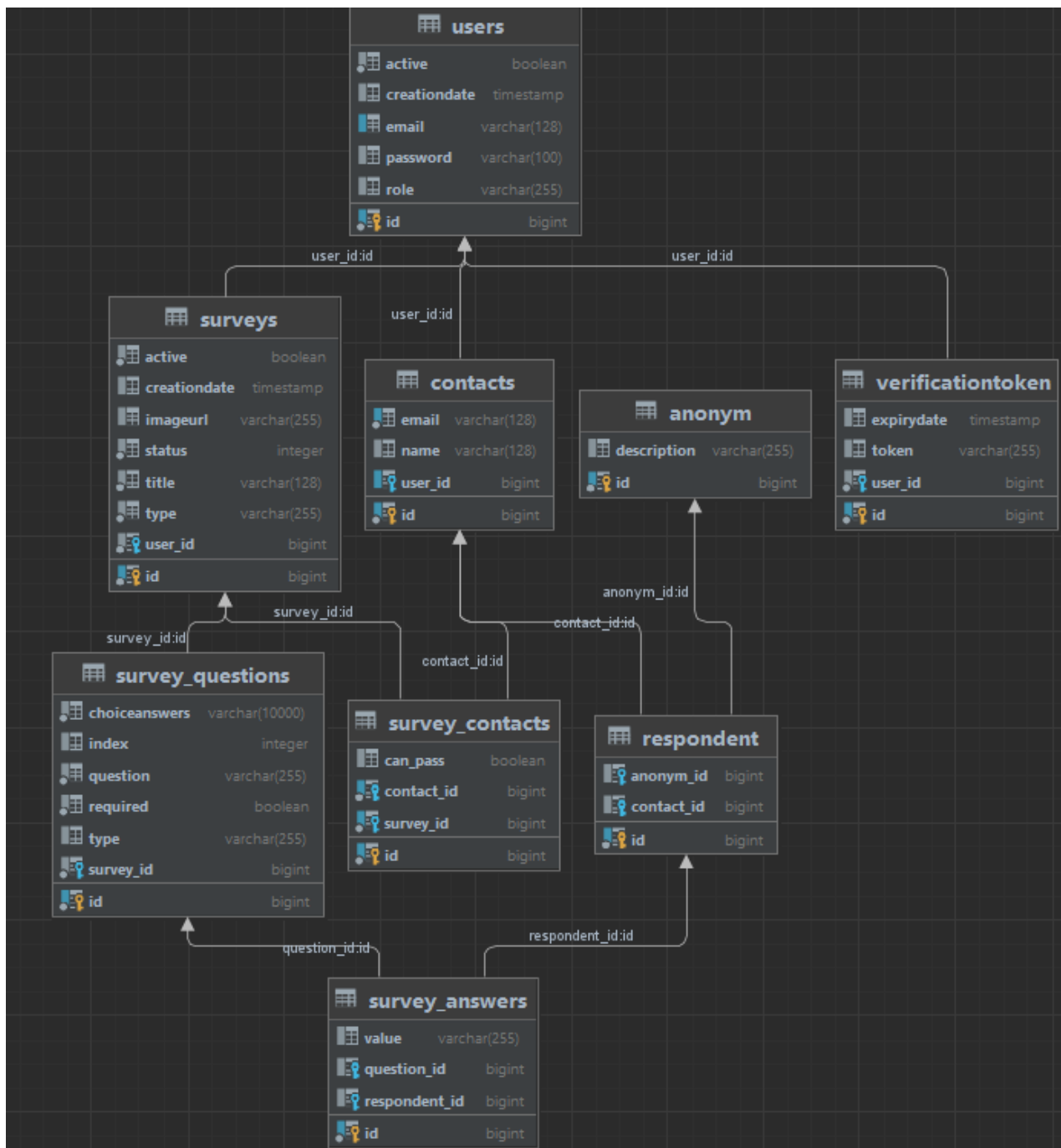


Рисунок Б.1 – Діаграма бази даних «SURVEY<X>»

## Додаток В. Структура програми застосунку

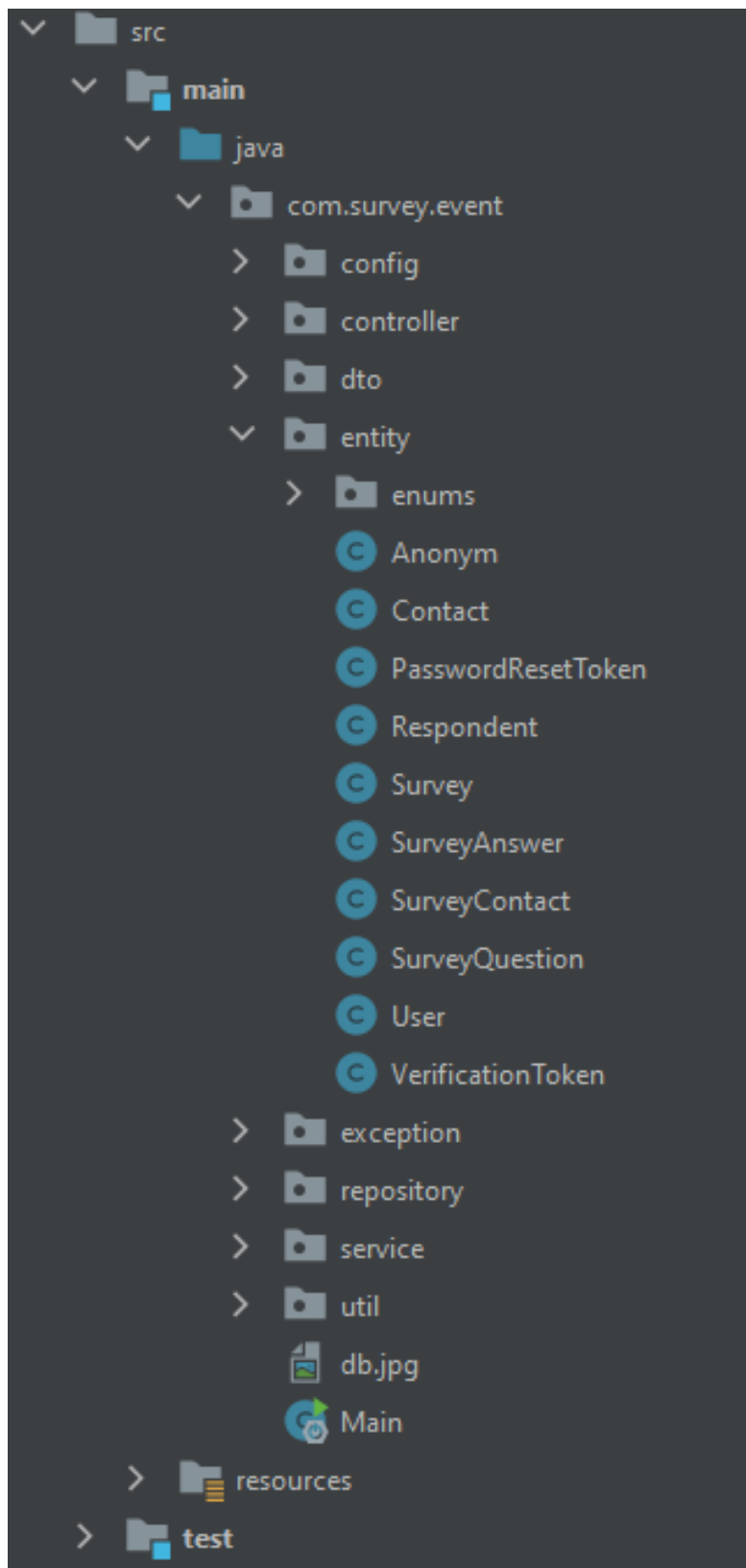


Рисунок В.1 – Структура програми