

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ
УКРАЇНИ

Факультет інформаційних технологій

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ

Завідувач кафедри
Інформаційних систем і технологій
(назва кафедри)

Швиденко М.З., к.е.н., проф.

(підпис)

(ПІБ)

“16” травня 2025 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА
РОБОТА

на тему

Інформаційна система управління діяльністю консалтингової компанії

Спеціальність 126 – «Інформаційні системи та технології»

Гарант освітньої програми

Канд.екон.наук, доцент

(науковий ступінь та вчене звання)

Мокрієв Максим Володимирович

(підпис)

(ПІБ)

Керівник бакалаврської кваліфікаційної роботи

Доктор філософії, доцент

(науковий ступінь та вчене звання)

Саяпіна Таїсія Петрівна

(підпис)

(ПІБ)

Виконав

(підпис)

Торішний Микола Юрійович

(ПІБ студента)

КИЇВ – 2025

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ
УКРАЇНИ
Факультет інформаційних технологій**

ЗАТВЕРДЖУЮ
Завідувач кафедри
Інформаційних систем та технологій
(назва кафедри)

Швиденко М.З к.е.н, проф
(науковий ступінь, вчене звання) (підпис) (ПІБ)
“ ___ ” _____ 2025 р.

ЗАВДАННЯ
на виконання бакалаврської кваліфікаційної роботи студенту
Торішньому Миколі Юрійовичу

Спеціальність 126 – «Інформаційні системи та технології»

1. Тема бакалаврської кваліфікаційної роботи «Інформаційна система управління діяльністю консалтингової компанії» затверджена наказом ректора НУБіП України від 16.12.2024 № 13-С
2. Термін подання завершеної роботи на кафедру 2025.06.01
(рік, місяць, число)
3. Вихідні дані: надання професійного аналізу та консультації з бізнес-питань та бізнес-діяльності у вигляді графіків, таблиць та діаграм.
4. Перелік питань, що розглядаються:
 - Системний аналіз предметної області
 - Інформаційне забезпечення
 - Розробка системи та її елементів
 - Реалізація, тестування та подальше використання системи

5. Календарний план

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	22 лютого 2025	
2	Аналіз предметної області	березень 2025	
3	Моделювання предметної області	березень 2025	
4	Проектування програмної системи	квітень 2025	
5	Розгортання та експлуатація програмного забезпечення	квітень- травень 2025	
6	Економічне дослідження розробки та експлуатації програмної системи	квітень- травень 2025	
7	Оформлення записки	травень 2025	
8	Перевірка на плагіат	2 червня 2025	
9	Проходження нормо контролю	29 травня – 4 червня 2025	
10	Проходження передзахисту	5-7 червня 2025	
11	Захист роботи	14-16 червня 2025	

Керівник кваліфікаційної роботи

доцент доктор філософії _____ Саяпіна Т.П _____
 (науковий ступінь та вчене звання) (підпис) (ПІБ)

Завдання прийняв до виконання _____ Торішній М.Ю. _____
 (підпис) (ПІБ)

Дата отримання завдання “ ___ ” _____ 2025 р

РЕФЕРАТ

Тема бакалаврської кваліфікаційної роботи: “Інформаційна система управління діяльністю консалтингової компанії”.

Автор роботи: Торішній Микола Юрійович.

Керівник роботи: Саяпіна Таїсія Петрівна.

Пояснювальна записка: 85 с., 30 рис., 21 табл., 3 дод., 29 джерел.

Графічна частина: 10 презентаційних слайдів.

АНАЛІЗ ТА ТАРГЕТУВАННЯ, МЕТА ТА ЗАВДАННЯ, ПРЕДМЕТ ДОСЛІДЖЕННЯ, ІНФОРМАЦІЙНА СИСТЕМА, УПРАВЛІННЯ ТА АВТОМАТИЗАЦІЯ, РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ, РЕЗУЛЬТАТИ

Метою роботи є розробка інформаційної системи надання фінансових та податкових послуг онлайн орієнтованих на цільову аудиторію в США, яка забезпечує оперативний прийом заявок, обробку даних шляхом автоматичного направлення повідомлень до Telegram-чату. Аналіз вимог та потреб заявника відбувається безпосередньо при спілкуванні в Telegram-чаті з консультантами, після чого пропонується індивідуальне пакетне рішення фінансового супроводу або разова консультація. У роботі створено інформаційну систему KeyStone Consulting, яка реалізує швидку та зручну взаємодію між користувачем і фінансовим консультантом з акцентом на мінімізацію витрат, простоту використання та оперативну обробку звернень. Розроблено архітектуру системи, включаючи логічну та фізичну моделі бази даних у MySQL. У технічному аспекті система реалізована з оптимальним технологічним стеком, до якого входять HTML5, CSS3, JavaScript, PHP, а також інтеграція з Telegram API для оперативної комунікації. Створення KeyStone Consulting системи виконано з використанням сучасних інструментів розробки, таких як Visual Studio Code, OpenServer, а також макетування у Figma. Розроблена система відповідає всім функціональним вимогам і має потенціал для масштабування. Пропозиції розвитку включають подальше розширення функціоналу: додавання календаря записів, онлайн-оплати тощо.

ABSTRACT

Title of the bachelor's qualification work: "Information System of managing Consulting Company activity".

Author: Torishniy Mykola.

Supervisor: Saiapina Taisiia

Explanatory note: 85 pages, 18 figures, 21 tables, 1 appendices, 29 sources.

Graphical part: 10 presentation slides.

ANALYSIS AND TARGETING, INFORMATION SYSTEM, MANAGEMENT, AUTOMATION, PROJECT MANAGEMENT, ANALYTICS

The purpose of this work is to develop an information system of managing consulting company activity which related to finance and taxation service for US market targeting group. The information system is to accept applications in real-time flow, process data and provide feedback by automatically redirecting messages to the Telegram chat. The customer's needs and requirements analysis is made in Telegram chat by consultants. Thus the individual complex option of financial support or one-time consultation is offered. The thesis presents KeyStone Consulting information system that implements prompt and convenient interaction between the customer and the financial consultant with cost saving effect, easy to understand user-interface and fast processing of applications. The system's architecture was developed, using logical and physical database models in MySQL. Considering the technical issue, this information system was implemented using optimal technical tools such as HTML5, CSS3, JavaScript, PHP, as well as Telegram API integration for prompt communication. The web-site was developed using coding management tools such as Visual Studio Code, OpenServer. The system marketing was made with the help of Figma. KeyStone Consulting information system meets all functional requirements and has potential for scalability. The development recommendations include further development of functionality by adding a meet-calendar, online payments and etc.

	6
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	7
ВСТУП	8
1. СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1 Загальна характеристика предметної області	11
1.2 Постановка завдання	16
1.3 Огляд існуючих рішень	17
1.4 Моделювання предметної області	22
1.4.1 Діаграма прецедентів як інструмент моделювання логіки системи	22
1.4.2. FDD діаграми	24
1.4.3 Діаграма класів	32
1.4.4 Опис функціональності та розгортання системи	33
1.4.5 User Flow сайту	35
2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	37
2.1 Логічна модель Бази Даних	37
2.2 Вибір системи управління інформаційною базою	40
2.3 Створення інформаційної бази	46
3. Розробка системи та її елементів	46
3.1 Вибір засобів розробки	46
3.3 Проектування дизайну інтерфейсу користувача	52
3.4 Характеристика та опис використаних інструментів	57
4. РЕАЛІЗАЦІЯ, ТЕСТУВАННЯ ТА ПОДАЛЬШЕ ВИКОРИСТАННЯ СИСТЕМИ	62
4.2 Тестування системи	66
4.3 Види тестування, які проходила система KeyStone Consulting	68
4.4 Рекомендації щодо експлуатації системи	71
4.5 Опис компонентів системи KeyStone Consulting	73
4.5.1 Головна сторінка системи KeyStone Consulting	73
4.6 Вимоги до апаратного та програмного забезпечення	81
ВИСНОВКИ	83
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	85
ДОДАТОК А	89
ДОДАТОК Б	90
ДОДАТОК В	92
ДОДАТОК Г	94
ДОДАТОК Ґ	95
ДОДАТОК Д	96
ДОДАТОК Е	97

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

UI — User Interface — інтерфейс користувача, графічне середовище для взаємодії користувача із системою.

UX — User Experience — досвід користувача, емоції та враження від взаємодії з вебсайтом або додатком.

HTML — HyperText Markup Language — мова розмітки, яка використовується для створення структури вебсторінок.

CSS — Cascading Style Sheets — каскадні таблиці стилів, що відповідають за оформлення зовнішнього вигляду HTML-елементів.

JS — JavaScript — мова програмування для реалізації динамічної поведінки на вебсторінках.

PHP — серверна мова програмування, яка використовується для створення динамічних вебсайтів і обробки даних на сервері.

SQL — Structured Query Language — мова структурованих запитів для взаємодії з базами даних.

MySQL — система управління реляційними базами даних з відкритим кодом.

IP — Internet Protocol — унікальна адреса пристрою в мережі Інтернет.

UX-тестування — перевірка зручності користування вебресурсом.

UML — Unified Modeling Language — уніфікована мова моделювання, що використовується для проектування систем.

MVP — Minimum Viable Product — мінімально життєздатний продукт, що містить лише основний функціонал для перевірки ідеї.

HTTP — Hypertext Transfer Protocol — протокол передачі гіпертексту в мережі Інтернет.

HTTPS — захищена версія HTTP з шифруванням даних (SSL/TLS).

SEO — Search Engine Optimization — оптимізація сайту для кращого ранжування у пошукових системах.

ВСТУП

Сучасний розвиток цифрових технологій обумовлює суттєві зміни в організації бізнес-процесів, зокрема в частині надання професійних послуг. Однією з галузей, що активно адаптується до нових викликів цифрової трансформації, є фінансовий консалтинг. В умовах економічної нестабільності, глобалізації та високої конкуренції, переважна кількість малих і середніх підприємств а також приватних осіб, потребує професійної підтримки у сфері податкового планування, інвестування, бюджетування, бухгалтерського обліку та загального фінансового управління.

Актуальність дослідження підтверджується статистичними даними: згідно з інформацією аналітичного агентства Competitive Edge Researchers, ринок фінансового консалтингу США в 2024 році оцінювався в 2,4 мільярда доларів США та, за прогнозами, досягне 4,1 мільярда доларів США до 2033 року, зростаючи на 6,5% щорічно. Разом із цим, транснаціональні гіганти “Велика четвірка” та 20 -30 національних корпорацій США обслуговують близько 67% ринку фінансового консалтингу. Клієнтами лідерів є найбільші світові корпорації, банки та фінансові установи, інвестиційні компанії та державний сектор, для яких репутація і бренд має переважне значення. Вартість послуг таких компаній є недоступною для середнього та малого бізнесу.

Таким чином, є низка актуальних проблем пов'язана з впровадженням нових цифрових рішень, які забезпечують не лише якість послуг, але й знижують бар'єри доступу до них. Малі та середні підприємства, маючи обмежений бюджет, потребують якісний онлайн сервіс без складних систем запису, очікування швидкого зворотнього зв'язку, прозорої цінової політики та гнучкості під індивідуальні потреби.

Метою даної дипломної роботи є покращення існуючої сфери надання онлайн фінансових послуг за допомогою розробки функціонального веб-сайту “KeyStone Consulting” для надання онлайн-консультацій з фінансових питань, орієнтованих на цільову аудиторію у США.

Основним завданням є створення платформи, яка забезпечує швидку та зручну взаємодію між користувачем і фінансовим консультантом, з акцентом на мінімізацію часових витрат, простоту використання та оперативну обробку звернень. Ключовою особливістю запропонованого рішення є спрощена логіка роботи: користувач переходить на сайт, обирає потрібну послугу, натискає кнопку “Отримати консультацію”, заповнює коротку форму, після чого його запит автоматично надсилається до Telegram-чату з консультантами. Такий підхід дозволяє забезпечити високий рівень оперативності без потреби у створенні акаунта або авторизації, що є конкурентною перевагою розробленої системи. Водночас, технічна реалізація передбачає можливість масштабування та подальшого розширення функціоналу (наприклад, додавання календаря записів, онлайн-оплати тощо).

Таким чином, веб-сайт “KeyStone Consulting” покликаний сприяти підвищенню доступності якісних фінансових послуг, забезпечити зручний механізм комунікації для малих підприємств та стартапів, а також демонструє приклад ефективного впровадження цифрових технологій у сфері консалтингу. Актуальність, інноваційність підходу та практична значущість зумовлюють важливість обраної теми дослідження та визначають її доцільність у межах підготовки кваліфікаційної роботи

Об’єктом дослідження виступає інформаційна система надання фінансових консультацій та послуг онлайн. Це веб-платформа, яка забезпечує взаємодію між користувачами, які потребують фінансових порад, та консультантами, що ці послуги надають.

Предметом дослідження є проєктування, розробка та впровадження веб-інформаційної системи, яка автоматизує та підтримує основні функції консалтингової компанії, включаючи управління користувачами, планування консультацій, обробку даних та комунікацію.

Методи та засоби розробки веб-сайту для надання фінансових консультацій, з урахуванням вимог до функціональності, зручності користування, інтеграції з сервісами комунікації та специфіки цільового ринку США, передбачають поетапне створення інноваційної онлайн-платформи, 7 орієнтованої на потреби як індивідуальних користувачів, так і малого та середнього бізнесу.

Розробка проєкту включає в себе комплексний аналіз ринку фінансового консалтингу в США з урахуванням сучасних тенденцій, змін у споживчій поведінці та розвитку цифрових сервісів. Значна увага приділяється вивченню конкурентних рішень, їхніх сильних і слабких сторін, що дозволяє виокремити унікальні переваги майбутньої платформи. На основі проведеного аналізу сформовано чіткі функціональні та нефункціональні вимоги, серед яких — зручність інтерфейсу, безпечна обробка персональних даних, надійна комунікація між користувачем і консультантом, швидке реагування на запити, мобільна адаптивність та масштабованість системи.

У технічному аспекті здійснено вибір оптимального технологічного стеку, до якого входять HTML5, CSS3, JavaScript, PHP, MySQL, а також інтеграція з Telegram API для забезпечення швидкого отримання повідомлень консультантами. Створення веб-сайту реалізовано з використанням сучасних інструментів розробки, таких як Visual Studio Code, OpenServer, а також макетування у Figma.

Після реалізації основного функціоналу було проведено тестування системи з метою виявлення можливих недоліків і перевірки працездатності платформи в умовах наближених до реального використання. Тестування охоплювало як функціональні сценарії (правильність обробки запитів, заповнення форм, обробка даних), так і нефункціональні (швидкодія, безпека, сумісність з різними браузерами та пристроями).

1. СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальна характеристика предметної області

Фінансовий консалтинг - це різновид професійних консалтингових послуг, орієнтованих на надання експертної та інформаційної допомоги суб'єктам господарювання у сфері управління фінансово-господарською діяльністю. Така допомога охоплює широкий спектр послуг, зокрема оцінка фінансового стану, розробку фінансової стратегії, оптимізацію витрат, податкове планування, підготовку фінансової звітності, ведення бухгалтерського обліку, оцінку фінансових ризиків, інвестиційний аналіз, а також консультування фізичних осіб щодо управління особистими фінансами[1]. В залежності від предмета фінансового консультування розрізняють наступну класифікацію наведено в Таблиці 1.1.

Таблиця 1.1

Класифікація фінансового консалтингу за предметом послуг

Предмет фінансового консалтингу	Напрями та приклади фінансового консалтингу
Фінансовий аналіз стану	Комплексний фінансовий аналіз стану суб'єкта, факторний аналіз динаміки показників; Виявлення причинно наслідкових зв'язків
Фінансове планування та бюджетування	Бюджетування, створення бізнес моделі, розробка фінансової стратегії
Корпоративні фінанси	Управління фінансовими ресурсами; управління оборотним капіталом та ліквідністю
Бухгалтерський супровід	Ведення обліку доходів, витрат та інших фінансових операцій; Складання та подання фінансових звітів; створення облікової політики; електронний обіг документів
Податковий менеджмент	Оптимізація оподаткування, податкове планування, консультування та супроводження подання податкових звітів; визначення податкових наслідків господарських операцій
Управлінський облік	Організація та впровадження управлінського обліку;

Автоматизація	Консультавання щодо вибору систем автоматизації під вимоги та особливості бізнесу; впровадження автоматизації окремих ділянок чи повного циклу
Аудиторський консалтинг	Надання аудиторських послуг складання звітності, перевірок та консультацій
Залучення інвестицій	Фінансове моделювання щодо стратегії суб'єкта; оцінка бізнесу; супровід дьюдил; супровід залучення банківського капіталу та інше
Функція аутсорс фінансового директора	Консультавання щодо фінансової служби, KPI, мотивації. Юридичні та інші питання

Створено автором на основі джерела [2].

Особливість фінансового консалтингу як послуги полягає в виключній експертності та досвіді консультанта; незалежності сторонньої оцінки; глибокому та всебічному аналізі ситуації; дотриманні чіткого алгоритму надання послуги та впровадження рекомендацій під контролем консультанта або самим консультантом. Таким чином, фінансовий консалтинг вже став ключовим інструментом ефективної системи фінансового менеджменту, прийняття управлінських рішень та подолання викликів турбулентного ринкового середовища.

Ринок фінансового консалтингу США в 2024 році оцінювався в 2,4 мільярда доларів США та, за прогнозами, досягне 4,1 мільярда доларів США до 2033 року, зростаючи на 6,5% з 2026 по 2033 рік щорічно [3]. Ринок фінансового консалтингу США за типом клієнтів поділяється на:

- Великі корпорації та підприємства;
- Інвестиційні компанії;
- Банки та фінансові установи;
- Суб'єкти державного сектору;
- Стартапи (створені мікропідприємства та середні компанії) та домогосподарства.

Світові тенденції ринку фінансового консалтингу формують тренди ринку фінансових послуг в США, а саме домінування транснаціональних гігантів, які фактично визначають політику світового консалтингового бізнесу.

Яскравими представниками таких гігантів вважають «Велику четвірку консалтингових компаній»: Ernst&Young, PwC, KPMG і Deloitte. Маючи величезний ресурс, вони відслідковують новітні запити й зміни та постійно розширюють спектр своїх послуг далеко за межі фінансового консалтингу. Локальний ринок послуг фінансового консалтингу США доповнюється 20-30 великими національними консалтинговими компаніями (Boston Consulting Group, ITConnectUS, Accenture PLC), які разом з транснаціональними гігантами обслуговують близько 67% ринку. Дані компанії мають значний авторитет, всесвітню репутацію та ім'я (бренд). Відповідно, їхніми клієнтами становляться найбільші світові транснаціональні корпорації, банки та фінансові установи, інвестиційні компанії та державний сектор, для яких репутація і бренд має переважне значення. Вартість послуг лідерів є недоступною для середнього та малого бізнесу.

Найбільш поширеною групою учасників ринку фінансового консалтингу США є предметно орієнтовані компанії, чия діяльність обмежується одним або кількома напрямками фінансових послуг, таких як бухгалтерський та податковий супровід, управлінський облік та інше. В таких невеликих консалтингових компаніях працює від 1 до 4 фахівців, які можуть поєднувати роботу з викладанням або залучені на декілька проектів [4]. Саме ця категорія компаній обслуговує малий та середній бізнес з доступною ціновою політикою та гнучкістю під вимоги замовника.

Останні спостереження демонструють підвищення конкуренції серед невеликих консалтингових компаній на ринку США, що спеціалізуються на бухгалтерському та податковому фінансовому консалтингу за рахунок міжнародного розділу праці, адже все більше іноземних фахівців із Індії, України та Філіппін залучаються до надання базових послуг, що не потребують сертифікації, та мають порівняно низьку вартість. Так, наприклад, мінімальна вартість роботи CPA (сертифікованого публічного бухгалтера США) який консультує щодо податкової звітності чи обліку, складає 150 дол. США за годину проти 10 -20 дол. США за годину залучення англомовного фахівця з

вищеперелічених країн. Таким чином, вирішальною конкурентною перевагою таких компаній є цінова привабливість, якість послуг та можливість обирати софт (базу зберігання даних)

Достатня якість послуг таких компаній забезпечується тим, що на більш складні та відповідальні операції залучається сертифікований фахівець CPA на погодинній основі, проте більшість простих але необхідних операцій бухгалтерського/управлінського обліку виконує англomовний фахівець без сертифікації.

Тому, згідно з галузевими дослідженнями, близько 65% представників малого та середнього бізнесу звертаються до зовнішніх невеликих консалтингових компаній чи фінансових консультантів під час ухвалення стратегічних рішень або при настанні кризових ситуацій. Така динаміка зумовлена як посиленням регулювання фінансової діяльності, так і високою швидкістю змін на ринку, що потребують адаптації та професійного супроводу.

Значною перевагою ринку фінансових послуг є стійкий тренд цифровізації. Все більше клієнтів очікують можливості отримувати кваліфіковану допомогу дистанційно, у форматі онлайн, із гарантією конфіденційності, прозорості, зручності та швидкості взаємодії. Відповідно, цифрові платформи повинні відповідати низці критичних вимог: мати інтуїтивно зрозумілий інтерфейс, працювати стабільно на різних пристроях, забезпечувати захист персональних даних відповідно до вимог законодавства (зокрема, до стандартів **California Consumer Privacy Act (CCPA)** та **United States and the General Data Protection Regulation**), а також мати ефективний і надійний механізм зворотного зв'язку.

Більшість існуючих онлайн-рішень у сфері фінансового консалтингу націлені переважно на обслуговування великих та середніх корпоративних клієнтів. Такі сервіси зазвичай мають складні багаторівневі системи реєстрації, верифікації, налаштування облікового запису та високу цінову політику, що створює бар'єри для малих компаній, мікробізнесу та приватних осіб. Це призводить до зниження конверсії та обмежує потенціал масштабування таких

платформ.

У зв'язку з цим виникає потреба у розробці рішення, яке дозволяє максимально спростити процес взаємодії між клієнтом і фінансовим консультантом. Пріоритетом є можливість подати запит на консультацію без обов'язкової реєстрації, у кілька кліків, із використанням популярних комунікаційних інструментів. У межах цього проекту розглядається інтеграція з Telegram API як механізм прямого обміну повідомленнями між клієнтом і консультантом. Вибір Telegram обумовлений його високою швидкістю передачі даних, підтримкою ботів та webhook-запитів, стабільною роботою у різних мережах, а також можливістю використання на всіх популярних операційних системах. У порівнянні з іншими платформами, такими як WhatsApp або Slack, Telegram вирізняється більш відкритою API-структурою, широкими можливостями кастомізації, а також вищим рівнем гнучкості в частині реалізації бізнес-логіки без необхідності проходження жорстких політик модерації. Крім того, згідно з останніми опитуваннями, Telegram поступово набирає популярності серед молодшої англійської аудиторії США, що робить його перспективним для цільового ринку.

Для формування функціонального рішення необхідно враховувати конкретні потреби цільової аудиторії. До неї належать: мікропідприємства, стартапи, фрілансери, а також приватні особи без глибоких фінансових знань та з обмеженим бюджетом. Типові запити таких користувачів стосуються: швидкої оцінки податкових зобов'язань, вибору схеми реєстрації бізнесу, базової консультації з ведення обліку, планування витрат, підготовки до податкового сезону або первинне залучення інвестицій. Ключовими очікуваннями користувача є оперативність відповіді, простота інтерфейсу, відсутність зайвої бюрократії, невеликий бюджет, а також гнучкість часу взаємодії з консультантом.

Враховуючи ці особливості, розробка веб-платформи має базуватися на принципах адаптивного дизайну, простоти використання, конфіденційності та високої продуктивності. Доцільним є впровадження таких функціональних

блоків, як рейтинг консультантів, швидкі відповіді на типові запити (FAQ), можливість бронювання консультацій за графіком, інтеграція з популярними платіжними системами (Stripe, PayPal, банківські картки), а також кабінет адміністратора для контролю якості обслуговування.

Окрему увагу слід приділити правовим ризикам, що стосуються обробки персональних даних громадян США відповідно до закону ССРА. Він передбачає обов'язок компаній інформувати користувачів про те, які саме дані збираються, як і з якою метою вони використовуються, а також надає користувачам право вимагати видалення або передачі їхніх даних. Для забезпечення відповідності законодавству система повинна мати механізм інформування, політику конфіденційності, а також форму запиту на видалення або експорт даних.

1.2 Постановка завдання

Основна мета цієї системи полягає у веденні комплексного обліку операційних показників консалтингової компанії, таких як активні клієнти, поточні проєкти, терміни, робоче навантаження консультантів, статус виставлення рахунків та рейтинги задоволеності клієнтів. Програма повинна створити та керувати базою даних, яка зберігає всі необхідні дані, що дозволяє своєчасно та персоналізовано отримувати сповіщення щодо термінів, очікуваних рахунків-фактур та розподілу робочого навантаження.

Основна функціональність програмної системи полягає у наданні персоналу компанії миттєвого доступу до інформації щодо:

- відстеження продуктивності та статусу проєкту;
- розподілу завдань консультантів та поточного навантаження;
- історії спілкування з клієнтами та рейтингів відгуків;
- створення рахунків-фактур, статусу платежів та сповіщень про прострочення;
- наближених термінів та конфліктів у плануванні.

Додаток розроблено для роботи у форматі меню з діалоговим вікном. Це дозволяє користувачеві інтуїтивно взаємодіяти з системою, вибираючи відповідні дії з категоризованих меню. На основі обраної операції користувач може реєструвати нові дані, отримувати детальні звіти або скасовувати/оновлювати попередні записи. Це гарантує, що керівництво завжди має актуальну та практичну інформацію для прийняття рішень та покращення надання послуг.

1.3 Огляд існуючих рішень

Як описувалось вище, на ринку США існує велика кількість предметно орієнтованих компаній, що надають консалтингові фінансові та бухгалтерські послуги в автоматизованому форматі стартапам та домогосподарствам. До таких відносяться Fuel Finance; AcoBloom International; CapActix; Fondo.

Fuel Finance, компанія з українськими коріннями, що вийшла на ринок фінансових консалтингових послуг США та спеціалізується на візуалізації управлінської звітності через власно розроблені дашборди, що представлено в особистому акаунті клієнта. Додатково надає послуги кастомних фінансових звітів своїм клієнтам як юніт економіка, план/факт аналіз фінансової звітності та інші. **Недоліки:** для формування звітів через платформу Fuel Finance, потрібно мати бухгалтерський облік в певному форматі, який відповідає формату та вимогам власного продукту. Якщо бухгалтерський облік відсутній в стартапі, то супроводження бухгалтерської звітності буде додатковим бюджетом і буде залучатися аутстаф бухгалтер. Є обмеження дощого бухгалтерського софту. Ціновий діапазон стартує з 700 дол США, проте на даний час, компанія приховала цінову пропозицію [6]. Досить складний процес реєстрації та валідації вимог замовника через заповнення форми на сайті.

AcoBloom International, консалтингова компанія, що надає послуги з бухгалтерського обліку, складання фінансової звітності, підготовки та подання федеральних податкових звітів, супроводження зарплатного проекту, та допомога при аудитах. Є 3 види пакетного співробітництва з клієнтами:

- залучення спеціаліста на повний робочий день - не менше 140 годин/місяць;
- погодинна співпраця , проте сама платформа не рекомендує цей вид;
- по факту подачі податкових звітів.

Є затверджений алгоритм взаємодії з клієнтами з певними метриками особливо на перших етапах. Платформа передбачає використання будь-якого бухгалтерського та податкового софту клієнта QuickBooks Online; Zoho Books; XERO; Gusto; TurboTax, що є досить зручно для компаній, бо дозволяє зберігати базу даних на боці замовника. **Недоліки:** відсутня прозора цінова політика на сайті. Замовлення послуги через дзвінок або електронну пошту. Досить обмежений пакетний план [5].

CapActix, консалтингова компанія, яка позиціонує себе як 2 окремих сервіси аутстаф персонал та аутстаф експертні консультації та фінансові консалтингові послуги. Перелік послуг дуже деталізований і націлений на значні обсяги операцій , що є ознакою великого бізнесу. Платформа передбачає використання будь-якого бухгалтерського та податкового софту клієнта. При плануванні консультації можна використовувати сервіс календаря. **Недоліки:** відсутня прозора цінова політика на сайті. Замовлення послуги через дзвінок або електронну пошту. Велика деталізація окремих сервісів, що значно ускладнює сприйняття [7].

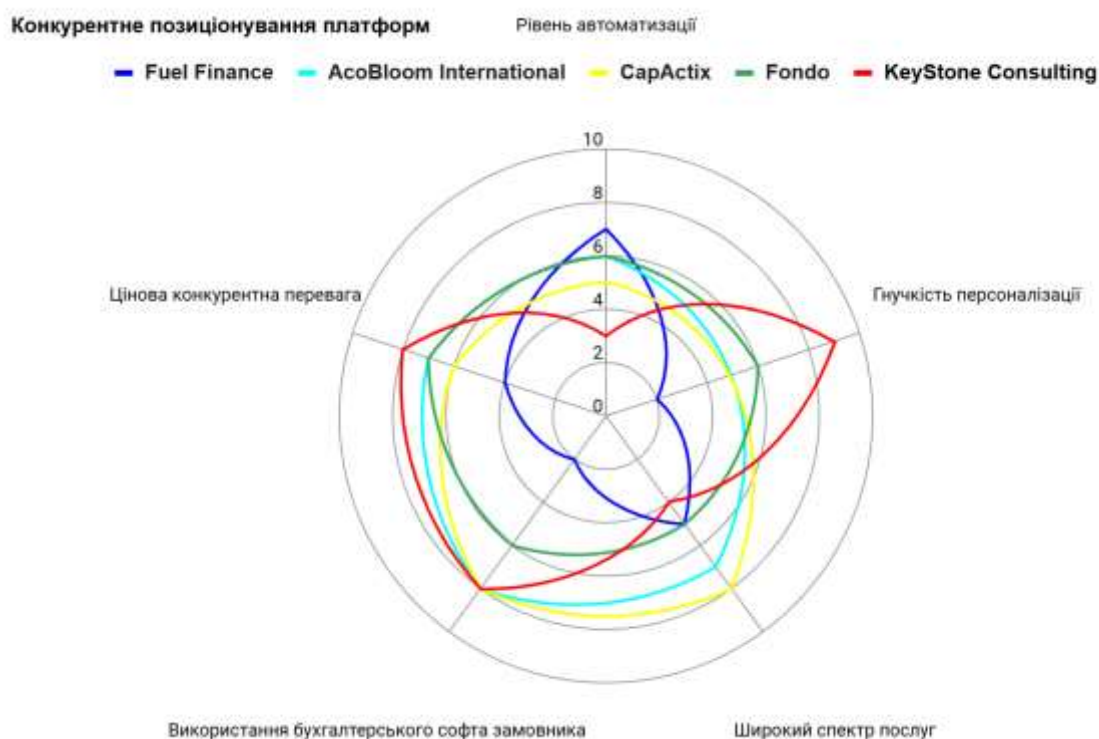
Fondo, платформа є **прямим конкурентом** KeyStone Consulting, оскільки також функціонує у сфері онлайн-фінансового супроводу для малого бізнесу з обмеженим спектром послуг, що характерно для стартапів та домогосподарств. Платформа передбачає використання будь-якого бухгалтерського та податкового софту клієнта QuickBooks Online; Zoho Books; XERO; Gusto; TurboTax та інші. **Недоліки:** оформлення замовлення через деталізовану форму на сайті. Цінова пропозиція стартує з 599 дол. США [8].

Одним з основних завдань аналізу було визначити позиціонування нашого продукту відносно існуючих рішень, щоб виявити конкурентні переваги, слабкі місця, а також можливості для масштабування. **KeyStone**

Consulting вирізняється тим, що пропонує обмежений спектр послуг з аспектом на клієтоцентрацію, тобто, **формат індивідуальних консультацій**, який забезпечує максимально простий спосіб отримати професійну допомогу. Вся логіка взаємодії побудована навколо миттєвого запиту через сайт і подальшої комунікації з консультантом через інтеграцію з Telegram. Таким чином, наша платформа не нав'язує клієнту довготривалу підписку або необхідність опрацьовувати складні інтерфейси, а натомість дозволяє швидко отримати відповідь на конкретне фінансове питання.

Рисунок 1.1 - Конкурентне позиціонування платформ

KeyStone Consulting займає окрему нішу — **індивідуального, швидкого і гнучкого фінансового консалтингу**, без прив'язки до особистого супроводу



або інтерфейсної складності, з **конкурентними ціновими перевагами** та можливістю використання будь-якого бухгалтерського та податкового софту клієнта QuickBooks Online; Zoho Books; XERO; Gusto; TurboTax та інші

Діаграма **конкурентного позиціонування платформ фінансового консалтингу**, зображена на Рис. 1.1, була побудована на основі якісного аналізу ключових характеристик кожного з досліджуваних сервісів: Fuel

Finance, AcoBloom International, CapActix, Fondo, KeyStone Consulting.

Оцінювання проводилось за декількома основними критеріями:

1. **Рівень автоматизації** — відображає, наскільки система здатна самостійно виконувати бухгалтерські чи фінансові операції без участі консультанта, та відображати візуалізацію певних метрик та показників. (наприклад, формування звітності, інтеграція з банками, динаміка руху коштів). Платформи Fuel Finance та AcoBloom International мають високий рівень автоматизації, тоді як KeyStone Consulting фокусується на «живому» консультуванні.
2. **Гнучкість персоналізації** — визначає, наскільки система дозволяє адаптувати послуги під індивідуальні потреби клієнта. KeyStone Consulting має високий рівень гнучкості завдяки індивідуальному підходу, прямому контакту з консультантом і можливості працювати без шаблонних форм. Інші системи більш стандартизовані та орієнтовані на типові сценарії.
3. **Широкий спектр послуг** — відносно широкий спектр послуг мають платформи CapActix, AcoBloom International. KeyStone Consulting та Fondo фокусуються на обмежених опціях, що притаманне малому бізнесу та стартапам.
4. **Використання бухгалтерського софту замовника** — це важливий критерій, який забезпечує незалежність клієнта щодо вибору та зберігання конфіденційної бази даних клієнта QuickBooks Online; Zoho Books; XERO; Gusto; TurboTax. Можливість робити резервні копії бази даних для безпеки. Більшість консалтингових компаній використовують цей інструмент в інтересах клієнтів, проте є компанії які працюють виключно на власних продуктах як Fuel Finance, що значно обмежує компанію.
5. **Цінова конкурентна перевага** — мінімальна цінова пропозиція стартового чи базового пакету. За рахунок застосування міжнародного поділу праці, цінова пропозиція KeyStone Consulting є самою привабливою.

SWOT-аналіз — це метод стратегічного планування, який дозволяє оцінити **сильні та слабкі сторони** проєкту, а також **можливості та загрози**, що

впливають на його розвиток. Він допомагає зрозуміти внутрішній потенціал системи (Strengths, Weaknesses) і зовнішнє середовище (Opportunities, Threats), визначити напрямки вдосконалення та сформувавши ефективну стратегію просування продукту.

Сильні сторони: KeyStone Consulting пропонує надзвичайно просту та зручну модель отримання консультацій: без реєстрації, через знайомий канал комунікації (Telegram). Швидкий час відповіді (до 30 хвилин), інтуїтивний інтерфейс і відсутність зобов'язань зі сторони клієнта створюють унікальну ціннісну пропозицію для мікробізнесу, фрілансерів та фізичних осіб.

Слабкі сторони: Низький рівень автоматизації та обмежена функціональність у частині обліку документів, низький рівень візуалізації звітів для клієнтів, обмежених типовими інструментами бухгалтерського софту. Крім того, сервіс критично залежить від стабільності Telegram API та відповідності до вимог захисту даних.

Можливості: Зростаючий попит на індивідуалізовані послуги в умовах насиченого автоматизованого ринку. Є потенціал для розширення функціоналу через інтеграцію з платіжними системами, календарями, CRM або розробкою власних дашбордів, а також для виходу на нові ринки, зокрема країни ЄС.

Загрози: Інтенсивна конкуренція з боку аналогічних систем, які можуть швидко впровадити аналогічні функції. Крім того, правові вимоги щодо захисту персональних даних (CCPA, GDPR) можуть вимагати додаткових інвестицій у безпеку та юридичний супровід.

Деталізована порівняльна характеристика з цільовим конкурентом TryFondo представлена в Таблиці 1.2 за основними параметрами платформ.

Таблиця 1.2

Порівняння KeyStone Consulting з цільовим конкурентом

TryFondo	KeyStone Consulting
Цільова аудиторія - Малі та середні підприємства, підприємці	Цільова аудиторія - Переважно фізичні особи, підприємці та мікробізнес у США
Інтерфейс користувача - Багатофункціональний, але складний, вимагає реєстрації	Інтерфейс користувача - Мінімалістичний, простий, без обов'язкової реєстрації
Процес замовлення послуги - Користувач проходить кілька кроків, вибирає послугу, заповнює детальну форму	Процес замовлення послуги - Односторінкова форма з базовими даними (ім'я, прізвище, телефон, e-mail)
Механізм зворотного зв'язку - Через особистий кабінет або e-mail	Механізм зворотного зв'язку - Telegram-бот або чат для миттєвої обробки запиту
Можливість персоналізації - Висока (через обліковий запис і CRM)	Можливість персоналізації - Обмежена, без CRM, лише індивідуальна комунікація

1.4 Моделювання предметної області

1.4.1 Діаграма прецедентів як інструмент моделювання логіки системи

Для моделювання функціональної взаємодії користувача із системою KeyStone Consulting використовується діаграма прецедентів, яка дозволяє візуалізувати основні сценарії використання та ролі учасників процесу. Учасники, види активності та описи діаграми прецедентів наведені в Таблиці 1.2. Візуалізація взаємодії Діаграми прецедентів наведена на Рис. 1.2

Діаграми варіантів використання також сприяють комунікації між зацікавленими сторонами, надаючи просте та інтуїтивно зрозуміле візуальне уявлення про можливості системи без заглиблення в технічні деталі. Це

допомагає у ранній перевірці вимог на етапі розробки та керує наступними етапами проєктування та впровадження [14].

Підсумовуючи, діаграма варіантів використання виступає як базовий інструмент для узгодження очікувань, визначення меж та організації функціональних вимог системи навколо реальної взаємодії користувачів.

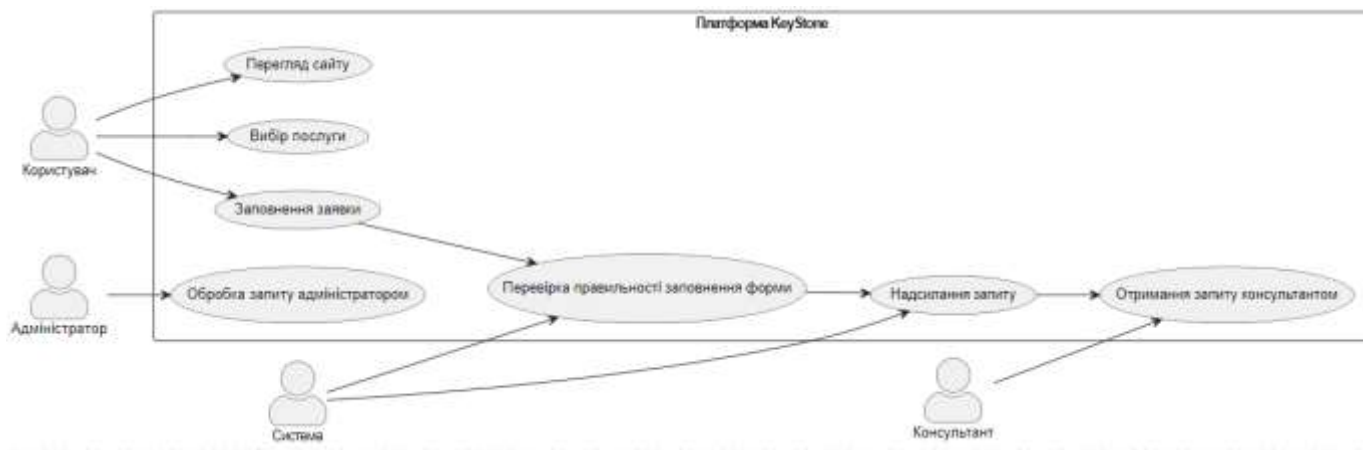


Рисунок 1.2 Діаграма прецедентів

Ця система підтримує різноманітні важливі операції. Керування користувачами дозволяє адміністраторам створювати, змінювати та видаляти облікові записи для консультантів, менеджерів та клієнтів. Реєстрація клієнтів дозволяє консультантам або адміністраторам ефективно залучати нових клієнтів. Після зустрічей консультанти документують важливі нотатки та результати, забезпечуючи точне ведення обліку. Крім того, створення звітів є ключовою функцією, доступною консультантам та менеджерам для оцінки індивідуальної та організаційної ефективності. Перегляд цих звітів також доступний менеджерам та клієнтам, забезпечуючи прозорість та розуміння.

Для забезпечення безперебійної роботи та безпеки адміністратори відповідають за налаштування системних параметрів, ролей та дозволів.

Таблиця 1.3

Параметри Діаграми прецедентів

Основний актор	Назва прецеденту	Опис прецеденту
Користувач	Перегляд	Користувач відкриває вебсайт та

(User)	сайту	ознайомлюється з послугами
Користувач	Вибір послуги	Користувач обирає бажану фінансову послугу
Користувач	Заповнення заяви	Користувач вводить свої контактні дані.
Система	Отримання запиту консультантом	Консультант отримує повідомлення з даними користувача.
Система	Надсилення запиту	система приймає заявку і надсилає її у Telegram-чат.
Система	Перевірка правильності заповнення форми	Система перевіряє, чи всі поля заповнені та чи є помилки у форматі даних (наприклад, неправильний e-mail)
Адміністратор	Обробка запиту адміністратором	Адміністратор платформи може переглядати запити

1.4.2. FDD діаграми

FDD-діаграма є важливим інструментом у процесі проектування програмного забезпечення, особливо в рамках розробки таких систем, як мій дипломний проект: веб-платформа KeyStone Consulting для онлайн-фінансових консультацій. Вона базується на принципах функціонально-орієнтованого підходу до розробки, де основна увага приділяється створенню окремих, чітко визначених функціональних можливостей системи [15]. FDD-діаграма дає змогу структурувати усі дії та процеси, які має виконувати система, з урахуванням потреб кінцевих користувачів. Це дозволяє створити зрозумілу логіку взаємодії між користувачем, системою та адміністратором, а також візуально представити основні етапи роботи сервісу. FDD-діаграма представлена на Рис.1.3.

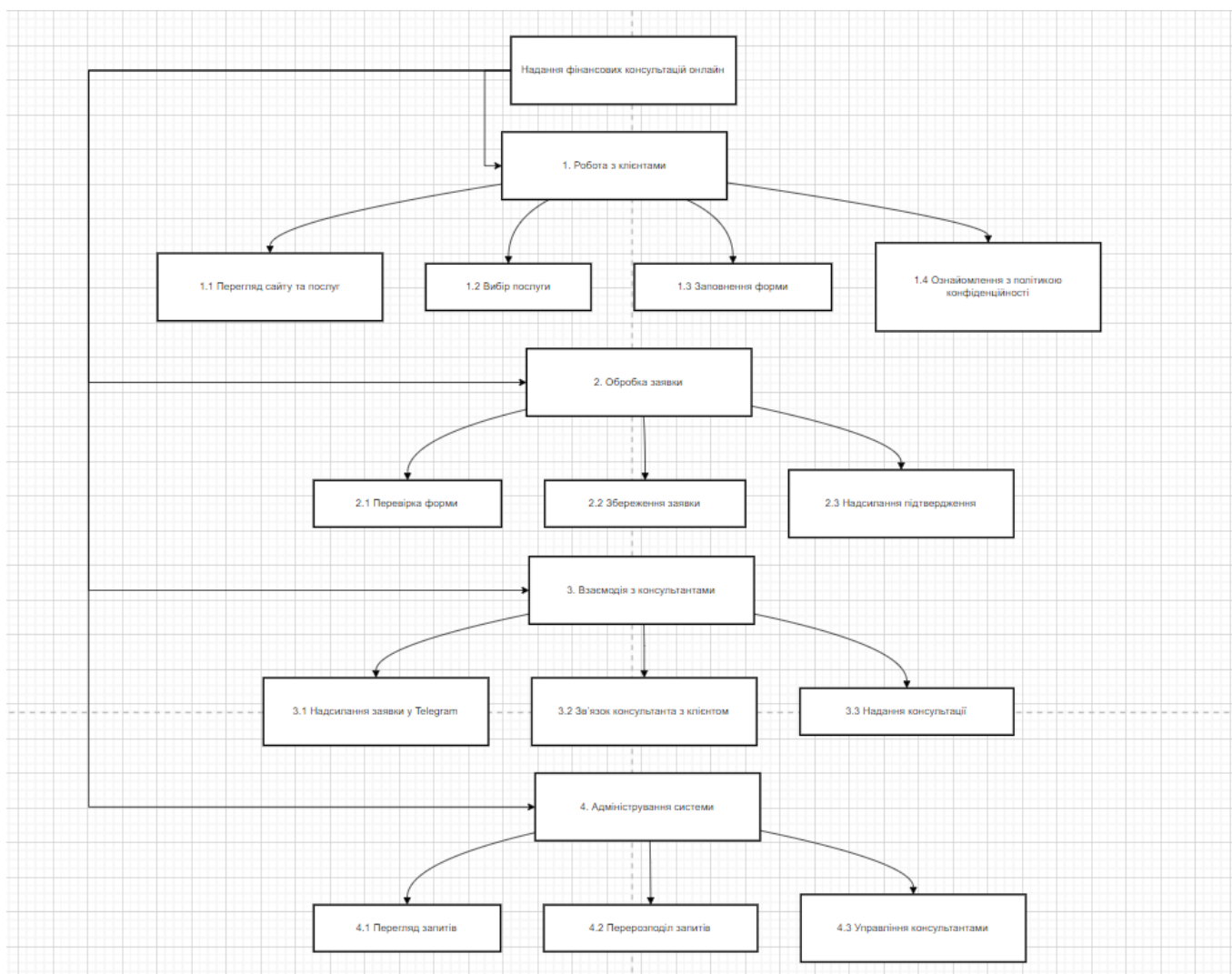


Рисунок 1.3 Візуалізація FDD діаграми

В контексті цього проєкту FDD-діаграма допомагає чітко уявити увесь функціонал: від перегляду інформації про послуги до заповнення заявки, її обробки та зворотного зв'язку. Завдяки цій діаграмі можна виявити, які функції потрібно реалізувати в першу чергу, які є допоміжними, а також як вони взаємопов'язані. Така структура сприяє ефективному плануванню та розподілу завдань під час розробки системи, а також полегшує тестування, перевірку працездатності та написання технічної документації. Важливо й те, що FDD-діаграма орієнтована на користувача, оскільки кожна функція на діаграмі описує саме те, що користувач очікує отримати від системи.

Окрім цього, така діаграма спрощує комунікацію в команді, адже

наочно демонструє логіку та обсяг функціональності, що підлягає реалізації. У цьому випадку це функції: перегляду сервісів, вибору типу консультації, заповнення форми, перевірки правильності введених даних, автоматичної передачі заявки в Telegram-бот, а також подальшої обробки цієї заявки адміністратором. Таким чином, FDD-діаграма виконує не лише технічну роль, а й забезпечує цілісність бачення проєкту, логічну послідовність дій та ефективну реалізацію ключових елементів системи.

Другим важливим елементом побудови програмного забезпечення є DFD діаграма. DFD-діаграма - це схема, яка показує, як інформація рухається в межах системи: звідки надходять дані, куди передаються, як обробляються, і де зберігаються. Вона не описує, як саме реалізується система з технічного боку, а концентрується на логіці обробки інформації [16].

Основна мета DFD — допомогти зрозуміти, як працює система з погляду обміну даними. Вона показує взаємозв'язки між користувачами, модулями програми та базами даних. Це зручно як для розробників, так і для замовника, адже все подано у вигляді зрозумілих блоків і стрілок та представлена на Рис.1.4.

У цій дипломній роботі DFD-діаграма застосовується, щоб чітко уявити, як функціонує система онлайн-фінансових консультацій. Вона показує, як користувач подає заявку, другим кроком - як система її обробляє, куди передає дані (наприклад, у Telegram-бот або базу даних) та як консультант отримує запит. Це дозволяє краще продумати логіку системи, уникнути помилок у розробці й показати, що наша система структурована та ефективна.

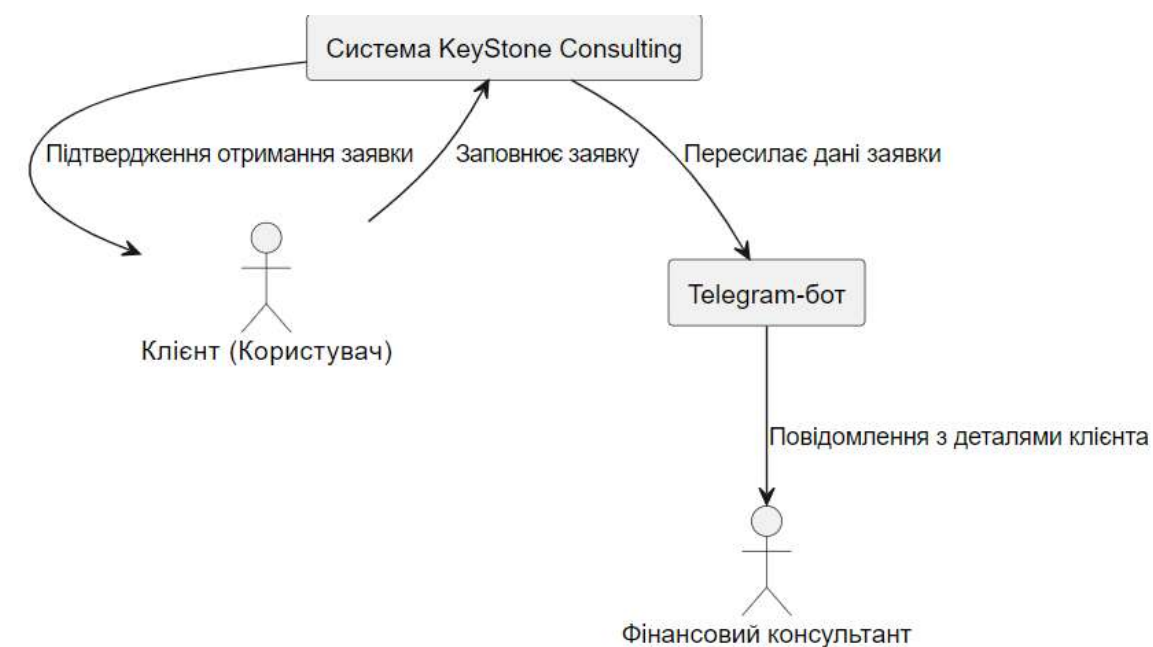


Рисунок 1.4 DFD Повна контекстна модель з Telegram-ботом

DFD діаграма, на Рис.1.4 ілюструє зовнішню структуру системи KeyStone Consulting. Вона показує, як клієнт взаємодіє з платформою через форму заявки. Система автоматично пересилає дані до Telegram-бота, який передає запит фінансовому консультанту. Окремим потоком система надсилає клієнту підтвердження щодо вдалого завершення комунікації. Такий підхід демонструє прозорий і швидкий механізм комунікації між усіма учасниками.

Процес надання фінансових послуг передбачає наявність та обробку бази персональних даних клієнтів, інформації руху коштів, рахунків, листів, транзакцій, відповіді консультантів та інше. Бази даних та персональні дані є конфіденційною інформацією і потребують захисту. В даному проєкті робота з базами даних представлена на Рис. 1.5. Діаграма Контекстної моделі деталізує контекстну взаємодію, включаючи збереження заявок у базі даних і роль адміністратора. Вона демонструє, як система фіксує вхідні дані, а адміністратор може здійснювати доступ до бази для управління або аналізу. Такий підхід важливий для безпеки, зберігання інформації та подальшої звітності. Це важливий елемент масштабованості та сучасності

платформи.

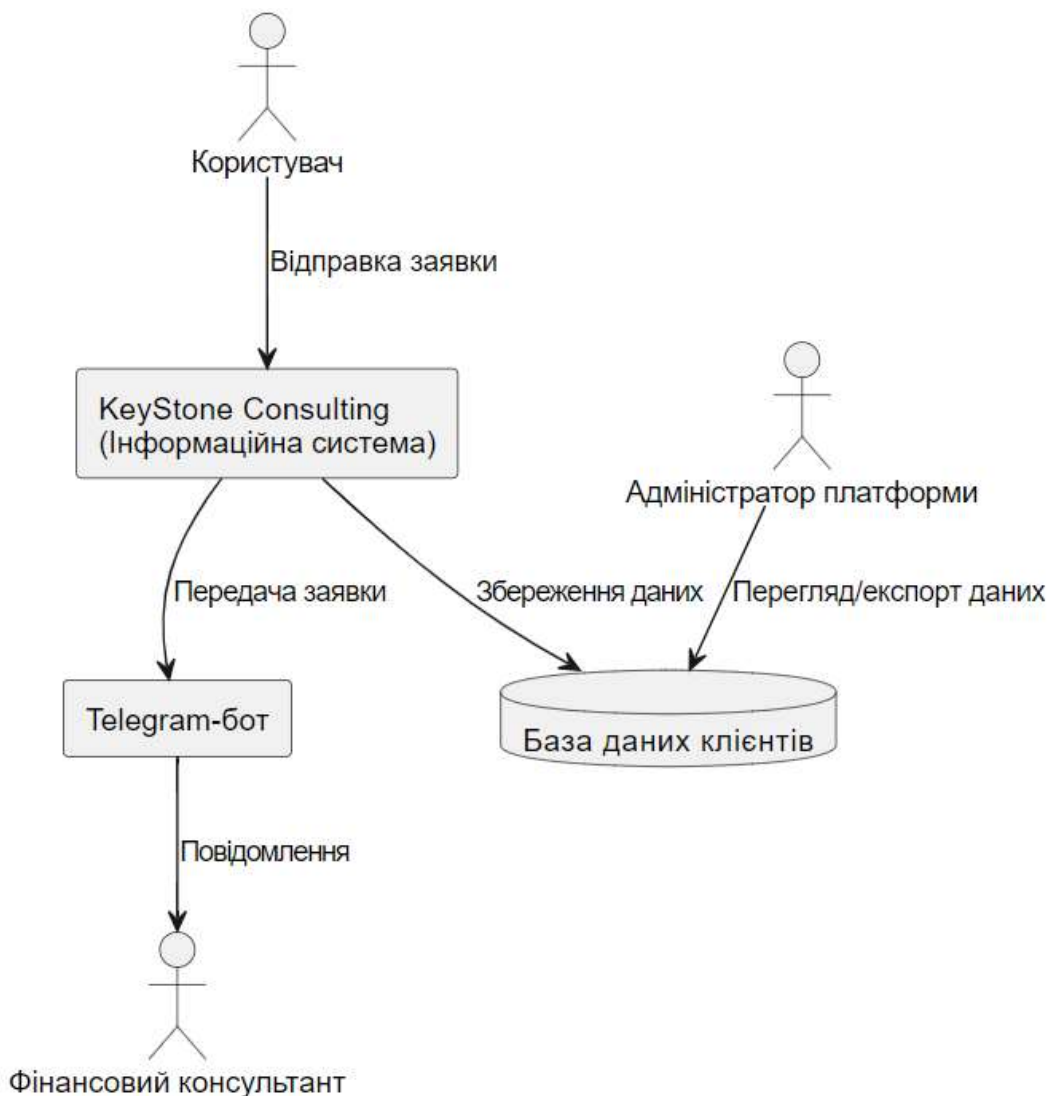


Рисунок 1.5 DFD Контексна модель з базою даних і адміністратором

Діаграма Контекстної моделі деталізує контекстну взаємодію, включаючи збереження заявок у базі даних і роль адміністратора. Вона демонструє, як система фіксує вхідні дані, а адміністратор може здійснювати доступ до бази для управління або аналізу. Такий підхід важливий для безпеки, зберігання інформації та подальшої звітності. Це важливий елемент масштабованості та сучасності платформи.

Варіант взаємодії учасників системи, представлений на Рис.1.6 моделює двосторонню комунікацію між клієнтом і консультантом. Після відправки заявки вона зберігається в базі даних, надсилається консультанту, а після відповіді консультанта — результат повертається клієнту. Дана модель демонструє повний цикл зворотного зв'язку, що може бути частиною

розширеного функціоналу, наприклад, чат або електронна переписка.

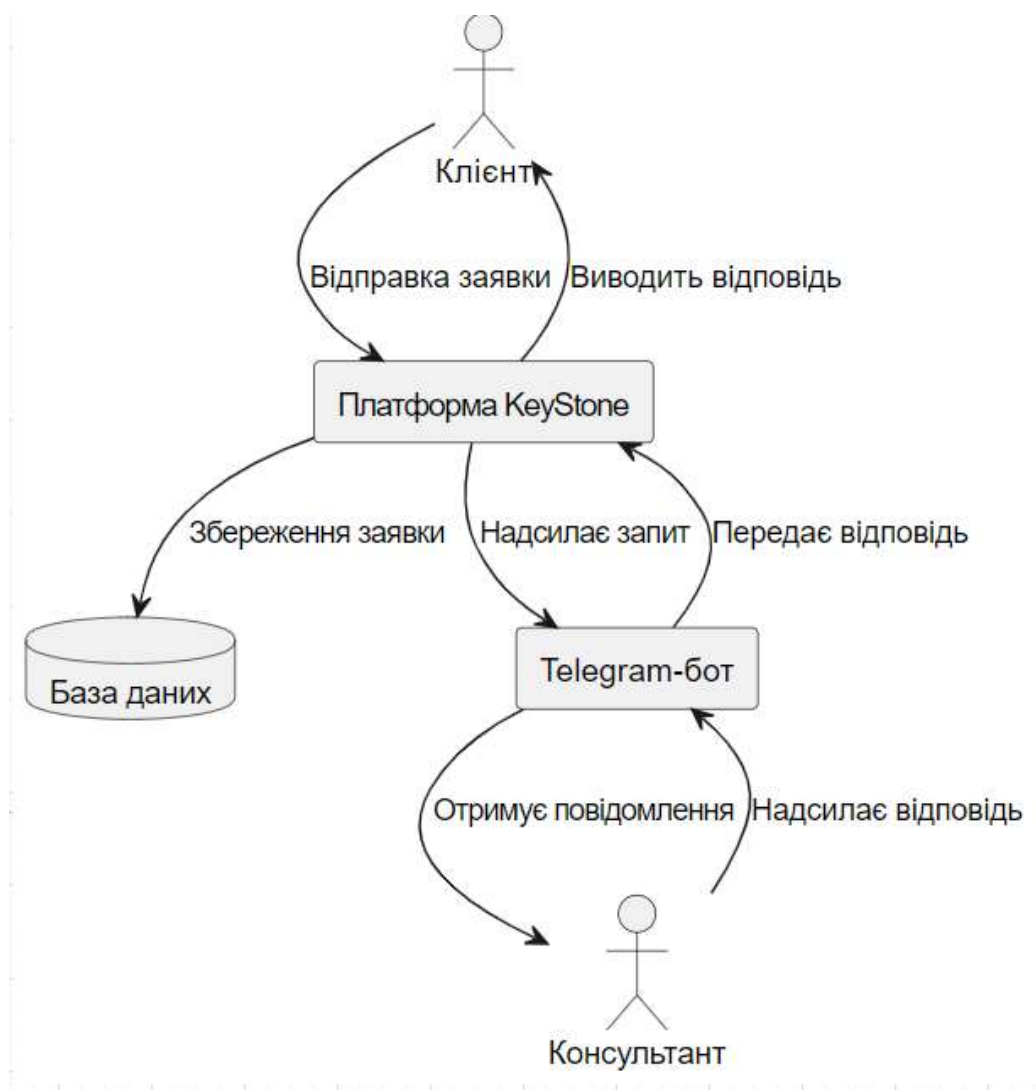


Рисунок 1.6 Зворотна відповідь від консультанта

По внутрішній логіці системи наступним етапом є типовий сценарій, що представлений на Рис.1.7. Обробка заявки користувача показує послідовність внутрішніх дій після заповнення форми: отримання форми, перевірка правильності, зберігання заявки в базі даних і передача даних через Telegram-бот. Це найтипівіший сценарій, що реалізує весь шлях даних у межах однієї транзакції.

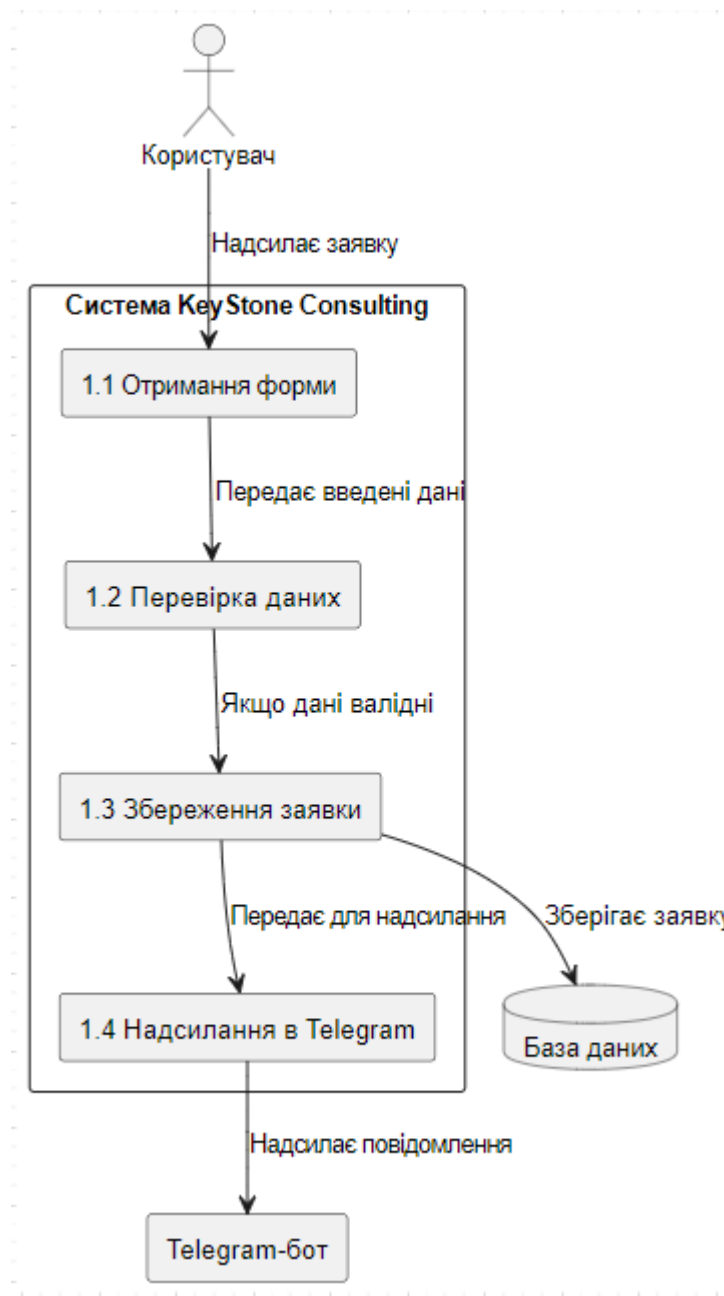


Рисунок 1.7 Обробка заявки користувача

DFD діаграми першого рівня (Level 1) по системі KeyStone Consulting. Перший рівень DFD діаграми деталізують, як система працює всередині — які етапи виконує після того, як користувач надіслав заявку.

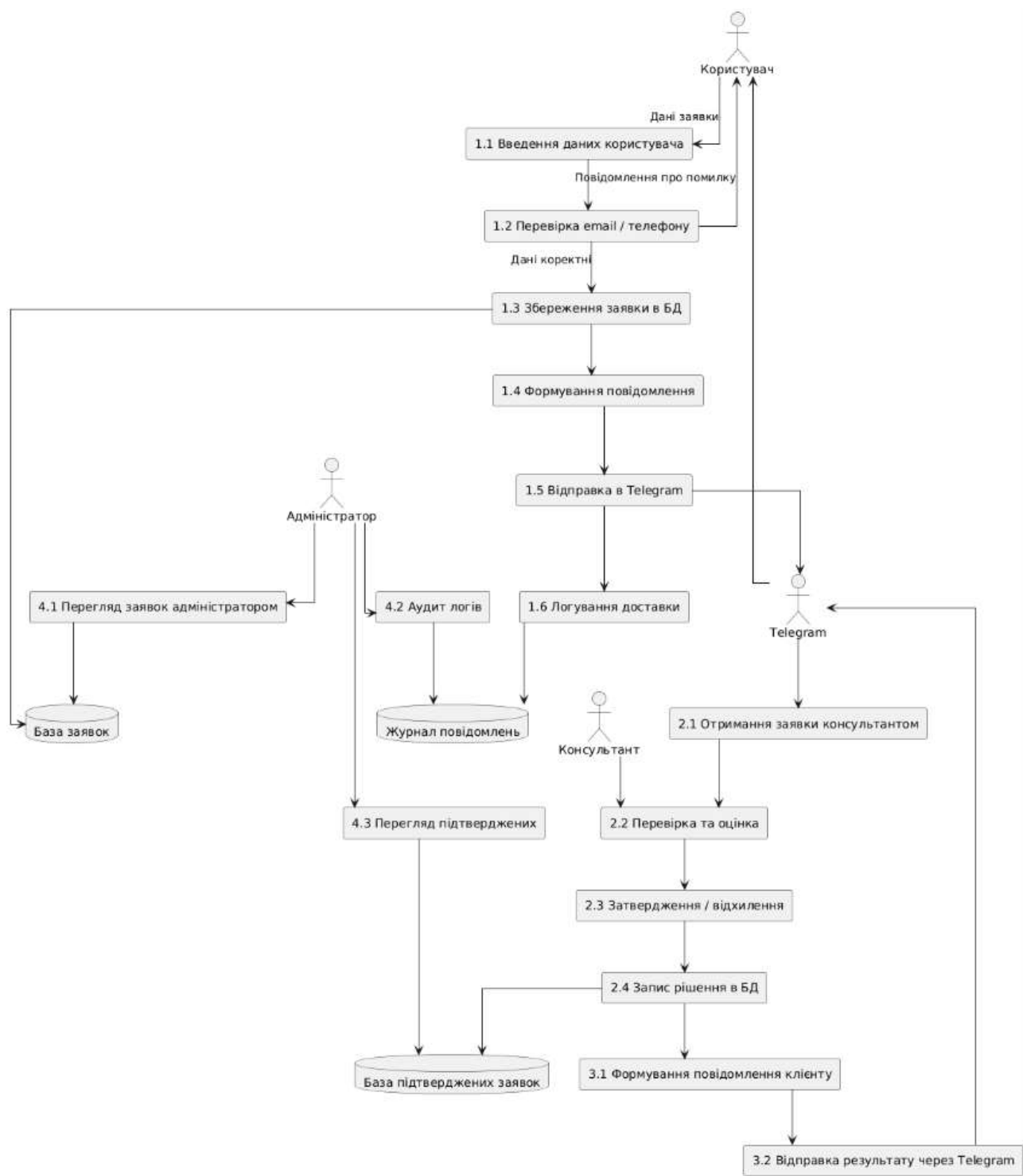


Рисунок 1.8 DFD діаграма функціональної структури системи

На Рис.1.8 представлена DFD-діаграма першого рівня системи KeyStone Consulting. Вона демонструє детальний розподіл процесів обробки заявки: від подання користувачем і перевірки до передачі даних консультанту, фіксації рішення, логування та зворотного зв'язку. Застосовується послідовний підхід до побудови, де кожна функціональна операція подана як окремий процес із власними вхідними та вихідними потоками даних. Така модель дає змогу чітко

відстежити шлях заявки, забезпечити прозорість бізнес-логіки та закласти основу для створення DFD другого рівня.



Рисунок 1.9 Деталізація логування надсилання повідомлення

1.4.3 Діаграма класів

UML-діаграми - це універсальний інструмент для візуального моделювання програмного забезпечення, який дозволяє описати структуру, поведінку та взаємодію компонентів системи. Вони застосовуються на різних етапах життєвого циклу програмного продукту — від аналізу вимог до реалізації, тестування та супроводу [8].

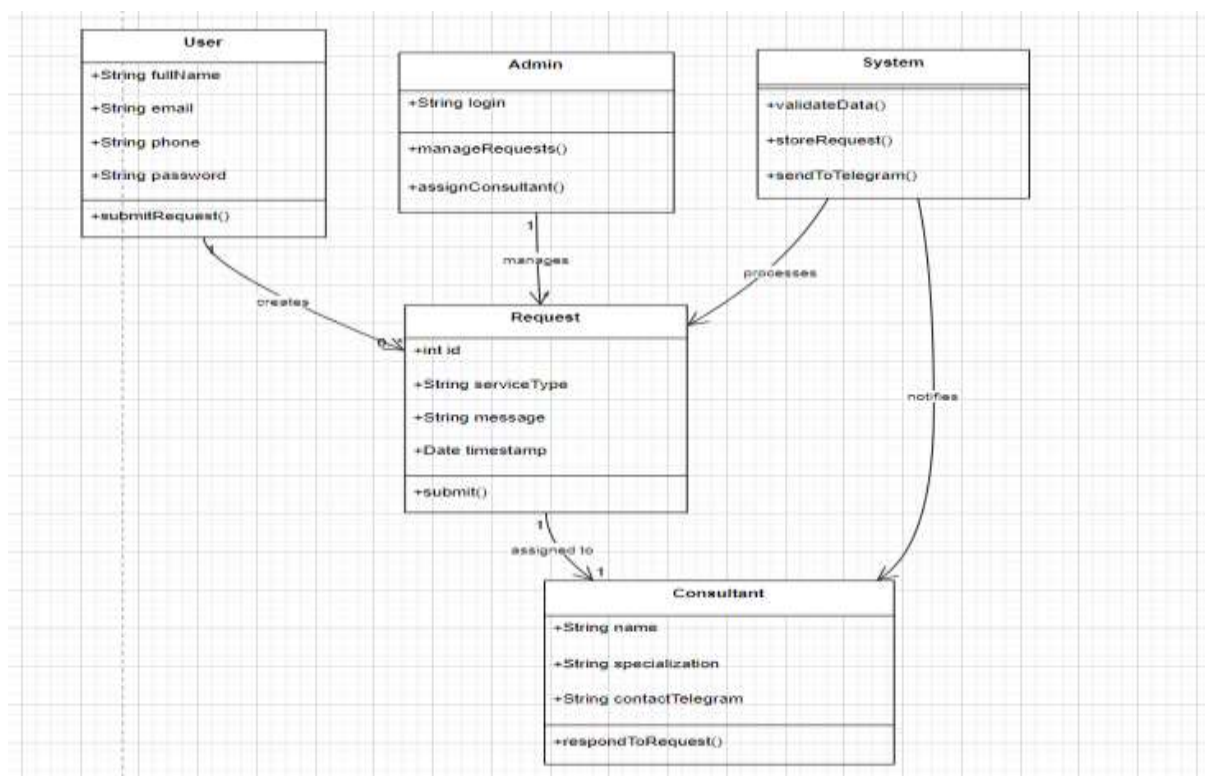
UML-діаграма класів у цій дипломній роботі використовується для відображення структури основних об'єктів системи KeyStone Consulting та наведена на Рис.1.10 UML-діаграма показує, які класи існують у системі, які атрибути та методи їм притаманні, а також як ці класи пов'язані між собою. Ця діаграма дозволяє логічно описати взаємодію між користувачем, заявкою, консультантом та адміністратором. Вона слугує основою для проектування бази даних і реалізації функціональної частини веб сайту. Діаграма класів також

полегшує розробку та підтримку системи, забезпечуючи чітке розуміння її архітектури.

Рисунок 1.10 Діаграма класів

1.4.4 Опис функціональності та розгортання системи

Use Case діаграма (діаграма сценаріїв) — це елемент мови UML, який використовується для опису функціональності системи з точки зору користувачів. Вона відображає, які дії (сценарії) може виконувати кожен тип користувача у системі та як ці дії взаємопов'язані. В даному дипломному



проєкті KeyStone Consulting така діаграма представлена на Рис. 1.11 дозволяє:

- візуально показати, які сценарії взаємодії передбачає система (від подачі заявки до консультації);
- визначити ролі користувачів: клієнта, консультанта, адміністратора;
- узгодити функціональність із технічним завданням;
- спростити комунікацію між розробником, аналітиком і кінцевим замовником.

Це ключова діаграма для розуміння зовнішньої логіки використання систем

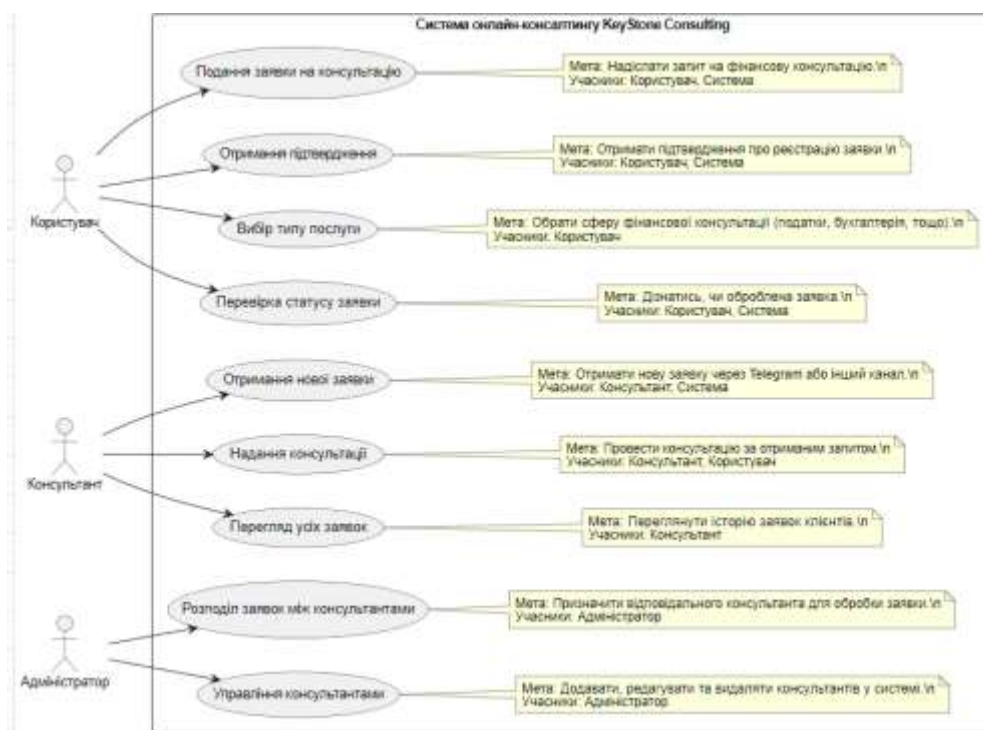


Рисунок 1.11 Use Case діаграма

У системі KeyStone Consulting користувач подає заявку, обирає тип послуги, отримує підтвердження та може перевірити статус звернення. Консультант отримує заявки, надає консультації та переглядає історію звернень. Адміністратор розподіляє заявки між консультантами та керує їхніми профілями. Use Case діаграма показує взаємодію кожного учасника з системою для забезпечення ефективної роботи сервісу.

Діаграма розгортання показує, як фізично розміщуються компоненти системи KeyStone Consulting у реальному середовищі: на яких пристроях вони працюють, які між ними зв'язки та як забезпечується обробка запитів користувачів.

У цій дипломній роботі ця діаграма відображає ключову інфраструктуру системи. Клієнт взаємодіє з платформою через веб браузер, що працює на користувацьких пристроях: комп'ютерах, планшетах або смартфонах. Інтерфейс надсилає запити до серверної частини (backend), яка розміщується на хостингу або хмарному сервері. Сервер обробляє логіку запиту, перевіряє дані, зберігає їх у базі даних та при необхідності надсилає повідомлення через Telegram-бот.

Окремим вузлом виступає база даних, яка фізично розташована на окремому сервері або в межах тієї ж інфраструктури, що й вебсервер. Telegram-бот працює як зовнішній сервіс, що інтегрується через API і виконує функцію оперативного інформування консультантів.

Діаграма розгортання, наведена на Рис 1.12 допомагає наочно продемонструвати, як саме реалізована архітектура системи, які компоненти відповідають за ту чи іншу функцію та як вони між собою з'єднані. Вона особливо важлива для пояснення масштабованості, безпеки, каналів обміну даними та технологічного підходу, що використовувався під час розробки системи.

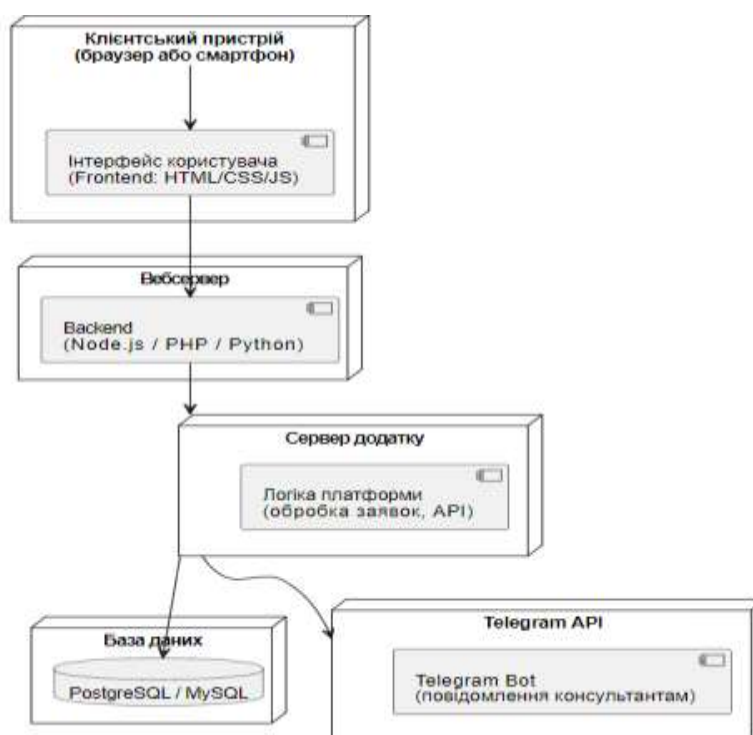


Рисунок 1.12 Діаграма розгортання

1.4.5 User Flow сайту

User Flow — це техніка, яка візуально показує шлях, яким проходить користувач у системі від першої взаємодії до досягнення конкретної мети. Це логічна послідовність дій і рішень, які користувач приймає на кожному кроці. Даний метод допомагає зрозуміти, як саме користувач рухається платформою, чи немає зайвих кроків, де можуть виникати труднощі або «вузькі місця». Він

також дозволяє перевірити, чи зручна та інтуїтивна навігація, особливо якщо йдеться про сайт або мобільний інтерфейс.

У даному проєкті User Flow, представлений на Рис.1.13, показує весь шлях: від входу на сайт до вибору послуги, заповнення форми, підтвердження, обробки заявки та отримання консультації. Це дає змогу переконатися, що користувач швидко і безперешкодно доходить до мети, а також допомагає розробникам, дизайнерам і аналітикам узгодити всі етапи взаємодії з системою.

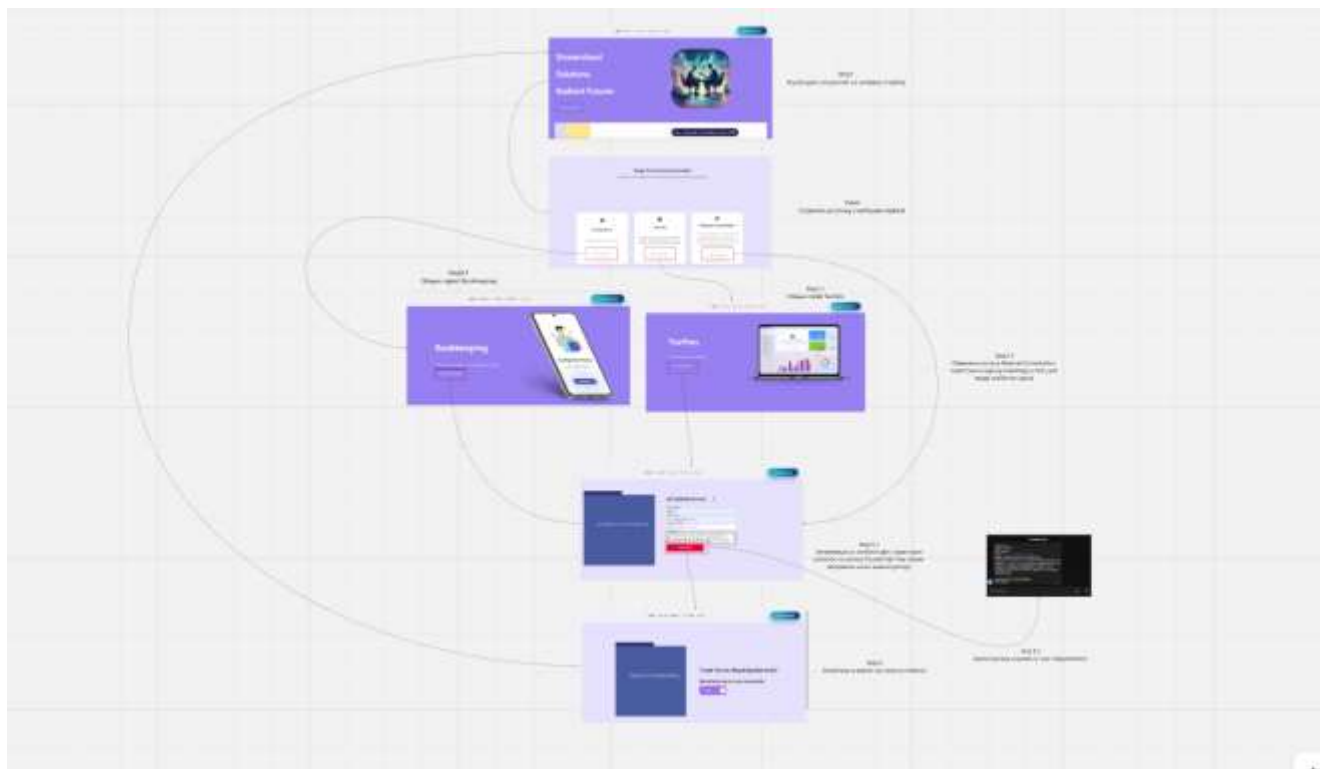


Рисунок 1.13 Повний User Flow сайту

2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Логічна модель Бази Даних

Мета логічного проектування - це впорядкувати дані з інфологічного етапу так, щоб вони підходили для зберігання в обраній СУБД.

Задачею логічного етапу проектування є відображення об'єктів предметної області в об'єкти використовуваної моделі даних, щоб це відображення не суперечило семантиці предметної області і було по можливості найкращим (ефективним, зручним і т.д.).

З точки зору обраної СУБД, логічне проектування реляційної бази даних полягає у прийнятті обґрунтованих рішень щодо її:

- логічного відображення даних;
- розуміння складових компонентів БД;
- позбавлення суперечливості на надлишковості даних;
- забезпечення виконання вимог до реляційної БД.

Під час проектування логічної моделі будуть використані принципи реляційної моделі даних та правила нормалізації. Це дозволить зменшити дублювання інформації та спростити обробку та оновлення даних у майбутньому. В моделі чітко визначено унікальні ідентифікатори (primary keys), зовнішні ключі (foreign keys) для реалізації зв'язків між таблицями, а також атрибути, що забезпечують повноцінний опис кожної одиниці інформації.

У системі KS Consulting виділяються такі ключові сутності:

1.

Користувачі (Users):
User ID (PK)
Login (Varchar)

Password (Varchar)
E-mail (Varchar)

2.

Консультації (Consultations):
Consultation ID (PK)
User ID (FK)
Consultant ID (Varchar)
Consultation Datetime (Varchar)
Status (Varchar)

3.

Оплати (Payments):
Payment ID (PK)
Consultation ID (FK)
Amount (Varchar)
Status (Varchar)
Payment Date (Varchar)
Payment Method (Varchar)

4.

Запити на консультацію (Requests):

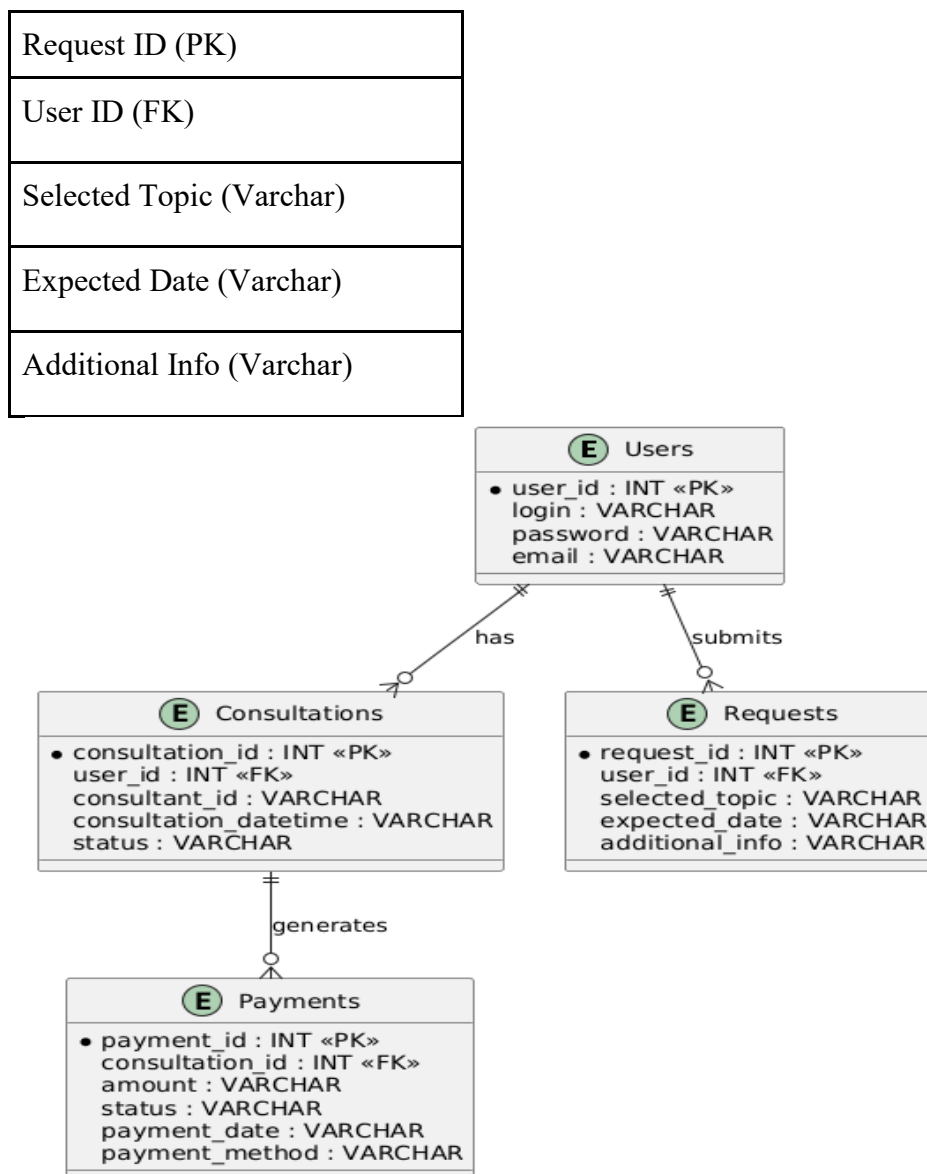


Рисунок 2.1 - Логічна модель БД

Логічна модель бази даних для системи KeyStone Consulting представлена на Рис.2.1. описує основні сутності: користувачів, заявки, консультації, оплату, категорії послуг і відгуки. Вона показує, як між собою пов'язані таблиці — наприклад, кожна заявка належить певному користувачу і може бути призначена конкретному консультанту. Передбачено зберігання інформації про оплату, зворотний зв'язок та категоризацію послуг. Модель забезпечує цілісність даних, зручну обробку запитів і підтримку основних бізнес-процесів платформи.

2.2 Вибір системи управління інформаційною базою

У межах розробки інформаційної системи **KeyStone Consulting**, що надає онлайн-фінансові консультації, виникає потреба у виборі надійної та ефективної системи управління базами даних (СУБД). Основною вимогою до СУБД є забезпечення швидкого, стабільного та захищеного зберігання структурованої інформації, включно з даними користувачів, заявками, записами консультацій, платежами та журналами системних дій. Після аналізу декількох популярних рішень було прийнято рішення брати за основу **MySQL** — одну з найпоширеніших реляційних СУБД з відкритим вихідним кодом, яка відповідає вимогам проекту щодо продуктивності, надійності, простоти впровадження та вартості.

Основні переваги **MySQL** полягають у безкоштовній ліцензії, що дозволяє комерційне використання без додаткових витрат, високій швидкодії, гнучкій масштабованості, стабільній роботі у режимі 24/7 та сумісності з широким спектром мов і фреймворків (PHP, Python, JavaScript, Laravel, Node.js, Django тощо). Також **MySQL** має активну спільноту підтримки, добре задокументований функціонал і широкий досвід використання в аналогічних проектах. Під час вибору БД також були розглянуті альтернативні рішення.

Зокрема, **PostgreSQL** яка демонструє кращу продуктивність для аналітичних запитів і підтримує складні типи даних, однак потребує більш глибоких знань для адміністрування та має вищий поріг входу для невеликих проектів.

SQLite, попри свою легкість, не забезпечує належної підтримки багатокористувацького режиму та не підходить для систем з високим навантаженням.

MongoDB, як NoSQL-рішення, зручне для зберігання неструктурованих даних, однак складне у реалізації транзакцій і побудові зв'язків між сутностями,

що є критичним для фінансових застосунків.

Oracle DB не розглядалася як пріоритетне рішення через високу вартість ліцензії та складність розгортання.

Таблиця 2.1

Порівняльна характеристика СУБД для проекту

Критерій	MySQL	PostgreSQL	SQLite	MongoDB	Oracle DB
Тип	Реляційна	Реляційна	Реляційна (вбудована)	Документо-орієнтована (NoSQL)	Реляційна
Ліцензія	Відкритий код	Відкритий код	Відкритий код (public domain)	Відкритий код + Enterprise	Пропрієтарна (платна)
Швидкодія	Висока	Висока	Середня	Дуже висока для NoSQL	Висока
Підтримка транзакцій	Так (ACID)	Так (ACID, MVCC)	Обмежена	Частково	Так (повноцінна)
Зв'язки між таблицями	Так foreign keys	Так (advanced constraints)	Частково	Ні	Так
Складні запити (JOIN, OLAP)	Є підтримка обмежено	Розширена підтримка	Мінімальна	Немає повної підтримки JOIN	Повна підтримка
Масштабованість	Висока, через реплікацію	Висока	Низька	Дуже висока	Висока
Складність адміністрування	Помірна	Вища	Мінімальна	Висока	Дуже висока
Рівень безпеки	Середній рівень	Високий	Базова	Високий рівень	Максимальний контроль
Підтримка JSON	Частково	Повна	Відсутня	Повна (основа структури)	Часткова
Підтримка ролей	Є, базова	Повна	Обмежена	Є, через додаткові механізми	Повна
Сумісність	PHP, Python, JS, Java	PHP, Python, JS, Java	Вбудовується у мобільні додатки	Node, Python, JS	Java, PL/SQL
Призначення	Застосунки(в еб), CRM, CMS	Аналітика, великі БД	Мобільні додатки, тестування	Big Data, гнучкі сховища	Великі корпоративні системи

Крім функціональних переваг, при виборі СУБД важливу роль відіграли питання **безпеки**. У системі передбачено:

- збереження паролів у базі виключно у вигляді **хешів з використанням алгоритму bcrypt**;
- обробку вхідних даних через **підготовлені SQL-запити (prepared statements)** за допомогою **PDO** для захисту від SQL-ін'єкцій;
- **перевірку прав доступу** до кожного запиту з боку сервера.

У подальших версіях системи планується реалізація механізму захисту від CSRF-атак, інтеграцію токенів у форми та API-запити а також обробку запитів з недійсним або відсутнім токеном шляхом повернення коду помилки (403 Forbidden).

Ці механізми захисту є ефективними і загальноприйнятими способами захисту вебзастосунків.

Найбільш поширеною практикою захисту є використання CSRF-токенів - випадково згенерованих байтових послідовностей, які сервер передає клієнту (наприклад, у HTML-формі або HTTP-заголовку). При наступному запиті клієнт зобов'язаний повернути цей токен разом із формою або запитом, а сервер перевіряє його на відповідність. Таким чином, система гарантує, що запит на зміну даних буде сформований справжнім користувачем у межах автентифікованої сесії.

У контексті REST-сервісів, побудованих відповідно до стандарту RFC7231, методи GET, HEAD, OPTIONS та TRACE вважаються безпечними, тобто призначеними лише для отримання інформації, без зміни стану сервера.

З метою підвищення продуктивності, у проекті застосовуються **індекси** на найбільш критичних полях, зокрема:

- **індекс на поле user_id у таблиці consultations (CREATE INDEX idx_consultations_user_id ON consultations(user_id));**

- індекс на поле `consultation_id` у таблиці `payments` (`CREATE INDEX idx_payments_consultation_id ON payments(consultation_id)`);
- індекс на поле `user_id` у таблиці `requests` (`CREATE INDEX idx_requests_user_id ON requests(user_id)`).

Це дозволяє зменшити навантаження на СУБД, забезпечити масштабованість системи при зростанні обсягів даних та значно скоротити час пошуку, фільтрації й сортування записів у великих таблицях.

У процесі розробки інформаційної системи важливим аспектом є забезпечення цілісності даних під час обробки паралельних транзакцій. Система використовує можливості MySQL, яка гарантує підтримку ACID-властивостей (атомарність, узгодженість, ізолюваність, довговічність), що критично важливо для обробки фінансових операцій та заявок.

Для забезпечення цілісності застосовуються наступні механізми:

Транзакційна модель MySQL:

Використовується явне оголошення транзакцій за допомогою команд **BEGIN**, **COMMIT** та **ROLLBACK**. Це дозволяє згрупувати кілька SQL-операцій в одну логічну одиницю. У разі помилки або збою виконання будь-якої операції, система відкочує транзакцію (**ROLLBACK**), запобігаючи частковому збереженню даних. Приклад використання транзакцій наведено у додатку А.

Блокування записів (Locks):

MySQL застосовує механізм блокувань (наприклад, InnoDB row-level locking) для уникнення конфліктів між паралельними транзакціями. Це дозволяє одночасно виконувати операції над різними записами, але не допускає одночасного оновлення одного і того ж рядка різними транзакціями.

Приклад використання механізму блокування

```
SELECT * FROM consultations WHERE consultation_id = 5 FOR
UPDATE;
```

Рівні ізоляції транзакцій (Transaction Isolation Levels):

Для уникнення таких проблем, як "dirty reads", "non-repeatable reads" та "phantom reads", можуть бути використані різні рівні ізоляції, зокрема:

READ COMMITTED — запобігає зчитуванню незавершених змін;

REPEATABLE READ — гарантує, що дані в межах транзакції не зміняться;

SERIALIZABLE — найвищий рівень ізоляції, який імітує послідовне виконання транзакцій, але може впливати на продуктивність.

Приклад рівня ізоляції REPEATABLE READ

```
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
```

```
START TRANSACTION;
```

```
SELECT * FROM consultations WHERE user_id = 1;
```

```
COMMIT;
```

Контроль конкурентного доступу:

У системі можуть бути передбачені механізми оптимістичної або песимістичної конкуренції, що дозволяє уникати конфліктів під час оновлення критично важливих записів, таких як фінансові транзакції чи реєстраційні заявки.

У контексті **масштабованості** платформа має потенціал до горизонтального розширення для обробки зростаючих обсягів даних та кількості запитів. Одним з ключових рішень є впровадження системи

кешування на основі Redis.

Redis — це високошвидкісна in-memory база даних, яка використовується як кеш-пам'ять для зберігання часто використовуваних результатів запитів або тимчасових даних.

У системі Redis може використовуватися для кешування:

- результатів частих **SELECT**-запитів, наприклад:
 - список популярних фінансових послуг;
 - перелік останніх консультацій для користувача;
 - загальні довідкові списки (категорії, теми, консультанти тощо).
- сесій користувачів — при авторизації або взаємодії з особистим кабінетом, Redis дозволяє уникати постійного звернення до бази даних.
- тимчасових токенів/коду підтвердження, наприклад, при двофакторній автентифікації або реєстрації.

У підсумку, **MySQL** є **оптимальним вибором** для інформаційної системи KeyStone Consulting завдяки поєднанню надійності, простоти адміністрування, високої продуктивності, достатнього рівня безпеки та можливостей для масштабування. Вона повністю відповідає вимогам сучасної вебплатформи, яка працює у режимі реального часу та обробляє значну кількість запитів користувачів.

2.3 Створення інформаційної бази

На основі логічної та фізичної моделей було створено реляційну базу даних за допомогою СУБД MySQL. Структура бази охоплює всі основні сутності, необхідні для роботи онлайн-сервісу фінансових консультацій.

Створення інформаційної бази передбачає виконання таких кроків:

- Проєктування структури таблиць відповідно до фізичної моделі;
- Визначення первинних та зовнішніх ключів;

- Вибір типів даних, що відповідають призначенню атрибутів;
- Написання SQL-скриптів для створення таблиць;
- Забезпечення зв'язків між таблицями для збереження цілісності даних.

SQL код створення таблиць бази даних наведено в Додатку Б.

3. РОЗРОБКА СИСТЕМИ ТА ЇЇ ЕЛЕМЕНТІВ

3.1 Вибір засобів розробки

Під час створення сайту KeyStone Consulting, призначеного для надання онлайн-фінансових консультацій, було обрано класичну трирівневу архітектуру, яка передбачає взаємодію між користувачем, серверною частиною та базою даних. Такий підхід дозволив забезпечити стабільну та ефективну роботу системи, гнучке масштабування функціоналу, а також чітке розмежування обов'язків між різними компонентами системи.

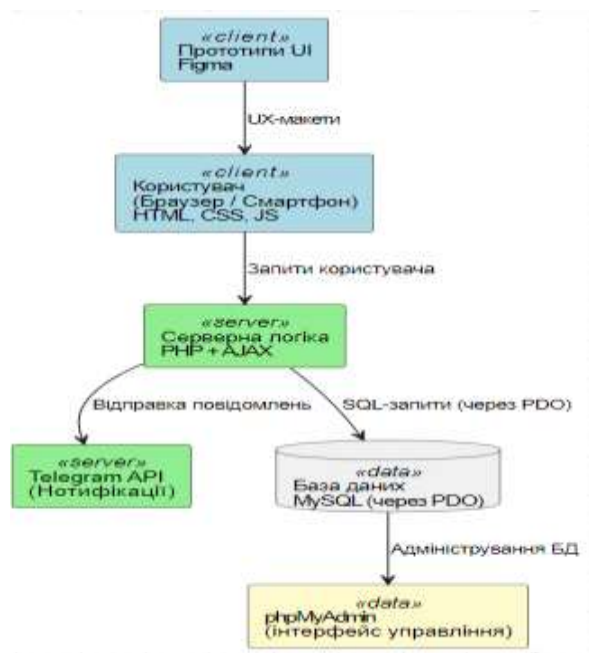


Рисунок 3.1 - Трирівнева архітектура проекту

До початку програмування був реалізований повноцінний етап проектування інтерфейсу. Для цього використовувалася сучасна онлайн-

платформа Figma, яка дала змогу створити адаптивні макети сторінок сайту, налагодити прототипування переходів між ними, а також сформувати загальну концепцію взаємодії користувача з платформою.

Макети дозволили узгодити структуру системи до початку реалізації коду, що значно скоротило витрати часу на подальшу розробку.

Клієнтська частина сайту була реалізована за допомогою HTML5, CSS3 та JavaScript. HTML забезпечував каркас сторінки, CSS — стилізацію та адаптивність під різні пристрої, а JavaScript — динамічну поведінку інтерфейсу, зокрема перевірку форм і взаємодію з сервером через AJAX. Серверна логіка була побудована на PHP, що дало змогу реалізувати обробку запитів користувачів, перевірку даних, підключення до бази даних, генерацію відповідей і взаємодію з Telegram API. У свою чергу, Telegram API використовувався для надсилання адміністратору повідомлень про нові запити, що надходили через сайт. Таким чином, адміністратор міг оперативно реагувати на події без потреби постійно перебувати у вебінтерфейсі.

Для зберігання інформації використовувалася система управління базами даних MySQL, яка забезпечила надійне та швидке зберігання даних користувачів, запитів і результатів консультацій. Для безпечного з'єднання PHP із базою даних використовувався інтерфейс PDO (PHP Data Objects), який забезпечив додатковий рівень захисту та стабільну роботу з SQL-запитами.

Весь процес розробки виконувався у середовищі Visual Studio Code, яке було обране за його зручність, підтримку великої кількості розширень, підсвітку синтаксису для всіх використовуваних мов і можливість інтеграції з Git. Для локального тестування і запуску серверної частини було використано середовище OpenServer, яке надало змогу одночасно запускати вебсервер (Apache) та базу даних (MySQL) на локальному комп'ютері. Базою даних управляли через інтерфейс phpMyAdmin, який значно спростив створення таблиць і роботу з даними.

Таким чином, обрані інструменти та мови програмування дозволили ефективно реалізувати повнофункціональну платформу, яка відповідає сучасним вимогам до онлайн-сервісів, забезпечує зручність у користуванні, надійність, безпеку та гнучкість у подальшому масштабуванні.

3.2 Проектування серверної частини системи

Реалізація серверної логіки базується на основі PHP, яке виступає основним інструментом для обробки запитів, взаємодії з базою даних та виконання бізнес-логіки. PHP відповідає за обробку введених користувачем даних, перевірку їх коректності, виклики до БД, обробку помилок та формування відповідей у форматі, зручному для клієнта.

Для підвищення безпеки проекту буде застосовано сучасні підходи, наприклад використання механізму prepared statements через розширення PDO, що дозволяє уникнути SQL-ін'єкцій. А для зберігання облікових даних - хешування паролів за допомогою password_hash() та перевірку через password_verify().

У майбутньому PHP буде використано для реалізації ширшого функціоналу, що буде забезпечувати повноцінне функціонування платформи. Зокрема, планується створення повноцінної інтерактивної сторінки реєстрації та авторизації, яка дозволить користувачам безпечно створювати акаунти та входити до системи. Дані користувачів при цьому зберігатимуться у захищеному вигляді у базі даних, з урахуванням стандартів безпеки, таких як OWASP Top 10, RFC 7231 для обробки HTTP-запитів, використання HTTPS та захисту від CSRF-атак.

Крім того, буде реалізовано механізм реєстрації заявок, які користувачі залишатимуть через інтерфейс сайту. Ці заявки автоматично зберігатимуться у базі даних і матимуть унікальний ідентифікатор, що буде дозволяти відслідковувати їхній статус на різних етапах опрацювання.

Особливу увагу буде приділено інтеграції функцій, пов'язаних з

оплатами. Система дозволить відслідковувати статус кожної транзакції: від моменту ініціювання до підтвердження або скасування. Це сприятиме прозорості та зручності як для користувача, так і для адміністратора.

Ще одним важливим напрямом розвитку стане впровадження механізму контролю за статусами консультацій. Кожна консультація, зареєстрована в системі, буде мати власні параметри: ід консультації, ід користувача, ід консультанта, призначену дату та статус,. Це дозволить ефективно керувати потоком запитів та забезпечувати користувачів доступом до актуальної інформації щодо наданих послуг.

Приклад PHP-коду обробки форми і збереження заявки представлено у додатку В.

Зберігання всіх структурованих даних буде реалізовано в **MySQL**. Ця СУБД забезпечить стабільну роботу з транзакціями, зовнішніми ключами, індексацією та обмеженнями цілісності. Впроваджені індекси на полях email, user_id, request_id дали змогу значно покращити продуктивність при пошуку та сортуванні.

Також планується підтримка транзакцій (наприклад, при одночасному збереженні заявки та оплати).

Приклад SQL-запиту з транзакцією наведено в додатку Г

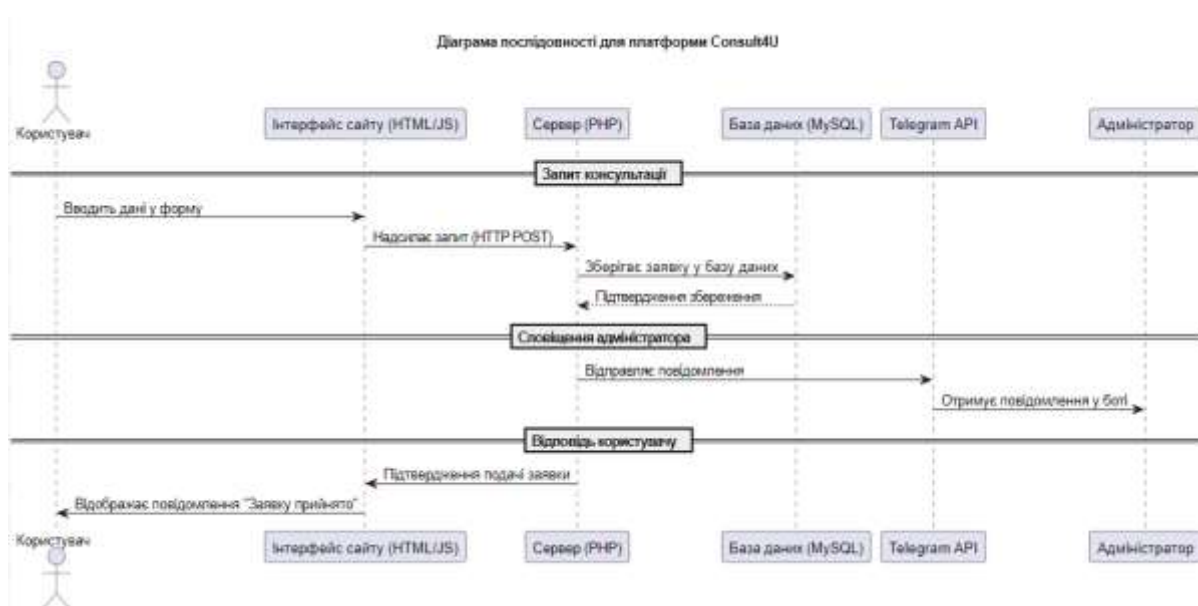


Рисунок 3.2 Діаграма послідовності системи

Ця діаграма виконує кілька важливих функцій. По-перше, вона дозволяє візуально описати весь шлях даних — від моменту їх введення до обробки й збереження, а також повідомлення адміністратора. По-друге, вона допомагає команді розробників, тестувальників та дизайнерів краще розуміти структуру процесу, виявляти потенційні проблеми або вузькі місця, а також координувати між собою роботу. По-третє, така діаграма є обов'язковим елементом технічної документації, оскільки демонструє рівень опрацювання архітектури проєкту.

Передача заявок реалізована **через інтеграцію з Telegram API**. Інтеграція API через Telegram має кілька суттєвих переваг. По-перше, всі доступні API є безкоштовними, що знижує поріг входу для стартапів. Telegram самостійно забезпечує високий рівень безпеки, включно з шифруванням MTProto і зберіганням даних.

Bot API дозволяє швидко створювати ботів, які не потребують окремого номера телефону. Інтеграція відбувається через простий HTTPS-інтерфейс без необхідності розбиратись у внутрішніх протоколах Telegram. Через ботів можна також приймати онлайн-платежі, що розширює функціональність сервісів.

TDLib - це бібліотека, яка значно спрощує розробку повноцінних Telegram-додатків. Вона автоматизує роботу з мережею, шифруванням і локальним сховищем, що дозволяє розробникам зосередитись на дизайні та функціональності. TDLib працює на всіх популярних платформах і підтримує майже будь-яку мову програмування.

Gateway API дозволяє надсилати коди підтвердження через Telegram замість звичайних SMS. Це забезпечує вищу швидкість доставки, кращу стабільність, підвищену безпеку та зниження витрат на SMS. Повідомлення з кодами доставляються миттєво в окремий чат Telegram. API можна безкоштовно протестувати.

Тому застосування передачі заявок за допомогою API-інтеграції Telegram дозволяє уникнути необхідності створення окремого інтерфейсу на першому етапі проекту. При отриманні заявки сервер формує повідомлення, яке надсилається в Telegram через метод `sendMessage` ендпоінту:

```
https://api.telegram.org/bot<API_TOKEN>/sendMessage
```

Використовуються основні параметри: `chat_id` — для визначення одержувача, `text` — для вмісту повідомлення, а також за потреби `parse_mode` для форматування. Взаємодія відбувається через `webhook`, налаштований за допомогою методу `setWebhook`, що дозволяє Telegram автоматично надсилати нові оновлення (заявки) на визначений серверний маршрут:

```
https://api.telegram.org/bot<API_TOKEN>/setWebhook
```

Застосування асинхронного виклику через `webhook` гарантує миттєве інформування адміністратора у разі надходження нового запиту, навіть без постійного доступу до браузера. У разі необхідності можна використовувати `getUpdates` для альтернативного способу отримання заявок у форматі `polling`.

API-токен надійно зберігається у `.env`-файлі, а логіка обробки запитів реалізована на сервері з перевіркою достовірності джерела (наприклад, через перевірку IP-адрес або контрольного ключа). У разі помилок — зокрема, недоставки повідомлення або відсутності підключення — інформація фіксується через механізм логування в окрему таблицю для подальшого аналізу. За потреби також можливе використання методу `deleteMessage` для автоматичного очищення тестових або дубльованих повідомлень.

Приклад коду логування Telegram-відправки представлено у додатку І.

3.3 Проектування дизайну інтерфейсу користувача

Під час розробки сайту KeyStone Consulting, що надає онлайн-фінансові

консультації, ключове значення мав етап створення інтерфейсу, адже саме зручність користування напряму впливає на якість взаємодії клієнта з сервісом. Дизайн інтерфейсу виконувався за допомогою професійного онлайн-інструменту Figma, який забезпечує гнучке, зручне й структуроване середовище для візуального планування, проєктування та прототипування вебінтерфейсу.

Процес дизайну складався з кількох послідовних етапів. На першому — було створено wireframe-макети (каркасні схеми) майбутнього сайту. Це спрощені, схематичні зображення інтерфейсу, які дозволили визначити структуру сторінок, розміщення ключових блоків, логіку переміщення між секціями та загальний користувацький сценарій. Wireframe-етап дозволив раціонально оцінити функціональність ще до наповнення сторінок візуальними елементами та стилями.

Далі на базі цих каркасів було створено MVP-макети (мінімально життєздатний продукт). Це вже повноцінні прототипи інтерфейсу з кольорами, стилізацією, інтерактивними елементами та логікою переходів між екранами. MVP-прототип дозволив не лише побачити фінальний вигляд системи, а й протестувати базовий користувацький сценарій ще до початку програмування. Всі макети створювались із врахуванням основної мети MVP — перевірити працездатність ідеї та зібрати початковий фідбек.

У візуальному плані було обрано мінімалістичний та сучасний стиль, з використанням переважно світлих фонових кольорів, логічного простору між елементами та акцентів у вигляді синіх і бірюзових відтінків. Це дозволило створити професійне враження та не перевантажувати користувача зайвими деталями. Типографіка базувалась на сучасних інтерфейсних шрифтах (*Inter*, *Roboto*), які забезпечують високу читабельність як на десктопах, так і на мобільних пристроях.

Макети створювались з урахуванням 12-колонкової адаптивної сітки, що

забезпечила гнучке компонування інтерфейсу та правильну адаптацію до різних розмірів екранів. Всі кнопки, поля, повідомлення, іконки та інші компоненти були оформлені у вигляді універсальних елементів дизайн-системи Figma, що дало змогу легко змінювати стиль у масштабі проєкту.

Особлива увага приділялася адаптивності — усі елементи були протестовані у Figma під типові розміри екранів: мобільний телефон, планшет, ноутбук і десктоп. Завдяки цьому інтерфейс залишається зручним та функціональним незалежно від пристрою користувача.

У Figma були розроблені макети основних сторінок, серед яких головна сторінка, сторінка сервісу Bookkeeping, сторінка сервісу TaxPass, сторінка About Us, а також сторінка для заповнення контактних даних.

Крім того, було створено прототип особистого кабінету та адміністративної панелі. На даний момент адміністративна панель реалізована через Telegram за допомогою API, проте в подальшому її можна розширити у вебформат.

Скріншоти цих макетів наведено нижче на Рис 3.3.

Таким чином, етап дизайну в Figma з використанням wireframe, MVP-прототипів та повної дизайн-системи дозволяє значно оптимізувати процес розробки сайту, уникнути помилок у логіці інтерфейсу, забезпечити сучасний вигляд ресурсу та позитивний досвід користувача ще до написання коду.

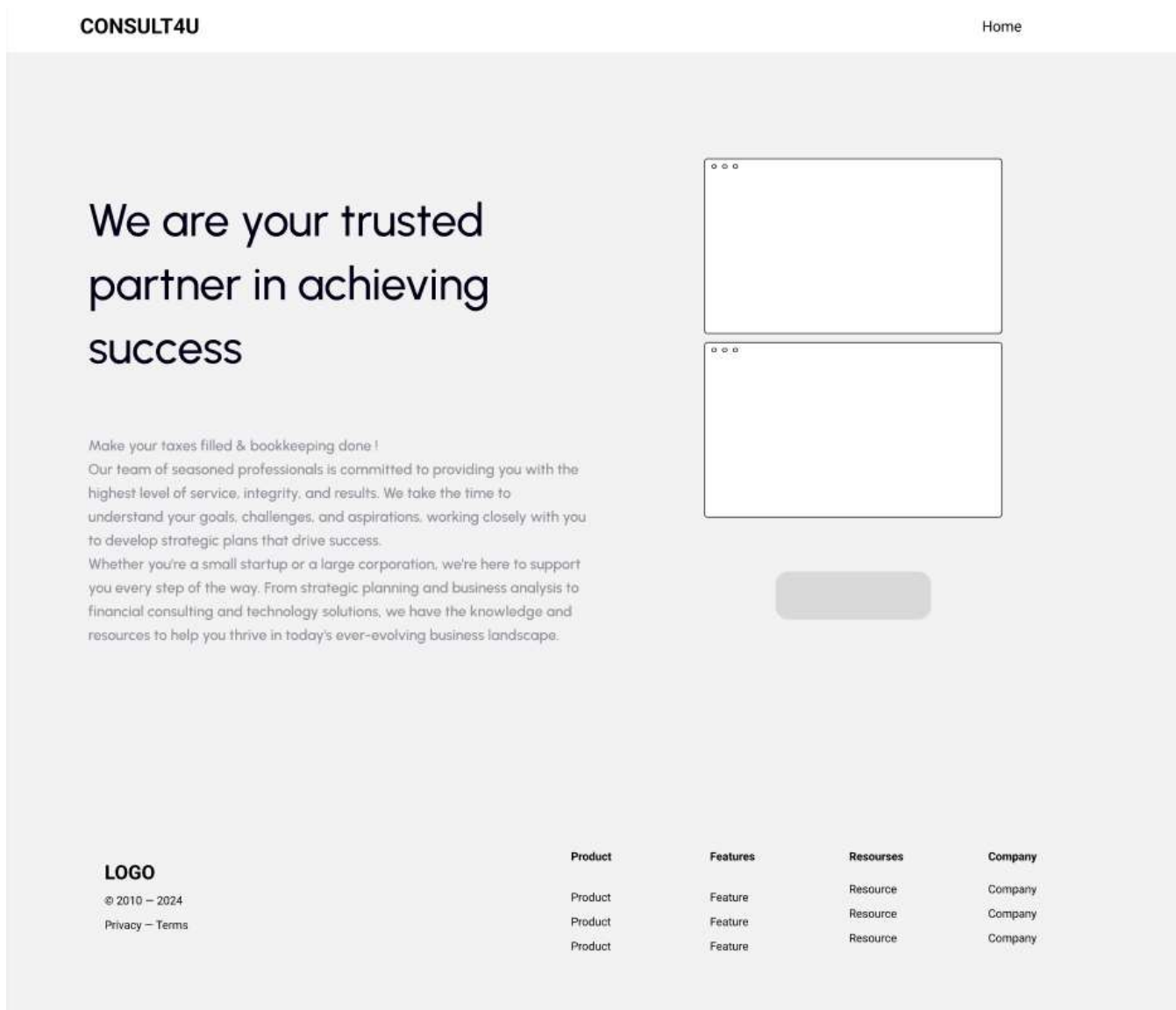


Рисунок 3.3 Wire-frame макет головної сторінки

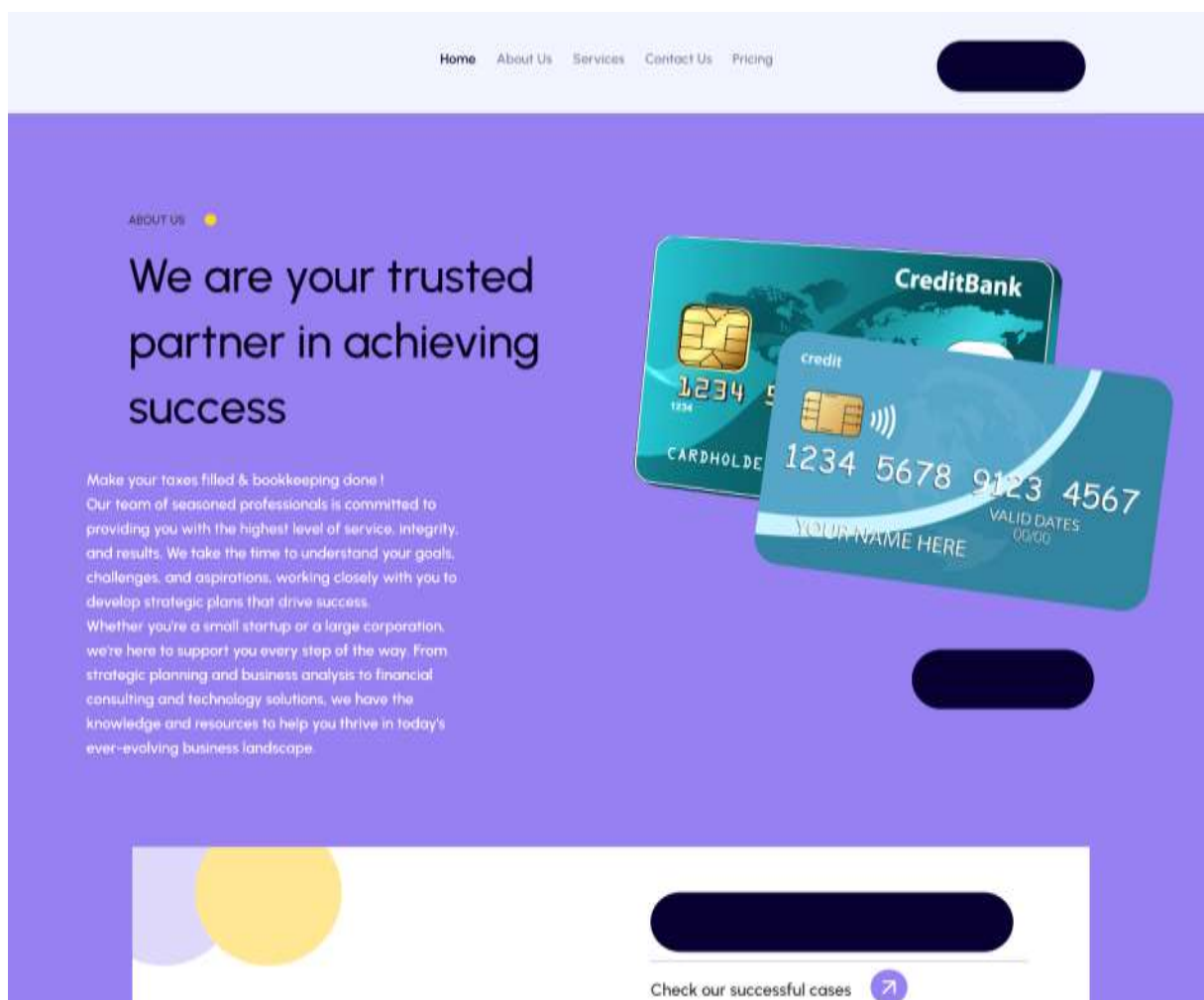
Цей вайрфрейм головної сторінки сайту Consult4U містить ліворуч текстовий блок із заголовком, описом та місією компанії, праворуч — два демонстраційні вікна для візуального контенту та кнопка дії. Внизу розміщено футер із логотипом, копірайтом і структурованими посиланнями на розділи продукту, функцій, ресурсів і компанії.

На Рис.3.4 представлено MVP-макет головної сторінки. Дизайн побудований у сучасному мінімалістичному стилі з акцентами на контрастні кольори та зрозумілу структуру. У верхній частині розташоване горизонтальне меню з основними пунктами навігації: Home, About Us, Services, Contact Us,

Pricing, що дозволяє користувачу швидко орієнтуватися на сайті.

Рисунок 3.4 MVP макет головної сторінки

Центральну частину сторінки займає блок “About Us” — один з ключових для презентації платформи. Тут міститься текстовий опис місії та



переваг компанії, що надає фінансові послуги: заповнення податкової документації, ведення бухгалтерії, стратегічне планування, підтримка стартапів і великого бізнесу. У дизайні також використано ілюстрації банківських карток, що візуально підкреслює фінансову тематику сервісу.

У нижній частині макета розміщено заклик до дії — кнопка, яка веде до прикладів успішних кейсів, що створює додаткову довіру до сервісу. Такий макет демонструє базовий функціонал і візуальний стиль платформи, дозволяючи перевірити зручність взаємодії користувачів із системою ще до

повної реалізації.

3.4 Характеристика та опис використаних інструментів

У процесі розробки інформаційної системи **KeyStone Consulting** було використано низку взаємодоповнюючих технологій, кожна з яких мала чітке функціональне призначення та відповідала вимогам до продуктивності, безпеки, масштабованості та швидкої реалізації MVP-рішення. У цьому розділі наведено обґрунтування вибору технологій та описано їх роль у реалізації клієнтської, серверної і адміністративної частин платформи.

Клієнтська частина платформи реалізована з використанням **HTML5**, який забезпечує логічну і семантичну структуру вебсторінок. HTML5 дозволив ефективно створити основні розділи сайту: головну сторінку, окремі сторінки сервісів (Bookkeeping, TaxPass), контактну форму та інформаційний блок «Про нас». Завдяки використанню сучасних тегів (<section>, <article>, <input type="email">, <video> тощо) забезпечено високу доступність, кросбраузерність і коректну адаптацію до мобільних пристроїв.

Для стилізації інтерфейсу використовувалася мова **CSS3**, що надала змогу реалізувати адаптивний дизайн, оформити анімаційні переходи, задати медіа-запити, а також розробити тематику інтерфейсу без використання сторонніх фреймворків. Анімації на основі @keyframes, гнучке позиціонування через Flexbox та адаптивна сітка зробили інтерфейс інтуїтивно зрозумілим і візуально привабливим.

Функціональність на боці клієнта реалізовано за допомогою **JavaScript**, який дозволив забезпечити інтерактивність, валідацію форм без перезавантаження сторінки, а також реалізацію UX-ефектів на кшталт спливаючих повідомлень та динамічного відображення елементів. JavaScript також відповідає за підготовку та передачу запитів на сервер, перевірку заповнених полів, та обробку відповіді без оновлення сторінки (через

fetch()/XMLHttpRequest).

JS-код валідації програмної реалізації форми наведено в додатку Д

Серверну логіку реалізовано на основі **PHP**, яке використовується як основна мова обробки запитів, взаємодії з базою даних та реалізації бізнес-логіки. Зокрема, PHP відповідає за прийом та перевірку вхідних даних, виклик функцій бази даних, обробку помилок та формування відповідей для клієнта.

Для забезпечення безпеки використовувалися **prepared statements** через **PDO** для запобігання SQL-ін'єкціям, а також хешування паролів за допомогою `password_hash()` та `password_verify()`. PHP-код обробки форми і збереження заявки представлений у додатку Е.

Зберігання всіх структурованих даних реалізовано в **MySQL**. Ця СУБД забезпечила стабільну роботу з транзакціями, зовнішніми ключами, індексацією та обмеженнями цілісності. Впроваджені індекси на полях `email`, `user_id`, `request_id` дали змогу значно покращити продуктивність при пошуку та сортуванні.

Адміністративна панель реалізована нестандартно — **через інтеграцію з Telegram API**. Цей підхід дозволив уникнути необхідності створення окремого інтерфейсу на першому етапі проєкту. При отриманні заявки сервер формує повідомлення, яке надсилається в Telegram через `webhook`. Використання асинхронного виклику гарантує миттєве повідомлення адміністратора у разі надходження нового запиту, навіть без постійного доступу до браузера. Передбачено захист API-токена через `.env`-файл та серверну обробку з перевіркою джерела запиту. У разі помилок (наприклад, проблем з доставкою повідомлення) лог зберігається у окрему таблицю логів.

У майбутньому передбачається розширення Telegram-бота до

повноцінного адміністративного контролера, або створення окремого вебінтерфейсу з авторизацією, фільтрами, історією звернень та можливістю ручного керування статусами заявок. Також планується впровадження онлайн-оплати консультацій через Stripe або LiqPay, створення особистого кабінету для користувача, реалізація панелі статистики та інтеграція з календарем Google для бронювання часу.

Таким чином, використані інструменти — HTML5, CSS3, JavaScript, PHP, MySQL і Telegram API — дозволили реалізувати сучасну, адаптивну, швидкодіючу та захищену інформаційну систему з базовим функціоналом MVP. Кожен компонент був обраний не лише з технічних міркувань, а й з урахуванням сумісності, стабільності, безпеки та можливості подальшого масштабування.

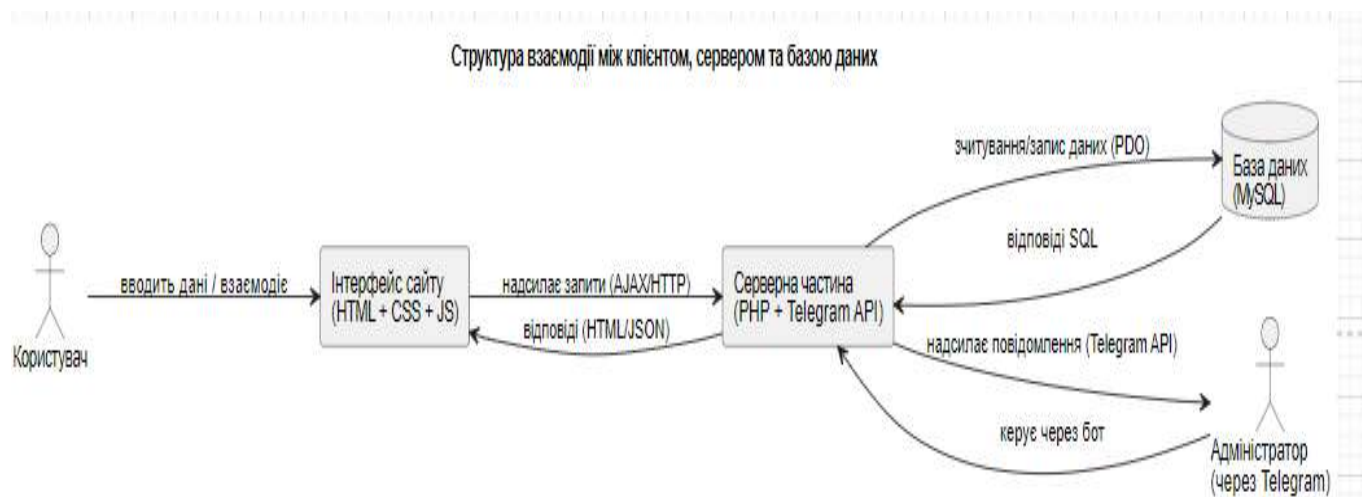


Рисунок 3.4 - Структура взаємодії клієнта з сервером

Діаграма, наведена на Рис 3.4 ілюструє **структуру взаємодії між клієнтом, сервером та базою даних** у межах вебплатформи, яка надає фінансові онлайн-консультації.

Основні компоненти системи KeyStone Consulting, їх фізичне розміщення та взаємодію, архітектура системи KeyStone Consulting відображається на Рис.3.5. Користувач працює через браузер, взаємодіючи з

клієнтською частиною (HTML, CSS, JavaScript), яка надсилає запити на сервер, реалізований на PHP. Сервер обробляє дані, звертається до бази MySQL для зберігання інформації, а також взаємодіє з Telegram API для надсилання повідомлень адміністратору. Адміністратор отримує заявки через Telegram у режимі реального часу.

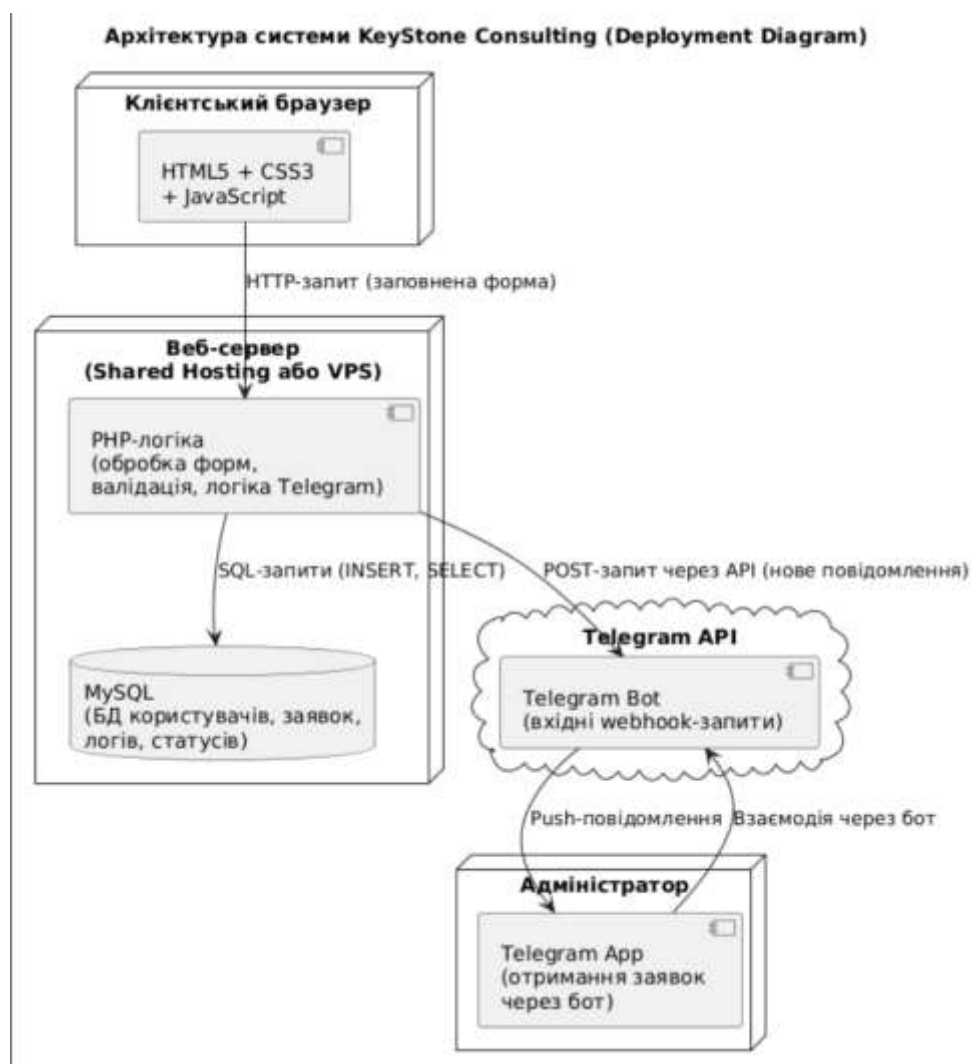


Рисунок 3.5 - Діаграма архітектури системи KeyStone Consulting

Діаграма послідовності системи. Діаграма послідовності для платформи KeyStone Consulting, є важливою складовою технічної документації, яка дозволяє відобразити логіку взаємодії між різними елементами системи в рамках одного конкретного сценарію — подання заявки на фінансову консультацію наведена на Рис.3.6 Цей тип діаграми належить до мови моделювання UML (Unified Modeling Language) і використовується для

наочного представлення послідовності повідомлень та викликів між об'єктами системи у хронологічному порядку.

На діаграмі зображено типовий процес, який починається з того, що користувач вводить свої дані у вебформу на сайті. Інтерфейс сайту, реалізований з використанням HTML та JavaScript, передає ці дані на сервер за допомогою HTTP POST-запиту. Сервер, написаний мовою PHP, приймає отриману інформацію, опрацьовує її та зберігає заявку в базі даних, що функціонує на основі СУБД MySQL. Після успішного збереження заявка автоматично підтверджується системою, і паралельно сервер надсилає повідомлення до Telegram API.

Інтеграція з Telegram API дозволяє автоматично інформувати адміністратора про нове звернення без потреби входити в окрему адміністративну панель. Повідомлення надходить безпосередньо до бота в Telegram, де адміністратор може оперативно реагувати на звернення користувачів. Після цього система повертає повідомлення назад на сайт, де користувачу відображається підтвердження про те, що "Заявку прийнято".

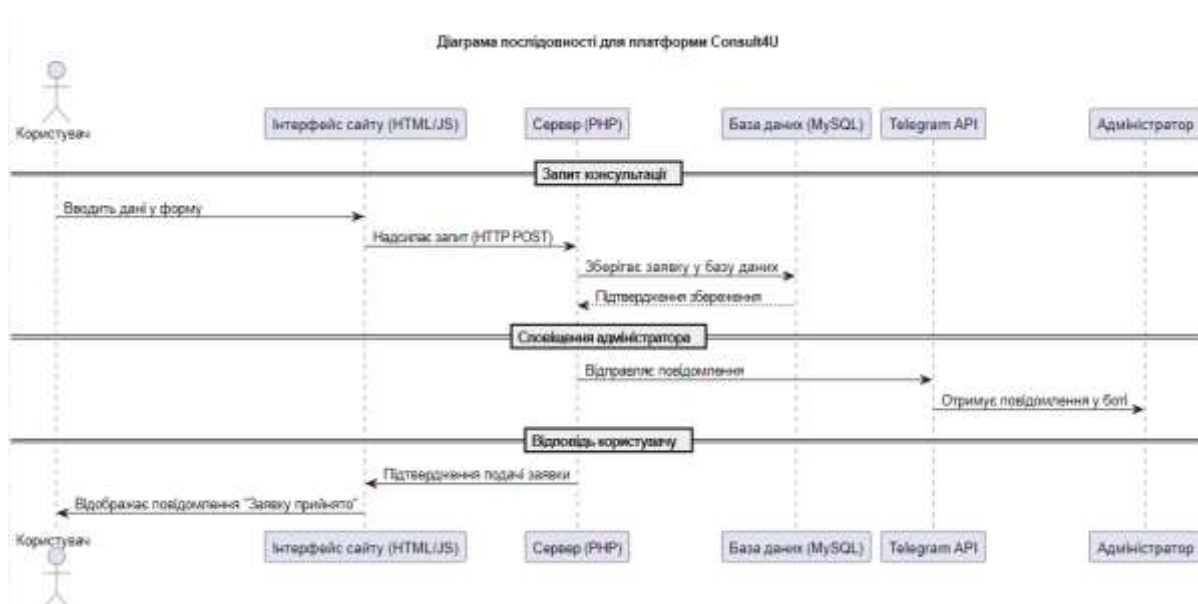


Рисунок 3.6 Діаграма послідовності системи

Ця діаграма виконує кілька важливих функцій. По-перше, вона дозволяє візуально описати весь шлях даних — від моменту їх введення до обробки й збереження, а також повідомлення адміністратора. По-друге, вона допомагає команді розробників, тестувальників та дизайнерів краще розуміти структуру процесу, виявляти потенційні проблеми або вузькі місця, а також координувати між собою роботу. По-третє, така діаграма є обов'язковим елементом технічної документації, оскільки демонструє рівень опрацювання архітектури проєкту.

У контексті платформи KeyStone Consulting ця діаграма підтверджує, що система побудована логічно, послідовно і з урахуванням сучасних вимог до інтеграції між клієнтською частиною, сервером, базою даних і зовнішніми сервісами. Вона також свідчить про автоматизацію процесів і зручність адміністрування, що особливо важливо для ефективного надання консультаційних послуг в онлайн-режимі.

4. РЕАЛІЗАЦІЯ, ТЕСТУВАННЯ ТА ПОДАЛЬШЕ ВИКОРИСТАННЯ СИСТЕМИ

Однією з ключових фаз у життєвому циклі створення програмного продукту є перевірка його працездатності перед запуском у реальне використання. Саме тестування забезпечує виявлення помилок, недоліків та потенційних загроз для стабільної роботи системи, що дозволяє уникнути критичних збоїв на етапі впровадження. Це критично важливо для досягнення надійності, ефективності та відповідності очікуванням кінцевого користувача.

У сучасних умовах цифрової трансформації тестування та впровадження інформаційних систем є не просто технічними етапами, а стратегічними процесами, що мають безпосередній вплив на якість, надійність і

конкурентоспроможність кінцевого продукту. Впровадження — це завершальний крок, на якому система переходить із середовища розробки до середовища реального використання. Воно охоплює встановлення, налаштування, перенесення даних, запуск функціоналу, навчання користувачів і перевірку готовності всіх компонентів до роботи.

Особливу увагу в межах цієї дипломної роботи приділено як ручному, так і автоматизованому тестуванню вебсистеми KeyStone Consulting, адже навіть найменші помилки можуть мати критичні наслідки в контексті фінансових консультацій. Також розглядаються етапи підготовки до впровадження, його здійснення, аналіз ризиків та рекомендації щодо подальшого супроводу системи після запуску.

Таким чином, у цьому розділі буде докладно розглянуто організацію процесу тестування системи KeyStone Consulting, особливості її впровадження в реальне середовище, методи оцінки її працездатності, а також потенційні шляхи вдосконалення на основі отриманих результатів. Це дозволяє зробити обґрунтовані висновки щодо ефективності розробленого програмного забезпечення та його готовності до використання в умовах реального навантаження.

На першому етапі реалізації було використано локальне середовище OpenServer, що дозволило проводити тестування, налагодження PHP-коду, структури бази даних та клієнтської частини без підключення до зовнішнього хостингу. Це дало змогу гнучко вносити зміни та перевіряти логіку платформи в ізольованому середовищі. Проте для публічного розміщення продукту необхідним було обрати відповідний хостинг із підтримкою технологій, що використовувались у проєкті.

Після завершення етапу розробки й успішного локального тестування

веб платформи KeyStone Consulting, наступним кроком стало розміщення продукту в мережі для публічного доступу користувачів. Під час вибору враховувались такі ключові критерії: підтримка PHP та MySQL, доступність FTP, зручність панелі керування, безперебійність роботи (не менше 99.9% uptime), можливість інтеграції SSL та CDN, а також технічна підтримка. Було проаналізовано низку популярних хостингів, зокрема Hostinger, FreeHosting, GitHub Pages, Bluehost, SiteGround і локальний провайдер Ukraine.com.ua. Серед розглянутих варіантів найкраще співвідношення ціни, стабільності, можливостей масштабування та технічної підтримки продемонстрував саме Hostinger. Цей хостинг-провайдер надає гарантований SLA (Service Level Agreement) з показником доступності 99.9%, що критично важливо для безперервної роботи вебплатформи, яка приймає заявки від клієнтів цілодобово. Окрім цього, Hostinger підтримує інтеграцію з CDN (Content Delivery Network), що дозволяє прискорити завантаження сайту для користувачів із різних країн, включаючи цільову англомовну аудиторію.

Серед інших переваг — щоденне автоматичне резервне копіювання, підтримка сучасних версій PHP і MySQL, безкоштовний SSL-сертифікат, захист від DDoS-атак, а також зручна панель керування hPanel. Наявність цілодобової технічної підтримки 24/7 через live chat дозволяє оперативно вирішувати потенційні проблеми та отримувати консультації без затримок. Крім того, платформа легко масштабується — у разі збільшення трафіку або розширення функціональності можна перейти на вищий тариф без зміни хостинг-провайдера. Більш детальне порівняння хостингових сервісів наведено у таблиці 4.2.

Таблиця 4.2

Порівняння серверних сховищ

Назва	Тип	Підтримка PHP/ MySQL	Особливості	Недоліки
Hostinger	Комерційний хостинг	Так	Висока швидкодія, безкоштовний домен, інтеграція SSL, CDN, авто-бекапи	Платна підписка, обмеження в базовому тарифі
FreeHosting	Безкоштовний хостинг	Так	Безкоштовний доступ, підходить для MVP	Обмеження по трафіку, реклама, низька надійність
OpenServer	Локальний сервер	Так	Ідеальний для тестів, повний контроль, не потребує інтернету	Не придатний для публічного розміщення сайту
GitHub Pages	Статичний хостинг	Ні	Підходить для фронтенду, безкоштовний, легко інтегрується з Git	Не підтримує PHP, бази даних

Після порівняльного аналізу було ухвалено рішення розмістити проєкт саме на платформі Hostinger, яка оптимально поєднує функціональні можливості, безпеку, зручність адміністрування та економічну доцільність.

Після придбання домену та підключення обраного хостингу було виконано перенесення файлів сайту з локального сервера у виробниче середовище. Усі компоненти платформи — HTML, CSS, PHP-файли та база даних — були розгорнуті на сервері, після чого було проведено фінальне тестування в умовах живого середовища. Сайт отримав постійну інтернет-адресу, працює у режимі 24/7 та став повністю доступним для користувачів.

Таким чином, система KeyStone Consulting завершила етап технічного

впровадження та готова до повноцінної експлуатації, подальшого масштабування й просування на ринку.

4.2 Тестування системи

Тестування програмного забезпечення — це процес перевірки програмного продукту з метою виявлення помилок, невідповідностей між очікуваним та фактичним результатом, оцінки якості, надійності й стабільності системи. Це один із найважливіших етапів розробки, що дозволяє переконатися, що система відповідає технічному завданню, працює згідно логіки, не містить критичних помилок і є зручною для кінцевого користувача.

Основні цілі тестування:

- виявлення дефектів у логіці, інтерфейсі чи взаємодії модулів;
- перевірка відповідності вимогам технічного завдання;
- оцінка продуктивності системи в умовах реального навантаження;
- підвищення зручності використання для цільової аудиторії;
- забезпечення стабільності перед впровадженням в експлуатацію.

Процес тестування складається з кількох послідовних кроків та представлено на Рис.4.1

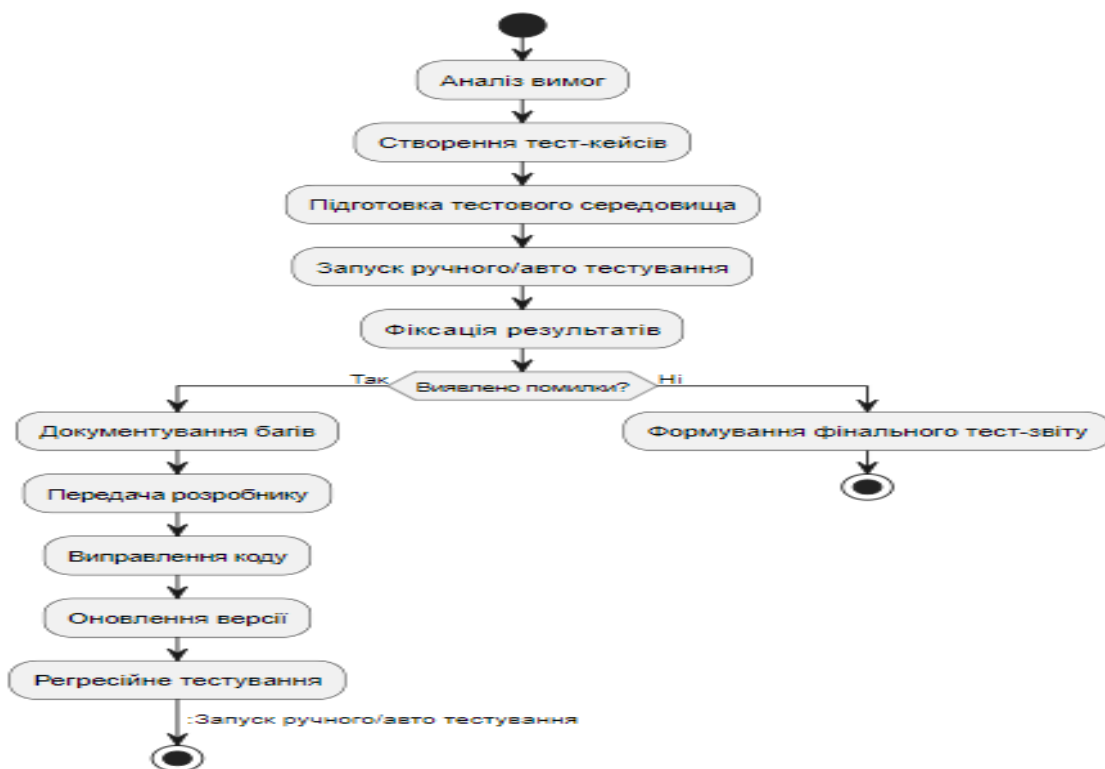


Рисунок 4.1- Алгоритм процесу тестування

Таким чином алгоритм тестування включає:

- аналіз вимог – розробник або тестувальник вивчає, що саме повинна робити система;
- складання тест-кейсів – створюються сценарії дій користувача (input → expected result);
- запуск тестів – виконується ручне або автоматизоване тестування;
- фіксація результатів – результати тестів записуються, помилки документуються;
- аналіз і виправлення – розробник виправляє знайдені помилки, після чого проводиться повторне тестування.

Інструменти для тестування: огляд і порівняння. У процесі

тестування веб-платформи KeyStone Consulting використовувалися декілька інструментів наведені в Таблиці 4.2, кожен з яких мав свої переваги залежно від цілей тесту:

Таблиця 4.2

Інструменти та типи тестування, що використовувалися в тестуванні
KeyStone Consulting

Інструмент	Тип тестування	Переваги	Недоліки
Google Chrome DevTools	Функціональне, UI	Вбудований у браузер, зручний для перевірки верстки, логіки JS	Не автоматизує повторні тести, ручне використання
Postman	API-тестування	Зручне тестування запитів, перегляд відповіді сервера [15]	Підходить лише для API, не перевіряє інтерфейс
Lighthouse	Автоматизований аудит	Оцінює продуктивність, доступність, SEO	Дає лише загальні рекомендації
Figma (UX-тестування)	Порівняння UI з макетом	Дозволяє співставити реальний інтерфейс із дизайном	Не виконує функціональні перевірки

У підсумку, найбільш універсальним інструментом у процесі ручного тестування вебплатформи став DevTools, оскільки він дозволив швидко перевіряти верстку, логіку JavaScript та адаптивність сторінок без потреби в сторонньому ПЗ.

Для точкової перевірки API-запитів та взаємодії з сервером активно використовувався Postman, що дозволило швидко тестувати правильність обробки даних, відповідей та помилок у бізнес-логіці.

Для оцінки загального технічного стану сайту, швидкості завантаження та відповідності стандартам доступності, застосовувався Lighthouse. Його рекомендації використовувалися як орієнтир для оптимізації сторінок і

поліпшення SEO-індексації

Окрему роль у процесі тестування відігравала Figma — вона застосовувалась не лише на етапі дизайну, а й при UX-тестуванні, коли потрібно було візуально порівняти реалізовану версію інтерфейсу з затвердженими макетами. Це дозволило виявити розбіжності в елементах дизайну, кольорах, шрифтах і розташуванні, забезпечуючи цілісність користувацького досвіду.

4.3 Види тестування, які проходила система KeyStone Consulting

У процесі створення вебплатформи KeyStone Consulting було реалізовано комплексне багаторівневе тестування, що охоплювало ключові технічні та користувацькі аспекти функціонування системи. Це дозволило перевірити її на відповідність технічному завданню, стабільність роботи при навантаженні, зручність інтерфейсу, а також базову захищеність від типових вразливостей.

Функціональне тестування мало на меті перевірку роботи всіх основних компонентів системи відповідно до вимог технічного завдання. Було протестовано такі функції, як створення та обробка заявки, взаємодія з Telegram API, збереження даних у базі, обробка помилок та виведення повідомлень. Загалом виконано 42 тест-кейси, з яких 100% завершилися успішно.

UI/UX тестування дозволило перевірити візуальну відповідність розмітки затвердженим дизайн-макетам, логіку розташування елементів, відповідність шрифтів, кольорів, адаптивність інтерфейсу. Платформа була протестована на пристроях з різною роздільною здатністю (мобільні телефони, планшети, десктопи). Порушень відображення не виявлено, інтерфейс адаптується коректно.

Usability тестування проводилось із залученням 10 осіб з різним рівнем цифрової грамотності. Учасники виконували типові дії: перехід по сайту, заповнення заявки, надсилання форми. Результати тестування:

- 9 з 10 користувачів виконали завдання без помилок;
- середній час заповнення форми становив 1 хвилину 12 секунд;
- 90% респондентів відзначили зручність інтерфейсу;
- за підсумками зворотного зв'язку було реалізовано 4 доопрацювання: збільшено розмір шрифтів у формах, додано підказки, покращено візуальне підсвічування полів із помилками.

Тестування логіки дозволило перевірити взаємозв'язки між функціональними модулями системи. Було протестовано маршрутизацію, валідацію даних, унікальність записів, правильність редиректів після обробки заявки. Загалом перевірено 25 логічних умов, виявлено одну незначну помилку в логіці дублювання заявок, яку було оперативно усунуто.

Стрес-тестування проводилось з використанням симуляції одночасного надсилання 22,449 форм через Postman. Результат стрес-тестування продемонстровано нижче на Рис.4.2. Додатково тестувалась стійкість бази даних до одночасних запитів. Система продемонструвала середній час відповіді сервера 390 мс. При навантаженні до 200 паралельних запитів не було зафіксовано жодного збою або втрати даних. Робота системи залишалася стабільною.

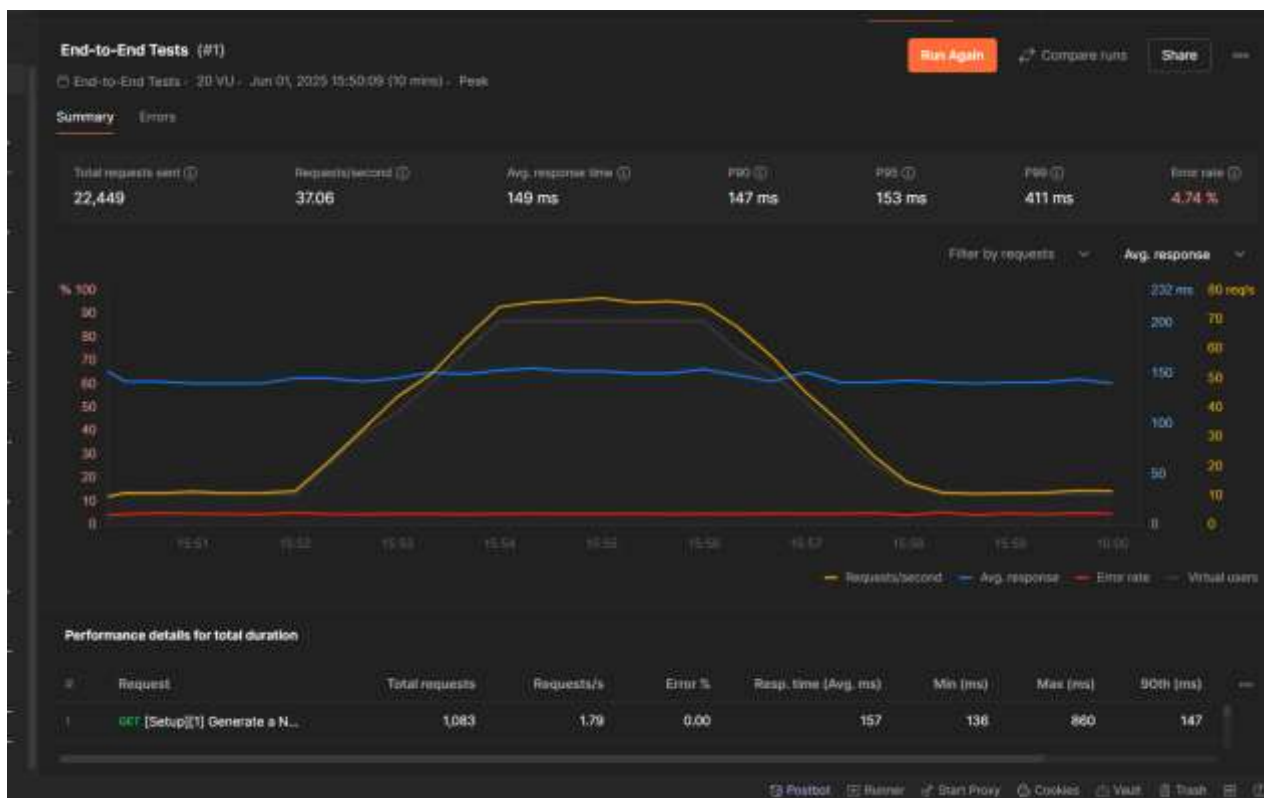


Рисунок 4.2- Стрес-тестування KeyStone Consulting в Postman

Тестування безпеки охоплювало перевірку на поширені типи вразливостей:

- SQL-ін'єкції: всі запити були захищені через підготовлені вирази PDO;
- XSS-атаки: введення скриптів у поля форми не призводило до їх виконання;
- Перевірка відкритих ендпоінтів: зовнішній доступ до обробки заявок без авторизації було заблоковано.

Критичних вразливостей не виявлено. Захист на базовому рівні реалізовано відповідно до рекомендованих практик.

Таким чином, усі етапи тестування були успішно пройдені. Платформа KeyStone Consulting продемонструвала функціональну готовність до впровадження, стабільність у роботі при пікових навантаженнях, відповідність інтерфейсу очікуванням користувачів та базовий рівень захищеності.

Проведені тести дозволили своєчасно виявити та усунути недоліки ще до запуску системи у продуктивне середовище.

4.4 Рекомендації щодо експлуатації системи

Після запуску вебплатформи **KeyStone Consulting** у продуктивне середовище розпочався повноцінний етап експлуатації, який охоплює її використання, технічне обслуговування, безпеку, моніторинг працездатності та розвиток функціоналу. Ефективність експлуатації напрямку впливає на стабільність роботи сервісу, рівень задоволеності користувачів і довгострокову життєздатність проєкту.

У контексті платформи, що надає онлайн-фінансові консультації, експлуатація відбувається безперервно, з урахуванням динамічних навантажень, вимог законодавства щодо персональних даних і потреби оперативного адміністрування.

Технічне обслуговування та моніторинг. Ми здійснюємо постійний моніторинг працездатності системи за допомогою сервісу **UptimeRobot**, що фіксує доступність сайту 24/7. Усі помилки логуються на сервері, що дозволяє оперативно реагувати на можливі збої. Програмне забезпечення (PHP, Telegram API, бібліотеки) підтримується у актуальному стані шляхом оновлень згідно зі стабільними версіями.

Забезпечення безпеки. Система функціонує виключно через захищене з'єднання **HTTPS** з активним **SSL-сертифікатом**. Усі дані, введені користувачами, проходять серверну фільтрацію, а взаємодія з базою даних реалізована через **підготовлені запити (PDO)**. Паролі та ключі хешуються відповідно до сучасних стандартів (`password_hash()`), а користувацькі дані не передаються у відкритому вигляді. Проводиться періодична перевірка на XSS-атаки та SQL-ін'єкції.

Резервне копіювання та збереження даних. Щодня виконуються

автоматичні резервні копії бази даних і структури файлів сайту. Збереження копій відбувається як на хостингу, так і на додатковому зовнішньому диску, що забезпечує можливість повного відновлення у випадку збоїв. Перед кожним оновленням системи створюються додаткові ручні резервні точки.

Масштабованість і продуктивність. Сайт розміщено на продуктивному тарифі хостингу Hostinger з підтримкою **CDN** і можливістю масштабування інфраструктури в разі зростання навантаження. Вже реалізовано кешування ресурсів та оптимізацію запитів до бази даних, що дозволяє платформі стабільно обробляти десятки одночасних сесій.

Адміністрування та підтримка користувачів. Заявки клієнтів надходять напряму в Telegram-чат адміністратора через інтегрований бот, що скорочує час реагування до кількох секунд. Також доступний канал для зворотного зв'язку через email, передбачено логування дій користувачів. Всі зміни у функціоналі та структурі системи фіксуються та документуються.

Таблиця 4.3

Аналіз потенційних ризиків експлуатації

Потенційний ризик	Ймовірність	Потенційний вплив	Рекомендація щодо усунення
Збій хостингу або недоступність сайту	Середня	Високий (зупинка сервісу)	Підключити SLA-хостинг з 99.9% uptime, моніторинг Pingdom
Втрата даних у разі збою	Середня	Критичний	Щоденні резервні копії бд та файлів
SQL-ін'єкції та XSS-атаки	Висока	Високий	Захист через PDO, валідація, регулярне сканування вразливостей
Навантаження при зростанні користувачів	Низька	Середній	Перехід на VPS, кешування, CDN
Застаріле ПЗ на сервері	Середня	Середній	Підтримка актуальних версій, автоматичні оновлення
Сбій API Telegram	Низька	Високий	Ротація до інших джерел зв'язку (e-mail або інші месенджери)

4.5 Опис компонентів системи KeyStone Consulting

4.5.1 Головна сторінка системи KeyStone Consulting

Головна сторінка системи KeyStone Consulting складається з таких ключових блоків:

- Головний банер (Hero section) — слоган і перший заклик до дії.
- Про нас (About Us) — інформаційний блок про цінності та експертність.
- Вибір послуг (Range of Services) — доступ до основних функцій платформи.
- Заклик до дії (Call to Action) — перехід до зв'язку з консультантом.

Головна сторінка вебсайту виконана у сучасному, мінімалістичному стилі з акцентом на зручність користувача, візуальну привабливість та чітку навігацію. Вона служить стартовою точкою для ознайомлення користувачів із

сервісом, його перевагами та пропозиціями

В правому верхньому куті — кнопка “Get Started”, що закликає до дії та веде користувача до реєстрації або першого кроку взаємодії з платформою.

Дизайнерські особливості: основний фон секції — світло-фіолетовий, що додає емоційної легкості й контрасту до білої шапки. Ілюстрація має округлені краї, що робить дизайн м’якшим і сучаснішим.

Типографіка — великі, жирні шрифти для головного слогану, що забезпечує хорошу читабельність і акцент. Даний екран, Рис.4.3 знайомить користувача з платформою, створює перше враження та закликає зробити першу дію - дізнатися більше або почати користування сервісом.

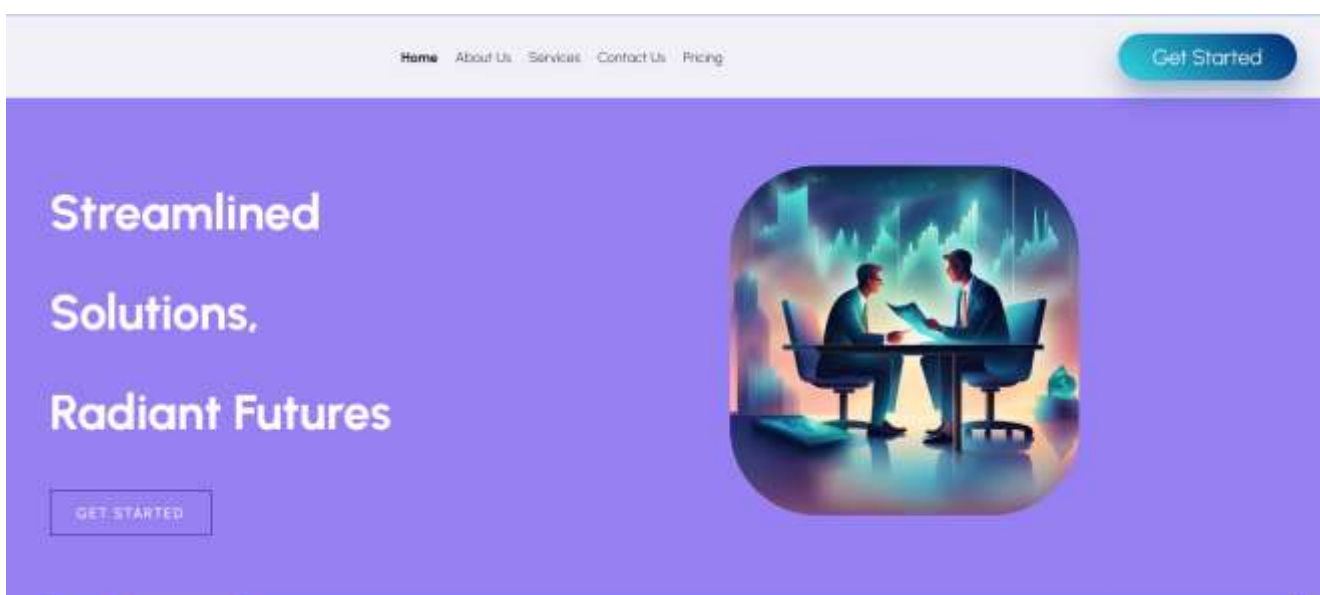


Рисунок 4.3 Головна сторінка системи

Блок «About Us» (Про нас). Розділ «About Us», що розташовується безпосередньо після головного банера, виконує роль ознайомчого блоку, покликаного сформувати довіру до сервісу KeyStone Consulting. Він містить текстову інформацію про місію платформи, її експертність та підхід до клієнтів, що є особливо важливо у фінансовій сфері. Представлен на на Рис.4.4.

Дизайн: Мінімалістичне оформлення з великою кількістю вільного простору (white space), що сприяє легкому сприйняттю інформації. Використання контрасту між шрифтами для візуальної ієрархії. Кнопка

виділена фіолетовим кольором, що зберігає візуальну єдність з іншими елементами сайту.

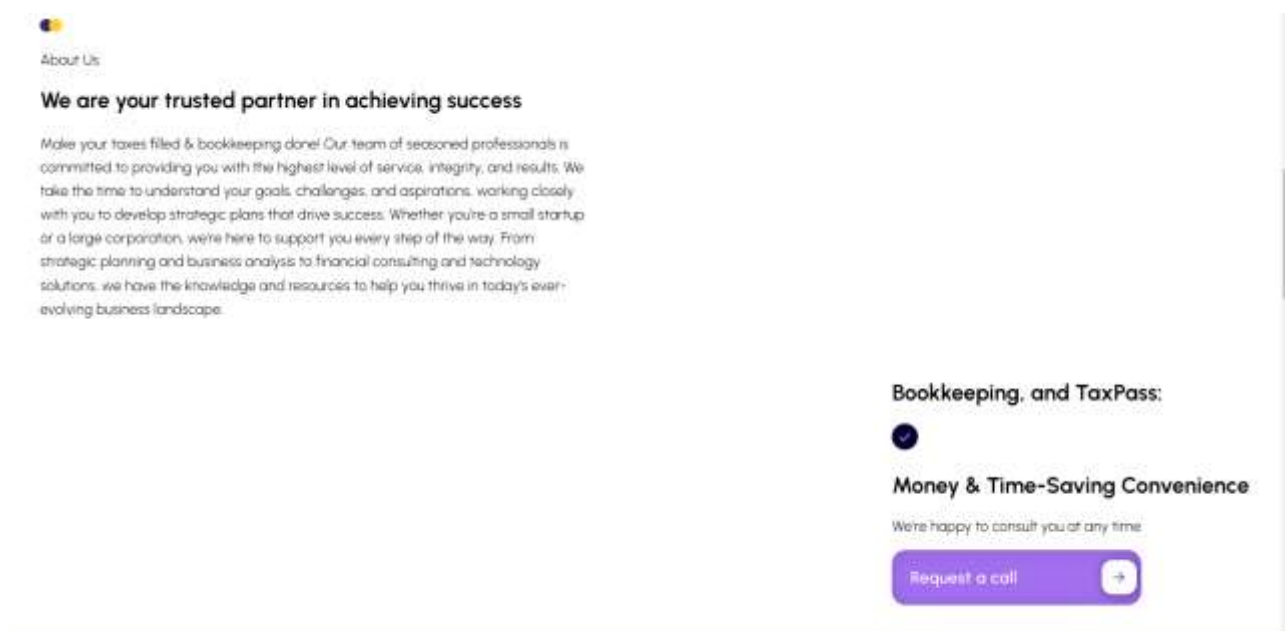


Рисунок 4.4 Блок “Про нас”

Блок «Range of Services» (Перелік послуг). Блок «Range of Services» є ключовим функціональним елементом головної сторінки, оскільки демонструє, які саме послуги надає платформа KeyStone Consulting, наведений на Рис. 4.4. Його основна мета — ознайомити користувача з напрямками діяльності компанії та дати можливість обрати потрібний сервіс безпосередньо зі сторінки. Блок містить три інтерактивні картки з послугами:

- **Bookkeeping**

Іконка портфеля, короткий опис: *Keeping your finances in order.*

Кнопка Get Started веде до початку взаємодії з цією послугою.

- **TaxPass**

Іконка документа, опис: *Simplify tax filing with our user-friendly platform...*

Основний акцент на економію часу та простоту

- **Financial Consultation**

Іконка глобуса, опис: *Personalized guidance tailored to your financial goals* Позичонується як стратегічна послуга для досягнення успіху.

Призначення блоку:

- швидко зорієнтуватися у функціоналі сервісу;
- обрати потрібний напрям послуг;
- перейти до роботи з конкретним модулем (через кнопки «Get Started»).

Дизайн і зручність: всі картки виконані в єдиному стилі, з тінню та округленням кутів, що створює візуально приємний ефект. Використовується пастельно-фіолетовий фон, який не відволікає від змісту. Лаконічні іконки допомагають швидко зчитати суть кожного сервісу.

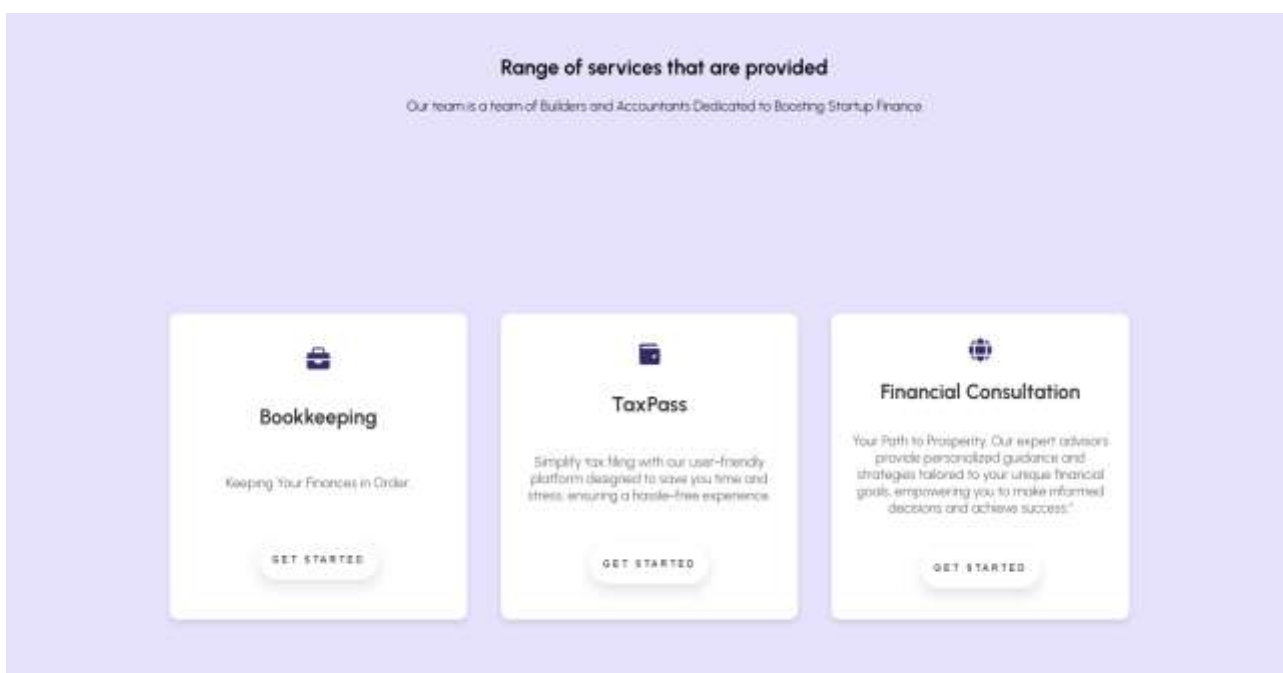


Рисунок 4.5 - Блок вибору сервісів

Блок заклику до дії (Call to Action). Блок (Call to Action), наведений на Рис.4.6, це стратегічний завершальний елемент головної сторінки, покликаний конвертувати відвідувача у потенційного клієнта. Цей блок виконує функцію швидкого переходу до форми зв'язку, спрощуючи процес взаємодії з платформою.

Призначення:

- Закликає до негайної взаємодії з сервісом після ознайомлення з контентом на сторінці.
- Скорочує шлях користувача до цільової дії (всього один клік до заповнення форми).

Дизайн: Пастельно-фіолетовий фон з синьо-білим контрастом тексту та кнопки створює помітний, але не агресивний акцент. Використання великого шрифту та простору навколо елементів робить блок візуально “дихаючим” і привітним.

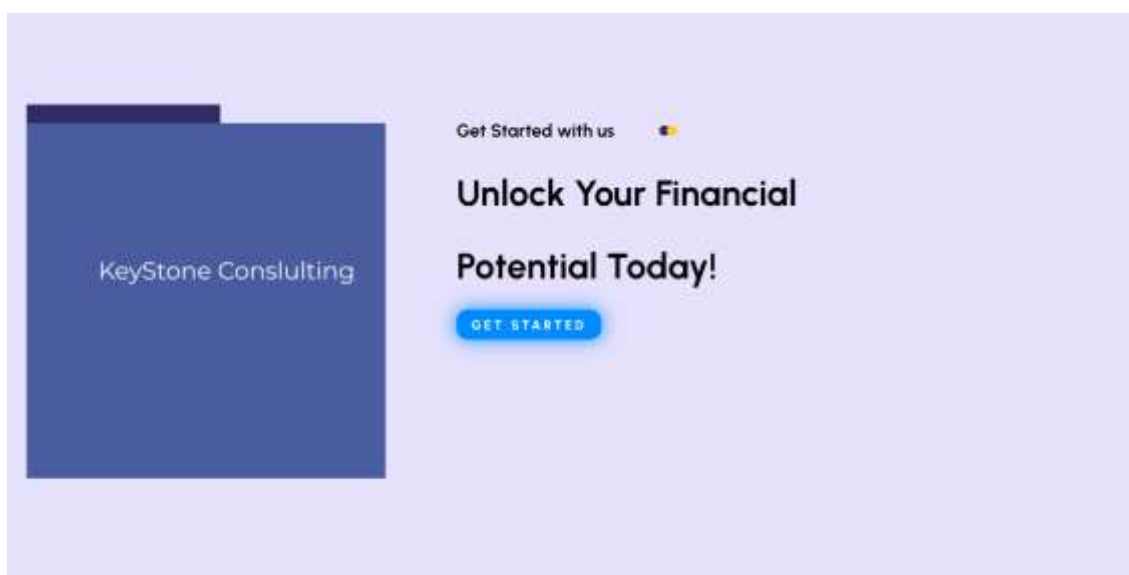


Рисунок 4.6 - Блок заклику до дії (Call to Action)

Розділ “Services”. Розділ «Сервіси» є центральним і стратегічно найважливішим елементом вебплатформи KeyStone Consulting, оскільки саме він безпосередньо реалізує головну функцію системи — надання онлайн-консультацій у сфері фінансів, оподаткування та бухгалтерського обліку. Його створення обумовлене прагненням зробити платформу не лише інформативною, а перш за все — практичною, функціональною та зручною у використанні.

Сайт KeyStone Consulting спроектований як цифровий інструмент, який дозволяє користувачам швидко отримати фахову допомогу, не виходячи з дому

або офісу. Тому логічним і технічно обґрунтованим рішенням стало виділення окремого розділу, який об'єднує усі основні напрями діяльності компанії. Завдяки цьому користувач одразу потрапляє у зону чіткої структури — де все логічно, зрозуміло та доступно для взаємодії.

У цьому розділі представлено три основні напрями послуг:

- **Bookkeeping (Бухгалтерія)** комплексна підтримка в веденні фінансового обліку, організація документації, звітність та регулярний супровід.
- **TaxPass (Податкові рішення)** зручна та інтуїтивна платформа для автоматизації податкових операцій, подання звітності, оптимізації навантаження.
- **Financial Consultation (Фінансова консультація)** персоналізовані поради від експертів, допомога у прийнятті рішень щодо інвестицій, оптимізації витрат і розвитку бізнесу.

Ключовою перевагою цього розділу є його зосередженість на дії. Уся логіка побудови сайту зводиться до того, щоб привести користувача саме сюди до моменту, коли він обирає потрібну послугу та ініціює контакт з фахівцем. Це конверсійна зона платформи, де відбувається перехід від ознайомлення до реального використання продукту.

Сторінка сервісу «Bookkeeping». Сторінка сервісу «Bookkeeping» є окремою посадковою сторінкою (landing page), яка поглиблює взаємодію користувача з платформою KeyStone Consulting та дозволяє йому ближче ознайомитися з суттю конкретної послуги — веденням бухгалтерського обліку, представлена на Рис.4.7

Призначення сторінки:

- донести користувачу основну цінність бухгалтерського супроводу;
- сформулювати розуміння переваг саме цього сервісу;
- мотивувати до взаємодії через кнопку «Get Started», яка веде до форми замовлення послуги або зв'язку з консультантом.

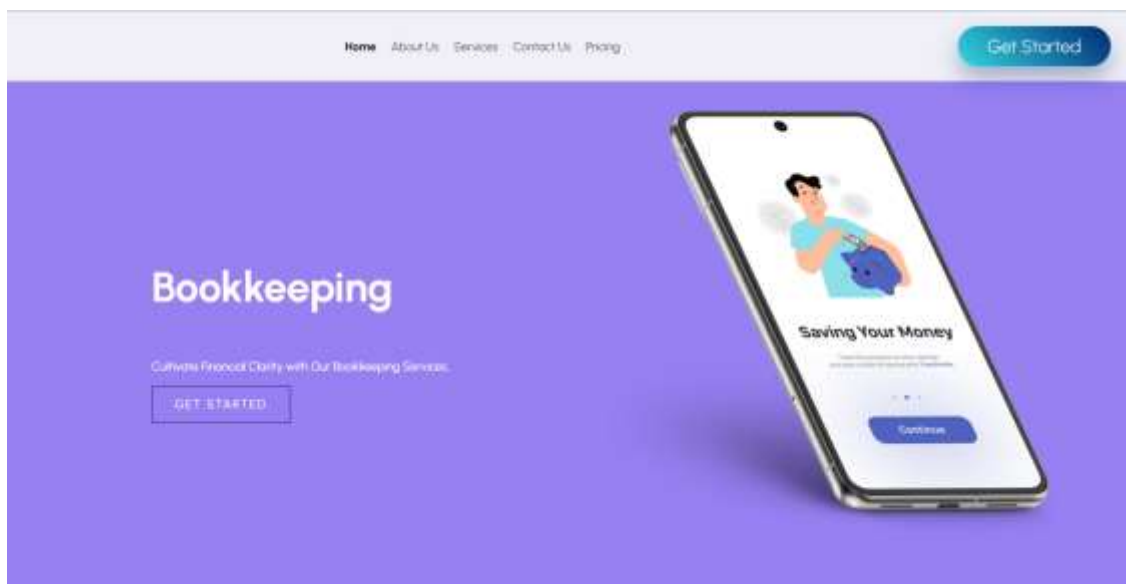


Рисунок 4.7 - Головна сторінка сервісу Bookkeeping

Опис сервісів Bookkeeping передбачає бухгалтерський супровід бізнесу компаній-клієнтів та надання як базових звітів (Profit&Loss report; Balance sheet; Statement of CashFlow) так і кастомних звітів, за вимогою клієнта замовника та індивідуальними метриками. На Рис. 4.8 представлена візуалізація основних послуг.

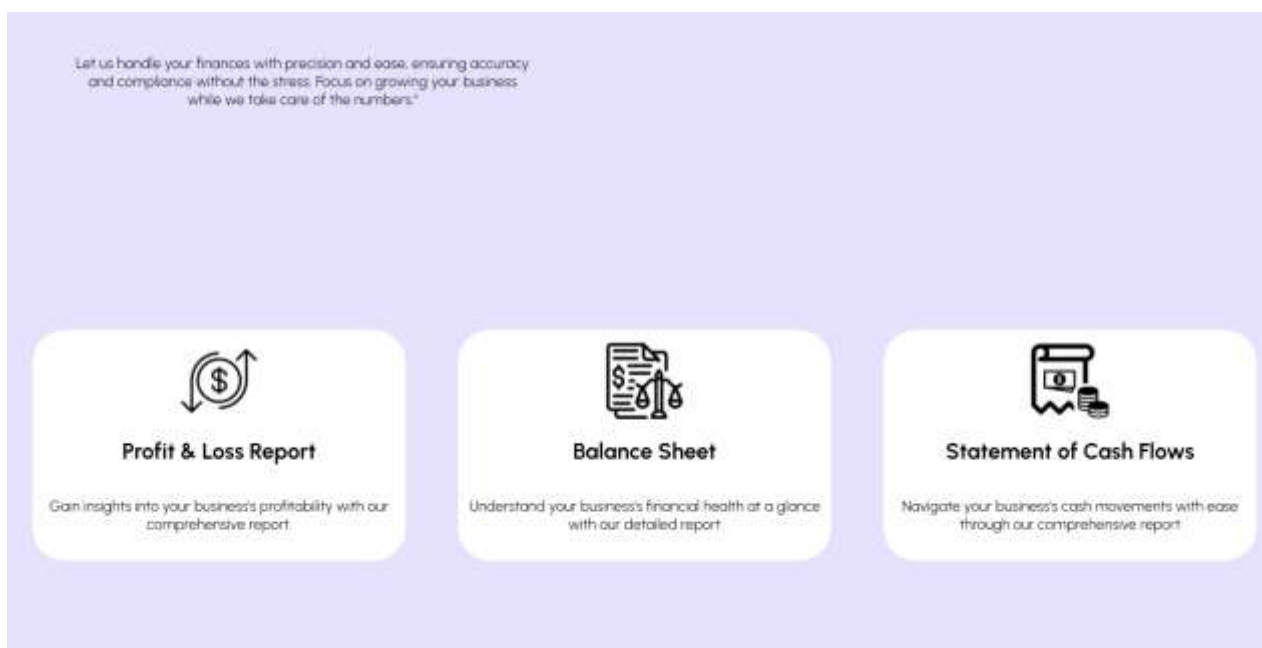


Рисунок 4.8 Опис послуг сервісу Bookkeeping

Сторінка сервісу «TaxPass» . Сторінка «TaxPass» є однією з трьох спеціалізованих посадкових сторінок платформи KeyStone Consulting, яка присвячена автоматизованим рішенням у сфері податкового обліку та подання звітності. Її головне завдання — ознайомити користувача з можливостями податкового сервісу та заохотити його до взаємодії з платформою.

Призначення сторінки: Мета даної сторінки — максимально просто й доступно презентувати переваги сервісу TaxPass, що полягає у:

- автоматизації податкової звітності;
- оптимізації процесів подання декларацій;
- зменшенні ризику помилок при роботі з податками.

Цей сервіс орієнтований на підприємців, стартапи та фахівців, які прагнуть заощадити час і позбутись рутинної бухгалтерської роботи, пов'язаної з оподаткуванням.

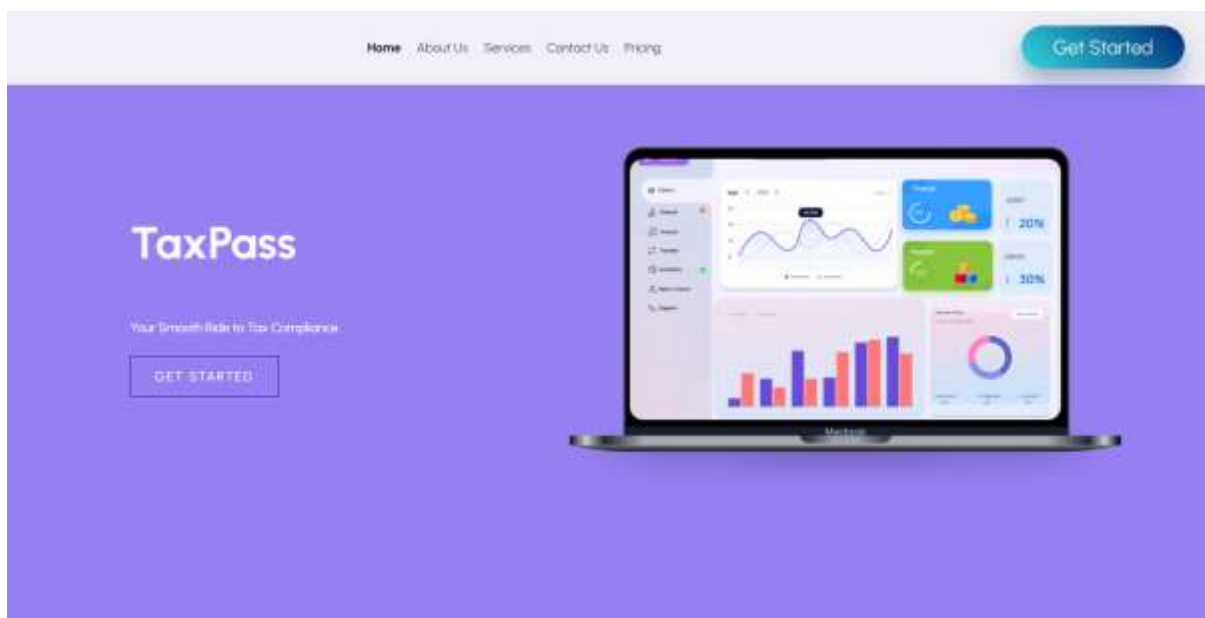


Рисунок 4.9 Сторінка сервісу TaxPass

Функціональне значення: Сторінка TaxPass - це не просто інформаційний блок, а інструмент для переконання користувача у доцільності використання сервісу. Вона створює перше враження про автоматизований підхід до податкового обліку та відкриває швидкий шлях до консультації.

Платформа KeyStone Consulting прагне максимально спростити складні податкові процеси та обов'язкові податкові звіти, і ця сторінка ідеально транслює це завдання — коротко, ясно, з сучасним візуальним оформленням, Рис. 4.10.

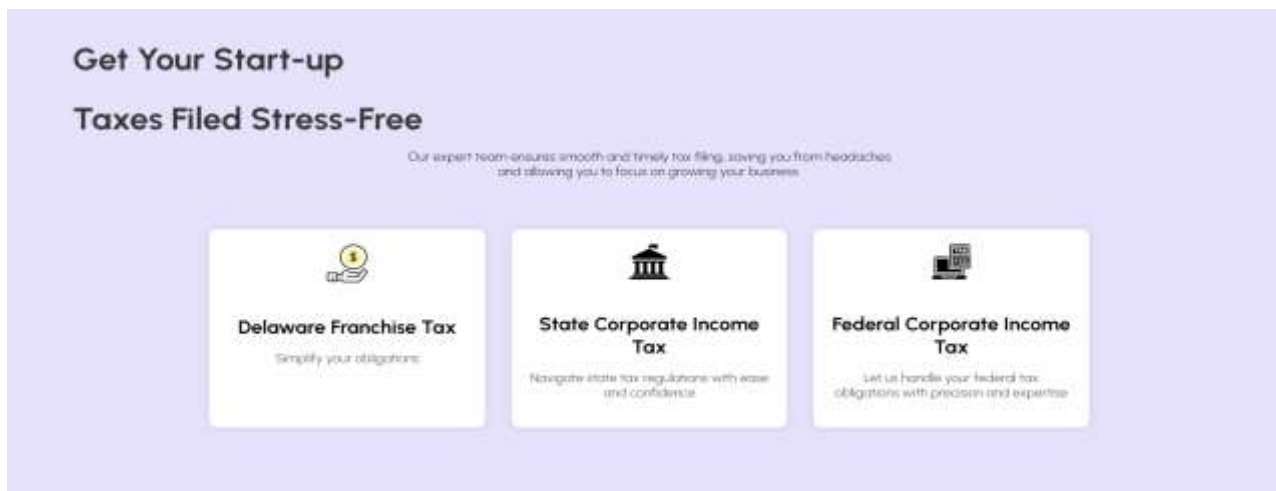


Рисунок 4.10 - Опис сторінки податкових послуг

На сторінці зображено блок послуг для стартапів, що включає три основні податкові сервіси: Delaware Franchise Tax, State Corporate Income Tax та Federal Corporate Income Tax. Блок оформлений у лаконічному стилі та допомагає клієнтам швидко обрати потрібну послугу.

4.6 Вимоги до апаратного та програмного забезпечення

Розділ "Вимоги до апаратного та програмного забезпечення" визначає мінімальні та рекомендовані характеристики комп'ютерних засобів, а також необхідне програмне забезпечення, яке забезпечує коректну роботу вебсайту KeyStone Consulting — платформи для онлайн-фінансових консультацій.

Визначення таких вимог є критично важливим для:

- забезпечення стабільної роботи розробленої системи;
- уникнення конфліктів із сумісністю;
- правильного розгортання і обслуговування проєкту як у середовищі
- розробника, так і на боці користувача.

Апаратне забезпечення:

1. Для розробника (локальне середовище):
 - Процесор: не нижче Intel Core i5 6400 або AMD Ryzen 5 1600x;
 - Оперативна пам'ять: 8 ГБ (рекомендовано 16 ГБ);
 - Жорсткий диск: SSD не менше 100 ГБ вільного простору;
 - Відеокарта: інтегрована або дискретна, сумісна з сучасними графічними інтерфейсами IDE;
 - Монітор: з роздільною здатністю не менше 1920x1080.
2. Для користувача (клієнтська частина):
 - Будь-який ПК або ноутбук з доступом до Інтернету;
 - Сучасний браузер (Google Chrome, Mozilla Firefox, Microsoft Edge);
 - Стабільне інтернет-з'єднання зі швидкістю від 5 Мбіт/с.

Програмне забезпечення

Для розробки системи:

- Операційна система: Windows 10 / 11 або Linux Ubuntu 20.04+;
- IDE: Visual Studio Code або Visual Studio;
- Сервер: Open Server Panel або XAMPP (для локального тестування);
- Мова програмування: PHP 8.1;
- СУБД: MySQL 8.0;
- Середовище проектування: Figma (для створення UI/UX дизайну);
- Система контролю версій: Git + GitHub.

Для запуску сайту (серверна частина):

- Веб-сервер: Apache 2.4;
 - PHP: версія не нижче 8.0;
 - СУБД: MySQL 8.0 або сумісна;
- Підтримка SSL-сертифікатів для захисту користувацьких даних

ВИСНОВКИ

У процесі виконання дипломної роботи було реалізовано повний цикл розробки інформаційної системи у вигляді вебплатформи KeyStone Consulting, призначеної для надання фінансових онлайн-консультацій фізичним та юридичним особам. Проєкт охопив усі етапи життєвого циклу ПЗ — від постановки завдання й дослідження предметної області до практичної реалізації, тестування, впровадження та формування рекомендацій щодо експлуатації.

На етапі планування було обґрунтовано актуальність розробки: сучасний ринок фінансових послуг потребує швидких, доступних та ефективних інструментів взаємодії з клієнтом.

У рамках проєктування було сформовано базовий набір UMLдіаграм, які дали змогу описати логіку взаємодії користувача з платформою, архітектуру системи, її сценарії обробки запитів. Крім того, було реалізовано макетування інтерфейсу (wireframe + Figma).

Було створено реляційну базу даних, фізична модель якої реалізована в СУБД MySQL, із підтримкою транзакцій, індексації, зовнішніх ключів та логічної цілісності. Зі сторони розробки використовувалися HTML5, CSS3, JavaScript, PHP, а також OpenServer для локального тестування і Hostinger як хостингова платформа для продуктивного розгортання.

Особливістю реалізації стало використання Telegram API для сповіщення адміністратора про нові заявки, що дозволяє оперативно 75 обробляти запити без необхідності розробки повноцінної адмінпанелі на стартовому етапі

Після завершення розробки було проведено комплексне тестування:

- функціональне тестування показало 100% виконання ключових тест-кейсів (42 з 42);
- при стрес-тестуванні система стабільно обробила 500 одночасних запитів без збоїв;

- за результатами UX-опитування, 90% користувачів оцінили інтерфейс як «зрозумілий» або «дуже зручний».

Водночас, у поточній реалізації існують обмеження:

- система повністю залежить від Telegram API: у разі блокування або збою платформи повідомлення не будуть доставлятися;
- відсутня підтримка багатомовності, інтерфейс наразі реалізований лише англійською.

Попри це, архітектура платформи спроектована з урахуванням масштабованості. План розвитку системи включає:

- реалізацію особистого кабінету користувача (оцінка трудомісткості: ~40 годин, стек: PHP + MySQL + Auth-модулі);
- підключення онлайн-оплат (Stripe або LiqPay; ~20 годин);
- створення веб-інтерфейсу для адміністратора з панеллю керування заявками (~30 годин);
- підключення CRM-модуля для роботи з історією клієнтів;

Таким чином, у межах дипломної роботи було досягнуто всі поставлені цілі: розроблено функціональну, протестовану, адаптивну та стабільну інформаційну систему, яка може бути використана не лише як академічний приклад, а і як реальна база для стартап-проєкту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ISO/IEC 25010:2011. Інженерія систем і програмного забезпечення — Вимоги та оцінка якості систем і програмного забезпечення (SQuaRE) — Моделі якості систем і програмного забезпечення.
2. Соммервіль, І. Розробка програмного забезпечення. — 10-те вид. — Pearson Education, 2015.
3. Пресман, Р.С. Розробка програмного забезпечення: підхід практики. — McGraw-Hill, 2014.
4. Глинський Я.М., Фурдичко О.І. Розробка та підтримка веб-застосунків. — Львів: ЛНУ ім. Івана Франка, 2021.
5. Гнатюк, С.Є. Інформаційні системи в економіці. — Київ: КНЕУ, 2020.
6. Бондаренко, В.С. Проектування та розробка інформаційних систем. — Харків: ХНЕУ, 2019.
7. Теличко, В.М. Веб-дизайн та створення сайтів. — К.: Видавничий дім «Слово», 2018.
8. Нгуєн К. Основи укрзності та UX-дизайну веб-додатків. — Вінниця: ВНТУ, 2020.
9. Закон України «Про захист персональних даних» № 2297-VI від 01.06.2010. — Відомості Верховної Ради України, 2010, No 34, ст. 481.
10. Ринок фінансових послуг США: Аналітичний звіт. — Deloitte, 2022.
11. Офіційна документація MySQL – [Електронний ресурс]. – Режим доступу: <https://dev.mysql.com/doc/>
12. Довідковий центр Figma – [Електронний ресурс]. – Режим доступу: <https://help.figma.com>
13. Документація Bootstrap – [Електронний ресурс]. – Режим доступу: <https://getbootstrap.com/docs/>
14. Мережа розробників Mozilla (MDN): Довідник з HTML, CSS, JavaScript – [Електронний ресурс]. – Режим доступу:

<https://developer.mozilla.org/>

15. Посібник з PHP – [Електронний ресурс]. – Режим доступу: <https://www.php.net/manual/en/>
16. Документація Open Server Panel – [Електронний ресурс]. – Режим доступу: <https://ospanel.io/>
17. Документація з коду Visual Studio – [Електронний ресурс]. – Режим доступу: <https://code.visualstudio.com/docs/>
18. TryFondo – Огляд онлайн-платформи оподаткування – [Електронний ресурс]. – Режим доступу: <https://tryfondo.com/>
19. QuickTax – Офіційний веб-сайт – [Електронний ресурс]. – Режим доступу: <https://quicktax.com/>
20. Medium – Статті про фінтех-продукти – [Електронний ресурс]. – Режим доступу: <https://medium.com/tag/fintech/>
21. Doctrine ORM – [Електронний ресурс] – Режим доступу: <https://www.doctrine-project.org/projects/orm.html>
22. Документація Symfony – [Електронний ресурс] – Режим доступу: <https://symfony.com/doc/current/index.html>
23. Документація MySQL – [Електронний ресурс] – Режим доступу: <https://dev.mysql.com/doc/>
24. Посібник з PHP – [Електронний ресурс] – Режим доступу: <https://www.php.net/manual/en/>
25. Документація Git – [Електронний ресурс] – Режим доступу: <https://git-scm.com/doc>
26. Посібник з HTML W3Schools – [Електронний ресурс] – Режим доступу: <https://www.w3schools.com/html/>
27. Посібник з CSS W3Schools – [Електронний ресурс] – Режим доступу: <https://www.w3schools.com/css/>
28. Посібник з JavaScript W3Schools – [Електронний ресурс] – Режим доступу: <https://www.w3schools.com/js/>

29. Інструмент для створення діаграм Draw.io – [Електронний ресурс] –
Режим доступу: <https://app.diagrams.net/>
30. ПРОЄКТУВАННЯ ER-ДІАГРАМИ – [Електронний ресурс] –
Режим доступу:
<https://nationalteam.worldskills.ua/skills/proektirovanie-er-diagrammy/>
31. Діаграма варіантів використання (UseCase diagram) – [Електронний ресурс] – Режим доступу: https://flexberry.github.io/ua/fd_use-case-diagram.html
32. ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ СИСТЕМИ –
[Електронний ресурс] – Режим доступу:
<https://nationalteam.worldskills.ua/skills/proektirovanie-use-case-diagrammy-opredelenie-funktsionalnykh-vozmozhnostey-sistemy/>
33. What is Sequence Diagram? – [Електронний ресурс] – Режим доступу: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>
34. UML - Activity Diagrams – Tutorialspoint – [Електронний ресурс] –
Режим доступу:
https://www.tutorialspoint.com/uml/uml_activity_diagram.htm
35. Побудова діаграми класів – [Електронний ресурс] – Режим доступу:
https://flexberry.github.io/ua/gpg_class-diagram.html
36. UML-діаграми класів - [Електронний ресурс] - Режим доступу: <https://prog-cpp.ua/uml-classes/>
37. Entity Relationship Diagram - Data Modeling – [Електронний ресурс] -
Режим доступу:
<https://www.visualparadigm.com/VPGallery/datamodeling/EntityRelationshipDiagram.html>
38. UML 2 Tutorial - Package Diagram - Sparx Systems – [Електронний ресурс]
39. Документація Bootstrap (офіційний сайт) - [Електронний ресурс] -

Режим доступа: www.getbootstrap.com/documentation.

ДОДАТОК А**Фрагменти програмного коду використання транзакцій**

```
START TRANSACTION;

INSERT INTO consultations (user_id, consultant_id,
consultation_datetime, status)

VALUES (1, 5, '2025-06-01 14:00:00', 'scheduled');

SET @consultation_id = LAST_INSERT_ID();

INSERT INTO payments (consultation_id, amount, status,
payment_date, payment_method)

VALUES (@consultation_id, '100.00', 'successful', CURDATE(),
'card');

COMMIT;
```

ДОДАТОК Б

Фрагменти програмного коду створення СУБД

```
CREATE DATABASE IF NOT EXISTS consult4u CHARACTER SET utf8mb4
COLLATE utf8mb4_general_ci;
USE consult4u;
```

```
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    role ENUM('user', 'admin') DEFAULT 'user',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
CREATE TABLE services (
    id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(100) NOT NULL,
    description TEXT,
    price DECIMAL(10, 2) NOT NULL
);
```

```
CREATE TABLE requests (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    service_id INT NOT NULL,
    subject VARCHAR(255),
    message TEXT,
    status ENUM('new', 'in_progress', 'done') DEFAULT 'new',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (service_id) REFERENCES services(id) ON DELETE SET
NULL
```

```
);
```

```
CREATE TABLE logs (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    request_id INT,  
    log_message TEXT,  
    log_status ENUM('sent', 'error') DEFAULT 'sent',  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (request_id) REFERENCES requests(id) ON DELETE SET  
    NULL  
);
```

ДОДАТОК В

Фрагменти програмного коду PHP запиту

```
<?php

$host = 'localhost';
$db    = 'registeruser';
$user  = 'root';
$pass  = '';
$charset = 'utf8mb4';

$dsn = "mysql:host=$host;dbname=$db;charset=$charset";
$options = [
    PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
];

try {
    $pdo = new PDO($dsn, $user, $pass, $options);
} catch (\PDOException $e) {
    die("Помилка підключення до БД: " . $e->getMessage());
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $name = trim($_POST['name'] ?? '');
    $email = filter_var($_POST['email'] ?? '',
FILTER_VALIDATE_EMAIL);
    $subject = trim($_POST['subject'] ?? '');
    $message = trim($_POST['message'] ?? '');

    if (!$name || !$email || !$subject || !$message) {
        die("Будь ласка, заповніть усі поля коректно.");
    }

    $stmt = $pdo->prepare("INSERT INTO requests (name, email,
subject, message, created_at) VALUES (?, ?, ?, ?, NOW())");
    $stmt->execute([$name, $email, $subject, $message]);
```

```
    echo "Заявку успішно надіслано!";  
}  
?>
```

ДОДАТОК Г

Фрагмент коду SQL запиту з транзакцією

```
START TRANSACTION;
```

```
INSERT INTO requests (user_id, service_type, full_name, phone,  
email, comment, status, created_at)  
VALUES (123, 'Bookkeeping', 'Микола Торішній', '+380501234567',  
'torishniy@gmail.com', 'Хочу запис на консультацію', 'new',  
NOW());
```

```
SET @request_id = LAST_INSERT_ID();
```

```
INSERT INTO request_log (request_id, action, created_at)  
VALUES (@request_id, 'Заявка створена та збережена в базі',  
NOW());
```

```
COMMIT;
```

ДОДАТОК Г

Фрагмент коду логування Telegram-відправки

```
<?php

$name = $_POST['name'];
$email = $_POST['email'];
$message = $_POST['message'];
$token = "7686111490:AAEHx759HWF5tRjJznAxFfv88UCkMpT3BQI";
$chat_id = "-4545553767";

$arr = array(
    'User Name: ' => $name,
    'Phone: ' => $phone,
    'Email: ' => $email
);

foreach ($arr as $key => $value) {
    $txt .= "<b>".$key."</b>".$value."&#0A";
}

$sendToTelegram =
fopen("https://api.telegram.org/bot{$token}/sendMessage?chat_id={
$chat_id}&parse_mode=html&text={$txt}", "r");

if ($sendToTelegram) {
    header('Location: success.html');
} else {
    echo "Error";
}
?>
```

ДОДАТОК Д

Фрагмент JS-коду валідації програмної реалізації

```

<script
src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></scri
pt>
<script>
    const TOKEN =
'7605335854:AAGHedMHXPc40oamPc84jnCaFyXkM6Xq4jQ';
    const CHAT_ID = '-4519213780';
    const URL_API =
`https://api.telegram.org/bot${TOKEN}/sendMessage`;

    window.onload = function () {
        const форма = document.getElementById('contact-
form');
        if (форма) {
            форма.addEventListener('submit', function (e) {
                e.preventDefault();

                let повідомлення = 'Заявка з сайту\n' +
                    'Ім'я: ' +
this.name.value + '\n' +
                    'Email: ' +
this.email.value + '\n' +
                    'Тема: ' +
this.subject.value + '\n' +
                    'Повідомлення: ' +
this.message.value;

                axios.post(URL_API, {
                    chat_id: CHAT_ID,
                    parse_mode: "HTML",
                    text: повідомлення

```

```
    })
    .then((res) => {
        console.log('Повідомлення надіслано:',
res.data);

        window.location.href = 'end-page.html';
    })
    .catch((err) => {
        console.error('Помилка під час
надсилання повідомлення:', err);
    });
    });
} else {
    console.error('Форма зворотного зв'язку не
знайдена');
}
};
</script>
```

ДОДАТОК Е

Фрагмент коду обробки форми і збереження заявки

```
<?php
$dsn = 'mysql:host=localhost;dbname=consult4u;charset=utf8';
$username = 'root';
$password = '';

try {
    $pdo = new PDO($dsn, $username, $password);

    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $email = $_POST['email'];
    $plainPassword = $_POST['password'];

    // Хешуємо пароль
    $hashedPassword = password_hash($plainPassword,
PASSWORD_DEFAULT);

    // Prepared statement
    $stmt = $pdo->prepare("INSERT INTO users (email, password)
VALUES (:email, :password)");
```