

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ**

**Кафедра комп'ютерних наук**

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**  
на тему: **«РОЗРОБКА СЕРВЕРНОЇ ЧАСТИНИ САЙТУ**  
**АНАЛІЗУ ЕФЕКТИВНОСТІ ПРОЕКТІВ У**  
**БУДІВНИЦТВІ»**

Виконав: студент 4 курсу, групи 6.1229  
спеціальності 122 комп'ютерні науки  
(шифр і назва спеціальності)  
освітньої програми комп'ютерні науки  
(назва освітньої програми)

М.О. Українець  
(ініціали та прізвище)

Керівник завідувачка кафедри комп'ютерних наук, доцент,  
д.т.н. Шило Г. М.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент завідувач кафедри фундаментальної та прикладної  
математики, професор, д.т.н. Гребенюк С. М.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Факультет математичний

Кафедра комп'ютерних наук

Рівень вищої освіти бакалавр

Спеціальність 122 комп'ютерні науки

(шифр і назва)

Освітня програма комп'ютерні науки

**ЗАТВЕРДЖУЮ**

Завідувач кафедри комп'ютерних наук,  
д.т.н., професор

Чопоров С.В.

(підпис)

“ 30 ” січня 2023 р.

**З А В Д А Н Н Я**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**

Українцю Максиму Олеговичу

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка серверної частини сайту аналізу ефективності проєктів  
у будівництві

керівник роботи Шило Галина Миколаївна, д.т.н., в.о.завідувача кафедри  
комп'ютерних наук

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 26 » січня 2023 року № 102-с

2. Строк подання студентом роботи 07.06.2023

3. Вихідні дані до роботи 1. Постановка задачі.

2. Методика розрахунку ефективності

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Постановка задачі

2. Показники і методи розрахунку ефективності проєктів у будівництві

3. Інформаційна модель задачі

4. Розробка серверної частини сайту

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 30.01.2023

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	03.04.2023	
2.	Збір вихідних даних.	11.04.2023	
3.	Обробка методичних та теоретичних джерел.	23.04.2023	
4.	Проектування серверної частини веб-сайту	27.04.2023	
5.	Розробка першого, другого та третього розділу.	24.05.2023	
6.	Оформлення та нормоконтроль кваліфікаційної роботи бакалавра.	01.06.2023	
7.	Захист кваліфікаційної роботи.	20.06.2023	

Студент \_\_\_\_\_  
(підпис)

М.О. Українець  
(ініціали та прізвище)

Керівник роботи \_\_\_\_\_  
(підпис)

Г.М. Шило  
(ініціали та прізвище)

### Нормоконтроль пройдено

Нормоконтролер \_\_\_\_\_  
(підпис)

О.Г. Спиця  
(ініціали та прізвище)

## РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка серверної частини сайту аналізу ефективності проектів у будівництві»: 67 с., 19 рис., 2 табл., 19 джерел, 2 додатки.

БУДІВНИЦТВО, ВЕБ-СЕРВЕР, ЕФЕКТИВНІСТЬ, КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА, ПРОГРАМНЕ ЗАБЕСПЕЧЕННЯ, ПРОЕКТНИЙ АНАЛІЗ, САЙТ, СЕРВЕР, СУБД.

Об'єкт дослідження – засоби автоматизації аналізу ефективності проектів у будівництві.

Мета роботи: розробка серверної частини сайту аналізу ефективності проектів у будівництві.

Методи дослідження: діалектичний метод пізнання; логічний та формально-логічний методи; метод аналогії, порівняння, узагальнення, систематизації та синтезу; системний і комплексний підходи; методи економічного аналізу, моделювання, алгоритмізації.

Розглянуто особливості підходів до аналізу ефективності проектів у будівництві. Проаналізовано програмні засоби інформаційно-комп'ютерних технологій фінансового планування та контролю. Розглянуто передові SaaS-рішення для створення, моніторингу і візуалізації ключових показників ефективності проектів. Виявлені невирішені проблеми в автоматизації оцінки ефективності проектів в будівництві та сформульовано завдання для розробки програмного забезпечення.

Запропоновано архітектуру клієнт-серверної системи оцінки ефективності проектів у будівництві. Обрано технології для створення серверної частини сайту для вирішення поставленого завдання. Розроблено програмний код веб-сервера для обслуговування запитів клієнтів та передачі даних між клієнтом та сервером.

## SUMMARY

Bachelor's qualifying theses "Development of the server part of the site for the projects effectiveness analyzing in the building": 67 pages, 19 figures, 2 tables, 19 references, 2 supplements.

BUILDING, WEB-SERVER, EFFECTIVENESS, CLIENT-SERVER SYSTEM, SOFTWARE, PROJECT ANALYSIS, SITE, SERVER, DBMS.

The object of the study is means of automating the analysis of the projects effectiveness in the building.

Aim of the study: development of the server part of the site for the projects effectiveness in the building analyzing.

Methods of research: dialectical of cognition; logical and formal-logical; method of analogy, comparison, generalization, systematization and synthesis; systemic and complex approaches; economic analysis, modeling, algorithmization. Peculiarities of approaches to the analysis of the projects effectiveness in construction are considered. The software tools of information and computer technologies of financial planning and control are analyzed. Advanced SaaS solutions for creating, monitoring and visualizing key performance indicators of projects are considered. Unsolved problems in the automation of the evaluation of the projects effectiveness of construction were identified and the tasks for software development were formulated.

The architecture of the client-server system for assessing the projects effectiveness in construction is proposed. Technologies were chosen for creating the server part of the site to solve the task. Developed web-server code to serve client requests and between client and server data transfer.

## ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
РЕФЕРАТ .....	4
SUMMARY .....	5
ВСТУП.....	8
1 МЕТОДИ ТА ЗАСОБИ АНАЛІЗУ ЕФЕКТИВНОСТІ ПРОЕКТІВ У БУДІВНИЦТВІ .....	10
1.1 Методологічні засади аналізу ефективності проектів .....	10
1.2 Програмні засоби аналізу ефективності проектів .....	16
1.3 Постановка завдання для розробки .....	20
1.3.1 Мета та завдання .....	20
1.3.2 Вхідні дані .....	21
1.3.3 Результати роботи програми .....	23
1.3.4 Вимоги до інтерфейсу системи .....	24
1.3.5 Вимоги до надійності системи .....	25
1.4 Висновки до розділу.....	26
2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ .....	27
2.1 Проектування архітектури .....	27
2.2 Проектування бази даних.....	30
2.3 Вибір засобів реалізації програмного забезпечення .....	34
2.3.1 Огляд серверних технологій .....	35
2.3.2 Вибір технологій для вирішення задач .....	38
2.4 Висновки до розділу.....	39
3 РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ.....	40
3.1 Розробка інтерфейсу.....	40
3.2 Розробка алгоритмів .....	44
3.3 Результати тестування системи.....	47
ВИСНОВКИ .....	52

ПЕРЕЛІК ПОСИЛАНЬ .....	53
ДОДАТОК А Скрипт генерації бази даних .....	55
ДОДАТОК Б Фрагменти програмного кода .....	63

## ВСТУП

Післявоєнна розбудова України потребує виконання великої кількості будівельних проектів по відновленню пошкодженого житла та будівництву нового замість зруйнованого. В умовах дефіциту ресурсів актуальним стає аналіз та контроль ефективності проектів будівництва на різних стадіях їх життєвого циклу за допомогою відповідних інструментів та індикаторів, варіантного проектування та варіантного оцінювання проектів за низкою показників. Проведення оціночного аналізу надає можливість виявлення резервів підвищення ефективності проектів, а також проблем, виробничих і фінансових ризиків. Але такий аналіз потребує великої обчислювальної роботи над значними масивами різномірної інформації. В сучасних умовах розвитку інформаційних технологій бізнес-аналітику можна покладати на обчислювальні машини з відповідними програмними комплексами, що актуалізує тему даної кваліфікаційної роботи.

Об'єктом дослідження для кваліфікаційної роботи обрано методи і технології автоматизації аналізу ефективності проектів у будівництві.

Предметом дослідження є теоретичні, методичні та технологічні аспекти розробки серверної частини сайту для аналізу ефективності проектів у будівництві.

Мета роботи полягає у розробці серверної частини сайту аналізу ефективності проектів у будівництві на підставі дослідження методик аналізу ефективності та сучасних технологій вирішення задач економічного моніторингу.

Для досягнення поставленої мети у роботі потрібно вирішити наступні завдання:

- проаналізувати предметну область: дослідити методи оцінки ефективності проектів у будівництві;



- проаналізувати програмні засоби для вирішення аналогічних задач та особливості їх практичного використання;
- виявити недоліки та не вирішені завдання у питанні автоматизації аналізу ефективності проектів у будівництві та сформулювати завдання для їх усунення;
- спроектувати структуру програмного забезпечення для вирішення поставленого завдання;
- розробити алгоритми обраних методів розрахунків ефективності проектів;
- зробити огляд серверних технологій та спроектувати архітектуру серверної частини сайту;
- розробити програмні коди для серверної частини програмного забезпечення;
- розробити інтерфейс;
- виконати тестування системи;
- оформити пояснювальну записку до кваліфікаційної роботи.

Для виконання кваліфікаційної роботи використано наступні методи дослідження: системний і комплексний підходи; діалектичний метод пізнання; логічний та формально-логічний методи; метод аналогії, порівняння, узагальнення, систематизації та синтезу; методи економічного аналізу, моделювання, алгоритмізації, об'єктно-орієнтованого програмування.

Теоретичною базою для виконання роботи стали наукові статті та підручники з економічної теорії, з розробки інформаційних систем та web-технологій, керівництво до мов програмування та до використання програмних платформ.

Кваліфікаційну роботу викладено на 67 сторінках проілюстровано 19 рисунками, 2 таблицями та 2 додатками.

# 1 МЕТОДИ ТА ЗАСОБИ АНАЛІЗУ ЕФЕКТИВНОСТІ ПРОЕКТІВ У БУДІВНИЦТВІ

## 1.1 Методологічні засади аналізу ефективності проектів

Проектні організації сьогодні стикаються із значними змінами умов своєї детальності на ринку, такими як зменшення рентабельності розробки проектів; поява та ускладнення комплексних інфраструктурних проектів, вимоги максимального скорочення термінів та зниження цін контрактів. Крім цього ефективність будівельних проектів у сучасному суспільстві оцінюється за критеріями екологічності, енергоефективності, соціальної ефективності.

Це актуалізує питання, пов'язані з оцінкою ефективності проектів будівництва на всіх етапах його життєвого циклу – від проектування до утилізації. Вагомий внесок у вирішення проблем управління будівельними проектами зробили як зарубіжні, так і вітчизняні дослідники (Р. Ньютон, С.М. Картавих, Ю. Ю. Сайковська, І. П. Вяльцева та ін.)

Поняття «ефективність» вельми широко трактується як в якісному, так і в кількісному плані багатою кількістю фахівців у достатньо широкому аспекті [2]. Це веде до неоднозначності розуміння цієї дефініції та зростанню кількості її показників [9]

В економічному аспекті ефективність господарської діяльності визначається відношенням результату (ефекту від діяльності) до витрат, які понесені для отримання цього ефекту. «Ефективність показує не сам результат, а те якою ціною він був досягнутий» [13].

Отже, основними визначальними факторами ефективності є результат та витрати, а для оцінки ефективності потрібні відповідно оцінки результату та оцінки витрат за їх видами. Вочевидь, в кожному виді господарської діяльності різні вимоги до таких оцінок.

Специфіка оцінки ефективності проектів у будівництві обумовлена тим, що житло, з одного боку, є об'єктом задоволення найважливіших соціальних потреб, а з іншого сприймається як інвестиційний проект. Аналогічно, і промислова забудівля є об'єктом задоволення потреб того чи іншого виробництва, а з іншого це інвестиційний проект. На даний час вже сформувалася методологічна база підготовки та прийняття до реалізації інвестиційних рішень [7, 12]. Проте ці підходи, засновані на комерційних принципах, що неспроможні повною мірою бути застосовані до проектів будівництва житла з їх соціальної спрямованістю, бо проекти будівництва повинні задовольняти не тільки (і не стільки) потреби проектних організацій та забудовників у отриманні прибутків, а в першу чергу потреби споживачів, регіону, громади у якісному, економічно ефективному в експлуатації житлі та необхідних промислових та інфраструктурних об'єктах.

Згідно програмних матеріалів організації з питань промислового розвитку ООН (UNIDO) будь-який інвестиційний проект проходить три головні фази свого розвитку:

- передінвестиційну (попередні дослідження до остаточного прийняття інвестиційного рішення);
- інвестиційну (проекування, договори, будівництво);
- експлуатаційну (експлуатація новостворених об'єктів) [6].

Ці етапи проходить кожний проект в будівництві, і на кожній стадії висуваються додаткові критерії ефективності. В табл. 1.1 наведена матриця критеріїв оцінювання ефективності проекту будівництва від різних учасників процесу на різних життєвих стадіях проекту – проектування, будівництво, експлуатація. Основними учасниками проекту є проектувальна організація (проекувальник), організація-збудувник, інвестор проекту (споживач) та суспільство в якості держави, регіону, громади, на території якої реалізується проект. Аналіз табл. 1.1 показує, що ефективність проекту по різному оцінюється різними суб'єктами процесу, отже проблема критерію оцінки ефективності проекту є наріжною.

Таблиця 1.1 – Критерії оцінювання ефективності проекту на різних стадіях різними учасниками процесу

	Проектувальник	Забудовник	Споживач	Суспільство
Проектування	Мінімальна трудомісткість проектних робіт та мінімальні терміни, отримання прибутку	Уніфікація конструкцій та технологій	Оптимальне співвідношення ціна – якість проекту, задоволення потреб	Вирішення соціальних, екологічних економічних проблем регіону
Будівництво	Мінімальні зміни проекту	Мінімальні витрати на працю, матеріали, обладнання, отримання прибутку	Мінімальні терміни, висока якість	Екологічність Мінімальні матеріальні витрати
Експлуатація	Якість, ергономічність, безпека	Задоволення та безпека споживача	Якість, ергономічність, безпека, економічність, задоволення потреб	Енергоефективність, безпека, естетичність, довготривала придатність для експлуатації, економічність

За визначенням фахівців [10] «під критерієм розуміється величина, яка визначає відповідність будь-якого процесу, явища або предмета заданим умовам». Умови визначаються при постановці завдання. Словесно або математично формулюється кінцева мета, яку потрібно досягти в результаті вирішення завдання. Отже, постановка задачі містить оцінку кінцевого

результату і є підставою для розробки критеріїв, обмежень, складу вихідних даних проекту. За допомогою критеріїв визначається вектор пошуку кращих варіантів проектних рішень, а отримані результати порівнюються з заданими цілями. Тому вибір критерію оцінки ефективності проекту є важливим етапом розробки методичного забезпечення систем автоматизованого проектування.

На даний час номенклатура і зміст критеріїв оцінки проектів будівництва не знайшли повного відображення в нормативних документах. У існуючих системах автоматизованого проектування розроблені, з тим або іншим ступенем повноти, власні методики критеріїв оцінки проектів.

Аналіз робіт, пов'язаних з автоматизацією архітектурно-будівельного проектування [3], показує, що найчастіше застосовуються наступні критерії, які можна поділити на дві групи.

Перша група включає критерії витрат:

- накладні витрати повні або часткові;
- одноразові витрати на об'єкт будівництва або його елементи;
- експлуатаційні витрати на об'єкт (повні або часткові);
- вартість транспортних та інженерних комунікацій;
- трудові витрати на зведення об'єкта;
- витрати на матеріали (сталь, цемент, залізобетон і ін.);
- цінність ділянки забудови.

Друга група включає критерії, які відображають функціональність та структуру об'єкта будівництва:

- площа або об'єм будівлі та його елементів;
- ступінь уніфікації проекту (проліт, крок, поверховість);
- умови взаємного розташування елементів (зонування, сумісність);
- організація людських потоків (протяжність, напруженість);
- організація транспортних та інженерних комунікацій (протяжність, потоковість, доступність).

На практиці здебільше оцінка проектів в будівництві проводиться за техніко-економічними показниками, які відображають кількісні витрати

ресурсів (грошей, матеріалів, труда, обладнання та часу) на реалізацію проекту та на його експлуатацію. За базу оцінки приймаються прототипові проекти, а порівняння варіантів проводиться по кошторисній вартості об'єктів, хоча, вочевидь, оцінка таких витрати не відображає дійсних витрат, вартість яких змінюється у часі. Тому більш обґрунтованим є використання в якості узагальнюючого економічного критерію приведених витрат ( $\Pi$ ), що включають одноразові витрати на зведення будівлі (кошторисна вартість  $C$ ) і експлуатаційні витрати в період функціонування об'єкта  $E$ .

Приведені витрати проекту розраховуються по формулі:

$$\Pi_i = E_i + K_i * C_i, \quad (1.1)$$

де  $E_i$  – річні витрати на експлуатації об'єкта по  $i$ -му проекту;

$K_i$  – нормативний коефіцієнт ефективності капітальних вкладень, що дорівнює 0,05 – 0,03, в залежності від терміну експлуатації (амортизації) об'єкта по  $i$ -му проекту;

$C_i$  – кошторисні витрати об'єкта по  $i$ -му проекту [8].

Оцінка проектних рішень за приведеними витратами найбільш об'єктивна, хоча і не відображає повністю якості проектних рішень. Однак її використання для оцінки складних рішень не може бути надійним інструментом для вибору кращих варіантів, оскільки інтегральна ефективність включає, крім економічної ефективності, ще й характеристику соціально-технічних властивостей об'єкта.

Для оцінки соціально-технічних властивостей об'єктів в даний час використовуються комплекси показників, які розроблені для різних типів будівель.

Крім того, оцінка ефективності проекту у будівництві ускладнюється тим, що, проекти можуть бути дуже різними і вимагати різних метрик для оцінки їх ефективності. Отже, для оцінки ефективності проекту тим чи іншим замовником потрібно сформулювати функцію мети, яка б відображала вподобання

замовника, його оцінку важливості того чи іншого показника. Для цього використовується нормалізація показників та оцінка їх рангів.

Для отримання єдиного показника ефективності за проектом всі показники за функціями мети нормалізуються до єдиної розмірності та приводяться до безрозмірних одиниць. Для цього використовується міра відхилення значень показників варіантів проектів від значення показників «еталонного» проекту. За «еталонний» приймається проект із найкращим значення показника по вибраній функції мети із бази знань.

Відносне відхилення показника, що досліджується, від «еталонного» значення показника дорівнює:

$$\Delta P_i = PE_i / PR_i, \quad (1.2)$$

де  $PR_i$  – значення показника  $i$ -тої функції в варіанті проекту будівництва;  
 $PE_i$  – «еталонне» значення показника  $i$ -ої функції.

Для отримання інтегрального показника виконується згортка показників в єдиний за допомогою рангів значущості показників.

Узагальнена функція мети має вигляд:

$$IP = \sum_{i=1}^n \Delta P_i \cdot r_i, \quad (1.3)$$

де  $r_i$  – ранг  $i$ -го показника,  $\sum_{i=1}^n r_i = 1$ .

$n$  – кількість показників ефективності проекту, обраних для оцінки.

За оптимальний варіант обирається той проект, в якому значення інтегрального показника відхилень (1.3) мінімальне.

Функція мети може будуватися за двома основними принципами:

- визначення об'ємно-планувальних параметрів при оптимізації витрат (вартості, праці, матеріалів) або при витратах, що не перевищують

певної межі, з дотриманням усіх необхідних вимог і регламентних умов;

- досягнення оптимального співвідношення між показниками якості і показниками витрат.

Перший принцип формування функції мети застосовується здебільше для проектів будівництва невиробничої сфери, наприклад, житла. оскільки якість житла на певному етапі розвитку суспільства визначена як еталон.

Для проектів виробничих будівель існує велике розмаїття технологічних процесів, типів промислових будівель, вимог до їх функціонування. Тому функції мети в таких завданнях будуються на другому критерію, а вибір проекту відбувається відповідно до заданих оціночних показників. Наприклад, функція мети за критерієм зв'язку мінімізує довжину комунікаційних ліній в промисловій будівлі.

Отже, для інтегральної оцінки ефективності проекту будівництва потрібно виконати великий обсяг розрахунків, тому доцільно використовувати для цієї роботи обчислювальну техніку та спеціальну програму.

## **1.2 Програмні засоби аналізу ефективності проектів**

Ринок програмних продуктів в Україні на даний час достатньо насичений пропозиціями продуктів для бізнес-аналітики, проектного менеджменту, бізнес-планування, в тому числі із урахуванням специфіки будівельних організацій. Для проектних організацій пропонується низка систем автоматизованого проектування класу САПР. Розглянемо найбільш популярні продукти.

Для вирішення завдань фінансового планування та контролю інвестиційних проектів успішно використовують ряд продуктів такі як MS Project, Time Lime, Prima Vera, Comrad, Project Expert, Альт-Інвест та ін.

Для розробка практичного бізнес-плану інвестиційного проекту популярним є пакет прикладних програм (ППП) "Project Expert". Це система



стратегічного інвестиційного проектування, в основу якої покладені методика оцінки інвестиційних проектів ЮНІДО та методика фінансового аналізу, яка відповідає міжнародним стандартам обліку. Програма дозволяє провести детальний аналіз інвестиційного проекту з урахуванням впливу інфляції, курсу валют, динаміки масштабів і структури витрат; враховувати вартість основних виробничих фондів; особливості отримання коштів за реалізований продукт, умови залучення капіталу, можливі зміни у системі оподаткування та ін.

ППП "Project Expert" можна використовувати для фінансової оцінки як на стадії розробки проекту, так і відстежувати його економічні показники на стадії втілення. Це можливо за рахунок швидкого перерахунку бізнес-плану та обґрунтування економічної ефективності проекту. Результати фінансового аналізу проекту подаються у вигляді таблиць, діаграм і графіків. Користувач може провести інтегральну оцінку проекту одночасно за багатьма критеріями.

Отже, Project Expert достатньо перспективний та зручний інструмент для розробки і обґрунтуванні економічної ефективності проектів і стратегічних планів. До недоліків програми можна віднести недостатню гнучкість у виборі методів розрахунків та оцінок ефективності проектів, націленість здебільш на проекти виробництва, а не будівництва.

Інший підхід використовується у ППП Waytobi [1]. Це передове SaaS-рішення для створення, ведення, складення звітів і візуалізації ключових показників ефективності (KPI), розроблене для спрощення процесу управління проектами. Waytobi KPI Аналітика може дати менеджменту бізнесу ясне і чітке уявлення про короткострокові та довгострокові тенденції, перспективи розвитку.

Waytobi KPI Інфопанелі (рис. 1.1) являють собою функціональну платформу для відображення і візуалізації кожного ключового аспекту й показника ефективності бізнесу, відділу, підприємства чи окремого проекту. Всі фінансові показники, перешкоди, стратегії, плани дій можна об'єднати в інфопанелі для подальшого регулювання процесів і візуального відображення інформації. Waytobi KPI Інфопанелі дозволяють аналізувати такі показники

ефективності, як рентабельність проектів, доходи від продажів, та також все необхідне для визначення першопричин фінансових ускладнень або будь-яких завдань іншого роду.

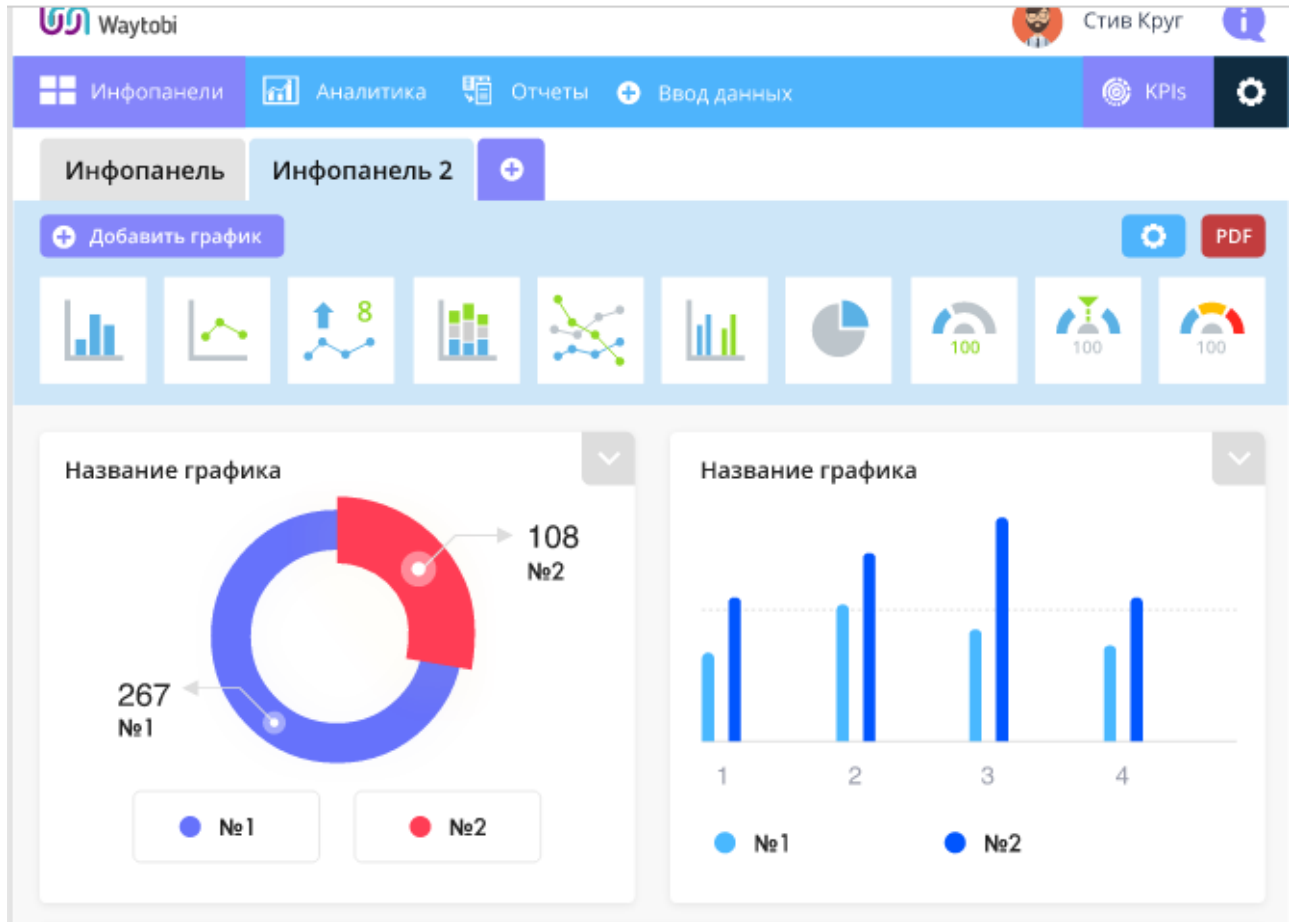


Рисунок 1.1 – Інтерфейс Waytobi KPI Інфопанелі

KPI Інфопанелі дозволяють контролювати необхідні показники ефективності команди, відділу і компанії в цілому. Можливо надати спільний доступ до KPI Інфопанелей різним співробітникам, а також будь-яким особам за межами організації. Всі дані та показники ефективності можна відобразити у вигляді різних форматів від простих лінійних графіків, таблиць до складних діаграм і лічильників. Набір ключових показників ефективності обирається провідним менеджером за допомогою інструментів ППП Waytobi. Тобто, аналіз ефективності проекту може виконуватись за тими показниками, які вважає за потрібне користувач.

BAS Будівництво. Це система класу ERP, яка підтримує організаційну стратегію інтеграції виробництва та операцій управління трудовими ресурсами, фінансового менеджменту та управління активами, орієнтована на безперервне балансування та оптимізацію ресурсів підприємства за допомогою спеціалізованого інтегрованого пакета прикладного програмного забезпечення, що підтримує загальну модель даних та процесів будівельної компанії [16].

Ключовими функціями системи є:

- планування ресурсів: забезпечення матеріалами, трудовими ресурсами;
- формування календарних планів робіт: діаграма Ганта по роботах, матеріалам і ресурсам, мережева діаграма, ведення календарних планів в розрізі проектів;
- облік виробництва у будівництві: контроль виконання виробничих показників, забезпечення ресурсами; ведення складського обліку: надходження і списання матеріалів, резервування матеріалів по складам будівництва;
- облік будівельно-монтажних робіт: ведення обліку будівельно-монтажних робіт, приймання субпідрядних робіт, реалізація з формуванням акту про приймання виконаних робіт.

Також в системі «BAS Будівництво» передбачено можливість завантаження кошторису з ПП "АВК 5" з формату OUTBOX, як окремий проект, з подальшим перетворенням в календарний план та обмін даними з ПП "BAS Будівництво. Керування фінансами", в якому, реалізована вся функціональність для ведення бухгалтерського і податкового обліку.

Отже, на даний час на ринку програмних продуктів пропонуються продукти різної функціональності: програмне забезпечення для фінансових розрахунків інвестиційних проектів, організації та контролю виконання будівельних робіт, складання кошторису будівництва, моніторингу ключових показників ефективності бізнесу. Але для оцінки інтегральної ефективності проектів будівництва продуктів не виявлено, а платформи бізнес-аналітики не

виконують багатокритеріальну оцінку та автоматичне завантаження техніко-економічні характеристики типових проектів будівництва. Навпаки, платформи управління будівництвом мають доступ до бази типових проектів, але не мають інструментів багатокритеріальної оцінки їх ефективності.

### **1.3 Постановка завдання для розробки**

#### **1.3.1 Мета та завдання**

Метою проектування є розробка серверної частини сайту для аналізу ефективності проектів у будівництві на підставі дослідження методик аналізу ефективності та сучасних технологій вирішення задач економічного моніторингу.

Для досягнення мети у роботі потрібно вирішити наступні завдання:

- проаналізувати предметну область;
- проаналізувати програмні засоби для вирішення аналогічних задач;
- виявити недоліки та не вирішені завдання у питанні автоматизації аналізу ефективності проектів у будівництві та сформулювати завдання для їх усунення;
- спроектувати структуру програмного забезпечення для вирішення поставленого завдання;
- розробити алгоритми обраних методів розрахунків ефективності проектів;
- зробити огляд серверних технологій та спроектувати архітектуру серверної частини сайту;
- розробити програмні коди серверної частини програмного забезпечення;
- розробити інтерфейс;
- виконати тестування системи;

- оформити пояснювальну записку до кваліфікаційної роботи.

Програмний продукт повинен дозволяти виконувати наступні функції:

- отримувати з БД інформацію про основні техніко-економічні характеристики заданого типу проектів (житлові, соціальні, виробничі);
- присвоювати рангову оцінку техніко-економічним характеристикам проекту для формування інтегрального критерію оцінки проектів;
- виконувати ранжирування проектів по ефективності за заданою функцією мети: найкращий, хороший, середній, допустимий, недопустимий;
- надавати користувачу інформацію про найбільш ефективний за інтегральною оцінкою проект.

### **1.3.2 Вхідні дані**

Вхідними даними – є основні характеристики проектів відповідно до трьох типів проектів:

- житлові;
- промислові;
- громадські.

Характеристики проектів на будівництво громадських будинків:

- найменування будинку, місце розташування;
- кошторисна вартість будівництва, тис. грн. ;
- витрати на охорону навколишнього середовища, тис. грн. ;
- поверховість будинку;
- загальна площа, кв. м;
- корисна площа, кв. м;
- будівельний об'єм будинку, куб. м;
- площа забудови, кв. м;

- площа ділянки, кв. м;
- кількість створених робочих місць;
- питома теплова потужність опалення, Гкал;
- питома річне тепло споживання, Гкал;
- річна потреба в паливі, Гкал;
- річна потреба воді, л;
- річна потреба електричній енергії, кВт;
- річна потреба тепловій енергії, Гкал;
- потужність;
- місткість, осіб;
- пропускна спроможність, осіб/годину;
- проектна організація (назва, адреса, мейл).

Характеристики проектів на будівництво житлових будинків:

- найменування будинку, місце його розташування;
- кошторисна вартість будівництва, тис. грн. ;
- витрати на охорону навколишнього середовища, тис. грн. ;
- поверховість будинку;
- загальна площа будинку, кв. м;
- загальна площа квартир у будинку, кв. м;
- загальний будівельний об'єм;
- площа забудови, кв. м;
- площа ділянки, кв. м;
- кількість створених робочих місць, од. ;
- питома теплова потужність опалення, Гкал;
- питома річне тепло споживання, Гкал;
- річна потреба в паливі, Гкал;
- річна потреба воді, л;
- річна потреба електричній енергії, кВт;
- річна потреба тепловій енергії, Гкал;
- загальна кількість однокімнатних квартир у будинку;

- загальна кількість двокімнатних та більше квартир у будинку;
- найменша площа квартири, кв. м;
- найбільша площа квартири, кв. м;
- площа вбудованих нежитлових приміщень, кв. м;
- проектна організація (назва, адреса, мейл).

Характеристики проектів на будівництво виробничих об'єктів:

- найменування та місце його розташування;
- кошторисна вартість будівництва, тис. грн. ;
- витрати на охорону навколишнього середовища, тис. грн. ;
- площа території, кв. м;
- потужність об'єкта обсяг продукції;
- потужність об'єкта місткість;
- потужність об'єкта пропускна спроможність;
- загальна кількість працюючих;
- продуктивність праці за рік;
- собівартість продукції;
- термін окупності інвестицій;
- вартість основних фондів (будинку, споруди);
- вартість основних фондів, які вибувають у процесі будівництва;
- тривалість будівництва;
- трудомісткість будівництва (у люд.-днях) ;
- річна потреба підприємства сировина та матеріали;
- річна потреба енергоресурси (млн. Квт/год);
- річна потреба теплоенергія в млн. Гкал;
- річна потреба нафтопродукти в тис. т);
- річна потреба вода (в тис. м3);
- транспорт зовнішній (тис. т.) ;
- прибуток;
- рентабельність;
- матеріаломісткість;

- енергомiсткiсть,
- питома теплова потужнiсть опалення;
- питоме рiчне тепло споживання;
- проектна органiзацiя (назва, адреса, мейл).

### **1.3.3 Результати роботи програми**

За результатами роботи програми користувач отримує характеристики проекту будiвництва, який є найбільш ефективним у вiдповiдностi з заданими користувачем критерiями оцiнки.

Програмний продукт повинен оптимiзувати витрати обмеженого бюджету на базi встановлених прiоритетiв i ранжирування проектiв з метою визначення серед альтернативних найбільш ефективного не тiльки в економiчному, а в широкому сенсi.

Для цього за введеними користувачем критерiями програма має надати iнформацiю про найбільш ефективний за iнтегральною оцiнкою проект будiвництва.

### **1.3.4 Вимоги до iнтерфейсу системи**

Iнтерфейс системи складається з iнтерфейсу адмiнiстратора та iнтерфейсу користувача.

Iнтерфейс користувача повинен бути дружнiм, контекстна зрозумiлим та забезпечувати:

- вибiр типу проекту: громадський, житловий, виробничий;
- вибiр iнтервальних значень характеристик проекту вiдповiдно до обраного типу проекту;



- заповнення рангових оцінок характеристикам проекту відповідно до обраного типу проекту;
- відображення назви та характеристик найкращого за критеріями проекту, знайденого по результатам розрахунку інтегральної оцінки.

Інтерфейс адміністратора повинен бути дружнім, контекстна зрозумілим та забезпечувати:

- перегляд, додавання, редагування, видалення проекту та його характеристик за типами проектів;
- пошук проекту за параметрами;
- надання/видалення для проекту мітки «еталон»;
- перегляд та видалення (за потреби) користувача з БД;
- додавання / редагування даних по проектним організаціям до бази даних.

### **1.3.5 Вимоги до надійності системи**

Надійність системи повинна забезпечуватись за такими напрямками: ●

- забезпечення працездатності компонентів програмної системи;
- збереження даних.

Забезпечення працездатності повинно підтримувати працездатність системи при відмові одного або декількох компонентів шляхом їх резервування.

Збереження даних має забезпечувати цілісність даних при програмно-апаратних відмовах, помилках, шляхом використання програмно-апаратних засобів та рішень, резервного копіювання, транзакційності при змінах даних. Авторизація повинна забезпечити захист від несанкціонованого доступу до зміни даних.

## 1.4 Висновки до розділу

У цьому розділі розглянуто методичні підходи до оцінки проектів у будівництві. Виділено різноплановий критеріальний підхід до оцінювання ефективності будівельних проектів, та групи критеріїв, за якими інтерна оцінка проектів для проектувальників, забудовників, споживачів, суспільства.

Проведений огляд програмних продуктів, які можна використовувати для оцінки ефективності проектів в будівництві показав, що проблемним питанням залишається автоматизація оцінювання будівельних проектів за інтегральними критеріями, сформульованими користувачем (експертом), який би враховував не тільки техніко-економічні фактори проекту, а і екологічні, соціальні та інші, задані користувачем.

Завдання, яке поставлено перед проектуванням – розробити програмний продукт, який дозволяв би користувачу обирати найбільш ефективний проект заданого типу по інтегральному критерію. Для цього користувач повинен мати змогу надавати ранги значущості всім характеристикам проекту.

## 2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Проектування архітектури

Архітектура програмного продукту визначається тими завданнями, які цей продукт має виконувати. Складемо у Rational Rose діаграму варіантів використання програмного продукту «BudExpert» (див. рис. 2.1).

Основними акторами системи є користувач та адміністратор.

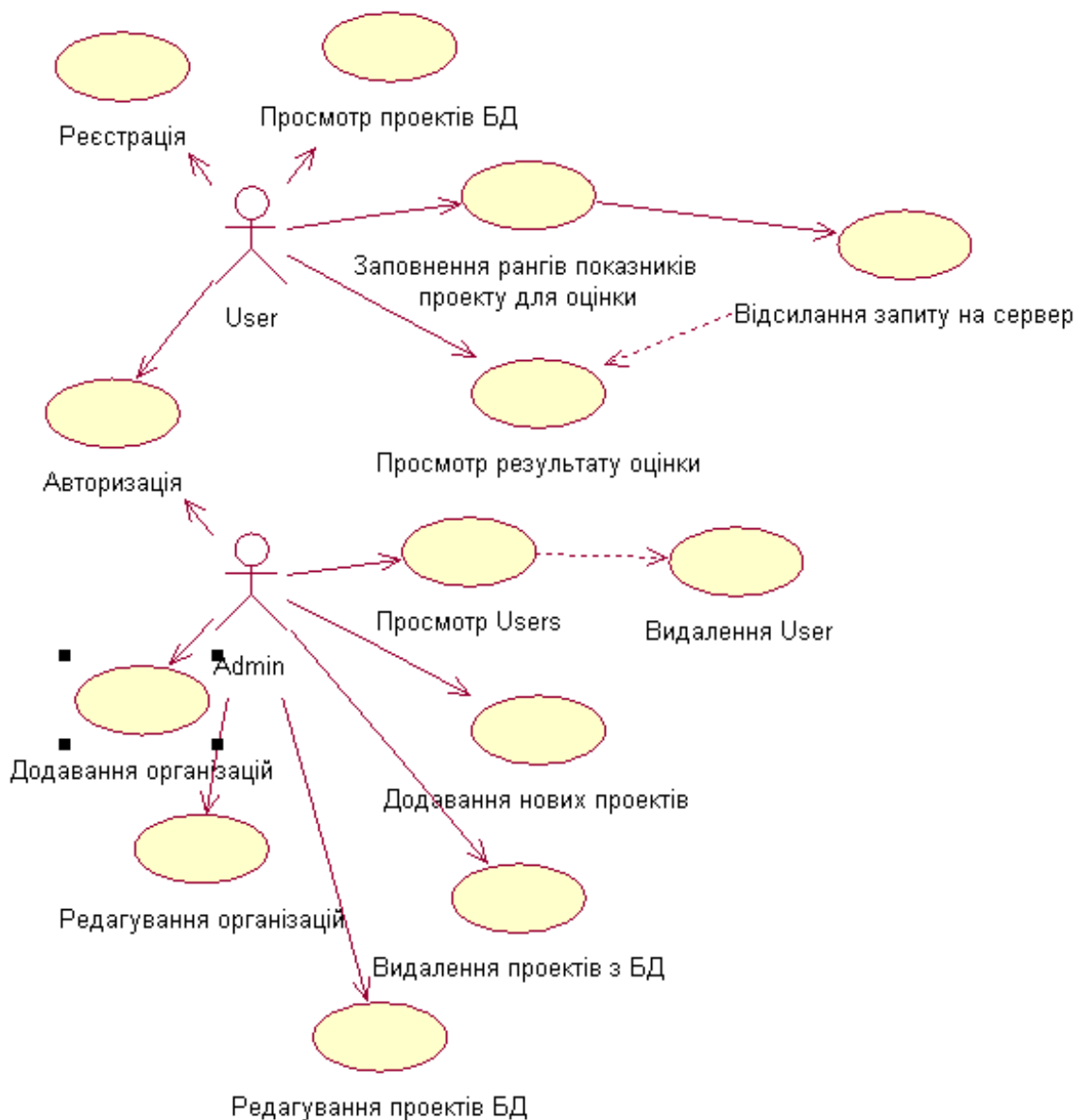


Рисунок 2.1 – Діаграма варіантів використання ПП «BudExpert»

Адміністратор має доступ до БД проектів, може додавати, видаляти, редагувати проекти, надавати проекту позначку «еталон» для використання у порівняльних розрахунках.

Користувач після авторизації/реєстрації має змогу на своєму робочому місці формувати запит для вибору проекту у відповідній екранній формі та отримувати від віддаленого сервера результат пошуку та розрахунку.

Отже, програмний продукт «BudExpert» має забезпечувати:

- зберігання та редагування структурованих даних бази будівельних проектів різних типів;
- формування запиту користувача на відбір проекту по заданим критеріям;
- виконання розрахунків та відбір найбільш ефективних проектів по запиту користувача; надання результату.

Для вирішення поставлених завдань найбільш доцільною, на нашу думку, є клієнт-серверна взаємодія (див. на рис. 2.2), під якою розуміється доступ клієнта до ресурсів, які зберігаються в сервері, з метою отримання інформації та обробки даних за допомогою серверних програм. Клієнт-серверна архітектура дозволяє організувати доступ до БД багатьох користувачів, а використання Інтернету для комунікацій робить такий доступ можливим для як завгодно віддалених користувачів.

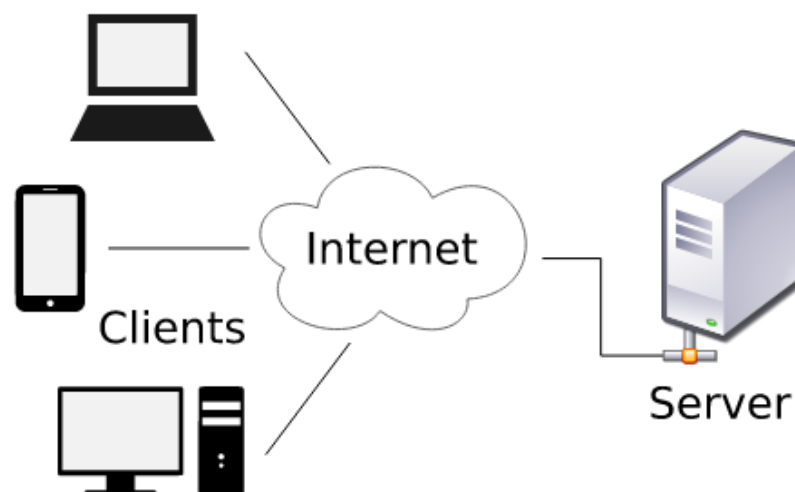


Рисунок 2.2 – Архітектура клієнт-серверної взаємодії

У клієнт-серверній архітектурі завдання чи мережні навантаження розподілені між постачальниками послуг (серверами) та замовниками послуг (клієнтами). Клієнт та сервер – це програмне забезпечення, розташоване на різних обчислювально-комунікаційних пристроях, яке взаємодіє через Інтернет-мережу за допомогою мережевих протоколів.

Технології клієнт-серверної взаємодії різняться способом реалізації інтерактивності та розподілом функцій між сервером та клієнтом.

При серверному виконанні браузер клієнта виконує лише отримання та візуальне представлення інформації з сервера, а також формування запитів до сервера

До переваг серверного виконання програм відносяться:

- відсутність проблем із сумісністю платформ;
- відсутність проблем із дотриманням прав власності на програмне забезпечення;
- відсутність небезпеки несанкціонованого доступу до даних клієнта;
- вимоги до характеристик клієнтських комп'ютерів знижуються.

Недоліками серверного виконання є:

- небезпека перевантаження сервера за великої кількості клієнтів;
- підвищені вимоги до пропускної спроможності мережі під час інтерактивної роботи користувачів.

Отже, обираємо саме серверне динамічне виконання ПП «BudExpert».

Програми серверу очікують від клієнтських програм запити та надають їм ресурси у вигляді даних або сервісних функцій. Оскільки одна програма-сервер може виконувати запити від багатьох програм-клієнтів, її розміщують на спеціально виділеній обчислювальній машині, налаштованій особливим чином, як правило, спільно з іншими програмами-серверами, тому продуктивність такої машини повинна бути високою.

На діаграмі рис. 2.3 наведені основні складові архітектури ПП «BudExpert» і пронумеровані ярлики послідовності операцій при зверненні користувача з запитом до БД.

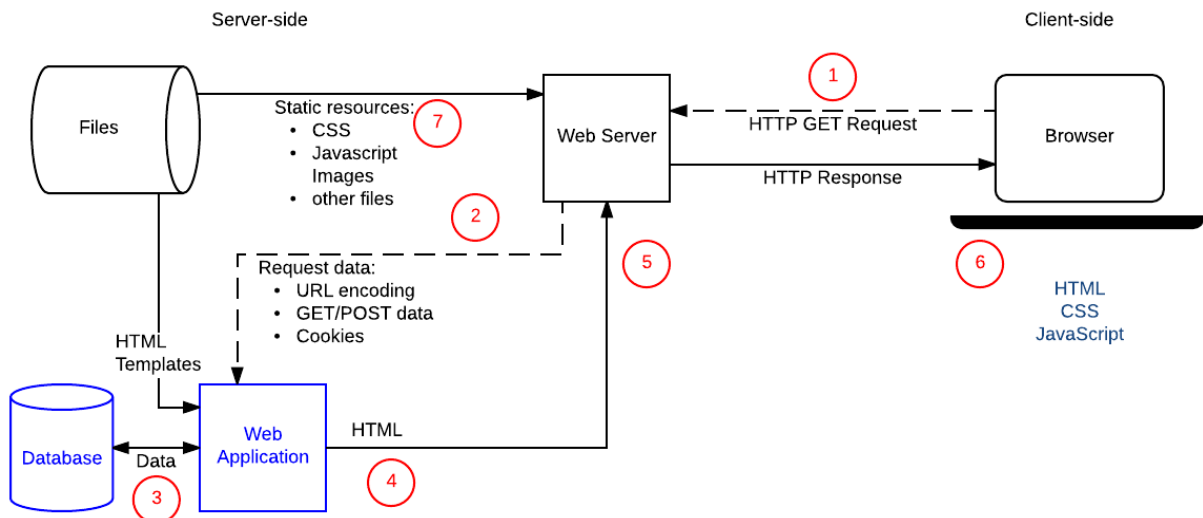


Рисунок 2.3 – Клієнт-серверна архітектура ПП «BudExpert»

Частинами сайту, які роблять його динамічним, є:

- веб-сервер, що обробляє HTTP-запити і повертає HTTP-відповіді,
- база даних, яка містить інформацію про будівельні проекти,
- HTML -шаблони.

Після того, як користувач відправить форму з запитом, послідовність операцій буде наступною:

1) Веб-браузер відправляє HTTP-запит GET на сервер з використанням базової URL-адреси ресурсу (/best) та кодування у формі URL-параметрів проекту. Запит GET використовується тому, що йдеться лише про запит вибірки даних (а не про їх зміну).

2) Веб-сервер визначає, що запит є «динамічний» і пересилає його до веб-додатку для обробки (веб-сервер визначає, як обробляти різні URL-адреси на основі правил зіставлення шаблонів, визначених у його конфігурації).

3) Веб-додаток визначає ціль запиту – отримати «список кращих проектів» на основі URL (/best/) і дізнатися ім'я проекту з URL-адреси. Потім веб-додаток отримує необхідну інформацію з бази даних (використовуючи додаткові «внутрішні» параметри, щоб визначити, які проекти є «кращими»).

4) Веб-додаток динамічно створює HTML-сторінку з даними із бази даних, заповнюючи HTML-шаблон.

5) Веб-додаток повертає генеровану HTML-сторінку у веб-браузер (через

веб-сервер) разом із кодом стану HTTP 200 («успіх»). Якщо виникає перешкода поверненню HTML-сторінки, веб-додаток повертає інший код, наприклад, 404, що вказує, що команда не виконана.

6) Веб-браузер обробляє повернуту HTML-сторінку і надсилає окремі запити, щоб отримати будь-які інші файли CSS або JavaScript, на які сторінка посилається.

7) Веб-сервер завантажує статичні файли з файлової системи і повертає їх безпосередньо до браузера (знову ж таки, правильна обробка файлів заснована на правилах конфігурації та зіставленні шаблонів URL).

Операція з оновлення запису в базі даних, яка доступна адміністратору, обробляється аналогічно, за винятком того, що HTTP-запит із браузера має бути закодований як запит POST.

Отже, обираємо клієнт-серверну архітектуру ПП «BudExpert» серверного виконання, що дозволяє вирішити поставлені перед проектуванням завдання, а також розширювати функціональні можливості продукту у перспективі.

## **2.2 Проектування бази даних**

Проектування БД виконуємо за допомогою платформи моделювання ERwin і починаємо з визначення основних сутностей моделі даних.

Основні сутності інформаційної моделі логічного рівня бази даних це:

- проекти житлових будинків (House\_list);
- проекти громадських будинків (Social\_list);
- проекти виробничих будинків (Prom\_list);
- зареєстровані користувачі програми (Users).

Отже, інформаційна модель у вигляді відображення сутностей наведена на рис. 2.4. При проектуванні цієї моделі повстало питання – чи не є всі проекти однією сутністю. Не є, тому що мають більшість різних атрибутів, хоч деякі з атрибутів і однакові для всіх трьох сутностей.

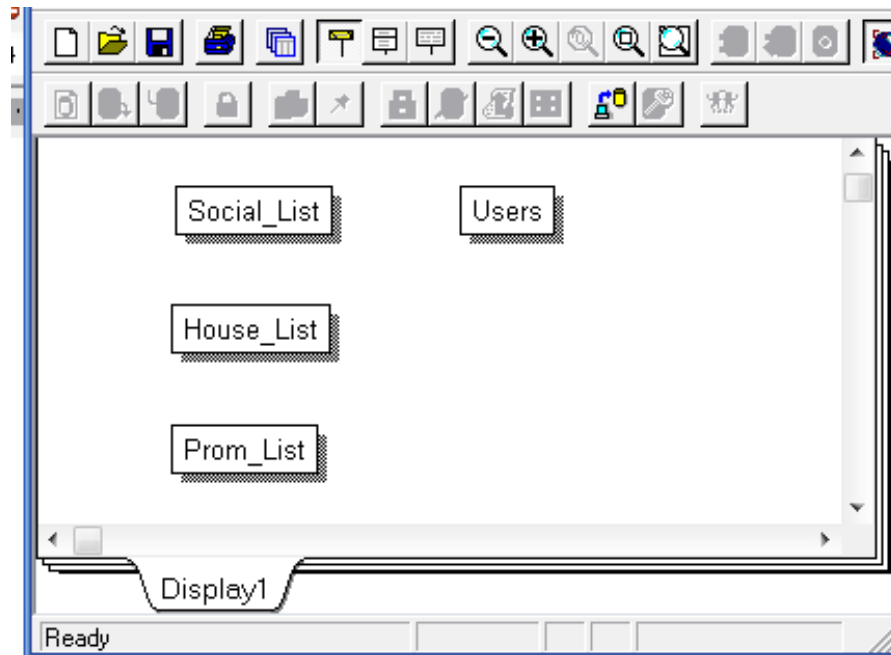


Рисунок 2.4 – Інформаційна модель у вигляді відображення сутностей

Атрибути проектів трьох типів задані у п. 1.3.2 даної роботи. Аналіз атрибутів різних типів проектів показує, що частково вони однакові для всіх типів проектів – назва проекту, вартість будівництва, витрати на природне зовнішнє середовище, ресурсні витрати, проектна організація.

Але більша частина атрибутів відрізняються для різних типів проектів. Тому доцільним є виділення кожного типу проектів у окрему сутність та збереження інформації по кожному типу проектів у окремій таблиці. Таке розбиття та нормалізація даних надасть змогу сформуванню кращої структури.

Нормалізація – це процес, у результаті якого можна позбавитися дефектів проектування бази даних [8, с. 67].

Метою нормалізації є:

- зменшення об'єму для зберігання даних;
- підвищення ефективності роботи БД. Процедура нормалізації виконується поетапно.

В ідеалі при нормалізації треба домогтися, щоб будь-яке значення зберігалось в базі в одному примірнику, причому значення це не має бути отримано розрахунковим шляхом з інших даних, що зберігаються в базі [8].



Для забезпечення цілісності даних виділяємо сутність Проектні Організації і встановлюємо відношення 1:М цієї сутності Organization з кожною з трьох сутностей проектів – Social\_List, House\_List, Prom\_List. Для цього користуємося інструментом Relationship Editor, що відображено на рис. 2.5.

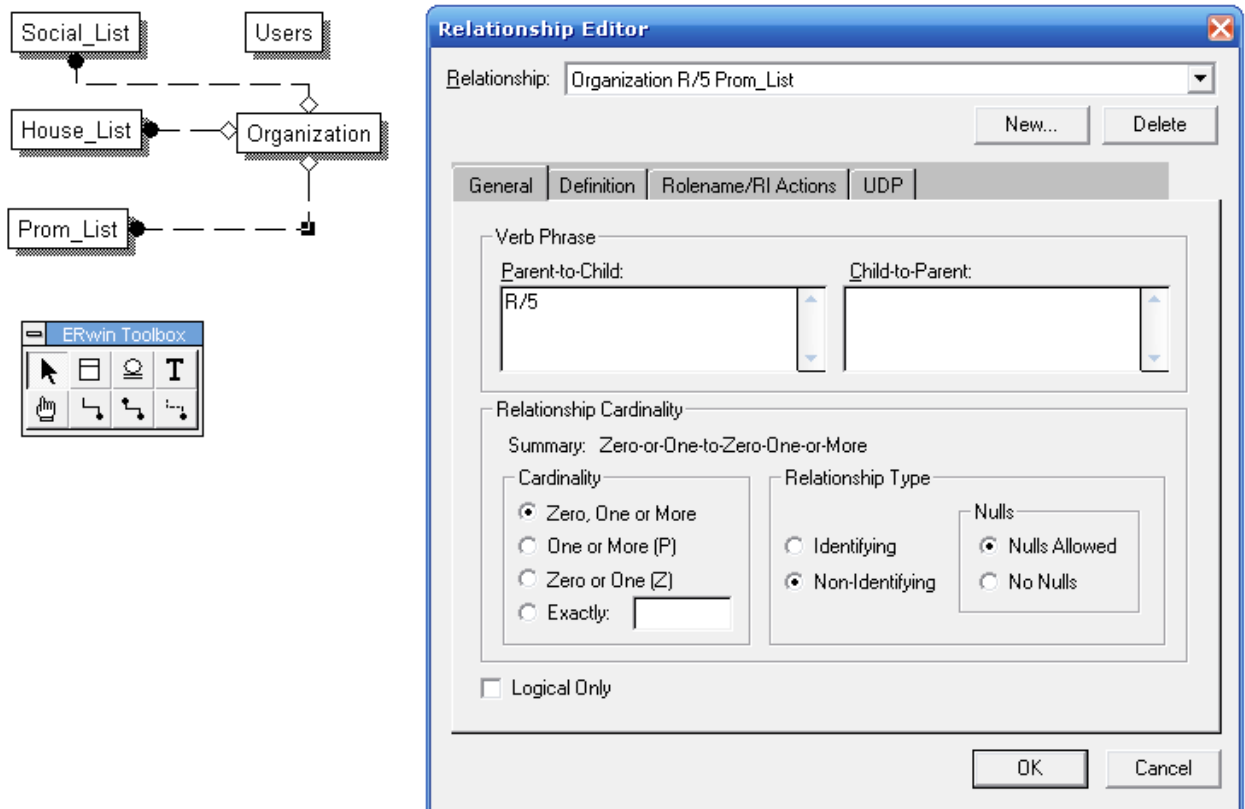


Рисунок 2.5 – Встановлення зв'язків сутностей інформаційної моделі за допомогою вікна Relationship Editor

Для кожної сутності додаємо атрибути відповідно до вимог, наведених у п. 1.3.2. Крім перелічених у завданні атрибутів додаємо атрибут первісного ключа-ідентифікатора Kod, та атрибут Etalon, який може використовуватись адміністратором для встановлення позначки еталонного проекту. В результаті отримуємо логічну модель БД, наведену на рис. 2.6.

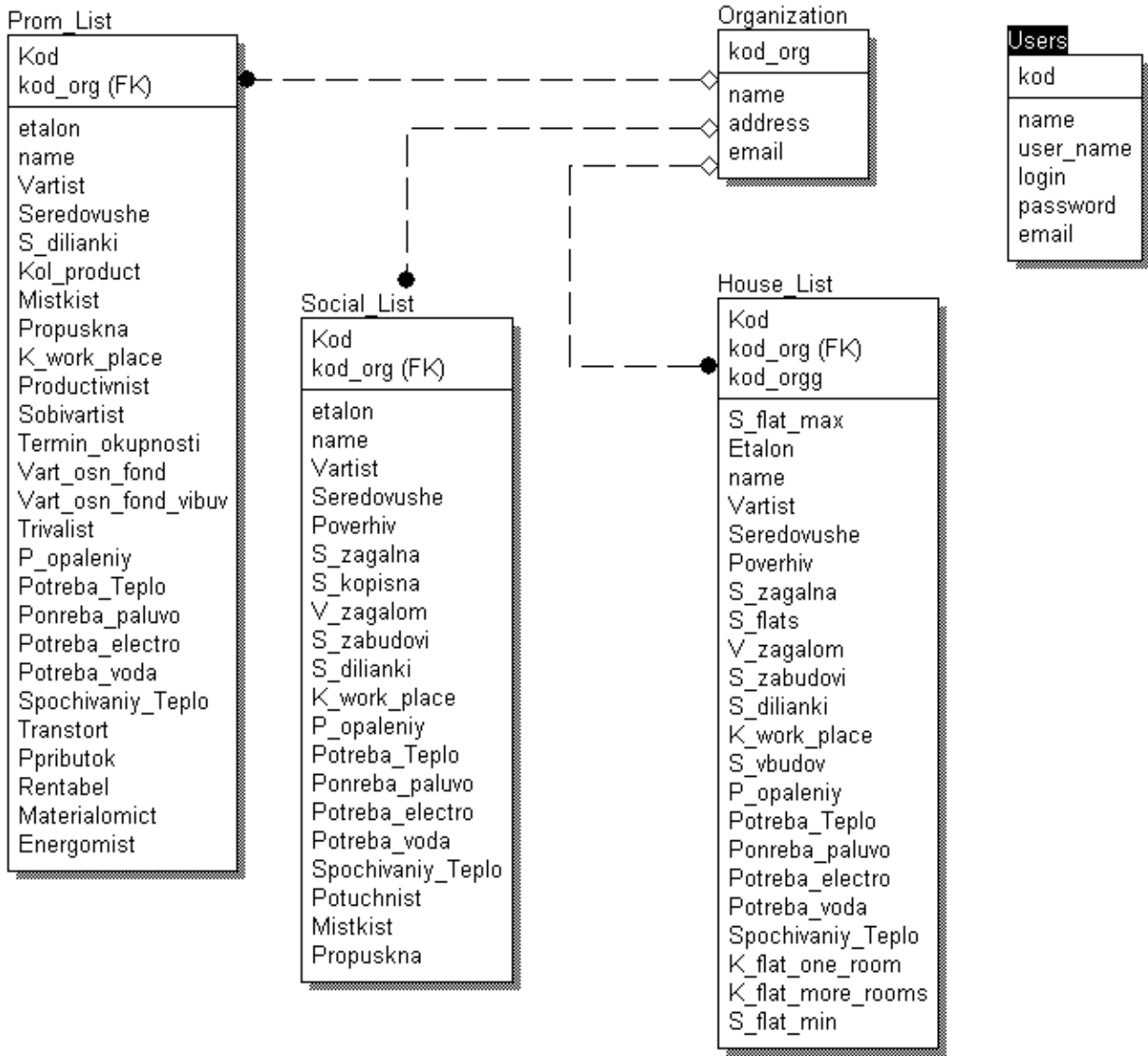


Рисунок 2.6 – Інформаційна модель логічного рівня

В процесі додавання атрибутів обираємо для кожного з них тип даних, відображення, назначаємо ключові поля. Для зв'язку з сутностями проектів із сутністю організації додаємо до атрибутів сутностей проектів атрибут kod\_org та назначаємо його вторинним ключем.

В результаті отримуємо фізичну інформаційну модель, яку надано на рис. 2.7. Скрипт генерації бази даних наведено у Додатку А.

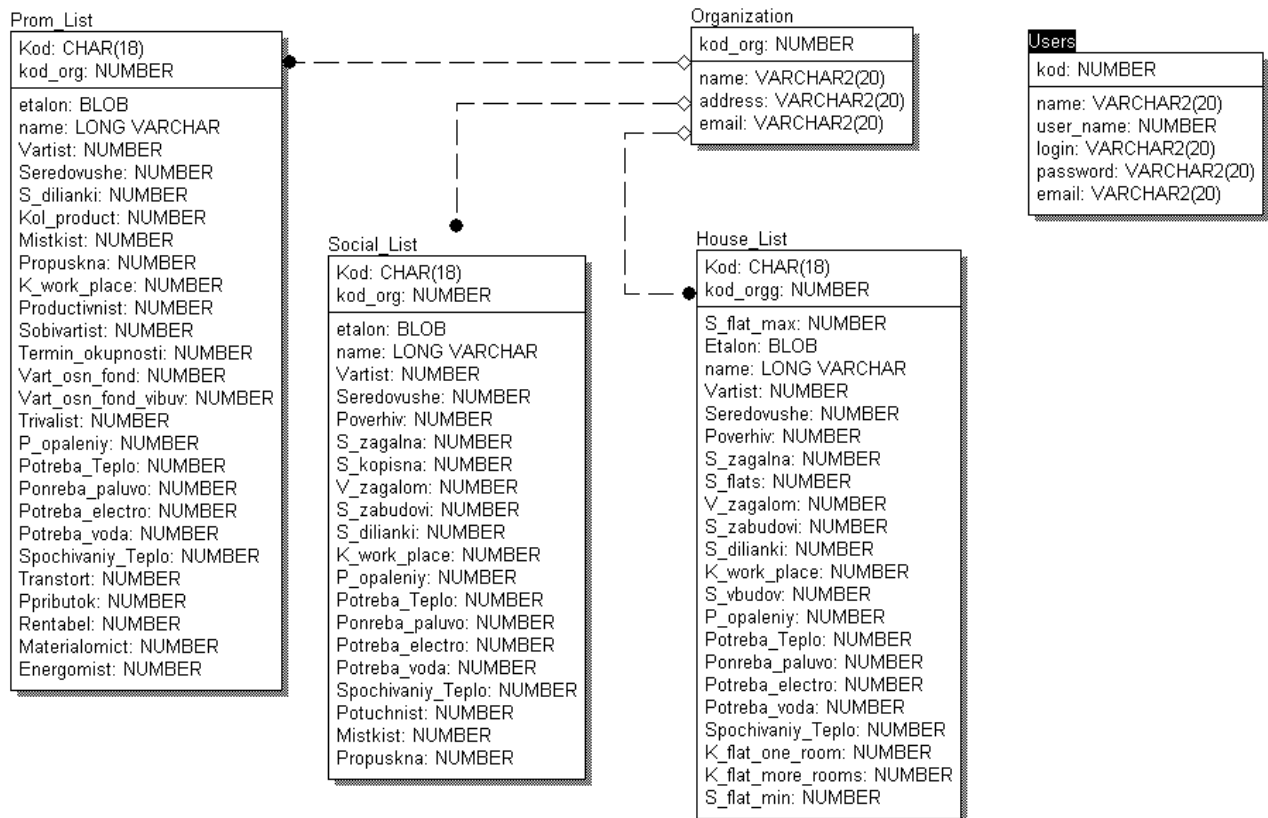


Рисунок 2.7 – Інформаційна модель фізичного рівня

Відповідно до обраної архітектури програмного продукту, база даних буде зберігатися на сервері, а доступ до даних БД буде організовано за допомогою серверного веб-додатку.

### 2.3 Вибір засобів реалізації програмного забезпечення

Створення клієнт-серверних систем потребує використання спеціальних технологій та підходів під час проектування серверної частини.

Вибір правильних технологій та підходів може суттєво вплинути на успішність проекту, а вибір правильної архітектури сервера може значно вплинути на продуктивність, масштабованість та безпеку системи.

### *2.3.1 Огляд серверних технологій*

В даний час більшість серверів працює під управлінням програм або Internet Information Server (IIS) в середовищі Windows, або Apache в середовищі ОС UNIX. Кожен сервер має доменне ім'я, зареєстроване на сервері DNS. Зв'язок зазвичай здійснюється за протоколом http (HyperText Transfer Protocol). На серверній стороні вона реалізується за допомогою програми, яка називається http-сервером, який обробляє запити, що надходять від браузерів, шукає запитаний документ і видає клієнту або вміст знайденого файлу, або повідомлення про помилку, якщо такий файл не було знайдено або доступ до нього заборонено [13].

Браузер клієнта призначений лише для отримання та обробки інформації з сервера. Вплив на серверні дані з боку клієнта можливий лише за допомогою спеціальних засобів, наприклад, на основі технологій CGI або FTP.

До основних Web-технологій відносяться:

- включення сценарію в HTML-документ (мови сценаріїв JavaScript, VBScript);
- активні серверні сторінки (ASP) – сценарій виконується на сервері;
- CGI (Common Gateway Interface) - за кожним запитом сервер створює процес (додаток) у своєму адресному просторі;
- ISAPI (Internet Server Application Programming Interface) — програми створюються у загальному адресному просторі сервера;
- аплети (Applets) - додаток на Java передається до виконання клієнту;
- сервлети (Servlets) - додаток Java виконується на сервері;
- ActiveX — компоненти керування, які можуть виконуватися у вузлах сервера чи клієнта.

Однією з ключових технологій розробки клієнт-серверних систем є використання високопродуктивних обчислювальних систем та хмарних технологій. Хмарні технології дозволяють створювати гнучкі і масштабовані системи, здатні адаптуватися до потреб, що змінюються.

Однією з ключових технологій під час проектування серверної частини є використання мікросервісної архітектури. Ця архітектура дозволяє розбити програму на невеликі сервіси, кожен з яких може бути запущений на окремому сервері і розроблятися і масштабуватися незалежно.

Для забезпечення роботи та запуску серверної програми використовуються різні технології. Сутність та приклади деякі з них наведені у табл. 2.1.

Таблиця 2.1 — Технології забезпечення роботи та запуску серверної програми

Назва технології	Приклади	Сутність технології
Веб-сервери	Apache, Nginx, Microsoft IIS	Обслуговування запитів клієнтів та передача даних між клієнтом та сервером. Робота з протоколами HTTP, HTTPS, FTP, SMTP та ін.
СУБД	MySQL, PostgreSQL, Microsoft SQL Server та Oracle	Зберігання та керування даними
Системи контролю версій	Git, SVN та Mercurial	Зберігання коду серверної програми та керування версіями програми
Контейнеризація	Docker, Kubernetes	Створення та запуск програми в ізольованих контейнерах
Автентифікації та авторизації	SSL, TLS, OAuth	Контроль доступу, запобігання несанкціонованому доступу до сервера та даних
Системи автоматичного розгортання	Jenkins, GitLab CI, GitHub, Bitbucket та Travis C	Автоматизація процесу розгортання сервера

Для забезпечення масштабованості системи можна використовувати технології контейнеризації, такі як Docker та Kubernetes. Контейнеризація дозволяє запакувати додаток та його залежності в єдиний пакет, який може бути запущено на будь-якому сервері, забезпечуючи максимальну гнучкість та масштабованість.

Ще однією важливою технологією є використання баз даних та їх оптимізація. Дані можуть бути розподілені на кілька серверів для збільшення швидкості обробки та доступу до них. Також важливо правильно вибирати тип бази даних, залежно від характеристик проекту.

СУБД: СУБД (Система Управління Базами Даних) використовуються для зберігання та керування даними, що використовуються серверним додатком. Різні СУБД, такі як MySQL, PostgreSQL, Microsoft SQL Server та Oracle, надають різні можливості для обробки та зберігання даних [13].

Веб-сервери: Веб-сервери, такі як Apache, Nginx та Microsoft IIS, використовуються для обслуговування запитів клієнтів та передачі даних між клієнтом та сервером. Вони можуть працювати з різними протоколами, такими як HTTP, HTTPS, FTP, SMTP, та багато інших.

Для забезпечення безпеки даних та захисту від злому необхідно використовувати сучасні технології автентифікації та авторизації, такі як SSL, TLS, OAuth та інші. Також для інформаційної безпеки важливо використовувати механізми контролю доступу, щоб запобігти несанкціонованому доступу до сервера та даних.

Системи контролю версій: Системи контролю версій, такі як Git, SVN та Mercurial, використовуються для зберігання коду серверної програми та керування версіями програми. Вони дозволяють команді розробників спільно працювати над програмою та відстежувати зміни у коді.

Контейнеризація: Контейнеризація, наприклад Docker, дозволяє створювати та запускати програми в ізольованих контейнерах. Це полегшує розгортання програми та управління її залежностями.

Системи автоматичного розгортання: Системи автоматичного розгортання, такі як Jenkins, GitLab CI, GitHub, Bitbucket та Travis CI, дозволяють автоматизувати процес розгортання сервера. Всі вони мають приватні репозиторії на безкоштовних тарифах та підтримують роботу з Git.

### ***2.3.2 Вибір технологій для вирішення задачі***

Аналіз існуючих технологій реалізації серверного додатка показав, що найбільш доцільним для даного проекту є наведений нижче набір технологій.

Мова програмування: Python – це проста у використанні та інтерпретована мова програмування, яка дозволяє швидко створювати прототипи та розробляти програми. Python також широко використовується в наукових обчисленнях та аналізі даних, що може бути корисним для багатьох серверних додатків.

Фреймворк: Flask – це легкий і простий у використанні фреймворк для створення веб-застосунків на Python. Flask не нав'язує жорстких обмежень на структуру програми та дозволяє швидко створювати прототипи.

СУБД: SQLite – це проста у використанні реляційна база даних, яка зберігається в одному файлі. SQLite не потребує встановлення окремого сервера бази даних і підтримує SQL-запити, що дозволяє легко працювати з даними.

Веб-сервер: Gunicorn – це швидкий та надійний веб-сервер для Python, який може обробляти безліч запитів одночасно. Gunicorn також інтегрується з Flask, що полегшує його використання.

Контейнеризація: Docker – це інструмент для керування контейнерами, які дозволяють запакувати додаток та його залежності в єдиний образ. Docker полегшує розгортання та масштабування програми на різних серверах.

Вибір даних технологій обґрунтовується їх простотою у використанні, ефективністю та широким використанням у програмуванні. Цей набір

технологій дозволить швидко створювати прототипи та розробляти серверні програми, які будуть надійно працювати та легко масштабуватись.

Архітектура клієнт-серверного додатка є сукупністю клієнтської та серверної частин, які взаємодіють між собою через інтернет. Клієнтська частина може бути написана на різних технологіях, але в даному випадку ми будемо використовувати JavaScript та ReactJS.

Серверна частина складатиметься з Flask-фреймворку мовою програмування Python, який працюватиме з базою даних SQLite та веб-сервером Gunicorn.

Крім того, будемо використовувати Docker для упаковки програми та її залежностей у контейнер.

Клієнтська частина взаємодітиме з серверною частиною через API, який буде реалізований у Flask-додатку. Для обміну даними між клієнтом та сервером ми можемо використовувати протокол HTTP та формат даних JSON.

Таким чином, для проектування ПП «BudExpert» обираємо наступні інструменти:

- клієнтська частина на JavaScript та ReactJS,
- серверна частина на Python з використанням Flask,
- база даних SQLite;
- веб-сервер Gunicorn,
- контейнеризація програми Docker.

## ***2.4 Висновки до розділу***

У цьому розділі розглянуто варіанти використання програмного продукту та обґрунтовано вибір для ПП «BudExpert» клієнт-серверної архітектури серверного виконання та розроблена така архітектура. Спроектвана база даних відповідно до вхідних даних проекту. Обрано технології виконання клієнтської та серверної частини продукту, бази даних, веб-серверу. У наступному розділі виконаємо реалізацію компонентів системи за обраними технологіями.



## 3 РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

### 3.1 Розробка інтерфейсу

Для доступу до сховища даних через браузер потрібен веб-інтерфейс, який забезпечував функції системи доступні як для адміністратора, так і для користувача. Адміністратору надається більше функцій, а саме редагування (додання, видалення) проектів бази даних, тому потрібною є авторизація в системі.

Для розробки карти сайту скористаємось безкоштовним онлайн сервісом gloomaps [19], який має оптимальний набір функцій та широкі можливості експорту готового проекту. Розроблена карта сайту наведена на рис. 3.1.

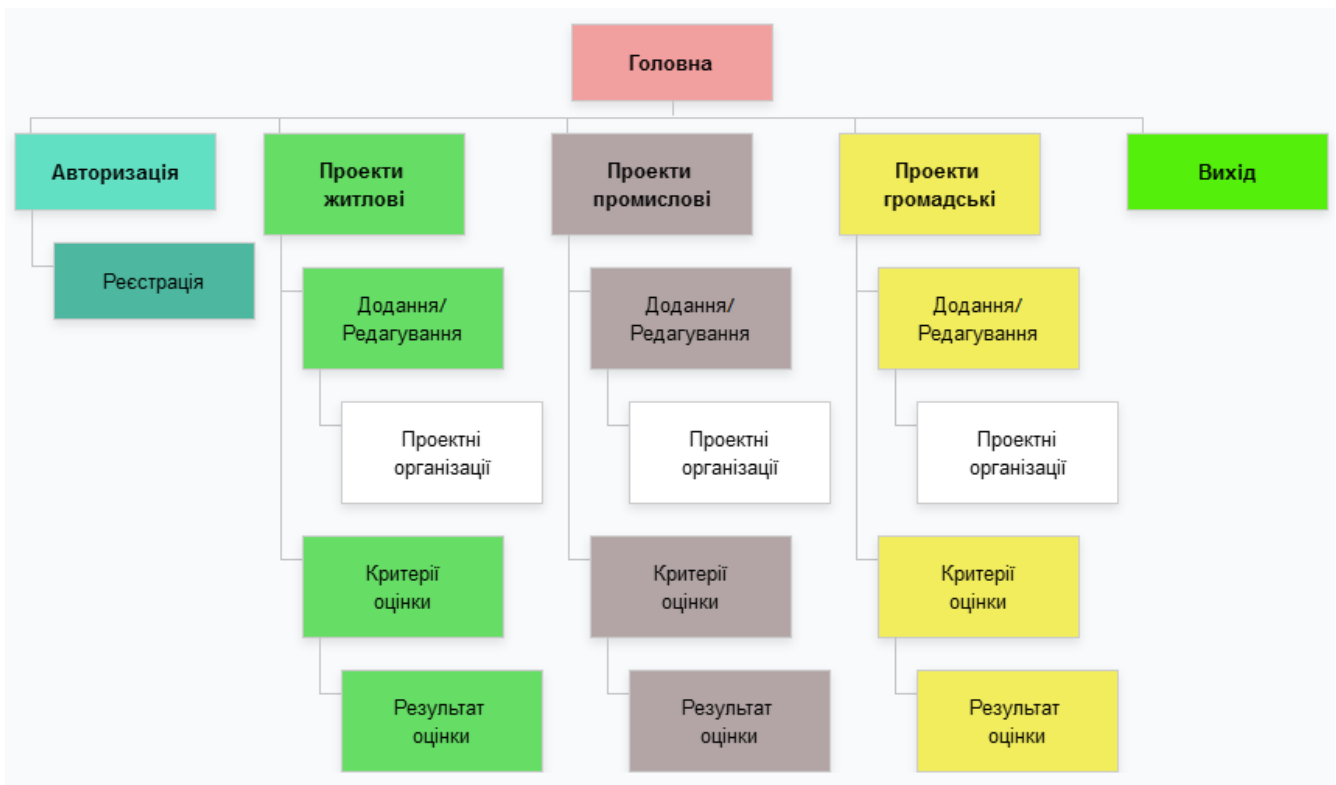


Рисунок 3.1 – Карта сайту BudExpert

Для ПП «BudExpert» ми обрали деревовидну структуру сайту. Оскільки кожний тип проектів значно відрізняється набором атрибутів, то для кожного з типів проектів створено свої форми при однаковій функціональності. Треба зауважити, що оскільки темою кваліфікаційної роботи є серверна частина сайту, то будемо використовувати мінімалістський дизайн клієнтського інтерфейсу, який би забезпечував функціональність системи оцінки ефективності проектів.

Для функціонування системи розроблено наступні форми у вигляді HTML-сторінок:

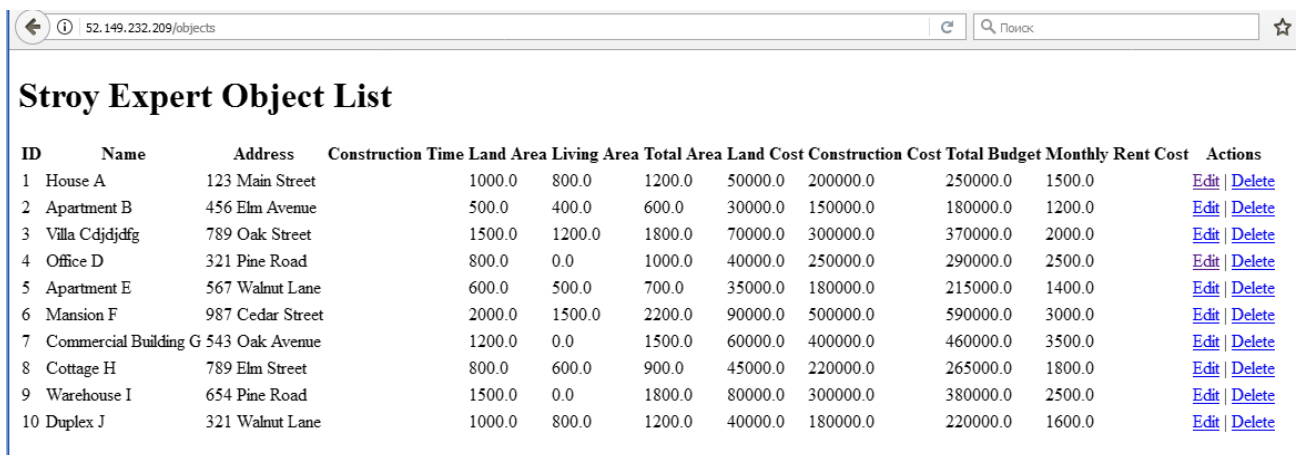
- авторизації;
- реєстрації;
- головна;
- житлові проекти;
- громадські проекти;
- промислові проекти;
- проектні організації;
- додання/редагування житлового проекту;
- додання/редагування громадського проекту;
- додання/редагування промислового проекту;
- призначення критеріїв оцінки житлових проектів;
- призначення критеріїв оцінки громадських проектів;
- призначення критеріїв оцінки промислових проектів;
- відображення результатів оцінки проектів.

Послідовність активізації подій визначається картою сайту (див. рис. 3.1). Всі операції по формуванню змісту веб-сторінок, підключенню до БД, отриманню з неї потрібної інформації виконується серверним додатком. До клієнтського браузера передається заповнена HTML-сторінка. При редагуванні або додаванні нового запису до БД, на сервер передається інформація, записана користувачем у поля форми браузера для обробки та коректного запису до БД. Тому для відображення і для редагування (додання) інформації з бази даних створюємо окремі об'єкти форм.

Приклад створення екземпляру класу форми ObjectForm та оголошення її полів наведено нижче:

```
class ObjectForm(FlaskForm):
    id = HiddenField('id')
    name = StringField('Name')
    address = StringField('Address')
    construction_time = StringField('Construction Time')
    land_area = FloatField('Land Area')
    living_area = FloatField('Living Area')
    total_area = FloatField('Total Area')
    land_cost = FloatField('Land Cost')
    construction_cost = FloatField('Construction Cost')
    total_budget = FloatField('Total Budget')
    monthly_rent_cost = FloatField('Monthly Rent Cost')
```

Вигляд форми реєстру житлових проектів наведено на рис. 3.2.



ID	Name	Address	Construction Time	Land Area	Living Area	Total Area	Land Cost	Construction Cost	Total Budget	Monthly Rent Cost	Actions
1	House A	123 Main Street	1000.0	800.0	1200.0	50000.0	200000.0	250000.0	1500.0	<a href="#">Edit</a>   <a href="#">Delete</a>	
2	Apartment B	456 Elm Avenue	500.0	400.0	600.0	30000.0	150000.0	180000.0	1200.0	<a href="#">Edit</a>   <a href="#">Delete</a>	
3	Villa Cdjdjdfg	789 Oak Street	1500.0	1200.0	1800.0	70000.0	300000.0	370000.0	2000.0	<a href="#">Edit</a>   <a href="#">Delete</a>	
4	Office D	321 Pine Road	800.0	0.0	1000.0	40000.0	250000.0	290000.0	2500.0	<a href="#">Edit</a>   <a href="#">Delete</a>	
5	Apartment E	567 Walnut Lane	600.0	500.0	700.0	35000.0	180000.0	215000.0	1400.0	<a href="#">Edit</a>   <a href="#">Delete</a>	
6	Mansion F	987 Cedar Street	2000.0	1500.0	2200.0	90000.0	500000.0	590000.0	3000.0	<a href="#">Edit</a>   <a href="#">Delete</a>	
7	Commercial Building G	543 Oak Avenue	1200.0	0.0	1500.0	60000.0	400000.0	460000.0	3500.0	<a href="#">Edit</a>   <a href="#">Delete</a>	
8	Cottage H	789 Elm Street	800.0	600.0	900.0	45000.0	220000.0	265000.0	1800.0	<a href="#">Edit</a>   <a href="#">Delete</a>	
9	Warehouse I	654 Pine Road	1500.0	0.0	1800.0	80000.0	300000.0	380000.0	2500.0	<a href="#">Edit</a>   <a href="#">Delete</a>	
10	Duplex J	321 Walnut Lane	1000.0	800.0	1200.0	40000.0	180000.0	220000.0	1600.0	<a href="#">Edit</a>   <a href="#">Delete</a>	

Рисунок 3.2 – Форма реєстру житлових проектів (налагоджувальний варіант)

Для заповнення форми інформацією з БД виконуються події, відображені у наступному фрагменті коду серверного додатку:

```
import sqlite3
// Підключення до бази даних
def get_all_objects():
    conn = sqlite3.connect('stroy.db')
    cursor = conn.cursor()
// Запит до БД
    cursor.execute('SELECT * FROM objects')
    rows = cursor.fetchall()
//Считування масиву даних
    objects = []
    for row in rows:
        object_data = {
            'id': row[0],
            'name': row[1],
            'address': row[2],
            'build_time': row[3],
            'land_area': row[4],
            'living_area': row[5],
            'total_area': row[6],
            'land_cost': row[7],
            'construction_cost': row[8],
            'total_budget': row[9],
            'monthly_rent_cost': row[10]
        }
        objects.append(object_data)
// Відключення від БД
    conn.close()
// передача масиву даних
    return objects
```

На рис. 3.3 відображена форма редагування (або додавання нового проекту). Ця форма на відміну від реєстру проектів призначена для формування масиву даних, який буде записано до БД по натисканню кнопки Save.

The image shows two side-by-side screenshots of web forms. The left screenshot is titled 'Edit Object' and has a URL of '52.149.232.209/edit\_object/1'. It contains several input fields with pre-filled values: Name: House A, Address: 123 Main Street, Construction Time: (empty), Land Area: 1000.0, Living Area: 800.0, Total Area: 1200.0, Land Cost: 50000.0, Construction Cost: 200000.0, Total Budget: 250000.0, and Monthly Rent Cost: 1500.0. A 'Save' button is at the bottom. The right screenshot is titled 'New Object' and has a URL of '52.149.232.209/add\_objectform'. It contains the same set of input fields but they are all empty. A 'Save' button is also at the bottom.

Рисунок 3.3. – Форма редагування та додання нового житлового проекту (налагоджувальний варіант)

Функції наведеної на рис. 3.3 форми виконуються за допомогою наступного програмного фрагменту:

```
# Ініціалізуємо форму з даними проекту
form = ObjectForm(request.form)
print(form.validate())
if request.method == 'POST' and form.validate():
    # Оновлюємо дані проекту з форми
    # object.id =
```

```

# Зберігаємо оновлені дані в БД
# update_object(obj)
print(form.name.data)

save_object( form.name.data, form.address.data, form.construction_time.data,
form.land_area.data, form.living_area.data, form.total_area.data, form.land_cost.data,
form.construction_cost.data, form.total_budget.data, form.monthly_rent_cost.data)

return jsonify({'message': 'Created Successfully!!!'})

# Спрямовуємо користувача до реєстру проектів
# return

#redirect(url_for('objects'))

```

Треба зауважити, що інтерфейс клієнтської частини сайту у даній роботі виконує завдання демонстрації функціонування серверної частини сайту, тому повинен бути простим і максимально демонструвати роботу з сервером, яка забезпечується за розробленими алгоритмами, розглянутими у наступному підрозділі.

### ***3.2 Розробка алгоритмів***

Основними алгоритмами функціонування ПП «BudExpert» є алгоритм обміну даними клієнта з сервером та алгоритм пошуку найбільш ефективного проекту за заданим інтегральним критерієм ефективності проекту.

Послідовність обробки та обміну даними при відборі найбільш ефективного проекту зручно відобразити за допомогою UML-діаграми станів (рис. 3.4).

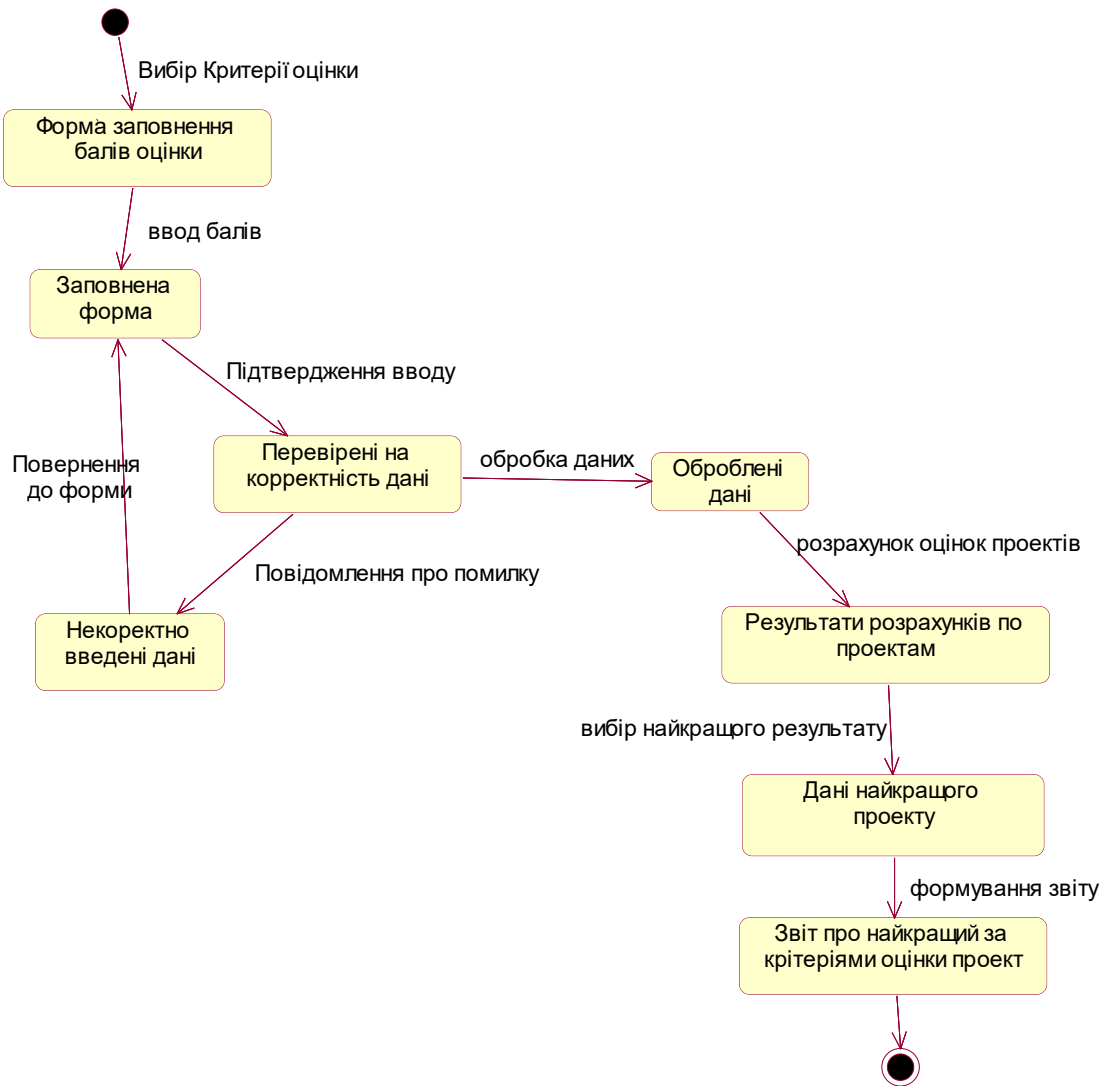


Рисунок 3.4 – Діаграма станів при виконанні пошуку найкращого проекту за інтегральною оцінкою ефективності

Відповідно до наведеної діаграми, для пошуку найбільш ефективного проекту користувач обирає форму «Критерії оцінки» для відповідного типу проекту (житлового, або громадського, або виробничого). У формі користувач надає бали значущості показникам проекту відповідно до своєї мети. Заповнені дані перевіряються у клієнтському додатку на коректність і в разі невідповідності даних вимогам видається повідомлення про помилку та користувач повертається до зміни даних. Інакше дані обробляються та передаються на сервер.

У серверному додатку відповідно до отриманого запиту виконується алгоритм розрахунку інтегральної оцінки для кожного із проектів, який відповідає вимогам запиту.

По масиву отриманих оцінок обирається найкращий проект і з даними цього проекту готується звіт і передається в браузер клієнта.

Алгоритм розрахунку інтегральної оцінки проекту наведено на рис. 3.5.

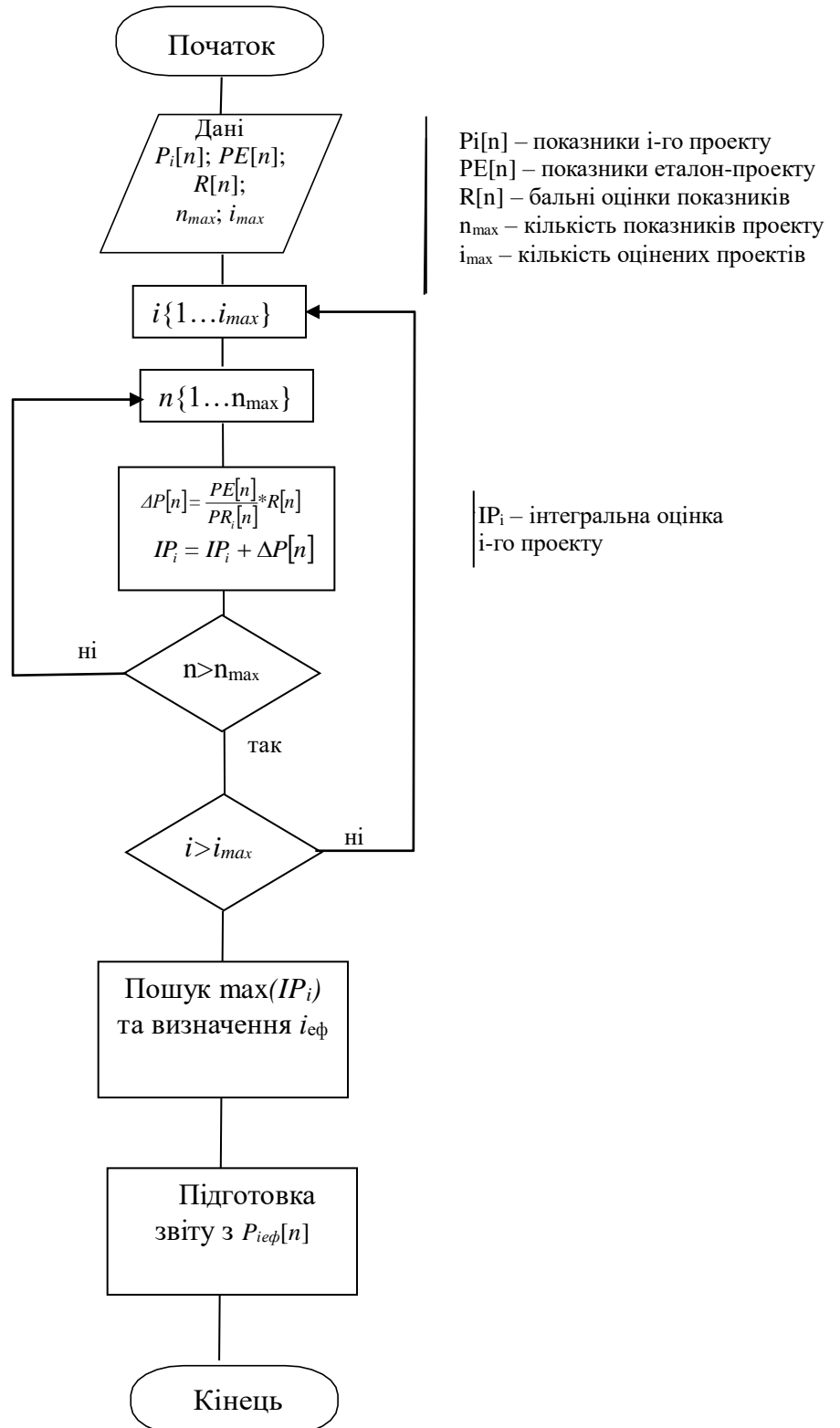




Рисунок 3.5 – Алгоритм розрахунку найбільш ефективного проекту

### ***3.3 Результати тестування системи***

Для перевірки працездатності системи проведено тестування логічної схеми бази даних і роботи форм додатку. Аналіз логічної схеми було проведено розділі 2. Форми тестувались під час їх розробки та під час розробки запитів. Аналіз роботи форм показав, що відхилень немає, вони працюють коректно, всі поля та кнопки правильно виконують свої функції, виведення, оновлення та вставка даних працюють вірно.

Для роботи системи була генерована БД на віддаленому сервері. Скрипт генерації БД наведено у додатку А. Після генерації БД необхідно заповнити всі таблиці тестовими даними.

Для прикладу протестуємо процес реєстрації нового користувача. Для цього в браузері набираємо URL адресу <http://52.149.232.209> для виклику форми авторизації Login form (рис. 3.6).

← → ⓘ 52.149.232.209/login

## ☾ BudExpert Login form

Username:

Password:

Don't have an account? [Register](#)

Рисунок 3.6 – Форма авторизації

Для реєстрації нового користувача нажимаємо кнопку Register. З'являється форма реєстрації нового користувача (див. на рис. 3.7), в якій потрібно заповнити всі вказані поля. Після заповнення всіх даних нажимаємо кнопку Register.

← ⓘ 52.149.232.209/regform

## Registration Form

Check your email for account activation!

← ⓘ 52.149.232.209/regform

## Registration Form

Check your email for account activation!

Рисунок 3.7 – Форма реєстрації нового користувача до і після введення даних

При невірно заповненому полі або при введенні даних, які вже зберігається в базі, з'являється повідомлення про помилку, наприклад «{"message":"Email already exists"}».

Якщо введена вірна на оригінальна інформація, то на указаному мейлі з'являється лист з проханням підтвердження реєстрації на сайті. Реєстрація нового користувача завершена. Головне меню користувача наведено на рис. 3.8.

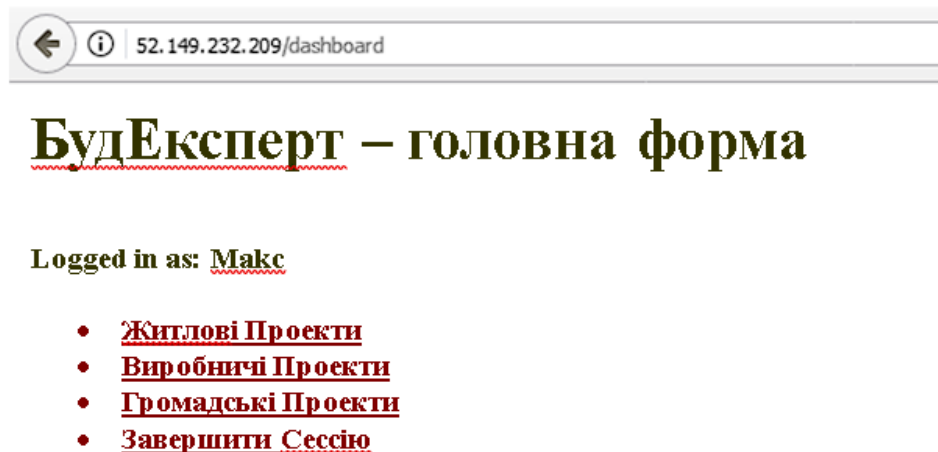


Рисунок 3.8 – Головна форма користувача

З головного меню можна вийти на перегляд реєстру кожного з видів проектів у вигляді таблиці та на редагування характеристик проектів (див на рис. 3.7) (доступно для адміністратора).

З реєстру виробничих проектів кнопкою Edit активуємо форму редагування проекту (рис. 3.9). Вносимо зміни до характеристик проекту «Швейна фабрика» і кнопкою «Save» зберігаємо змінені дані у БД.

Повертаємось до реєстру виробничих проектів. Спостерігаємо змінені дані у проекті «Швейна фабрика».

Найменування виробництва	Швейна фабрика
Характер будівництва	
Потужність об'єкта (натуральних)	100
Потужність об'єкта (вартість)	200
Кількість робочих місць	50
Загальна кількість працюючих	100
Продуктивність праці	500
Собівартість основних продуктів	1000
Кошторисна вартість будівництва	5000
Витрати на охорону навколишнього природного середовища	1000
Термін окупності капітальних вкладень	5
Вартість основних фондів	10000
Вартість основних фондів (бувають)	20000
Доля участі у будівництві	Дольова участь
Тривалість будівництва	2
Трудомісткість будівництва	1000
Річна потреба підприємства у сировинних матеріалах	500
Річна потреба підприємства у енергоресурсах	1000
Річна потреба підприємства у воді	1000
Річна потреба підприємства у зовнішньому транспорті	200
Витрати основних будинкових матеріалів	5000
Інші показники	1000
Прибуток	100000
Рентабельність	0,2
Матеріаломісткість	0,5
Енергомісткість власних потреб	0,3

Рисунок 3.9 – Форма редагування характеристика виробничого проекту

Натискаємо кнопку «Оцінка проекту» у формі оцінки вносимо оціночні бали по всім показникам виробничого проекту (див. рис. 3.10). За наданими балами система на підставі розрахунку інтегральної оцінки знайде найбільш ефективний виробничий проект при натисканні кнопки «Execute».

← | 52.149.232.209/rang\_promobj/3

### Оцінити значущість показників проекту

Потужність об'єкта (натуральних)	2
Потужність об'єкта (вартість)	2
Кількість робочих місць	1
Загальна кількість працюючих	1
Продуктивність праці	3
Собівартість основних продуктів	3
Кошторисна вартість будівництва	5
Витрати на охорону навколишнього природного середовища	5
Термін окупності капітальних вкладень	4
Вартість основних фондів	1
Вартість основних фондів (бувають)	1
Доля участі у будівництві	1
Тривалість будівництва	2
Трудомісткість будівництва	2
Річна потреба підприємства у сировинних матеріалах	3
Річна потреба підприємства у енергоресурсах	3
Річна потреба підприємства у воді	1
Річна потреба підприємства у зовнішньому транспорті	1
Витрати основних будинкових матеріалів	1
Інші показники	0
Прибуток	5
Рентабельність	5
Матеріаломісткість	1
Енергомісткість власних потреб	2
Питома теплотна потужність опалення	3
Питоме річне теплоспоживання	3
Площа території	3

Рисунок 3.10 – Форма внесення рангів значення характеристик проекту для оцінки ефективності проекту

Результатом розрахунків та порівняння інтегральної оцінки для всіх проектів обраного типу буде форма з інформацією по найбільш ефективному за критерієм проекту (рис. 3.11). Такий же пошук можна виконати і для інших груп проектів за відповідними оцінками.

← 52.149.232.209/bestpromobj/4

### Найбільш ефективний проект по інтегральному критерію

Найменування виробництва	Завод мікрочип
Характер будівництва	
Потужність об'єкта (натуральних)	300000
Потужність об'єкта (вартість)	2000000
Кількість робочих місць	50
Загальна кількість працюючих	30
Продуктивність праці	5000
Собівартість основних продуктів	500
Кошторисна вартість будівництва	5000000
Витрати на охорону навколишнього природного середовища	50000
Термін окупності капітальних вкладень	3
Вартість основних фондів	1000000
Вартість основних фондів (бувають)	20000
Доля участі у будівництві	
Тривалість будівництва	1
Трудомісткість будівництва	38000
Річна потреба підприємства у сировинних матеріалах	50000
Річна потреба підприємства у енергоресурсах	10000
Річна потреба підприємства у воді	10000
Річна потреба підприємства у зовнішньому транспорті	20000
Витрати основних будинкових матеріалів	500000
Інші показники	
Прибуток	400000
Рентабельність	0,2

Рисунок 3.11 – Результат оцінки проектів по інтегральному критерію та вибору найкращого

Отже, тестування ПП «BudExpert» показало працездатність сайту і виконання ним поставлених завдань.

## ВИСНОВКИ

Питання вдосконалення методів оцінки ефективності проектів мають важливе значення в зв'язку з великими обсягами будівництва в нашій країні і завданням підвищення якості будівельної продукції при скороченні вартості, трудомісткості та термінів будівництва. Застосування ІТ-технологій в проектуванні стало реальною основою для активної розробки методів оцінки ефективності проектів, спрямоване на підвищення рівня експертизи проектів.

В результаті виконаної роботи було досліджено варіанти оцінки ефективності проектів в будівництві, і на цій основі описано один з методів оцінки ефективності на основі інтегральних оцінок. Було створено програмний засіб, що автоматизує розрахунки та враховує необхідну інформацію. Поставлена мета, яка полягала у спрощенні визначення рівня ефективності будівельних проектів за різними критеріями, було досягнуто.

Створене програмне забезпечення дозволяє робити основні дії з базою даних будівельних проектів. Воно забезпечує користувачам легкий доступ до інформації у зручному вигляді, а адміністратору, крім того, може дозволити додавати та видаляти записи. Реалізовано спеціальні методи, які забезпечують автоматичний розрахунок для вирішення задачі визначення найбільш ефективного будівельного проекту за заданим інтегральним критерієм.

Програма містить дружній інтерфейс, що забезпечує зручність у використанні. Програмний засіб забезпечує надійність у використанні, оскільки передбачає наявність унікальних логінів та пошти для кожного користувача.

Таким чином, було створено продукт, за допомогою якого можна підвищити ефективність прийняття рішень щодо вибору та оцінки проектів в будівництві, що буде корисним для менеджерів, аналітиків та інвесторів. У перспективі можливий розвиток програми, як створення кабінету користувача зі збереженням його критеріальних оцінок, а також додання альтернативних моделей оцінки ефективності проектів.

## ПЕРЕЛІК ПОСИЛАНЬ

- 1.Галузеве звітно-аналітичне програмне забезпечення для контролю КРІ процесів і розвитку вашого бізнесу // <https://waytobi.com/ua/ua-kpi.html>
- 2.Годзь Ю. В., Бикова В. Г. Критеріальний підхід в управлінні ефективністю функціонування підприємства // [http://confcontact.com/2014\\_04\\_25\\_ekonomika\\_i\\_menedgment/tom2/13\\_Godz.htm](http://confcontact.com/2014_04_25_ekonomika_i_menedgment/tom2/13_Godz.htm) (дата звернення: 25.05.2023).
- 3.Казарін О.В. Надійність і безпека програмного забезпечення. – К., 2018.
- 4.Картавих С. М. Моделі та методи оцінки будівельних проектів в умовах композиційної невизначеності /С. М. Картавих, С. А. Теренчук // Управління розвитком складних систем: зб. наук. праць.– К.: КНУБА, 2019.– № 39.– С.75-80
- 5.Кириченко О.А. Інвестування: підручник / О.А. Кириченко, С.А. Єрохін. – К.: Знання, 2009. – 573 с.
- 6.Майорова Т. В. Активізація інвестиційного процесу в Україні: кол. моногр. /Т. В. Майорова, М. І. Дибя, С. В. Онишко / – К.:КНЕУ, 2012.– 473 с.
- 7.Міжнародний стандарт по управлінню проектами ISO 21500:2012
- 8.Нормалізація баз даних [Електронний ресурс]: Режим доступу: [http://wiki.fizmat.tnpu.edu.ua/index.php/Нормалізація\\_баз\\_даних](http://wiki.fizmat.tnpu.edu.ua/index.php/Нормалізація_баз_даних) (дата звернення: 27.05.2023).
- 9.Петухов О.М. Інвестування: навч. посіб. / О.М.Петухов – К.: Центр учбової літератури, 2014. – 336 с.
- 10.Приведені витрати – європейський вектор розвитку // <https://www.google.com/url?esrc=s&q=&rct=j&sa=U&url=https://eurodev.duan.edu.ua/images/PDF/2013/1/20.pdf&ved=2ahUKEwirx-SH5cb-AhXL57sIHUCjC30QFnoECAkQAg&usg=AOvVaw2WFz-3nwz0ISPeGwXMxZrE> (дата звернення: 29.05.2023).
- 11.Сайковська Ю. Ю., Вяльцева І. П. Теоретичні основи ефективності діяльності будівельного виробництва // [http://confcontact.com/2014\\_04\\_25\\_ekonomika\\_i\\_menedgment/tom2/49\\_Saykovska.htm](http://confcontact.com/2014_04_25_ekonomika_i_menedgment/tom2/49_Saykovska.htm) (дата звернення: 25.05.2023).



12.Сидора Т. Ю. Критерії оцінювання ефективності діяльності підприємства /Т. Ю. Сидора // Ефективна економіка: електронне наукове фахове видання. – 2012. – № 2. – Режим доступу: <http://www.economy.nauka.com.ua/?op=1&z=1298> (дата звернення: 01.06.2023).

13.Сучасні клієнт-серверні технології та їх застосування при вивченні систем управління базами даних / Н.Р. Балик; В.І. Мандзюк <http://enpuir.npu.edu.ua/handle/123456789/1338> (дата звернення: 25.05.2023).

14.Чевганова В. Я. Проектний аналіз: навч. посіб. / В. Я. Чевганова, В. В. Биба, А. С. Скрильник – К.: Центр учбової літератури, 2014. – 258 с.

15.Череп А. В. Ефективність як економічна категорія / А. В. Череп, Є. М. Стрілець // Ефективна економіка: електронне наукове фахове видання. – 2013. – № 1. – Режим доступу: <http://www.economy.nauka.com.ua/?op=1&z=1727> (дата звернення: 04.06.2023).

16.Федоренко О. Як вибрати показники, які дійсно важливі для вашого бізнесу // <https://waytobi.com/ua/blog/how-to-choose-kpis-and-metrics-that-matter-to-your-business.html> (дата звернення: 04.06.2023).

17.Ширяєв А.О. Критерії загальної оцінки ефективності інвестиційних проектів // Збірник наукових праць МННЦ ІтіС, 2007, вип.12. – С. 95-102

18.BAS Будівництво. Управління будівельним виробництвом // <https://www.bas-soft.eu/soft/bas-sectoral/bas-construction-building-production-management/> (дата звернення: 28.05.2023).

19.Сервіс розробки карти сайту // <https://www.gloomaps.com/4mjbme7Zhq> (дата звернення: 04.07.2023).

## ДОДАТОК А

## Скрипт генерації бази даних

```
CREATE TABLE Organization (  
    kod_org          int NOT NULL,  
    name            varchar(20) NULL,  
    address         varchar(20) NULL,  
    email           varchar(20) NULL  
)  
go  
  
CREATE UNIQUE INDEX XAK1Organization ON Organization  
(  
    kod_org  
)  
go  
  
ALTER TABLE Organization  
    ADD PRIMARY KEY (kod_org)  
go  
  
CREATE TABLE House_List (  
    Kod              char(18) NOT NULL,  
    kod_orgg         int NOT NULL,  
    S_flat_max       int NULL,  
    Etalon           binary NULL,  
    name            text NULL,  
    Vartist         int NULL,  
    Seredovushe     int NULL,  
    Poverhiv        int NULL,  
    S_zagalna       int NULL,  
    S_flats         int NULL,  
    V_zagalom       int NULL,  
    S_zabudovi      int NULL,  
    S_dilianki      int NULL,  
    K_work_place    int NULL,  
    S_vbudov        int NULL,  
    P_opalenyi      int NULL,  
    Potreba_Teplo  int NULL,  
    Ponreba_paluvo  int NULL,  
    Potreba_electro int NULL,  
    Potreba_voda    int NULL,  
    Spochivaniy_Teplo int NULL,  
    K_flat_one_room int NULL,  
    K_flat_more_rooms int NULL,  
    S_flat_min      int NULL
```

```
)
go
```

```
ALTER TABLE House_List
    ADD PRIMARY KEY (Kod, kod_orgg)
go
```

```
CREATE TABLE Users (
    kod                int NOT NULL,
    name               varchar(20) NULL,
    user_name          int NULL,
    login              varchar(20) NULL,
    password            varchar(20) NULL,
    email              varchar(20) NULL
)
go
```

```
ALTER TABLE Users
    ADD PRIMARY KEY (kod)
go
```

```
CREATE TABLE Social_List (
    Kod                char(18) NOT NULL,
    kod_org            int NOT NULL,
    etalon             binary NULL,
    name               text NULL,
    Vartist            numeric NULL,
    Seredovushe        numeric NULL,
    Poverhiv           numeric NULL,
    S_zagalna          numeric NULL,
    S_kopisna          numeric NULL,
    V_zagalom          numeric NULL,
    S_zabudovi         numeric NULL,
    S_dilianki         numeric NULL,
    K_work_place       numeric NULL,
    P_opaleny          numeric NULL,
    Potreba_Teplo      numeric NULL,
    Ponreba_paluvo     numeric NULL,
    Potreba_electro    numeric NULL,
    Potreba_voda       numeric NULL,
    Spochivaniy_Teplo numeric NULL,
    Potuchnist         numeric NULL,
    Mistkist           numeric NULL,
    Propuskna          numeric NULL
)
go
```

```
CREATE INDEX XIE1Social_List ON Social_List
(
    kod_org,
```

```

        Kod
    )
go

ALTER TABLE Social_List
    ADD PRIMARY KEY (Kod, kod_org)
go

CREATE TABLE Prom_List (
    Kod                char(18) NOT NULL,
    kod_org            int NOT NULL,
    etalon             image NULL,
    name               text NULL,
    Vartist            numeric NULL,
    Seredovushe        numeric NULL,
    S_dilianki         numeric NULL,
    Kol_product        numeric NULL,
    Mistkist           numeric NULL,
    Propuskna          numeric NULL,
    K_work_place       numeric NULL,
    Productivnist      numeric NULL,
    Sobivartist        numeric NULL,
    Termin_okupnosti numeric NULL,
    Vart_osn_fond       numeric NULL,
    Vart_osn_fond_vibuv int NULL,
    Trivalist          int NULL,
    P_opaleny          numeric NULL,
    Potreba_Teplo      numeric NULL,
    Ponreba_paluvo     numeric NULL,
    Potreba_electro    numeric NULL,
    Potreba_voda       numeric NULL,
    Spochivaniy_Teplo  numeric NULL,
    Transtort          numeric NULL,
    Ppributok          int NULL,
    Rentabel           int NULL,
    Materialomict      int NULL,
    Energomist         int NULL
)
go

ALTER TABLE Prom_List
    ADD PRIMARY KEY (Kod, kod_org)
go

ALTER TABLE Social_List
    ADD FOREIGN KEY (kod_org)
        REFERENCES Organization
go

```

```

ALTER TABLE Prom_List
    ADD FOREIGN KEY (kod_org)
                                REFERENCES Organization
go

create trigger tD_Organization on Organization for DELETE as
/* DELETE trigger on Organization */
begin
    declare @errno int,
            @errmsg varchar(255)
            /* Organization R/5 Prom_List ON PARENT DELETE SET NULL */
    update Prom_List
        set
            /* %SetFK(Prom_List,NULL) */
            Prom_List.kod_org = NULL
        from Prom_List,deleted
        where
            /* %JoinFKPK(Prom_List,deleted," = "," and") */
            Prom_List.kod_org = deleted.kod_org

            /* Organization R/4 Social_List ON PARENT DELETE SET NULL */
    update Social_List
        set
            /* %SetFK(Social_List,NULL) */
            Social_List.kod_org = NULL
        from Social_List,deleted
        where
            /* %JoinFKPK(Social_List,deleted," = "," and") */
            Social_List.kod_org = deleted.kod_org
    return
error:
    raiserror @errno @errmsg
    rollback transaction
end
go

create trigger tU_Organization on Organization for UPDATE as
/* UPDATE trigger on Organization */
begin
    declare @numrows int,
            @nullcnt int,
            @validcnt int,
            @inskod_org int,
            @errno int,
            @errmsg varchar(255)

    select @numrows = @@rowcount
    /* Organization R/5 Prom_List ON PARENT UPDATE SET NULL */
    if
        /* %ParentPK(" or",update) */

```

```

    update(kod_org)
begin
    update Prom_List
    set
        /* %SetFK(Prom_List,NULL) */
        Prom_List.kod_org = NULL
    from Prom_List,deleted
    where
        /* %JoinFKPK(Prom_List,deleted," = "," and") */
        Prom_List.kod_org = deleted.kod_org
end

/* Organization R/4 Social_List ON PARENT UPDATE SET NULL */
if
    /* %ParentPK(" or",update) */
    update(kod_org)
begin
    update Social_List
    set
        /* %SetFK(Social_List,NULL) */
        Social_List.kod_org = NULL
    from Social_List,deleted
    where
        /* %JoinFKPK(Social_List,deleted," = "," and") */
        Social_List.kod_org = deleted.kod_org
end

return
error:
    raiserror @errno @errmsg
    rollback transaction
end
go

create trigger tI_Social_List on Social_List for INSERT as
/* INSERT trigger on Social_List */
begin
    declare @numrows int,
            @nullcnt int,
            @validcnt int,
            @errno int,
            @errmsg varchar(255)

    select @numrows = @@rowcount
    /* Organization R/4 Social_List ON CHILD INSERT SET NULL */
    if
        /* %ChildFK(" or",update) */
        update(kod_org)
    begin
        update Social_List
        set

```

```

        /* %SetFK(Social_List,NULL) */
        Social_List.kod_org = NULL
    from Social_List,inserted
    where
        /* %JoinPKPK(Social_List,inserted," = "," and") */
        Social_List.Kod = inserted.Kod and
        Social_List.kod_org = inserted.kod_org and
        not exists (
            select * from Organization
            where
                /* %JoinFKPK(inserted,Organization," = "," and") */
                inserted.kod_org = Organization.kod_org
        )
    end

    return
error:
    raiserror @errno @errmsg
    rollback transaction
end
go

create trigger tU_Social_List on Social_List for UPDATE as
/* UPDATE trigger on Social_List */
begin
    declare @numrows int,
            @nullcnt int,
            @validcnt int,
            @insKod char(18),
            @inskod_org int,
            @errno int,
            @errmsg varchar(255)

    select @numrows = @@rowcount
    /* Organization R/4 Social_List ON CHILD UPDATE SET NULL */
    if
        /* %ChildFK(" or",update) */
        update(kod_org)
    begin
        update Social_List
        set
            /* %SetFK(Social_List,NULL) */
            Social_List.kod_org = NULL
        from Social_List,inserted
        where
            /* %JoinPKPK(Social_List,inserted," = "," and") */
            Social_List.Kod = inserted.Kod and
            Social_List.kod_org = inserted.kod_org and
            not exists (
                select * from Organization
                where

```

```

        /* %JoinFKPK(inserted,Organization," = "," and") */
        inserted.kod_org = Organization.kod_org
    )
end

return
error:
    raiserror @errno @errmsg
    rollback transaction
end
go

create trigger tI_Prom_List on Prom_List for INSERT as
/* INSERT trigger on Prom_List */
begin
    declare @numrows int,
            @nullcnt int,
            @validcnt int,
            @errno int,
            @errmsg varchar(255)

    select @numrows = @@rowcount
    /* Organization R/5 Prom_List ON CHILD INSERT SET NULL */
    if
        /* %ChildFK(" or",update) */
        update(kod_org)
    begin
        update Prom_List
        set
            /* %SetFK(Prom_List,NULL) */
            Prom_List.kod_org = NULL
        from Prom_List,inserted
        where
            /* %JoinPKPK(Prom_List,inserted," = "," and") */
            Prom_List.Kod = inserted.Kod and
            Prom_List.kod_org = inserted.kod_org and
            not exists (
                select * from Organization
                where
                    /* %JoinFKPK(inserted,Organization," = "," and") */
                    inserted.kod_org = Organization.kod_org
            )
    end

return
error:
    raiserror @errno @errmsg
    rollback transaction
end
go

```



```

create trigger tU_Prom_List on Prom_List for UPDATE as
/* UPDATE trigger on Prom_List */
begin
    declare @numrows int,
            @nullcnt int,
            @validcnt int,
            @insKod char(18),
            @inskod_org int,
            @errno int,
            @errmsg varchar(255)

    select @numrows = @@rowcount
    /* Organization R/5 Prom_List ON CHILD UPDATE SET NULL */
    if
        /* %ChildFK(" or",update) */
        update(kod_org)
    begin
        update Prom_List
        set
            /* %SetFK(Prom_List,NULL) */
            Prom_List.kod_org = NULL
        from Prom_List,inserted
        where
            /* %JoinPKPK(Prom_List,inserted," = "," and") */
            Prom_List.Kod = inserted.Kod and
            Prom_List.kod_org = inserted.kod_org and
            not exists (
                select * from Organization
                where
                    /* %JoinFKPK(inserted,Organization," = "," and") */
                    inserted.kod_org = Organization.kod_org
            )
    end

    return
error:
    raiserror @errno @errmsg
    rollback transaction
end
go

```

## ДОДАТОК Б

### Фрагменти програмного коду серверної частини сайту

```

#om flask import Flask, request, jsonify, render_template
#m flask import session, redirect, url_for
from flask_mail import Mail, Message
import bcrypt
import secrets
import sqlite3
from threading import Thread
from flask_wtf import FlaskForm
import smtplib
from email.mime.text import MIMEText
#ort sqlalchemy
import os
import subprocess
from flask import Flask, render_template, request, redirect, url_for,
session, jsonify
#ort stroydbfunc
#from stroydbfunc import ObjectForm, PrombudForm # Import the ObjectForm
class from your forms module
from wtforms import HiddenField, FloatField, StringField, TextAreaField,
IntegerField, DecimalField
from wtforms.validators import DataRequired
#port sqlite3
import psychopg2
from wtforms.validators import InputRequired, NumberRange
from wtforms import Form
from flask_wtf.csrf import CSRFProtect, CSRFError

app = Flask(__name__)

#csrf= CSRFProtect(app)

app.config['SECRET_KEY'] = '2e3e4a5a4748593748586c'
app.config['WTF_CSRF_SECRET_KEY'] = '2e3e4485858584848f5f3a2a'

env = os.environ.get('FLASK_ENV', 'dev')
app.config.update(dict( SECRET_KEY="e3e32e4a1ae",
                        WTF_CSRF_SECRET_KEY="2e3a4a6e",
                        WTF_CSRF_ENABLED=False))
#app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///stroy.db'
#db = SQLAlchemy(app)
#csrf = CSRFProtect(app)
#csrf.init_app(app)

```

```

#csrf.init_app(app)
#csrf= CSRFProtect(app)
app.config['MAIL_SERVER'] = 'smtp.gmail.com' # Настройте SMTP-сервер для
отправки писем
app.config['MAIL_PORT'] = 587
app.config['MAIL_USE_TLS'] = True
app.config['MAIL_USERNAME'] = 'oleg.uzhakin@gmail.com'
app.config['MAIL_PASSWORD'] = 'itb108Q_'
password='hbkyehqanappjysx'
sender='oleg.uzhakin@gmail.com'

mail = Mail(app)

def send_email(subject, body, sender, recipients, password):
    msg = MIMEText(body)
    msg['Subject'] = subject
    msg['From'] = sender
    msg['To'] = ', '.join(recipients)
    print(sender)
    print(password)
    with smtplib.SMTP_SSL('smtp.gmail.com', 465) as smtp_server:
        smtp_server.login(sender, password)
        smtp_server.sendmail(sender, recipients, msg.as_string())
    print("Message sent!")

#app.route('/printrouts123')

@app.route('/checkdb')
def check_db():
    import sqlite3

    try:
        # Подключение к базе данных
        conn = sqlite3.connect('stroy.db')

        # Создание курсора
        cursor = conn.cursor()

        # Пример выполнения запроса
        cursor.execute('SELECT * FROM users')
        result = cursor.fetchall()

        # Вывод результатов
        for row in result:
            print(row)

        # Закрытие курсора и соединения с базой данных
        cursor.close()
        conn.close()

```

```

except sqlite3.Error as e:
    print(f"An error occurred: {e}")

check_db()

def create_admin_user():
    # Подключение к базе данных
    conn = sqlite3.connect('stroy.db')
    cursor = conn.cursor()

    # Проверяем, есть ли записи в таблице users
    cursor.execute('SELECT COUNT(*) FROM users')
    result = cursor.fetchone()

    # Если таблица пуста, добавляем пользователя admin
    if result[0] == 0:
        name = 'admin'
        fio = 'admin'
        username = 'admin'
        email = 'admin@example.com'
        password = 'admin'

        # Хешируем пароль
        hashed_password = bcrypt.hashpw(password.encode('utf-8'),
bcrypt.gensalt())

        # Вставляем данные пользователя в таблицу
        cursor.execute('INSERT INTO users (name, fio, username, email,
password, id) VALUES (?, ?, ?, ?, ?, ?)',
            (name, fio, username, email, hashed_password, 1))
        conn.commit()

        print("Admin user added successfully.")

    # Закрываем курсор и соединение с базой данных
    cursor.close()
    conn.close()

# Вызываем функцию для создания пользователя admin, если таблица users
пуста
create_admin_user()

try:
    subprocess.run(['python', 'ssdbcheck.py'])
    return 'Database check completed successfully'
except Exception as e:
    return f'Error occurred during database check: {str(e)}'

@app.route('/sslogin', methods=['GET', 'POST'])

```

```

def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']

        conn = sqlite3.connect('stroy.db')
        cursor = conn.cursor()

        cursor.execute('SELECT * FROM users WHERE username = ?',
            (username,))
        user = cursor.fetchone()
        hpw=password #bcrypt.hashpw(password.encode('utf-8'),
bcrypt.gensalt())

        cursor.close()
        conn.close()
        print( hpw)
        print (user[5])
        if user:
            if hpw== user[5]:
                session['user_id'] = user[0]
                session['username'] = user[3]
                return redirect(url_for('dashboard'))

        return jsonify({'message': 'Invalid username or password'})

    return render_template('login_form.html')

@app.route('/ssregform', methods=['GET'])
def registration_form():
    return render_template('registration_form.html')

@app.route('/ssregister', methods=['POST'])
def register():
    name = request.form['name']
    fio = request.form['fio']
    username = request.form['username']
    email = request.form['email']
    password = request.form['password']

    # Проверяем, что все поля заполнены
    if not name or not fio or not username or not email or not password:
        return jsonify({'message': 'All fields are required'})

    # Проверяем, существует ли пользователь с таким же логином или email
    if user_exists(username):
        return jsonify({'message': 'Username already exists'})
    if email_exists(email):
        return jsonify({'message': 'Email already exists'})

```

```

# Хешируем пароль
hashed_password = password# bcrypt.hashpw(password.encode('utf-8'),
bcrypt.gensalt())

# Создаем уникальный токен активации
activation_token = secrets.token_urlsafe(16)

# Сохраняем данные пользователя в базе данных
save_user(name, fio, username, email, hashed_password,
activation_token)

# Отправляем письмо с ссылкой на активацию аккаунта
activation_link =
f'http://52.149.231.209/activate?token={activation_token}'
send_activation_email(email, activation_link)

return jsonify({'message': 'Registration successful'})

def save_user(name, fio, username, email, password, activation_token):
    conn = sqlite3.connect('stroy.db')
    cursor = conn.cursor()

    try:
        cursor.execute('INSERT INTO users (name, fio, username, email,
password, activation_token) VALUES (?, ?, ?, ?, ?, ?)',
                        (name, fio, username, email, password,
activation_token))
        conn.commit()
        return True
    except sqlite3.Error as e:
        print("Error saving user to database:", e)
        return False
    finally:
        cursor.close()
        conn.close()

```