

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕКОНОМІКИ І МЕНЕДЖМЕНТУ
Кафедра кібернетики та інформатики

КВАЛІФІКАЦІЙНА РОБОТА

освітній ступінь – «Бакалавр»

на тему:

**«Розробка вебзастосунку аналізу
тарифів мобільних операторів УКРАЇНИ»**

Виконав:

студент спеціальності
126 «Інформаційні системи та технології»
ОП «Інформаційні системи та технології»

Федірко Данило Дмитрович

Науковий керівник:

Толбатов Андрій Володимирович
доцент кафедри кібернетики та інформатики

Рецензент:

Гриценко Костянтин Григорович
доцент кафедри економічної кібернетики ННІ бізнесу,
економіки та менеджменту Сумського державного університету,
к.т.н., доцент

Суми – 2024

СУМСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ

Факультет	Економіки і менеджменту
Кафедра	Кібернетики та інформатики
Освітній ступінь	«Бакалавр»
Спеціальність	126 «Інформаційні системи та технології» ОП «Інформаційні системи та технології»

Затверджую:
Завідувач кафедри _____ Світлана АГАДЖАНОВА

« _____ » _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Федірку Данилу Дмитровичу

1. Тема роботи: “Розробка вебзастосунку аналізу тарифів мобільних операторів України”

2. База практичного дослідження: Сумський національний аграрний університет м. Суми.

керівник роботи: Толбатов Андрій Володимирович, к.т.н., доцент

затверджена наказом по університету від « _____ » _____

3. Строк подання студентом закінченого проєкту
(роботи) _____ «28» червня 2024 року

4. Вихідні дані до проєкту
(роботи):
результати аналітичних досліджень зарубіжних та вітчизняних вчених, інтернет-джерела,

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

- виконати постановку задачі;
- вибрати середовище для реалізації;
- розробити технічне завдання;
- розробити програмний продукт відповідно до поставлених задач;

6. Дата видачі завдання: _____ « _____ » _____ 2024р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів проєкту	Примітка
1	Визначення об'єкту, предмету дослідження, формулювання мети та задач кваліфікаційної роботи, складання плану	січень 2024 р.	виконано
2	Підбір та вивчення літературних джерел, технічної документації	січень 2024 р.	виконано
3	Розробка технічного завдання	січень 2024 р.	виконано
4	Розробка вебзастосунку	січень-травень 2024 р.	виконано
5	Оформлення пояснювальної записки	до 30 травня 2024 р.	виконано
6	Оформлення презентації	до 1 червня 2024 р.	виконано
7	Опрацювання зауважень керівника, перевірка на автентичність	2 червня 2024 р.	виконано
8	Підготовка доповіді та її попередній захист	6 червня 2024 р.	виконано
9	Перевірка з академічної доброчесності кваліфікаційної роботи	10 червня 2024 р.	виконано
10	Представлення кваліфікаційної роботи на кафедрі, деканат	23 червня 2024 р.	виконано
11	Захист кваліфікаційної роботи	28 червня 2024р.	виконано
Студент		(підпис)	Федірко Данило

Керівник роботи

(підпис)

Андрій Толбатов

Перевірку на автентичність проведено

(підпис)

Надія БАРАНІК

Кваліфікаційна робота допущена до захисту

(підпис)

Маргарита ЛИШЕНКО

АНОТАЦІЯ

Федірко Д.Д. Розробка вебзастосунку аналізу тарифів мобільних операторів України.

Кваліфікаційна робота зі спеціальності 126 «Інформаційні системи та технології» ОП Інформаційні системи та технології, СНАУ, Суми-2024р. – Рукопис.

Ця робота присвячена створенню веб-додатку для аналізу тарифів мобільних операторів України. Основною метою є розробка зручного та ефективного інструменту для користувачів, що дозволить порівнювати та аналізувати тарифні плани різних операторів для вибору оптимального варіанту.

У дослідженні проведено огляд існуючих платформ для порівняння тарифів мобільних операторів, визначено ключові проблеми найпопулярніших платформ та запропоновано архітектуру веб-додатку. Визначено мету та засоби реалізації.

Для реалізації веб-додатку використовувались сучасні технології програмування та баз даних, що забезпечують його швидке та надійне функціонування. Розроблено модулі для введення даних, виконання розрахунків, створення звітів та аналізу результатів. Проведено автоматизоване тестування веб-додатку.

Результати тестування підтвердили коректність функціонування веб-додатку та його відповідність вимогам і очікуванням користувачів. Отримані результати дозволяють зробити висновок про доцільність впровадження розробленого веб-додатку для аналізу тарифів мобільних операторів України. Він сприятиме покращенню якості обслуговування користувачів, а також забезпечить зручний доступ до необхідної інформації для прийняття обґрунтованих рішень щодо вибору оптимального тарифного плану.

Ключові слова: веб-додаток, мобільні оператори, тарифи, аналіз, порівняння, Python, Flask, JSON.

SUMMARY

Fefirko D.D. Development of a Web Application for Analyzing Mobile Operator Tariffs in Ukraine.

Qualification Work in the specialty 126 "Information Systems and Technologies," Educational Program "Information Systems and Technologies," Sumy National Agrarian University, Sumy-2024. – Manuscript.

This work is dedicated to the creation of a web application for analyzing mobile operator tariffs in Ukraine. The main objective is to develop a convenient and efficient tool for users that allows them to compare and analyze tariff plans of different operators to choose the optimal option.

The study provides a review of existing platforms for comparing mobile operator tariffs, identifies key issues of the most popular platforms, and proposes the architecture of the web application. The goal and means of implementation are defined.

Modern programming and database technologies were used to implement the web application, ensuring its fast and reliable operation. Modules for data entry, calculations, report generation, and results analysis have been developed. Automated testing of the web application was conducted.

The testing results confirmed the correctness of the web application's functionality and its compliance with the users' requirements and expectations. The obtained results allow concluding the feasibility of implementing the developed web application for analyzing mobile operator tariffs in Ukraine. It will contribute to improving the quality of user service and provide convenient access to the necessary information for making informed decisions regarding the choice of the optimal tariff plan.

Keywords: web application, mobile operators, tariffs, analysis, comparison, Python, Flask, JSON.

ЗМІСТ

Вступ.....	7
1. Аналіз предметної області.....	9
1.1 Огляд останніх досліджень і публікацій.....	9
1.2 Аналіз програмних продуктів-аналогів.....	17
1.3 Постановка задачі.....	19
2. Моделювання та проектування.....	21
2.1 Моделювання.....	21
2.2 Проектування інформаційної системи.....	23
2.3 Проектування моделі бази даних	25
3. Розробка веб-додатка для аналізу тарифів мобільних операторів України.....	31
3.1 Архітектура програмного додатку.....	31
3.2 Програмна реалізація.....	36
3.3 Використання програмного додатку	38
Висновки.....	41
Список використаних джерел	42
Додаток А. Технічне завдання	44
Додаток Б. Планування робіт.....	51
Додаток В. Лістинг програмного коду.....	54

ВСТУП

У сучасному світі мобільний зв'язок є однією з ключових складових інформаційного простору, що значно полегшує спілкування та забезпечує доступ до різноманітних сервісів. Швидкий розвиток технологій та постійна конкуренція між мобільними операторами призводить до появи великої кількості тарифних планів та пропозицій, що часто ускладнює вибір оптимального варіанту для користувачів. У цьому контексті *актуальним* стає розробка веб-додатка для аналізу тарифів мобільних операторів України.

Метою дослідження є розробка веб-додатка для аналізу тарифів мобільних операторів України, який надасть користувачам зручний і ефективний інструмент для порівняння та аналізу тарифних планів, запропонованих різними мобільними операторами.

Завдання дослідження:

- проаналізувати існуючі платформи для аналізу тарифів мобільних операторів, визначити їх переваги та недоліки;
- спроектувати архітектуру веб-додатка для аналізу тарифів мобільних операторів України;
- спроектувати базу даних веб-додатка для зберігання інформації про тарифи мобільних операторів;
- реалізувати архітектуру веб-додатка, застосовуючи сучасні технології програмування;
- розробити прототип інтерфейсу користувача;
- фізично реалізувати базу даних веб-додатка;
- протестувати функціональність створеного веб-додатка;
- оцінити результати розробки та надати рекомендації щодо її подальшого удосконалення.

Об'єктом дослідження є процес розробки веб-додатка для аналізу тарифів мобільних операторів України.

Предметом дослідження є інформаційні технології розробки веб-додатка

для аналізу тарифів мобільних операторів України.

Наукова новизна полягає у розробці веб-додатка, який забезпечить ефективний аналіз тарифів мобільних операторів, враховуючи сучасні вимоги користувачів і технологічні тенденції. Розробка ініційована з причини відсутності певних рішень на ринку, тому є унікальною та науково цінною з технічної точки зору. Запропоноване рішення орієнтовано на зручність використання, інтуїтивно зрозумілий інтерфейс, адаптивний дизайн та можливість інтеграції з іншими сервісами.

Практичне значення одержаних результатів полягає у можливості використання веб-додатка для аналізу тарифів у різних сферах мобільного зв'язку, включаючи професійні та особисті потреби.

У дослідженні застосовані такі методи:

- метод експерименту, що відноситься до емпіричних методів і застосовувався під час тестування веб-додатка;
- метод моделювання, що є комплексним методом наукового дослідження і використовувався для створення проєкту веб-додатка та його бази даних;
- метод абстрагування, що також відноситься до комплексних методів і застосовувався при аналізі існуючих на ринку веб-додатків;
- метод сходження від абстрактного до конкретного, застосований при аналізі технологій, що були використані у процесі розробки веб-додатка. Цей метод наукового дослідження відноситься до теоретичних.

Структура і обсяг роботи. Кваліфікаційна робота складається із вступу, трьох розділів, висновків та пропозицій, списку використаних джерел з 13 найменувань, додатків. Основний текст викладений на 71 сторінці комп'ютерного тексту, робота містить 16 рисунків, 4 додатки.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень та публікацій

Україна, як країна з високим рівнем розвитку телекомунікаційного сектору, характеризується динамічним ринком мобільних операторів, які пропонують широкий спектр тарифних планів та послуг. Провідні гравці ринку, такі як Kyivstar, Vodafone (раніше MTC) та lifecell, активно конкурують як між собою, так і з місцевими операторами, що створює сприятливі умови для розвитку та впровадження інновацій.

Мобільні оператори в Україні пропонують широкий спектр тарифних планів, які відрізняються за вартістю, обсягом даних, кількістю хвилин та SMS. Існують як мінімальні тарифи для легкого користування, так і розширені плани з необмеженими послугами, що дозволяє користувачам вибирати оптимальний варіант відповідно до їхніх потреб. Окрім основних послуг, оператори також надають додаткові сервіси, такі як роумінг, пакети міжнародних дзвінків, доступ до контенту та інші послуги, що підвищують зручність та функціональність користування мобільним зв'язком [1].

Користувачі мобільних послуг в Україні активно шукають оптимальні тарифи, які задовольняють їхні потреби у зв'язку та доступі до Інтернету за вигідною ціною. Зростаюче використання мобільного Інтернету та соціальних мереж вимагає наявності тарифів з великим обсягом даних. Крім того, користувачі віддають перевагу тарифам з прозорим ціноутворенням та високоякісним обслуговуванням, що дозволяє їм уникати прихованих платежів і непередбачуваних витрат.

Конкурентна боротьба між операторами стимулює впровадження нових технологій та інноваційних послуг. Нові тарифні плани, акції та промо-кампанії стають невід'ємною частиною стратегії кожного оператора для залучення та

утримання клієнтів. Завдяки цьому користувачі отримують більше можливостей для вибору та персоналізації своїх тарифних планів. Це включає можливість підбору тарифу відповідно до індивідуальних вимог щодо обсягу даних, кількості хвилин та SMS, а також доступу до додаткових послуг.

Регулювання телекомунікаційного сектору відіграє важливу роль у забезпеченні чесної конкуренції та захисту прав споживачів. Впровадження нових правил та стандартів може вплинути на стратегії ціноутворення та надання послуг, однак при належному балансі це сприяє розвитку ринку та підвищенню якості послуг для користувачів. Зокрема, регулювання може забезпечити прозорість умов надання послуг, захист прав споживачів та стимулювати операторів до підвищення рівня обслуговування.

Ринок мобільних тарифів в Україні є жвавим та динамічним, з численними можливостями для користувачів та операторів. Постійна конкуренція та інновації сприяють розвитку якісних та доступних послуг. Зростаючі потреби користувачів вимагають від операторів гнучкості та відповідності їхнім вимогам. Регулювання галузі є важливим для забезпечення рівних умов конкуренції та захисту прав споживачів. З урахуванням цих факторів мобільні оператори можуть досягти успіху на ринку та задовольнити потреби своїх клієнтів.

Мобільні оператори в Україні не лише конкурують за клієнтів за допомогою різноманітних тарифів, а й пропонують додаткові послуги та акції, щоб привернути увагу та задовольнити потреби споживачів. Порівняння послуг та акцій відображає широкий спектр можливостей, доступних для користувачів мобільного зв'язку. Інтернет-трафік, пропонований кожним оператором, варіюється від обмежених до необмежених планів, причому важливою є також швидкість доступу до мережі, яка може різнитися між операторами. Тарифні плани включають різні кількості хвилин для внутрішньої мережі, дзвінків на інші мережі та кількість безкоштовних SMS, з можливістю необмежених планів для дзвінків та повідомлень. Умови для роумінгу та міжнародних дзвінків також можуть різнитися, зокрема, деякі оператори пропонують спеціальні пакети для подорожуючих або знижки на міжнародні дзвінки. Додаткові послуги, такі як мобільне телебачення, стримінгові сервіси та

мобільні платежі, також можуть бути доступні, що підвищує функціональність та привабливість пропозицій операторів [2].

Акції та спеціальні пропозиції включають знижки на тарифні плани, бонуси або додатковий обсяг даних за поповнення рахунку, подарункові карти або призи серед клієнтів та знижки або додаткові послуги для вірних клієнтів. Це створює конкурентне середовище, яке стимулює операторів до постійного удосконалення та розробки нових пропозицій для задоволення потреб своїх клієнтів.

Публікації, що розглядають питання розробки веб-додатків та вибору між мобільним додатком та веб-сайтом, мають велике значення для професіоналів у галузі веб-розробки та для тих, хто цікавиться цією сферою.

Стаття «Життєвий цикл розробки вебзастосунку: від планування до обслуговування» за авторством Іщенка М. Д. та Білоуса М. Ю. [3] пропонує детальний огляд етапів розробки вебзастосунків, починаючи від планування та завершуючи обслуговуванням. Автори розглядають кожен етап життєвого циклу, надаючи корисні поради та вказівки щодо ефективної реалізації кожного з них.

Стаття починається з визначення мети та значення життєвого циклу вебзастосунку. Вона розглядає сучасні тенденції веброзробки та вказує на важливість правильного розуміння вимог клієнта на першому етапі. Автори підкреслюють, що успішне планування є фундаментом для подальших етапів розробки, оскільки воно дозволяє визначити основні цілі та завдання проекту, обрати правильні інструменти та методи для їх досягнення.

Подальші розділи статті охоплюють планування проекту, структурне моделювання та вибір технічного стеку. Особлива увага приділяється важливості створення карти та каркасу сайту, що є ключовими етапами у визначенні архітектури та функціональності майбутнього вебзастосунку. Автори наголошують на необхідності ретельного опрацювання цих етапів, оскільки це дозволяє уникнути багатьох проблем у процесі розробки та забезпечити високу якість кінцевого продукту.

Автори також розглядають питання дизайну та його взаємодії з користувачем. Вони підкреслюють, що якісний дизайн не лише покращує

зовнішній вигляд вебзастосунку, але й сприяє зручності його використання, що є важливим фактором для залучення та утримання користувачів. У статті наводяться рекомендації щодо вибору кольорової гами, типографіки, розміщення елементів інтерфейсу та інших аспектів, що впливають на зручність взаємодії з користувачем.

Після розгляду етапів планування та розробки, стаття переходить до тестування та розгортання вебзастосунку. Вона розглядає різноманітні методи тестування, від прийняття користувачами до функціонального тестування та тестування продуктивності. Автори підкреслюють важливість проведення всебічного тестування для виявлення та усунення можливих помилок до розгортання вебзастосунку. Особлива увага приділяється регресійному тестуванню, яке дозволяє переконатися, що нові функції або виправлення помилок не впливають на роботу вже існуючих функцій.

Також наголошується на важливості обслуговування програмного забезпечення після його розгортання. Автори вказують на необхідність регулярного оновлення вебзастосунку, впровадження нових функцій та виправлення помилок для підтримки його високої якості та відповідності потребам користувачів. Вони наводять приклади кращих практик у сфері обслуговування програмного забезпечення, що дозволяють забезпечити стабільну та безперебійну роботу вебзастосунку.

Стаття надає чіткий та докладний огляд життєвого циклу розробки вебзастосунку, надаючи корисні поради та рекомендації для кожного етапу. Вона є цінним джерелом інформації для професіоналів у сфері веброзробки та тих, хто планує вивчити цю галузь. Стаття охоплює всі ключові аспекти розробки вебзастосунків, від планування та проектування до тестування та обслуговування, що робить її корисною як для початківців, так і для досвідчених розробників.

Стаття «Створення додатку чи веб-сайту: У чому ключова різниця?» [4] порушує важливе питання вибору між створенням мобільного додатку та веб-сайту для досягнення бізнес-цілей. Автор висвітлює різницю між цими двома варіантами та допомагає читачам зрозуміти, які можливості та функції притаманні кожному з них. У статті розглядаються ключові аспекти, такі як функціональність мобільних

додатків та веб-сайтів, вибір відповідно до потреб бізнесу та розуміння відмінностей між ними.

Автор пояснює, що мобільні додатки орієнтовані на персоналізований та інтерактивний досвід користувачів завдяки використанню різних функцій, таких як push-повідомлення та доступ до апаратного забезпечення пристрою. Мобільні додатки дозволяють користувачам отримувати миттєві повідомлення, використовувати геолокацію, камеру, мікрофон та інші функції пристрою, що значно підвищує їхню залученість та зручність у використанні. Крім того, мобільні додатки зазвичай працюють швидше і можуть функціонувати в режимі офлайн, що робить їх незамінними для ряду бізнес-задач, таких як мобільний банкінг, електронна комерція, соціальні мережі та інші сервіси, які вимагають високого рівня інтерактивності та персоналізації.

З іншого боку, веб-сайти частіше використовуються для поширення інформації серед ширшої аудиторії. Веб-сайти забезпечують легкий доступ до інформації з будь-якого пристрою з підключенням до Інтернету і не вимагають встановлення додаткового програмного забезпечення. Вони ідеально підходять для бізнесів, які орієнтовані на надання інформації, публікацію контенту, підтримку клієнтів та проведення маркетингових кампаній. Веб-сайти також простіші в оновленні та підтримці, оскільки зміни можуть бути внесені централізовано, без необхідності оновлення на кожному окремому пристрої.

Автор наголошує на важливості розуміння власних потреб та цілей бізнесу для вибору найбільш підходящого варіанту - мобільного додатку або веб-сайту. Вибір між ними залежить від характеру бізнесу, цільової аудиторії та специфічних вимог до функціональності. Мобільні додатки можуть бути кращим вибором для бізнесів, які потребують високого рівня інтерактивності та персоналізації, тоді як веб-сайти підходять для тих, хто хоче забезпечити широке поширення інформації та легкий доступ до своїх послуг.

Через розуміння відмінностей у функціональності та призначенні мобільних додатків і веб-сайтів, читачі можуть краще визначити, який варіант відповідає їхнім потребам. Стаття допомагає читачам усвідомити, що вибір між мобільним

додатком та веб-сайтом не є взаємовиключним і в багатьох випадках може бути доцільним поєднання обох рішень для досягнення максимального ефекту.

Вебзастосунок є інтерактивним застосуванням, яке складається з двох основних частин: перша частина завантажується у браузері, коли користувач відкриває відповідну сторінку, друга частина розміщена на веб-сервері і містить усі дані, необхідні для роботи застосунку, отримуючи запити від користувачів, знаходячи та відправляючи інформацію у браузер. Для більшості користувачів вебзастосунки ззовні майже не відрізняються від звичайних веб-сайтів, але мають значно більше вбудованих функцій, таких як робота з графікою, таблицями та оформлення замовлень онлайн.

Вебзастосунки мають ряд переваг: вони не потребують встановлення на пристрій користувача, користувач не має турбуватися про оновлення програми, оскільки вони відбуваються автоматично і централізовано, усі користувачі отримують доступ до однієї версії вебзастосунку, що уникає непорозумінь. Вебзастосунок розробляється з урахуванням можливості його відкриття з будь-якого браузера та операційної системи. Вебзастосунки зменшують витрати як для бізнесу, так і для користувачів, оскільки потребують менше ресурсів для роботи та обслуговування. Вони прості у використанні і не потребують спеціальних знань або навичок.

Вебзастосунки створюються з різними цілями, включаючи взаємодію з клієнтами та налагодження бізнес-процесів всередині компанії. Ці інструменти мають безліч можливостей, включаючи обмін інформацією між співробітниками та клієнтами, проведення транзакцій, керування файлами, роботу з нотатками та списками справ тощо.

Вебзастосунок відрізняється від веб-сайту складнішою структурою та необхідністю компіляції для запуску. Він має більше функціональних можливостей для інтеракції з користувачем. Ці фактори роблять вебзастосунки ефективними інструментами для різних видів бізнесу та комунікації з аудиторією [5].

Одиничні вебзастосунки (Single Page Applications, SPA) є типом вебзастосунків, які завантажують лише одну сторінку HTML і автоматично

оновлюються через JavaScript, не потребуючи повторного завантаження сторінки. Наприклад, коли приходить новий лист у пошту на Gmail, сама сторінка не перезавантажується, а лише підвантажується потрібний елемент. Цей процес відбувається швидко і без перерв. SPA забезпечують високу швидкість завантаження, оскільки всі елементи відразу завантажуються, а зміни на сторінці відбуваються миттєво. Більшість роботи відбувається в браузері без необхідності звертання до сервера, що робить їх надзвичайно ефективними за часом. Вони також можуть швидко кешувати дані, відправляючи лише один запит, після чого можуть працювати офлайн.

Водночас вебзастосунки потребують постійного з'єднання з Інтернетом, оскільки без нього використовувати додаток майже неможливо. Вебзастосунки можуть призвести до додаткового навантаження на браузер, особливо якщо сторінка складна, що може спричинити зависання на старих або слабких пристроях. Крім того, вони вимагають підтримки JavaScript браузером і можуть мати проблеми з SEO, оскільки пошукові системи не завжди здатні індексувати всі модулі.

Мультисторінковий вебзастосунок (Multi Page Application, MPA) є багатосторінковим застосунком, який вимагає постійного звертання до веб-сервера при взаємодії з ним. Кожна дія призводить до оновлення сторінки, що робить їх менш ефективними за швидкістю порівняно з SPA. MPA ідеально підходять для інтернет-магазинів з великим асортиментом товарів, оскільки сторінки можна легко оптимізувати під ключові запити і використовувати у браузерах, що не підтримують JavaScript. Проте, MPA є складнішими, важчими та менш масштабованими за SPA, що відповідно вимагає більше часу і коштів для їх розробки.

Прогресивні вебзастосунки (Progressive Web Applications, PWA) можна розглядати як гібрид між веб-сторінкою і мобільним додатком. Вони можуть бути запуснені через браузер на десктопі або смартфоні, або ж встановлені на робочий екран пристрою, що дозволяє їм працювати автономно. PWA мають багато функцій, характерних для нативних мобільних додатків, а візуально вони мало відрізняються від мобільних версій веб-сайтів. PWA працюють на різних

операційних системах на смартфонах, ПК та терміналах у торгових залах, швидко встановлюються та працюють, не потребуючи створення окремого веб-сайту. Оновлення відбуваються автоматично, і немає потреби звертатися до магазину додатків для встановлення. Однак, деякі браузері погано підтримують технологію PWA, і через обмеження iOS розробники часто не мають доступу до всіх нативних функцій пристроїв Apple.

Додаток, який розробляється, представляє собою SPA (Single Page Application), який забезпечує можливість коригування інформації адміністратором після авторизації, а також відображення цієї інформації клієнтам без авторизації. У контексті сучасного технологічного ландшафту, де мобільні телефони стали невід'ємною частиною повсякденного життя, розробка додатків для аналізу тарифів мобільних операторів України є нагальною потребою. Споживачі активно користуються мобільними послугами, і доступ до інформації про тарифи стає ключовим для їхнього раціонального вибору. Оскільки на сучасному ринку існує велика кількість мобільних операторів, кожен із яких пропонує різноманітні тарифні плани з різними умовами та вартістю, додаток для аналізу тарифів стає незамінним інструментом для користувачів. Мета такого додатку полягає в тому, щоб надати користувачам зручний та швидкий доступ до інформації про наявні тарифні плани мобільних операторів України. Головна мета полягає у спрощенні процесу вибору оптимального тарифу, дозволяючи користувачам порівняти різні пропозиції та обрати найбільш вигідний варіант з урахуванням їхніх індивідуальних потреб та вимог.

Цільова аудиторія вебзастосунку включає широкий спектр користувачів мобільного зв'язку в Україні. Вона охоплює молодіжний сегмент, який активно використовує мобільний Інтернет та різноманітні мобільні послуги, а також бізнес-клієнтів, які шукають оптимальні умови для корпоративного користування. Крім того, серед потенційних користувачів можуть бути ті, хто планує перехід до нового оператора зв'язку або оновлення свого поточного тарифного плану.

Оскільки інформація про тарифні плани мобільних операторів зазвичай представлена на їхніх вебсайтах, вебзастосунок для аналізу тарифів може

використовувати API (інтерфейс програмування застосунків) для отримання актуальної інформації з цих джерел. Таким чином, користувачам надається можливість перегляду та порівняння тарифів безпосередньо зі своїх мобільних пристроїв, забезпечуючи зручний та доступний інструмент для прийняття інформованих рішень щодо вибору тарифного плану.

Ці публікації дають змогу вивчити, що потрібно зробити: веб-додаток чи сайт, і як поетапно розробити веб-додаток, що є корисним для даної роботи.

1.2 Аналіз програмних продуктів-аналогів

У контексті розробки вебзастосунку для аналізу тарифів мобільних операторів важливо провести аналіз програмних продуктів-аналогів, які вже існують на ринку. Це дозволить отримати уявлення про функціональні можливості, інтерфейс, а також про те, які аспекти та функції користувачі вважають важливими при виборі тарифів. У цьому розділі будуть розглянуті деякі програмні продукти, що надають можливість порівняння тарифів мобільних операторів.

Одним з таких програмних продуктів є Tariffic (Швейцарія), додаток, який дозволяє користувачам аналізувати їхні поточні тарифні плани та порівнювати їх з іншими доступними варіантами на ринку. Програма надає детальну інформацію про вартість, обсяг послуг (дзвінки, SMS, мобільний Інтернет), а також допомагає вибрати найбільш вигідний тариф для користувача. Tariffic - це інноваційний додаток, спрямований на забезпечення економічних міжнародних дзвінків. З Tariffic користувачам не потрібен план даних, оскільки дзвінки маршрутизуються традиційним шляхом, без використання VoIP [6].

Основні особливості Tariffic включають безкоштовний стартовий кредит для перевірки якості звуку та зручності використання, низькі тарифи на переадресацію, відсутність потреби у плані даних, можливість налаштування другого номера викликаючого абонента та вибір номера набору з доступних країн. Додаток підтримує англійську та німецьку мови. Для власників плоскої тарифної ставки на міські номери або включених хвилин на міські номери, Tariffic дозволяє

здійснювати вигідні дзвінки, генеруючи місцевий міський номер для дорогих міжнародних напрямків.

Для тестування Tariffic, кожна нова реєстрація отримує стартовий кредит у розмірі €1, який можна використовувати для обмежених напрямків.

My Data Manager (США) допомагає користувачам контролювати використання мобільного Інтернету, дзвінків та SMS, що дозволяє їм ефективно керувати своїми тарифами. Додаток аналізує споживання даних та рекомендує оптимальні тарифи на основі користувальницьких звичок. My Data Manager вважається одним з найкращих додатків для контролю використання даних на телефонах і економії коштів. Понад 11,5 мільйонів користувачів вже використовують цей додаток, щоб відслідковувати, скільки даних вони використовують, та отримувати сповіщення перед перевищенням ліміту, уникаючи зайвих витрат [7].

Основні функції My Data Manager включають відслідковування використання даних, зокрема мобільного трафіку, Wi-Fi та роумінгу, а також можливість встановлювати сповіщення про перевищення трафіку. Додаток також дозволяє відслідковувати, які програми використовують найбільше даних, слідкувати за використанням даних кожним членом сім'ї або іншими особами у випадку спільного тарифу, керувати використанням даних на кількох пристроях, оцінювати історію використання даних для підключення до найкращого тарифного плану та економити гроші за рахунок ефективного використання даних. Розробником додатку є data.ai Basics, якій довіряють більше 1 мільйона користувачів. data.ai є провідним світовим постачальником оцінок продуктивності мобільних додатків, допомагаючи розробникам створювати кращі додатки.

Data Monitor є додатком, що дозволяє користувачам контролювати використання своїх мобільних даних та забезпечує ефективну економію. За допомогою Data Monitor користувачі можуть відстежувати своє використання даних, налаштовувати ліміти та отримувати сповіщення про наближення до ліміту. Основні функції Data Monitor включають відстеження використання даних з детальними статистичними даними, налаштування лімітів використання даних для

уникнення перевищення ліміту та додаткових витрат, отримання сповіщень про наближення до ліміту використання даних, аналіз додатків для визначення тих, що споживають найбільше даних, та перегляд історії використання даних для аналізу та вдосконалення своїх звичок використання даних. Data Monitor є ефективним інструментом для керування мобільними даними та забезпечення економії.

1.3 Постановка задачі

Постановка задачі полягає в чіткому визначенні цілей і завдань, які необхідно вирішити під час розробки додатку для аналізу тарифів мобільних операторів України. Основні завдання включають розробку функціоналу додатку, визначення технічних вимог, розробку дизайну та інтерфейсу користувача, тестування, впровадження та підтримку програмного забезпечення.

Визначення функціоналу додатку включає реалізацію можливості перегляду та порівняння тарифів мобільних операторів, надання користувачам можливості ознайомитися з усіма доступними тарифними планами. Додаток має підтримувати актуальні дані про наявні тарифні плани, використовуючи API мобільних операторів або інші джерела для отримання та оновлення інформації. Він також повинен забезпечувати можливість фільтрації тарифів за різними параметрами, такими як ціна, обсяг послуг, терміни та умови. Відображення додаткових переваг та обмежень кожного тарифного плану, включаючи технічні характеристики, швидкість Інтернету, обмеження на обсяг даних та інші параметри, є обов'язковим. Реалізація інструментів для розрахунку вартості послуг дозволяє користувачам розраховувати загальну вартість послуг залежно від обраного тарифу та їх персональних потреб. Інтерфейс додатку повинен бути зручним та інтуїтивно зрозумілим, що забезпечить легкість використання та навігацію.

Визначення технічних вимог включає вибір платформи для розробки додатку, враховуючи потреби цільової аудиторії. Вибір мов програмування та технологій має базуватися на ефективності реалізації функціоналу, швидкості розробки та можливостях підтримки. Розробка бази даних для збереження та

оновлення інформації про тарифні плани мобільних операторів є ключовим етапом, що потребує оптимізації структури для забезпечення ефективної роботи додатку. Забезпечення високої швидкодії та безпеки включає оптимізацію коду, використання надійних протоколів зв'язку та заходи захисту даних користувачів.

Розробка дизайну та інтерфейсу користувача має на меті створення зручного та привабливого інтерфейсу. Важливо ретельно продумати розміщення елементів інтерфейсу, розмір кнопок, функціональність жестикуляції та забезпечити відповідність сучасним тенденціям та стандартам. Дизайн повинен включати гармонійну колірну палітру, зручні шрифти та ергономічне розміщення елементів на екрані.

Тестування додатку включає виконання різноманітних видів тестування, зокрема функціонального тестування для перевірки відповідності кожної функції вимогам та очікуванням користувача. Регресійне тестування забезпечує перевірку раніше випробуваних функцій після внесення змін у код, щоб впевнитися, що нові функції або виправлення помилок не вплинули на роботу вже існуючих функцій. Тестування на різних пристроях та платформах включає перевірку сумісності додатку з різними версіями операційних систем (Android, iOS) та різними типами пристроїв (смартфони, планшети). Адаптивність інтерфейсу до різних розмірів екранів та роздільних здатностей є критично важливою для забезпечення універсальності додатку.

РОЗДІЛ 2

МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

2.1 Моделювання

Моделювання вебзастосунку є процесом створення абстрактної моделі або концепції вебзастосунку перед його реальною розробкою. Цей процес дозволяє розробникам та зацікавленим сторонам краще зрозуміти функціонал та взаємодію елементів застосунку перед тим, як розпочати його реалізацію. Моделювання вебзастосунків включає створення схем, діаграм, прототипів та інших інструментів для візуалізації та аналізу різних аспектів застосунку.

На першому етапі, збору вимог, проводяться зустрічі з клієнтами або стейкхолдерами, опитування користувачів, аналіз конкурентів та інші заходи для визначення функціональності та інших аспектів застосунку. Це дозволяє зібрати необхідну інформацію, яка стане основою для подальшого проектування. Після збору вимог розробники створюють схеми та діаграми, що відображають структуру та взаємозв'язки між різними елементами застосунку. Наприклад, можуть бути створені діаграми класів, діаграми сутностей-зв'язків та інші схематичні представлення, які допомагають зрозуміти архітектуру майбутнього застосунку.

Розробка прототипів є наступним важливим етапом моделювання. Прототипи дозволяють візуалізувати вигляд та поведінку застосунку на ранніх етапах. Це можуть бути прототипи інтерфейсу користувача (UI), які демонструють розташування елементів та навігацію, або прототипи функціоналу (UX), що демонструють роботу різних функцій. Аналіз та оцінка прототипів проводяться для перевірки їх ефективності та відповідності вимогам. На цьому етапі можуть бути виявлені та виправлені недоліки або недоречності в дизайні чи функціональності.

Після аналізу прототипів вносяться необхідні зміни до схем, діаграм та прототипів для вирішення виявлених проблем або покращення загального дизайну та функціоналу. Цей етап уточнення та оптимізації є критично важливим для забезпечення високої якості кінцевого продукту. Після успішного аналізу та уточнень прототипу він може бути затверджений для подальшої розробки. На цьому етапі планується детальний графік реалізації функціоналу та розвитку застосунку.

Моделювання вебзастосунку є важливим етапом у його розробці, оскільки дозволяє розробникам та стейкхолдерам краще зрозуміти, яким має бути кінцевий продукт, та уникнути помилок чи недорозумінь під час реалізації. Вебзастосунок для порівняння тарифів мобільних операторів України розробляється на основі комплексної моделі, що включає різноманітні складові. Однією з ключових складових є модель даних, яка буде відображена за допомогою UML-діаграми класів. У цій діаграмі передбачено такі класи, як `User`, `Tariff`, `Operator` та `ComparisonResult`. Клас `User` відображає користувача системи, який може мати багато вибраних тарифів, представлених класом `Tariff`. Кожен тариф, у свою чергу, належить до одного оператора, який відображається класом `Operator`. Результати порівняння тарифів зберігаються в класі `ComparisonResult`.

Окрім того, в рамках розробки застосунку передбачено створення прототипу інтерфейсу користувача. Головна сторінка застосунку буде містити можливість вибору тарифів та перегляду порівняння. Користувач матиме можливість переглянути список доступних тарифів та вибрати необхідний, а також застосувати фільтри для швидкого пошуку за оператором. На сторінці результатів порівняння будуть відображені переваги та недоліки кожного тарифу, щоб користувач міг зробити обдуманий вибір. Крім того, у застосунку буде доступна сторінка контактів, за допомогою якої користувач може зв'язатися з адміністратором у разі виникнення питань або потреби в допомозі.

Вебзастосунок забезпечить зручний та інтуїтивно зрозумілий інтерфейс, що дозволить користувачам легко орієнтуватися та здійснювати необхідні дії. Такий підхід до розробки забезпечить ефективне використання застосунку та задоволення потреб користувачів у сфері порівняння тарифів мобільних операторів.

2.2 Проектування інформаційної системи

Проектування інформаційної системи є процесом розробки архітектури, компонентів та функціональності системи з урахуванням потреб користувачів і бізнес-вимог. Цей процес включає декілька ключових етапів, починаючи з аналізу вимог. На цьому етапі здійснюється збір і аналіз вимог до системи від користувачів і бізнесу, що включає визначення функціональних вимог, які визначають, що система повинна робити, та нефункціональних вимог, таких як продуктивність, безпека, доступність.

Проектування архітектури є наступним етапом, на якому визначається загальна архітектура системи, включаючи розподіл компонентів, інтерфейси, технології, бази даних та інші ключові аспекти. Це може включати створення схем баз даних, діаграм класів, компонентів та послідовності. На етапі розробки бази даних створюється структура бази даних системи, включаючи таблиці, поля, зв'язки та індекси, що базується на аналізі вимог до даних та потреб користувачів.

Етап розробки програмного забезпечення включає створення компонентів програмного забезпечення системи, таких як серверна та клієнтська частини, інтеграція з базою даних, створення веб-інтерфейсів та інші функціональні елементи. Після розробки програмного забезпечення проводяться різноманітні види тестування, включаючи функціональне та регресійне тестування, для забезпечення якості роботи додатку та відповідності вимогам та очікуванням користувачів.

Після успішного завершення тестування система впроваджується в експлуатацію. Впровадження включає навчання персоналу, підтримку користувачів та вирішення можливих проблем під час експлуатації. Також

здійснюється постійний моніторинг роботи системи та внесення необхідних змін і покращень.

Моделювання інформаційної системи для додатку порівняння тарифів мобільних операторів України включає створення комплексної моделі, що забезпечує можливість користувачам переглядати і порівнювати тарифи мобільних операторів без необхідності авторизації. Адміністраторам, які авторизувалися, надається розширений функціонал, що включає керування контентом додатку, додавання акцій, редагування тарифів та інші можливості.

Діаграма потоків даних (DFD) відображає обмін даними між користувачами та системою. Потік даних від користувача до системи включає вибір тарифів мобільних операторів, де користувач може вибрати необхідні параметри, такі як тип послуги або обсяги, для отримання переліку доступних тарифів. Потік даних від системи до користувача передає результати порівняння тарифів, які формуються системою на основі вибраних користувачем параметрів. Користувач отримує перелік тарифів разом із порівняльними характеристиками та рекомендаціями.

Створення контекстної діаграми DFD дозволяє узагальнити та уявити, як відбувається обмін даними між різними частинами системи та її користувачами. На діаграмі будуть відображені основні актори, такі як клієнт без авторизації та адміністратор з авторизацією. Потоки даних будуть представлені переглядом та порівнянням тарифів, а також управлінням контентом системи. Цей підхід сприяє розумінню роботи системи та ефективному її розробленню, забезпечуючи зручний та інтуїтивно зрозумілий інтерфейс для користувачів (рис 2.1).

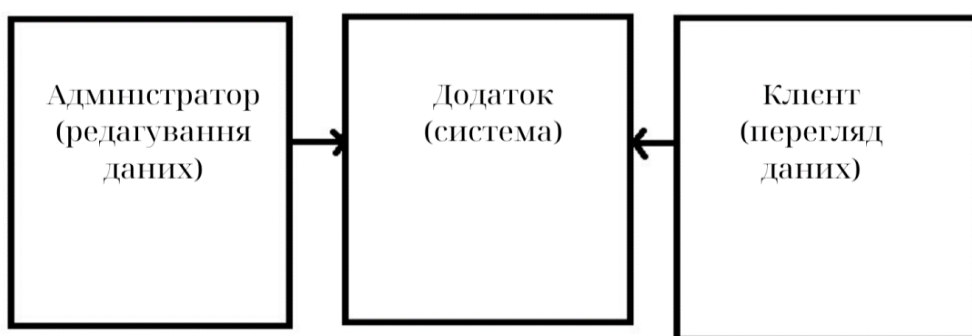


Рисунок 2.1– Контекстна діаграма DFD

Розроблення діаграм рівнів DFD - система буде розбита на рівні, що включають в себе перегляд та порівняння тарифів, управління контентом, а також авторизацію користувачів (рис 2.2).

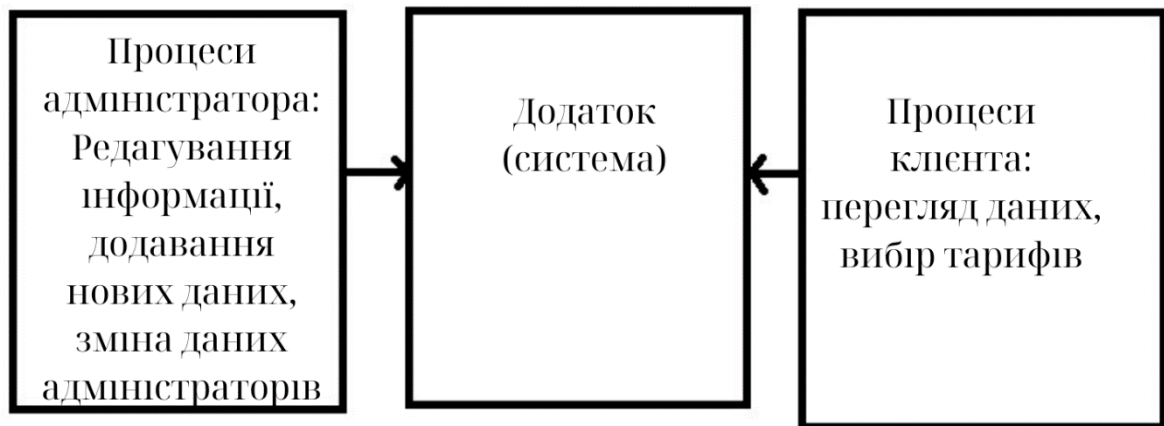


Рисунок 2.2 – Діаграма 1 рівень DFD

2.3 Проектування моделі бази даних

Проектування моделі бази даних - це важливий етап у розробці інформаційної системи, який визначає структуру та організацію даних, що зберігаються в системі. Основні кроки проектування моделі бази даних включають визначення сутностей та атрибутів, визначення зв'язків між сутностями, нормалізацію, визначення індексів та ключів, а також створення фізичної моделі бази даних. Сутності представляють об'єкти або поняття, про які потрібно зберігати інформацію. Наприклад, якщо розглядається система управління студентами, сутностями можуть бути "Студент", "Курс", "Викладач" і т.д. Для кожної сутності визначаються атрибути - це конкретні характеристики або властивості сутностей. Наприклад, для сутності "Студент" атрибутами можуть бути "Ім'я", "Прізвище", "Дата народження" і т.д. Після визначення сутностей потрібно встановити, як вони взаємодіють між собою. Зв'язки визначаються відносинами між сутностями і вказують, які сутності пов'язані між собою і яка є природа цих зв'язків. Наприклад, у системі управління студентами може бути зв'язок "Бере участь" між сутностями "Студент" і "Курс", що показує, які студенти записані на які курси. Нормалізація -

це процес організації даних в базі даних з метою усунення дублювання даних і забезпечення ефективності зберігання та обробки інформації. Нормалізація допомагає уникнути аномалій даних і забезпечити консистентність бази даних. Індеси та ключі використовуються для швидкого доступу до даних в базі даних. Вони дозволяють швидко знаходити конкретні записи та забезпечують унікальність даних. Наприклад, для кожної таблиці в базі даних може бути визначений первинний ключ, який є унікальним ідентифікатором кожного запису. Створення фізичної моделі бази даних визначає, як саме будуть зберігатися дані в базі даних, включаючи типи даних, розміри полів, індеси та інші аспекти. Фізична модель бази даних визначається конкретною системою управління базами даних (СУБД), яку буде використовуватися для реалізації системи.

Під час розробки веб-додатку для порівняння тарифів мобільних операторів України було визначено кілька ключових аспектів моделювання. По-перше, створено базу даних, що включає файли для збереження інформації про користувачів, тарифи, операторів та відгуки. Для організації взаємодії з користувачами розроблено UML діаграму варіантів використання, яка включає отримання списку тарифів, вибір тарифів для порівняння, виконання порівняння та перегляд результатів. Далі, розроблено діаграми послідовності для опису процесу вибору та порівняння тарифів, а також для збереження обраних тарифів до файлів. У відповідності з клієнт-серверною архітектурою, додаток складається з клієнтської та серверної частин. Клієнтська частина відповідає за інтерфейс користувача, дозволяючи клієнтам переглядати тарифи, вибирати їх для порівняння та переглядати результати. Серверна частина містить логіку порівняння та збереження даних, обробляючи запити від клієнтської частини. Всі ці аспекти спрямовані на створення зручного та ефективного інструменту для користувачів, що дозволить їм отримувати необхідну інформацію про тарифи мобільних операторів та здійснювати обдумані вибори.

Для нашого додатку, який дозволяє коригувати інформацію адміністраторам та переглядати її клієнтам без авторизації, проектування моделі бази даних буде мати такі кроки. Визначення сутностей та атрибутів: сутності включають

користувача, тарифи, операторів та відгуки. Користувач має атрибути: ім'я, прізвище, логін, пароль, роль (адміністратор). Тарифи мають атрибути: оператор, план, ціна. Оператори мають атрибути: ім'я. Відгуки мають атрибути: ім'я, email, повідомлення. Визначення зв'язків між сутностями: зв'язок "Має доступ" між сутностями "Користувач" і "Тарифи", що показує, який користувач має доступ до яких даних. Нормалізація: перевірка на аномалії дублювання даних і вирішення їх шляхом розбиття файлів та створення зв'язків. Визначення індексів та ключів: визначення первинного ключа для кожного файлу, наприклад, ідентифікатор користувача та ідентифікатор тарифів. Встановлення індексів для полів, які часто використовуються для пошуку, наприклад, логін користувача. Створення фізичної моделі бази даних: вибір конкретного формату для реалізації бази даних, наприклад, JSON. Визначення типів даних для кожного атрибуту, наприклад, STRING для текстових полів, NUMBER для чисел. Створення файлів з урахуванням визначених сутностей та зв'язків між ними. Встановлення необхідних обмежень, наприклад, унікальності логінів користувачів. Створення індексів для полів, які потребують швидкого доступу.

Ось приклад структури JSON-файлів, що використовується у нашому додатку:

Файл `admins.json`:

```
[ {
  "username": "admin",
  "password": "123"
}]
```

Файл `operators.json`:

```
[ {
  "name": "Life"
}, {
  "name": "Vodafone"
}, {
  "name": "Київстар"
```

```
}]
```

Файл `tariffs.json`:

```
[ {
  "operator": "Life",
  "plan": "Тариф 1",
  "price": 200
}, {
  "operator": "Vodafone",
  "plan": "Тариф 2",
  "price": 250
}, {
  "operator": "Київстар",
  "plan": "Тариф 3",
  "price": 300
}]
```

Файл `promotions.json`:

```
[ {
  "operator": "Life",
  "promotion": "Акція 1",
  "date": "2024-05-01"
}, {
  "operator": "Vodafone",
  "promotion": "Акція 2",
  "date": "2024-05-02"
}, {
  "operator": "Київстар",
  "promotion": "Акція 3",
  "date": "2024-05-03"
}]
```

Файл `feedback.json`:

```
[ {
  "name": "Іван",
  "email": "ivan@example.com",
  "message": "Чудовий сервіс!"
}]
```

Це проектування бази даних з використанням JSON-файлів забезпечує гнучкість і простоту у реалізації додатку, що дозволяє адміністраторам коригувати інформацію, а клієнтам переглядати її без авторизації.

Інтерфейс додатку робимо у кавово-бежевих тонах на мові HTML. Робимо шість сторінок: вхід, сторінка адміністратора, сторінка клієнта, сторінка оператора, сторінка відгуку, сторінка для списку відгуків (рис. 2.3-2.9)

Рисунок 2.3 – Сторінка для входу адміністратора

Рисунок 2.4 – Сторінка адміністратора

Оператор	Тариф	Ціна
Life	Basic Plan	200
Київстар	Basic Plan	210
Vodafone	Basic Plan	220
Life	Premium Plan	250
Київстар	Premium Plan	260
Vodafone	Premium Plan	270
Київстар	Річний	1500

Рисунок 2.5 – Сторінка клієнта

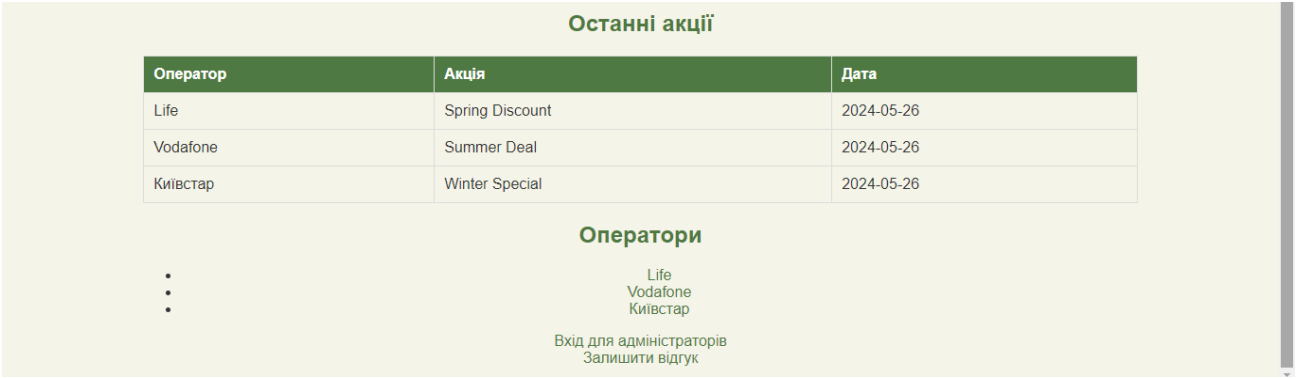


Рисунок 2.6 – Сторінка клієнта продовження

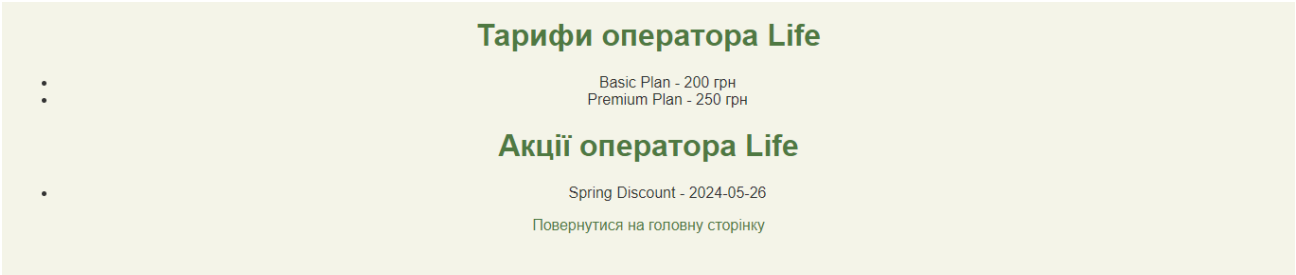


Рисунок 2.7 – Сторінка оператора

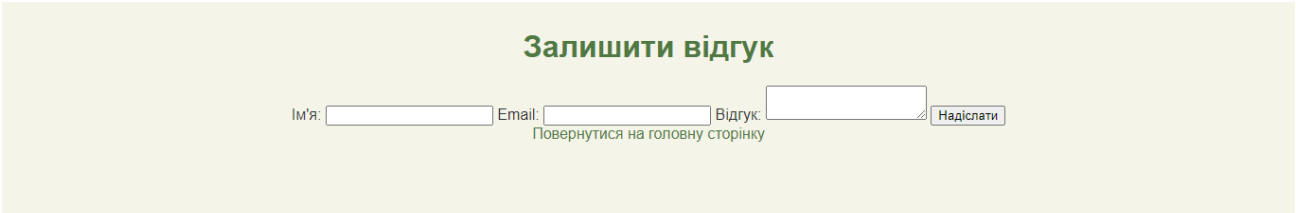


Рисунок 2.8 – Сторінка відгуків

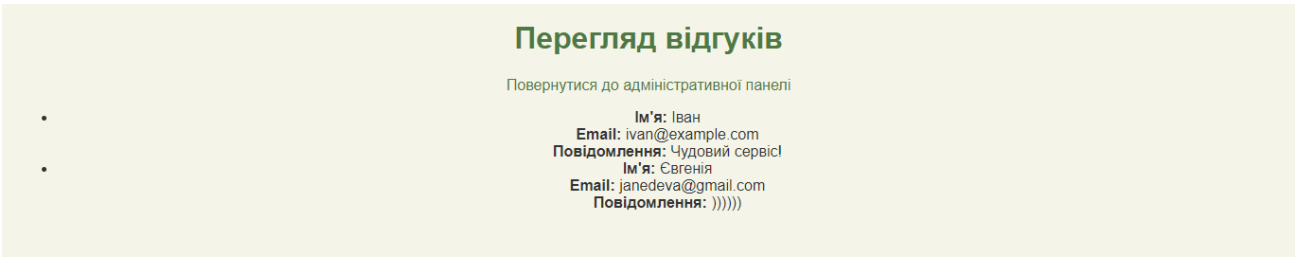


Рисунок 2.9 – Сторінка перегляду відгуків

РОЗРОБКА ВЕБ-ДОДАТКУ ДЛЯ АНАЛІЗУ ТАРИФІВ МОБІЛЬНИХ ОПЕРАТОРІВ УКРАЇНИ

3.1 Архітектура програмного додатку

Розробка програмного додатка для аналізу тарифів мобільних операторів України є складним та важливим завданням в сучасній інформаційній технологічній сфері. Під час проектування такого додатка велика увага приділяється його архітектурі, оскільки вона визначає загальну структуру, розподіл функцій та взаємозв'язки між різними компонентами. У цьому розділі ми детально розглянемо архітектурні аспекти програмного додатка для аналізу тарифів мобільних операторів України.

Першим кроком у розробці архітектури додатка є визначення його основних компонентів та взаємозв'язків між ними. У нашому випадку основними компонентами є клієнтська та серверна частини. Клієнтська частина відповідає за інтерфейс користувача, який забезпечує можливість перегляду та аналізу тарифів. Вона доступна через веб-браузер, що дозволяє користувачам отримувати доступ до додатка з будь-якого пристрою, що має підключення до Інтернету. Серверна частина відповідає за обробку запитів від клієнтської частини, виконання аналізу та порівняння тарифів, а також збереження даних у JSON-файлах.

Одним з ключових аспектів архітектури додатка є його масштабованість. Додаток повинен бути готовий працювати з великою кількістю користувачів та обробляти великі обсяги даних. Тому під час проектування архітектури слід враховувати можливість горизонтального та вертикального масштабування. Горизонтальне масштабування передбачає розподіл навантаження між різними серверами, що дозволяє збільшити продуктивність та надійність системи. Вертикальне масштабування включає у себе збільшення потужності окремих серверів шляхом додавання ресурсів, таких як процесори, пам'ять та сховища даних.

Для забезпечення безпеки додатка важливо враховувати заходи захисту інформації та забезпечення конфіденційності даних користувачів. Зокрема, важливо використовувати шифрування для передачі та збереження конфіденційної інформації, регулярно оновлювати програмне забезпечення для запобігання використанню вразливостей та здійснювати аудит безпеки для виявлення можливих загроз.

Ще одним важливим аспектом архітектури додатка є його ефективність та продуктивність. Для забезпечення швидкої відповіді на запити користувачів і обробки великого обсягу даних важливо правильно вибрати технології та інструменти розробки, оптимізувати запити до файлів даних та використовувати кешування для збереження результатів попередніх запитів.

Узагальнюючи, успішна розробка програмного додатка для аналізу тарифів мобільних операторів України вимагає ретельного проектування архітектури, яка враховує масштабованість, безпеку, ефективність та інші важливі аспекти. Наявність добре структурованої та оптимізованої архітектури грає ключову роль у забезпеченні успіху проекту та задоволення потреб користувачів.

Основні компоненти архітектури включають в себе клієнтську та серверну частини. Клієнтська частина відповідає за інтерфейс користувача, який може бути доступний через веб-браузер. Вона має надати зручність у користуванні та інтуїтивно зрозумілі функції для вибору та порівняння тарифів різних мобільних операторів. Серверна частина відповідає за обробку запитів від клієнтів та виконання всіх необхідних операцій для аналізу та порівняння тарифів. Вона має забезпечувати швидку та ефективну роботу, включаючи доступ до файлів даних, обробку даних та відправку результатів до клієнта.

Файли даних відіграють ключову роль у збереженні та організації інформації про доступні тарифи мобільних операторів, а також результати порівняння. Вони повинні бути оптимізовані для швидкого доступу до даних та забезпечувати цілісність і безпеку інформації. Під час проектування архітектури такого додатку також варто врахувати можливості масштабування. Система повинна бути готова працювати з великим обсягом даних та великою кількістю користувачів, що може

вимагати горизонтального масштабування, наприклад, за допомогою використання кластеризації серверів.

Щоб забезпечити безпеку додатку, важливо врахувати заходи захисту, такі як шифрування даних, перевірку коректності введених даних, контроль доступу до функціоналу додатку та захист від атак злому. У підсумку, ефективна архітектура програмного додатку для аналізу тарифів мобільних операторів України повинна бути гнучкою, масштабованою та забезпечувати надійність та безпеку усіх його компонентів.

Для додатку, який використовує JSON-файли для зберігання даних, архітектура має такі компоненти. Клієнтська частина, створена з використанням HTML, CSS і JavaScript, відповідає за відображення інтерфейсу користувача. HTML забезпечує структуру сторінок, CSS відповідає за стилізацію відображення, а JavaScript додає інтерактивність, дозволяючи користувачам взаємодіяти з додатком.

Серверна частина, реалізована за допомогою Flask, обробляє запити від клієнтської частини та взаємодіє з файлами даних. Flask є мікрофреймворком для Python, який дозволяє швидко створювати веб-застосунки. Він підтримує використання шаблонів Jinja2 для динамічного генерування HTML-сторінок, що дозволяє створювати більш інтерактивні та динамічні інтерфейси.

JSON-файли використовуються для зберігання даних про адміністраторів, операторів, тарифи, акції та відгуки. Ці файли забезпечують простий спосіб збереження та доступу до даних без використання реляційної бази даних. JSON є легким форматом для обміну даними, який легко читається як людиною, так і машиною, що робить його ідеальним для зберігання структурованої інформації в додатку.

Додаток для аналізу тарифів мобільних операторів України розроблений з використанням мови програмування Python та веб-фреймворка Flask. Проект організовано таким чином, щоб забезпечити легкість підтримки, масштабованість і зручність розробки. Архітектура додатку розроблена для забезпечення ефективного та безперебійного функціонування, з урахуванням потреб

користувачів у швидкому та зручному доступі до інформації про тарифи мобільних операторів.

Клієнтська частина забезпечує зручний і привабливий інтерфейс для користувачів, дозволяючи їм легко навігувати та знаходити необхідну інформацію. Серверна частина відповідає за логіку додатку та обробку даних, забезпечуючи стабільну роботу та швидку реакцію на запити користувачів. JSON-файли служать основним сховищем даних, що дозволяє зберігати інформацію у структурованому вигляді та забезпечує легкий доступ до неї.

Такий підхід до організації архітектури додатку дозволяє легко адаптувати його до змін і вимог користувачів, забезпечуючи високу гнучкість і надійність роботи системи. Проектування додатку з використанням Flask і JSON-файлів дозволяє створювати ефективні та зручні рішення для аналізу тарифів мобільних операторів, забезпечуючи користувачам можливість швидкого та зручного доступу до необхідної інформації.

Структура проекту включає кореневу папку проекту під назвою `TariffAnalysis`. У цій папці містяться основні файли та підпапки, необхідні для функціонування додатку.

Основний файл додатку `app.py` ініціалізує Flask-додаток та визначає основні маршрути. У цьому файлі визначені функції для відображення головної сторінки додатку з тарифами та акціями, сторінки входу для адміністраторів, виходу з адмін-панелі, відображення адміністративної панелі, перегляду тарифів та акцій окремого оператора, додавання нового тарифу, оператора та акції, форми для відгуків користувачів та перегляду відгуків користувачів.

Файл `views.py` містить функції обробки запитів, відомі як `view functions`, та логіку обробки даних, яка взаємодіє з JSON файлами. Ці файли використовуються для збереження даних про адміністраторів, мобільних операторів, акції, тарифи та відгуки користувачів. Папка `data` містить ці файли: `admins.json` для збереження даних про адміністраторів, `operators.json` для збереження даних про мобільних операторів, `promotions.json` для збереження даних про акції мобільних операторів,

tariffs.json для збереження даних про тарифи мобільних операторів та feedback.json для збереження даних про відгуки користувачів.

Папка static містить файл styles.css, який відповідає за стилізацію зовнішнього вигляду сторінок додатку. Папка templates містить HTML шаблони для різних сторінок додатку. Серед них є такі шаблони: about.html для сторінки Про нас, admin.html для адміністративної панелі, яка дозволяє адміністраторам керувати даними операторів, тарифів та акцій, contact.html для сторінки Контакти, feedback.html для форми відгуків користувачів, index.html для головної сторінки додатку, layout.html як основний шаблон для інших HTML сторінок, login.html для сторінки входу адміністраторів, operator.html для відображення інформації про окремого оператора, promotions.html для відображення акцій та view_feedbacks.html для відображення відгуків користувачів.

Основні файли та функції проекту включають app.py і views.py. У файлі app.py визначені маршрути для різних сторінок додатку, таких як головна сторінка з тарифами та акціями, сторінка входу та виходу для адміністраторів, адміністративна панель, перегляд тарифів та акцій окремого оператора, додавання нового тарифу, оператора та акції, форма для відгуків користувачів та перегляд відгуків користувачів. Файл views.py містить функції для обробки цих запитів і логіку обробки даних.

HTML шаблони включають admin.html для адміністративної панелі, яка дозволяє керувати даними операторів, тарифів та акцій, feedback.html для форми відгуків користувачів, де користувачі можуть залишати свої відгуки або запити, view_feedbacks.html для відображення відгуків користувачів, де адміністратори можуть переглядати всі отримані відгуки. Інші шаблони включають index.html для головної сторінки додатку, login.html для сторінки входу адміністраторів, operator.html для відображення інформації про окремого оператора та promotions.html для відображення акцій.

Ця структура проекту дозволяє легко організовувати та підтримувати додаток, забезпечуючи його масштабованість та зручність використання. Завдяки чіткій організації файлів і папок, розробники можуть легко орієнтуватися в проекті,

швидко знаходити необхідні файли та вносити зміни без зайвих ускладнень. Це сприяє ефективному розробленню, тестуванню та впровадженню нових функцій, забезпечуючи при цьому високу якість кінцевого продукту.

3.2 Програмна реалізація

Програмна реалізація включає створення основних компонентів додатку.

Файл `app.py` є основним файлом програми, який ініціалізує Flask-додаток та визначає основні маршрути для обробки запитів користувачів. Він забезпечує основну логіку роботи додатку, включаючи взаємодію з користувачами та обробку їх запитів.

Файли JSON використовуються для зберігання даних про адміністраторів, операторів, тарифи, акції та відгуки. Ці файли зберігаються в папці `data` і забезпечують структуру зберігання даних у форматі, який легко читати та редагувати.

Файли HTML зберігаються в папці `templates` та визначають зовнішній вигляд сторінок додатку. Кожен HTML файл відповідає за відображення певної сторінки, наприклад, головної сторінки, сторінки входу, адміністративної панелі тощо, забезпечуючи інтерфейс для взаємодії користувачів із додатком.

Після реалізації функціональності додатку, необхідно провести тестування, щоб переконатися, що всі компоненти працюють коректно. Це включає в себе тестування користувацького інтерфейсу, серверної логіки та взаємодії з файлами JSON. Тестування допомагає виявити можливі помилки та забезпечити стабільну роботу додатку.

Після успішного тестування додаток можна розгортати на виробничому сервері. Для цього необхідно налаштувати серверну інфраструктуру та забезпечити надійний доступ до додатку для кінцевих користувачів. Це включає в себе налаштування серверного оточення, розгортання додатку та забезпечення його доступності для користувачів.

У підсумку, програмна реалізація додатку для аналізу тарифів мобільних

операторів України охоплює весь цикл розробки, від проектування до розгортання. Дотримання цих кроків забезпечує створення надійного, масштабованого та зручного інструменту для користувачів.

Код, представлений у додатку В, створює вебзастосунок для аналізу тарифів мобільних операторів України за допомогою Flask, веб-фреймворка на мові програмування Python. Код містить маршрути для обробки запитів користувачів, завантаження та збереження даних у форматі JSON, а також відображення різних сторінок.

Імпорт необхідних модулів включає `datetime` для роботи з датами та часом, `Flask` та його компоненти `render_template`, `request`, `redirect`, `url_for`, `session`, `flash` для створення веб-застосунку та роботи з запитами, `json` для роботи з JSON файлами, а також `os` для роботи з файловою системою.

Ініціалізація додатка включає створення об'єкта `Flask`, встановлення секретного ключа для сесій та визначення шляху до папки з даними. Завантаження та збереження даних здійснюються за допомогою функцій `load_data(file_name)`, яка завантажує дані з JSON файлу, та `save_data(file_name, data)`, яка зберігає дані у JSON файл.

Маршрути для обробки запитів включають кілька ключових функцій. Функція `home()` обробляє запити до головної сторінки, завантажує тарифи, акції та операторів з JSON файлів, відсортовує тарифи за ціною та відображає останні акції кожного оператора, повертаючи шаблон `index.html`. Функція `login()` обробляє запити до сторінки входу для адміністраторів, перевіряє облікові дані та зберігає ім'я користувача в сесії у разі успіху, повертаючи шаблон `login.html`. Функція `logout()` обробляє запити для виходу з сесії адміністратора, видаляючи відповідні дані з сесії. Функція `admin()` обробляє запити до адміністративної панелі, перенаправляючи користувача на сторінку входу, якщо він не увійшов до системи, і повертає шаблон `admin.html`.

Інші маршрути включають `select_operator()`, яка обробляє запити для вибору оператора та перенаправляє на сторінку перегляду обраного оператора, `operator_view(name)`, яка обробляє запити до сторінки перегляду інформації про

оператора, завантажуючи тарифи та акції обраного оператора і повертаючи шаблон `operator.html`. Функція `add_tariff()` обробляє запити для додавання нового тарифу, додаючи новий тариф до JSON файлу та перенаправляючи на адміністративну панель. Функція `add_operator()` обробляє запити для додавання нового оператора, додаючи його до JSON файлу та перенаправляючи на адміністративну панель. Функція `add_promotion()` обробляє запити для додавання нової акції, додаючи її до JSON файлу та перенаправляючи на адміністративну панель. Функція `feedback_form()` обробляє запити до сторінки відгуків, зберігаючи новий відгук у JSON файл у разі запиту методом POST та повертаючи шаблон `feedback.html`. Функція `view_feedbacks()` обробляє запити для перегляду відгуків, перенаправляючи користувача на сторінку входу, якщо він не увійшов до системи, та повертаючи шаблон `view_feedbacks.html`.

Такий підхід до організації програмної реалізації забезпечує ефективне використання додатку, високу якість та надійність його роботи, задовольняючи потреби користувачів у зручному інструменті для аналізу тарифів мобільних операторів.

3.3 Використання програмного додатку

Заходимо на головну сторінку та зразу входимо на сторінку адміністратора (рис 3.1), вводимо нового оператора DJuse, новий тариф до нього і нову акцію до нього (рис 3.2, рис 3.3). Переходимо на головну сторінку (рис 3.4).

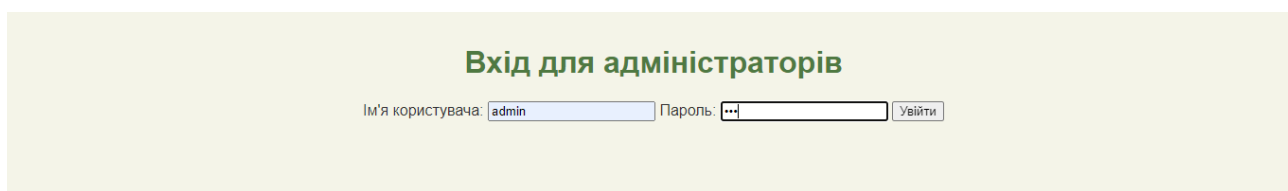


Рис 3.1 Вхід на сторінку адміністратора

Адміністративна панель

Вийти

Додати оператора

Назва:

Додати тариф

Оператор: План: Ціна:

Додати акцію

Оператор: Акція: Дата:

Перегляд відгуків

Переглянути відгуки

Рис 3.2 Введення нового оператора

Адміністративна панель

Вийти

Додати оператора

Назва:

Додати тариф

Оператор: План: Ціна:

Додати акцію

Оператор: Акція: Дата:

Перегляд відгуків

Переглянути відгуки

Рис 3.3 Введення тарифу та акції

Тарифи

Оператор	Тариф	Ціна
Life	Basic Plan	200
Київстар	Basic Plan	210
Vodafone	Basic Plan	220
Life	Premium Plan	250
Київстар	Premium Plan	260
Vodafone	Premium Plan	270
Київстар	Річний	1500
DJuse	Річний	2000

Останні акції

Оператор	Акція	Дата
Life	Spring Discount	2024-05-26
Vodafone	Summer Deal	2024-05-26
Київстар	Winter Special	2024-05-26
DJuse	Зима	2024-01-01

Оператори

•

•

•

•

Life
Vodafone
Київстар
DJuse

Рис 3.4 Головна сторінка після додавання нового оператора з тарифом та акцією
Залишимо відгук (рис 3.5), переходимо на перегляд відгуків (рис 3.6).

Залишити відгук

Ім'я: Email: Відгук:

Повернутися на головну сторінку

Рис 3.5 Залишаємо відгук

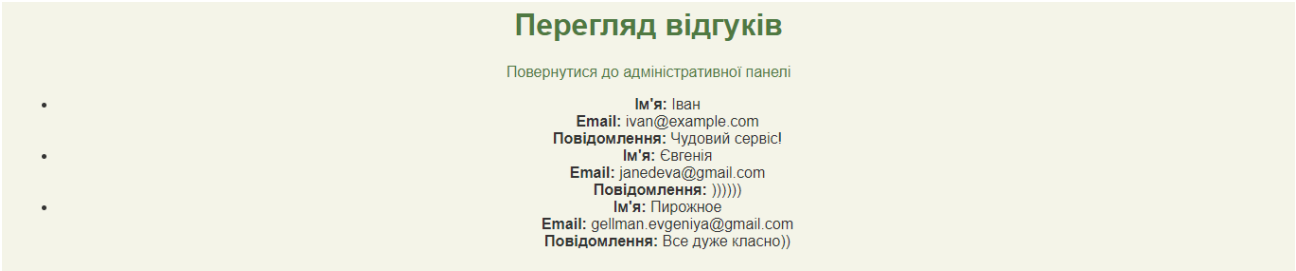
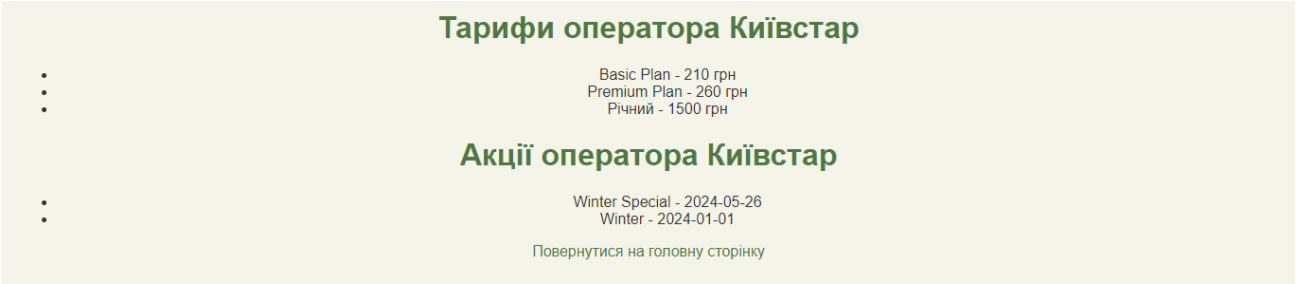


Рис 3.6 Перегляд відгуків

Заходимо на сторінку оператора Київстар (рис 3.7)



Програма працює згідно очікуваних результатів.

ВИСНОВКИ

У результаті виконання цієї роботи було розроблено веб-додаток для аналізу тарифів мобільних операторів України. Було здійснено детальний аналіз предметної області, зокрема, досліджено ринок мобільних операторів в Україні та їхні тарифні плани. Визначено вимоги до функціоналу додатку, що включають можливість порівняння тарифів, перегляду акцій та збереження відгуків користувачів.

Розроблено архітектуру додатку, що включає клієнтську та серверну частини. Клієнтська частина забезпечує зручний інтерфейс користувача, серверна частина відповідає за обробку запитів та взаємодію з файлами даних. Використано технології HTML, CSS, JavaScript для клієнтської частини та Flask для серверної частини. Дані зберігаються у форматі JSON.

Реалізовано функціонал для перегляду тарифів, порівняння тарифів, додавання нових тарифів та акцій, збереження відгуків користувачів. Створено зручну адміністративну панель для керування даними.

Проведено тестування додатку для перевірки його працездатності та відповідності вимогам. виправлено виявлені помилки та покращено загальну стабільність роботи додатку.

Додаток готовий до розгортання на виробничому сервері. Налаштовано інфраструктуру для надійного доступу користувачів до додатку.

Загалом, розробка веб-додатку для аналізу тарифів мобільних операторів України дозволяє забезпечити користувачам зручний та ефективний інструмент для порівняння тарифних планів, що сприяє прийняттю обґрунтованих рішень щодо вибору найбільш вигідного тарифу.

Для відкриття проекту, збереженого за допомогою Visual Studio, необхідно запустити файл ``runserver.py``.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. L. P. S., Vasilkova A. O. Analysis of the Market of Mobile Services in Ukraine. Modern Economics. 2018. Vol. 12, no. 1. P. 179–184. URL: [https://doi.org/10.31521/modecon.v12\(2018\)-27](https://doi.org/10.31521/modecon.v12(2018)-27) (date of access: 23.06.2024).
2. Smahliuk A., Nadtochii A. Analysis of the Confectionery Goods Market in Ukraine and the Perspectives of Its Development. International scientific journal "Internauka". Series: "Economic Sciences". 2016. No. 2(34). URL: <https://doi.org/10.25313/2520-2294-2020-2-5659> (date of access: 23.06.2024).
3. Іщенко М. Д. Білоус М. Ю. Життєвий цикл розробки веб-додатка: від планування до обслуговування. ElAr :: Головна. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/c50f8e04-ee6c-439c-8c0a-37925545532d/content> (date of access: 31.03.2024).
4. Створення додатку чи веб-сайту: У чому ключова різниця?. Ranktracker. URL: <https://www.ranktracker.com/uk/blog/building-an-app-vs-a-website-what-is-the-key-difference/> (date of access: 31.03.2024).
5. Як створити веб-додаток: типи, переваги, принцип роботи - Wezom. ІТ-компанія повного цикла розробки програмних продуктів WEZOM - Київ, Україна. URL: <https://wezom.com.ua/ua/blog/kak-sozdat-veb-prilozhenie> (date of access: 02.04.2024).
6. Tariffic. Softonic. URL: <https://tariffic.softonic.ru/iphone> (date of access: 02.04.2024).
7. House K. S. Data Monitor: Simple Net-Meter - Apps on Google Play. Android Apps on Google Play. URL: <https://play.google.com/store/apps/details?id=info.kfsoft.datamonitor> (date of access: 02.04.2024).
8. Лисновська К. Тарифи мобільних операторів України – рейтинг вигідних цін. novyny.live. URL: <https://finance.novyny.live/reiting-naideshevshikh-tarifiv-mobilnikh-operatoriv-ukrayini-146394.html> (date of access: 31.03.2024).
9. Порівняння базових тарифів мобільного зв'язку: що пропонують українські оператори у 2023 році. ІТС.ua. URL: <https://itc.ua/ua/articles/porivnyannya-bazovyh-taryfiv-mobilnogo-zv-yazku-shho-proponuyut-ukrayinski-operatory-u-2023-rotsi/> (date

of access: 31.03.2024).

10. Gleeson P. How to build a JSON API with Python. freeCodeCamp.org. URL: <https://www.freecodecamp.org/news/build-a-simple-json-api-in-python/> (date of access: 23.06.2024).

11. How to Build RESTful APIs with Python and Flask | Codementor. Codementor | Get live 1:1 coding help, hire a developer, & more. URL: <https://www.codementor.io/@dongido/how-to-build-restful-apis-with-python-and-flask-12qto530jd> (date of access: 23.06.2024).

12. Real Python. Build a Scalable Flask Web Project From Scratch – Real Python. Python Tutorials – Real Python. URL: <https://realpython.com/flask-project/> (date of access: 23.06.2024).

13. Python for web development / U. Patkar et al. International Journal of Computer Science and Mobile Computing. 2022. Vol. 11, no. 4. P. 36 – 48. URL: <https://doi.org/10.47760/ijcsmc.2022.v11i04.006> (date of access: 10.06.2024).

ДОДАТКИ

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ

**На розробку вебзастосунку аналізу тарифів мобільних операторів
України**

Погоджено:

Кандидат технічних наук, доцент кафедри
кібернетики та інформатики _
Толбатов А.В.

Студент групи ІСТ 2001

Федірко Данило Дмитрович

СУМИ 2024

1. Призначення й мета створення вебзастосунку

1.1 Призначення вебзастосунку

Вебзастосунок призначений для аналізу тарифів мобільних операторів України. Він дозволяє користувачам переглядати, порівнювати та аналізувати різні тарифні плани мобільних операторів, щоб вибрати найбільш підходящий варіант.

1.2 Мета створення вебзастосунку

Розробка зручного інструменту для користувачів, який дозволить ефективно порівнювати та аналізувати тарифні плани мобільних операторів України, з метою обрання оптимального тарифу відповідно до індивідуальних потреб та вимог користувачів

1.3 Цільова аудиторія

Цільова аудиторія вебзастосунку включає широкий спектр користувачів мобільного зв'язку в Україні, зокрема:

1. Молодіжний сегмент, який активно використовує мобільний Інтернет та різноманітні мобільні послуги.
2. Бізнес-клієнти, які шукають оптимальні умови для корпоративного користування.
3. Ті, хто планує перехід до нового оператора зв'язку або оновлення свого поточного тарифного плану.

2 Вимоги до вебзастосунку

2.1 Вимоги до вебзастосунку в цілому

2.1.1 Вимоги до структури й функціонування вебзастосунку

Вебзастосунок має бути реалізований у вигляді сайту, доступного в мережі Інтернет. Архітектура вебзастосунку реалізує клієнт-серверний підхід з використанням server-side rendering, що забезпечує роботу як у браузері клієнта, так і на сервері. Сайт складається з взаємозалежних розділів з чітко розділеними функціями для забезпечення логічної навігації та швидкого доступу до потрібної інформації.

2.1.2 Вимоги до персоналу

Персонал, який буде підтримувати та експлуатувати вебзастосунок, не повинен мати спеціальних технічних навичок, окрім базових знань роботи з персональним комп'ютером та стандартним веб-браузером (наприклад, Google Chrome).

2.1.3 Вимоги до збереженні вебзастосунку

Вся інформація у вебзастосунку буде зберігатися у базі даних, реалізованій засобами системи управління базами даних JSON.

2.1.4 Вимоги до розмежування доступу

Вебзастосунок не є приватним.

Користувачів можна поділити на клієнтів та адміністратора відповідно до прав доступу.

Клієнти можуть переглядати сторінку.

Адміністратор може редагувати зовнішній вигляд та наповнення вебзастосунку після авторизації за логіном та паролем.

2.2 Структура вебзастосунку

2.2.1 Навігація

– У шапці сайту буде створена система контентного меню для швидкого переміщення користувачів по всіх доступних сторінках. Меню відображатиметься на всіх сторінках, що дозволить користувачам у будь-який момент перейти на потрібну сторінку.

2.2.2 Наповнення вебзастосунку (контент)

– Сторінки всіх розділів сайту створюватимуться автоматично на основі даних з бази даних на сервері. Редагування вмісту розділів можливе через адміністративний веб-інтерфейс, що дозволяє редагувати інформаційний вміст сторінок без необхідності програмування. Наповнення сторінок здійснюється за допомогою шаблонів сайту.

2.3 Вимоги до функціонування системи

2.3.1 Потреби користувача

Система керування контентом повинна надавати можливість додавання, редагування й видалення вмісту сторінок. Має бути передбачена можливість додавання інформації без її відображення на сайті для користувачів.

2.3.2 Функціональні вимоги

На головній сторінці будуть представлені такі елементи:

- меню навігації

- список доступних для перегляду тарифів

2.3.3 Системні вимоги

Системні вимоги включають:

- наявність реєстрації та авторизації
- можливість редагування адміністратором тарифів
- створення, редагування, видалення даних про тарифи адміністратором
- можливість перегляду клієнтами тарифів

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до інформаційного забезпечення

Реалізація вебзастосунку здійснюється з використанням:

- Python
- Flask
- CSS3
- HTML5
- JSON

2.4.2 Вимоги до лінгвістичного забезпечення

Вебзастосунок має бути виконаний українською мовою.

2.4.3 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Веб-браузер: Internet Explorer 7.0 і вище, або Firefox 4.5 і вище, або

Safari 3.3.1 і вище, або Chrome 3 і вище.

3 Склад і зміст робіт зі створення вебзастосунку

Докладний опис етапів роботи зі створення вебзастосунку наведено в таблиці А.1.

Таблиця А.1 – Етапи створення вебзастосунку

№	Склад і зміст робіт	Строк розробки
1	Аналіз та планування	4 дні
2	Вивчення аналогів	2 дні
3	Вибір технологічного стеку	3 дні
4	Розробка функціональності	6 днів
5	Розробка структури вебзастосунку	4 дні
6	Розробка бази даних	3 дні
7	Веб-дизайн та UX/UI	5 днів
8	Інтеграція з JSON-файлами	3 дні
9	Комплексне тестування	2 дні
10	Виправлення помилок та вдосконалення	2 дні
	Загальна тривалість робіт	34 дні

4. Вимоги до складу й змісту робіт із введення вебзастосунку в експлуатацію

Для того, щоб вебзастосунком могли користуватися робітники товариства необхідно розмістити його у мережі Інтернет, тому необхідно придбати доменне ім'я та місце на хостингу. На хостинг переноситься вебзастосунок і наповнення бази даних з подальшою їх доробкою. Для коректного переносу вебзастосунку на хостинг необхідно, щоб параметри хостинга відповідали вимогам, зазначеним у ТЗ.

ДОДАТОК Б

ПЛАНУВАННЯ РОБІТ

У сучасному світі мобільні технології відіграють ключову роль у повсякденному житті, надаючи можливість швидкого доступу до інформації, комунікації та різноманітних послуг. Мобільні оператори постійно пропонують нові тарифні плани та акції, що робить вибір оптимального тарифу складним завданням для споживачів.

Розробка вебзастосунку для аналізу тарифів мобільних операторів України має на меті спростити цей процес, забезпечуючи користувачів зручним інструментом для порівняння та вибору тарифних планів відповідно до їхніх потреб. Такий вебзастосунок дозволить користувачам швидко та ефективно знаходити найвигідніші пропозиції на ринку, враховуючи всі актуальні параметри та умови.

Для досягнення успіху в розробці вебзастосунку необхідно чітко визначити початок і завершення завдань, розподілити робочі навантаження, встановити контрольні точки та постійно моніторити їх виконання порівняно з початковим планом. Важливо також коригувати графік за потреби, оптимізувати використання людських і матеріальних ресурсів та інформувати команду про статус проекту, заплановані завдання та строки їх виконання. У цьому допомагає діаграма Ганта (див. рисунок Б.1), яка є важливим інструментом для планування та контролю проекту.

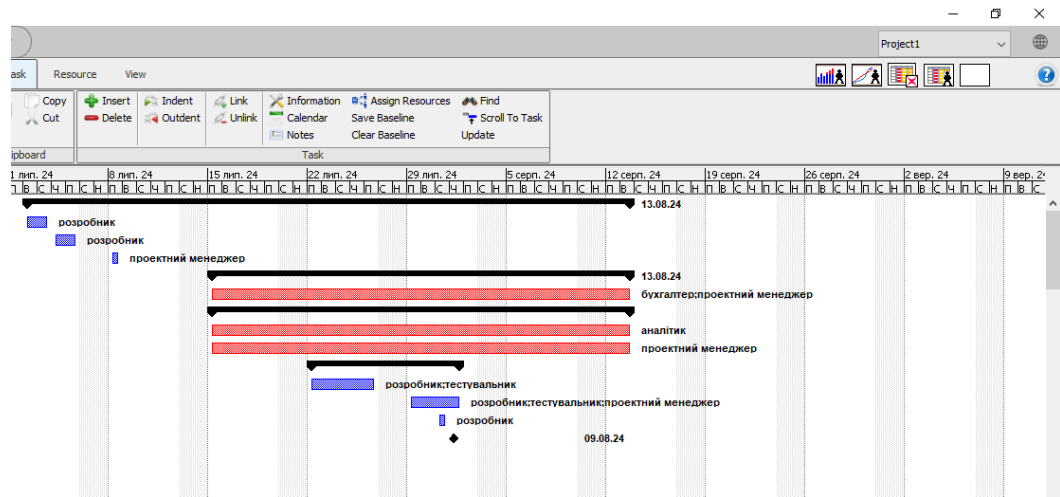


Рисунок Б.1 – Діаграма Ганта

Оцінка ризиків у проєкті створення вебзастосунку для аналізу тарифів мобільних операторів України відіграє ключову роль у забезпеченні успішного виконання проєкту. Вона включає ідентифікацію, аналіз і управління ризиками, що можуть вплинути на досягнення цілей проєкту. Основними компонентами оцінки ризиків є опис ризику, ймовірність виникнення, вплив на проєкт, стратегії управління та відповідальні особи.

Застосування цих компонентів дозволяє команді проєкту своєчасно виявляти потенційні загрози, оцінювати їх серйозність і розробляти заходи для мінімізації або усунення негативних наслідків. Процес оцінки ризиків починається з ідентифікації можливих ризиків, таких як технічні проблеми, затримки у розробці, бюджетні обмеження або зміни у вимогах замовника.

Визначення відповідальних осіб за управління кожним ризиком є важливим аспектом, оскільки це забезпечує чітку відповідальність та своєчасну реалізацію заходів щодо управління ризиками. Регулярний моніторинг та перегляд ризиків протягом усього життєвого циклу проєкту допомагають вчасно виявляти нові ризики та коригувати стратегії управління.

Отже, систематична оцінка ризиків (див. таблицю Б.1) є невід'ємною частиною управління ІТ-проєктами, яка забезпечує їх успішне виконання та досягнення поставлених цілей.

Таблиця Б.1 – Оцінка ризиків ІТ-проєкту.

Ризики	Імовірність виникнення	Вплив на проєкт	Стратегії управління	Відповідальні особи
Втрата даних	Низька	Висока	Регулярне резервне копіювання даних, застосування шифрування	ІТ-фахівець
Відмова обладнання	Низька	Середня	Планування та впровадження запасних/резервних пристроїв, регулярне технічне обслуговування	ІТ-фахівець
Відмова сервера через перевищення навантаження	Середня	Середній	Масштабування серверної інфраструктури, впровадження механізмів автоматичного моніторингу	ІТ-фахівець
Затримка в розробці через зміни вимог	Середня	Середній	Узгодження докладних вимог до функціоналу, впровадження гнучких методологій розробки	Проектний менеджер
Проблеми з безпекою	Низька	Високий	Регулярні аудити безпеки, впровадження стандартів безпеки, навчання персоналу з питань безпеки	ІТ-фахівець
Зміна нормативних вимог	Низька	Високий	Моніторинг законодавства, залучення юридичних консультантів, адаптація до нових вимог	Юридичний консультант
Збої в програмному забезпеченні	Низька	Середній	Регулярна перевірка ПЗ, тестування, планування альтернативних варіантів	ІТ-фахівець

ДОДАТОК В

ЛІСТІНГИ ПРОГРАМ

Сторінка адміністратора:

```
<!DOCTYPE html>
```

```
<html lang="uk">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>Адміністративна панель</title>
```

```
  <link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">
```

```
</head>
```

```
<body>
```

```
  <h1>Адміністративна панель</h1>
```

```
  <a href="/logout">Вийти</a>
```

```
  <h2>Додати оператора</h2>
```

```
  <form action="/add_operator" method="post">
```

```
    <label for="name">Назва:</label>
```

```
    <input type="text" id="name" name="name" required>
```

```
    <button type="submit">Додати оператора</button>
```

```
  </form>
```

```
  <h2>Додати тариф</h2>
```

```
  <form action="/add_tariff" method="post">
```

```
    <label for="operator">Оператор:</label>
```

```
    <select id="operator" name="operator" required>
```

```
      {% for operator in operators %}
```

```
        <option value="{{ operator.name }}">{{ operator.name }}</option>
```

```
      {% endfor %}
```

```

</select>
<label for="plan">План:</label>
<input type="text" id="plan" name="plan" required>
<label for="price">Ціна:</label>
<input type="number" id="price" name="price" required>
<button type="submit">Додати тариф</button>
</form>
<h2>Додати акцію</h2>
<form action="/add_promotion" method="post">
  <label for="operator">Оператор:</label>
  <select id="operator" name="operator" required>
    {% for operator in operators %}
      <option value="{{ operator.name }}">{{ operator.name }}</option>
    {% endfor %}
  </select>
  <label for="promotion">Акція:</label>
  <input type="text" id="promotion" name="promotion" required>
  <label for="date">Дата:</label>
  <input type="date" id="date" name="date" required>
  <button type="submit">Додати акцію</button>
</form>
<h2>Перегляд відгуків</h2>
<a href="{{ url_for('view_feedbacks') }}">Переглянути відгуки</a>
</body>
</html>
Сторінка відгуків:
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">

```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">
<title>Залишити відгук</title>
</head>
<body>
  <div class="container">
    <h1>Залишити відгук</h1>
    <form action="{{ url_for('feedback_form') }}" method="post">
      <label for="name">Ім'я:</label>
      <input type="text" id="name" name="name" required>
      <label for="email">Email:</label>
      <input type="email" id="email" name="email" required>
      <label for="message">Відгук:</label>
      <textarea id="message" name="message" required></textarea>
      <button type="submit">Надіслати</button>
    </form>
    <a href="{{ url_for('home') }}">Повернутися на головну сторінку</a>
  </div>
</body>
</html>
```

Головна сторінка:

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">
  <title>Головна сторінка</title>
</head>
<body>
```

```
<div class="container">
  <h1>Тарифи</h1>
  <table>
    <tr>
      <th>Оператор</th>
      <th>Тариф</th>
      <th>Ціна</th>
    </tr>
    {% for tariff in tariffs %}
    <tr>
      <td>{{ tariff.operator }}</td>
      <td>{{ tariff.plan }}</td>
      <td>{{ tariff.price }}</td>
    </tr>
    {% endfor %}
  </table>
  <h2>Останні акції</h2>
  <table>
    <tr>
      <th>Оператор</th>
      <th>Акція</th>
      <th>Дата</th>
    </tr>
    {% for promotion in promotions %}
    <tr>
      <td>{{ promotion.operator }}</td>
      <td>{{ promotion.promotion }}</td>
      <td>{{ promotion.date }}</td>
    </tr>
    {% endfor %}
```

```

</table>
<h2>Оператори</h2>
<ul>
  {% for operator in operators %}
  <li><a href="{{ url_for('operator_view', name=operator.name) }}">{{
operator.name }}</a></li>
  {% endfor %}
</ul>
<a href="{{ url_for('login') }}">Вхід для адміністраторів</a>
<br>
<a href="{{ url_for('feedback_form') }}">Залишити відгук</a>
</div>
</body>
</html>

```

Сторінка входу:

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>Вхід для адміністраторів</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">
</head>
<body>
  <h1>Вхід для адміністраторів</h1>
  <form action="/login" method="post">
    <label for="username">Ім'я користувача:</label>
    <input type="text" id="username" name="username" required>
    <label for="password">Пароль:</label>
    <input type="password" id="password" name="password" required>
    <button type="submit">Увійти</button>
  </form>

```

```
</form>
</body>
</html>
Сторінка оператора:
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>Оператор {{ name }}</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">
</head>
<body>
  <h1>Тарифи оператора {{ name }}</h1>
  <ul>
    {% for tariff in tariffs %}
    <li>{{ tariff.plan }} - {{ tariff.price }} грн</li>
    {% endfor %}
  </ul>
  <h1>Акції оператора {{ name }}</h1>
  <ul>
    {% for promo in promotions %}
    <li>{{ promo.promotion }} - {{ promo.date }}</li>
    {% endfor %}
  </ul>
  <a href="/">Повернутися на головну сторінку</a>
</body>
</html>
```

Сторінка перегляду відгуків:

```
<!DOCTYPE html>
<html lang="uk">
```

```

<head>
    <meta charset="UTF-8">
    <title>Перегляд відгуків</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">
</head>
<body>
    <h1>Перегляд відгуків</h1>
    <a href="/admin">Повернутися до адміністративної панелі</a>
    <ul>
        {% for feedback in feedbacks %}
        <li>
            <strong>Ім'я:</strong> {{ feedback.name }}<br>
            <strong>Email:</strong> {{ feedback.email }}<br>
            <strong>Повідомлення:</strong> {{ feedback.message }}<br>
        </li>
        {% endfor %}
    </ul>
</body>
</html>

```

Код перегляду програми:

```

from datetime import datetime
from flask import render_template, request, redirect, url_for, session, flash
from TariffAnalysis import app
import json
import os

data_dir = os.path.join(app.root_path, 'data')

def load_data(file_name):

```

```
with open(os.path.join(data_dir, file_name), 'r', encoding='utf-8-sig') as f:
    return json.load(f)
```

```
def save_data(file_name, data):
```

```
    with open(os.path.join(data_dir, file_name), 'w', encoding='utf-8') as f:
        json.dump(data, f, indent=4, ensure_ascii=False)
```

```
@app.route('/')

```

```
@app.route('/home')
```

```
def home():
```

```
    """Renders the home page."""
```

```
    tariffs = sorted(load_data('tariffs.json'), key=lambda x: x['price'])
```

```
    promotions = load_data('promotions.json')
```

```
    operators = load_data('operators.json')
```

```
    latest_promotions = []
```

```
    for operator in operators:
```

```
        operator_promotions = [promo for promo in promotions if promo['operator'] ==
operator['name']]
```

```
        if operator_promotions:
```

```
            latest_promotion = max(operator_promotions, key=lambda x: x['date'])
```

```
            latest_promotions.append(latest_promotion)
```

```
    return render_template(
```

```
        'index.html',
```

```
        title='Home Page',
```

```
        year=datetime.now().year,
```

```
        tariffs=tariffs,
```

```
        promotions=latest_promotions,
```

```
        operators=operators
```

)

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

```
def login():
```

```
    if request.method == 'POST':
```

```
        username = request.form['username']
```

```
        password = request.form['password']
```

```
        admins = load_data('admins.json')
```

```
        admin = next((admin for admin in admins if admin['username'] == username and
admin['password'] == password), None)
```

```
        if admin:
```

```
            session['admin'] = username
```

```
            return redirect(url_for('admin'))
```

```
        else:
```

```
            flash('Невірні облікові дані')
```

```
    return render_template('login.html')
```

```
@app.route('/logout')
```

```
def logout():
```

```
    session.pop('admin', None)
```

```
    return redirect(url_for('home'))
```

```
@app.route('/admin')
```

```
def admin():
```

```
    if 'admin' not in session:
```

```
        return redirect(url_for('login'))
```

```
    operators = load_data('operators.json')
```

```
    tariffs = load_data('tariffs.json')
```

```
    promotions = load_data('promotions.json')
```

```
    return render_template('admin.html', operators=operators, tariffs=tariffs,
```

```
promotions=promotions)
```

```
@app.route('/operator')
```

```
def select_operator():
```

```
    operator_name = request.args.get('operator_name')
```

```
    if operator_name:
```

```
        return redirect(url_for('operator_view', name=operator_name))
```

```
    return redirect(url_for('home'))
```

```
@app.route('/operator/<name>')
```

```
def operator_view(name):
```

```
    tariffs = load_data('tariffs.json')
```

```
    promotions = load_data('promotions.json')
```

```
    operator_tariffs = [tariff for tariff in tariffs if tariff['operator'] == name]
```

```
    operator_promotions = [promo for promo in promotions if promo['operator'] ==  
name]
```

```
    return render_template('operator.html', name=name, tariffs=operator_tariffs,  
promotions=operator_promotions)
```

```
@app.route('/add_tariff', methods=['POST'])
```

```
def add_tariff():
```

```
    new_tariff = request.form.to_dict()
```

```
    new_tariff['price'] = int(new_tariff['price']) # Перетворюємо ціну на числовий тип
```

```
    tariffs = load_data('tariffs.json')
```

```
    tariffs.append(new_tariff)
```

```
    save_data('tariffs.json', tariffs)
```

```
    return redirect(url_for('admin'))
```

```
@app.route('/add_operator', methods=['POST'])
```

```
def add_operator():
```

```
new_operator = request.form.to_dict()
operators = load_data('operators.json')
operators.append(new_operator)
save_data('operators.json', operators)
return redirect(url_for('admin'))
```

```
@app.route('/add_promotion', methods=['POST'])
```

```
def add_promotion():
```

```
    new_promotion = request.form.to_dict()
    promotions = load_data('promotions.json')
    promotions.append(new_promotion)
    save_data('promotions.json', promotions)
    return redirect(url_for('admin'))
```

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

```
def feedback_form():
```

```
    if request.method == 'POST':
        feedback = request.form.to_dict()
        feedback_list = load_data('feedback.json')
        feedback_list.append(feedback)
        save_data('feedback.json', feedback_list)
        return redirect(url_for('home'))
    return render_template('feedback.html')
```

```
@app.route('/admin/feedbacks')
```

```
def view_feedbacks():
```

```
    if 'admin' not in session:
        return redirect(url_for('login'))
    feedback_list = load_data('feedback.json')
    return render_template('view_feedbacks.html', feedbacks=feedback_list)
```

Код запуску Flask:

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
app.secret_key = 'supersecretkey'
```

```
import TariffAnalysis.views
```

Код стиля:

```
body {
```

```
    font-family: Arial, sans-serif;
```

```
    background-color: #f4f4e8;
```

```
    color: #333;
```

```
    text-align: center;
```

```
    padding: 10px;
```

```
}
```

```
h1, h2 {
```

```
    color: #4f7942;
```

```
}
```

```
.container {
```

```
    width: 80%;
```

```
    margin: 0 auto;
```

```
}
```

```
.form-control {
```

```
    display: block;
```

```
    width: 100%;
```

```
    padding: 10px;
```

```
    margin-bottom: 10px;
```

```
border-radius: 5px;
border: 1px solid #ccc;
}
```

```
.btn {
  background-color: #4f7942;
  color: white;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
```

```
.btn:hover {
  background-color: #3e5934;
}
```

```
table {
  width: 100%;
  border-collapse: collapse;
  margin: 20px 0;
}
```

```
table, th, td {
  border: 1px solid #ddd;
}
```

```
th, td {
  padding: 10px;
  text-align: left;
```

```
}
```

```
th {
```

```
    background-color: #4f7942;
```

```
    color: white;
```

```
}
```

```
a {
```

```
    color: #4f7942;
```

```
    text-decoration: none;
```

```
}
```

```
    a:hover {
```

```
        text-decoration: underline;
```

```
    }
```

Код файлу запусків:

```
from flask import Flask, render_template, request, redirect, url_for, session, flash
```

```
import json
```

```
import os
```

```
from datetime import datetime
```

```
app = Flask(__name__)
```

```
app.secret_key = 'supersecretkey' # Секретний ключ для сесій
```

```
data_dir = os.path.join(app.root_path, 'data')
```

```
def load_data(file_name):
```

```
    with open(os.path.join(data_dir, file_name), 'r', encoding='utf-8-sig') as f:
```

```
        return json.load(f)
```

```
def save_data(file_name, data):
```

```
with open(os.path.join(data_dir, file_name), 'w', encoding='utf-8') as f:
    json.dump(data, f, indent=4, ensure_ascii=False)
```

```
@app.route('/')
```

```
@app.route('/home')
```

```
def home():
```

```
    """Renders the home page."""
```

```
    tariffs = sorted(load_data('tariffs.json'), key=lambda x: x['price'])
```

```
    promotions = load_data('promotions.json')
```

```
    operators = load_data('operators.json')
```

```
    latest_promotions = []
```

```
    for operator in operators:
```

```
        operator_promotions = [promo for promo in promotions if promo['operator'] ==
operator['name']]
```

```
        if operator_promotions:
```

```
            latest_promotion = max(operator_promotions, key=lambda x: x['date'])
```

```
            latest_promotions.append(latest_promotion)
```

```
    return render_template(
```

```
        'index.html',
```

```
        title='Home Page',
```

```
        year=datetime.now().year,
```

```
        tariffs=tariffs,
```

```
        promotions=latest_promotions,
```

```
        operators=operators
```

```
    )
```

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

```
def login():
```

```
if request.method == 'POST':
    username = request.form['username']
    password = request.form['password']
    admins = load_data('admins.json')
    admin = next((admin for admin in admins if admin['username'] == username and
admin['password'] == password), None)
    if admin:
        session['admin'] = username
        return redirect(url_for('admin'))
    else:
        flash('Невірні облікові дані')
return render_template('login.html')
```

```
@app.route('/logout')
```

```
def logout():
```

```
    session.pop('admin', None)
    return redirect(url_for('home'))
```

```
@app.route('/admin')
```

```
def admin():
```

```
    if 'admin' not in session:
        return redirect(url_for('login'))
    operators = load_data('operators.json')
    tariffs = load_data('tariffs.json')
    promotions = load_data('promotions.json')
    return render_template('admin.html', operators=operators, tariffs=tariffs,
promotions=promotions)
```

```
@app.route('/operator/<name>')
```

```
def operator_view(name):
```

```
tariffs = load_data('tariffs.json')
promotions = load_data('promotions.json')
operator_tariffs = [tariff for tariff in tariffs if tariff['operator'] == name]
operator_promotions = [promo for promo in promotions if promo['operator'] ==
name]

return render_template('operator.html', name=name, tariffs=operator_tariffs,
promotions=operator_promotions)
```

```
@app.route('/add_tariff', methods=['POST'])
```

```
def add_tariff():
```

```
    new_tariff = request.form.to_dict()
    tariffs = load_data('tariffs.json')
    tariffs.append(new_tariff)
    save_data('tariffs.json', tariffs)
    return redirect(url_for('admin'))
```

```
@app.route('/add_operator', methods=['POST'])
```

```
def add_operator():
```

```
    new_operator = request.form.to_dict()
    operators = load_data('operators.json')
    operators.append(new_operator)
    save_data('operators.json', operators)
    return redirect(url_for('admin'))
```

```
@app.route('/add_promotion', methods=['POST'])
```

```
def add_promotion():
```

```
    new_promotion = request.form.to_dict()
    promotions = load_data('promotions.json')
    promotions.append(new_promotion)
    save_data('promotions.json', promotions)
```

```
return redirect(url_for('admin'))
```

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```