

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет комп'ютерних наук, фізики та математики
Кафедра комп'ютерних наук та програмної інженерії**

**Програмне середовище навчального призначення з теми
«Задача про найкоротші шляхи у орієнтованому графі»**

Кваліфікаційна робота (проєкт)

на здобуття ступеня вищої освіти «магістр»

Виконав: здобувач 2 курсу 231М групи
Спеціальності: 122 Комп'ютерні науки
Освітньо-професійної програми: «Комп'ютерні науки»

Черниш Ярослав Миколайович

Керівник: – доктор фізико-математичних наук,
професор Львов М.С.

Рецензент: Кузьмич В.І. професор кафедри
алгебри, геометрії та математичного аналізу
Херсонського державного університету

Івано-Франківськ – 2023

АНОТАЦІЯ

Пояснювальна записка: 100 с., 20 рисунків, 10 таблиць, 18 джерел.

Об'єкт розробки: Програмне середовище навчального призначення з теми «Задача про найкоротші шляхи у орієнтованому графі».

Предмет дослідження : веб – орієнтовані програмні засоби для проектування та розробки програмного середовища навчального призначення.

Мета дипломного проекту: розробка веб-додатку для автоматизації діяльності процесів надання інформації навчального призначення з теми «Задача про найкоротші шляхи у орієнтованому графі».

У першому розділі наведено теоретичні аспекти предметної галузі, надано визначення задачі про найкоротші шляхи, наведено основні поняття задачі про найкоротші шляхи, обумовлено актуальність розв'язання поставленої задачі та алгоритми вирішення завдання.

У другому розділі проведено аналіз та визначення інструментів для вирішення поставленої задачі, проведено аналіз та порівняльну характеристика технологій для розробки проектованого ПЗ, зроблено вибір інструментів для вирішення поставленого завдання на основі проведеного аналізу.

У третьому розділі проведено практичну реалізацію проектованої системи, надано інформаційне, технічне та системне забезпечення розробки.

Актуальність розробки веб-сервісу визначається великим попитом на подібні системи, оскільки автоматизація робить процес навчання більш гнучким, зрозумілим, відкритим й стандартизованим, а також надає значні переваги : мобільність, безконтактність, ціна, масштабованість і т.ін.

Список ключових слів: АВТОМАТИЗАЦІЯ, ВЕБ-ДОДАТОК, БАЗА ДАНИХ, PHP, HTML, CSS,MYSQL.

ABSTRACT

Explanatory note: 90 p., 20 figures, 10 tables, 18 sources.

Object of development: Software environment for educational purposes on the topic "Problem of the shortest paths in a directed graph".

Research subject: web-oriented software tools for designing and developing educational software environment.

The goal of the diploma project: the development of a web application for automating the activity of the processes of providing educational information on the topic "The problem of the shortest paths in a directed graph".

In the first chapter, the theoretical aspects of the subject area are given, the definition of the problem of the shortest paths is given, the basic concepts of the problem of the shortest paths are given, the relevance of the solution of the given problem and the algorithms of the problem are determined.

In the second section, the analysis and definition of tools for solving the given task was carried out, the analysis and comparative characteristics of technologies for the development of the designed software were carried out, the selection of tools for solving the given task was made based on the analysis.

In the third section, practical implementation of the designed system was carried out, informational, technical and system development support was provided.

The relevance of web service development is determined by the great demand for such systems, since automation makes the learning process more flexible, understandable, open and standardized, and also provides significant advantages: mobility, non-contact, price, scalability, etc.

List of keywords: AUTOMATION, WEB APPLICATION, DATABASE, PHP, HTML, CSS, MYSQL.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ

ПК – персональний комп'ютер

HTML (Hyper Text Markup Language) – мова розмітки гіпертекстових документів

CSS (Cascading Style Sheets) – каскадна таблиця стилів

PHP - скриптова мова загального призначення, що інтенсивно використовується для розробки веб-додатків. В даний час підтримується переважною більшістю хостинг-провайдерів і є одним з лідерів серед мов, що застосовуються для створення динамічних веб-сайтів

MySQL - ободня реляційна система управління базами даних. Розробку та підтримку MySQL здійснює корпорація Oracle.

ЗМІСТ

ВСТУП.....	10
1. ТЕОРЕТИЧНІ АСПЕКТИ ПРЕДМЕТНОЇ ГАЛУЗІ.....	12
1.1 Визначення задачі про найкоротші шляхи.....	12
1.2 Основні поняття задачі про найкоротші шляхи.....	13
1.3 Актуальність розв’язання поставленої задачі.....	15
1.4 Алгоритми вирішення завдання.....	17
1.4.1 Алгоритм Дейкстри.....	17
1.4.2 Алгоритм Белмана-Форда.....	20
1.4.3 Алгоритм Флойда-Уоршелла.....	23
2. АНАЛІЗ ТА ВИЗНАЧЕННЯ ІНСТРУМЕНТІВ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕННОЇ ЗАДАЧІ.....	27
2.1 Аналіз існуючих варіантів вирішення поставленого завдання.....	27
2.2 Порівняльна характеристика технологій для розробки проектного програмного продукту.....	35
2.3 Аналіз існуючих СУБД для вирішення поставленої задачі.....	38
2.4 Порівняльна характеристика можливих СУБД для використання в розробці програмного середовища навчального призначення з теми «Задача про найкоротші шляхи у орієнтованому графі».....	42
2.5 Вибір інструментів для вирішення поставленого завдання на основі проведеного аналізу.....	45
3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЕКТОВАНОГО ПРОГРАМНОГО ПРОДУКТУ.....	46
3.1. Вимоги до програмного продукту.....	46
3.1.1. Вимоги до інформаційної безпеки.....	46
3.1.2. Вимоги до складу та параметрів технічних засобів.....	47
3.1.3. Вимоги до інформаційної та програмної сумісності.....	48
3.2 Структура програмного продукту.....	49
3.3 Проектування архітектури розроблюваної системи.....	51

3.4	Проектування структури бази даних.....	52
3.5	Програмна реалізація головної сторінки.....	54
3.6	Програмна реалізація сторінки обраного алгоритму.....	58
3.7	Програмна реалізація сторінки відгуків.....	62
3.8	Програмна реалізація сторінки реєстрації.....	64
3.9	Програмна реалізація сторінки авторизації.....	67
3.10	Розробка інтерфейсу адміністрування системи.....	69
ВИСНОВОК.....		73
СПИСОК ЛІТЕРАТУРИ.....		74
ДОДАТКИ.....		76

ВСТУП

Автоматизація процесів навчання – завдання на сьогоднішній день відоме та поширене. У сучасному світі не одна галузь не може обійтися без інтеграції в свою роботу інформаційної системи в якомусь її представленні, оскільки це істотно підвищує як зручність і якість взаємодії між користувачами, так і значно підвищує охоплення клієнтської аудиторії та можливості взаємодії з нею.

Можна говорити про велику кількість успішно впроваджених проектів у різних галузях, у різних за масштабом компаніях, з застосуванням різних засобів впровадження процесів автоматизації. Коли ми говоримо про розробку програмного середовища навчального призначення з теми «Задача про найкоротші шляхи у орієнтованому графі», ми маємо на увазі розробку веб - додатку для автоматизованого надання тематичних навчальних матеріалів студентам з можливостями взаємодіяти з ними шляхом перевірки рівню вивчення отриманого матеріалу. Фактично у наш час поняття веб – додатку стало невід'ємною частиною процесів отримання освіти. Результати аналізу існуючих даних свідчать, що нині більшість як великих вузів, так і невеликих локальних тією чи іншою мірою переглянули свій підхід до структури та процесів керування процесами надання знань у бік автоматизації.

Метою кваліфікаційної роботи є удосконалення процесу розробки програмного середовища навчального призначення з теми «Задача про найкоротші шляхи у орієнтованому графі» шляхом розробки та впровадження Web-сервісу з для автоматизованої взаємодії студентів з навчальним матеріалом.

Для досягнення мети в роботі поставлено й вирішено такі теоретичні й практичні **завдання**:

- Провести аналіз предметної газулі для визначення актуальної інформації для надання у додатку.
- Провести аналіз необхідних матеріалів для роботи з мовами програмування.
- Провести аналіз необхідних матеріалів для роботи з базами даних.
- Визначити архітектуру веб – додатку.
- Розробити інтерфейс користувача додатку.
- Розробити бекенд частину додатку для взаємодії з базою даних
- Провести тестування системи
- Провести аналіз результатів роботи

Об’єктом дослідження є програмне середовище навчального призначення з теми «Задача про найкоротші шляхи у орієнтованому графі».

Предметом дослідження є веб – орієнтовані програмні засоби для проектування та розробки програмного середовища навчального призначення.

Методи дослідження. У процесі роботи застосовувались загальнонаукові та спеціальні методи, які дозволили вивчити предмет та об’єкт дослідження, дослідити напрями та шляхи оптимізації процесу проектування та розробки програмного середовища навчального призначення.

Практичне значення одержаних результатів. Практичне значення веб-сервісу визначається великим попитом на подібні системи, оскільки автоматизація робить процес навчання більш гнучким, зрозумілим, відкритим й стандартизованим, а також надає значні переваги : мобільність, безконтактність, ціна, масштабованість і т.ін.

РОЗДІЛ 1

ТЕОРЕТИЧНІ АСПЕКТИ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Визначення задачі про найкоротші шляхи

Задача про найкоротший шлях є одним із найважливіших класичних завдань теорії графів. Граф - математична модель, представлена сукупністю безлічі вершин та зв'язків між ними. Вона може бути визначена для неорієнтованого, орієнтованого чи змішаного графа. У наведеній роботі завдання вирішується у межах орієнтованого графа.

Завдання пошуку найкоротшого шляху є класичним завданням теорії графів – розділу дискретної математики, що займається вивченням якості графів. Цей розділ знаходить застосування у багатьох сферах нашого життя: у геоінформаційних системах, у будівництві доріг, мережевих комунікаціях та багатьох інших.

Історія виникнення теорії графів бере свій початок із завдання про Кенігсберзькі мости - старовинної математичної задачі, в якій питалося, як можна пройти по всіх семи мостах Кенігсберга, не проходячи ні по одному з них двічі. Дане завдання вперше було вирішено німецьким математиком Леонардом Ейлером у 1736 році. У своєму листі до італійського математика та інженера Мариноні від 13 березня 1736 він формулює основні висновки теорії графів, що і послужило початком цієї математичної дисципліни.

Однак ця стаття Ейлера була єдиною майже сто років. Інтерес до розвитку теорії графів відродився межі XIX-XX століть. Було безліч причин для такого поживлення вивчення графів. Природні науки вплинули на це завдяки дослідженню електричних мереж і молекулярних схем. Також

поштовх до розвитку теорія графів отримала, коли різко зросла кількість робіт у галузі топології та комбінаторики.

1.2 Основні поняття задачі про найкоротші шляхи

Предметом перших завдань у теорії графів були різні зміни, що складаються з точок та ліній, що їх з'єднують. З цього опису можна скласти математичне визначення графа [1]:

$$G = G(V),$$

де G - граф з безліччю вершин V і деяким сімейством поєднань (пар) виду

$$E = (a_i, a_{i+1}), \quad a \in V,$$

що вказує, які вершини вважаються з'єднаними. Відповідно до геометричного подання графа кожна конкретна пара називається ребром графа, вершини a_i і a_{i+1} якого називаються кінцями ребра E . Вершини a_i і a_{i+1} інцидентні E , а також ребро E інцидентно вершинам a_i і a_{i+1} . Дві вершини (або два ребра) інцидентними бути не можуть за визначенням.

Важливо враховувати, чи береться до уваги розташування двох кінців ребра: якщо цей порядок не суттєвий, то ребро вважається неорієнтованим, інакше орієнтованим. Граф називається неорієнтованим, якщо кожне його ребро не орієнтоване, і орієнтованим, якщо всі його ребра орієнтовані (рисунок 1.1).

Граф називається позначеним (або перенумерованим), якщо його вершини відрізняються одна від одної будь-якими позначками.

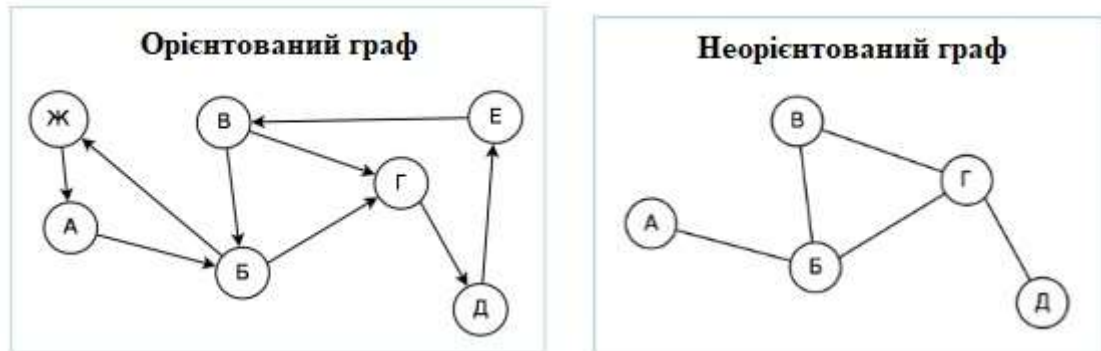


Рисунок 1.1 - Приклад перенумерованих графів: орієнтованого та неорієнтованого

Нерідко актуально розглядати змішані графи, мають як орієнтовані, і неорієнтовані ребра.

Виваженим називається граф, кожному ребру якого надано значення (вага ребра). Зазвичай вага - речове число, в такому випадку його можна інтерпретувати як «довжину» ребра. [2]

Шлях (маршрут) на графі визначається послідовністю вершин та ребер

$$a_1 E_1 a_2 E_2 a_3 \dots a_i E_i a_{i+1},$$

де a_i - вершини, $E_i = (a_i, a_{i+1})$ – ребро.

Для орієнтованого графа існує два типи маршрутів: неорієнтований та орієнтований. Неорієнтованим маршрутом називається послідовність вершин a_1, a_2, \dots, a_i така, що для кожного $i = 1, 2, \dots, n-1$ хоча б одне з ребер (a_i, a_{i+1}) , (a_{i+1}, a_i) належить графу. Маршрут називається орієнтованим, якщо для кожного i пара (a_i, a_{i+1}) є ребром графа. Таким чином, при русі неорієнтованим маршрутом в орієнтованому графі ребра можуть

проходиться як у напрямку орієнтації, так і в зворотному, а при русі вздовж орієнтованого маршруту - тільки за напрямом орієнтації ребра.

Відповідно до двох типів маршрутів визначаються і два типи зв'язності орієнтованих графів. Орграф називається зв'язковим (або слабо зв'язковим), якщо для кожної пари вершин у ньому є маршрут, що з'єднує їх, він називається сильно зв'язковим, якщо для кожної впорядкованої пари вершин (a_i, a_{i+1}) у ньому є орієнтований маршрут, що веде з a_i до a_{i+1} .

У загальному випадку задачу найкоротшого шляху можна сформулювати наступним чином: дано зважений орієнтований граф $G = (V, E)$ з ваговою функцією $w = E \rightarrow R$, що відображає ребра на їх ваги, значення яких виражаються дійсними числами. Нам необхідно знайти мінімальне значення ваги шляху $p = \langle v_0, v_1, \dots, v_k \rangle$ між потрібними нам вершинами, що дорівнює сумарній вазі окремих ребер, що входять до нього:

$$w(p) = \sum_{i=1}^{\infty} w(v_{i-1}, v_i)$$

Сформульовані вище визначення для графа достатні для виконання багатьох завдань, зокрема і завдання найкоротшого шляху, яке реалізується у даній роботі. [3]

1.3 Актуальність розв'язання поставленої задачі

Проблема знаходження найкоротших шляхів залишається актуальною і важливою в сучасному світі з численними практичними застосуваннями. Наведемо кілька сфер, де ця проблема є актуальною:

- Транспорт і маршрутизація: Системи навігації та маршрутизації в автомобільних навігаторах та мобільних додатках допомагають водіям

- знаходити найшвидший або найкоротший шлях до пункту призначення, що зменшує час та паливо, які витрачаються на подорож.
- Транспортні системи міст: Міські влади та транспортні компанії використовують алгоритми найкоротших шляхів для планування маршрутів громадського транспорту, що допомагає пасажиром знаходити швидкий та зручний спосіб дістатися до своїх місць призначення.
 - Логістика та постачання: Компанії, які займаються логістикою та постачанням, використовують алгоритми найкоротших шляхів для оптимізації маршрутів доставки товарів та зменшення витрат на транспортування.
 - Мережі телекомунікацій: У сфері телекомунікацій важливо знаходити найкоротший шлях для передачі даних між вузлами мережі. Це допомагає зменшити затримки та покращити продуктивність мережі.
 - Біологічні дослідження: В біології і генетиці аналіз генетичних мереж та інших біологічних систем може включати пошук найкоротших шляхів для подорожування інформації в клітині або організмі.

У сучасному світі розвиток геопросторових технологій та аналітичних методів дозволяє нашим системам навігації та маршрутизації працювати більш ефективно та надавати користувачам більше можливостей для оптимізації своїх маршрутів, що робить проблему знаходження найкоротших шляхів надзвичайно актуальною в сучасному світі. Завдяки розвитку технологій, таких як географічні інформаційні системи (ГІС), комп'ютерне зорове розпізнавання та штучний інтелект, можливості знаходження найкоротших шляхів стали більш точними і швидкими, що робить їх актуальними для багатьох галузей життя.

1.4 Алгоритми вирішення завдання

Існує безліч різних постановок задачі, які враховують такі параметри як наявність негативних ваг, необхідність підраховувати не тільки найкоротший шлях, а й ребра, що входить до його складу, і т.п. Наведемо найбільш популярні алгоритми розв'язання поставленого завдання:

- Алгоритм Дейкстри (знаходить найкоротший шлях від однієї з вершин графа до всіх інших; алгоритм працює тільки для графів без ребер негативної ваги)
- Алгоритм Белмана-Форда (знаходить найкоротші шляхи від однієї вершини графа до решти у зваженому графі; вага ребер може бути негативним)
- Алгоритм Флойда-Уоршелла (знаходить найкоротші шляхи між усіма вершинами зваженого орієнтованого графа).

Розглянемо детальніше кожен за наведених алгоритмів.

1.4.1 Алгоритм Дейкстри

Алгоритм Дейкстри по праву вважається одним з найбільш ефективних алгоритмів розв'язання задачі віднайти найкоротший шлях у графі для випадку негативних ваг дуг. Він був розроблений нідерландським математиком Едсгером Дейстром у 1959 році. Даний алгоритм широко застосовується у програмуванні, а також у протоколах маршрутизації (таких як OSPF та IS-IS).

Суть алгоритму полягає в наступному: зіставимо кожній вершині мітку – мінімальна відома відстань від даної вершини до a . Алгоритм працює крок за кроком, на кожному наступному кроці відвідуючи одну з вершин, намагаючись зменшити величину мітки. Робота алгоритму завершиться, коли

всі вершини будуть відвідані. Перша мітка, з якої починається алгоритм, відповідає 0, а мітки інших вершин - нескінченності (мітка нескінченності означає, що відстань цієї вершини поки невідома).

До того моменту, коли всі вершини виявляться відвіданими, вибиратиметься деяка з ще не відвіданих вершин, що має мінімальну мітку. Якщо всі відвідані вершини, алгоритм завершується. В іншому випадку з ще не відвіданих вершин вибирається вершина u , що має мінімальну мітку.

Вершини, які ведуть ребра з u , називаються сусідами цієї вершини. Для кожного сусіда вершини u , крім зазначених як відвідані, розглядається нова довжина шляху, що дорівнює сумі значень поточної мітки u та довжини ребра, що з'єднує u з цим сусідом. Якщо отримане значення довжини менше значення мітки сусіда, значення мітки замінюється на отримане значення довжини. Розглянувши всіх сусідів, вершина u позначається як відвідана і крок алгоритму повторюється.

Розглянемо переваги та недоліки наведеного алгоритму.

Переваги:

- 1) Ефективність для додатних ваг ребер: Алгоритм Дейкстри працює дуже ефективно, коли всі ваги ребер позитивні. Його часова складність в цьому випадку становить $O(V^2)$, де V - кількість вершин у графі. Це робить його досить швидким для більшості практичних ситуацій.
- 2) Локальність: Алгоритм Дейкстри локальний, тобто він розглядає вершини і ребра тільки в межах певної області навколо поточної вершини, що робить його ефективним для обчислень на певному відрізку маршруту.
- 3) Оптимальність для додатних ваг ребер: Якщо всі ваги ребер не від'ємні, алгоритм Дейкстри гарантує знаходження оптимальних найкоротших шляхів.

Недоліки:

- 1) Неспроможність обробляти ваги ребер від'ємні або нульові: Алгоритм Дейкстри не може обробляти ребра з від'ємними або нульовими вагами, оскільки в цьому випадку алгоритм не збудує коректні найкоротші шляхи.
- 2) Неефективність для графів із багатою кількістю вершин та ребер: Для графів із великою кількістю вершин і ребер алгоритм Дейкстри може стати неефективним через квадратичну часову складність.
- 3) Використання локальних структур даних: Алгоритм Дейкстри вимагає використання локальних структур даних, таких як пріоритетна черга, що може бути неефективним для деяких застосувань.
- 4) Відсутність обробки циклів: Алгоритм Дейкстри не може обробляти графи з циклами, оскільки може призвести до нескінченності під час пошуку найкоротших шляхів.

Узагальнюючи, алгоритм Дейкстри є ефективним та оптимальним для графів з додатними вагами ребер, але він має обмеження у відношенні до роботи з вагами ребер, які можуть бути від'ємними або нульовими, і неефективний для великих графів. Вибір алгоритму повинен визначатися конкретними вимогами вашого завдання та властивостями вашого графа.

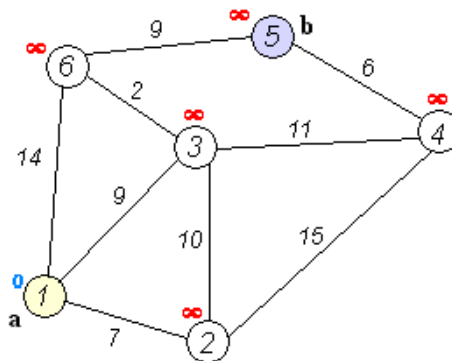


Рисунок 1.2 – Алгоритм Дейкстри

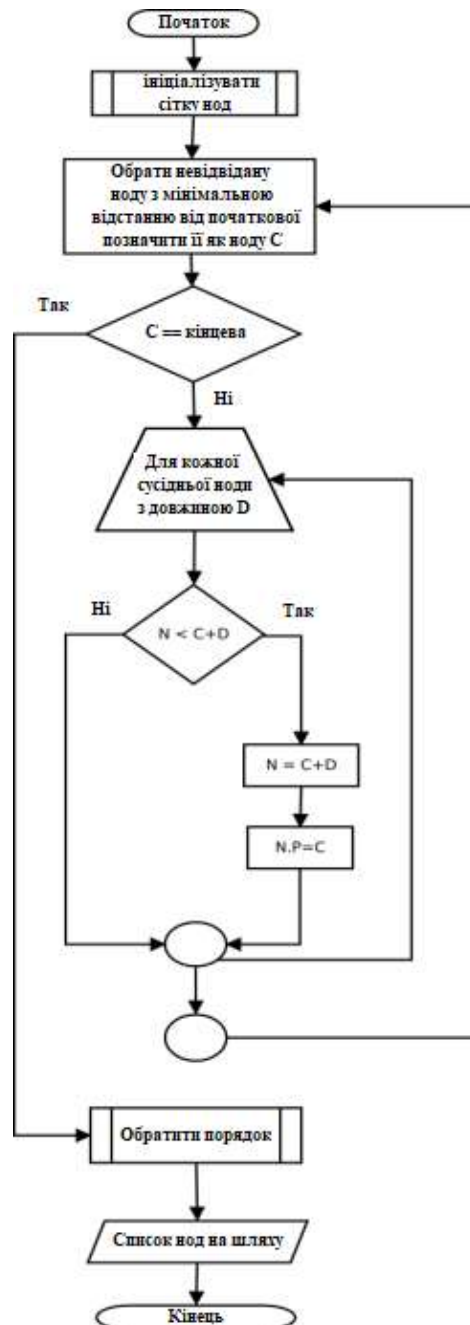


Рисунок 1.3 – Блок - схема роботи алгоритму Дейкстри

1.4.2 Алгоритм Белмана-Форда

Алгоритм Белмана-Форда - це алгоритм для знаходження найкоротших шляхів в напрямленому або ненапрямленому зваженому графі, включаючи графи з від'ємними вагами ребер (але без циклів від'ємної ваги, які доступні з вершини старту). Цей алгоритм найчастіше використовується

для знаходження найкоротших шляхів в графах, де алгоритм Дейкстри не підходить через ваги ребер.

Основна ідея алгоритму полягає в тому, що він починає з вершини-джерела і послідовно розширює зону впливу для знаходження коротших шляхів до всіх інших вершин. Алгоритм працює ітеративно та повторюється $V-1$ разів, де V - кількість вершин у графі.

Основні кроки алгоритму Беллмана-Форда:

- 1) Ініціалізувати відстані для всіх вершин як "нескінченні" (крім вершини-джерела, яка має відстань 0).
- 2) Провести $V-1$ ітерацій. На кожній ітерації переглянути всі ребра графу і оновити відстані до сусідніх вершин, якщо можливо скоротити шлях через поточну вершину.
- 3) Після $V-1$ ітерацій отримаємо коротші відстані від вершини-джерела до всіх інших вершин.

Алгоритм Беллмана-Форда також використовує перевірку на наявність циклів від'ємної ваги. Якщо на останній ітерації відстані до вершин продовжують зменшуватися, то це означає наявність циклу від'ємної ваги в графі.

Алгоритм Беллмана-Форда є корисним і важливим інструментом для розв'язання різних задач, і він має декілька важливих властивостей і застосувань:

- Негативні ваги ребер: Однією з основних переваг алгоритму Беллмана-Форда є можливість обробки графів з ребрами негативної ваги. Він може розв'язувати задачі, де шляхи мають вагу як позитивну, так і негативну.
- Визначення наявності циклів негативної ваги: Алгоритм Беллмана-Форда може бути використаний для виявлення наявності циклів негативної ваги в графі. Якщо після $V-1$ ітерацій відстані до вершин

продовжують зменшуватися, то це означає наявність циклу негативної ваги.

- Знаходження найкоротших шляхів: Алгоритм знаходить найкоротші шляхи від вершини-джерела до всіх інших вершин в графі, навіть враховуючи ваги ребер. Це корисно для задач, пов'язаних з маршрутизацією, логістикою та іншими областями.
- Робота з графами, які мають цикли: Іноді графи можуть мати цикли або зв'язані компоненти, які не доступні з вершини-джерела за допомогою алгоритму Дейкстри. В цьому випадку алгоритм Беллмана-Форда залишається корисним.

Однак важливо враховувати, що алгоритм Беллмана-Форда має часову складність $O(VE)$, де V - кількість вершин, E - кількість ребер. Це робить його менш ефективним для великих графів порівняно з алгоритмом Дейкстри, який має складність $O(V^2)$ або $O(V \log(V))$ залежно від реалізації. Тому важливо вибирати алгоритм відповідно до конкретних потреб і властивостей наведеного графа.

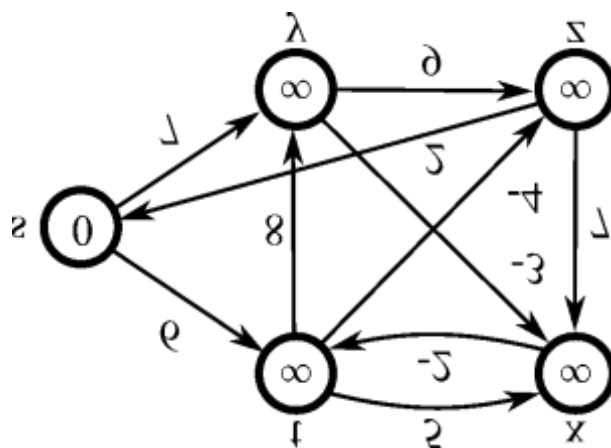


Рисунок 1.4 – Алгоритм Белмана - Форда

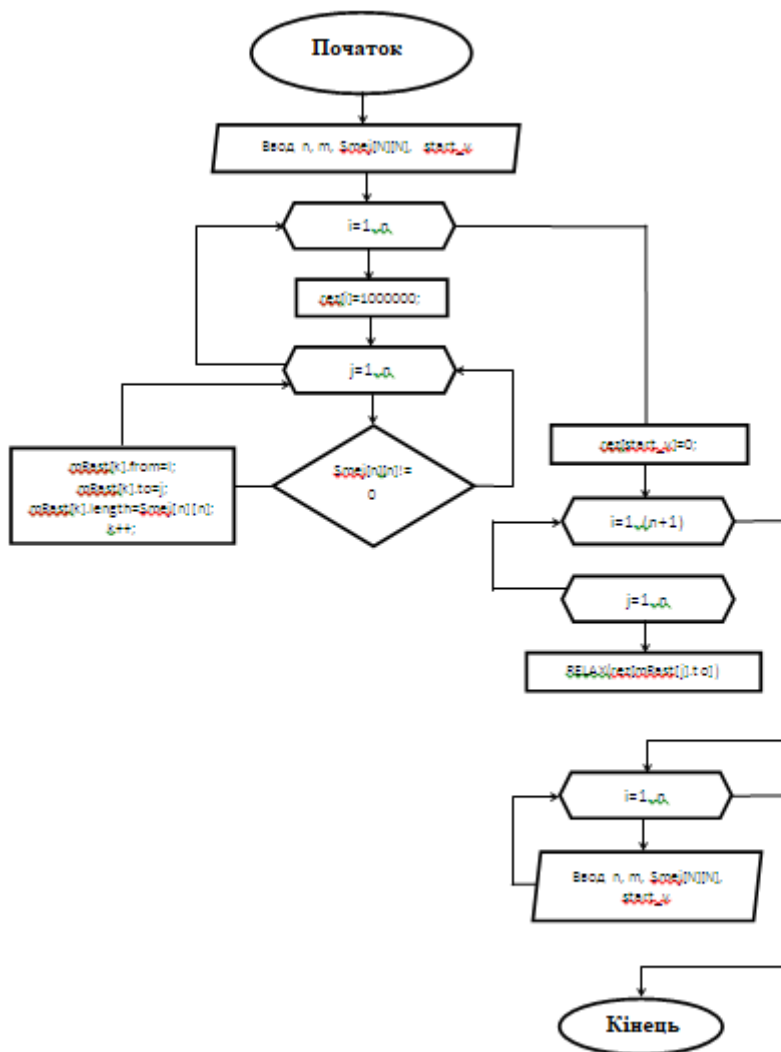


Рисунок 1.5 – Блок - схема роботи алгоритму Белмана - Форда

1.4.3 Алгоритм Флойда-Уоршелла

Алгоритм Флойда-Уоршелла - це класичний алгоритм для знаходження найкоротших шляхів між всіма парами вершин у напрямленому або ненаправленому зваженому графі. Цей алгоритм дозволяє знайти найкоротші шляхи між всіма вершинами графу навіть у випадку наявності ваг ребер і циклів негативної ваги. Важливо зазначити, що він підходить для графів з великою кількістю вершин.

Основні кроки алгоритму Флойда-Уоршелла:

- Ініціалізація матриці відстаней: Спочатку ініціалізується матриця відстаней D , де $D[i][j]$ - це відстань між вершинами i і j . Початкові значення встановлюються рівними вагам ребер, але якщо немає прямого зв'язку між вершинами, то вони ініціалізуються як "нескінченні".
- Ітерації для знаходження найкоротших шляхів: Проводяться ітерації для кожної вершини як проміжної вершини (k) між всіма парами вершин (i, j). У кожній ітерації матриця відстаней D оновлюється наступним чином:

$$D[i][j] = \min(D[i][j], D[i][k] + D[k][j]),$$

де $\min()$ обирає мінімальне значення між поточним значенням $D[i][j]$ і сумою $D[i][k]$ і $D[k][j]$. Це означає, що алгоритм шукає шляхи, які проходять через вершину k і можуть бути коротшими за поточні значення.

- Завершення ітерацій: Після завершення всіх ітерацій отримуємо оновлену матрицю D , де $D[i][j]$ містить найкоротший шлях між вершинами i та j .
- Виявлення від'ємних циклів: Якщо після всіх ітерацій буде знайдено $D[i][i] < 0$ для хоча б однієї вершини i , це вказує на наявність в графі від'ємних циклів, і алгоритм Флойда-Уоршелла не може правильно знайти найкоротші шляхи.

Наведемо переваги та обмеження цього алгоритму.

Переваги:

- 1) Універсальність: Алгоритм Флойда-Уоршелла дозволяє знайти найкоротші шляхи між всіма парами вершин у графі, незалежно від того, чи є цикли, чи від'ємні ваги.

- 2) Зручність в роботі з матрицями: Алгоритм легко реалізувати в мові програмування за допомогою матриці відстаней, що спрощує роботу із знаходженням найкоротших шляхів у великих графах.
- 3) Можливість виявлення від'ємних циклів: Алгоритм дозволяє виявити наявність від'ємних циклів у графі, що може бути корисним для аналізу та вирішення проблем у графах, де вони недопустимі.

Обмеження:

- 1) Часова складність: Алгоритм Флойда-Уоршелла має часову складність $O(V^3)$, де V - кількість вершин у графі. Це робить його неефективним для великих графів.
- 2) Пам'ять: Для зберігання матриці відстаней потрібна квадратична кількість пам'яті, що може бути проблемою для великих графів.
- 3) Неефективність для розріджених графів: Якщо граф має багато відсутніх або нульових ваг ребер, алгоритм Флойда-Уоршелла все одно витрачає час на їх обробку, що є надмірним.
- 4) Можлива втрата точності: У великих графах з великою кількістю вершин і великими вагами ребер може виникнути втрата точності через переповнення деяких значень в матриці відстаней.

Алгоритм Флойда-Уоршелла корисний для знаходження найкоротших шляхів у графах, де необхідно знати відстані між всіма парами вершин. Це може використовуватися в транспортних мережах, маршрутизації даних, в графах міської інфраструктури та багатьох інших застосуваннях. Однак важливо враховувати, що алгоритм Флойда-Уоршелла має квадратичну складність і може бути неефективним для великих графів.

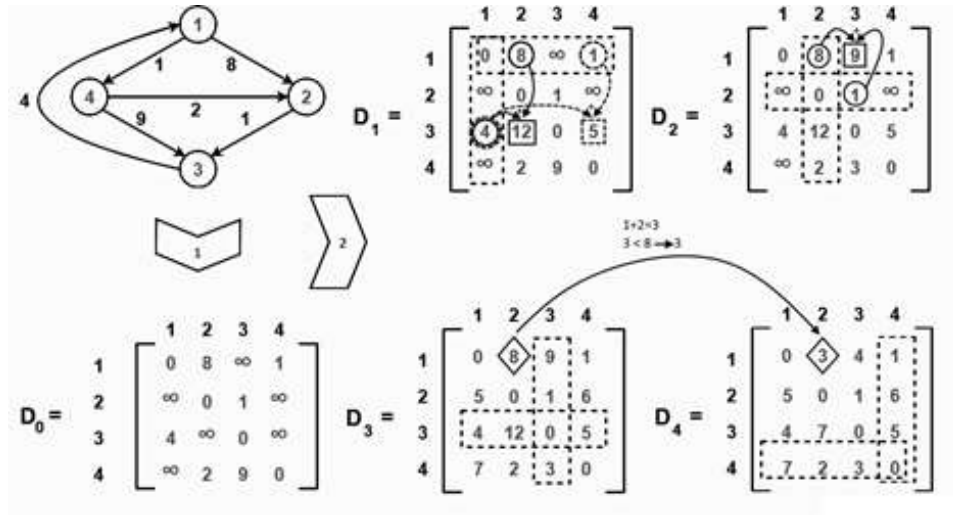


Рисунок 1.6 – Алгоритм Флойда-Уоршелла

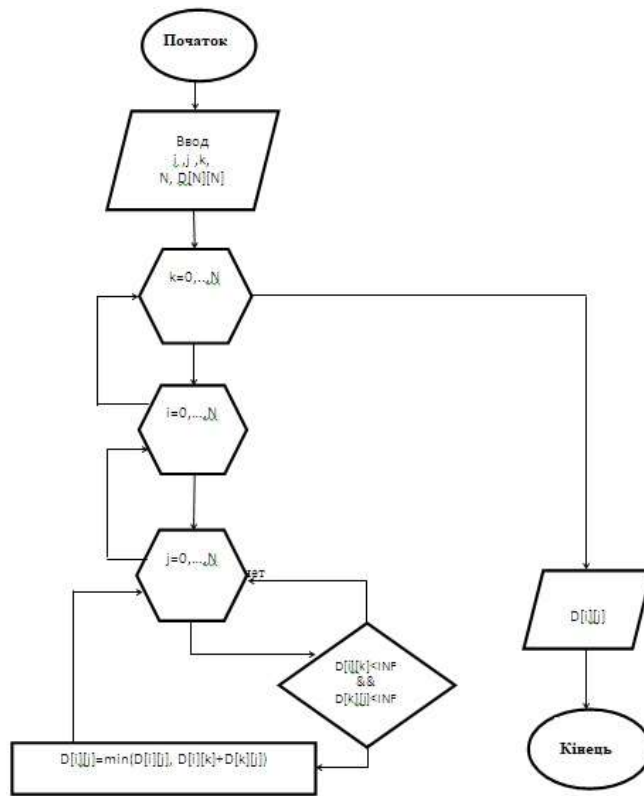


Рисунок 1.7 – Блок – схема алгоритму Флойда-Уоршелла

РОЗДІЛ 2

АНАЛІЗ ТА ВИЗНАЧЕННЯ ІНСТРУМЕНТІВ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕННІ ЗАДАЧІ

2.1 Аналіз існуючих варіантів вирішення поставленого завдання

Коли починається етап розробки програмного середовища навчального призначення з теми «Задача про найкоротші шляхи у орієнтованому графі», перше з чим розробник обов'язково зіштовхнеться це HTML (мова гіпертекстової розмітки) і CSS (каскадні таблиці стилів), але це всього лише інструменти для розмітки сторінки, для побудови чогось більшого нам обов'язково доведеться зробити вибір мови програмування, що найбільше підходить для вирішення нашого завдання.

HTML та CSS є мовами розмітки, які використовуються для створення електронних документів (або сторінок) та для розробки сайтів відповідно. Тим часом, мови веб-програмування складніші і можуть бути поділені на дві категорії: внутрішні та зовнішні мови веб-розробки.

Зазвичай веб-розробка включає кодування на стороні сервера (внутрішня частина), кодування на стороні клієнта (зовнішня частина) та технології баз даних [1].



Рисунок 2.1 - Схема роботи типового веб- додатку

Як приклад візьмемо LinkedIn - найбільшу у світі професійну мережу. Щоб дозволити кандидатам з різних регіонів спілкуватися з іншими, шукати роботу або навіть набувати нових навичок, LinkedIn був запрограмований трьома різними мовами веб-програмування: JavaScript, Java та Scala, а потім використав Voldemort як розподілену базу даних для зберігання величезної кількості профілів.

Відповідно, мови для веб-розробки можна визначити як "складні логічні інструкції та процеси", які допомагають створювати сайти, що відповідають певним вимогам.

Отже, розглянемо найпопулярніші мови веб-розробки та визначимо, яка з них найкраще підійде для вирішення нашого завдання [2].

JavaScript

Щоразу, коли мова заходить про веб-розробку, швидше за все, у 9 із 10 разів буде використано JavaScript. Згідно з щорічними звітами різних популярних платформ, таких як Stack Overflow та Octoverse, JavaScript є однією з найбільш кращих та провідних мов програмування у світі технологій.

Одна з основних причин цього полягає в тому, що конкретна мова може використовуватися як для веб - розробки переднього плану, так і для веб-розробки внутрішнього інтерфейсу.

Дивлячись на деякі попередні тенденції та статистику, можна сказати, що популярність Node.js якимось чином збільшила використання JavaScript як внутрішньої мови для веб-розробки. Тим часом, мова надає вам чудові функції для серверної розробки, таких як полегшена мова сценаріїв, динамічна типізація, інтерпретація, підтримка об'єктно-орієнтованого програмування, перевірка на стороні клієнта, величезна підтримка спільноти та багато іншого.

Фреймворки JavaScript для серверної веб-розробки: Next.js, Express, MeteorJS і т.д.

Популярні веб-сайти, що використовують JavaScript: Facebook, Google, eBay і т.д [3].

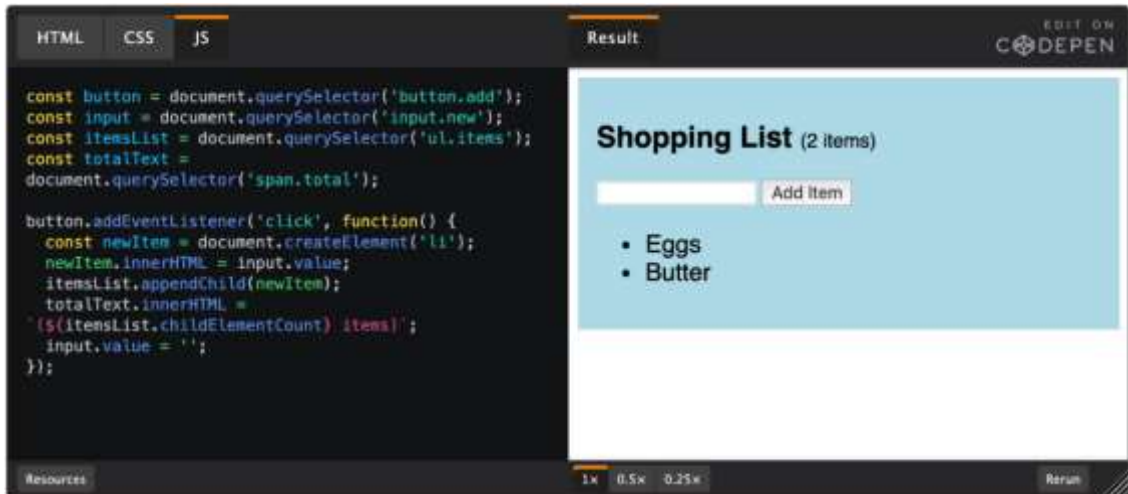


Рисунок 2.2 - Приклад розробки на JavaScript

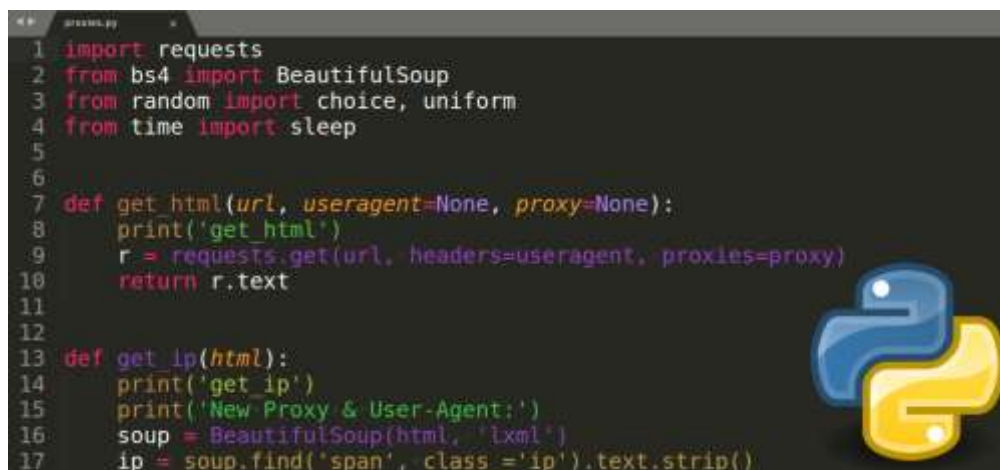
Python

Хоча Python досить відомий серед своєю сумісністю з передовими технологіями, такими як машинне навчання, IoT, наука про дані тощо, ця збагачувальна мова програмування широко використовується і дуже підходить для серверної веб-розробки. також. Навіть один із провідних ІТ-гігантів сучасності Google значною мірою покладається на Python, і це одна з трьох основних мов, які використовуються Google (дві інші — Java та C++). Однією з основних переваг використання Python для веб-розробки є величезний набір стандартних бібліотек, які роблять роботу розробників порівняно простішою та ефективнішою. Додаткові видатні та унікальні функції Python, такі як краща читаємість коду, більш проста інтеграція з іншими мовами, підтримка програмування з графічним інтерфейсом,

переносимість та багато інших факторів роблять його більш переважною мовою серед веб-розробників.

Фреймворки Python для серверної веб-розробки: Django, Flask, Pyramid тощо.

Популярні веб-сайти, що використовують Python: Spotify, Pinterest, Instacart тощо [4].



```

1 import requests
2 from bs4 import BeautifulSoup
3 from random import choice, uniform
4 from time import sleep
5
6
7 def get_html(url, useragent=None, proxy=None):
8     print('get_html')
9     r = requests.get(url, headers=useragent, proxies=proxy)
10    return r.text
11
12
13 def get_ip(html):
14    print('get_ip')
15    print('New Proxy & User-Agent:')
16    soup = BeautifulSoup(html, 'lxml')
17    ip = soup.find('span', class='ip').text.strip()

```

Рисунок 2.3 - Приклад розробки на Python

PHP

PHP (або, можна сказати, препроцесор гіпертексту) є дуже затребуваною мовою програмування у світі веб-розробки. Ця серверна мова сценаріїв з відкритим вихідним кодом створена в 1994 році і спеціально використовується для веб-розробки. Оскільки це мова, що інтерпретується, вона також не потребує компілятора, а також може працювати практично у всіх основних операційних системах, таких як Windows, Linux, macOS, Unix і т. д., а також має багато переваг, таких як : простоту в освоєнні, крос-платформну сумісність, функції ООП, підтримка різних стандартних баз даних, таких як MySQL, SQLite і т. д., величезна підтримка спільноти та багато іншого. Крім цього, PHP дуже безпечний як мова сценаріїв на стороні

сервера, оскільки в PHP доступно безліч хеш-функцій для шифрування даних користувача. Зокрема, якщо ви є новачком, ви можете вибрати PHP для серверної веб-розробки.

PHP-фреймворки для серверної веб-розробки: Laravel, CodeIgniter, Symfony і т.д.

Популярні веб-сайти, що використовують PHP: WordPress, MailChimp, Flickr і т.д [5].

```
<?php
namespace Email;

class Client
{
    public function send(string $emailAddress, Message $message): bool
    {
        if (!$this->validateParameters($emailAddress, $message)) {
            return false;
        }

        return $this->sendToMailServer([
            'to' => $emailAddress,
            'text' => $message->text(),
        ]);
    }
}
```

Рисунок 2.4 - Приклад розробки на PHP

Ruby

Ruby - ще одна мова програмування, яка відмінно підходить для веб-розробки. Подібно до PHP і Python, Ruby також простий в освоєнні і підходить для початківців.

Що робить Ruby особливим для веб-розробки, так це середовище Ruby on Rails, на якому працюють такі сайти, як Github, Shopify, Airbnb, Groupon, GoodReads та Kickstarter.

Якщо ви не знаєте, Rails – це середовище модель-представлення-контролер (MVC), що надає стандартні структури для бази даних, веб-служби та веб-сторінок. Він заохочує та полегшує використання веб-стандартів, таких як JSON або XML, для передачі даних та HTML, CSS та JavaScript для взаємодії з користувачем.

Сама мова є високорівневою та чисто об'єктно-орієнтованою, що означає, що «кожне значення є об'єктом», і в Ruby немає примітивних типів даних. Він також має строгу динамічну типізацію та автоматичне складання сміття – форму для управління пам'яттю.

Синтаксис Ruby порівняно схожий на синтаксис Python та Perl і досить гнучкий. Хоча це робить Ruby легким для читання програмістами, він може легко створювати непередбачувані помилки під час виконання, які важко налагоджувати [6].

```

1  require "time"
2
3  class InvoiceItem
4  attr_reader :id, :item_id, :invoice_id, :created_at
5
6  attr_accessor :quantity, :unit_price, :updated_at
7
8  def initialize(params)
9  @id = params[:id].to_i
10 @item_id = params[:item_id].to_i
11 @invoice_id = params[:invoice_id].to_i
12 @quantity = params[:quantity].to_i
13 @unit_price = BigDecimal(params[:unit_price])
14 @created_at = Time.parse(params[:created_at].to_s)
15 @updated_at = Time.parse(params[:updated_at].to_s)
16 end
17
18 def unit_price_to_dollars
19 @unit_price.to_f
20 end
21 end
22

```

Рисунок 2.5 - Приклад розробки на Ruby

Java

Java – ще одна зразкова мова програмування для серверної веб-розробки. Ця об'єктно-орієнтована мова програмування широко

використовується для розробки веб-додатків масштабу підприємства, а також для розробки додатків для Android, настільних додатків, наукових додатків тощо. Основною перевагою використання Java є те, що він працює за принципом «написати один раз і запустити будь-де», тобто. скомпільований код Java може виконуватись на будь-якій платформі, що підтримує Java, без необхідності повторної компіляції. Говорячи конкретніше, код Java спочатку компілюється в байт-код, який залежить від машини, та був цей байт-код запускається на JVM незалежно від базової архітектури.

Java Frameworks для серверної веб-розробки: Spring, Struts, Grails.

Популярні веб-сайти, що використовують Java: LinkedIn, IRCTC, Yahoo і т.д [7].

```
def get_symbols(file_name):
    with open(file_name, "r") as in_file:
        records = []
        count = 0
        symbol_set = ""
        for line in in_file:
            symbol_set = symbol_set + line[:-1] + ','
            count = count + 1
            if count % 50 == 0:
                records.append(symbol_set)
                symbol_set = ""

        symbols.append(symbol_set)
    return records
```

Рисунок 2.6 - Приклад розробки на Java

C#

C# - одна з тих небагатьох мов, які останні кілька років постійно входять до п'ятірки найкращих мов програмування за різними стандартними індексами. Однак слід зазначити, що ця мова загального призначення спочатку була розроблена Microsoft в першу чергу для платформи .Net.

Поряд із серверною веб-розробкою в даний час С# широко використовується в багатьох областях, таких як розробка програм для Windows, розробка ігор і т. д. Мова надає вам різні корисні функції, такі як швидка компіляція, функціональна сумісність, масштабованість та оновлюваність компонентно-орієнтованість, структурована мова та багато інших. Крім того, С# пропонує багатий набір бібліотек, які допомагають розробникам прискорити та підвищити ефективність процесу розробки. Отже, якщо ви шукаєте мову програмування для веб-розробника, ви також можете спробувати С#.

Фреймворки С# для серверної веб-розробки: .NET

Популярні веб-сайти, що використовують С#: GoDaddy, Marketwatch, Stack Overflow і т.д [8].

```
public async Task<User> RegisterAsync(string login, string password)
{
    var hash = Hash(password);
    var exist = await _repository.Where<User>(s => s.Login == login && s.Password == hash).AnyAsync();

    if (exist)
        throw new AppError("User already exist");

    var id = Guid.NewGuid();
    var user = new User(id, login, hash);

    await _repository.SaveAsync(user);
    await _emailService.SendAsync(user.Login, "Добро пожаловать", "Бла бла бла");

    return user;
}
```

Рисунок 2.7 - Приклад розробки на С#

2.2 Порівняльна характеристика технологій для розробки проектного програмного продукту

Для аналізу характеристик зазначених у першому пункті технологій, складемо порівняльні таблиці для проведення порівняння за такими характеристиками як: Парадигми, Типізація, Управління пам'яттю, Об'єктно-орієнтованими можливостями та частотою використання.

Парадігми

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Імперативна	+	+	+	+	+	+
Об'єктно-орієнтована	+	+	+	+	+	+
Функціональна	+/-	+/-	+/-	+/-	+	+
Рефлексивна	+/-	+/-	+	+	+	+
Узагальна	+	+	+	+	+	+
Логічна	-	-	-	-	-	-
Декларативна	+/-	-	+/-	+	+	+
Розподілена	+/-	+	-	+	+/-	+/-

Таблиця 2.1 - Порівняльна характеристика за парадигмами

Типізація

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Статична типізація	+	+	-	-	-	-
Динамічна типізація	+	-	+	+	+	+
Явна типізація	+/-	+	-	+/-	+/-	-
Неявна типізація	+	-	+	+	+	+
Неявне приведення типів	-	+	+	+	+	+

Таблиця 2.2 - Порівняльна характеристика за типізацією

Керування пам'яттю

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Створення об'єктів на стеку	+	-	-	-	-	-
Некеровані вказівники	+	-	-	-	-	-

Ручне керування пам'яттю	+	-	-	-	-	-
Збирання сміття	+	+	+	+	+	+

Таблиця 2.3 - Порівняльна характеристика за можливостями керування пам'яттю

Функціональні можливості

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Декларації чистоти функцій	-	-	-	-	-	-
First class функції	+	-	+	+	+	+
Анонімні функції	+	+	+	+	+	+
Лексичні замкнення	+	+	+	+	+	+
Часткове застосування	-	-	+	-	+	+
Карування	+	-	+	+	+	+

Таблиця 2.4 - Порівняльна характеристика за функціональними можливостями

Об'єктно – орієнтовані можливості

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Інтерфейси	+	+	+	+	+	-
Мультиметоди	+/-	-	-	-	-	-
Міксини	-	+	+	+	+	+

Перейменування при спадкуванні	-	-	-	-	-	-
Множинне спадкування	-	-	-	-	+	-

Таблиця 2.5 - Порівняльна характеристика за об'єктно – орієнтованими
МОЖЛИВОСТЯМИ

Частота використання

Проведемо аналіз усіх діючих сайтів та получимо наступний графік :

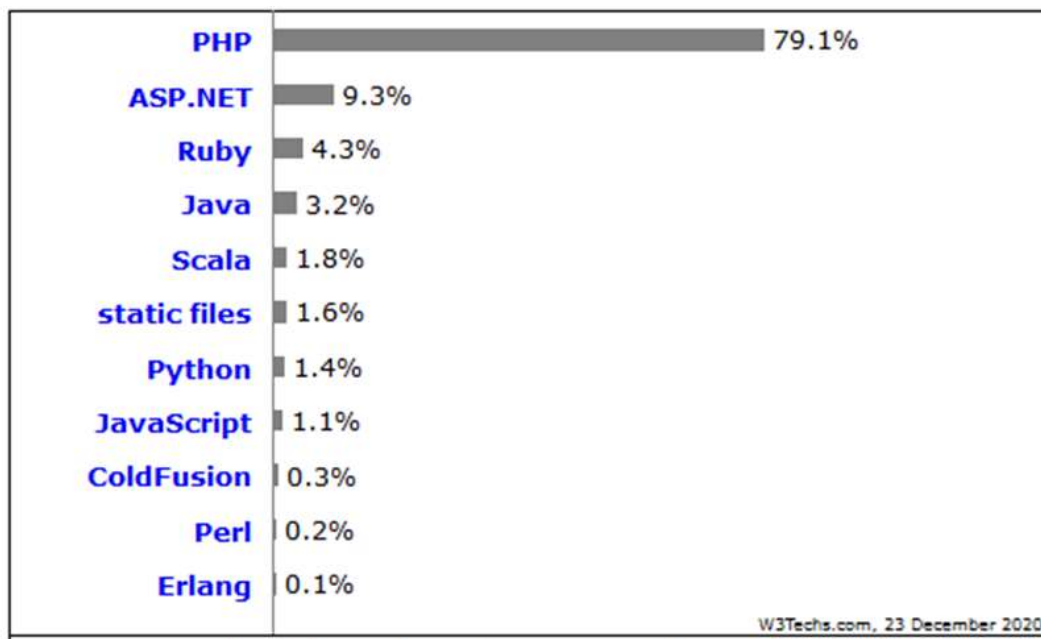


Рисунок 2.9 - Частота використання у існуючих веб-сайтах
З проведеного аналізу можна побачити, що біля 79% сайтів, про бекенд яких відомо, використовують PHP у своїй конструкції [9].

2.3 Аналіз існуючих СУБД для вирішення поставленої задачі

Незалежно від того, яка інформаційна система та з використанням яких мов програмування буде проектуватися, для повного її функціонування будуть використані бази даних.

База даних є систематизований набір даних. Вони підтримують електронне зберігання та маніпулювання даними. Бази даних значно спрощують управління даними.

Наведемо приклад бази даних: онлайн телефонний довідник використовує базу даних для зберігання даних про людей, телефонні номери та інші контактні дані. Постачальник електроенергії використовує базу даних для управління виставленням рахунків, проблемами, пов'язаними з клієнтами, обробкою даних про несправності тощо.

Розглянемо також Facebook. Він повинен зберігати, обробляти та подавати дані, пов'язані з учасниками, їх друзями, діями учасників, повідомленнями, рекламою та багатьом іншим. Можна надати безліч прикладів використання баз даних [10].



Рисунок 2.10 - Схема роботи бази даних

Отже, розглянемо найпопулярніші системи управління базами даних та визначимо, яка з них найкраще підійде для вирішення поставленого завдання.

MySQL

Це одна з найпопулярніших систем реляційних баз даних. Спочатку рішення з відкритим вихідним кодом, MySQL тепер належить корпорації Oracle. Сьогодні MySQL є основою прикладного програмного забезпечення LAMP. Це означає, що він є частиною стека Linux, Apache, MySQL та

Perl/PHP/Python. Маючи під капотом C і C++, MySQL добре працює з системними платформами, як Windows, Linux, MacOS, IRIX та іншими.

Плюси MySQL:

Безкоштовне встановлення. Версія MySQL для спільноти доступна для безкоштовного завантаження. Версія MySQL Community Edition з базовим набором інструментів для індивідуального використання є гарним варіантом для початку.

Простий синтаксис та невелика складність. Структура та стиль MySQL дуже прості. Розробники навіть вважають MySQL базою даних із людською мовою. MySQL часто використовується у тандемі з мовою програмування PHP.

Хмарна сумісність. Орієнтована на бізнес за своєю природою та спочатку розроблена для Інтернету, MySQL підтримується найпопулярнішими хмарними провайдерами. Він доступний на таких провідних платформах, як Amazon, Microsoft та інші [11].

MariaDB

MariaDB, форк MySQL з відкритим вихідним кодом, має комерційну підтримку. Він працює під Стандартною громадською ліцензією GNU і має ті ж команди, API та бібліотеки, що й MySQL.

Плюси MariaDB:

Шифрування. Для MariaDB відкритий вихідний код означає небезпечний. Крім внутрішньої безпеки та перевірки паролів, MariaDB надає такі функції, як автентифікація PAM та LDAP, Kerberos та ролі користувачів.

У поєднанні із зашифрованими табличними просторами, таблицями та журналами він створює надійний рівень захисту даних.

Широкий функціонал. За останні кілька років MariaDB представила багато нових функцій. Наприклад, підтримка ГІС передбачає плавне зберігання координат та запити даних про місцезнаходження. Динамічні стовпці дозволяють одній СУБД забезпечувати обробку даних SQL та NoSQL для різноманітних потреб.

Висока продуктивність. Хоча MariaDB походить від двигуна MySQL, він дуже далеко просунувся з погляду продуктивності. Розширені функції оптимізації покращують управління пулом потоків та обробку даних. Таким чином, при видаленні рядків з таблиці операційна система відразу звертається до вільного простору, усуваючи прогалини в табличному просторі [12].

Oracle

Oracle – це система управління реляційними базами даних, створена та керована корпорацією Oracle. В даний час вона підтримує кілька моделей даних, таких як документ, граф, реляційна модель та модель "ключ-значення" в рамках однієї бази даних. У останніх випусках він переорієнтувався на хмарні обчислення.

Плюси Oracle:

Інновації для щоденного робочого процесу. Починаючи з версії Oracle 12c, коли програмне забезпечення набуло ери гібридної хмари, регулярно з'являлися нові технології хмарних обчислень. З кожним новим випуском Oracle намагається не відставати від темпів інновацій, приділяючи особливу увагу інформаційній безпеці, включаючи активний захист даних, поділ, покращене резервне копіювання та відновлення.

Потужна техпідтримка та документація. Oracle забезпечує гідну підтримку клієнтів та надає вичерпну технічну документацію щодо кількох

ресурсів. Таким чином, ви, ймовірно, знайдете рішення для будь-яких проблем.

Велика ємність. Мультимодельне рішення Oracle дозволяє розміщувати та обробляти величезну кількість даних. Завдяки нещодавно випущеній функції мультиарендності архітектура бази даних тепер спрощує упаковку багатьох баз даних та забезпечує плавне керування ними [13].

PostgreSQL

Ця система управління базами даних поділяє свою популярність із MySQL. Це об'єктно-реляційна СУБД, в якій об'єкти користувача та табличні підходи об'єднуються для створення більш складних структур даних. Крім того, PostgreSQL має багато спільного з MySQL. Він спрямований на зміцнення стандартів відповідності та розширюваності. Отже, він може обробляти будь-яке робоче навантаження як для одномашинних продуктів, так і для складних додатків.

Плюси PostgreSQL:

Відмінна масштабованість. Вертикальна масштабованість є відмінністю PostgreSQL, на відміну від СУБД MySQL. Враховуючи, що майже будь-яке спеціалізоване програмне рішення має тенденцію до зростання, що призводить до розширення бази даних, цей конкретний варіант, безумовно, сприяє зростанню та розвитку бізнесу.

Підтримка типів даних користувача. PostgreSQL спочатку підтримує велику кількість типів даних за промовчанням, таких як JSON, XML, H-Store та інші. PostgreSQL використовує цю перевагу, будучи однією з небагатьох реляційних баз даних із сильною підтримкою функцій NoSQL.

Легко інтегровані сторонні інструменти. Система управління базами даних PostgreSQL має потужну підтримку додаткових інструментів, як безкоштовних, так і комерційних. У їх обсяг входять розширення для покращення багатьох аспектів. Наприклад, ClusterControl забезпечує

вважаючи допомогу в управлінні, моніторингу та масштабуванні баз даних з відкритим вихідним кодом SQL та NoSQL.

Підтримка з відкритим вихідним кодом та спільнотою. Postgres має повністю відкритий вихідний код та підтримується спільнотою, що зміцнює його як повноцінну екосистему [14].

2.4 Порівняльна характеристика можливих СУБД для використання в розробці програмного середовища навчального призначення з теми «Задача про найкоротші шляхи у орієнтованому графі»

Для аналізу зазначених СУБД складемо порівняльні таблиці для проведення порівняння за такими характеристиками як: Переваги, недоліки, частота використання.

Назва	Переваги	Недоліки
MySQL	<p>Чудово документована</p> <p>Пропонує багато функцій, навіть у безкоштовній версії</p> <p>Пакет MySQL включений у стандартні репозиторії найпоширеніших дистрибутивів операційної системи Linux, що дозволяє встановлювати її просто</p> <p>Підтримує набір інтерфейсів користувача</p> <p>Може працювати з іншими базами даних, включаючи DB2 та Oracle.</p>	<p>Потрібно потратити багато часу і зусиль, щоб змусити MySQL виконувати нескладні завдання, хоча інші системи роблять це автоматично, наприклад: створювати інкрементні резервні копії.</p> <p>Відсутня підтримка XML або OLAP.</p> <p>Для безкоштовної версії доступна лише платна підтримка.</p>

Oracle	<p>Найсвіжіші інновації та вражаючий функціонал уже впроваджено в цьому продукті, оскільки Oracle прагне тримати планку навіть на тлі інших розробників СУБД.</p> <p>Oracle є вкрай надійною, фактично це еталон надійності серед таких систем.</p>	<p>Вартість Oracle може бути непомірно високою, особливо для невеликих організацій.</p> <p>Система може вимагати значних ресурсів відразу після установки, тому можливо потрібно модернізувати обладнання для впровадження Oracle.</p>
MariaDB	<p>Продуктивність</p> <p>Індикатори дадуть вам знати, як обробляється запит.</p> <p>Розширювана архітектура та плагіни дозволяють налаштувати інструмент відповідно до ваших потреб.</p> <p>Шифрування доступне в мережі, сервері та рівні програми.</p>	<p>На сьогодні стабільність нижча, ніж у MySQL, тому навіть на нових проєктах можна рекомендувати встановлювати mysql.</p> <p>Двигун досить новий, тому поки що немає жодних гарантій подальших оновлень.</p> <p>Як і в інших безкоштовних базах даних, вам доведеться платити за підтримку.</p>
PostgreSQL	<p>Є масштабованим рішенням та дозволяє обробляти терабайти даних.</p> <p>Підтримує формат json.</p> <p>Існує безліч визначених функцій.</p> <p>Доступний ряд інтерфейсів.</p>	<p>Документація туманна, тому, можливо, відповіді на деякі запитання доведеться шукати в Інтернеті.</p> <p>Конфігурація може збентежити непідготовленого користувача.</p> <p>Швидкість роботи може зменшуватися під час пакетних операцій або виконання запитів читання.</p>

Частота використання

Проведемо аналіз усіх випадків використання СУБД та отримаємо наступний графік :



Рисунок 2.11 - Частота використання СУБД у веб - розробці

Як можна побачити, близько 50% сайтів, про бекенд яких відомо, використовують базу даних MySQL у своїй конструкції [15].

2.5 Вибір інструментів для вирішення поставленого завдання на основі проведеного аналізу

Провівши аналіз та порівняльну характеристику між актуальними у наш час та найбільш підходящими для вирішення нашого завдання інструментами, маємо можливість зробити висновок, які з них використовуватимемо у процесі виконання роботи.

Для написання візуальної частини інтерфейсу програмного середовища навчального призначення з теми «Задача про найкоротші шляхи у орієнтованому графі» буде використано HTML як основну розмітку для сторінок, які відображатимуться перед користувачем та CSS для стилізації цих сторінок та надання їм необхідного нам вигляду. В якості мови для виконання бекнд частини буде використаний PHP, проведений аналіз показав, що ця мова найбільш підходить для виконання поставленого завдання, так як вона має досить зрозумілу структуру, є абсолютно вільно

використовуваною, має хороші показники в області продуктивності і відмінно підходить для написання веб - додатків, що недвозначно підтверджується частотою її застосування. В якості СУБД була обрана MySQL, так як вона достатньо проста у використанні, має великий функціонал, безпечна, має хорошу швидкість роботи, а також має інтерфейс для взаємодії з PHP. Як IDE для розробки було вибрано PHPStorm, а для візуалізації роботи з базою даних PHPMyAdmin.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЕКТОВАНОГО ПРОГРАМНОГО ПРОДУКТУ

3.1. Вимоги до програмного продукту

3.1.1. Вимоги до інформаційної безпеки

Для забезпечення інформаційної безпеки особистих даних користувачів, ці дані буде зашифровано за допомогою алгоритмів симетричного шифрування. Симетричне шифрування передбачає використання одного й того самого ключа і для зашифрування, і для розшифрування. До симетричних алгоритмів застосовуються дві основні вимоги: повна втрата всіх статистичних закономірностей в об'єкті шифрування та відсутність лінійності. Прийнято розділяти симетричні системи на блокові та потокові. У блокових системах відбувається розбиття вихідних даних на блоки з подальшим перетворенням ключем. У поточкових системах виробляється певна послідовність (вихідна гамма), яка у подальшому накладається саме повідомлення, і шифрування даних відбувається потоком у міру генерування гами. Схема зв'язку з використанням симетричної криптосистеми представлена малюнку. Де M - відкритий текст, K - секретний ключ, що передається по закритому каналу, $E_K(M)$ - операція зашифрування, а $D_K(M)$ - операція розшифрування.

Операція підстановки виконує першу вимогу до симетричного шифру, позбавляючись будь-яких статистичних даних шляхом перемішування бітів повідомлення за певним заданим законом. Перестановка необхідна до виконання другої вимоги – надання алгоритму нелінійності. Досягається за рахунок заміни певної частини повідомлення заданого обсягу на стандартне значення шляхом звернення до вихідного масиву.

Симетричні системи мають свої переваги, так і недоліки перед асиметричними. До переваг симетричних шифрів відносять високу швидкість шифрування, меншу необхідну довжину ключа за аналогічної стійкості, велику вивченість і простоту реалізації. Недоліками симетричних алгоритмів вважають насамперед складність обміну ключами через велику ймовірність порушення секретності ключа при обміні, який необхідний, і складність управління ключами у великій мережі.

+ Параметры

		id	login	name	lname	password	country
<input type="checkbox"/>		7	VGvk	VGvkZHk=	QnJvd24=	MTIzNDU=	0KPQutGA0LDQuNC90LA=
<input type="checkbox"/>		8	VG9t	VG9tbXk=	eGdmZGdz	ZHNnc2Rn	0KPQutGA0LDQuNC90LA=
<input type="checkbox"/>		9	QWxleA==	QWxleGFuZHI=	QmFyc3Vr	MTIzNDU=	0KDQvtGB0YHQuNGP

Рис. 3.1. Зберігання зашифрованих даних у базі даних

3.1.2. Вимоги до складу та параметрів технічних засобів

Одним з найважливіших факторів, який необхідно врахувати під час розробки програми, є відповідність потреб у ресурсах наявному технічному забезпеченні. Програма має коректно працювати на сумісному з ним обладнанні.

Реалізація проекту проводиться в системі розробки сценаріїв (PHP), включає інтерпретатор мови, набір функцій для доступу до баз даних та різних служб Інтернет. Використання цього продукту пред'являє такі вимоги до обладнання та програмного забезпечення:

- процесор Intel Pentium III 866 МГц та вище (рекомендується Intel Pentium IV / Celeron 1800 МГц);
- оперативна пам'ять 1024 Мбайт та вище;
- дисковий простір щонайменше 100 Мб;
- відеокарта 256Мб;

- мінімальна роздільна здатність екрана 1024x768;
- 32-розрядна операційна система Windows (2000/XP/Vista/Seven/10);
- підтримка технології Flash;
- Інтернет-браузер: Opera, Mozilla Firefox, Internet Explorer, Google Chrome;
- маніпулятор "миша";
- клавіатура.

Для коректного запуску програмного продукту на сервері, визначено наступні вимоги до серверного обладнання та програмного забезпечення:

- Операційна система – Linux (CentOs, Debian тощо).
- PHP 7, Mariadb, Nginx, PHP-FPM
- Оперативна пам'ять від 2 Гб.
- Наявність SSD
- Процесор щонайменше 2-х ядер.
- Vesta, ISPmanager

3.1.3. Вимоги до інформаційної та програмної сумісності

Програмне середовище навчального призначення з теми «Задача про найкоротші шляхи у орієнтованому графі» може використовуватись на комп'ютерах із встановленню операційною системою Windows XP, Vista, 7, 10.

Програма має повну сумісність із офісними продуктами компанії Microsoft, та не викликає конфліктів з іншими програмами.

3.2 Структура програмного продукту

Інформаційна модель відображує загальний склад інформаційних об'єктів автоматизованої системи, яка проектується та зв'язки між ними. Структура інформаційної моделі системи наведена на рис. 3.2.

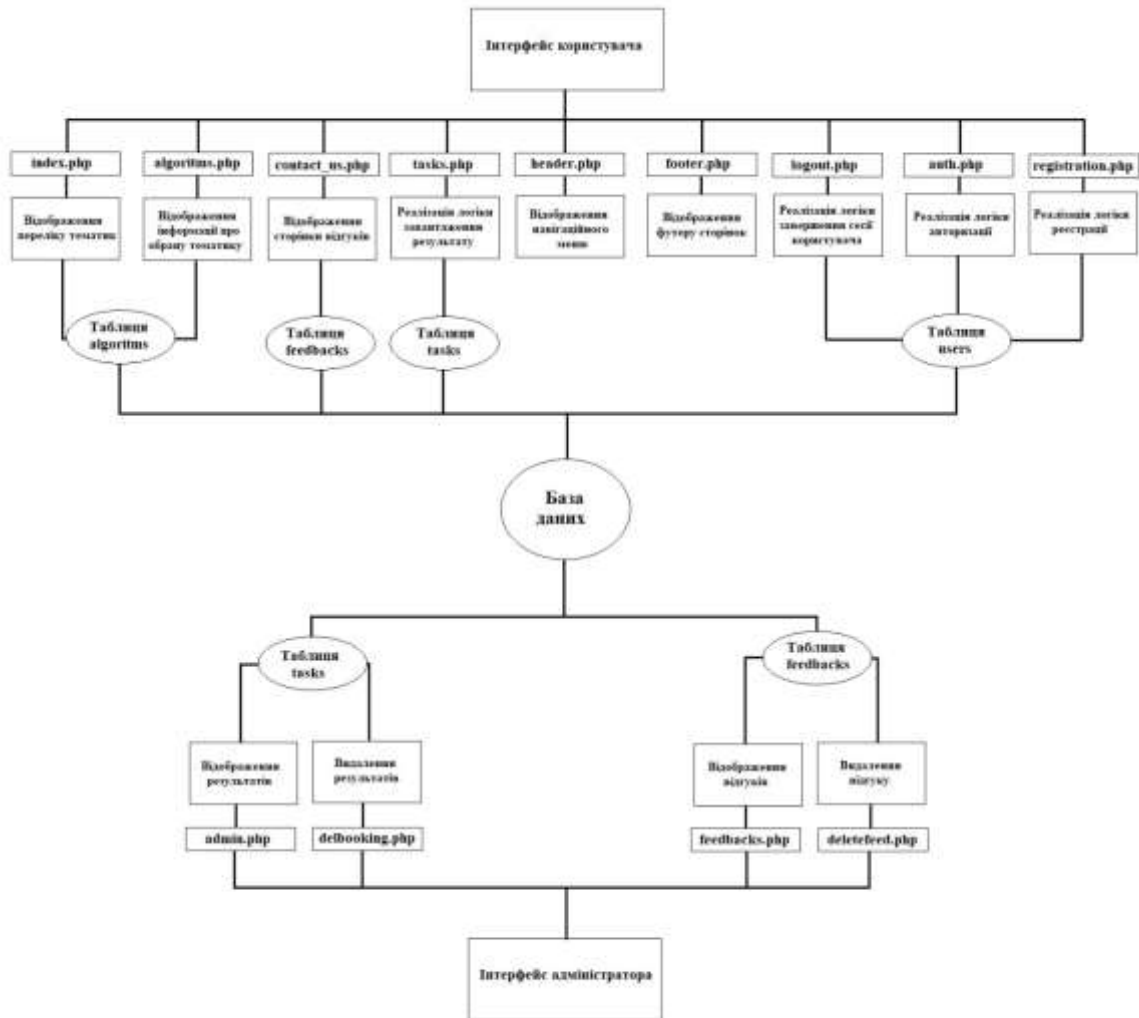


Рис. 3.2 - Структура розроблюваного ПО

Програма складається з набору програмних модулів, кожен з яких виконує одну чи декілька функцій. У таблиці 3.2 наведено призначення програмних модулів.

№ з/п	Позначення	Призначення
1	Admin	Модуль відображення результатів виконаних завдань.
2	Tasks	Модуль для реалізації логіки відправлення виконаного завдання до бази даних.
3	Feedbacks	Модуль для перегляду існуючих у базі даних відгуків про роботу додатку. Містить блок з інформацією з бази та кнопками редагування та видалення.
4	Deletefeedback	Модуль для видалення обраного відгуку з бази даних.
5	Index	Модуль для відображення інформації про доступні навчальні матеріали за тематикою, інформація про які зберігається у базі даних та витягується за запитом користувача. Також має блоки для відображення інформації щодо роботи з додатком та відеоуроків щодо тематики.
6	Algoritm	Модуль для відображення навчальної інформації щодо обраної теми. Містить наведену інформацію та список завдань для виконання.
7	Contact_us	Модуль для надання можливості користувачу залишити відгук про роботу додатку. Містить форму для заповнення особових даних та безпосередньо самого відгуку. Після підтвердження, введені дані буде занесено до

		бази даних відгуків.
8	Header	Модуль для виведення на верхню частину сторінок навігаційної панелі з меню.
9	Footer	Модуль для виведення у нижній частині сторінок контактної інформації та загальну інформацію про додаток.
10	Auth	Модуль для реалізації логіки авторизації користувача у додатку та відображення форми авторизації на інтрефейс користувача.
11	Registration	Модуль для реалізації логіки реєстрації користувача у додатку та відображення форми реєстрації на інтрефейс користувача.
12	Logout	Модуль для реалізації логіки завершення сесії користувача.

Таблиця 3.2 - Призначення програмних модулів

3.3 Проектування архітектури розроблюваної системи

Проектований веб-додаток використовує 3 основних компоненти: веб-браузер, веб-сервер і базу даних, тому буде використовуватися архітектура клієнт-сервер. На малюнку 3.3 зображено діаграму компонентів, що відображає залежність між модулями програмного забезпечення.



Рисунок 3.3 – Діаграма компонентів клієнт-сервер

Архітектура клієнт-сервер є розподіленою структурою програми, яка поділяє завдання або робочі навантаження між постачальниками ресурсу або сервісу (серверами) та ініціаторами запитів (клієнтами). Часто клієнти та сервери взаємодіють через комп'ютерну мережу на окремому устаткуванні, але й клієнт та сервер може перебувати в тій самій системі. Прикладами комп'ютерних програм, які використовують модель клієнт-сервер, є електронна пошта, мережевий друк та мережа Інтернет.

Для проєктованого веб - додатку на першому рівні архітектури знаходиться браузер, з якого виробляються всі операції користувача. На другому – логіка системи, тобто як веб-сервер обробляє інформацію. Третій рівень – це СУБД, що зберігає дані для сервера.

3.4 Проєктування структури бази даних

Система, що розробляється, повинна бути інтегрована з веборієнтованим продуктом, а це означає, що потрібно розробити рекомендаційну структуру бази даних, в якій будуть міститися дані, необхідні для функціонування проєктованого додатку. База даних є необхідною складовою реалізації функціоналу, а саме: збереження галереї зображень, збереження відгуків, збереження інформації про навчальні матеріали тощо. Інтеграція бази даних потрібна для оптимізації роботи веб-сайту з критично важливими даними, що зберігаються під час використання додатку.

Першим кроком є створення таблиці ідентифікаторів, яка наведена у таблиці 3.3.

Об'єкт	Власність	Тип	Розмірність	Ідентифікатор
Таблиця виконаних завдань	Айді	Varchar	100	Id
	Назва алгоритму	Varchar	100	Algo
	Номер завдання	Varchar	100	Number

	Айді користувача	Varchar	100	Studentid
	Відповідь 1	Varchar	100	Firstanswer
	Відповідь 2	Varchar	100	Secondanswer
	Відповідь 3	Varchar	100	Therdanswer
Таблиця відгуків	Айді	Varchar	100	msgID
	Им'я відправника	Varchar	100	senderfname
	Прізвище відправника	Varchar	100	senderlname
	Email відправника	Varchar	100	sendereMail
	Відгук	Varchar	500	Senderfeedback
Таблиця користувачів	Айді	Varchar	100	Id
	Логін	Varchar	100	Login
	Пароль	Varchar	100	Password
	Им'я	Varchar	100	Name
	Прізвище	Varchar	100	Lname
	Група	Varchar	100	Group
Таблиця алгоритмів	Айді	Varchar	100	serviceId
	Зображення	Varchar	100	serviceimg
	Назва	Varchar	100	serviceTitle
	Посилання	Varchar	100	servicelink

Таблиця 3.3 - Таблиця ідентифікаторів

3.5 Програмна реалізація головної сторінки

На головній сторінці були розміщені блоки з алгоритмами вирішення завдання про найкоротші шляхи у орієнтованому графі, які на даний момент актуальні у додатку. Дані щодо алгоритмів знаходяться у базі даних та підгружаються на сторінку за запитом користувача. Також на сторінці представлено підказки по використанню додатку та відеоуроки за тематикою завдання.

```

<?php
$link = mysqli_connect("localhost", "root", "", "graf");
$sql = "SELECT * FROM algoritms";
?>
<header></header>
<div id="home-section-1" class="movie-show-container">
  <h1>Алгоритми пошуку</h1>
  <h3>Оберіть алгоритм, інформацію щодо якого ви хочете отримати</h3>
  <div class="movies-container">
    <?php
      if($result = mysqli_query($link, $sql)){
        $lines = mysqli_num_rows($result);
        if(mysqli_num_rows($result) > 0){
          for ($i = 0; $i <= $lines - 1; $i++){
            $row = mysqli_fetch_array($result);
            echo '<div class="movie-box">';
            echo '';
            echo '<div class="movie-info ">';
            echo '<h3>'. $row['algoTitle'] .</h3>';
            echo '<a href="'. $row['algotlink'] .'"><i
class="fas fa-ticket-alt"></i> Перейти </a>';
            echo '</div>';
            echo '</div>';
          }
          mysqli_free_result($result);
        } else{
          echo '<h4 class="no-annot">На даний момент немає
актуальних алгоритмів</h4>';
        }
      } else{
        echo "ERROR: Could not able to execute $sql. " .
mysqli_error($link);
      }

      // Close connection
      mysqli_close($link);
    ?>

  </div>
</div>

```

У даній ділянці коду реалізується підключення до бази даних за допомогою SQL запити та побудова списку алгоритмів, навчальні матеріали щодо яких є у базі даних додатку. При натисканні на обраний алгоритм користувача буде перенесено на сторінку навчальних матеріалів за обраною тематикою.

```

<div id="home-section-2" class="services-section">
  <h1>Як користуватися додатком ?</h1>
  <h3>3 простих кроки !</h3>

  <div class="services-container">
    <div class="service-item">
      <div class="service-item-icon">

```

```

        <i class="fas fa-4x fa-video"></i>
      </div>
      <h2>1. Авторизуйтеся у системі</h2>
      <p>Авторизуйтеся у додатку за допомогою даних, що було вказано
під час реєстрації</p>
    </div>
    <div class="service-item">
      <div class="service-item-icon">
        <i class="fas fa-4x fa-credit-card"></i>
      </div>
      <h2>2. Оберіть алгоритм</h2>
      <p>Оберіть алгоритм, інформацію щодо якого ви хочете отримати
!</p>
    </div>
    <div class="service-item">
      <div class="service-item-icon">
        <i class="fas fa-4x fa-theater-masks"></i>
      </div>
      <h2>3. Вивчайте отриману інформацію</h2>
      <p>Інформація подається вам у максимально зручному форматі</p>
    </div>
    <div class="service-item"></div>
  </div>
</div>

```

У цій ділянці коду реалізується меню з підказками щодо роботи з додатком. Підказки представлені у вигляді динамічних іконок, що підсвічуються при наведенні.

```

<div id="home-section-3" class="trailers-section">
  <h1 class="section-title">Хочете отримати додаткові знання з теми ?</h1>
  <h3>Ознайомтеся з відеоуроками щодо заданої тематики !</h3>
  <div class="trailers-grid">
    <div class="trailers-grid-item">
      
      <div class="trailer-item-info" data-
video="7orC5euJ3CM?si=oE1LBmFPcsK-f01M">
        <h3>Алгоритм Дейкстри</h3>
        <i class="far fa-3x fa-play-circle"></i>
      </div>
    </div>
    <div class="trailers-grid-item">
      
      <div class="trailer-item-info" data-
video="ckzErtS0QZI?si=SmKlM620M3vF9oAh">
        <h3>Алгоритм Беллмана - Форда</h3>
        <i class="far fa-3x fa-play-circle"></i>
      </div>
    </div>
    <div class="trailers-grid-item">
      
      <div class="trailer-item-info" data-video="1R-
V4ketoR0?si=TEUEIbd-s8_V-DBF">
        <h3>Пошук в глибину на графі</h3>
        <i class="far fa-3x fa-play-circle"></i>
      </div>
    </div>
  </div>

```

```

    <div class="trailers-grid-item">
      
      <div class="trailer-item-info" data-
video="GQkRvlziRf0?si=Af4LSucIIEuHLlyu">
        <h3>Пошук в ширину на графі</h3>
        <i class="far fa-3x fa-play-circle"></i>
      </div>
    </div>
    <div class="trailers-grid-item">
      
      <div class="trailer-item-info" data-
video="3TEybSp0H6k?si=WwFl5Th3_exNDn_u">
        <h3>Алгоритм Дейкстри</h3>
        <i class="far fa-3x fa-play-circle"></i>
      </div>
    </div>
    <div class="trailers-grid-item">
      
      <div class="trailer-item-info" data-
video="4ddTXt_Znz0?si=y1XC0k_kq6zj1Ax">
        <h3>Пошук найкоротшого шляху</h3>
        <i class="far fa-3x fa-play-circle"></i>
      </div>
    </div>
  </div>

```

У даній ділянці коду реалізований блок з тематичними відеоуроками основних алгоритмів, які вирішують завдання про найкоротші шляхи у орієнтованому графі для отримання додаткової інформації. При натисканні на відео розпочнеться відтворення. Інтерфейс сторінки наведено на рисунку 3.4.

Орієнтований граф

Головна Категорія Сторінки Додаток

Алгоритми пошуку

Оберіть алгоритм, інформацію щодо якого ви хочете отримати

Як користуватися додатком ?

З простих кроків

- 1. Авторизуйтесь у системі**
Авторизуйтесь у додатку за допомогою даних, що були введені під час реєстрації
- 2. Оберіть алгоритм**
Оберіть алгоритм, інформацію щодо якого ви хочете отримати
- 3. Вивчайте отриману інформацію**
Інформація надається вам у вигляді відео, друкуємої форми

Хочете отримати додаткові знання з теми ?

Станьте членом з відеореєстрації лекцій заданої тематики

Розглянемо приклад

Соціальні Мережі

- Facebook
- Twitter
- Instagram

Зв'яжіться з нами

Залишайтеся з нами, щоб зрозуміти нові терміни, графіки, будь-які інші аспекти будівництва графів.
Телефон: +380 980 990 99 99
E-mail: info@myself.com

© 2023 Diploma. Created by Myself

Contact Privacy Policy Advertising Settings

Рисунок 3.4 - Вигляд головної сторінки веб – додатку

3.6 Програмна реалізація сторінки обраного алгоритму

На сторінці обраного алгоритму було розміщено загальну інформацію навчального призначення щодо тематики, яку було обрано студентом на головній сторінці додатку. Інформація про обрану тематику знаходиться у базі даних та витягується звідти за запитом користувача. Також на сторінці було реалізовано функціонал отримання завдань по обраній тематиці. Після виконання завдань та завантаження відповідей у відповідні поля форми, користувач може відправити їх до бази даних, звідки їх буде відображено у інтерфейсі адміністрування додатку.

```
<?php
    session_start();
    $login = $_SESSION['itlogin'];
    $id = $_GET['id'];
    $link = mysqli_connect("localhost", "root", "", "graf");
    $movieQuery = "SELECT * FROM algoritms WHERE movieID = $id";
    $movieImageById = mysqli_query($link, $movieQuery);
    $row = mysqli_fetch_array($movieImageById);
    $name = $row['algoTitle'];
?>

<body style="background-color:#6e5a11; ">
    <div class="booking-panel-section booking-panel-section4"
style="width: 96%; margin-left: 2%; margin-bottom: 50px; ">
    <button class="btn btn-warning" onclick="window.location =
'index.php';" ><span> На головну </span></button>
    <div class="movie-box" style="width: 20%; margin-left: 40%;">
        <?php
            echo '';
        ?>
    </div>
    <div class="title" style="text-align: center;"><?php echo $name;
?></div>
    <div style="margin-left: 5%; padding-right: 5%;">
<h5 class="text">Алгоритм нідерландського вченого Едстера Дейкстри знаходить
всі найкоротші шляхи
    з однієї наперед заданій вершини графа до всіх інших. З його допомогою,
при
    наявності всієї необхідної інформації, можна, наприклад, дізнатися яку
послідовність
    доріг краще використовувати, щоб дістатися з одного міста до кожного з
багатьох
    інших, або в які країни вигідніше експортувати нафту тощо.
    Мінусом даного методу є неможливість обробки графів, в яких є ребра з
негативним
    вагою, тобто, якщо, наприклад, деяка система передбачає збиткові для
Вашої фірми
    маршрути, то для роботи з нею варто скористатися відмінним від алгоритму
Дейкстри
    методом.</h5>
```

```

<h5 class="text">Для програмної реалізації алгоритму використовуємо два
масиви: логічний visited -
    для зберігання інформації про відвідані вершини та числовий distance ,
в який будуть
    заноситися знайдені найкоротші шляхи. Отже, є граф  $G = (V, E)$  . Кожна з
вершин
    входять в множину  $n$  , спочатку відзначена що не відвідана, тобто,
елементам
    масиву visited присвоєно значення false .
    Оскільки найвитідніші шляхи тільки належить знайти, в кожен елемент
вектора distance записується таке число, яке точно більше будь-якого
потенційного
    шляху (зазвичай це число називають нескінченністю, але в програмі
використовують,
    наприклад максимальне значення конкретного типу даних). В якості
вихідного пункту
    вибирається вершина  $s$  і їй приписується нульовий шлях:  $distance[s] = 0$ 
, тобто,
    немає ребра з  $s$  в  $s$  (метод не передбачає петель).
</h5>

```

У цій частині коду виконується підключення до бази даних, з бази даних витягується інформація про поточну тематику, що було обрано студентом за її айді, яке передається методом GET разом із завантаженням сторінки. Далі за допомогою цих даних формується сторінка з інформацією щодо обраної тематики.

```

<div class="tasks" id="tasks">
    <form action="tasks.php" method="post">
        <input type="hidden" name="algo" value="<?php echo $name; ?>">
        <h4 class="text" style="color: red">Завдання 1</h4>
        <h5 class="text" style="margin-bottom: 2%;">Дана мережа автомобільних
доріг, що з'єднують міста Львівської області. Знайти найкоротшу відстань від
Львова до кожного міста області, якщо рухатись можна тільки по дорогах.
        </h5>
        <textarea placeholder="Місце для відповіді" class="form-control"
style="width: 80%; margin-left: 10%;" name="first" required></textarea>
        <h4 class="text" style="color: red; margin-top: 2%;">Завдання 2</h4>
        <h5 class="text" style="margin-bottom: 2%;">Дана карта велосипедних
доріжок Латвії та Білорусі. Знайти мінімальну відстань, яку треба проїхати,
щоб дістатися від Риги до Бобруйська.
        </h5>
        <textarea placeholder="Місце для відповіді" class="form-control"
style="width: 80%; margin-left: 10%;" name="second" required></textarea>
        <h4 class="text" style="color: red; margin-top: 2%;">Завдання 3</h4>
        <h5 class="text">Є план міста з нанесеними на нього місцями
розміщення пожежних частин. Знайти найближчу до кожного дому пожежну станцію.
        </h5>
        <textarea placeholder="Місце для відповіді" class="form-control"
style="width: 80%; margin-left: 10%;" name="third" required></textarea>
        <button class="btn btn-warning" style="width: 40%; background: green;
margin-left: 30%; margin-top: 2%;"><span> Відправити відповіді
</span></button>
    </div>
    <button class="btn btn-warning" type="button" style="width: 40%; margin-

```



```

left: 30%; margin-top: 1%; margin-bottom: 1%;" onclick="tasks();" ><span
id="span"> Показати завдання </span></button>
</form>
</div>
<script>
    var status = 'none';
function tasks(){
    if (status == 'none'){
        status = 'block'
        document.getElementById('tasks').style.display = 'block';
        document.getElementById('span').innerHTML = 'Сховати завдання';
    }else if (status == 'block'){
        document.getElementById('span').innerHTML = 'Показати завдання';
        document.getElementById('tasks').style.display = 'none';
    }
}
</script>

```

Наведена частина коду відповідає за реалізацію форми, яка відображує на інтерфейс користувача завдання за обраною тематикою. Користувач має можливість ввести відповіді у спеціальні поля форми та відправити їх на перевірку.

```

session_start();
$login = $_SESSION['itlogin'];
$link = mysqli_connect("localhost", "root", "", "graf");
echo $_POST["algo"];
$insert_query = "INSERT INTO
                tasks ( algo,
                        studentid,
                        firstanswer,
                        secondanswer,
                        thirdanswer)
                VALUES (
                        '$_POST["algo"]',
                        '$login',
                        '$_POST["first"]',
                        '$_POST["second"]',
                        '$_POST["third"]' );";
mysqli_query($link,$insert_query);
echo "<script>window.location.href='index.php'</script>";

```

Ця частина коду реалізує обробку отриманих від користувача даних та якщо вони валідні – занесення їх до бази даних додатку для подальшого відображення на інтерфейсі адміністрування.

Навчання

Алгоритм Дейкстри

Алгоритм нідерландського вченого Едсгера Дейкстри знаходить всі найкоротші шляхи з однієї наперед заданої вершини графа до всіх інших. З його допомогою, при наявності всієї необхідної інформації, можна, наприклад, дізнатися яку послідовність доріг краще використовувати, щоб дістатися з одного міста до кількох з багатьох інших, або в якій країні влітати на територію нафти тощо. Мислом даного методу є неможливість обробки графа, в яких є ребра з негативним вагою, тобто, якщо, наприклад, деяка система передбачає штрафні для Вашої фірми маршрути, то для роботи з нею варто скористатися відходом від алгоритму Дейкстри методом.

Для програми реалізації алгоритму використовують два масиви: посілений visited - для зберігання інформації про відвідані вершини та числовий distance, в який будуть заноситися знайдені найкоротші шляхи. Отже, є граф $G = (V, E)$. Кожен з вершин входить в множину n , спочатку відзначаємо що не відвідано, тобто, елементом масиву visited призначаємо значення false. Оскільки найкоротший шлях тільки належить аналізу, в кожен елемент вектора distance записується таке число, яке точно більше будь-якого потенційного шляху (зазвичай це число називають нескінченністю, для в програмі використовують, наприклад, максимальне значення конкретного типу даних). В якості вихідного пункту вибирається вершина s і їй призначається нульовий шлях: $distance[s] = 0$, тобто, немає ребра з s в s (метод не передбачає петель). Далі, знаходяться всі сусідні вершини (якщо є ребро з s) (якщо такий будуть t і u) і по черзі досліджується, а саме обчислюється вага маршруту з s черзі в кожен з них:

$$distance[t] = distance[s] + \text{веса_інцидентного } s \text{ і } t \text{ ребра};$$

$$distance[u] = distance[s] + \text{веса_інцидентного } s \text{ і } u \text{ ребра}$$

Але шлях імовірно, що в ту чи іншу вершину z може іти два шляхи, тому вагість шляху в таку вершину в масиві distance доведеться порівняти, тоді найбільше (неоптимальне) значення ігнорується, а найменше ставиться у відповідність вершині. Після обробки суміжних з s вершин вона позначається як відвідана: $visited[s] = true$. Так само стає та вершина, шлях з s в яку мінімальний. Припустимо, шлях з s в u коротший, ніж з s в t , отже, вершина u стане активною і буде описана способом досліджується її сусіді, за винятком вершини s . Далі, u позначається як пройдена: $visited[u] = true$, активне стає вершини t і вся процедура повторюється для неї. Алгоритм Дейкстри триває до тих пір, поки всі доступні з s вершини не будуть досліджені.

Тепер на конкретному графі прослідкуємо роботу алгоритму, знайдемо всі найкоротші шляхи між початковою та всіма іншими вершинами. Розмір (кількість ребер) зображеного нижче графа дорівнює $7 (|E| = 7)$, а порядок (кількість вершин) - $6 (|n| = 6)$. Це звичайний граф, кожному з його ребер поставлено у відповідність деяке числове значення, тому довжину маршруту необхідно визначитися числом ребер, що лежать між парами вершин.

З усіх вершин входить в множину n виберемо одну, ту, від якої необхідно знайти найкоротші шляхи до інших доступних вершин. Нехай це буде вершина 1. Додаток шляху до всіх вершин, крім самої, спочатку дорівнює нескінченності, а до неї - 0, тобто, граф не має петель.

У вершині 1 рівно 3 сусідні (вершини 2, 3, 5), і щоб обчислити довжину шляху до них потрібно додати вагу дуг, що лежать між вершинами: $1+2, 1+3, 1+5$ до значення першої вершини (є нульом):

2	-	1 + 2
3	-	1 + 3
5	-	1 + 5

Як уже зазначалося, отримані значення присвоюються вершинам, лише в тому випадку якщо вони «кращі» (мають менше значення) ніж інші на даний момент. А так як кожна з трьох чисел менше нескінченності, вони стають новими активними, що означають довжину шляху з вершини 1 до вершин 2, 3 і 5.

Далі, активна вершина позначається як відвідана, статус «активної» (вершини колої) переходить до однієї з її сусідок, а саме до вершини 2, оскільки вона найближча до раніше активної вершини.

Рисунок 3.5 – Частина навчального інтерфейсу на прикладі алгоритму Дейкстри



Рисунок 3.6 – Інтерфейс виконання завдань з обраної тематики

3.7 Програмна реалізація сторінки відгуків

На цій сторінці була розміщена основна інформація щодо розробленого додатку та вбудована інтерактивна карта. Нижче була розташована форма, за допомогою якої користувач може залишити свій відгук про використання додатку для покращення його функціоналу.

```
<?php
    $link = mysqli_connect("localhost", "root", "", "graf");
?>
<header></header>
<div class="gmap_canvas"><iframe id="gmap_canvas"
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3262.5815823728194
!2d-%20Charlotte!5e0!3m2!1sru!2sua!4v1654828127938!5m2!1sru!2sua" width="600"
height="450" style="border:0;" allowfullscreen="" loading="lazy"
referrerpolicy="no-referrer-when-downgrade"></iframe>
</div>
<div class="contact-us-container">
    <div class="contact-us-section contact-us-section1">
        <h1>Залиште ваш відгук</h1>
        <p>Сприяйте кращій роботі сервісу. Будь ласка, залиште ваш відгук
про використання.</p>
        <form action="" method="POST">
            <input placeholder="Ваше ім'я" name="fName" required><br>
            <input placeholder="Тема відгуку" name="lName" ><br>
            <input placeholder="Ваш E-mail адрес" name="email"
required><br>
            <textarea placeholder="Ваш відгук !" name="feedback"
rows="10" cols="30" required></textarea><br>
            <button type="submit" name="submit" value="submit">Відправити
```

```

Відгук</button>
    <?php
        if(isset($_POST['submit'])){
            $insert_query = "INSERT INTO
                feedbackTable ( senderfName,
                                senderlName,
                                sendereMail,
                                senderfeedback)
                VALUES (
                    '$_POST["fName"]','
                    '$_POST["lName"]','
                    '$_POST["eMail"]','
                    '$_POST["feedback"]' );
            mysqli_query($link,$insert_query);
        }
    </form>

</div>
<div class="contact-us-section contact-us-section2">
    <h1>Адреса & Інформація</h1>
    <h3>Наші номери</h3>
    <p><a href="tel:01011391148">+38 099 999 99 99</a><br>
        <a href="tel:01011391148">+38 099 999 99 99</a></p>
    <h3>Адрес</h3>
    <p>El Sherouk City , Suez Desert Road , Cairo 11837 - P.O. Box
43</p>
    <h3>E-mail</h3>
    <p><a href="#">diploma@gmail.com</a></p>
</div>
</div>
<footer></footer>
    <script src="scripts/jquery-3.3.1.min.js "></script>
    <script src="scripts/script.js "></script>
</body>
</html>

```

Наведена частина кдлу реалізує підключення до бази даних відгуків, а також відображення та функціонування форми для надсилання відгуку. Для надсилання користувач повинен заповнити поля форми і за натисканням кнопки надіслати дані якщо вони валідні будуть відправлені до бази даних відгуків для подальшої обробки.

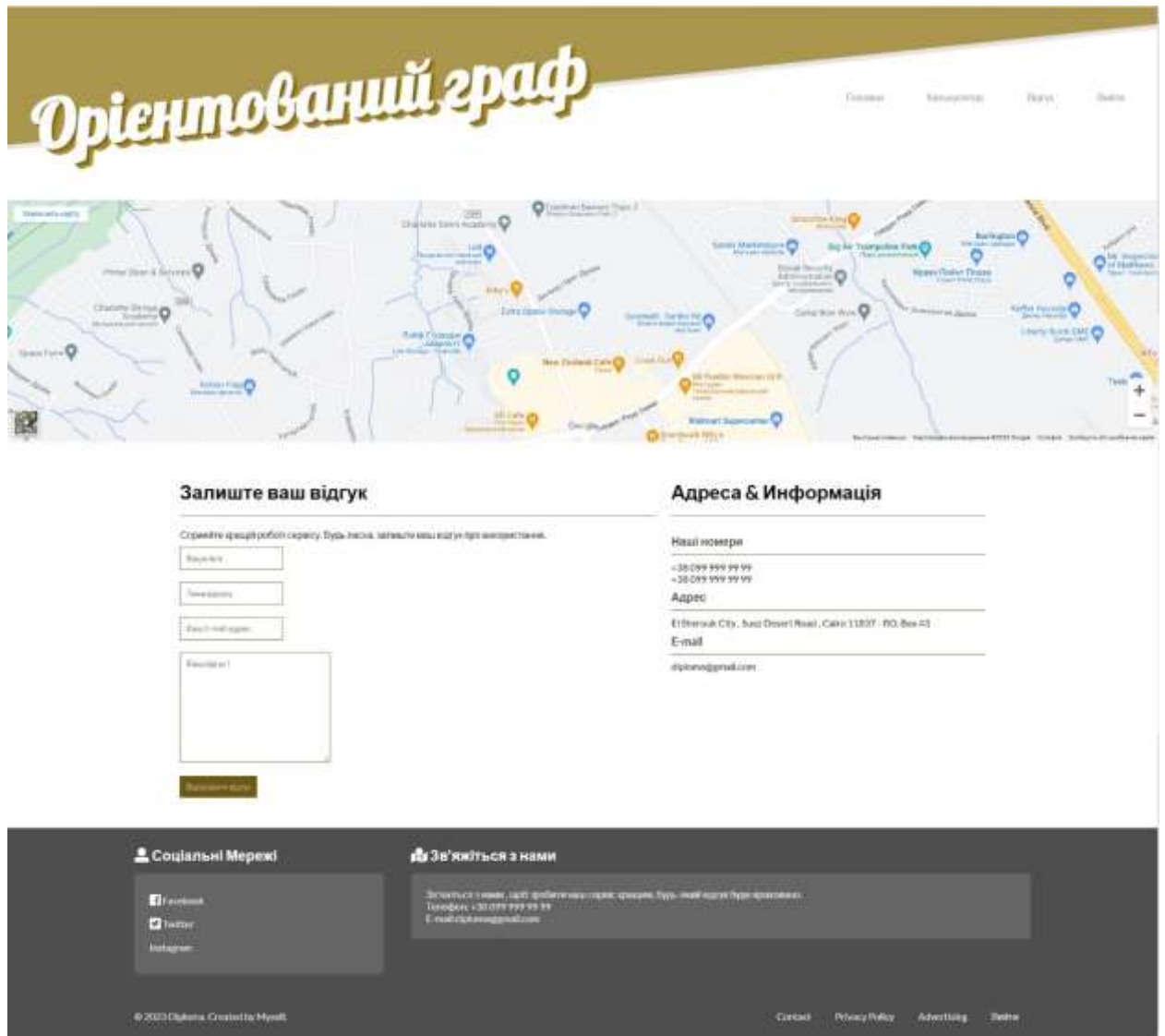


Рисунок 3.7 - Загальний вигляд сторінки відгуків

3.8 Програмна реалізація сторінки реєстрації

На даній сторінці було розміщено форму для реєстрації студента у системі розробленого додатку. Реєстрація обов'язкова для подальшого використання сервісу.

На сторінці реалізується форма реєстрації з полями логіну, паролю, ім'я, прізвища та групи студента. Після введення даних та натискання кнопки

“Зареєструватись”, введені дані буде перевірено на наявність у базі даних. Якщо користувача з таким логіном ще не існує у системі, реєстрацію буде проведено успішно, у іншому випадку – на інтерфейс користувача буде виведено інформацію щодо помилки реєстрації.

Логіка сторінки відповідає за перевірку даних з реєстраційної форми на валідність та якщо вони валідні – відправлення даних до бази даних користувачів. Програмну реалізацію наведено далі:

```
<div mt="150px" class="container">
  <div class="row">
    <div class="col-3"></div>
    <div class="col-6">
      <h1><center>Будьласка, заповніть форму для реєстрації у
дodatку</center></h1><br>
      <form action="" method="POST" >
        <div class="box-body">

          <div class="step_1">
            <div class="form-group">
              <label for="First_name">Логін:</label>
              <input type="text" class="form-control first_name
" name="name" required>
            </div><br>

            <div class="form-group">
              <label for="Second_name">Ім'я:</label>
              <input type="text" class="form-control
second_name" name="sname" required>
            </div><br>

            <div class="form-group">
              <label for="birthday">Прізвище:</label>
              <input class="form-control birthday" required
type="text" name="company" >
            </div><br>

            <div class="form-group">
              <label for="subject">Пароль:</label>
              <input class="form-control subject" required
type="text" name="position" >
            </div><br>

            <div class="form-group">
              <label for="country">Група:</label>
              <select class="form-control country"
name="country" id="country" >
                <?php
                $arr = array(1,2,3, 4);
                foreach ($arr as &$value) {
                  echo "<option>".$value."</option>";
                }?>
              </select>
            </div><br>
          <div class="box-footer">
```

```

        <button type="submit" style=" width: 100%;"
name="submit" value="submit" class="btn btn-danger">Зареєструватися</button>
    </div>
    <div class="auth_button">
        <h4 style="margin-top: 5%;">Вже
зареєстровані?</h4>
        <a href="auth.php">Увійти</a>
    </div>
</div>
</div>
<?php

if(isset($_POST['submit'])){
    $tyy = $_POST['name'];
    $sql = "SELECT id FROM users WHERE login = '$tyy'";
    $result = mysqli_query($link,$sql);
    if(mysqli_num_rows($result)==0){
        $insert_query = "INSERT INTO
        users ( login,
        name,
        lname,
        password,
        country
        )
        VALUES (
            '$_POST["name"]','
            '$_POST["sname"]','
            '$_POST["company"]','
            '$_POST["position"]','
            '$_POST["country"]','
        )";
        mysqli_query($link,$insert_query);
        $_SESSION['itlogin'] = $tyy;
        echo "<script>alert(\"Успішна
реєстрація!\");</script>";
        echo '<script> setTimeout(function () {
window.location.href = "index.php"; }); </script>';
    }else{
        echo "<script>alert(\"Цей логін вже
зайнятий!\");</script>";
    }
}
?>
</form>
</div>
</div>

```

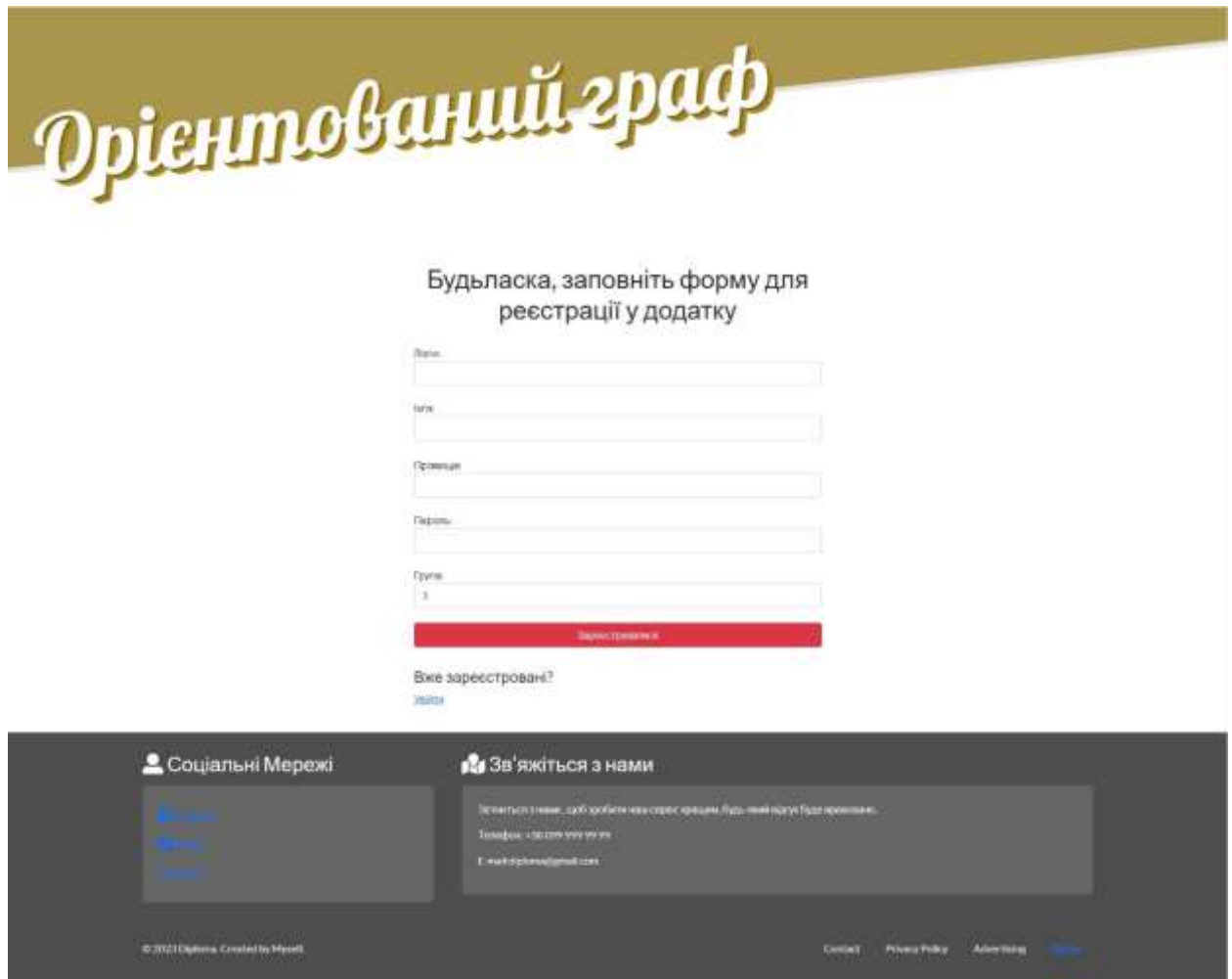


Рисунок 3.8 - Загальний вигляд сторінки реєстрації

3.9 Програмна реалізація сторінки авторизації

На цій сторінці була розміщена форма для авторизації у додатку. Введені у форму дані будуть перевірені на співпадання з базою даних користувачів, та якщо вони співпадають, користувача буде переміщено до головної сторінки додатку.

На сторінці реалізується підключення до бази даних, створення форми з полями імені та паролю та перевірку їх на наявність у базі даних сайту. Код сторінки наведено далі:


```

<div mt="150px" class="container">
  <div class="row" style="margin-bottom: 20%;">
    <div class="col-3"></div>
    <div class="col-6">
      <h1><center>Будь ласка, введіть свій логін та
пароль</center></h1><br>

      <form action="" method="POST" >
        <div class="box-body">

          <div class="step_1">
            <div class="form-group">
              <label for="First_name">Логін:</label>
              <input type="text" class="form-control first_name
" name="name" required>
            </div><br>

            <div class="form-group">
              <label for="subject">Пароль:</label>
              <input class="form-control subject" required
type="text" name="position" >
            </div><br>

            <div class="box-footer">
              <button type="submit" style="width: 100%;"
name="submit" value="submit" class="btn btn-danger">Увійти</button>
              <button type="button" style="width: 100%; margin-
top: 2%; background-color: #1b9b4c;" onclick="window.location.href =
'registration.php';" class="btn btn-danger">Зареєструватися</button>
            </div>
          </div>
        </div>
        <?php

        if(isset($_POST['submit'])){
          $tyy = $_POST['name'];
          $pass = $_POST['position'];
          $sql = "SELECT id FROM users WHERE login = '$tyy'";
          $check = "SELECT id FROM users WHERE login = '$tyy' and
password = '$pass'";
          $result = mysqli_query($link,$sql);
          $res = mysqli_query($link,$check);
          if(mysqli_num_rows($result)==0) {
            echo "Такого користувача не існує, будь ласка,
zareestruytes'!";
          }elseif(mysqli_num_rows($res)==0){
            echo "Введений пароль невірний, спробуйте знову";
          }else{
            $_SESSION['itlogin'] = $tyy;
            header ("location: /index.php");
          }
        }
        ?>
      </form>
    </div>
  </div>
</div>

```

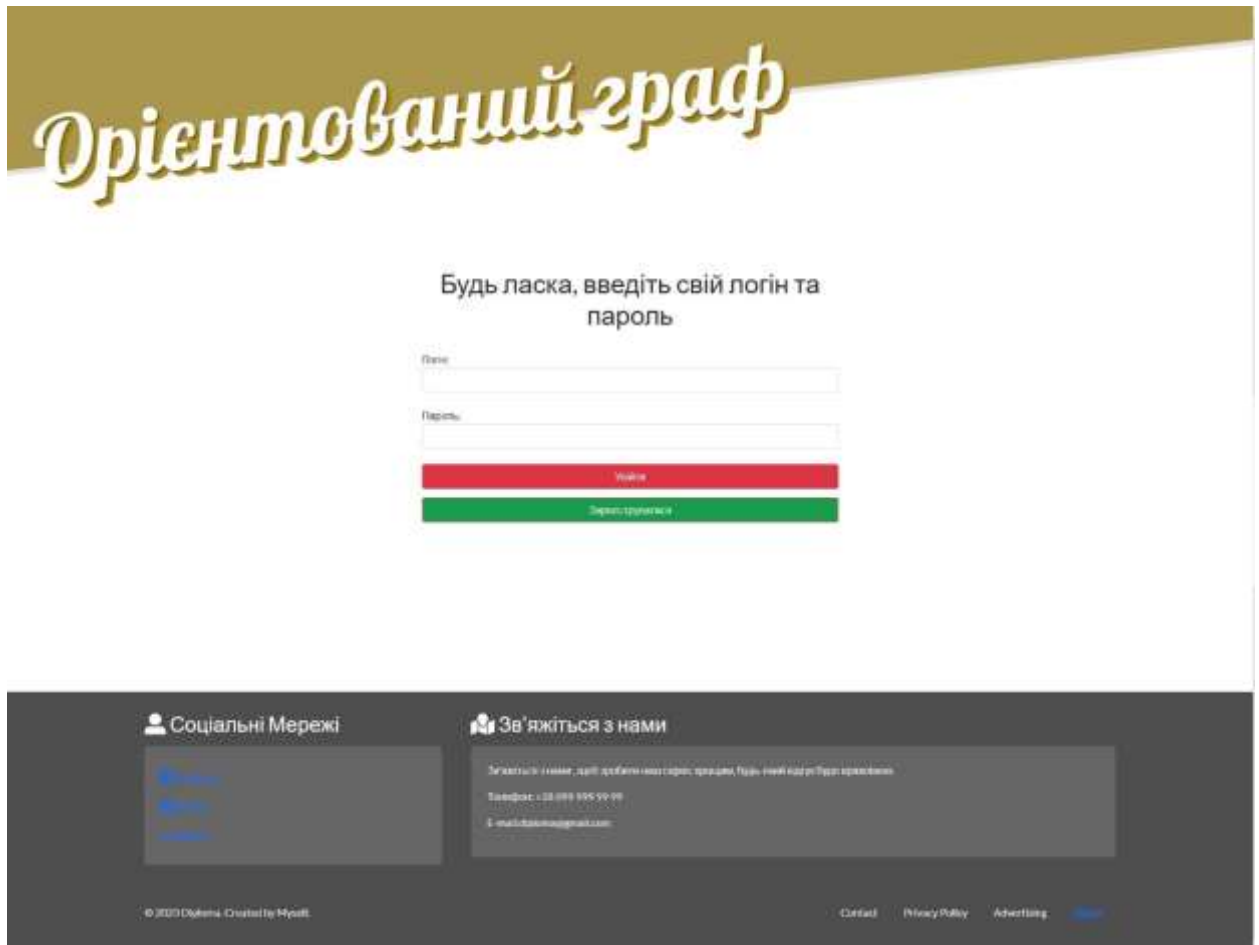


Рисунок 3.9 - Загальний вигляд сторінки авторизації

3.10 Розробка інтерфейсу адміністрування системи

Для адміністрування створеного веб - додатку була створена адміністративна панель, за допомогою якої адміністратор порталу має можливість ознайомитись з завданнями, що було виконано студентами та відгуками щодо використання розробленого додатку.

```
<?php
$link = mysqli_connect("localhost", "root", "", "graf");
$sql = "SELECT * FROM tasks";
$bookingsNo=mysqli_num_rows(mysqli_query($link, $sql));
$messageNo=mysqli_num_rows(mysqli_query($link, "SELECT * FROM
feedbackTable"));
$moviesNo=mysqli_num_rows(mysqli_query($link, "SELECT * FROM algoritms"));
$usersNo=mysqli_num_rows(mysqli_query($link, "SELECT * FROM users"));
```

```

?>
<div class="admin-section-header">
  <div class="admin-logo">
    Diploma Graf
  </div>
  <div class="admin-login-info">
    <i class="far fa-bell admin-notification-button"></i>
    <i class="far fa-comment-alt"></i>
    <a href="#">Welcome, Admin</a>
    
  </div>
</div>
<div class="admin-container">
  <div class="admin-section admin-section1 ">
    <ul>
      <a href="adminfeedbacks.php"> <li> <i class="fas fa-sliders-
h"></i>Відгуки <i class="fas admin-dropdown fa-chevron-right"></i></li></a>
      <li><div class="schedule-item schedule-item-selected"><i
class="fas fa-ticket-alt"></i>Виконані завдання <i ></i></div></li>

    </ul>
  </div>
  <div class="admin-section admin-section2">
    <div class="admin-section-column">
      <div class="admin-section-panel admin-section-stats">
        <div class="admin-section-stats-panel">
          <i class="fas fa-ticket-alt" style="background-color:
#cf4545"></i>
          <h2 style="color: #cf4545"><?php echo $bookingsNo ?></h2>
          <h3>Виконаних завдань</h3>
        </div>
        <div class="admin-section-stats-panel">
          <i class="fas fa-film" style="background-color:
#4547cf"></i>
          <h2 style="color: #4547cf"><?php echo $moviesNo ?></h2>
          <h3>Навчальних матеріалів</h3>
        </div>
        <div class="admin-section-stats-panel">
          <i class="fas fa-ticket-alt" style="background-color:
#bb3c95"></i>
          <h2 style="color: #bb3c95"><?php echo $usersNo ?></h2>
          <h3>Кількість студентів</h3>
        </div>
        <div class="admin-section-stats-panel" style="border: none">
          <i class="fas fa-envelope" style="background-color:
#3cbb6c"></i>
          <h2 style="color: #3cbb6c"><?php echo $messagesNo ?></h2>
          <h3>Відгуків</h3>
        </div>
      </div>
      <div class="admin-section-panel admin-section-panell1">
        <div class="admin-panel-section-header">
          <h2>Виконані завдання</h2>
          <i class="fas fa-ticket-alt" style="background-color:
#cf4545"></i>
        </div>
        <div class="admin-panel-section-content">
          <?php

          if($result = mysqli_query($link, $sql)){
            if(mysqli_num_rows($result) > 0){
              while($row = mysqli_fetch_array($result)){
                echo "<div class=\"admin-panel-section-

```

```

booking-item">\n";
                                echo "                                <div
class=\"admin-panel-section-booking-response\">\n";
                                echo "                                <a
href='deletetask.php?id=".$row['id']."'><i class=\"fas fa-check accept-
booking\" title=\"Reject booking\"></i></a>\n";
                                echo "                                <a
href='deletetask.php?id=".$row['id']."'><i class=\"fas fa-times decline-
booking\" title=\"Reject booking\"></i></a>\n";
                                echo "                                </div>\n";
                                echo "                                <div
class=\"admin-panel-section-booking-info\">\n";
                                echo "
<div>\n";
                                echo "
<h3>Студент ". $row['studentid'] . "</h3>\n";
                                echo "
</div>\n";
                                echo "
<div>\n";
                                echo "
<h3>Виконав завдання: ". $row['algo'] . " </h4>\n";
                                echo "
</div>\n";
                                echo "
<div>\n";
                                echo "
<h3>Відповідь на 1 запитання: ". $row['firstanswer'] . "</h4>\n";
                                echo "
</div>\n";
                                echo "
<div>\n";
                                echo "
<h3>Відповідь на 2 запитання: ". $row['secondanswer'] . "</h4>\n";
                                echo "
</div>\n";
                                echo "
<div>\n";
                                echo "
<h3>Відповідь на 3 запитання: ". $row['thirdanswer'] . "</h4>\n";
                                echo "
</div>\n";
                                echo "                                </div>\n";
                                echo "                                </div>";
                                }
                                mysqli_free_result($result);
                                } else{
                                echo '<h4 class="no-annot">На даний момент немає
виконаних завдань</h4>';
                                }
                                } else{
                                echo "ERROR: Could not able to execute $sql. " .
mysqli_error($link);
                                }
                                ?>
</div>

```

Ця частина коду відповідає за відображення інтерфейсу адміністратора та можливість взаємодіяти з вихідними даними з інтерфейсу користувача.

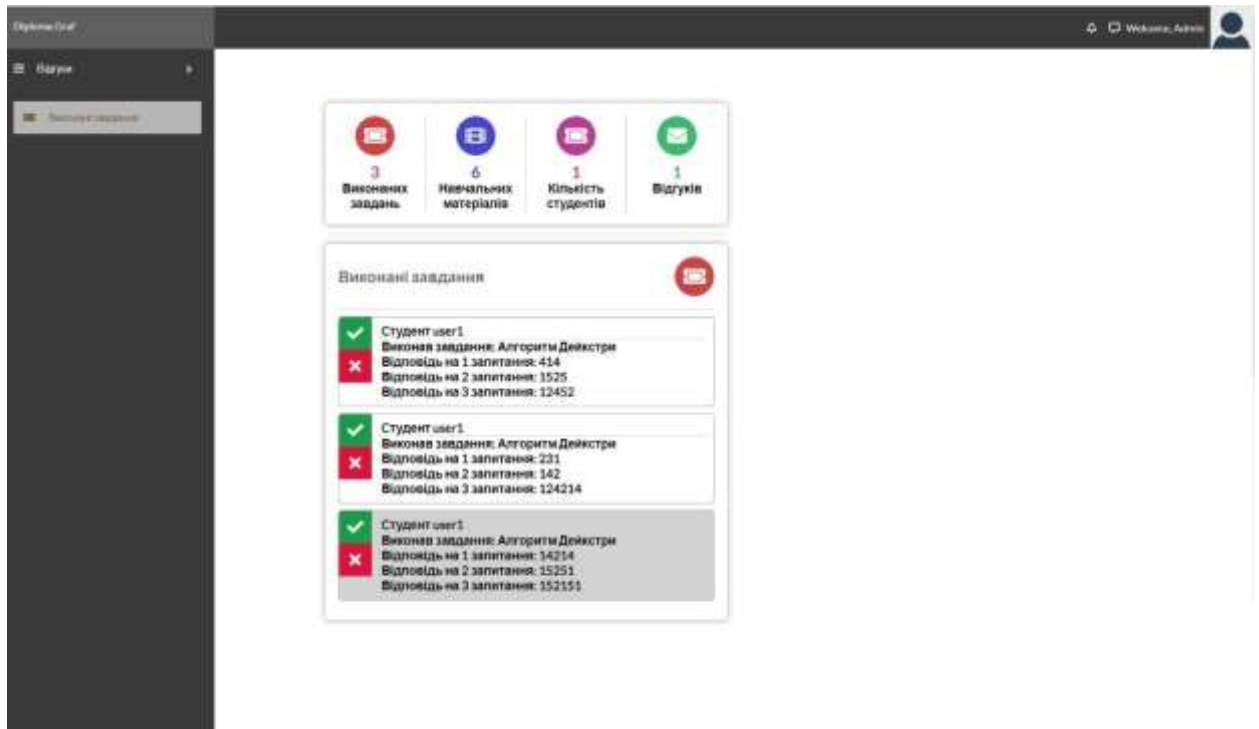


Рисунок 3.10 - Панель адміністратора веб-додатку

ВИСНОВОК

Під час виконання дипломної роботи була проведена робота з вивчення технологій розробки веб - додатків з використанням технологій PHP, MySQL, HTML, CSS. За підсумками виконання роботи було розроблено програмне середовище навчального призначення з теми «Задача про найкоротші шляхи у орієнтованому графі».

Реалізовано функції користувача та адміністратора, що забезпечить можливостями авторизації та реєстрації студента, перегляду основної інформації щодо тематичних матеріалів навчального призначення, отримання завдань та завантаження відповідей щодо завдань з обраної тематики, написання відгуків щодо використання додатку з боку користувача та адміністрування відповідей на завдання та відгуків студентів за допомогою панелі адміністратора.

Створена система дозволяє студенту використовувати функціонал додатку за допомогою інтерфейсу користувача, а адміністратору керувати вихідними даними.

Проведено аналіз предметної галузі, визначено навчальні матеріали з предметної галузі, що буде доречно представити у додатку. Проведено аналіз та вибір інструментів для вирішення поставленого завдання.

Наведено всі необхідні аналітичні процеси для проектування та програмування додатку та бази даних. Після реалізації системи за допомогою обраних інструментів було проведено випробування на працездатність, яке засвідчило, що система виконує усі зазначені нами функції.

Оскільки порушень не виявлено, всі функції відповідають вимогам, тому додаток готовий до впровадження та може бути використана навчальним закладом для автоматизації процесів надання інформації навчального призначення з теми «Задача про найкоротші шляхи у орієнтованому графі»..

СПИСОК ЛІТЕРАТУРИ

1. "PHP та MySQL Web Development (4th Edition)", Luke Welling, Laura Thomson 848 стор., 3 іл.; ISBN 978-5-8459-1574-0, 978-0-672-32916-6.
2. Веб Database Application with PHP and MYSQL», 2nd Edition By David Lane, Hugh E. Вільямс. © O'Reilly, May 2004. ISBN: 0-596-00543-1.
3. CMS List. Огляд cms. Сайт про системи керування сайтом. [Електронний ресурс] Режим доступу: <http://www.cmslist.com>
4. MySQL 4 – Строкові функції [Електронний ресурс] Режим доступу: <http://www.codenet.com/db/mysql/mystring4>
5. Web-розробка [Електронний ресурс] Режим доступу: <http://fcit.tneu.org/web-rozrobka/>
6. Бази даних: розробка та управління: Книга / Хансем Г., Хансем Дж. - М., 2010. - 704 с.
7. Технології розробки та тестування програм [Електронний ресурс]. Режим доступу: <http://moodle.ipr.kpi.ua/moodle/mod/resource/view.php>
8. Види тестування ПЗ [Електронний ресурс]. Режим доступу: <http://qlearning.com.ua/theory/lectures/material/testing-types-functional/>
9. Перенесення файлів із localhost на сервер [Електронний ресурс]. Режим доступу: <http://joomlaportal.com/faq/installation-and-update/51-perenos-sajta-slocalhost-na-server>
10. Архітектура клієнт-сервер [Електронний ресурс] Режим доступу: <http://inter.ptngu.com/>
11. cyberleninka.com/ [Електронний ресурс]. – Режим доступу: <https://cyberleninka.com/article/n/rol-udovletvorennosti-trudom-personala-vreshenii-upravlencheskih-zadach-meditsinskoj-organizatsii/viewer>

12. hr-director.com [Электронный ресурс]. – Режим доступа: <https://www.hr-director.com/article/63014-otsenka-udovletvorennosti-personalauznaem-nastroeniya-rabotnikov>
13. hrliga.com [Электронный ресурс]. – Режим доступа: <https://hrliga.com/index.php?module=profession&op=view&id=1676>
14. lifewire.com [Электронный ресурс]. – Режим доступа: <https://www.lifewire.com/what-is-a-web-application-3486637>
15. habr.com [Электронный ресурс]. – Режим доступа: <https://habr.com/ua/post/450282/>
16. en.yeeply.com [Электронный ресурс]. – Режим доступа: <https://en.yeeply.com/blog/advantages-and-disadvantages-of-web-appdevelopment/>
17. svitla.com [Электронный ресурс]. – Режим доступа: <https://svitla.com/blog/web-application-architecture>
18. slideshare.net [Электронный ресурс]. – Режим доступа: <https://www.slideshare.net/ssusere5f319/web-66422826>
19. wikipedia.org [Электронный ресурс]. – Режим доступа: https://uk.wikipedia.org/wiki/Доменна_система_імен
20. wikipedia.org [Электронный ресурс]. – Режим доступа: <https://uk.wikipedia.org/wiki/HTML>
21. tutorialspoint.com [Электронный ресурс]. – Режим доступа: https://www.tutorialspoint.com/css/what_is_css.htm
22. wikipedia.org [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/PHP_93
23. web-creator.com [Электронный ресурс]. – Режим доступа: https://web-creator.com/articles/about_frameworks
24. habr.com [Электронный ресурс]. – Режим доступа: <https://habr.com/ua/post/464417/>