

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма: «Розробка програмного забезпечення»

Група: 4РП-08

Дипломний проект

здобувача освіти денної форми навчання

РП.08.04.000.ДП

***ВАСИЛЕВСЬКОГО
ДМИТРА КОСТЯНТИНОВИЧА***

м. Одеса
2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма: «Розробка програмного забезпечення»

Група: 4РП-08

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту на тему:

Розробка веб-застосунку для побудови діаграми компетентностей ІТ-фахівця

Проектний матеріал складається з пояснювальної записки на 74 сторінках та графічного (презентаційного) матеріалу на 12 аркушах (слайдах)

Дипломник _____ (Василевський Д.К.)
Керівник _____ (Закроєв Ю.М.)

Консультанти:

з економічного розділу _____ (Канський М.Ю.)
з розділу охорони праці та техніки безпеки _____ (Чорновол Н.І.)
з нормоконтролю _____ (Петрашова В.І.)
старший консультант _____ (Кривченко Ю.В.)

До захисту допущений

Голова циклової комісії _____ (Кривченко Ю.В.)
Завідувач відділення _____ (Краснокутська К.Г.)

Захист «26» 06 2025 р. Протокол ЕК № 2

Оцінка ЕК 5/90

Секретар ЕК _____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та ПІ
Спеціальність 121 «Інженерія програмного забезпечення»
Освітньо-професійна програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ:
Заст. дир. з НВР Беркань І.В.
« 12 » 08 2025 р.

ЗАВДАННЯ

на дипломний проект

Здобувачеві освіти Василевському Дмитру Костянтиновичу
(прізвище, ім'я, по батькові)

1. Тема проекту Розробка веб-застосунку для побудови діаграми компетентностей ІТ-фахівця.

затверджена наказом по коледжу від « 14 » листопада 2024р. № 246

2. Термін здачі закінченого проекту _____

3. Вихідні данні до проекту _____

1. Передбачити застосування прогресивного фреймворку;
2. Передбачити генерацію діаграм напряму через SVG або бібліотеку JavaScript;
3. Передбачити переключення тем (чорна/ світла);
4. Передбачити переключення мови (англійська/ українська);
5. Передбачити збереження діаграми PNG та даних у JSON.

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

1. Аналіз предметної області;
2. Технології та засоби розробки;
3. Проектування веб-застосунку;
4. Розробка клієнтського веб-застосунку;
5. Тестування створеного веб-застосунку;
6. Економічний розрахунок;
7. Аспекти охорони праці та техніки безпеки.

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)
Титул; Метрики компетенцій; Використані технології для розробки веб-застосунку; Файлова архітектура веб-застосунку; Навігація у веб-застосунку; Блок схема алгоритму генерації діаграми; Код генерації діаграми в SVG; Тестування створення діаграми; Тестування зміни мови; Тестування зміни теми; Тестування завантаження діаграми; Робота з хостингом.

6. Консультанти по проекту, із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основний розділ	Закросв Ю.М.		
Економічний розділ	Канський М.Ю.		
Розділ охорони праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Кривченко Ю.В.		

7. Дата видачі завдання 12.05.2025

Керівник

Закросв Ю.М.

(підпис)

Завдання прийняв до виконання

Василевський Д.К.

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Формування вступу	15.05.25	виконано
2	Дослідження предметної області	16.05.25	виконано
3	Огляд аналогів	19.05.25	виконано
4	Вибір технічної літератури	20.05.25	виконано
5	Аналіз технологій розробки	22.05.25	виконано
6	Проектування веб-застосунку	23.05.25	виконано
7	Розробка клієнтського веб-застосунку	26.05.25	виконано
8	Тестування створеного веб-застосунку	27.05.25	виконано
9	Оформлення пояснювальної записки	30.05.25	виконано
10	Підготовка графічних матеріалів	04.06.25	виконано
11	Економічний розрахунок	06.06.25	виконано
12	Опис аспектів охорони праці та техніки безпеки	10.06.25	виконано
13	Підведення висновків	13.06.25	виконано
14	Підготовка доповіді для захисту	16.06.25	виконано

Дипломник

(підпис)

Керівник

(підпис)

ЗМІСТ

Вступ.....	7
1 Основний розділ.....	8
1.1 Відомості про предметну область.....	8
1.1.1 Проблематика обліку професійних навичок працівників.....	8
1.1.2 Метрики введення обліку професійних навичок.....	9
1.1.3 Існуючі рішення на ринку введення обліку професійних навичок.....	11
1.2 Відомості про проєкт.....	15
1.2.1 Функціональність веб-застосунку.....	15
1.2.2 Обґрунтування обраних технологій.....	17
1.2.3 Архітектура веб-застосунку.....	19
1.2.4 Концепція дизайну.....	21
1.3 Розробка сайту.....	23
1.3.1 Налаштування оточення.....	23
1.3.2 Розробка компоненту навігації.....	26
1.3.3 Розробка механізму динамічних тем.....	29
1.3.4 Розробка механізму динамічної зміни мови.....	30
1.3.5 Розробка представлення «Створити діаграму».....	32
1.3.6 Розробка механізму збереження прогресу у JSON.....	38
1.3.7 Розробка представлення «Про діаграми».....	39
1.3.8 Розробка представлення «Про застосунок».....	41
1.4 Тестування веб-застосунку.....	42
1.4.1 Вивантаження веб-застосунку на хостинг.....	42
1.4.1 Обґрунтування обраного виду тестування.....	44
1.4.1 Тестування зручності веб-застосунку.....	45
1.4.2 Тестування основного функціоналу.....	49
2 Економічний розділ.....	51
2.1 Резюме.....	51
2.2 Визначення трудомісткості розробки ПЗ.....	51
2.3 Розрахунок ціни програмного продукту.....	54

					<i>РП 08. 04 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

3 Розділ охорони праці та техніки безпеки.....	56
3.1 Основні положення.....	56
3.2 Аналіз небезпечних та шкідливих виробничих чинників що мають вплив на користувача ПК	56
3.3 Заходи з охорони праці для безпечної роботи програміста	56
3.3.1 Вимоги до виробничих приміщень для експлуатації ПК	56
3.3.2 Нормалізація повітря робочої зони	57
3.3.3 Виробниче освітлення, шум	58
3.3.4 Електробезпека	59
3.3.5 Організація робочого місця	59
3.4 Пожежна безпека	60
3.5 Підведення підсумків	60
Висновки.....	61
Перелік використаних інформаційних джерел	62
Додаток А. Фрагмент коду VueJS компоненту створення діаграм.....	63
Додаток Б. Слайди мультимедійної презентації	68

ВСТУП

Ефективна оцінка професійних компетентностей фахівців у галузі інформаційних технологій є важливим елементом кадрової політики сучасних компаній. Особливу роль у цьому процесі відіграють HR-фахівці та рекрутери, які потребують інструментів для візуалізації та обліку навичок працівників, зокрема в контексті формування збалансованих команд, планування професійного розвитку та прийняття стратегічних рішень щодо персоналу. Традиційні форми обліку компетенцій, як-от таблиці чи текстові описи, часто виявляються недостатньо інформативними в умовах великого обсягу даних. Це ускладнює швидкий аналіз і порівняння кандидатів або поточних співробітників [1].

Одним з перспективних підходів до представлення компетентностей є використання так званих діаграм типу I-, T-, π-, V- та комбінованих (combi-shaped), які дозволяють у наочній формі відобразити як глибину експертизи у вузьких спеціалізаціях, так і широту знань в суміжних напрямках. Проте, незважаючи на їхню ефективність, на ринку програмного забезпечення спостерігається нестача зручних веб-інструментів, орієнтованих саме на потреби HR-відділів у сфері ІТ [2].

У межах даного дипломного проєкту поставлено завдання розробити веб-застосунок, який дозволяє створювати, редагувати та зберігати діаграми компетентностей ІТ-фахівців. Функціональні вимоги до системи включають використання прогресивного JavaScript-фреймворку Vue.js, застосування SVG-технології для векторної генерації діаграм, реалізацію можливості перемикання темного та світлого режимів інтерфейсу, підтримку двох мов (української та англійської), а також експорт діаграм у форматі PNG та збереження даних у форматі JSON.

Результатом роботи має стати інтерактивний веб-застосунок, здатний задовольнити потреби HR-фахівців у зручному інструменті для візуалізації професійних профілів співробітників галузі інформаційних технологій.

					<i>РП 08. 04 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

1 ОСНОВНИЙ РОЗДІЛ

1.1 Відомості про предметну область

1.1.1 Проблематика обліку професійних навичок працівників

У сучасних умовах динамічного розвитку інформаційних технологій ефективне управління людськими ресурсами є ключовим чинником конкурентоспроможності ІТ-компаній. Особливе значення набуває системний підхід до обліку, моніторингу та розвитку професійних навичок працівників. Відсутність централізованого інструменту для фіксації, візуалізації та аналізу компетентностей значно ускладнює процеси кадрового планування, формування команд, а також індивідуального професійного зростання спеціалістів.

Проблематика обліку навичок фахівців, зокрема у сфері ІТ, охоплює кілька ключових аспектів:

1. Невизначеність та фрагментарність даних: У багатьох організаціях інформація про навички працівників зберігається у вигляді неструктурованих записів (резюме, анкети, внутрішні документи), що ускладнює її аналіз і порівняння.
2. Відсутність уніфікованих критеріїв оцінки: Компетентність працівника часто визначається суб'єктивно, без чітких шкал, що призводить до нерівномірного розподілу завдань та необ'єктивних управлінських рішень.
3. Динаміка професійного розвитку: ІТ-сфера характеризується постійним оновленням технологій, тому навички працівників потребують регулярного оновлення. Відсутність механізмів відстеження цих змін призводить до втрати актуальності зібраної інформації.
4. Неefективна візуалізація компетентностей: Класичні методи, як-от текстові таблиці чи звіти, не дають цілісної картини рівня навичок окремого працівника або команди, що ускладнює прийняття стратегічних рішень.

Зважаючи на вищезазначене, постає потреба в розробці спеціалізованого інструменту – зокрема, веб-застосунку, який би дозволяв структурувати, оновлювати та візуалізувати інформацію про професійні навички працівників у

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

зручному вигляді. Такий застосунок має не лише враховувати технічні компетенції, але й підтримувати побудову діаграм, які відображають поточний стан та динаміку розвитку кожної навички.

У контексті управління персоналом це дозволяє:

- оперативно формувати ефективні проєктні команди;
- планувати навчальні заходи з урахуванням виявлених прогалів у знаннях;
- здійснювати об'єктивне оцінювання працівників під час атестацій або внутрішніх конкурсів;
- створювати індивідуальні траєкторії професійного розвитку.

Таким чином, проблематика обліку професійних навичок працівників вказує на актуальність і практичну значущість створення інформаційної системи, здатної оптимізувати цей процес. Розробка веб-застосунку побудови діаграм компетентностей ІТ-фахівця відповідає сучасним потребам галузі та створює основу для покращення управлінських рішень у сфері HR-менеджменту.

1.1.2 Метрики введення обліку професійних навичок

У контексті цифровізації управління персоналом і побудови систем розвитку фахівців, особливо в галузі інформаційних технологій, зростає потреба у візуалізації професійних навичок та їхнього обліку. Одним із сучасних підходів до опису та аналізу навичок є використання метафоричних моделей у вигляді діаграм компетентностей, таких як I-shaped, T-shaped, V-shaped, π -shaped та Comb-shaped. Кожна з цих моделей дозволяє представити уніфіковану метрику для оцінки глибини та ширини знань фахівця [1][2].

На рисунку 1.1 зображено візуальні моделі професійних навичок.

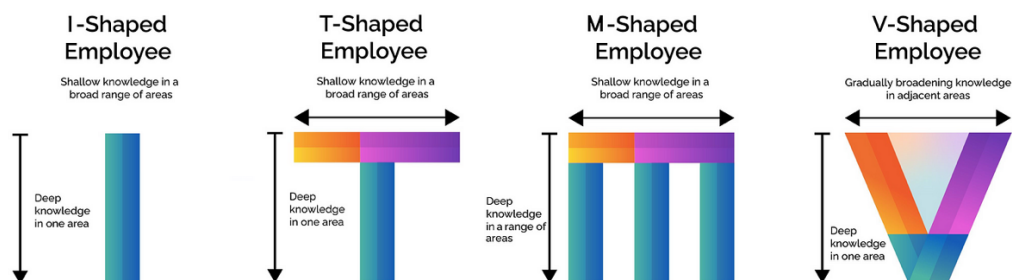


Рисунок 1.1. Моделі професійних навичок

					РП 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

Розглянемо кожну модель:

1. I-shaped діаграма: I-shaped модель відображає спеціаліста з глибокими знаннями у вузькій предметній галузі. Вона ілюструє вертикальну вісь глибокої спеціалізації без розгалужень, що вказує на високий рівень експертності в одному напрямку, але обмежену міждисциплінарність. Такий підхід застосовується для ролей, де потрібна глибока технічна експертиза, наприклад, системні програмісти або спеціалісти з баз даних.
2. T-shaped діаграма: T-shaped модель розширює попередню, поєднуючи глибоку експертизу в одній галузі (вертикальна вісь) із широким розумінням суміжних сфер (горизонтальна вісь). Це дозволяє фахівцеві ефективно співпрацювати з іншими командами та виконувати завдання в міждисциплінарному середовищі. У середовищі ІТ така модель часто застосовується до фахівців з досвідом у програмуванні, які також володіють навичками дизайну, DevOps або менеджменту.
3. V-shaped діаграма: V-shaped модель подібна до T-shaped, однак акцент робиться не лише на глибину та ширину, а й на здатність інтегрувати знання з різних джерел у логічну систему. Форма літери «V» символізує аналітичне мислення, глибоке розуміння проблеми з одночасним вивченням декількох шляхів її розв'язання. Це є цінним у ролях бізнес-аналітиків, системних архітекторів і науковців.
4. π-shaped діаграма: π-shaped модель характеризує фахівця, який має глибокі знання у двох ключових галузях (дві вертикальні осі) та загальну міждисциплінарну обізнаність (горизонтальна вісь). Це ідеальний варіант для ІТ-фахівців, які, наприклад, добре орієнтуються як у програмній інженерії, так і в керуванні проєктами або маркетингу.
5. Comb-shaped діаграма: Comb-shaped модель (або "гребінцева") описує спеціаліста з глибокими знаннями у кількох предметних областях, що представлені у вигляді кількох вертикальних осей, розміщених на горизонтальній базі широких знань. Ця модель відповідає концепції багатопрофільності (multiskilling) і є характерною для senior-розробників,

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

старших аналітиків, інженерів з широким технічним профілем. Вона дозволяє адаптуватися до змінних вимог, працювати в умовах гнучких команд і вирішувати завдання міждисциплінарного характеру.

Кожна з описаних моделей виконує роль візуальної метрики, яка дозволяє:

- Аналізувати структуру навичок фахівця;
- Оцінювати відповідність навичок до вимог конкретної ролі;
- Планувати професійний розвиток у межах індивідуальної траєкторії;
- Визначати потенціал до міждисциплінарної співпраці.

У рамках розробки веб-застосунку побудови діаграми компетентностей ІТ-фахівця зазначені моделі лягають в основу структуризації вхідних даних та візуалізації результатів. Метрики, що базуються на цих моделях, дозволяють систематизувати інформацію про професійні навички, підвищуючи прозорість і ефективність управління знаннями в організаціях.

1.1.3 Існуючі рішення на ринку введення обліку професійних навичок

На сучасному ринку інформаційних технологій спостерігається зростаючий попит на інструменти, що дозволяють систематизувати, візуалізувати та ефективно відстежувати професійні навички спеціалістів. Це зумовлено потребою у прозорому та об'єктивному оцінюванні компетентностей працівників, а також необхідністю планування навчання, розвитку персоналу й оптимізації кадрових процесів у сфері ІТ. У зв'язку з цим з'явилася низка рішень, що частково або повністю реалізують функціонал обліку навичок та побудови профілів компетентностей.

Одним із поширених рішень є LinkedIn Skills Assessment, який пропонує можливість проходження коротких тестів з окремих ІТ-галузей (наприклад, Python, JavaScript, HTML/CSS). Результати відображаються на профілі користувача як підтвердження його навичок. Перевагою такого підходу є інтеграція із соціальною мережею для професіоналів, що створює додаткові можливості для побудови кар'єри. Проте ця система не дозволяє будувати повноцінні діаграми компетентностей і має обмежений набір стандартних навичок без можливості гнучкого налаштування під конкретні вимоги компаній чи команд [3].

					РП 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

На рисунку 1.2 зображено сервіс LinkedIn Skills Assessment.

Skill Assessments

Check your skill level. Answer 15 multiple choice questions, score in the top 30%, and earn a skill badge. [Learn more](#)

The screenshot displays the LinkedIn Skills Assessment interface. At the top, there are filter buttons: 'All', 'Recommended' (highlighted in green), 'Industry Knowledge', and 'Tools & Technology'. A search bar is located on the right. Below the filters, a list of skill assessments is shown, each with a small icon, the skill name, a list of topics, and the number of people who took the assessment:

- Search Engine Optimization (SEO)**: Topics: On-Page, Ranking, Technical, Terminology, Tools. 1.4M people took this.
- Google Ads**: Topics: Bidding, Functionality, Optimizing, Setup, Terminology. 533k people took this.
- Google Analytics**: Topics: Analysis, Configuration, Functionality, Interface, Reports. 840k people took this.
- WordPress**: Topics: Core, Development, Performance and Accessibility, Plugins, Themes. 886k people took this.
- Microsoft Excel**: Topics: Analysis, Data Manipulation, Formulas, Setup. 19M people took this.
- HTML**: Topics: Beyond HTML; HTML Fundamentals; Images, Sound, and Video; Tables and Mathematics; Web for All. 2.5M people took this.

At the bottom of the list, there is a link: [Show more assessments](#).

Рисунок 1.1. Сервіс LinkedIn Skills Assessment

Ще одним прикладом є Skill Matrix – підхід, який активно використовується у внутрішніх HR-системах компаній. У такій матриці навички співробітників перераховуються у вигляді таблиці, де кожна особа оцінюється за заздалегідь визначеними критеріями (рівнями володіння). Такі системи зазвичай реалізуються у вигляді Excel-документів, Google Sheets або внутрішніх веб-сервісів. Вони забезпечують базову візуалізацію рівнів навичок, однак не мають інструментів для побудови складніших графічних моделей (наприклад, кругових або радіальних діаграм), а також не підтримують автоматичну синхронізацію з освітніми досягненнями чи проектною активністю [4].

На рисунку 1.2 зображено табличний підхід Skill Matrix.

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

	Skill 1	Skill 2	Skill 3	Skill 4	Skill 5	Skill 6
Employee Name 1	1		2	2	3	1
Employee Name 2	1	2	2	2	3	1
Employee Name 3	4	4	3	3	3	1
Employee Name 4	3	3	3	3	3	3
Employee Name 5	3	3	3	3	3	3
Employee Name 6	3	2	3	3	1	
Employee Name 7		3	3	4	3	3

	No Skill Present
	1 In Training
	2 Working Under Supervision
	3 Competent
	4 Expert / Trainer

Рисунок 1.2. Табличний підхід Skill Matrix

Pluralsight Skill IQ пропонує більш структурований підхід до оцінювання технічних компетенцій. Користувач проходить адаптивне тестування, після якого отримує числову оцінку володіння певною технологією. Система автоматично формує профіль навичок, який оновлюється після проходження нових курсів. Проте ця система орієнтована переважно на індивідуальне навчання і не дозволяє здійснювати колективний аналіз або порівняння працівників усередині команди. Також Pluralsight є комерційним продуктом з обмеженим доступом для широкого кола користувачів [5].

На рисунку 1.3 зображено сервіс Pluralsight.

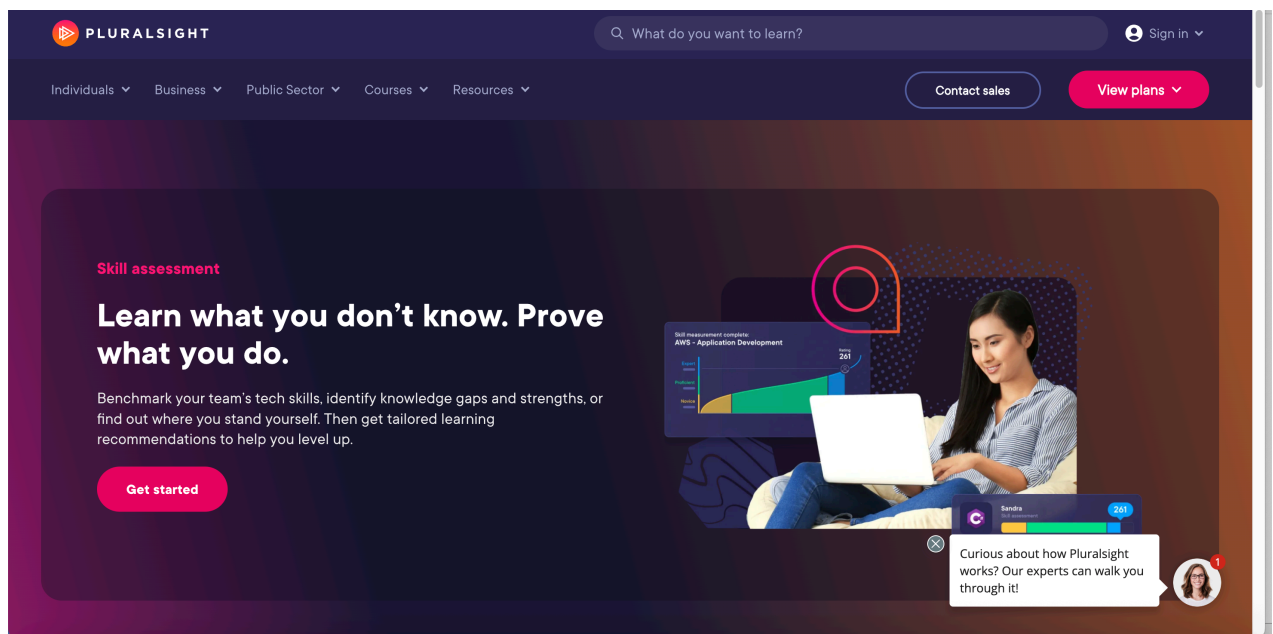


Рисунок 1.3. Сервіс Pluralsight

SkillsBoard – це корпоративна платформа, яка поєднує елементи менеджменту талантів, сертифікації навичок та побудови професійного профілю. Система підтримує створення динамічних діаграм навичок та їхнє оновлення в реальному часі на основі виконаних проєктів і зворотного зв'язку. Незважаючи на гнучкість і сучасний дизайн, рішення є орієнтованим на великий бізнес і складне у впровадженні в невеликі команди або освітні установи через складність налаштування та ліцензійну модель [6].

На рисунку 1.4 зображено сервіс SkillsBoard.

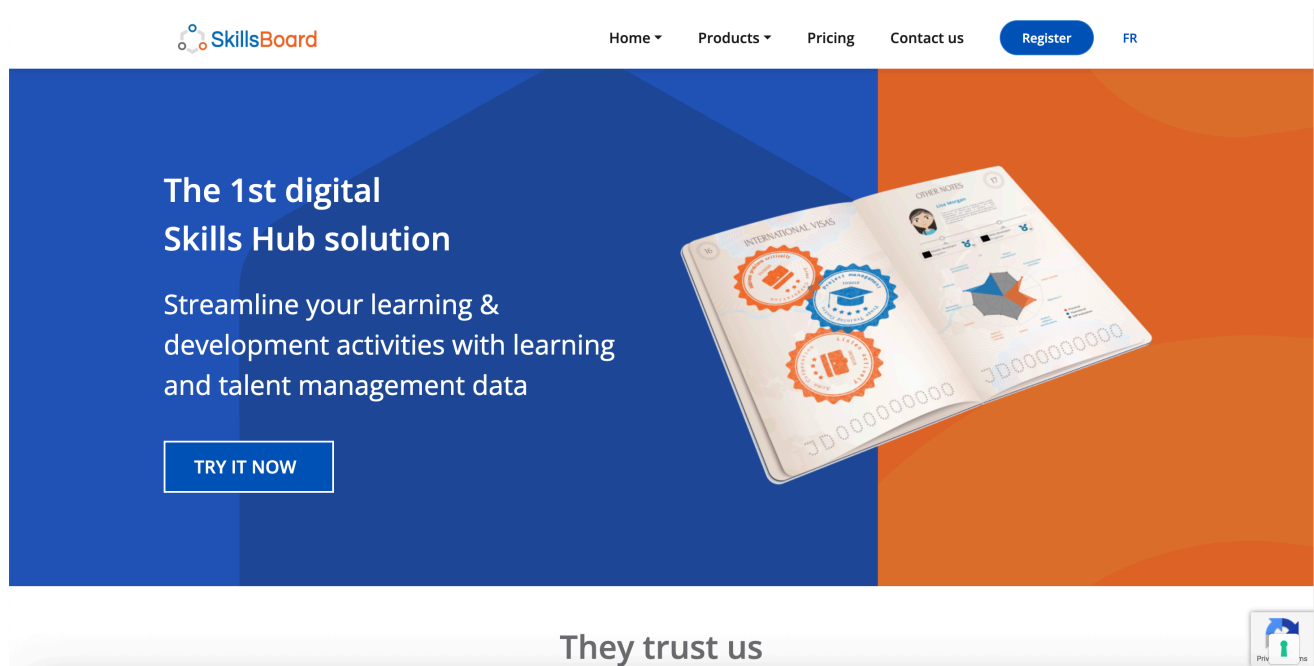


Рисунок 1.4. Сервіс SkillsBoard

CompetencyCore від компанії HRSG є ще одним прикладом професійного рішення, що дозволяє формувати профілі компетентностей, створювати карти кар'єрного розвитку та інтегрувати їх у загальну систему управління персоналом. Основна перевага CompetencyCore – використання стандартизованих моделей компетентностей, які можуть бути адаптовані під конкретні галузі. Водночас інтерфейс системи є складним для не ІТ-користувачів, а процес налаштування та підтримки вимагає залучення HR-фахівців із відповідним досвідом.

Таким чином, попри наявність на ринку значної кількості рішень для обліку та візуалізації професійних навичок, жодне з них не забезпечує достатньої гнучкості, відкритості й простоти використання у контексті індивідуального

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

розвитку IT-фахівців, невеликих команд або освітніх ініціатив. Це створює передумови для розробки спеціалізованого веб-застосунку, який би поєднував зручний інтерфейс, динамічну побудову діаграм компетентностей, гнучке налаштування під конкретні потреби та відкритість для інтеграції з іншими освітніми чи HR-системами [7].

На рисунку 1.5 зображено сервіс CompetencyCore.



Рисунок 1.5. Сервіс CompetencyCore

1.2 Відомості про проєкт

1.2.1 Функціональність веб-застосунку

У межах дипломного проєкту передбачається реалізація низки функціональних можливостей веб-застосунку для побудови діаграми компетентностей IT-фахівця. Запланований функціонал орієнтований на забезпечення зручного, адаптивного та інформативного інтерфейсу, який дозволить користувачам ефективно взаємодіяти із системою, створювати персоналізовані візуалізації та отримувати доступ до супровідної інформації.

1. Навігація. У веб-застосунку буде реалізовано зручну систему навігації, що забезпечуватиме швидкий доступ до основних розділів інтерфейсу: побудови діаграми, довідкового матеріалу та інформації про застосунок. Передбачається використання фіксованого меню з чіткими навігаційними

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

пунктами, доступними на будь-якому етапі роботи, що сприятиме інтуїтивному керуванню та підвищенню користувацького досвіду.

2. Зміна теми. З метою адаптації інтерфейсу до індивідуальних потреб користувача буде передбачено можливість перемикання між світлою (білою) та темною (чорною) темами. Така опція дозволить забезпечити комфорт при роботі в умовах різного освітлення, а також відповідатиме сучасним тенденціям персоналізації цифрового середовища.
3. Зміна мови. У застосунку буде реалізовано функцію перемикання мови інтерфейсу між українською та англійською. Такий підхід дозволить охопити ширше коло користувачів та зробить систему придатною як для локального, так і для міжнародного використання. Зміна мови відбуватиметься динамічно, без перезавантаження сторінки.
4. Побудова діаграми. Ключовою функціональністю стане можливість побудови діаграми компетентностей на основі введених або обраних параметрів. Користувачі зможуть вказувати рівень володіння окремими компетентностями, що дозволить сформувати візуальне уявлення про професійний профіль ІТ-фахівця у вигляді інтерактивної графічної моделі.
5. Завантаження діаграми. Для подальшого використання сформованої діаграми буде реалізовано функцію її збереження у вигляді зображення (формат PNG). Це забезпечить можливість інтеграції результатів до звітної документації, презентацій або портфоліо, сприяючи їх поширенню та аналізу.
6. Про діаграми (мануал). Для підтримки користувачів у процесі створення діаграм буде впроваджено довідковий розділ з інструкцією. У ньому буде подано рекомендації щодо введення даних, пояснення принципів побудови графіка, а також поради щодо тлумачення візуалізованої інформації.
7. Про застосунок. У спеціальному розділі буде надано загальну інформацію про веб-застосунок, включаючи його призначення, основні технічні характеристики, особливості розробки та сферу застосування. Також

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

планується додати контактну інформацію для отримання зворотного зв'язку або подання пропозицій щодо вдосконалення.

Передбачені функціональні можливості спрямовані на створення повноцінного інструменту для візуального представлення рівня професійної підготовки IT-фахівця, що поєднує зручність користування, доступність та практичну цінність.

1.2.2 Обґрунтування обраних технологій

У процесі реалізації веб-застосунку для побудови діаграми компетентностей IT-фахівця було обрано набір сучасних технологій, які забезпечують ефективну, адаптивну та масштабовану розробку. Обрання кожної технології базується на її відповідності до функціональних вимог, зручності використання, підтримці сучасних стандартів та наявності активної спільноти:

1. HTML (HyperText Markup Language) є основною мовою розмітки для структурування веб-інтерфейсу. Вона забезпечує логічну організацію контенту, визначення семантичної структури сторінок, а також слугує основою для інтеграції каскадних таблиць стилів (CSS) та сценаріїв JavaScript. Обрання HTML обумовлене його універсальністю, відповідністю стандартам W3C і широкою сумісністю з усіма сучасними браузерами.
2. SCSS (Sassy CSS) являє собою розширення мови CSS, що використовує синтаксис препроцесора SASS. Застосування SCSS дозволяє структурувати стилі у вигляді модулів, використовувати змінні, вкладеність та міксини. Такі можливості значно підвищують читаність, підтримуваність та повторне використання стилів. Обрання SCSS сприяє підвищенню ефективності розробки інтерфейсу веб-застосунку.
3. CSS (Cascading Style Sheets) забезпечує візуальне оформлення HTML-елементів, що є невід'ємною складовою фронтенд-розробки. Використання CSS необхідне для забезпечення адаптивності інтерфейсу, налаштування кольорової палітри, типографіки, розташування елементів та анімацій. CSS є стандартом для опису стилів і використовується разом із SCSS для формування фінального вигляду сторінок.

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

4. JSON (JavaScript Object Notation) являє собою формат обміну даними, який використовується для зберігання та передачі інформації між клієнтською та серверною частинами застосунку. JSON має просту синтаксичну структуру, що полегшує його обробку за допомогою JavaScript. Його використання доцільне для подання структури компетентностей, опису їхніх властивостей та передачі конфігураційних параметрів веб-застосунку.
5. JavaScript обрано як основну мову програмування для розробки логіки взаємодії користувача з інтерфейсом. Ця мова є стандартом для клієнтської сторони веб-застосунків і підтримується всіма сучасними браузерами. JavaScript дозволяє реалізувати динамічне оновлення вмісту, обробку подій, маніпулювання елементами DOM та взаємодію з API. Його популярність, гнучкість та активна екосистема зумовили вибір саме цієї мови для розробки застосунку.
6. Bootstrap використовується як CSS-фреймворк для швидкого створення адаптивного та уніфікованого інтерфейсу. Завдяки вбудованим компонентам, сітковій системі та типографіці, Bootstrap забезпечує зручне макетування елементів та сумісність з мобільними пристроями. Він сприяє скороченню часу розробки й забезпечує відповідність інтерфейсу сучасним вимогам до UX/UI.
7. Vue.js являє собою JavaScript-фреймворк, обраний для побудови реактивного інтерфейсу користувача. Його особливістю є простота інтеграції, двостороннє зв'язування даних, модульність та можливість компонентного підходу до розробки. Використання Vue.js дозволяє створювати масштабовані та гнучкі інтерфейси, забезпечуючи швидке оновлення стану застосунку без необхідності перезавантаження сторінки [8][9][10].

Загалом, обрані технології взаємодоповнюють одна одну та утворюють цілісне середовище для створення сучасного, функціонального й зручного веб-застосунку для візуалізації компетентностей ІТ-фахівця. Їх використання обґрунтоване не лише технічними характеристиками, але й орієнтацією на майбутню підтримку та масштабування проєкту.

На рисунку 1.6 зображено ієрархічну структурну схему використаних технологій.

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

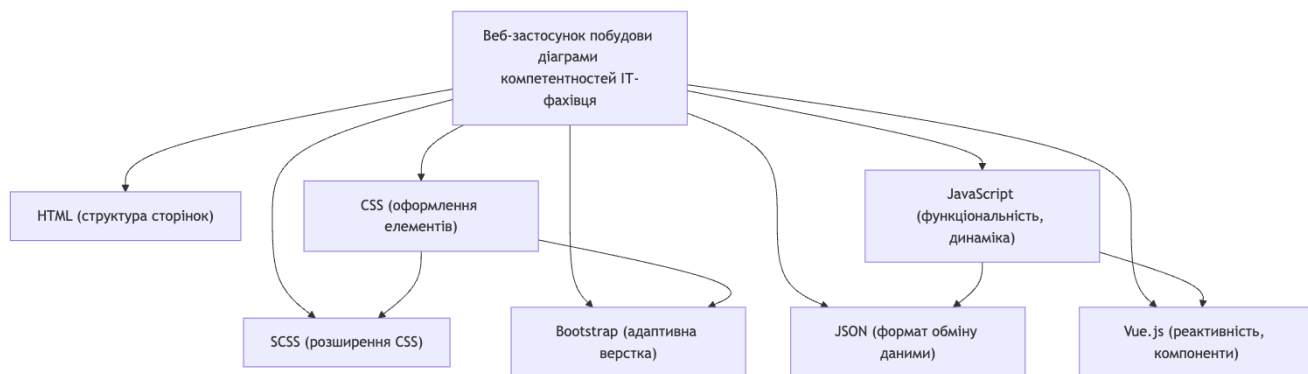


Рисунок 1.6. Ієрархічна структурна схема використаних технологій

1.2.3 Архітектура веб-застосунку

Архітектура веб-застосунку, розробленого для побудови діаграми компетентностей ІТ-фахівця, базується на концепції односторінкового застосунку (SPA – Single Page Application) із реалізацією клієнтського рендерингу (CSR – Client-Side Rendering). SPA-підхід передбачає, що веб-застосунок завантажується як єдина HTML-сторінка з мінімальним серверним вмістом і далі динамічно оновлює вміст на клієнті без перезавантаження всієї сторінки. Така архітектура сприяє швидкій взаємодії користувача із системою, плавності переходів між розділами та зменшенню навантаження на сервер, оскільки обробка логіки відбувається переважно на стороні браузера.

Ключовою особливістю SPA є використання JavaScript-фреймворків, серед яких Vue.js займає вагомe місце завдяки простоті інтеграції, реактивності та можливості компонування інтерфейсу у вигляді окремих модулів – компонентів. У межах CSR застосунок завантажує на клієнт повний набір необхідних ресурсів (HTML, CSS, JavaScript), а подальша генерація й оновлення інтерфейсу відбувається вже локально, що дозволяє досягти високої інтерактивності.

Проте, CSR має і певні обмеження: початкове завантаження сторінки може бути повільнішим через необхідність завантаження великих JavaScript-бандлів, а також спостерігаються складнощі з індексацією вмісту пошуковими системами, оскільки початковий HTML часто є майже порожнім. Для часткового розв’язання цих проблем іноді застосовують SSR (Server-Side Rendering), коли HTML генерується на сервері, що прискорює перше відображення сторінки та покращує SEO. Однак, у даному проекті переваги SPA і CSR як основи архітектури

дозволяють сфокусуватися на динамічності, зручності користування та масштабованості.

Структура папок Vue-проєкту, що використовується для реалізації веб-застосунку, має класичний вигляд, який забезпечує зрозумілість, модульність і простоту підтримки коду. Основні каталоги включають:

1. `src` – головна папка з вихідним кодом застосунку, що містить усі ключові підмодулі;
2. `components` – папка з багаторазовими Vue-компонентами, які реалізують окремі UI-елементи (кнопки, діалоги, графіки), що можуть повторно використовуватись у різних частинах застосунку;
3. `views` – компоненти вищого рівня, які відповідають за повноцінні сторінки або великі розділи, що формують навігаційну структуру SPA;
4. `assets` – статичні файли, такі як зображення, шрифти, файли стилів, які підключаються до застосунку та використовуються для візуального оформлення;
5. `router` – конфігураційні файли для Vue Router, що визначають маршрутизацію між сторінками SPA, налаштовують переходи та параметри URL;
6. `store` – централізоване сховище стану, зазвичай реалізоване через Vuex або Pinia, яке відповідає за управління даними застосунку, включно з інформацією про компетентності, поточний стан інтерфейсу та обробку дій користувача;
7. `plugins` (за наявності) – модулі для підключення сторонніх бібліотек та плагінів, які розширюють функціонал застосунку;
8. `utils` або `helpers` – допоміжні функції та сервіси, які забезпечують повторне використання логіки (наприклад, форматування даних, робота з API).

Використання такої організації структури проєкту забезпечує розподіл відповідальності між різними частинами коду, полегшує масштабування проєкту і сприяє командній розробці.

Отже, поєднання SPA-архітектури з CSR та продумана модульна структура Vue-проєкту створюють надійну, масштабовану та продуктивну основу для веб-застосунку побудови діаграм компетентностей ІТ-фахівця, що відповідає сучасним

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

вимогам інтерфейсної інтерактивності, швидкості відгуку та зручності користування.

На рисунку 1.7 зображено ієрархічну структурну схему папок застосунку vue.

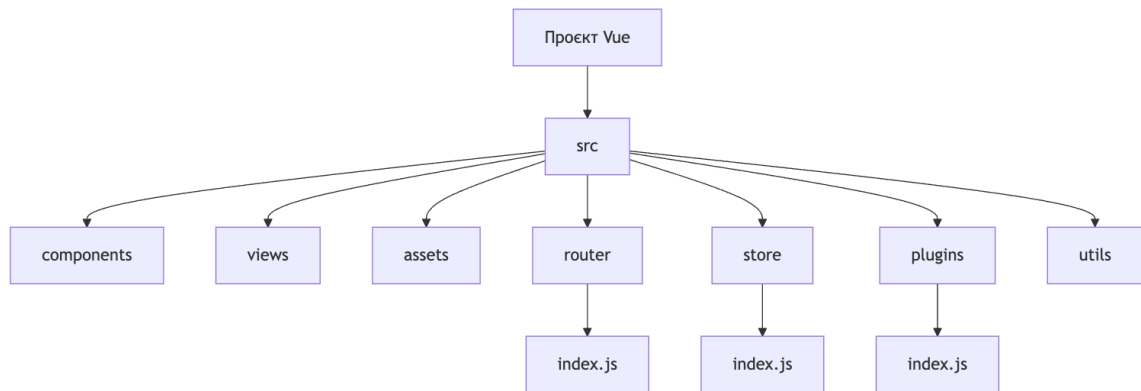


Рисунок 1.6. Ієрархічна структурна схема папок застосунку vue

На рисунку 1.7 зображено файлову структуру застосунку vue у редакторі коду Visual Studio Code.

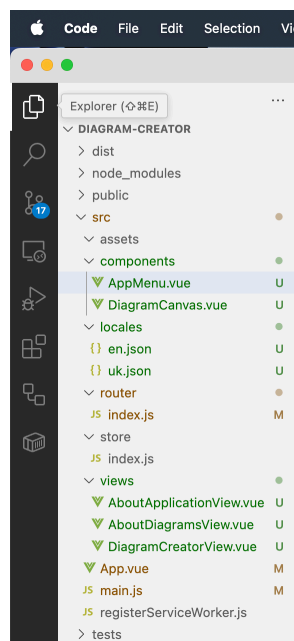


Рисунок 1.7. Файлова структура застосунку vue у редакторі коду MS VSCode

1.2.4 Концепція дизайну

Для розробки веб-застосунку побудови діаграми компетентностей ІТ-фахівця було обрано стратегічний підхід, що ґрунтується на принципах мінімалізму (minimalism) та плоского дизайну (flat design). Такий вибір

обумовлений прагненням забезпечити максимально простий, зрозумілий і водночас сучасний інтерфейс користувача, який не відволікає від основного функціоналу і сприяє зручній навігації.

Стратегія мінімалізму передбачає усунення зайвих елементів інтерфейсу, використання обмеженої кольорової палітри та акцент на чіткій типографіці. Це дозволяє користувачеві швидко орієнтуватися у застосунку та ефективно виконувати поставлені завдання без перевантаження візуальною інформацією.

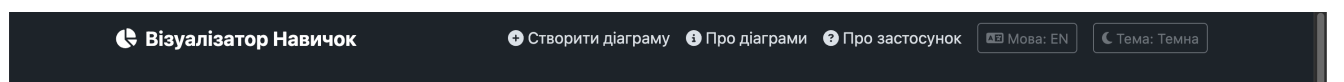
Принципи flat design реалізовані через застосування простих геометричних форм, відсутність складних градієнтів, тіней та тривимірних ефектів. Такий стиль забезпечує чистоту візуального оформлення і відповідає сучасним трендам у веб-дизайні, сприяючи одночасно швидкому завантаженню інтерфейсу та його адаптивності.

Навігація веб-застосунку реалізована у вигляді верхнього меню (navbar), що містить посилання на основні розділи: створення діаграми, інформацію про діаграми та відомості про застосунок. Інтерфейс підтримує двомовність (українська та англійська мови) з можливістю швидкого перемикавання, а також має функціонал зміни теми оформлення між світлою та темною, що підвищує комфорт користування застосунком у різних умовах освітлення.

Технічна реалізація навігаційного меню побудована на сучасних веб-технологіях із використанням фреймворку Vue.js та бібліотеки i18n для інтернаціоналізації. Динамічна зміна теми оформлення здійснюється через зміну атрибуту data-bs-theme у кореневому елементі документа, що дозволяє інтегрувати теми у стилізацію, засновану на CSS змінних. Кнопки перемикавання мови та теми інтуїтивно зрозумілі і забезпечують миттєву реакцію інтерфейсу без необхідності перезавантаження сторінки.

Концепція дизайну веб-застосунку орієнтована на створення зручного, функціонального та естетично привабливого інструменту, що сприяє ефективному використанню системи побудови діаграм компетентностей ІТ-фахівця.

На рисунку 1.8 зображено основну концепцію дизайну на прикладі навігації.



					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

Рисунок 1.8. Основна концепція дизайну на прикладі навігації

На рисунку 1.9 зображено ієрархічну структурну схеми навігації між розділами.

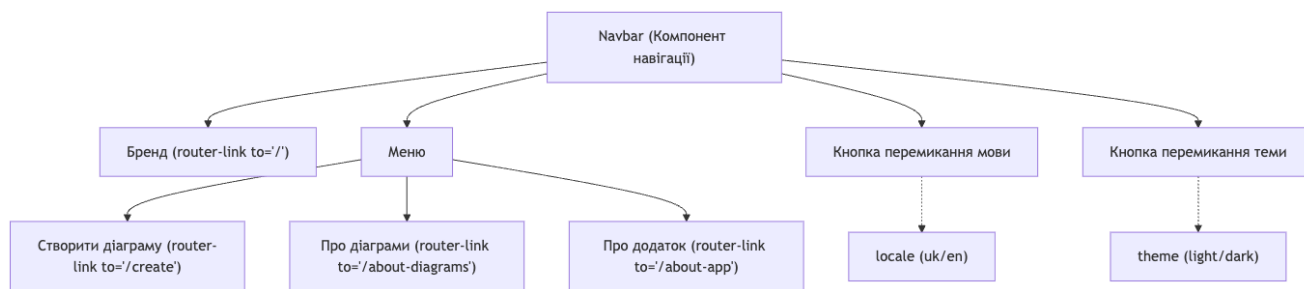


Рисунок 1.9. Ієрархічну структурну схеми навігації між розділами

1.3 Розробка сайту

1.3.1 Налаштування оточення

Налаштування оточення для розробки веб-застосунку побудови діаграми компетентностей ІТ-фахівця включає використання низки інструментів і бібліотек, кожен з яких виконує важливу роль у процесі створення програмного продукту. Нижче наведено детальний опис основних складових оточення.

1. Редактор коду Microsoft Visual Studio Code

Це сучасне середовище розробки, яке підтримує широкий спектр мов програмування, включно з JavaScript та Vue.js. Visual Studio Code забезпечує зручний інтерфейс, інтеграцію з системами контролю версій (Git), вбудоване відлагодження, а також підтримку численних розширень, що сприяють підвищенню продуктивності розробника. Наявність інтелектуальних підказок і автоматичного форматування коду допомагає підтримувати високу якість програмного продукту.

2. Vue.js (версія 3.2.13)

Основний фронтенд-фреймворк, що забезпечує реактивність інтерфейсу, компонентний підхід до розробки та ефективне управління станом додатка. Vue.js полегшує створення динамічних користувацьких інтерфейсів та забезпечує високу продуктивність.

3. Vue Router (версія 4.0.3)

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

Офіційний модуль маршрутизації для Vue.js, що дозволяє реалізувати односторінкові застосунки (SPA) з можливістю переходу між різними в'ю без повного перезавантаження сторінки. Забезпечує гнучке налаштування маршрутів і управління навігацією.

4. Vuex (версія 4.0.0)

Система централізованого збереження та управління станом додатка. Дозволяє підтримувати єдине джерело істини для даних застосунку, що спрощує їх синхронізацію між компонентами і підвищує передбачуваність поведінки.

5. vue-i18n (версія 11.1.5)

Бібліотека для реалізації інтернаціоналізації інтерфейсу. Дозволяє підтримувати кілька мов і локалізацій, що робить застосунок більш доступним для користувачів з різних регіонів.

6. Bootstrap (версія 5.3.6)

Популярний CSS-фреймворк для швидкої та адаптивної верстки інтерфейсу. Забезпечує готові стилі, компоненти і сіткову систему, що прискорює процес дизайну та підвищує зручність використання.

7. @fortawesome/fontawesome-svg-core, @fortawesome/free-solid-svg-icons, @fortawesome/vue-fontawesome

Набір бібліотек для роботи з іконками FontAwesome у вигляді SVG. Забезпечують високоякісні та масштабовані іконки, які легко інтегруються у Vue-компоненти для покращення візуального оформлення інтерфейсу.

8. html2canvas (версія 1.4.1)

Бібліотека, що дозволяє перетворювати DOM-елементи у зображення формату Canvas. Це є ключовим функціоналом для реалізації можливості збереження побудованих діаграм у вигляді графічних файлів.

9. register-service-worker (версія 1.7.2)

Інструмент для реєстрації сервіс-воркерів у браузері, що дозволяє реалізувати PWA (Progressive Web App) функціонал, покращуючи продуктивність і забезпечуючи кешування ресурсів для роботи в офлайн-режимі.

10. core-js (версія 3.8.3)

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

Бібліотека полівільлів (polyfills), що забезпечує сумісність сучасних JavaScript-функцій із різними браузерами, у тому числі з тими, які не підтримують останні стандарти ECMAScript.

11. Babel та пов'язані плагіни (@babel/core, babel-jest, @babel/eslint-parser)

Інструменти для транспіляції сучасного JavaScript-коду у сумісний зі старими браузерами формат, що забезпечує широку доступність застосунку.

12. ESLint та eslint-plugin-vue

Інструменти статичного аналізу коду, що допомагають підтримувати єдиний стиль програмування, виявляти помилки та потенційні проблеми в коді на ранніх стадіях розробки.

13. Jest та пов'язані пакети (@vue/cli-plugin-unit-jest, @vue/test-utils, @vue/vue3-jest)

Тестові фреймворки, що забезпечують написання та виконання юніт-тестів для перевірки коректності роботи окремих модулів застосунку.

14. Sass та sass-loader

Препроцесор для CSS, що розширює можливості стилізації за рахунок змінних, вкладеності, міксинів та інших зручностей, що підвищують підтримуваність стилів.

Таким чином, комбінування зазначених інструментів і бібліотек створює комплексне, сучасне та продуктивне оточення для розробки веб-застосунку, що відповідає актуальним стандартам і вимогам.

Нижче наведено фрагмент програмного коду файлу package.json, що має всі ці залежності:

```
"dependencies": {  
  "@fortawesome/fontawesome-svg-core": "^6.7.2",  
  "@fortawesome/free-solid-svg-icons": "^6.7.2",  
  "@fortawesome/vue-fontawesome": "^3.0.8",  
  "bootstrap": "^5.3.6",  
  "core-js": "^3.8.3",  
  "html2canvas": "^1.4.1",  
  "register-service-worker": "^1.7.2",  
  "vue": "^3.2.13",  
  "vue-i18n": "^11.1.5",  
  "vue-router": "^4.0.3",  
  "vuex": "^4.0.0"  
},
```

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

Нижче наведено фрагмент програмного код файлу App.vue, що є головною точкою в програму:

```
<template>
  <div id="app">
    <AppMenu />
    <router-view />
  </div>
</template>

<script setup>
import AppMenu from './components/AppMenu.vue'
</script>
```

Нижче наведено фрагмент програмного код файлу main.js, що підключає всі прт-пакети, що були завантажені з файлу package.json:

```
import { createApp } from 'vue'
import App from './App.vue'
import router from './router'
import { createI18n } from 'vue-i18n'

import en from './locales/en.json'
import uk from './locales/uk.json'

import 'bootstrap/dist/css/bootstrap.min.css'
import 'bootstrap'

// Імпорт Font Awesome
import { library } from '@fortawesome/fontawesome-svg-core'
import { FontAwesomeIcon } from '@fortawesome/vue-fontawesome'
import { faSun, faMoon, faGlobe, faLanguage, faChartPie, faPlusCircle, faInfoCircle, faCircleQuestion } from '@fortawesome/free-solid-svg-icons'

library.add(faSun, faMoon, faGlobe, faLanguage, faChartPie, faPlusCircle, faInfoCircle, faCircleQuestion)

// Додаємо іконки в бібліотеку
library.add(faSun, faMoon)

const i18n = createI18n({
  locale: 'uk',
  fallbackLocale: 'en',
  messages: { en, uk },
})

const app = createApp(App)

app.component('font-awesome-icon', FontAwesomeIcon)

app.use(router)
app.use(i18n)

app.mount('#app')
```

1.3.2 Розробка компоненту навігації

Компонент навігації Navbar у межах веб-застосунку побудови діаграми компетентностей ІТ-фахівця призначений для організації швидкого перемикання

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

між ключовими розділами системи. Він забезпечує постійну доступність головних маршрутів, дозволяючи користувачу легко орієнтуватися в інтерфейсі. Важливою особливістю цього компонента є наявність елементів, які забезпечують персоналізацію – перемикання мови інтерфейсу та теми візуального оформлення.

Функціональність компонента охоплює:

1. Відображення логотипа та назви застосунку;
2. Маршрутизацію до головної сторінки, створення діаграми, інформації про діаграми та опису застосунку;
3. Перемикання мови інтерфейсу між українською та англійською;
4. Перемикання теми між світлою та темною;
5. Підтримку адаптивного дизайну для різних розмірів екранів.

Основний код HTML-структури компонента наведено нижче:

```
<template>
  <nav class="navbar">
    <div class="container px-3">
      <router-link to="/" class="navbar-brand d-flex align-items-center gap-2">
        <font-awesome-icon icon="chart-pie" />
        <span><strong>{{ t('menu.app_title') }}</strong></span>
      </router-link>

      <div class="d-flex gap-3 align-items-center">
        <router-link to="/create" class="nav-link d-flex align-items-center gap-1">
          <font-awesome-icon icon="plus-circle" />
          {{ t('menu.create_diagram') }}
        </router-link>

        <router-link to="/about-diagrams" class="nav-link d-flex align-items-center gap-1">
          <font-awesome-icon icon="info-circle" />
          {{ t('menu.about_diagrams') }}
        </router-link>

        <router-link to="/about-app" class="nav-link d-flex align-items-center gap-1">
          <font-awesome-icon icon="circle-question" />
          {{ t('menu.about_app') }}
        </router-link>

        <!-- Кнопка перемикання мови -->
        <button @click="toggleLang" class="btn btn-outline-secondary btn-sm d-flex align-items-center gap-1">
          <font-awesome-icon :icon="locale.value === 'uk' ? 'globe' : 'language'" />
          Мова: {{ locale === 'uk' ? 'EN' : 'UA' }}
        </button>

        <!-- Кнопка перемикання теми -->
        <button @click="toggleTheme" class="btn btn-outline-secondary btn-sm d-flex align-items-center gap-1">
          <font-awesome-icon :icon="theme === 'light' ? 'sun' : 'moon'" />
          Тема: {{ theme === 'light' ? 'Світла' : 'Темна' }}
        </button>
      </div>
    </div>
  </nav>
```

					РП 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

```

        </button>
      </div>
    </div>
  </nav>
</template>

```

Пояснення до коду:

1. Тег <nav> реалізує основний контейнер навігаційної панелі.
2. Всередині контейнера розміщено логотип і назву застосунку, реалізовані за допомогою компонента router-link та іконки font-awesome-icon.
3. Основні пункти меню (Створити діаграму, Про діаграми, Про застосунок) також реалізовано через router-link, що забезпечує маршрутизацію без перезавантаження сторінки.
4. Кнопка перемикавання мови активує метод toggleLang, який змінює значення мови та оновлює текст відповідно до поточної локалі.
5. Кнопка перемикавання теми активує метод toggleTheme, що оновлює тему інтерфейсу на світлу або темну.
6. Усі елементи вирівняно за допомогою класів Bootstrap (d-flex, gap-1, align-items-center), що забезпечує адаптивний дизайн та зручність інтерфейсу.

Нижче наведено фрагмент програмного код файлу index.js у директорії router, що робить цю розмітку навігації працездатною:

```

import { createRouter, createWebHistory } from 'vue-router'
import DiagramCreatorView from '../views/DiagramCreatorView.vue'
import AboutDiagramsView from '../views/AboutDiagramsView.vue'
import AboutApplicationView from '../views/AboutApplicationView.vue'

const routes = [
  { path: '/', redirect: '/create' },
  { path: '/create', component: DiagramCreatorView },
  { path: '/about-diagrams', component: AboutDiagramsView },
  { path: '/about-app', component: AboutApplicationView }
]

const router = createRouter({
  history: createWebHistory(),
  routes
})

export default router

```

Компонент навігації виконує роль основного засобу перемикавання між функціональними модулями застосунку, при цьому забезпечуючи гнучкість відображення та підтримку користувацьких налаштувань, таких як мова та тема

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

оформлення. Його реалізація відповідає сучасним вимогам до SPA-застосунків та адаптивного дизайну.

1.3.3 Розробка механізму динамічних тем

Компонент реалізує механізм перемикання між світлою та темною темами інтерфейсу веб-застосунку. Його мета – надати користувачам можливість індивідуально налаштовувати вигляд інтерфейсу, що сприяє зручності використання, особливо в умовах різного освітлення. Назва компоненту безпосередньо не вказується у файлі, проте логічно він відповідає за навігаційну панель та налаштування візуального оформлення – умовно його можна ідентифікувати як `NavbarComponent`.

Компонент забезпечує такі функції:

1. динамічне перемикання між світлою (`light`) та темною (`dark`) темами;
2. встановлення початкової теми при завантаженні сторінки;
3. відображення актуальної теми в інтерфейсі;
4. інтерактивне оновлення теми інтерфейсу шляхом зміни відповідного атрибута HTML-документа (`data-bs-theme`), що є стандартним підходом у межах бібліотеки `Bootstrap 5`;
5. паралельно також реалізовано функцію перемикання мови, однак вона не розглядається в цьому розділі, оскільки не стосується тематики динамічного оформлення.

Основний код компоненту:

```
<script setup>
import { ref, watch } from 'vue'

const theme = ref('light')

function toggleTheme() {
  theme.value = theme.value === 'light' ? 'dark' : 'light'
}

watch(theme, (newTheme) => {
  document.documentElement.setAttribute('data-bs-theme', newTheme)
})

document.documentElement.setAttribute('data-bs-theme', theme.value)
</script>

<template>
  <!-- Кнопка перемикання теми -->
```

					РП 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

```

    <button @click="toggleTheme" class="btn btn-outline-secondary btn-sm d-flex align-items-center gap-1">
      <font-awesome-icon :icon="theme === 'light' ? 'sun' : 'moon'" />
      Тема: {{ theme === 'light' ? 'Світла' : 'Темна' }}
    </button>
  </template>

```

Пояснення основного коду:

1. `import { ref, watch } from 'vue'`: Імпортуються функції реактивності з фреймворку Vue, що дозволяє створювати та спостерігати за змінами стану.
2. `const theme = ref('light')`: Змінна `theme` є реактивним посиланням на поточну тему інтерфейсу, ініціалізовану як `'light'`.
3. `function toggleTheme() { ... }`: Функція перемикає значення змінної `theme` між `'light'` і `'dark'` кожного разу при натисканні кнопки.
4. `watch(theme, (newTheme) => { ... })`: Спостерігач автоматично реагує на зміну значення `theme` і відповідно оновлює атрибут `data-bs-theme` в кореневому HTML-елементі `document.documentElement`, що забезпечує негайне застосування нової теми в усьому інтерфейсі.
5. `document.documentElement.setAttribute('data-bs-theme', theme.value)`: Команда ініціалізації теми одразу при завантаженні сторінки – встановлює актуальне значення `theme` як атрибут HTML.
6. У шаблоні `<template>` розміщена кнопка, яка викликає `toggleTheme()` при кліку. Залежно від поточної теми, змінюється іконка (`sun` або `moon`) та відповідна назва ("Світла" або "Темна"), що покращує зручність користування.

Описаний механізм забезпечує гнучке та зручне перемикання між візуальними темами веб-застосунку без потреби в перезавантаженні сторінки, з дотриманням принципів реактивності, притаманних фреймворку Vue.js.

1.3.4 Розробка механізму динамічної зміни мови

1.3.3. Розробка механізму динамічних тем

Компонент, що реалізує механізм динамічних тем у веб-застосунку побудови діаграми компетентностей ІТ-фахівця, має назву `Navbar` і відповідає за управління інтерфейсом користувача, зокрема за динамічне перемикання між світлою та темною темами оформлення. Його основним призначенням є забезпечення

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

гнучкості візуального сприйняття застосунку шляхом адаптації інтерфейсу відповідно до побажань користувача, що покращує зручність і комфорт роботи з додатком.

Основний словник (dictionary) компоненту наведено нижче у форматі JSON:

```
{
  "menu": {
    "app_title": "Skill Visualizer",
    "create_diagram": "Create Diagram",
    "about_diagrams": "About Diagrams",
    "about_app": "About Application"
  },
  "aboutApp": {
    "title": "About Application",
    "lead": "This app helps you visualize your skills in different professional profile shapes (I, T, V, P, Combi).",
    "text": "Adjust skill levels using sliders and download your diagram for use in portfolios or personal websites."
  },
  "aboutDiagrams": {
    "title": "Diagram Types",
    "i_shaped": {
      "title": "I-Shaped",
      "description": "Deep expertise in one area."
    },
    "t_shaped": {
      "title": "T-Shaped",
      "description": "One deep skill with broad general skills."
    },
    "v_shaped": {
      "title": "V-Shaped",
      "description": "Deep analytical and methodological skills."
    },
    "p_shaped": {
      "title": "P-Shaped",
      "description": "Two deep skills (e.g., design + tech)."
    },
    "combi_shaped": {
      "title": "Combi-Shaped",
      "description": "Versatile across many domains."
    }
  },
  "canvas": {
    "name": "Name",
    "description": "Description",
    "date": "Date",
    "type": "Type",
    "based_on": "Based on",
    "support": "I-Shaped diagram supports only one skill!"
  },
  "create": {
    "title": "Create",
    "diagram": "Diagram",
    "createdAt": "Created at",
    "name": "Name",
    "description": "Description",
    "type": "Diagram Type",
    "basedOn": "Diagram is based on",
    "hardSkills": "Hard Skills",
  }
}
```

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

```

    "softSkills": "Soft Skills",
    "skillsComma": "Skills (comma-separated)",
    "addSkills": "Add Skills",
    "leftSkills": "Left Side Skills (comma-separated)",
    "rightSkills": "Right Side Skills (comma-separated)",
    "setVShape": "Set V-Shaped Skills",
    "level": "Level",
    "color": "Color",
    "download": "Download Diagram",
    "save": "Save to JSON",
    "load": "Load from JSON"
  }
}

```

Код основної логіки:

```

<script setup>
import { ref } from 'vue'
import { useI18n } from 'vue-i18n'

const { locale } = useI18n()

function toggleLang() {
  locale.value = locale.value === 'uk' ? 'en' : 'uk'
}
</script>

```

Пояснення основного коду зміни мови:

1. Імпортується реактивна функція `ref` та хук локалізації `useI18n` з бібліотеки `vue-i18n`.
2. З `useI18n` отримується об'єкт, з якого деструктурується реактивне посилання `locale` – поточна мова інтерфейсу.
3. Оголошується функція `toggleLang`, яка змінює значення `locale.value` між `'uk'` (українська) та `'en'` (англійська).
4. Зміна значення `locale.value` призводить до автоматичного оновлення текстів інтерфейсу відповідно до вибраної мови без перезавантаження сторінки.

Такий підхід забезпечує зручне і швидке перемикання мов, покращуючи інтернаціоналізацію застосунку.

1.3.5 Розробка представлення «Створити діаграму»

Представлення `CreateDiagram` призначений для інтерфейсної реалізації функціоналу створення діаграми компетентностей ІТ-фахівця. Він забезпечує користувача інструментами введення назв, описів, вибору типу діаграми (I-Shaped, T-Shaped, V-Shaped, P-Shaped, Combi-Shaped), а також формування набору навичок (Hard Skills або Soft Skills). Компонент служить основним інтерфейсом для

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

введення даних, які в подальшому використовуються для візуалізації діаграми компетентностей.

Функціональність компоненту

1. Основна функціональність компонента включає:
2. Введення та редагування назви та опису діаграми.
3. Вибір типу діаграми зі списку доступних форм.
4. Вибір категорії навичок: Hard Skills або Soft Skills.
5. Додавання навичок у вигляді текстового списку або у випадку V-Shaped діаграми – розподіл навичок за двома сторонами (ліва та права).
6. Встановлення рівня володіння навичкою (від 0 до 100%) за допомогою повзунка.
7. Вибір кольору для кожної навички.
8. Видалення навичок із діаграми.
9. Збереження даних діаграми у JSON-файл та завантаження у вигляді зображення.
10. Завантаження збережених діаграм з JSON-файлу для подальшого редагування.

Цей компонент взаємодіє з дочірнім компонентом DiagramCanvas, який відповідає за безпосередню візуалізацію діаграми.

Основний код компоненту

```
const addSkillsFromInput = () => {
  const names = skillsInput.value.split(',').map(s => s.trim()).filter(s => s !== '')
  names.forEach(name => {
    skills.value.push({
      name,
      level: 50,
      color: getRandomColor()
    })
  })
  skillsInput.value = ''
}

const setVShapeSkills = () => {
  const leftNames = leftSkillsInput.value.split(',').map(s => s.trim()).filter(s => s)
  const rightNames = rightSkillsInput.value.split(',').map(s => s.trim()).filter(s => s)
  const combined = []

  leftNames.forEach(name => {
    let skill = skills.value.find(s => s.name === name)
    if (!skill) {
```

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

```

        skill = { name, level: 50, color: getRandomColor() }
        skills.value.push(skill)
    }
    combined.push({ ...skill, side: 'left' })
})

rightNames.forEach(name => {
    let skill = skills.value.find(s => s.name === name)
    if (!skill) {
        skill = { name, level: 50, color: getRandomColor() }
        skills.value.push(skill)
    }
    combined.push({ ...skill, side: 'right' })
})

vShapeSkills.value = combined

// Очищення полів вводу
leftSkillsInput.value = ''
rightSkillsInput.value = ''
}

```

Пояснення основного коду

1. Функція `addSkillsFromInput` реалізує механізм додавання навичок користувача у вигляді коми розділеного рядка. Спершу вона розбиває введений текст на окремі назви навичок, очищає їх від пробілів та відкидає порожні значення. Для кожної навички створюється об'єкт із початковим рівнем володіння 50% та випадковим кольором, після чого навичка додається до списку `skills`. Після додавання поле вводу очищається.
2. Функція `setVShapeSkills` використовується для спеціального випадку V-Shaped діаграми, де навички розподіляються по двох боках. Вона обробляє два окремі поля вводу – для лівої та правої сторони. Кожне ім'я навички перевіряється у загальному списку `skills`, якщо навички немає – вона додається із стандартними параметрами. Потім для кожної навички створюється копія з позначкою сторони (`side: 'left'` або `'right'`), яка додається у спеціальний список `vShapeSkills`, що використовується для рендерингу V-Shaped діаграми. В кінці поля вводу очищуються для покращення UX.

Ці функції є ключовими для логіки формування набору навичок, що визначають зовнішній вигляд та зміст діаграми компетентностей у веб-застосунку.

Компонент `"DiagramCanvas"` призначений для візуалізації рівня професійних навичок ІТ-фахівця у вигляді інтерактивної діаграми різних типів (I-Shaped, T-Shaped, P-Shaped, Combi-Shaped, V-Shaped). Цей компонент забезпечує

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

наочне відображення компетентностей за обраною формою, що сприяє кращому аналізу сильних сторін та напрямків розвитку спеціаліста.

Компонент приймає набір навичок користувача у вигляді масиву об'єктів, де кожен об'єкт містить назву навички, рівень володіння нею (у відсотках), колір для візуального відображення та інші параметри. Залежно від обраної форми діаграми (shape) компонент динамічно формує відповідну SVG-графіку, де кожна навичка відображається у вигляді прямокутника з висотою, пропорційною рівню компетентності.

Підтримуються п'ять основних типів діаграм:

1. I-Shaped – одна навичка із вертикальним баром;
2. T-Shaped – одна основна навичка 100% і додаткові навички збоку;
3. P-Shaped – дві навички на 100% з розташуванням по краях;
4. Combi-Shaped – комбіноване розміщення навичок;
5. V-Shaped – розподіл навичок за двома "крилами" у формі V.

Крім того, компонент відображає інформацію про назву діаграми, опис, тип компетенції та дату її створення.

Нижче наведено ключовий фрагмент коду, що відповідає за обчислення позицій та побудову SVG-елементів діаграми залежно від обраної форми:

```
computed: {
  sortedSkills() {
    return [...this.skills].sort((a, b) => b.level - a.level);
  },
  shapeHundredPercentCount() {
    return this.sortedSkills.filter(skill => Number(skill.level) === 100).length;
  },
  computedEdgeMiddleSkills() {
    const sorted = this.sortedSkills;
    const total = sorted.length;

    if (total === 0) return [];

    if (total === 1) {
      return [{
        ...sorted[0],
        x: this.svgWidth / 2 - this.barWidth / 2,
      }];
    }

    if (total === 2) {
      return [
        { ...sorted[0], x: this.gap },
        { ...sorted[1], x: this.svgWidth - this.barWidth - this.gap },
      ];
    }
  }
}
```

					РП 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

```

    }

    const leftX = this.gap;
    const rightX = this.svgWidth - this.barWidth - this.gap;
    const middleSkills = sorted.slice(2);
    const middleCount = middleSkills.length;

    const totalMiddleSpace = rightX - leftX - this.barWidth;
    const gapCount = middleCount + 1;
    const gapSize = totalMiddleSpace / gapCount;

    const middlePositions = middleSkills.map((skill, i) => ({
      ...skill,
      x: leftX + this.barWidth + gapSize * (i + 1) - this.barWidth / 2,
    }));

    return [
      { ...sorted[0], x: leftX },
      ...middlePositions,
      { ...sorted[1], x: rightX },
    ];
  },
  computedVShapeSkills() {
    if (!this.skills.length) return [];

    const sorted = [...this.skills].sort((a, b) => b.level - a.level);
    const half = Math.ceil(sorted.length / 2);
    const leftSkills = sorted.slice(0, half);
    const rightSkills = sorted.slice(half);

    const centerX = this.svgWidth / 2 - this.barWidth / 2;
    const baseY = this.svgHeight - 60;
    const verticalStep = 50;
    const horizontalStep = 60;

    const positioned = [];

    if (leftSkills.length > 0) {
      positioned.push({
        ...leftSkills[0],
        x: centerX - horizontalStep / 2,
        y: baseY - leftSkills[0].level * this.scaleFactor,
      });
    }

    if (rightSkills.length > 0) {
      positioned.push({
        ...rightSkills[0],
        x: centerX + horizontalStep / 2,
        y: baseY - rightSkills[0].level * this.scaleFactor,
      });
    }

    for (let i = 1; i < leftSkills.length; i++) {
      positioned.push({
        ...leftSkills[i],
        x: centerX - horizontalStep * (i + 0.5),
        y: baseY - verticalStep * i - leftSkills[i].level * this.scaleFactor,
      });
    }

    for (let i = 1; i < rightSkills.length; i++) {

```

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

```

        positioned.push({
            ...rightSkills[i],
            x: centerX + horizontalStep * (i + 0.5),
            y: baseY - verticalStep * i - rightSkills[i].level * this.scaleFactor,
        });
    }

    return positioned;
}
}
}

```

В наведеному кодї реалізовано основну логіку розташування навичок на діаграмі. Обчислюються позиції прямокутників, що представляють рівень компетентності, залежно від типу діаграми:

1. `sortedSkills` – повертає відсортований за спаданням рівня масив навичок.
2. `shapeHundredPercentCount` – рахує кількість навичок із рівнем 100%, що є обмеженням для деяких типів діаграм.
3. `computedEdgeMiddleSkills` – для типів діаграм P-Shaped, Combi-Shaped та подібних розраховує координати по горизонталі для лівого, правого країв та проміжних навичок, рівномірно розподіляючи їх між двома фіксованими крайніми точками.
4. `computedVShapeSkills` – розподіляє навички у вигляді двох "крил" V-образної діаграми, розподіляючи позиції по горизонталі та вертикалі з урахуванням масштабу рівня навички.

Ця логіка дозволяє динамічно адаптувати візуалізацію залежно від даних, забезпечуючи гнучкий та інтуїтивно зрозумілий інтерфейс для користувача, що дозволяє легко інтерпретувати рівень компетентностей.

На рисунку 1.10 зображено блок схему алгоритму генерації SVG діаграм.

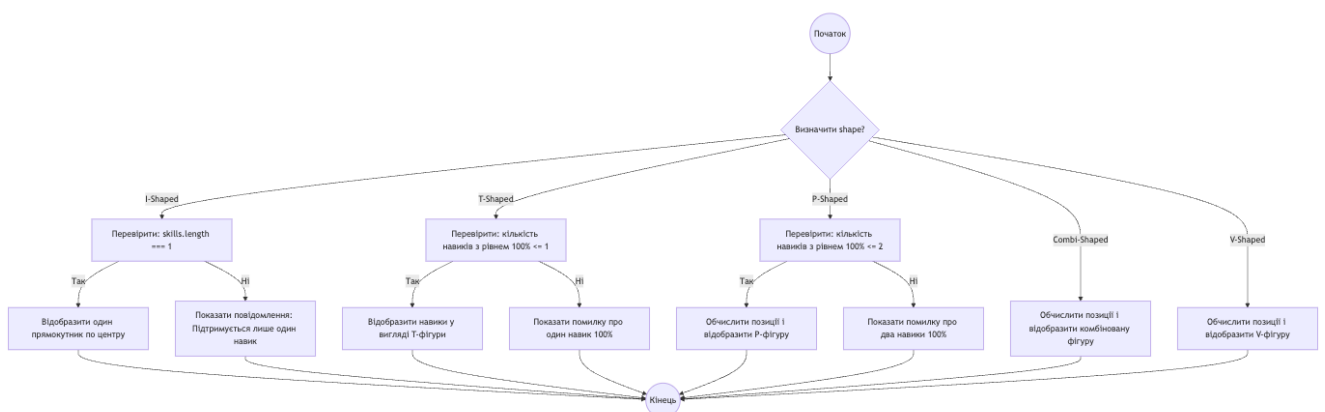


Рисунок 1.10. Блок схема генерації SVG діаграм

1.3.6 Розробка механізму збереження прогресу у JSON

Даний функціональний модуль реалізує механізм збереження та відновлення стану діаграми компетентностей ІТ-фахівця за допомогою форматування даних у форматі JSON. Це дозволяє користувачу зберігати створені діаграми локально у вигляді файлу, а також завантажувати їх пізніше для перегляду чи редагування. Такий підхід є критично важливим для забезпечення безперервності роботи користувача, персоналізації діаграм, а також для підтримки можливості обміну шаблонами між користувачами.

Функціональність компонента включає дві основні операції:

1. Збереження поточної діаграми у форматі JSON: на основі заповнених користувачем полів, таких як назва, опис, тип форми діаграми (I, T, V, тощо), перелік навичок та рівень їх розвитку, генерується структура об'єкта, яка серіалізується у JSON та завантажується як файл.
2. Завантаження діаграми з JSON-файлу: при виборі відповідного JSON-файлу виконується десеріалізація даних, які використовуються для відновлення попереднього стану інтерфейсу (всі навички, рівні, кольори, тип діаграми тощо).

Основний код представлення:

```
const saveToJson = () => {
  dateTime.value = getCurrentDateTime()
  const data = {
    diagramName: diagramName.value,
    diagramDescription: diagramDescription.value,
    selectedShape: selectedShape.value,
    skillType: skillType.value,
    dateTime: dateTime.value,
    skills: skills.value,
    vShapeSkills: vShapeSkills.value,
  }
  const jsonStr = JSON.stringify(data, null, 2)
  const blob = new Blob([jsonStr], { type: 'application/json' })
  const url = URL.createObjectURL(blob)
  const a = document.createElement('a')
  a.href = url
  a.download = `${diagramName.value || selectedShape.value}-Diagram.json`
  a.click()
  URL.revokeObjectURL(url)
}

const loadFromJSON = (event) => {
  const file = event.target.files[0]
  if (!file) return
```

					РП 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

```

const reader = new FileReader()
reader.onload = (e) => {
  try {
    const data = JSON.parse(e.target.result)
    diagramName.value = data.diagramName || ''
    diagramDescription.value = data.diagramDescription || ''
    selectedShape.value = data.selectedShape || 'I-Shaped'
    skillType.value = data.skillType || 'Hard Skills'
    dateTime.value = data.dateTime || getCurrentDateTime()
    skills.value = Array.isArray(data.skills) ? data.skills : []
    vShapeSkills.value = Array.isArray(data.vShapeSkills) ? data.vShapeSkills : []
  } catch (err) {
    alert('Failed to load diagram: Invalid JSON')
  }
}
reader.readAsText(file)
}

```

Пояснення основного коду:

1. Функція `saveToJson` відповідає за формування структури даних діаграми у форматі JSON та ініціалізацію процесу завантаження файлу. Всі параметри діаграми зчитуються з реактивних змінних `ref()` у `Vue.js`, включаючи список навичок (`skills`), тип діаграми (`selectedShape`), мета-інформацію (назва, опис, дата створення). За допомогою API `Blob` створюється об'єкт, який зберігається у тимчасовому URL для подальшого завантаження.
2. Функція `loadFromJson` виконує обробку обраного користувачем JSON-файлу. Використовується об'єкт `FileReader`, який зчитує вміст файлу як текст, після чого виконується спроба парсингу у JavaScript-об'єкт за допомогою `JSON.parse()`. Отримані значення присвоюються відповідним реактивним змінним, що дозволяє миттєво відобразити завантажені дані у візуальному інтерфейсі. Передбачено елементарну обробку помилок – у випадку невдалого парсингу користувачу виводиться повідомлення про помилку.

1.3.7 Розробка представлення «Про діаграми»

Представлення "Про діаграми" призначене для інформування користувачів веб-застосунку про основні типи діаграм компетентностей ІТ-фахівця, які підтримуються системою. Воно має на меті надати стислий опис кожного типу діаграми, що допомагає користувачеві зрозуміти їх сутність та застосування для подальшого вибору та побудови відповідної візуалізації.

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

Представлення реалізує відображення структурованого списку, який містить п'ять основних типів діаграм: I-shaped, T-shaped, V-shaped, P-shaped та комбіновані (combi-shaped). Для кожного типу виводиться заголовок та короткий опис, що береться з файлу локалізації, що забезпечує багатомовність інтерфейсу. Таким чином, користувач отримує зрозумілу інформацію про специфіку кожного виду діаграми компетентностей.

Основний код представлення:

```

<script setup>
import { useI18n } from 'vue-i18n'
const { t } = useI18n()
</script>

<template>
  <div class="container mt-4">
    <h2 class="mb-3">{{ t('aboutDiagrams.title') }}</h2>
    <ul class="list-group">
      <li class="list-group-item">
        <strong>{{ t('aboutDiagrams.i_shaped.title') }}</strong> {{
t('aboutDiagrams.i_shaped.description') }}
      </li>
      <li class="list-group-item">
        <strong>{{ t('aboutDiagrams.t_shaped.title') }}</strong> {{
t('aboutDiagrams.t_shaped.description') }}
      </li>
      <li class="list-group-item">
        <strong>{{ t('aboutDiagrams.v_shaped.title') }}</strong> {{
t('aboutDiagrams.v_shaped.description') }}
      </li>
      <li class="list-group-item">
        <strong>{{ t('aboutDiagrams.p_shaped.title') }}</strong> {{
t('aboutDiagrams.p_shaped.description') }}
      </li>
      <li class="list-group-item">
        <strong>{{ t('aboutDiagrams.combi_shaped.title') }}</strong> {{
t('aboutDiagrams.combi_shaped.description') }}
      </li>
    </ul>
  </div>
</template>

```

Пояснення основного коду:

1. Даний фрагмент коду реалізовано з використанням Vue 3 у варіанті <script setup>, що забезпечує спрощене та чисте оголошення логіки компонента. Використання хука useI18n дозволяє отримувати функцію перекладу t, яка звертається до локалізованих текстових ресурсів.
2. У шаблоні (<template>) створено контейнер, що містить заголовок і перелік типів діаграм у вигляді списку (), де кожен елемент списку () містить назву типу діаграми у вигляді жирного тексту та її опис. Обидва текстові поля

					РП 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

отримуються динамічно через ключі локалізації, що забезпечує підтримку багатомовного інтерфейсу.

Завдяки такій структурі, представлення є легким для масштабування, легко адаптується до нових типів діаграм і забезпечує зручне та зрозуміле подання інформації користувачу.

1.3.8 Розробка представлення «Про застосунок»

Представлення "Про застосунок" призначене для інформування користувачів про мету, основний функціонал та особливості веб-застосунку побудови діаграми компетентностей ІТ-фахівця. Воно виконує роль вступного інформаційного блоку, який пояснює користувачу значення застосунку, його користь та сферу застосування. Назва представлення чітко відображає його зміст, забезпечуючи зручний доступ до ключової інформації про систему.

Основною функціональністю представлення є відображення локалізованого текстового контенту, який складається із заголовка, короткого опису (lead) та детального пояснення. Завдяки використанню інтернаціоналізації (i18n), представлення підтримує багатомовність, що забезпечує доступність інформації для користувачів із різними мовними уподобаннями. Представлення не містить інтерактивних елементів, його метою є інформування користувача про призначення веб-застосунку.

Основний код представлення:

```
<script setup>
import { useI18n } from 'vue-i18n'
const { t } = useI18n()
</script>

<template>
  <div class="container mt-4">
    <h2 class="mb-3">{{ t('aboutApp.title') }}</h2>
    <p class="lead">{{ t('aboutApp.lead') }}</p>
    <p>{{ t('aboutApp.text') }}</p>
  </div>
</template>
```

Пояснення основного коду

1. Дане представлення реалізовано за допомогою фреймворку Vue 3 із використанням Composition API та локалізації через бібліотеку vue-i18n. В

					РП 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

секції `<script setup>` здійснюється імпорт функції `useI18n`, яка надає метод `t` для отримання перекладів текстових ключів.

- У шаблоні (`<template>`) створюється контейнер, який містить структуровані елементи: заголовок (`<h2>`), текстовий блок із класом `lead` для короткого опису та основний текстовий абзац. Вміст цих елементів формується за допомогою функції `t` з ключами `aboutApp.title`, `aboutApp.lead` та `aboutApp.text`, що забезпечує динамічне завантаження локалізованих текстів відповідно до поточної мови інтерфейсу.

Представлення є простим, але водночас гнучким інтерфейсним компонентом, що виконує роль інформаційного блока з підтримкою багатомовності, важливої для забезпечення широкого кола користувачів веб-застосунку.

1.4 Тестування веб-застосунку

1.4.1 Вивантаження веб-застосунку на хостинг

Для забезпечення доступності веб-застосунку побудови діаграми компетентностей ІТ-фахівця в мережі Інтернет необхідно здійснити його публікацію на хостинговому сервісі. В якості платформи для розміщення застосунку було обрано сервіс Netlify, який надає зручні інструменти для розгортання статичних веб-сайтів із підтримкою сучасних фронтенд-фреймворків, зокрема Vue.js. Особливістю Netlify є можливість швидкого деплою проекту за допомогою інтерфейсу drag and drop, що значно спрощує процес розгортання для розробників, зокрема початківців.

Перед розгортанням на хостинг необхідно здійснити збірку Vue-проекту з метою отримання статичних файлів, які будуть розміщені на сервері. Збірка виконується за допомогою командного рядка у середовищі розробки:

```
npm run build
```

Під час виконання команди відбувається компіляція вихідного коду застосунку та формування оптимізованої версії, яка зберігається у папці `dist` (за замовчуванням). Усі файли, необхідні для запуску веб-застосунку у браузері, знаходяться саме у цій директорії.

					РП 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

Для розгортання застосунку на Netlify використовувався метод drag and drop через веб-інтерфейс платформи. Цей спосіб є інтуїтивно зрозумілим і не вимагає додаткових налаштувань CI/CD:

1. Авторизація у особистому кабінеті Netlify через веб-браузер.
2. Перехід у розділ для створення нового сайту.
3. Вибір опції розміщення файлів вручну.
4. Завантаження папки dist, що містить результати збірки Vue-проекту, методом перетягування у відповідну область сторінки.
5. Після завершення завантаження система автоматично розгортає застосунок і надає унікальну URL-адресу для доступу.

Обраний метод розгортання має ряд переваг:

1. Швидкість і простота: відсутність необхідності конфігурувати складні налаштування серверів чи систем безперервної інтеграції.
2. Автоматичне масштабування: Netlify забезпечує стабільну роботу сайту навіть при збільшенні навантаження.
3. Безкоштовний тарифний план: дозволяє розміщувати невеликі проекти без додаткових витрат.
4. Підтримка HTTPS: автоматична генерація SSL-сертифікатів забезпечує безпечне з'єднання.

Використання платформи Netlify у поєднанні з Vue-проектом дозволило швидко та ефективно здійснити вивантаження веб-застосунку побудови діаграми компетентностей IT-фахівця. Метод drag and drop значно спрощує процес розгортання, що робить цю технологію оптимальним вибором для проектів середнього та малого масштабу, які не потребують складної інфраструктури.

На рисунку 1.11 зображено застосунок на хостингу Netify в live-режимі.

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

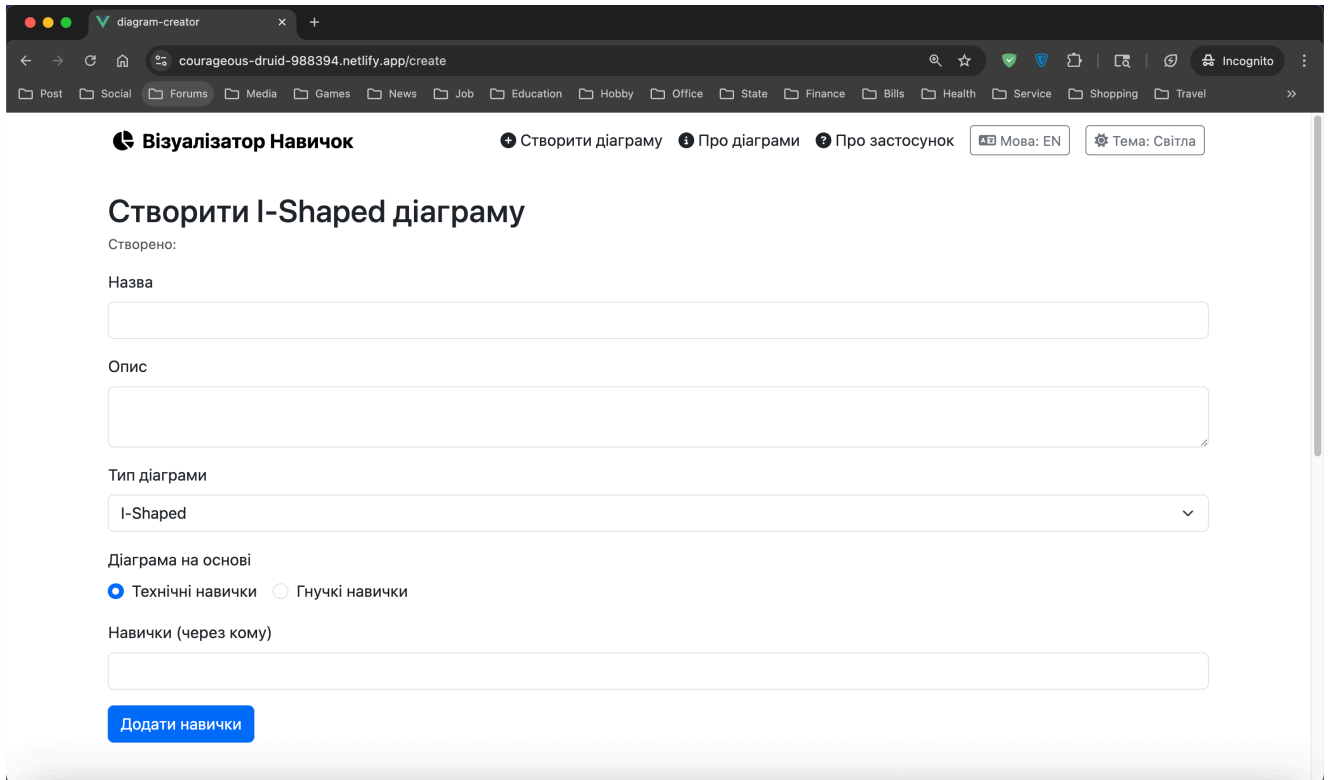


Рисунок 1.10. Застосунок на хостингу Netify в live-режимі

1.4.1 Обґрунтування обраного виду тестування

В процесі розробки веб-застосунку побудови діаграми компетентностей ІТ-фахівця важливим етапом є проведення тестування, яке дозволяє забезпечити відповідність функціональних можливостей системи вимогам та запобігти появі помилок у кінцевому продукті. З огляду на специфіку проєкту та характер поставлених завдань, було прийнято рішення використати ручне тестування як основний вид контролю якості програмного забезпечення.

Ручне тестування передбачає безпосередню участь тестувальника у перевірці функціоналу застосунку шляхом виконання серії запланованих тестових сценаріїв без застосування автоматизованих інструментів. Обґрунтування вибору саме ручного тестування ґрунтується на кількох ключових факторах.

По-перше, застосунок є новоствореним продуктом із відносно невеликою кількістю модулів та обмеженою функціональністю, що дозволяє ефективно і всебічно дослідити поведінку системи вручну. Ручне тестування забезпечує гнучкість у процесі оцінки користувацького інтерфейсу, логіки взаємодії компонентів, а також можливість виявлення нетипових помилок, які можуть бути

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

пропущені при автоматизованому підході.

По-друге, на даному етапі розробки відсутня стабільна та масштабна база для створення комплексних автоматичних тестів, що робить автоматизацію менш доцільною через додаткові витрати часу і ресурсів. Ручне тестування дозволяє оперативно адаптувати тестові сценарії до змін функціоналу та швидко реагувати на виявлені недоліки.

По-третє, ручне тестування сприяє глибшому розумінню розробниками та тестувальниками особливостей застосунку, що важливо для подальшого вдосконалення системи і формування більш ефективних стратегій тестування в майбутньому.

Враховуючи масштаб проекту, наявні ресурси та необхідність детальної перевірки функціональних аспектів веб-застосунку побудови діаграми компетентностей ІТ-фахівця, ручне тестування є найбільш раціональним та ефективним вибором для забезпечення якості програмного продукту.

1.4.1 Тестування зручності веб-застосунку

З метою перевірки коректності роботи навігаційної системи, мобільного відображення, зміни тем оформлення та мовної локалізації у веб-застосунку для побудови діаграми компетентностей ІТ-фахівця було проведено ручне тестування.

Під час тестування враховувалися наступні ключові чинники:

1. Функціональність навігації: перевірка правильності маршрутів між сторінками та коректності переходів при взаємодії з елементами інтерфейсу.
2. Адаптивність інтерфейсу: тестування вигляду й поведінки навігаційної панелі та інших елементів на мобільних пристроях і в різних розмірах екранів.
3. Зміна теми оформлення: перевірка стабільності перемикання між світлою та темною темами без порушення візуальної структури чи функціональності.
4. Мовна локалізація: тестування перемикання між українською та англійською мовами, включно з перевіркою відповідності перекладів і динамічного оновлення інтерфейсу.

Результати тестування:

					РП 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

1. Навігаційна система працює стабільно та інтуїтивно зрозуміло: всі основні посилання (головна сторінка, створення діаграми, перегляд діаграм, інформація про застосунок) ведуть на відповідні компоненти без затримок або збоїв.
2. Мобільна версія інтерфейсу зберігає повну функціональність і читабельність, панель навігації адаптується до вузьких екранів, усі елементи доступні для взаємодії.
3. Перемикання тем оформлення здійснюється миттєво без потреби перезавантаження сторінки; темна тема коректно застосовується до всіх елементів інтерфейсу.
4. Мовне перемикання між українською та англійською працює належним чином: текстові елементи оновлюються автоматично, переклади відповідають змісту, незалежно від обраної теми чи сторінки.

На рисунку 1.11 зображено головну сторінку.

Рисунок 1.11. Головна сторінка застосунку

На рисунку 1.12 зображено сторінку «Про діаграми».

					РП 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

Типи діаграм

I-подібний: Глибока експертиза в одній області.

T-подібний: Одна глибока навичка з широкими загальними навичками.

V-подібний: Глибокі аналітичні та методологічні навички.

P-подібний: Дві глибокі навички (наприклад, дизайн + технології).

Комбі-подібний: Універсальний в багатьох сферах.

Рисунок 1.12. Сторінка «Про діаграми»

На рисунку 1.13 зображено сторінку «Про застосунок».

Про застосунок

Цей застосунок допомагає візуалізувати ваші навички у різних професійних формах профілю (I, T, V, P, Combi).

Налаштуйте рівень навичок за допомогою повзунків і завантажте діаграму для використання в портфоліо або особистому сайті.

Рисунок 1.13. Сторінка «Застосунок»

На рисунку 1.14 зображено головну сторінку у мобільній версії.

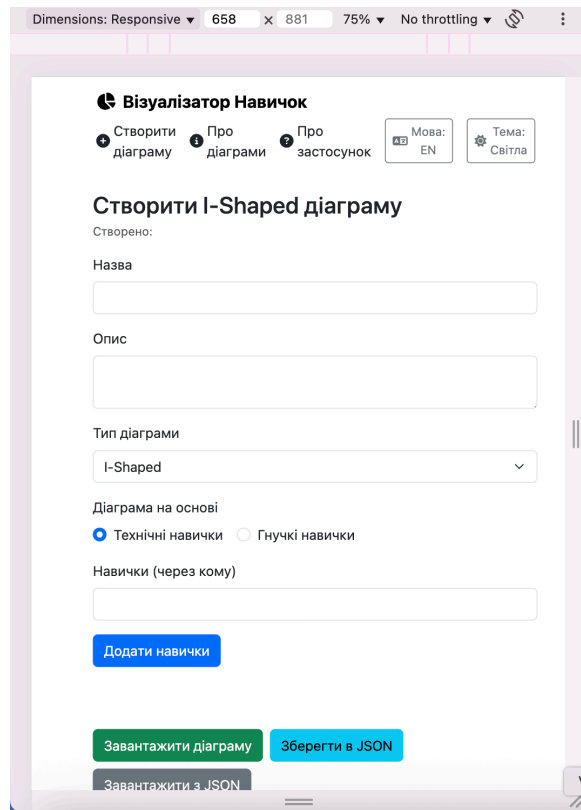


Рисунок 1.14. Головна сторінка у мобільній версії.

На рисунку 1.15 зображено темну тему застосунку.

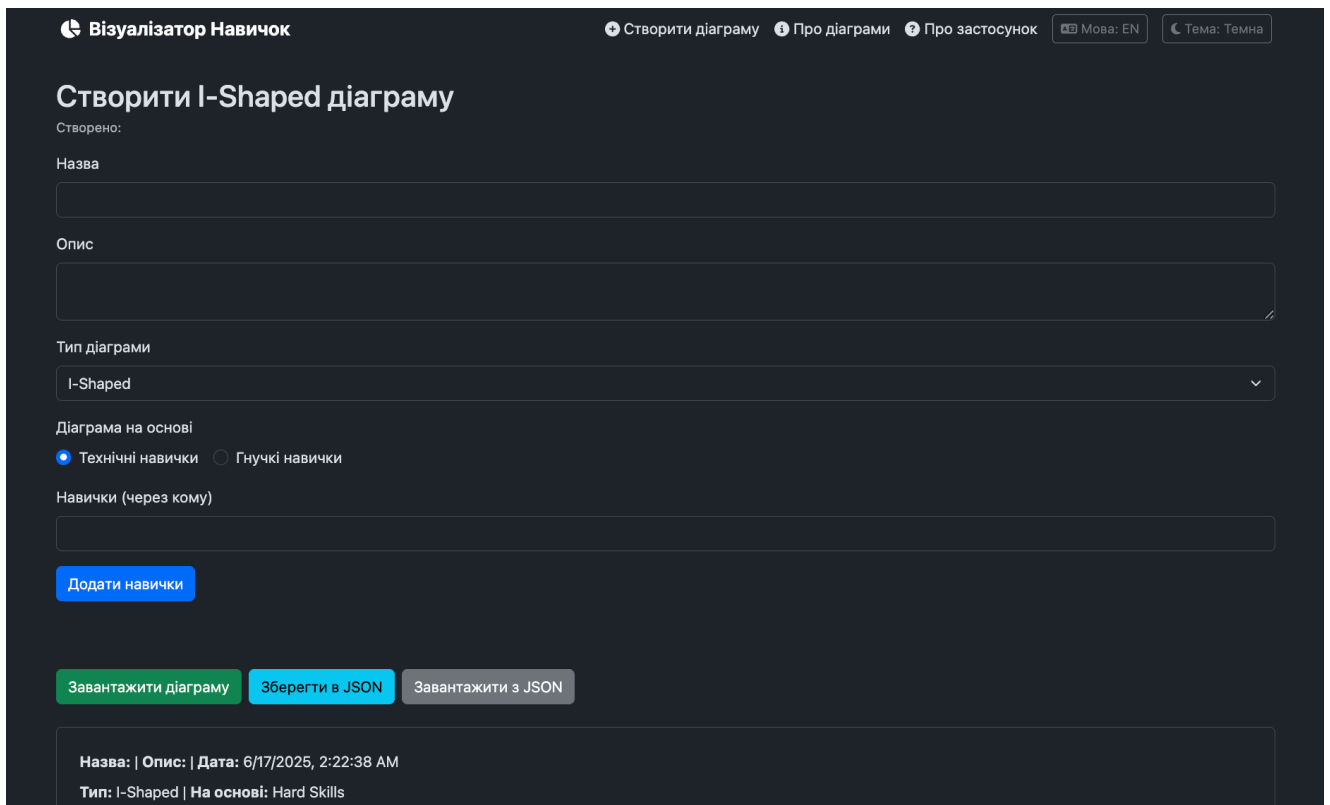


Рисунок 1.15. Темна тема застосунку

На рисунку 1.16 зображено англійську мову застосунку.

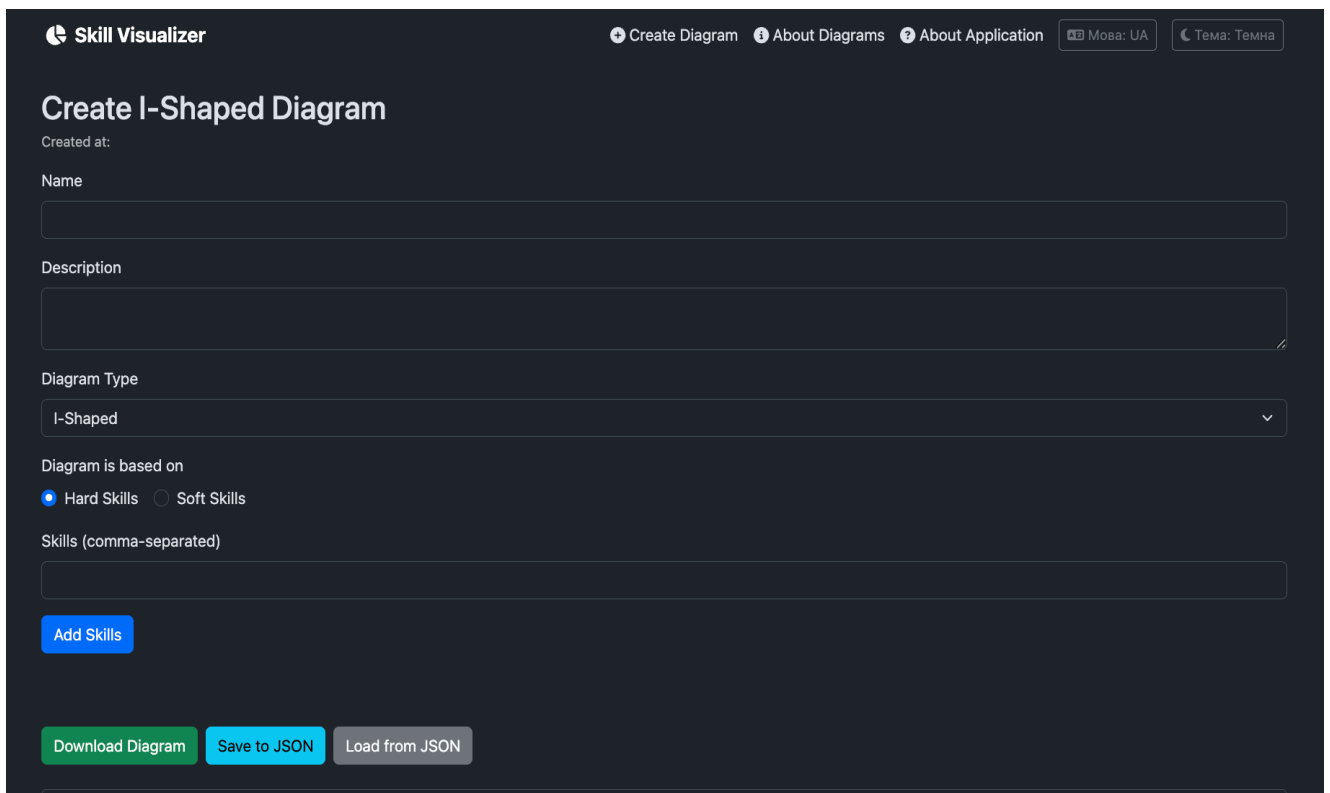


Рисунок 1.16. Англійська мова застосунку

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

У результаті тестування встановлено, що інтерфейс застосунку забезпечує комфортну взаємодію користувача з системою та відповідає сучасним вимогам до багатомовних адаптивних веб-застосунків.

1.4.2 Тестування основного функціоналу

Тестування модуля створення діаграм у веб-застосунку здійснювалося з метою перевірки працездатності ключових функцій, що забезпечують побудову діаграм компетентностей ІТ-фахівця. Було зосереджено увагу на перевірці логіки введення, обробки та візуалізації даних відповідно до обраного типу діаграми.

Основні чинники, які перевірялися під час тестування:

1. правильність вибору типу діаграми та відповідність логіки її побудови (I-, T-, V-, P-, Combi-Shaped);
2. коректне введення та обробка списку навичок (Hard Skills або Soft Skills);
3. динамічне додавання, редагування та видалення навичок із заданими рівнями та кольорами;
4. підтримка специфічної структури V-Shaped-діаграм із розділенням на ліву та праву частини;
5. функціональність збереження діаграми у форматі JSON та її подальшого відновлення;
6. коректна генерація зображення діаграми (PNG) за допомогою HTML-канвасу.

У результаті тестування було отримано такі основні висновки:

1. усі типи діаграм створюються відповідно до заданих параметрів;
2. навички додаються, редагуються та візуалізуються без помилок;
3. введення кольору та рівня компетентності працює стабільно;
4. формат JSON коректно зберігає повну структуру діаграми;
5. відновлення стану з JSON-файлу відбувається без втрати даних;
6. експорт до PNG забезпечує точне графічне представлення створеної діаграми.

На рисунку 1.17 зображено приклад створення діаграми (T-Shaped) для технічних навичок (Hard Skills).

					<i>РП 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

Programming ✕

Level: 89%

Color

Math ✕

Level: 70%

Color

Algorithms ✕

Level: 50%

Color

Data Structures ✕

Level: 50%

Color

Download Diagram
Save to JSON
Load from JSON

Name: Dima | **Description:** Dima's diagram | **Date:** 6/17/2025, 2:22:38 AM

Type: T-Shaped | **Based on:** Hard Skills

Skill	Level (%)
Programming	89%
Math	70%
Data Structures	50%
Algorithms	50%

Функціональність створення діаграм реалізовано успішно, що підтверджується стабільною роботою компонентів та відповідністю очікуваній логіці побудови профілю компетентностей.

2 ЕКОНОМІЧНИЙ РОЗДІЛ

2.1 Резюме

У дипломному проєкті розроблено веб-застосунок для побудови діаграми компетентностей ІТ-фахівця, який дозволяє візуалізувати навички, аналізувати рівень знань та відстежувати прогрес розвитку. Такий функціонал сприяє ефективному управлінню професійним розвитком та підвищенню кваліфікації спеціалістів. Якість системи оцінюється за критеріями зручності використання, продуктивності та практичної користі.

2.2 Визначення трудомісткості розробки ПЗ

Тривалість розробки програмного продукту залежить від його обсягу, складності, кваліфікації розробників і встановлених ринком термінів. Метод структурної аналогії дозволяє оцінити обсяг у тисячах умовних машинних команд на основі подібного програмного забезпечення.

Табл. 2.1 містить аналоги ПЗ з подібними функціями; обраний варіант виділено сірим.

Таблиця 2.1. Каталог аналогів

Найменування ПП	Обсяг функції ПП – V_0 , умовних. машинних командах
1. ПП автоматизації засобів по каталогу	680 – 7000
2. ПП автоматизованих розрахунків	1300 – 8600
3. ПП введення інформації	1060 – 5750

Після вибору аналога з обсягом V_0 (умовні машинні команди), трудомісткість визначається за табл. 2.2.

Таблиця 2.2. Обсяг ПП

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262
4.00	283
6.00	330
7.00	357

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
8.00	385
9.00	414
10.00	445

На основі отриманого значення за довідником визначають укрупнену норму часу, скориговану коефіцієнтом $K_k = 0,7-0,8$ для умов розробки на комп'ютері:

$$T_{ap} = 306 \times 0,8 = 244,8 \text{ (люд/годин)} \quad (2.1)$$

Трудомісткість визначається для кожного етапу окремо, з урахуванням складності, новизни та використання стандартних модулів, за відповідними формулами:

$$T_{T3} = T^a p \times L_1 \times K_H \quad (2.2)$$

$$T_{TII} = T^a p \times L_2 \times K_H \quad (2.3)$$

$$T_{PPI} = T^a p \times L_3 \times K_H \times K_T \quad (2.4)$$

Для розрахунку використовуються такі коефіцієнти:

- L_i – частка i -го етапу (табл. 2.3);
- K_H – коефіцієнт новизни (табл. 2.4);
- K_T – коефіцієнт використання типових програм (табл. 2.5).

Наш варіант виділено сірим.

Таблиця 2.3. Питомі коефіцієнти трудомісткості стадії у загальній трудомісткості розробки ПП

Код стадії	Ступінь новизни		
	А	Б	В
T3 (L_1)	0,15	0,12	0,12
TII (L_2)	0,16	0,15	0,11
PPI (L_3)	0,55	0,58	0,61

Таблиця 2.4. Значення коефіцієнта новизни

Код ступеня новизни	Ступінь новизни	Значення K_H
А	Принципово новий ПП	1,75 – 1,2
Б	ПП – розвиток визначеного параметричного ряду	1,0 – 0,8

Код ступеня новизни	Ступінь новизни	Значення K_n
В	ПП, що має аналог	0,7

Таблиця 2.5. Значення коефіцієнта використання типових програм

Ступінь охоплення реалізованих функцій розробленого ПП типовими програмами, %	Значення K_T
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Тепер розраховуємо трудомісткість для всіх етапів і зводимо у табл. 2.6:

Трудомісткість технічного завдання:

$$T_{tz} = Ta * L_1 * K_n = 244,8 * 0,12 * 0,8 = 23,50 \text{ (люд/годин)} \quad (2.2)$$

Трудомісткість розробки технічного проєкту:

$$T_{tp} = Ta * L_2 * K_n = 244,8 * 0,15 * 0,8 = 29,37 \text{ (люд/годин)} \quad (2.3)$$

Трудомісткість розробки робочого проєкту:

$$T_{rp} = Ta * L_3 * K_n * K_m = 244,8 * 0,58 * 0,8 * 0,7 = 79,51 \text{ (люд/годин)} \quad (2.4)$$

Для розрахунків визначили обсяг документації по етапах:

- технічне завдання $N_{tz}=2$ (стор);
- розробка ТП $N_{tp}=53$ (стор);
- розробка робочого проєкту $N_{rp}=9$ (стор);
- пояснювальна записка відповідно $N_{pz}=15$ (стор).

Таблиця 2.6. Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, годин		
1.ТЗ	$T_{РТЗ}=23,50$	$T_{кк}=0,7*N_{ТЗ}= 0,7*2=1,4$	$T_{нк}=0,15*N_{ТЗ}=0,15*2=0,30$
2.Розробка ТП	$T_{РТП}=29,37$	$T_{кк}=0,7*N_{ТП}=0,7*53=37,1$	$T_{нк}=0,15*N_{ТП}=0,15*53=7,95$
3.Розробка РП	$T_{Ррп}= 79,51$	$T_{кк}=0,7*N_{рп}=0,7*9=6,3$	$T_{нк}=0,15*N_{рп}=0,15*9=1,35$
4.Розробка ПЗ	$T_{Пз}=1,5*N_{Пз}=1,5*15=22,5$	$T_{кк}=0,7*N_{Тз}=0,7*15=10,5$	$T_{нк}=0,15*N_{Пз}=0,15*15=2,25$

Усього, в т.ч.:	213,53		
- на розробку	Тр=154,88		
- контроль керівника		Ткк=48,3	
- нормоконтроль			Тнк=10,35

2.3 Розрахунок ціни програмного продукту

Розраховуємо основну зарплату виконавців, матеріальні та загальні витрати на розробку ПП. Зарплата наведена в табл. 2.7. З 1 квітня 2024 мінімальна місячна зарплата – 8000 грн, погодинна ставка – 46 грн (згідно зі ст. 8 Закону про Держбюджет України).

Таблиця 2.7. Розрахунок основної заробітної плати виконавців

Найменування робіт	Трудомісткість робіт, години	Погодинна тарифна ставка, грн.	Розрахунок, грн.
1.Розробка ПП	154,88	55,35	8573,48
2.Контроль керівника	48,3	120,28	5810,00
3.Нормоконтроль	10,35	120,28	1245,00
Усього	-	-	З _о = 15628,48

Розраховуємо матеріальні витрати на розробку ПП та наведемо їх у табл. 2.8.

Таблиця 2.8. Розрахунок матеріальних витрат на розробку

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна одиниці, грн.	Вартість, грн.
Папір	Лист А4	70	4.0	280,0
Разом	-	-	-	$V_{M1} = 280,0$
Транспортно – заготівельні Витрати (10%)				$V_{тр_з} = 0.1 * V_{M1} = 0,1 * 280 = 28,00$
Усього				$V_M = V_{M1} + V_{тр_з} = 308,00$

За отриманими даними складена калькуляція планової собівартості ПП, наведена в табл. 2.9.

					РП 08. 04 002. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

Таблиця 2.9. Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	308,00	V_M (див. табл. 2.8)
2. Основна заробітна плата	15628,48	Z_o (див. табл. 2.7)
3. Додаткова заробітна плата	1562,84	$Z_d = 0.1 * Z_o = 15628,48 * 0,1$
4. Відрахування до єдиного фонду соціального внеску	3864,37	$V_{с.с.в.} = 0.22 * (Z_o + Z_d) = 0,22 * (15628,48 + 1562,84)$
5. Накладні витрати	6251,39	$V_{нак.} = 0.4 * Z_o = 0.4 * 15628,48$
6. Повна собівартість	27681,08	$C_{пов} = V_M + Z_o + Z_d + V_{с.с.в.} + V_{нак.} = 308,00 + 15628,48 + 1562,84 + 3864,37 + 6251,39$

Розмір прибутку розраховується за формулою:

$$П = (C_n * P) / 100 = (27681,08 * 10) / 100 = 2768,11 \text{ грн.} \quad (2.5)$$

Де p – плановий рівень рентабельності (10-15%).

Оптова ціна розраховується за формулою:

$$C_o = C_n + П = 27681,08 + 2768,11 = 30449,19 \text{ грн.} \quad (2.6)$$

За отриманими даними, ціна реалізації ПП за формулою становить:

$$C_p = C_o + ПДВ = 30449,19 + 30449,19 * 0.2 = 36539,03 \text{ грн.} \quad (2.7)$$

3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

3.1 Основні положення

Закон України "Про охорону праці" поширюється на всі підприємства та установи, незалежно від характеру їх діяльності. Він є ключовим нормативно-правовим актом, що регламентує основи охорони праці. У межах дипломного проекту розглядається питання створення веб-застосунку, тож об'єктом дослідження обирається робоче місце програміста.

3.2 Аналіз небезпечних та шкідливих виробничих чинників що мають вплив на користувача ПК

Професійна діяльність користувачів комп'ютерної техніки відбувається в конкретному виробничому середовищі, яке впливає на їхній фізіологічний стан. Найбільший вплив мають фізичні чинники, серед яких – електромагнітне випромінювання різних частот, електростатичні поля, шумове навантаження, мікрокліматичні параметри, недостатній рівень освітлення, а також інші світлотехнічні характеристики. Хімічні та, особливо, біологічні чинники мають набагато менше значення для користувачів комп'ютерів. Для забезпечення безпеки і максимальної ефективності праці програміста на його робочому місці необхідно створити відповідні умови.

3.3 Заходи з охорони праці для безпечної роботи програміста

Виробниче середовище – це простір, у якому людина виконує свою професійну діяльність, що включає предмети праці, засоби та інструменти виробництва, результати праці, а також умови, за яких здійснюється трудовий процес.

3.3.1 Вимоги до виробничих приміщень для експлуатації ПК

Об'ємно-планувальні параметри будівель і приміщень, призначених для роботи з персональними комп'ютерами, повинні відповідати вимогам ДСанПіН 3.3.2.007–98. Забороняється організація робочих місць із ПК у підвальних або

					<i>РП 08. 04 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

цокольних приміщеннях. Площа, що припадає на одне робоче місце, має бути не меншою за 6,0 м², а об'єм повітря – не менше ніж 20,0 м³.

Приміщення, де проводиться робота з ПК, повинні бути забезпечені як природним, так і штучним освітленням згідно з вимогами СНиП II-4-79. Природне освітлення має здійснюватись через віконні прорізи, орієнтовані головним чином на північ або північний схід, і забезпечувати коефіцієнт природної освітленості (КПО) не нижче 1,5 %.

У виробничих приміщеннях необхідно передбачити наявність шаф для зберігання документації, магнітних носіїв інформації, а також полицок, стелажів, тумб тощо з урахуванням встановлених норм площі. У приміщеннях із комп'ютерною технікою слід проводити щоденне вологе прибирання. Крім того, ці приміщення мають бути обладнані аптечками для надання першої медичної допомоги.

3.3.2 Нормалізація повітря робочої зони

Згідно з ДСанПіН 3.3.2.-007-98, робота з використанням відеодисплейних терміналів (ВДТ) електронно-обчислювальних машин (ЕОМ) та персональних електронно-обчислювальних машин (ПЕМ) за рівнем енерговитрат належить до категорії легких робіт (1а, 1б). У зв'язку з цим повинні бути дотримані відповідні санітарні вимоги.

У таблиці 3.1 представлено нормативні показники мікроклімату, які встановлені для приміщень, де здійснюється робота з ВДТ ЕОМ і ПЕМ.

Таблиця 3.1. Норми мікроклімату для приміщень з ВДТ ЕОМ та ПЕМ

Пора року	Категорія робіт	Температура повітря, °С, не більше	Відносна вологість повітря %	Швидкість руху повітря, м/с
Холодна	Легка-1а	22-24	40-60	0,1
	Легка-1б	21-23	40-60	0,1
Тепла	Легка-1а	23-25	40-60	0,1
	Легка-1б	22-24	40-60	0,1

Концентрації позитивних та негативних іонів у повітрі приміщень, де експлуатуються відеодисплейні термінали (ВДТ), повинні відповідати

встановленим санітарно-гігієнічним вимогам, визначеним у нормативному документі № 2152-80.

У таблиці 3.2 подано відповідні значення цих санітарно-гігієнічних норм згідно з вимогами документа № 2152-80.

Таблиця 3.2. Санітарно-гігієнічні нормам № 2152-80

Рівні	Число іонів в 1 см ³ повітря	Число іонів в 1 см ³ повітря
	n+	n-
Мінімально необхідні	400	600
Оптимальні	1500-3000	3000-5000
Максимально допустимі	50000	50000

Для підтримки в приміщенні нормального, що відповідає гігієнічним вимогам, складу повітря, видалення з нього шкідливих речовин використовують вентиляцію. При природній вентиляції (за допомогою вікон) повітря надходить у приміщення і видаляється внаслідок різниці температур. Але вона має низку недоліків. Тому дипломним проектом передбачено встановлення припливно-витяжної системи вентиляції, а також можливе застосування кондиціонерів.

3.3.3 Виробниче освітлення, шум

Освітлення в приміщенні може бути як природного, так і штучного походження. Природне світло надходить через віконні прорізи збоку, забезпечуючи бокове освітлення. Для штучного освітлення в приміщенні застосовуються люмінесцентні лампи, які мають низку важливих переваг порівняно з лампами розжарювання. Зокрема, їх спектральний склад світла наближений до природного, вони характеризуються вищою світловою віддачею та мають довший строк експлуатації. Рівень освітленості на робочих місцях має бути в межах 300–500 лк.

Щодо шумового впливу, то нормативний показник шуму на робочих місцях становить 50 дБ. Основними методами боротьби з шумом є усунення або зменшення його джерел на стадії проектування, використання звукопоглинальних матеріалів, а також раціональне розміщення елементів виробничого середовища.

3.3.4 Електробезпека

Сила електричного струму, що проходить через тіло людини, визначається рівнем прикладеної напруги та опором тієї ділянки тіла, до якої ця напруга прикладена. Джерелом живлення в більшості випадків є мережа змінного струму з напругою 220 В, на яку поширюється дія стандарту ГОСТ 25861-83.

Основними чинниками, що призводять до електротравматизму, є:

1. наявність напруги навіть на обладнанні, яке вважається відключеним;
2. раптова поява напруги через пошкодження ізоляції в місцях, де її за нормальних умов не повинно бути;
3. дотик струмопровідного обладнання до проводу, що знаходиться під напругою.

З метою запобігання ураженню електричним струмом необхідно суворо дотримуватись правил виконання робіт та технічної експлуатації. Потрібно виключити будь-який доступ оператора до частин обладнання, що знаходяться під небезпечною напругою, а також до неізольованих елементів, які призначені для роботи на низькій напрузі, але не підключені до системи захисного заземлення. Електроживлення персональних ЕОМ слід підводити через розетку за допомогою спеціальної вилки із заземлюючим контактом.

У приміщеннях, де одночасно експлуатуються п'ять і більше комп'ютерів, необхідно встановити службовий вимикач у доступному місці, що у разі аварійної ситуації дозволяє повністю відключити подачу електроенергії в кімнати.

3.3.5 Організація робочого місця

Робоче місце з відеодисплейним терміналом (ВДТ) повинне відповідати ергономічним вимогам, з урахуванням специфіки діяльності користувача (згідно з ДСанПіН 3.3.2.007-98). Конструкція меблів має передбачати індивідуальне регулювання для підтримання зручної робочої пози. Стіл рекомендується фарбувати матовою фарбою, а дисплей встановлювати на рівні очей, на відстані 60–90 см (оптимально – 70 см). Частота оновлення екрана повинна бути не менше 100 Гц.

Для зменшення втоми та покращення самопочуття необхідно впроваджувати

					<i>РП 08. 04 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

спеціальні вправи, передбачені санітарними нормами роботи з ВДТ..

На рисунку 3.1 наведено конструкцію робочого місця користувача ПК.

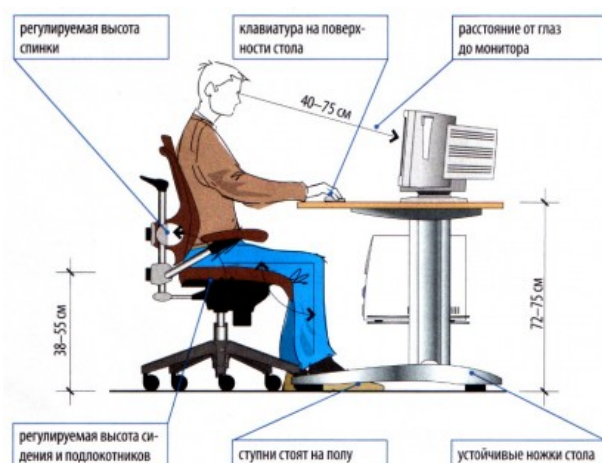


Рисунок 3.1. Конструкція робочого місця користувача ПК

3.4 Пожежна безпека

Згідно з ПУЕ, для пожежної безпеки потрібно ізолювати струмоведучі проводи, регулярно перевіряти їхню ізоляцію та дотримуватися норм протипожежного захисту на робочих місцях. Виконано вимоги СНиП щодо вогнестійкості будівель, евакуаційних шляхів і відстані робочих місць від виходів.

Приміщення обладнують вогнегасниками відповідно до правил пожежної безпеки України – у легкодоступних місцях, що не заважають евакуації. Для ПК використовують переносні вуглекислотні або порошкові вогнегасники об'ємом 3 л – по одному на 20 м². Після роботи ПК мають бути відключені від електромережі. Приміщення забезпечені запасними виходами з освітленими надписами «Запасний вихід», а план евакуації розміщений біля основного виходу..

3.5 Підведення підсумків

Безпечне та ергономічно облаштоване робоче місце програміста є необхідною умовою для збереження його здоров'я і підвищення ефективності роботи. Важливо дотримуватися норм мікроклімату, освітлення та електробезпеки, щоб уникнути професійних захворювань і травм. Регулярні перерви та спеціальні вправи допомагають зменшити втому і покращити самопочуття під час тривалого перебування за комп'ютером.

					<i>РП 08. 04 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

ВИСНОВКИ

Реалізація дипломного проєкту підтвердила важливість і необхідність створення спеціалізованого веб-застосунку для візуалізації компетентностей ІТ-фахівців, орієнтованого на потреби HR-відділів і рекрутерів. Розроблений інструмент дозволяє не лише систематизувати інформацію про професійні навички співробітників, а й здійснювати її наочне представлення у вигляді діаграм типу І-, Т-, π-, V- та комбінованих форм, що значно підвищує якість оцінки та порівняння кандидатів чи працівників.

Використання прогресивного JavaScript-фреймворку Vue.js забезпечило створення інтуїтивного і адаптивного інтерфейсу, що спрощує роботу користувача із застосунком. Застосування SVG-технології для генерації діаграм гарантує високу якість і масштабованість візуального контенту без втрати чіткості. Додатково, функції перемикання тем (світлої і темної) та підтримка двомовного інтерфейсу (українська та англійська мови) роблять застосунок зручним для широкого кола користувачів з різними уподобаннями та мовним середовищем.

Не менш важливим є впровадження можливості збереження діаграм у форматах PNG і JSON, що розширює функціонал системи та забезпечує легку інтеграцію з іншими кадровими та аналітичними інструментами. Це дозволяє не лише зберігати результати оцінки компетентностей, а й використовувати їх для подальшого аналізу, звітності або імпорту у зовнішні системи.

Розроблений веб-застосунок є комплексним рішенням, яке може суттєво підвищити ефективність кадрового менеджменту в ІТ-компаніях, сприяючи більш об'єктивній оцінці працівників і прийняттю стратегічних рішень щодо розвитку персоналу. Впровадження такого інструменту відповідає сучасним вимогам цифровізації бізнес-процесів і є важливим кроком у вдосконаленні управління людськими ресурсами.

					<i>РП 08. 04 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

ДОДАТОК А. Фрагмент коду VueJS компоненту створення

діаграм

```
<template>
  <div class="container my-4">
    <h2 class="mb-1">{{ t('create.title') }} {{ selectedShape }} {{ t('create.diagram') }}</h2>
    <small class="text-muted mb-3 d-block">{{ t('create.createdAt') }}: {{ datetime }}</small>

    <!-- Diagram form -->
    <form class="mb-4" @submit.prevent>
      <div class="mb-3">
        <label class="form-label">{{ t('create.name') }}</label>
        <input type="text" class="form-control" v-model="diagramName" />
      </div>

      <div class="mb-3">
        <label class="form-label">{{ t('create.description') }}</label>
        <textarea class="form-control" rows="2" v-model="diagramDescription" />
      </div>

      <div class="mb-3">
        <label class="form-label">{{ t('create.type') }}</label>
        <select class="form-select" v-model="selectedShape">
          <option>I-Shaped</option>
          <option>T-Shaped</option>
          <option>V-Shaped</option>
          <option>P-Shaped</option>
          <option>Combi-Shaped</option>
        </select>
      </div>

      <div class="mb-3">
        <label class="form-label">{{ t('create.basedOn') }}</label>
        <div>
          <div class="form-check form-check-inline">
            <input class="form-check-input" type="radio" value="Hard Skills" v-model="skillType" />
            <label class="form-check-label">{{ t('create.hardSkills') }}</label>
          </div>
          <div class="form-check form-check-inline">
            <input class="form-check-input" type="radio" value="Soft Skills" v-model="skillType" />
            <label class="form-check-label">{{ t('create.softSkills') }}</label>
          </div>
        </div>
      </div>

      <!-- Normal skills input for all except V-Shaped -->
      <div v-if="selectedShape !== 'V-Shaped'" class="mb-3">
        <label class="form-label">{{ t('create.skillsComma') }}</label>
        <input type="text" class="form-control" v-model="skillsInput" />
      </div>
      <button
        v-if="selectedShape !== 'V-Shaped'"
        type="button"
        class="btn btn-primary mb-4"
        @click="addSkillsFromInput"
      >
        {{ t('create.addSkills') }}
    </form>
  </div>
</template>
```

```

</button>

<!-- V-Shaped specific inputs -->
<div v-if="selectedShape === 'V-Shaped'">
  <div class="mb-3">
    <label class="form-label">{{ t('create.leftSkills') }}</label>
    <input type="text" class="form-control" v-model="leftSkillsInput" />
  </div>
  <div class="mb-3">
    <label class="form-label">{{ t('create.rightSkills') }}</label>
    <input type="text" class="form-control" v-model="rightSkillsInput" />
  </div>
  <button type="button" class="btn btn-secondary mb-4" @click="setVShapeSkills">
    {{ t('create.setVShape') }}
  </button>
</div>
</form>

<!-- Skill Cards for normal shapes -->
<div class="row" v-if="selectedShape !== 'V-Shaped'">
  <div class="col-md-4 mb-3" v-for="(skill, index) in skills" :key="index">
    <div class="card" :style="{ borderLeft: '6px solid ' + skill.color }">
      <div class="card-body">
        <div class="d-flex justify-content-between align-items-center">
          <h5 class="card-title">{{ skill.name }}</h5>
          <button class="btn-close" @click="removeSkill(index)"></button>
        </div>
        <div class="mb-2">
          <label class="form-label">{{ t('create.level') }}: {{ skill.level
}}%</label>
          <input type="range" class="form-range" min="0" max="100" v-
model="skill.level" />
        </div>
        <div class="mb-2">
          <label class="form-label">{{ t('create.color') }}</label>
          <input type="color" class="form-control form-control-color" v-
model="skill.color" />
        </div>
      </div>
    </div>
  </div>
</div>
</div>
</div>

<!-- Skill Cards for V-Shaped -->
<div class="row" v-if="selectedShape === 'V-Shaped'">
  <div
    class="col-md-4 mb-3"
    v-for="(skill, index) in vShapeSkills"
    :key="'v-' + index"
  >
    <div
      class="card"
      :style="{
        borderLeft: '6px solid ' + skill.color,
        backgroundColor: skill.side === 'left' ? '#e3f2fd' : '#fce4ec'
      }"
    >
      <div class="card-body">
        <div class="d-flex justify-content-between align-items-center">
          <h5 class="card-title">{{ skill.name }} ({{ skill.side }})</h5>
          <button class="btn-close" @click="removeVShapeSkill(index)"></button>
        </div>
      </div>
    </div>
  >

```

```

        <div class="mb-2">
          <label class="form-label">{{ t('create.level') }}: {{ skill.level
}}%</label>
          <input type="range" class="form-range" min="0" max="100" v-
model="skill.level" />
        </div>

        <div class="mb-2">
          <label class="form-label">{{ t('create.color') }}</label>
          <input type="color" class="form-control form-control-color" v-
model="skill.color" />
        </div>
      </div>
    </div>
    </div>
  </div>
  </div>

  <!-- Actions -->
  <div class="my-4 d-flex gap-2 flex-wrap">
    <button class="btn btn-success" @click="downloadDiagram">{{ t('create.download')
}}</button>
    <button class="btn btn-info" @click="saveToJSON">{{ t('create.save') }}</button>
    <button class="btn btn-secondary" @click="triggerLoadFile">{{ t('create.load')
}}</button>
    <input
      ref="fileInput"
      type="file"
      accept="application/json"
      @change="loadFromJSON"
      style="display:none"
    />
  </div>

  <!-- Diagram Canvas -->
  <DiagramCanvas
    :shape="selectedShape"
    :skills="selectedShape === 'V-Shaped' ? vShapeSkills : skills"
    :title="diagramName"
    :description="diagramDescription"
    :skillType="skillType"
    :dateTime="dateTime"
    ref="canvasRef"
  />
</div>
</template>

<script setup>
import { useI18n } from 'vue-i18n'

const { t } = useI18n()

import { ref } from 'vue'
import DiagramCanvas from '../components/DiagramCanvas.vue'
import html2canvas from 'html2canvas'

const diagramName = ref('')
const diagramDescription = ref('')
const selectedShape = ref('I-Shaped')
const skillType = ref('Hard Skills')

const skillsInput = ref('')
const skills = ref([])

```

```

const leftSkillsInput = ref('')
const rightSkillsInput = ref('')
const vShapeSkills = ref([])

const canvasRef = ref()
const fileInput = ref(null)

const dateTime = ref(getCurrentDateTime())

function getCurrentDateTime() {
  const now = new Date()
  return now.toLocaleString()
}

const getRandomColor = () => {
  const colors = ['#3498db', '#2ecc71', '#e67e22', '#9b59b6', '#1abc9c', '#e74c3c']
  return colors[Math.floor(Math.random() * colors.length)]
}

const addSkillsFromInput = () => {
  const names = skillsInput.value.split(',').map(s => s.trim()).filter(s => s !== '')
  names.forEach(name => {
    skills.value.push({
      name,
      level: 50,
      color: getRandomColor()
    })
  })
  skillsInput.value = ''
}

const removeSkill = (index) => {
  skills.value.splice(index, 1)
}

const setVShapeSkills = () => {
  const leftNames = leftSkillsInput.value.split(',').map(s => s.trim()).filter(s => s)
  const rightNames = rightSkillsInput.value.split(',').map(s => s.trim()).filter(s => s)
  const combined = []

  leftNames.forEach(name => {
    let skill = skills.value.find(s => s.name === name)
    if (!skill) {
      skill = { name, level: 50, color: getRandomColor() }
      skills.value.push(skill)
    }
    combined.push({ ...skill, side: 'left' })
  })

  rightNames.forEach(name => {
    let skill = skills.value.find(s => s.name === name)
    if (!skill) {
      skill = { name, level: 50, color: getRandomColor() }
      skills.value.push(skill)
    }
    combined.push({ ...skill, side: 'right' })
  })

  vShapeSkills.value = combined

  // Clear inputs after setting
  leftSkillsInput.value = ''
  rightSkillsInput.value = ''
}

```

```

const removeVShapeSkill = (index) => {
  vShapeSkills.value.splice(index, 1)
}

const downloadDiagram = async () => {
  const el = canvasRef.value.$el
  const canvasImage = await html2canvas(el)
  const link = document.createElement('a')
  link.download = `${diagramName.value || selectedShape.value}-Diagram.png`
  link.href = canvasImage.toDataURL()
  link.click()
}

const saveToJSON = () => {
  // Update datetime on save
  dateTime.value = getCurrentDateTime()

  const data = {
    diagramName: diagramName.value,
    diagramDescription: diagramDescription.value,
    selectedShape: selectedShape.value,
    skillType: skillType.value,
    dateTime: dateTime.value,
    skills: skills.value,
    vShapeSkills: vShapeSkills.value,
  }
  const jsonStr = JSON.stringify(data, null, 2)
  const blob = new Blob([jsonStr], { type: 'application/json' })
  const url = URL.createObjectURL(blob)

  const a = document.createElement('a')
  a.href = url
  a.download = `${diagramName.value || selectedShape.value}-Diagram.json`
  a.click()
  URL.revokeObjectURL(url)
}

const triggerLoadFile = () => {
  fileInput.value?.click()
}

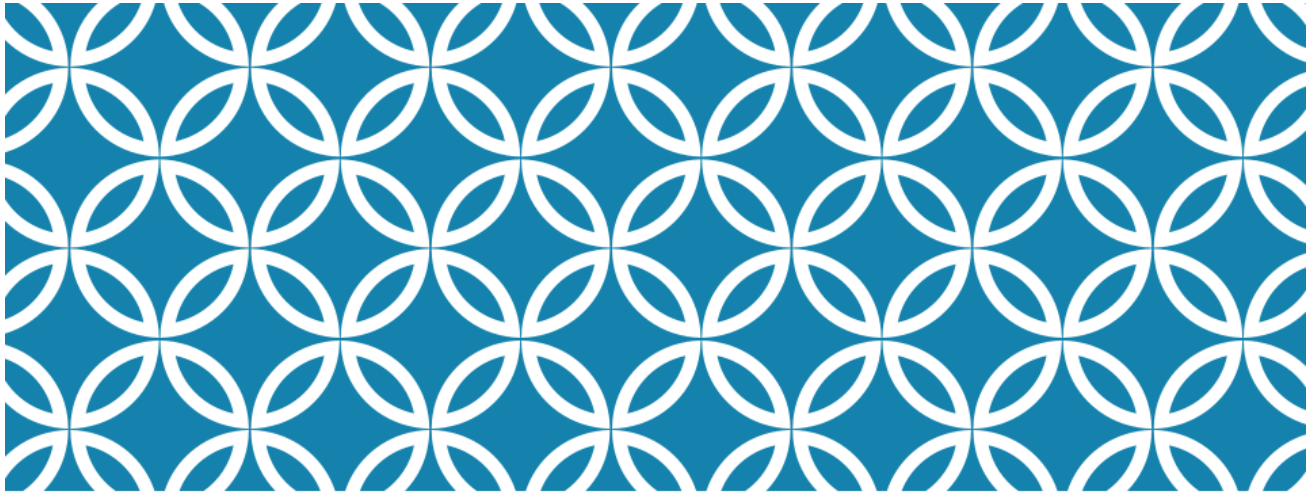
const loadFromJSON = (event) => {
  const file = event.target.files[0]
  if (!file) return

  const reader = new FileReader()
  reader.onload = (e) => {
    try {
      const data = JSON.parse(e.target.result)
      diagramName.value = data.diagramName || ''
      diagramDescription.value = data.diagramDescription || ''
      selectedShape.value = data.selectedShape || 'I-Shaped'
      skillType.value = data.skillType || 'Hard Skills'
      dateTime.value = data.dateTime || getCurrentDateTime()
      skills.value = Array.isArray(data.skills) ? data.skills : []
      vShapeSkills.value = Array.isArray(data.vShapeSkills) ? data.vShapeSkills : []
    } catch (err) {
      alert('Failed to load diagram: Invalid JSON')
      console.error(err)
    }
  }
}

reader.readAsText(file)

```

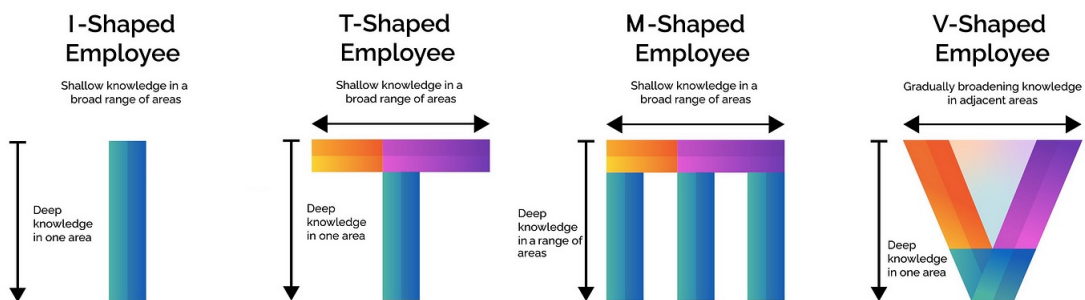
ДОДАТОК Б. Слайди мультимедійної презентації



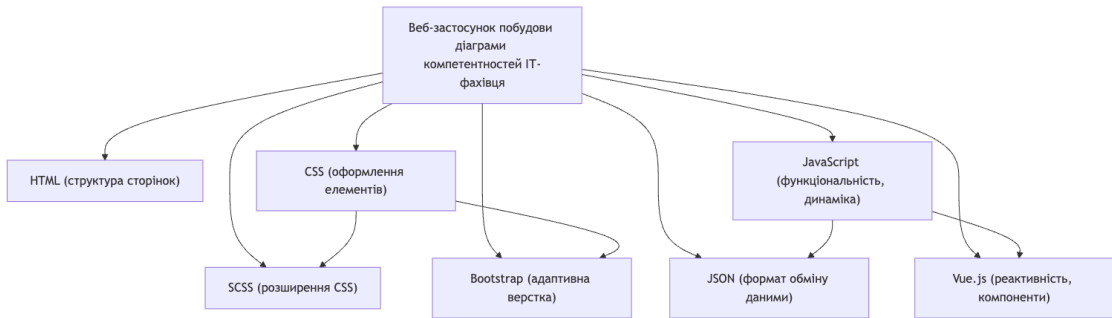
Розробка веб-застосунку для побудови діаграми компетентностей ІТ-фахівця

Виконав: студент групи 4РП-08 Василевський Д.К.
Керівник: викладач коледжу Жадан А.С.

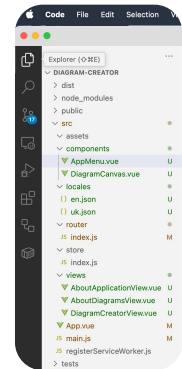
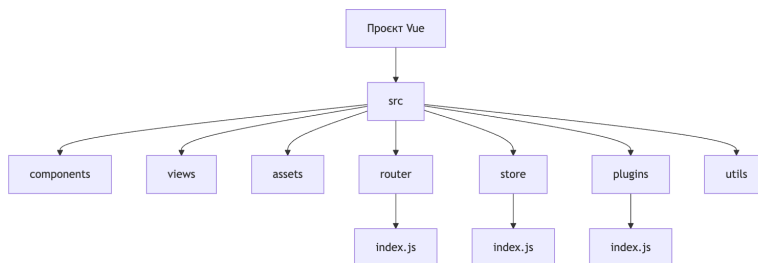
МЕТРИКИ КОМПЕТЕНЦІЙ



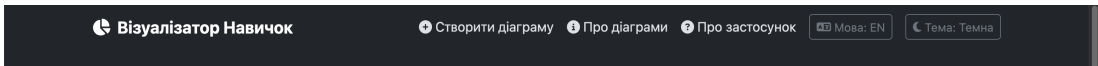
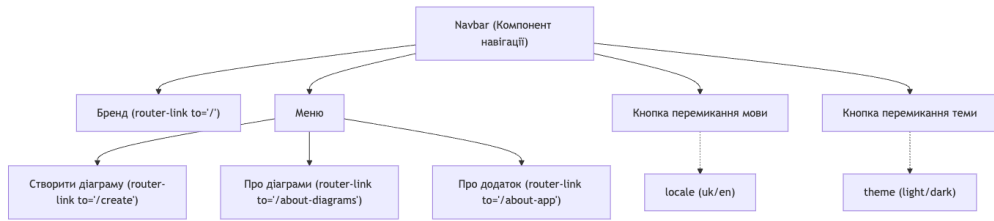
ВИКОРИСТАНІ ТЕХНОЛОГІЇ ДЛЯ РОЗРОБКИ ВЕБ-ЗАСТОСУНКУ



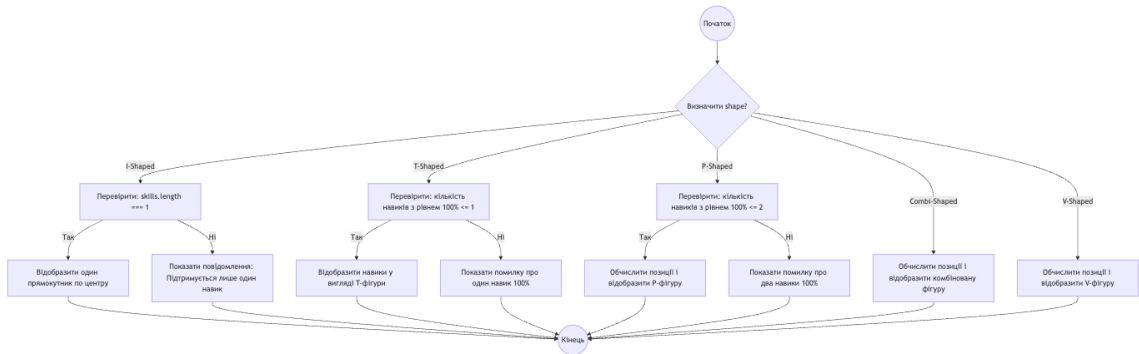
ФАЙЛОВА АРХІТЕКТУРА ВЕБ-ЗАСТОСУНКУ



НАВІГАЦІЯ У ВЕБ-ЗАСТОСУНКУ



БЛОК СХЕМА АЛГОРИТМУ ГЕНЕРАЦІЇ ДІАГРАМИ



КОД ГЕНЕРАЦІЇ ДІАГРАМИ В SVG

```
DiagramCanvas.vue |  
src > components > DiagramCanvas.vue > {} template > div.canvas-box-p-4.border.rounded > svg > g > template >  
7 <template>  
8 <div class="canvas-box p-4 border rounded">  
19 <svg :width="svgWidth" :height="svgHeight" style="background:#fff;">  
20 <g>  
46 <!-- T-Shaped -->  
47 <template v-else-if="shape === 'T-Shaped'">  
48 <template v-if="shapeHundredPercentCount <= 1">  
49 <g v-for="(skill, i) in sortedSkills" :key="i">  
50 <rect  
51 :x="computeTShapeX(i)"  
52 :y="baseTopY"  
53 :width="barWidth"  
54 :height="skill.level * scaleFactor"  
55 :fill="skill.color"  
56 />  
57 <text  
58 :x="computeTShapeX(i) + barWidth / 2"  
59 :y="baseTopY + skill.level * scaleFactor + 20"  
60 font-size="14"  
61 text-anchor="middle"  
62 fill="#000"  
63 >  
64 {{ skill.name }} ({{ skill.level }}%)  
65 </text>
```

ТЕСТУВАННЯ СТВОРЕННЯ ДІАГРАМИ

Skill Visualizer | Create Diagram | About Diagrams | About Application | EN | UA | TR | CL

Create I-Shaped Diagram

Created at: _____

Name: _____

Description: _____

Diagram Type: I-Shaped

Diagram is based on: Hard Skills Soft Skills

Skills (comma-separated): _____

[Add Skills](#)

[Download Diagram](#) [Save to JSON](#) [Load from JSON](#)

Name: | Description: | Date: 6/18/2025, 4:19:46 PM
Type: I-Shaped | Based on: Hard Skills

Programming Level: 88% | Color: Purple

Math Level: 70% | Color: Green

Algorithms Level: 50% | Color: Red

Data Structures Level: 50% | Color: Green

[Download Diagram](#) [Save to JSON](#) [Load from JSON](#)

Name: | Description: | Date: 6/18/2025, 4:19:46 PM
Type: I-Shaped | Based on: Hard Skills

ТЕСТУВАННЯ ЗМІНИ МОВИ

Skill Visualizer

Create Diagram About Diagrams About Application

Create I-Shaped Diagram

Created at:

Name

Description

Diagram Type

Diagram is based on

Skills (comma-separated)

Add Skills

Download Diagram Save to JSON Load from JSON

Name: | Description: | Date: 6/18/2025, 4:19:46 PM
Type: I-Shaped | Based on: Hard Skills

Створити I-Shaped діаграму

Сторінка:

Назва

Опис

Тип діаграми

Діаграма на основі

Навички (через коми)

Додати навички

Завантажити діаграму Зберегти в JSON Завантажити з JSON

Назва: | Опис: | Дата: 6/18/2025, 4:19:46 PM
Тип: I-Shaped | На основі: Hard Skills
I-Shaped діаграма підтримує лише одну навичку

ТЕСТУВАННЯ ЗМІНИ ТЕМИ

Створити I-Shaped діаграму

Сторінка:

Назва

Опис

Тип діаграми

Діаграма на основі

Навички (через коми)

Додати навички

Завантажити діаграму Зберегти в JSON Завантажити з JSON

Назва: | Опис: | Дата: 6/18/2025, 4:19:46 PM
Тип: I-Shaped | На основі: Hard Skills
I-Shaped діаграма підтримує лише одну навичку

Створити I-Shaped діаграму

Сторінка:

Назва

Опис

Тип діаграми

Діаграма на основі

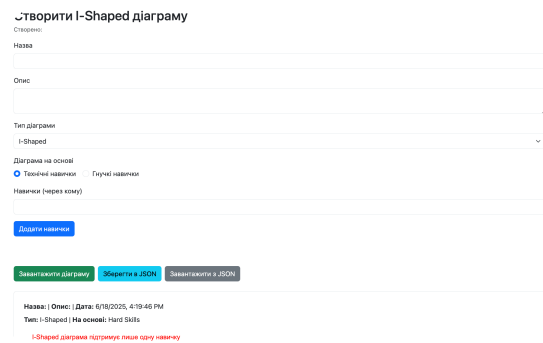
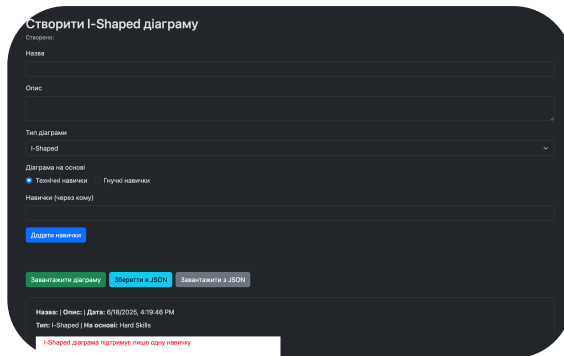
Навички (через коми)

Додати навички

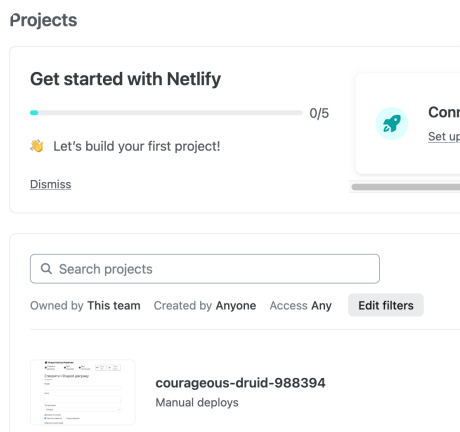
Завантажити діаграму Зберегти в JSON Завантажити з JSON

Назва: | Опис: | Дата: 6/18/2025, 4:19:46 PM
Тип: I-Shaped | На основі: Hard Skills
I-Shaped діаграма підтримує лише одну навичку

ТЕСТУВАННЯ ЗМІНИ ТЕМИ



РОБОТА З ХОСТИНГОМ



РЕЦЕНЗІЯ

на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Василевського Дмитра Костянтиновича

(прізвище, ім'я та по батькові)

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Розробка програмного забезпечення»

Керівник дипломного проекту (роботи) Закроєв Юрій Михайлович

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Розробка веб-застосунку для побудови діаграми компетентностей ІТ-фахівця

Обсяг розрахунково-пояснювальної записки 74 сторінок

Обсяг графічної (презентаційної) частини 12 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту завданню

Представлений на рецензію дипломний проект відповідає затвердженій темі та виконаний відповідно технічному завданню. Дипломний проект присвячений веб-розробці та складається з пояснювальної записки, додатку з програмним кодом та мультимедійної презентації, що містить приклади роботи програми.

б) характеристика виконання кожного розділу дипломного проекту

Пояснювальна записка складається з основного розділу (аналізу предметної області, проектування застосунку, реалізації застосунку, тестування застосунку), економічного розділу, розділу охорони праці та додатків. Перелічені розділи поетапно охоплюють розробку, виконані докладно та обґрунтовано. Розділ охорони праці містить загальну інформацію та вимоги до техніки безпеки оператора КТ. Економічний розділ проекту містить розрахунок витрат на НДР та реалізацію проекту.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту

Графічна частина складається з 12 слайдів мультимедійної презентації, виконаної у програмному продукті MS PowerPoint, які містять ілюстративні схеми, скріншоти роботи програмного застосунку, передбачені технічним завданням. Пояснювальна записка виконана акуратно та у відповідності до норм. Якість виконання графічної частини проекту та пояснювальної записки добра, розробку виконано у повному обсязі.

г) перелік позитивних якостей дипломного проекту Проект реалізовано як SPA із CSR, що відповідає актуальним підходам у фронтенд-розробці. Добре структурована архітектура застосунку.

Продумано дизайн: мінімалізм, flat design, адаптивність, підтримка темної/світлої теми та двомовності.

д) основні недоліки дипломного проекту _____

У коді не згадується використання типізації (наприклад, TypeScript), що є сучасною практикою для масштабування фронт-енд проєктів.

Деякі недоліки оформлення пояснювальної записки.

Оцінка розрахункової частини _____ Відмінно

Оцінка графічної частини _____ Добре

Загальна оцінка _____ Відмінно

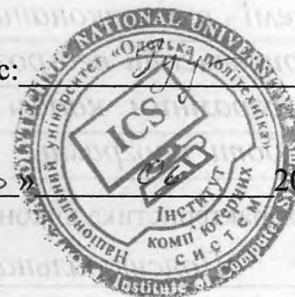
Прізвище, ім'я, по батькові рецензента _____ к.т.н. Рудніченко Микола Дмитрович

Місце роботи і посада рецензента Національний університет «Одеська політехніка», доцент кафедри інформаційних технологій

Підпис: _____

« 23 »

2025 р.



ВІДГУК

керівника на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Василевського Дмитра Костянтиновича

(прізвище, ім'я та по батькові)

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Тема дипломного проекту: Розробка веб-застосунку для побудови діаграми компетентностей ІТ-фахівця.

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) Дипломний проект виконано відповідно технічному завданню. Пояснювальна записка до дипломного проекту містить 74 сторінки. У пояснювальній записці описано етапи розробки веб-застосунку для побудови діаграми компетентностей ІТ-фахівця засобами HTML, CSS, JavaScript та Vue.js. Графічна частина складається з окремих 12 слайдів, оформлених у вигляді презентації, передбачених технічним завданням. Якість виконання пояснювальної записки та слайдів добра.

б) самостійність роботи над проектом: Протягом виконання дипломного проекту здобувач освіти поступово та послідовно виконував всі етапи, проявляла ініціативу в створенні загальної концепції та реалізації роботи. Всі роботи здобувач освіти виконував самостійно, з оглядом на рекомендації керівника.

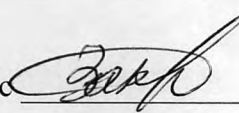
в) теоретична підготовка випускника (випускниці): Здобувач освіти під час роботи над дипломним проектом вивчив достатньо багато літературних та інтернет-джерел за даною тематикою. Вважаю, що теоретична підготовка дипломника достатня і він готовий до захисту проекту.

г) вміння розв'язувати виробничі та конструкторські питання Під час виконання дипломного проекту здобувач освіти показав вміння організовано працювати над поставленим завданням, застосовувати знання у галузі програмування та математики, розробляти, встановлювати та налаштовувати спеціалізоване програмне забезпечення, оформлювати слайди та складати презентації, користуючись сучасними комп'ютерними програмними засобами, такими як MS VS Code, HTML, CSS, JavaScript та Vue.js.

Оцінка розрахункової частини Добре
Оцінка графічної частини Відмінно
Загальна оцінка Відмінно

Прізвище, ім'я, по батькові керівника дипломного проекту Закроєв Юрій Михайлович

Місце роботи і посада керівника дипломного проекту Директор ТОВ «Біг ВОШ»

Підпис 

«20» 06 2025 р.

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
(ДИПЛОМНОГО ПРОЕКТУ)
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Василевський Дмитро Костянтинович,
здобувач освіти гр. 4РП-08, та

Закроєв Юрій Михайлович,
керівник дипломного проекту,

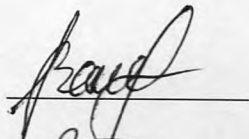
не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до дипломного проекту фахового молодшого бакалавра на тему:

«Розробка веб-застосунку для побудови діаграми компетентностей IT-фахівця» (автор роботи – Василевський Д.К., керівник роботи – Закроєв Ю.М.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2025 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

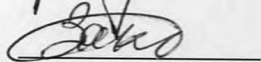
Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи і даємо згоду на обробку персональних даних.

Виконавець



/ Василевський Д.К. /

Керівник



/ Закроєв Ю.М. /

«16» червня 2025 р.

Д О В І Д К А

циклової комісії КТ та ПІ
про допуск до захисту дипломного проекту
здобувача (здобувачки) освіти ІV курсу
відділення комп'ютерних систем групи 4РП-08

Василевського Дмитра Костянтиновича

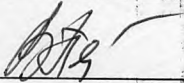
на тему Розробка веб-застосунку побудови діаграми
компетентностей ІТ-фахівця

Висновок відповідальної особи за проведення нормоконтролю:

пояснювальна записка до дипломного проекту виконана з некритичними

порушеннями ДСТУ та оформлена відповідно до вимог Положення про

дипломне проектування



(підпис)

20.06.2025

(дата)

Петрашова В.І.

(П.І.Б.)

Висновок відповідальної особи за перевірку роботи на наявність академічного
плагиату згідно звіту про перевірку від 18.06.2025 р. значення коефіцієнту
подібності в роботі становить 16,16%, коефіцієнт цитування – 5,45%.



(підпис)

20.06.2025

(дата)

Краснокутська К.Г.

(П.І.Б.)

Попередня експертиза (малий захист) дипломного проекту

здобувача (здобувачки) освіти

Василевського Д.К.

(П.І.Б.)

проведена « 20 » червня 2025 р.

Висновки Пояснювальна записка до дипломного проекту виконана у повному
обсязі. Випускна кваліфікаційна робота (дипломний проект) відповідає
вимогам Положення про дипломне проектування та рекомендована до
захисту.

Голова ЦК КТ та ПІ


(підпис)

Кривченко Ю.В.

(П.І.Б.)

Звіт подібності

метадані

Назва організації

Odesa Technical Professional College of Odesa National University of Technology

Заголовок

Розробка веб-застосунку побудови діаграми компетентностей IT-фахівця

Автор

Науковий керівник / Експерт

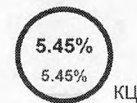
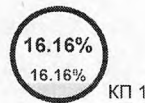
Василевський Дмитро Костянтинович Закросв Юрій Михайлович

підрозділ

Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету"

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фрази для коефіцієнта подібності 2

13704

Кількість слів

115680

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		18
Інтервали		0
Мікропробіли		1
Білі знаки		0
Парафрази (SmartMarks)	a	109

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

порядковий номер	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	Колір тексту
1	https://card-file.ontu.edu.ua/bitstreams/53ed22ad-8700-4162-b97a-082a1ad472d6/download	67 0.49 %
2	https://card-file.ontu.edu.ua/bitstreams/34a6756b-592f-4b77-a805-183aa03a6a26/download	61 0.45 %
3	Розробка 3D-гри у жанрі survival-horror з налаштуваннями рівнів складності 6/12/2025 Odesa Technical Professional College of Odesa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету")	47 0.34 %

4	https://card-file.ontu.edu.ua/server/api/core/bitstreams/995bdcec-4e4d-4321-8070-4d6badcb8e49/content	40 0.29 %
5	https://card-file.ontu.edu.ua/bitstreams/29489599-0581-4ce6-8890-c3b13d9f2e0e/download	37 0.27 %
6	https://card-file.ontu.edu.ua/bitstreams/549ee9fe-7574-4ae5-b500-9fe2711f33e6/download	37 0.27 %
7	https://card-file.ontu.edu.ua/bitstreams/dfa57ac3-98fa-4c22-86e7-0549d1254d89/download	35 0.26 %
8	https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content	31 0.23 %
9	https://card-file.ontu.edu.ua/bitstreams/53ed22ad-8700-4162-b97a-082a1ad472d6/download	30 0.22 %
10	https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download	29 0.21 %

з домашньої бази даних (0.64 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Розробка 3D-гри у жанрі survival-horror з налаштуваннями рівнів складності 6/12/2025 Odesa Technical Professional College of Odesa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету")	88 (4) 0.64 %

з програми обміну базами даних (1.14 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	ПРОГРАМНА РЕАЛІЗАЦІЯ МОНІТОРИНГУ ПРОДУКТИВНОСТІ ВЕБ-ДОДАТКІВ В РЕАЛЬНОМУ ЧАСІ 12/12/2024 Uzhhorod National University (Department)	52 (2) 0.38 %
2	Клієнт-серверна система аналізу даних для соціального опитування 6/15/2023 Uzhhorod National University (Department)	51 (6) 0.37 %
3	Система з підтримки вивчення методів тестування web-застосунків 3/15/2025 National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute)	29 (3) 0.21 %
4	Development of the mobile application "Apple of my eye" for people with limited vision 5/7/2024 International IT University (Информационные системы)	10 (1) 0.07 %
5	121 2024 магістр Стасько С.І 12/11/2024 Lutsk National Technical University (Lutsk National Technical University)	9 (1) 0.07 %
6	КРБ_KI_2025 6/16/2025 Uzhhorod National University (Department)	5 (1) 0.04 %

з Інтернету (14.38 %)

ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://card-file.ontu.edu.ua/server/api/core/bitstreams/44c16132-5f53-48e2-b6c0-61e9a2f0fd75/content	534 (45) 3.90 %
2	https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download	228 (15) 1.66 %

3	https://card-file.ontu.edu.ua/bitstreams/53ed22ad-8700-4162-b97a-082a1ad472d6/download	180 (10) 1.31 %
4	https://bibliotecadigital.udea.edu.co/bitstream/10495/43636/6/L%C3%B3pezSebastian_2024_GuiaRapidaEquipos.pdf	110 (9) 0.80 %
5	https://card-file.ontu.edu.ua/bitstreams/29489599-0581-4ce6-8890-c3b13d9f2e0e/download	88 (6) 0.64 %
6	https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content	84 (7) 0.61 %
7	https://card-file.ontu.edu.ua/bitstreams/34a6756b-592f-4b77-a805-183aa03a6a26/download	83 (2) 0.61 %
8	https://card-file.ontu.edu.ua/bitstreams/62baa43e-b968-4993-bb54-8cf8761a89b2/download	81 (8) 0.59 %
9	https://card-file.ontu.edu.ua/bitstreams/0e72a3b9-bdd7-4711-a3c6-dedc1d4287cc/download	65 (5) 0.47 %
10	https://card-file.ontu.edu.ua/bitstreams/549ee9fe-7574-4ae5-b500-9fe2711f33e6/download	61 (3) 0.45 %
11	https://card-file.ontu.edu.ua/bitstreams/035f6436-20b4-4ee6-8e99-bede670e308b/download	53 (5) 0.39 %
12	https://card-file.ontu.edu.ua/server/api/core/bitstreams/995bdcec-4e4d-4321-8070-4d6badcb8e49/content	47 (2) 0.34 %
13	https://card-file.ontu.edu.ua/bitstreams/bbed74c8-2ea7-44c5-8d00-0fe3fd9790ee/download	40 (4) 0.29 %
14	https://card-file.ontu.edu.ua/bitstreams/aed610a6-43ef-47e0-9066-e85c89456f3e/download	40 (4) 0.29 %
15	https://card-file.ontu.edu.ua/bitstreams/dfa57ac3-98fa-4c22-86e7-0549d1254d89/download	35 (1) 0.26 %
16	https://www.javaguides.net/2024/05/spring-boot-vuejs-jwt-authentication-example.html	32 (2) 0.23 %
17	http://eurasian-institute.org/images/public/chernigov/4.pdf	29 (1) 0.21 %
18	https://library.sumdu.edu.ua/images/FTP_materials/Novi_nahodzhennia_04_2023.pdf	26 (1) 0.19 %
19	https://card-file.ontu.edu.ua/bitstreams/bbaf3f38-16a8-4070-bead-5562769b7c71/download	23 (1) 0.17 %
20	https://studfile.net/preview/7191069/page:11/	21 (1) 0.15 %
21	https://stud.wiki/programming/3c0a65625a3ac69a4d43b89421206c26_1.html	15 (2) 0.11 %
22	https://card-file.ontu.edu.ua/bitstreams/9908b7a9-6b3e-46f5-a46e-84d83787cfd4/download	14 (1) 0.10 %
23	https://card-file.ontu.edu.ua/server/api/core/bitstreams/4bb7255e-46d4-4349-9726-9698476da02d/content	13 (1) 0.09 %
24	https://card-file.ontu.edu.ua/server/api/core/bitstreams/fc8a1853-39fc-4671-8807-2fd27ddb0779/content	13 (1) 0.09 %
25	https://card-file.ontu.edu.ua/bitstreams/6cf43324-8f08-4031-ba42-f80b18efbbc8/download	12 (2) 0.09 %
26	http://dspace.puet.edu.ua/bitstream/123456789/13982/1/Lukash_Denys_Igorovich-Knb-41-1.pdf	12 (2) 0.09 %
27	http://ukrefs.com.ua/print/page_1_171894-Proshivka-chipa-kartridzha-Samsung-SCX-4200.html	11 (1) 0.08 %
28	https://card-file.ontu.edu.ua/bitstreams/82a6d375-2b69-4233-b80f-fbfd149b7747/download	11 (1) 0.08 %
29	http://courses.webskill.com.ua/jsgamedev/courses/js/theme22/theme.html	10 (1) 0.07 %

Список принятых фрагментів (немає принятих фрагментів)

ПОРЯДКОВИЙ НОМЕР	ЗМІСТ	КІЛЬКІСТЬ ОДНАКОВИХ СЛІВ (ФРАГМЕНТІВ)
------------------	-------	---------------------------------------