

**МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ**

**ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І
СИСТЕМ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка web-додатку для аналізу відкритих даних податкових надходжень»

Здобувача вищої освіти другого (магістерського) рівня, II курсу, групи КМІз-21-1,
спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології в економіці»


(підпис)

Мандрик Владислав Володимирович

(прізвище, ім'я, по батькові)

Науковий керівник


(підпис)

Редич Олександр Володимирович, к.е.н., доцент

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

Керівник освітньо-професійної
програми


(підпис)

Горбовий Артур Юліанович, д.т.н., професор

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

Допущено до захисту за рішенням кафедри
протокол від «12» січня 2023 № 6

Завідувач
кафедри


(підпис)

Філоненко Михайло Миколайович, к.ф.-м.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Ірпінь – 2023

Реєстрація № 3

12.01.2023

(дата)



(підпис відповідальної особи)

Результати перевірки

до захисту
(до захисту, на доопрацювання)

12.01.2023
(дата)



(підпис)

Результати захисту

5 (відмінно)
(оцінка)

13.02.2023
(дата)



(підпис)

Голова комісії:



(підпис)

В.М. Франчук

(ініціали, прізвище)

Члени комісії:



(підпис)

М.М. Філоненко

(ініціали, прізвище)



(підпис)

О.В. Редич

(ініціали, прізвище)



(підпис)

Є.В. Котух

(ініціали, прізвище)

БЛАНК ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.П.: керівника, голови комісії членів комісії, секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст кваліфікаційної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	29		13.02.2023 Франчук В.М. <i>Венц</i> Філоненко М.М. <i>Філоненко</i>
2. Дотримання вимог щодо оформлення	від 0 до 10	8		Редич О.В. <i>Редич</i>
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	4		Котух С.В. <i>Котух</i>
4. Рецензент	від 0 до 5	5		Суховерхня Л.П. <i>Суховерхня</i>
РАЗОМ БАЛІВ*	від 0 до 50	46		

Оцінювання кваліфікаційної роботи науковим керівником - 46 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою	Дата	П.І.П.: голови, членів комісії, секретаря, (підпис)
92	A	5 (відмінно)	13.02.2023	Франчук В.М. <i>Венц</i> Філоненко М.М. <i>Філоненко</i> Редич О.В. <i>Редич</i> Котух С.В. <i>Котух</i> Суховерхня Л.П. <i>Суховерхня</i>

Правила переведення оцінок до національної шкали та шкали ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	
50-59	E	3 (задовільно)
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Спеціальність 122 «Комп'ютерні науки»
Освітня програма «Інформаційні управляючі системи і технології в економіці»

Затверджую

В.о. завідувача кафедри

 Федотова-Півень І.М.

«___» _____ 202__

**Завдання на кваліфікаційну роботу
здобувачу вищої освіти**

Мандрику Владиславу Володимировичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи

Розробка web-додатку для аналізу відкритих даних податкових надходжень

Затверджена наказом ректора від «05» січня 2022 № 4

2. Строк здачі закінченої роботи «11» січня 2023

3. Цільова установка: розробка додатку для аналізу відкритих даних податкових надходжень на основі використання відкритих даних державного веб-порталу Open budget

Мета роботи: провести дослідження технологій доступу та аналізу відкритих даних з використанням API та технології REST і на їх основі реалізувати веб-додаток

Об'єкт дослідження: середовище відкритих даних про надходження до бюджету України – openbudget.gov.ua

Предмет дослідження: технології розробки веб-додатків на базі Java Script, фреймворку ReactJS, API та технології REST.

4. Вихідні дані до роботи: відкриті дані про податкові надходження Open budget, інтерфейс програмування додатків API Open budget.

5. Перелік графічного матеріалу: рисунки та графіки, копії екранних вікон, алгоритми вирішення задач

6. Зміст кваліфікаційної роботи (перелік питань за кожним розділом)

Розділ 1. Огляд нормативного забезпечення та стану розвитку відкритих даних

1.1. Поняття відкритих даних. Огляд нормативних документів, що регламентують формування відкритих даних в Україні та ЄС

1.2. Систематизація веб-додатків, що використовують відкриті дані в ЄС та Україні

1.3. Порівняльний аналіз відкритих даних ЄС та України

Розділ 2. Огляд технологій і доступу до відкритих даних

2.1. Огляд та обґрунтування вибору технологій створення веб-додатків

2.2. Поняття API – використання, розробка та інтеграція в АІС

2.3. Технологія REST – використання, розробка та інтеграція в АІС

Розділ 3. Розробка веб-додатку для аналізу відкритих даних податкових надходжень

3.1. Основна мета та функції веб-додатку

3.2. Формування та програмування компонентів веб-додатку

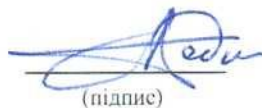
3.3. Розробка інтерфейсу та відображення даних

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	Фактично
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 27.12.2021 р.	27.12.2021
2	I розділ (розкрити сутність і стан дослідження наукової проблематики роботи, навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	26.05.2022 р.	26.05.2022
3	II розділ (аналіз експериментальної частини і методики досліджень)	20.11.2022 р.	20.11.2022
4	III розділ (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	30.11.2022 р.	30.11.2022
5	Висновки	15.12.2022 р.	15.12.2022
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	10.01.2023 р.	10.01.2023
7	Отримання відгуку керівника	15.01.2023 р.	03.02.2023
8	Отримання зовнішньої рецензії	25.01.2023 р.	06.02.2023

8. Дата видачі завдання «11» січня 2022

9. Науковий керівник кваліфікаційної роботи


(підпис)

Редич О.В.
(прізвище, ініціали)

10. Керівник освітньо-професійної програми


(підпис)

Горбовий А.Ю.
(прізвище, ініціали)

11. Завдання до виконання отримав здобувач вищої освіти


(підпис)

Мандрик В.В.
(прізвище, ініціали)

АНОТАЦІЯ

РОЗРОБКА WEB-ДОДАТКУ ДЛЯ АНАЛІЗУ ВІДКРИТИХ ДАНИХ ПОДАТКОВИХ НАДХОДЖЕНЬ

Мета роботи полягає в дослідженні технологій доступу та аналізу відкритих даних з використанням API та технології REST і реалізації на їх основі веб-додатку.

Завданнями роботи є розробка веб-додатку для аналізу відкритих даних податкових надходжень.

Предметом дослідження є технології розробки веб-додатків на базі JavaScript, фреймворку ReactJS, API та технології REST.

Об'єктом дослідження виступає середовище відкритих даних про надходження до бюджету України openbudget.gov.ua.

Базою дослідження виступає API та масив відкритої інформації про надходження до бюджету України на сайті openbudget.gov.ua.

За результатами дослідження сформульовані висновки щодо актуальності та розвитку відкритих даних; розглянуто сервіси, що надають доступ до відкритих даних; технології для отримання даних з сервера та роботи з ними; технології створення веб-сайтів та веб-додатків; розроблено веб-додаток для аналізу відкритих даних податкових надходжень.

Розроблений веб-додаток може отримувати відкриті дані надходжень до бюджету держави з державного веб-порталу OpenBudget через API та обробляти, відображати їх як у вигляді таблиці з можливістю фільтрації та подальшого їх збереження на ПК у одному з представлених форматів, так і у вигляді діаграм (з однією можна взаємодіяти) для кращого аналізу.

Одержані результати можуть бути використані у науковій діяльності освітніх та дослідницьких установ, в органах місцевого самоврядування для швидкого отримання актуальних даних про надходження до бюджету держави, здійснення їх базового графічного та статистичного аналізу.

У подальшій роботі над додатком можна приділити увагу розширенню функціональності у частині графічного представлення інформації, інтеграції методів машинного навчання, штучного інтелекту для прогнозування податків, класифікації регіонів, оцінювання економічних ризиків тощо.

Результати досліджень були апробовані у вигляді тез на Міжнародній науково-практичній конференції «Development of modern science, experience and trends», що проходила 11-14 жовтня 2022 р. у Бостоні, США.

Кваліфікаційна робота містить 68 сторінок, 36 рисунки, список використаних джерел з 43 найменування, 4 додатків.

Ключові слова: *відкриті дані; веб-додаток; веб-сайт; податкові надходження; надходження до бюджету; API; REST; React; HTTP; інтерфейс програмування додатків; HTML; CSS; JavaScript.*

Рік виконання кваліфікаційної роботи 2021-2023.

Рік захисту роботи 2023.

ANNOTATION

DEVELOPMENT OF A WEB APPLICATION FOR THE ANALYSIS OF OPEN DATA OF TAX REVENUES

The purpose of the work is to research open data access and analysis technologies using API and REST technology and to implement a web application based on them.

The tasks of the work are the development of a web application for the analysis of open data of tax revenues.

The subject of the research is web application development technologies based on Java Script, ReactJS framework, API and REST technology.

The object of the study is the environment of open data on receipts to the budget of Ukraine openbudget.gov.ua.

The basis of the study is the API and the array of open information on receipts to the budget of Ukraine on the website openbudget.gov.ua.

According to the results of the research, conclusions are formulated regarding the relevance and development of open data; considered web services providing access to open data; technologies for receiving data from the server and working with them; technologies for creating websites and web applications; a web application was developed for the analysis of open data of tax revenues.

The developed web application can receive open data of receipts to the state budget from the state web portal OpenBudget through API and process, display them both in the form of a table with the possibility of filtering and their subsequent saving on a PC in one of the presented formats, and display them in the form of diagrams (one can be interacted with) for better analysis.

The obtained results can be used in the scientific activity of educational and research institutions, in local self-government bodies to quickly obtain current data on receipts to the state budget, to carry out their basic graphical and statistical analysis.

In further work on the application, attention can be paid to expanding the functionality in terms of graphical presentation of information, integration of machine learning methods, artificial intelligence for tax forecasting, classification of regions, assessment of economic risks, etc.

The research results were tested in the form of theses at the International scientific and practical conference "Development of modern science, experience and trends", which took place on October 11-14, 2022 in Boston, USA.

The qualification thesis contains 68 pages, 36 figures, a list of used sources with 43 names, 4 appendices.

Keywords: open data; web application; website; tax revenues; revenue to the budget; API; REST; React; HTTP; application programming interface; HTML; CSS; JavaScript.

The year of completion of the thesis is 2021-2023.

The year of job protection is 2023.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ	3
ВСТУП.....	4
РОЗДІЛ 1. ОГЛЯД НОРМАТИВНОГО ЗАБЕЗПЕЧЕННЯ ТА СТАНУ РОЗВИТКУ ВІДКРИТИХ ДАНИХ.....	6
1.1. Поняття відкритих даних. Огляд нормативних документів, що регламентують формування відкритих даних в Україні та ЄС.....	6
1.2. Систематизація веб-додатків, що використовують відкриті дані в ЄС та Україні.....	9
1.3. Порівняльний аналіз відкритих даних ЄС та України	17
РОЗДІЛ 2. ОГЛЯД ТЕХНОЛОГІЙ І ДОСТУПУ ДО ВІДКРИТИХ ДАНИХ	20
2.1. Огляд та обґрунтування вибору технологій створення веб-додатків ...	20
2.2. Поняття API – використання, розробка та інтеграція в АІС	25
2.3. Технологія REST – використання, розробка та інтеграція в АІС	29
РОЗДІЛ 3. РОЗРОБКА ВЕБ-ДОДАТКУ ДЛЯ АНАЛІЗУ ВІДКРИТИХ ДАНИХ ПОДАТКОВИХ НАДХОДЖЕНЬ.....	38
3.1. Основна мета та функції веб-додатку	38
3.2. Формування та програмування компонентів веб-додатку	40
3.3. Розробка інтерфейсу та відображення даних.....	42
ВИСНОВКИ	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	58
ДОДАТКИ.....	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

БД – база даних

ВД – відкриті дані

Веб-додаток – клієнт-серверна програма, у якій клієнт взаємодіє з веб-сервером за допомогою браузера

ЄС – Європейський Союз

ЗУ – Закон України

ЗУ про доступ – Закон України “Про доступ до публічної інформації”

API – Application Programming Interface – міжпрограмний інтерфейс, який дозволяє двом програмам обмінюватися даними

CSS – Cascading Style Sheets – набір параметрів форматування, який застосовується до елементів документа, щоб змінити їх зовнішній вигляд

DOM – Document Object Model – інтерфейс програмування для веб-документів, за допомогою якого програми можуть змінювати структуру документа, його стиль та вміст

HTML – HyperText Markup Language – основний будівельний блок Інтернету, що визначає зміст та структуру веб-контенту

HTTP – HyperText Transfer Protocol – протокол передачі гіпертекстових документів, що є основою Всесвітньої павутини та використовується для завантаження веб-сторінок

JS – JavaScript – кросплатформна, об’єктно-орієнтована мова програмування

JSON – JavaScript Object Notation – текстовий формат для представлення структурованих даних на основі синтаксису об’єктів JavaScript, що зазвичай використовується для передачі даних у веб-додатках

REST – Representational State Transfer – архітектурний стиль проектування API та взаємодії компонентів розподіленого додатка в мережі

URL – Uniform Resource Locator – стандартизований спосіб запису адреси ресурсу в мережі Інтернет

ВСТУП

Відкриті дані є частиною широкого глобального руху, який не лише просуває науку та наукову комунікацію, але й трансформує сучасне суспільство та способи прийняття рішень. «Напочатку 2000-х років разом із поширенням інтернету за ініціативи електронного уряду та урядування почала набувати популярності ідея відкритих даних. Згодом цю ініціативу підхопила Україна» [19, с. 444]. За останнє десятиліття було досягнуто ряд прогресів у сфері відкритих даних, ініціаторами яких були переважно уряди. Відкриті дані сприяють науковому співробітництву, збагачують дослідження та розширюють аналітичні можливості для прийняття рішень. «Вони повинні бути опубліковані у машиночитанному форматі, оскільки без нього використання даних суттєво ускладнюється, дані є непридатними для автоматизованого оброблення інформаційними системами і, відповідно, втрачають свою “відкритість”» [19, с. 444].

Публікація даних ґрунтується на переконанні, що це приносить величезну користь громадянам, підприємствам і державним адміністраціям, водночас сприяючи тіснішій співпраці в Європі. Відкриті дані можуть принести користь у різних сферах, таких як охорона здоров'я, продовольча безпека, освіта, клімат, інтелектуальні транспортні системи та розумні міста, і вважаються «важливим ресурсом для економічного зростання, створення робочих місць і суспільного прогресу». Тож доступність відкритих даних створює можливості для різноманітних організацій, урядових установ і некомерційних організацій винаходити нові шляхи вирішення проблем суспільства.

Так, з розвитком сфери відкритих даних стає актуальним та набирає популярності створення веб-додатків, що обробляють відкриті дані (ВД) та надають їх як у зручному для користувача вигляді, так і машиночитанному форматі, надають інформацію на їх основі у корисному для користувача вигляді (наприклад, інтерактивна мапа доріг, що надає інформацію про те, де

обмежено рух або ремонтуються дороги), дозволяють аналізувати дані чи використовувати їх у своїх проектах.

Відкритими стають також і дані про фінансові надходження до бюджету держави. Завдяки цьому громадяни можуть стежити за тим, на що витрачаються їхні кошти, які вони сплачують у вигляді податків, аналізувати інформацію про те, як сплачують податки в Україні, перевірити ФОП та потенційних бізнес-партнерів на фінансову спроможність та доброчесність.

Використання інструментів контролю бюджетів та моніторинг державних закупівель допомагають викривати корупційні схеми, запобігають фінансовим втратам держави, полегшують їх аналіз, сприяють прозорості та більш ефективному використанню державних фінансів.

Тому завданням дипломного проекту є розробка веб-додатку для аналізу відкритих даних податкових надходжень, що буде надавати інформацію на основі ВД фінансових надходжень до бюджету держави у вигляді таблиці з можливістю їх фільтрації та подальшому збереженню у одному з запропонованих форматів, буде відображати інформацію по даних у вигляді діаграм для кращого їх аналізу.

Мета роботи полягає в дослідженні технологій доступу та аналізу відкритих даних з використанням Application Programming Interface (API) та технології REST і реалізації на їх основі веб-додатку.

Об'єктом дослідження виступає середовище відкритих даних про надходження до бюджету України openbudget.gov.ua.

Предметом дослідження є технології розробки веб-додатків на базі JavaScript, фреймворку ReactJS, API та технології REST.

Базою дослідження виступає API та масив відкритої інформації про надходження до бюджету України на веб-порталі openbudget.gov.ua.

Результати досліджень були апробовані у вигляді тез на Міжнародній науково-практичній конференції «Development of modern science, experience and trends», що проходила 11-14 жовтня 2022 р. у Бостоні, США.

РОЗДІЛ 1. ОГЛЯД НОРМАТИВНОГО ЗАБЕЗПЕЧЕННЯ ТА СТАНУ РОЗВИТКУ ВІДКРИТИХ ДАНИХ

1.1. Поняття відкритих даних. Огляд нормативних документів, що регламентують формування відкритих даних в Україні та ЄС

«Публічна інформація у формі відкритих даних (відкриті дані) – це дані, які будь-хто може вільно використовувати й поширювати» [41].

«Відкриті дані мають бути опубліковані в мережі Інтернет у машиночитаному форматі – форматі, придатному для автоматизованого оброблення інформаційними системами. Машиночитаний формат дає змогу інформаційним системам ідентифікувати, розпізнавати, перетворювати й отримувати конкретні дані без участі людини. Поширені машиночитані формати – CSV, XML, JSON.

Влада, бізнес, громадські організації, наукові установи і приватні особи можуть вільно публікувати, копіювати, поширювати, використовувати відкриті дані для одержання економічної, соціальної, культурної чи екологічної вигоди.

Відкриті дані можна вільно використовувати, зокрема з комерційною метою, у поєднанні з іншою інформацією або включаючи їх до складу власного продукту з обов’язковим зазначенням джерела їх отримання» [37].

Найкращою практикою оприлюднення відкритих даних є:

- первинні, неагреговані, гранулярні та максимально деталізовані дані;
- машиночитаний формат, який дає змогу інформаційним системам ідентифікувати, розпізнавати, перетворювати й отримувати конкретні дані без участі людини. Поширені машиночитані формати – CSV, XML, JSON;
- дані у вільному доступі для використання й поширення, зокрема з комерційною метою.

Сюди можна віднести дані у різних машиночитаних форматах, що надає Єдиний державний веб-портал відкритих даних – data.gov.ua; відкриті дані, що доступні через інтерфейс прикладного програмування (API),

зокрема машиночитані дані офіційного державного веб-порталу по надходженнях до бюджету держави – OpenBudget тощо.

Дані, що не належать до відкритих. Сюди відносяться електронні сервіси, що не надають вихідні дані у їх первинному вигляді, зокрема інтерактивні мапи, електронні черги, петиції, документообіг тощо, а також оприлюднена структурована публічна інформація у немашиночитаних форматах на офіційних сайтах, така як звіти, плани та інші документи. Наприклад, не є відкритими дані, що оприлюднюються у вигляді відсканованих таблиць; інтерактивна мапа доріг, що може лише надавати інформацію про те, де обмежено рух або ремонтуються дороги, проте не надає доступ до самих (первинних) даних, на основі яких і побудовано цю мапу.

Нормативно-правова база для організації роботи з відкритими даними в Україні. Головним регламентом в Україні щодо відкритих даних є ЗУ про доступ [41], що дає визначення поняття відкритих даних (стаття 10.1) та вказує умови їх використання (стаття 10.2).

«Стаття 10.1. Публічна інформація у формі відкритих даних – це публічна інформація у форматі, що дозволяє її автоматизоване оброблення електронними засобами, вільний та безоплатний доступ до неї, а також її подальше використання.

Розпорядники інформації зобов'язані надавати публічну інформацію у формі відкритих даних на запит, оприлюднювати і регулярно оновлювати її на єдиному державному веб-порталі відкритих даних та на своїх веб-сайтах» [41].

«Стаття 10.2. Публічна інформація у формі відкритих даних є дозволеною для її подальшого вільного використання та поширення.

Будь-яка особа може вільно копіювати, публікувати, поширювати, використовувати, у тому числі в комерційних цілях, у поєднанні з іншою інформацією або шляхом включення до складу власного продукту, публічну

інформацію у формі відкритих даних з обов'язковим посиланням на джерело отримання такої інформації» [41].

Для виконання цього закону Кабінетом Міністрів України (КМУ) було розроблено та ухвалено Постанову №835 [42], у якій визначаються формати, структура, вимоги, перераховуються набори ВД, що є обов'язковими до оприлюднення розпорядниками публічної інформації.

20 грудня 2017 року КМУ було внесено зміни до Постанови №835, що розширювали перелік наборів для оприлюднення.

Нормативно-правова база для організації роботи з відкритими даними в ЄС. 2003 року Європейський Союз прийняв Директиву 2003/98/ЄС, яка запроваджувала мінімальний набір правил для регулювання вторинного використання документів, що відносяться до органів публічного сектору. У даній Директиві зазначаються цілі стосовно створення умов, які допомагають розвивати послуги в усьому Співтоваристві та не допускати викривлення конкуренції на внутрішньому ринку.

20 червня 2019 року в ЄС було прийнято Директиву 2019/1024/ЄС. Вона впроваджувала набір нових правил регулювання щодо сфери вторинного використання інформації та відкритих даних.

У прийнятій Директиві можна визначити наступні ключові зміни у сфері відкритих даних:

- Безкоштовне вторинне використання. Вся інформація державного сектору, до якої можна дістати доступ за національними правилами держав-членів ЄС, в принципі має бути вільно (безкоштовно) доступною для подальшого використання.

- Відкритість за замовчанням. Держави-члени мають заохочувати органи публічного сектору та публічні підприємства створювати та робити доступною інформацію.

- Набори даних високої цінності. Вони повинні бути визначеними Європейською Комісією, доступними через API та бути безкоштовними. До поточного переліку тематичних категорій наборів даних високої цінності

входять: геопросторові дані; спостереження землі та навколишнього середовища; метеорологічні; статистика; компанії та володіння компаніями; транспорт.

– Динамічні дані. Інформація має бути машиночитаного формату та постійно чи в режимі реального часу оновлюватися, зокрема – через її непостійність або швидке застарівання.

– Комп'ютерні програми. Державам-членам ЄС дозволяється застосовувати норми Директиви 2019/1024/ЄС на комп'ютерні програми, якщо їх поширення не обмежується правом інтелектуальної власності (наприклад, програмне забезпечення з відкритим кодом).

– Дані досліджень. Сюди відносяться статистичні дані, результати експериментів, вимірювання, спостереження, отримані в результаті роботи на місцях, результати опитувань, записи інтерв'ю та зображення. Все це включає в себе метадані, специфікації та інші цифрові об'єкти.

1.2. Систематизація веб-додатків, що використовують відкриті дані в ЄС та Україні

«Великий обсяг інформації, що збирається та обробляється органами влади, робить діапазон використання відкритих даних надзвичайно широким. Користуватися відкритими даними або сервісами, створеними на їх основі, може кожен громадянин. Інформація, опрелюднена у формі відкритих даних, може вплинути на рішення у будь-яких сферах: від купівлі нерухомості (наприклад, інформація про нерухоме майно в заставі) до шляху на роботу (наприклад, інформація про громадський транспорт)» [19, с. 444].

Для зберігання та надання відкритих даних як в Україні, так і в ЄС, створені систематизовані веб-портали відкритих даних, що зберігають публічну інформацію у формі ВД та надають доступ до неї широкому колу осіб за принципами, що регламентуються у відповідних нормативних документах стосовно ВД.

В Україні створено багато систематизованих веб-порталів відкритих даних. Основний серед них – Єдиний державний веб-портал відкритих даних – data.gov.ua (рис. 1.1), який «створено на вимогу Закону України “Про доступ до публічної інформації” та постанови Кабінету Міністрів України від 21 жовтня 2015 року № 835 “Про затвердження Положення про набори даних, які підлягають оприлюдненню у формі відкритих даних”» [35].

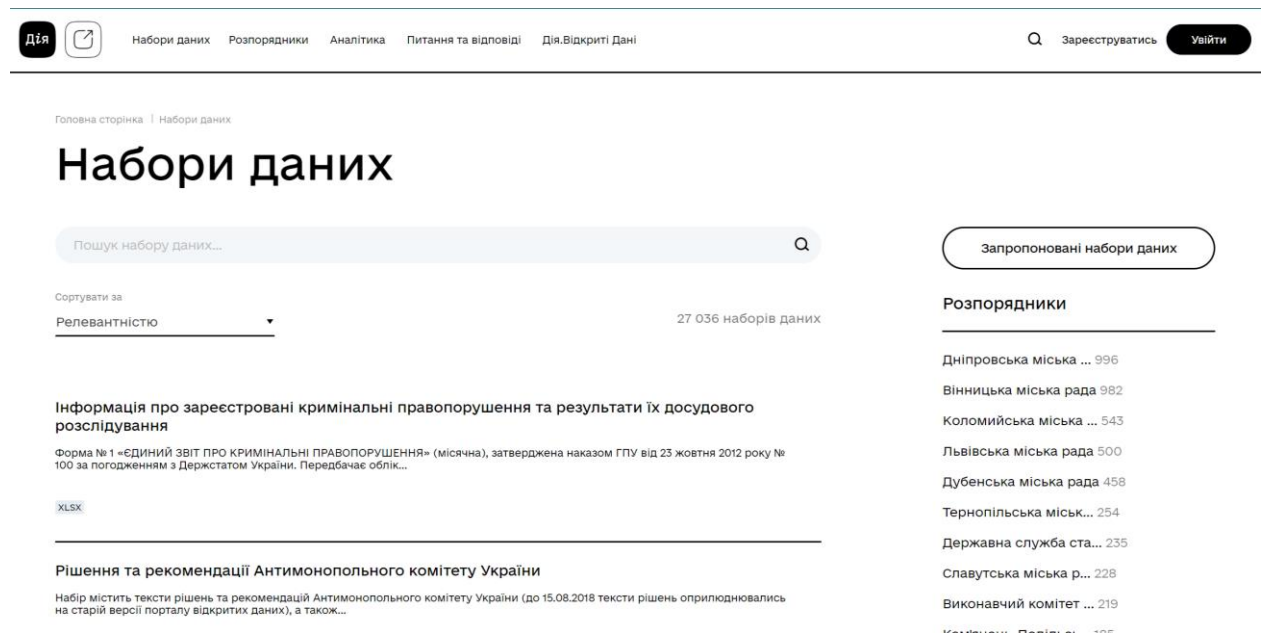


Рис. 1.1. Єдиний державний веб-портал відкритих даних [35]

Портал розроблено для забезпечення громадян доступом до публічної інформації та інформації органів влади у машиночитаному форматі у якості ВД із можливістю її подальшого використання, у тому числі в комерційних цілях [35].

Веб-портал працює на платформі CKAN, що дає змогу повноцінно використовувати API порталу як розпорядникам – для додавання наборів даних на портал так і користувачам – для завантаження наборів з порталу.

Open budget. «Open budget – це офіційний державний веб-портал, який відповідає кращому міжнародному досвіду у сфері відкритості публічних фінансів» [14], створений у 2018 році.

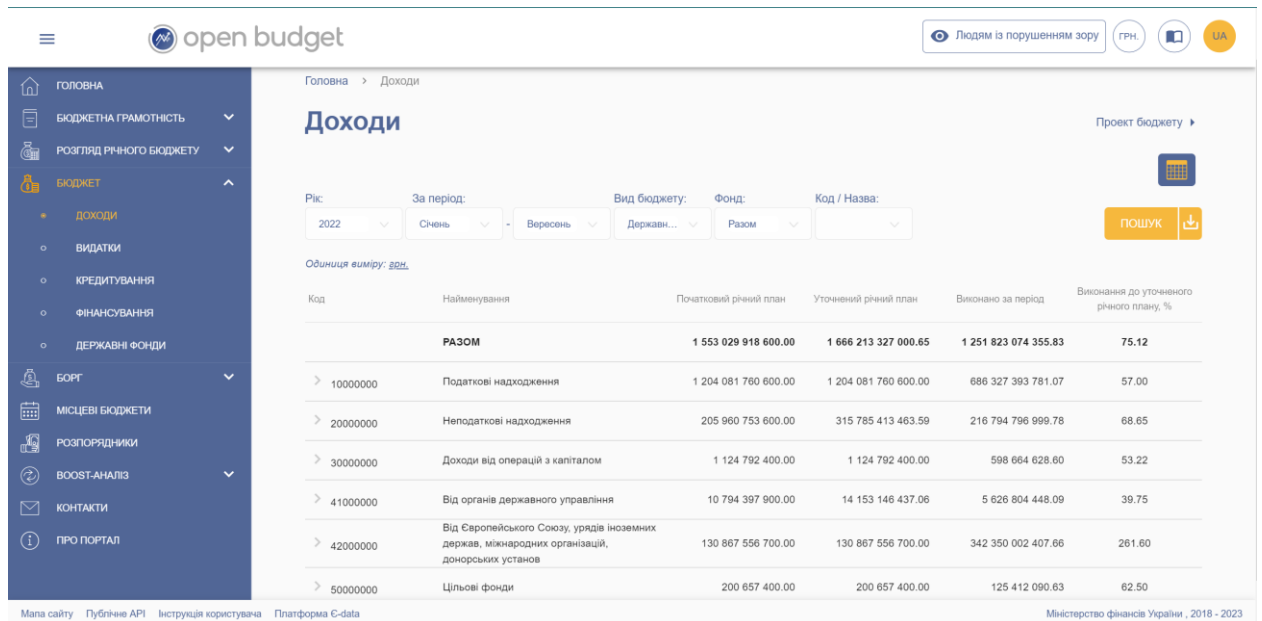


Рис. 1.2. Веб-портал Open budget [14]

Open budget надає дані про бюджетну політику України у відкритий доступ, щоб будь-хто з громадян міг дізнатися планування бюджету та перевірити, куди і на що витрачаються їхні кошти як платників податків. Також дозволяє дізнатися інформацію щодо доходів та видатків окремо по громадах.

Дані подаються у машиночитаному форматі, які можна вивантажити за допомогою публічного API, що надає портал Open budget.

Єдиний веб-портал використання публічних коштів. «Spending – це офіційний державний веб-портал з відкритими даними про використання бюджетних коштів в Україні» [34], який запрацював 15 вересня 2015 року (рис. 1.3).

Spending дозволяє отримати будь-якому громадянину дані «про використання публічних коштів розпорядниками та одержувачами коштів державного і місцевих бюджетів, суб'єктами господарювання державної і комунальної форм власності, Пенсійним фондом України та фондами загальнообов'язкового державного соціального страхування» [34].

ЗВЕРТАЄМО УВАГУ, ЗА НЕОПРИЛЮДНЕННЯ ІНФОРМАЦІЇ ПЕРЕДБАЧЕНО АДМІНІСТРАТИВНУ ВІДПОВІДАЛЬНІСТЬ					
ЛЮДЯМ ІЗ ПОРУШЕННЯМ ЗОРУ Про портал Інструкція користувача Статистика Службовий вхід Реєстрація					
spending ГОЛОВНА НОВИНИ ПОШУК ЗА РОЗПОРЯДНИКАМИ ПОШУК ЗА ТРАНСАКЦІЯМИ					
ГОЛОВНА > ПОШУК ЗА ТРАНСАКЦІЯМИ					
ТОП 100 НАЙБІЛЬШИХ ТРАНСАКЦІЙ З ПОЧАТКУ РОКУ (ЗА ДАНИМИ ДКСУ)					
Пошук на сторінці					
	ПЛАТНИК (код)	ОДЕРЖУВАЧ (код)	ПРИЗНАЧЕННЯ ПЛАТЕЖУ	ДАТА	СУМА
ПЛАТНИК	Мінфін (00013480)	Державна казначейська служба України, (37567646)	КЕКВ 0000; Перерахування за призначенням: Зарахування надходжень від випуску Військових ОВДП, постанова КМУ від 25.02.2016 №156.	09.03.2022	20 006 000 000.00 UAH
Код за ЄДРПОУ або назвою платника					
ОДЕРЖУВАЧ	Мінекономіки (37508596)	АТ "Ощадбанк" (00032129)	Надання допомоги в рамках Програми "Підтримка", що здійснюється на виконання ПКМУ09.12.21 №000 (зі змінами), подан. 20.03.22 №000-04/130, д. 16.12.21 №25	21.03.2022	4 404 991 500.00 UAH
Код за ЄДРПОУ або назвою одержу...					
ПРИЗНАЧЕННЯ	Мінекономіки (37508596)	АТ "Ощадбанк" (00032129)	Надання допомоги в рамках Програми "Підтримка", що здійснюється на виконання ПКМУ09.12.21 №000 (зі змінами), подан. 21.03.22 №000-04/131, д. 16.12.21 №252	22.03.2022	3 633 981 000.00 UAH
Призначення платежу					
СУМА	ДПС (43005393)	Пенсійний фонд України (00035323)	Кошти на загальнообов'язкове державне пенсійне страхування (до солідарної системи), клас профізику 201 (85,0000%)	08.02.2022	3 632 216 118.39 UAH
Від					
До					
ЗА ПЕРІОД*	Пенсійний фонд України (00035323)	Короткостр. позики, надз. а рах. коштів ЄК (37567646)	Дострокове погашення позики на виконання листа ПФУ від 25.02.2016 №6/н та п. 3.4.1 Договору 11.02.2016 №9-19/02/27-22	28.02.2022	3 613 180 675.08 UAH
09.04.2022					
30.04.2022					
УПРАВЛІННЯ ДКСУ	Виділення асигнувань по зф (37567646)	Мінфін (00013480)	Розпорядження МФУ 28.02.2016 №000	02.03.2022	3 379 352 109.40 UAH

Рис. 1.3. Єдиний веб-портал використання публічних коштів [34]

З його допомогою система державних фінансів і процедури бюджетування стають більш прозорими, задовольняються потреби громадськості в оперативному отриманні достовірних даних, а також портал впливає на підвищення відповідальності розпорядників та одержувачів бюджетних коштів на державному та місцевому рівнях.

Дані подаються у машиночитаному форматі, які можна вивантажити за допомогою публічного API, що надає веб-портал Spending.

Міжнародні фінансові проекти. «IFIs projects – це офіційний державний веб-портал, який містить вичерпну інформацію у форматі відкритих даних про проекти соціального і економічного розвитку України, які реалізуються із залученням коштів міжнародних фінансових організацій (МФО)» [39], створений у 2019 році (рис. 1.4).

Даний веб-портал надає у вільний доступ громадянам інформацію про співпрацю України з МФО, щодо реалізації проектів соціального та економічного розвитку, а також робить більш прозорою сферу публічних фінансів та являється важливим джерелом для громадськості та бізнесу, зокрема, щодо розвитку регіонів.

The screenshot shows the IFIs projects web portal. The left sidebar contains navigation links: ГОЛОВНА, ПОШУК ПРОЕКТІВ, СТАТИСТИКА, НОВИНИ, НОРМАТИВНА БАЗА, ПРОГРАМНИЙ ІНТЕРФЕЙС (API), ГЛОСАРІЙ, КОНТАКТИ, and ПРО ПОРТАЛ. The main content area displays the project title 'Транспортний зв'язок в Україні – Фаза I 87.617' with a date of update '14.12.2022'. Below this is a section 'Основна інформація про проект' containing a table with project details.

Основна інформація про проект	
Назва	Транспортний зв'язок в Україні – Фаза I 87.617
Стан	Ініціювання ✓ Підготовка ✓ Виконання 3 Завершений 4
Галузь реалізації	Транспорт, складське господарство
Мета проекту	Покращення економічних, безпекових та екологічних складових транспортного зв'язку в Україні шляхом інвестування в удосконалення серії окремих невеликих дорожніх об'єктів для покращення автодорожнього сполучення у Східному партнерстві.
Кредитор (МФО)	ЄІБ - Європейський Інвестиційний Банк
Ініціатор	Державне агентство автомобільних доріг України
Бенефіціар	Державне агентство автомобільних доріг України
Відповідальний виконавець	Державне агентство автомобільних доріг України
Вартість проекту	50 000 000.00 EUR
Орієнтовна сума позики	50 000 000.00 EUR
Строк реалізації проекту	22.09.2020 - 22.09.2025

Рис. 1.4. Веб-портал із реалізації міжнародних фінансових проектів [39]

Opendatabot. «Opendatabot – це платформа, що надає доступ до державних даних для громадян та бізнесу» [15]. Пошук інформації здійснюється виключно в офіційних та відкритих джерелах – у понад 20 державних реєстрах.

The screenshot shows the Opendatabot web portal. The header includes the Opendatabot logo, a search bar for company codes, and a 'Увійти' button. The main section is titled 'Опендатебот громадянам' with the subtitle 'Знати все про державу. Отримувати повідомлення про суди, борги та нерухомість'. A 'Підключитись' button is present. Below are three service cards: 'Людина' (person icon) with questions about court cases and debts; 'сім'я' (house icon) with questions about traffic fines and property; and 'Безпека' (shield icon) with questions about car safety, passport validity, and company checks.

Рис. 1.5. Веб-портал Opendatabot [15]

Веб-портал надає можливість громадянам швидко знайти необхідну інформацію про людину, бізнес, автомобіль, штрафи, нерухомість, судові справи та розклади тощо.

Відкритий парламент. «Портал призначений для забезпечення надання доступу до публічної інформації Верховної Ради України у формі відкритих даних та передбачає доступ до інформації з можливістю її наступного безкоштовного використання» [31] (рис. 1.6).

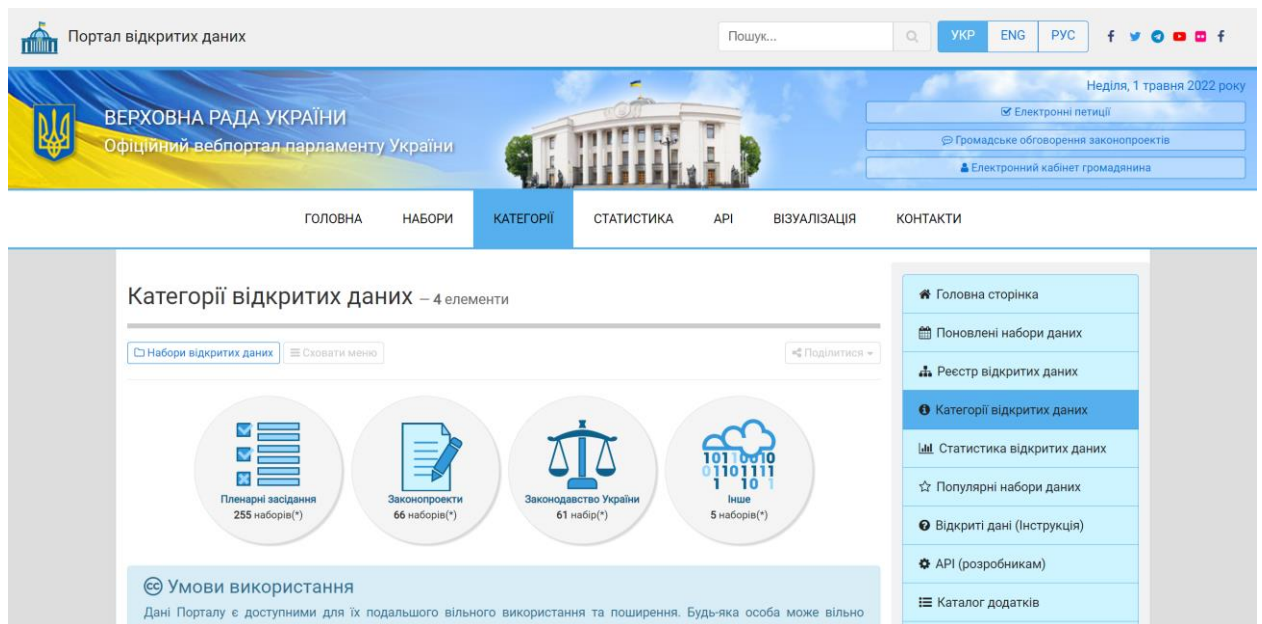


Рис. 1.6. Веб-портал відкритих даних парламенту України [31]

Веб-портал оприлюднює у якості ВД нормативно-правову базу України, інформацію про розгляд питань порядку денного ВР України, про зареєстровані законопроекти, про народних депутатів тощо.

Офіційний портал європейських даних. У країнах ЄС також є свої портали з відкритими даними. Для того, щоб зібрати всі такі дані в одному місці, був створений веб-портал The official portal for European data (рис. 1.7) – data.europa.eu (Офіційний портал європейських даних). Портал збирає метадані загальнодоступних даних по всій Європі, надає доступ до відкритих даних із міжнародних, європейських, національних, регіональних, місцевих та географічних порталів даних. Виходячи за рамки збору метаданих (даних

про дані), стратегічна мета portalу – покращити доступність та підвищити цінність відкритих даних.



Рис. 1.7. Офіційний портал європейських даних [21]

Офіційний портал європейських даних також, як і єдиний державний веб-портал відкритих даних, працює на платформі SKAN, що надає можливість використовувати API portalу як для додавання наборів даних на портал, так і для завантаження їх з portalу.

На веб-порталі є корисні розділи для користувачів стосовно пошук, надання та використання даних.

Євростат – європейська статистика. Євростат – це статистичне управління Європейського Союзу, відповідальне за публікацію високоякісної загальноєвропейської статистики та показників, які дозволяють порівнювати країни та регіони [6] (рис. 1.8).

Євростат має на меті [6]:

- надавати європейським інституціям та урядам держав-членів ЄС інформацію, необхідну для розробки, впровадження, моніторингу та оцінки європейської політики;

- розповсюджувати та надавати статистичні дані всім типам користувачів на основі рівних можливостей, щоб державні адміністрації, науковці, профспілки, студенти, підприємства та неурядові організації (НУО), зокрема, мали вільний і легкий доступ до даних;
- запровадити набір стандартів, методів та організаційних структур, які дозволяють створювати порівнянну, надійну та відповідну статистику в усьому ЄС, відповідно до принципів Кодексу практики європейської статистики;
- підтримати країни-члени ЄС та сприяти розвитку статистичних систем на міжнародному рівні.

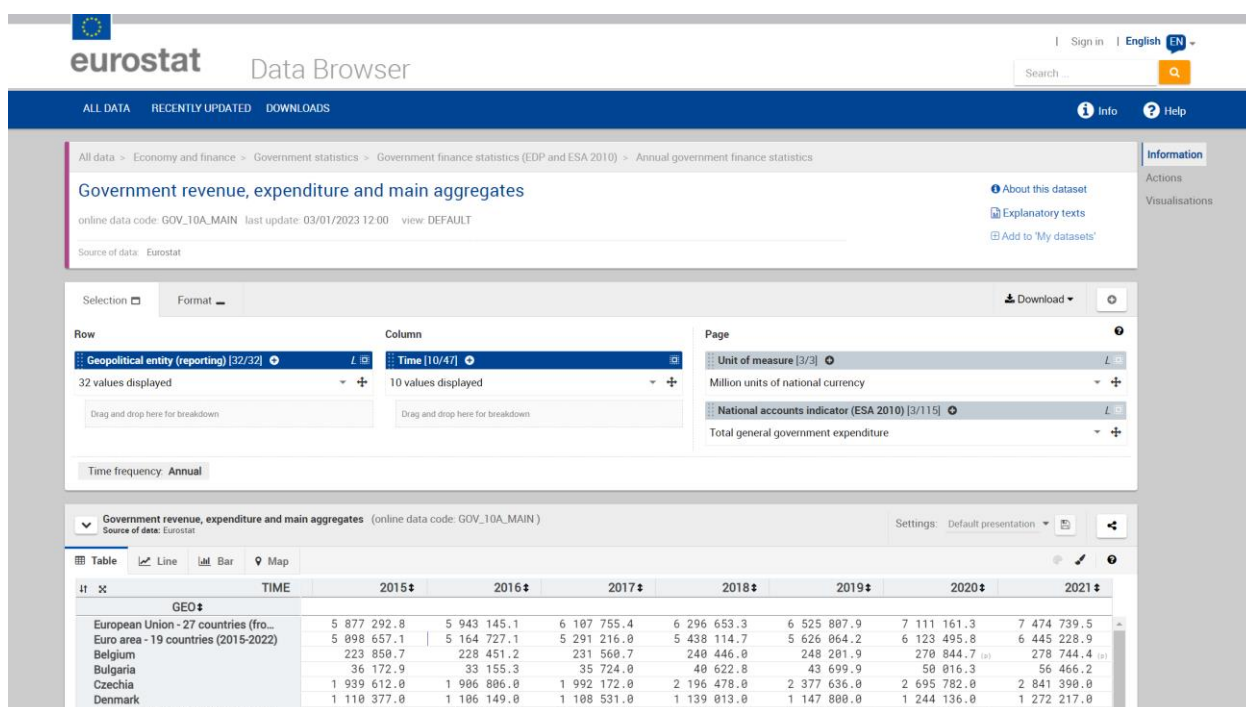


Рис. 1.8. Веб-портал Eurostat (дані річної статистики державних фінансів) [6]

Open Data Monitor. Веб-портал Open Data Monitor надає можливість отримати огляд доступних відкритих ресурсів даних країн ЄС, а також провести аналіз і візуалізацію існуючих каталогів даних за допомогою інноваційних технологій (рис. 1.9).

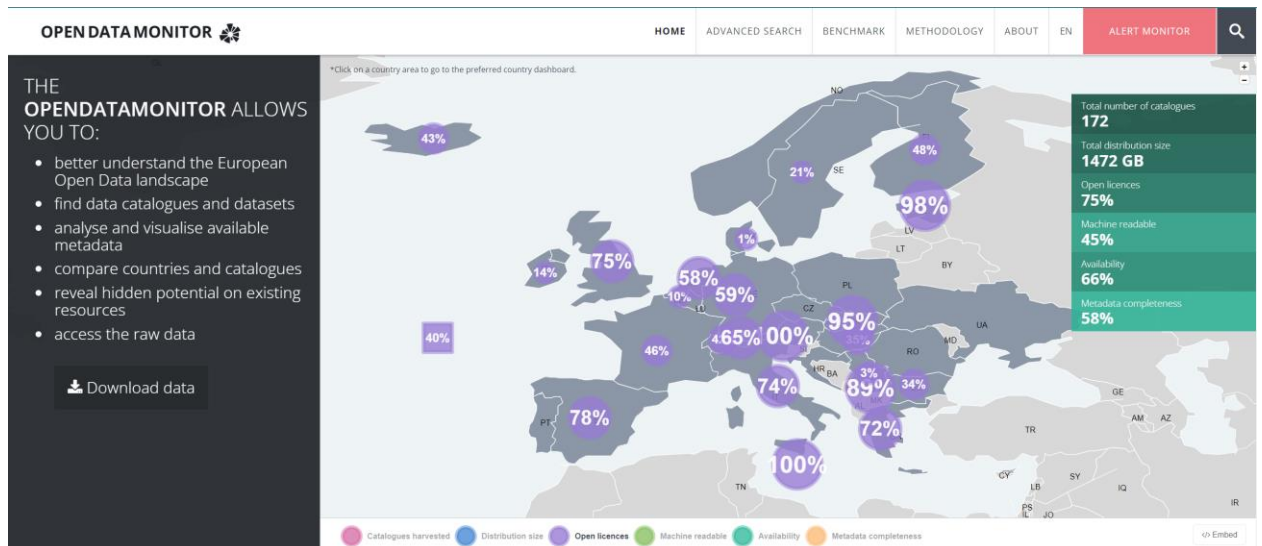


Рис. 1.9. Веб-портал Open Data Monitor [16]

Портал збирає набори даних, збираючи метадані з різних відкритих джерел даних. Зібрані метадані потім гармонізуються, оскільки зібрана інформація структурується та обробляється. Масштабовані методи аналітики та візуалізації дозволяють кінцевим користувачам дізнатися більше про склад регіональних, національних або загальноєвропейських сховищ відкритих даних. Наприклад, сукупність каталогів одного регіону чи країни можна легко візуалізувати, щоб скласти точну картину ситуації з відкритими даними та порівняти з іншими областями. Аналіз і візуалізація метаданих дозволить виявити прихований потенціал і важливу інформацію з наявних ресурсів і виявити прогалини, де потрібні додаткові відкриті дані.

Отже, на сьогодні все більше розвиваються відкриті дані та створюються на їх основі різні веб-портали, що дають можливість як звичайним громадянам, так і бізнесу, відповідно до своїх потреб знайти швидко та зручно ту чи іншу інформацію.

1.3. Порівняльний аналіз відкритих даних ЄС та України

Щорічно, починаючи з 2015 року, The official portal for European data (EDP, що являється основною точкою входу до даних, що публікують європейські країни) проводиться дослідження Open Data Maturity Report

(ODM Report), під час якого оцінюється розвиток сфери ВД європейських країн, а також надаються рекомендації стосовно покращення ситуації у цій сфері. 2020-го року вперше прийняли участь у дослідженні не лише країни ЄС, Європейської асоціації вільної торгівлі та Великої Британії, а й східні партнери, з-поміж яких була й Україна.

17 грудня того ж року EDP було опубліковано 6-й щорічний звіт ODM Report 2020 стосовно розвитку сфери ВД. Україна вперше приймала участь у цьому дослідженні і одразу ж стала однією з країн зі стрімким розвитком сфери ВД отримавши 2180 балів із 2600 можливих та зайнявши 17-те місце в рейтингу. Рівень зрілості сфери ВД склав 84%, у той час як середній показник у Європі – 78% (Додаток А рисунок А1).

Уже у наступному 2021 році Україна піднялася з 17-го місця на 6-те в рейтингу ODM Report 2021 ставши однією з країн, що задають тренди у сфері ВД. Рівень зрілості сфери ВД виріс до 94%, що на 10% більше, ніж у минулому році, а середній рівень країн ЄС склав 81% (Додаток А рисунок А3).

Оцінювався розвиток сфери ВД країн за наступними показниками:

- державна політика: Україна – 640 бали (98%), середній показник країн ЄС (СП ЄС) – 87%;
- функціонал національного порталу відкритих даних: Україна – 614 балів (94%), СП ЄС – 83%;
- вплив ВД у різних сферах – економічна, соціальна, екологічна, державне урядування: Україна – 615 балів (95%), СП ЄС – 78%;
- якість опублікованих на порталі даних, їх відповідність до міжнародного DCAT-AP стандарту: Україна – 578 бали (89%), СП ЄС – 77%.

У Додатку А на рисунку А4 представлена інфографіка за кожним критерієм оцінювання про розвиток ВД в Україні у 2021 році.

У 2022 році Україна за розвитком сфери ВД зайняла 2-ге місце в рейтингу ODM Report 2022. Рівень зрілості сфери ВД склав 97%, а середній показник країн ЄС склав 79% (Додаток А рисунок А5).

Україна покращила свої показники за всіма пунктами оцінки:

- політика відкритих даних: Україна – 640 балів (100%), СП ЄС – 86%;
- функціонування порталу ВД: Україна – 617 балів (95%), СП ЄС – 83%;
- вплив ВД: Україна – 600 балів (100%), СП ЄС – 71%;
- якість ВД: Україна – 605 балів (93%), СП ЄС – 77%.

У Додатку А на рисунку А6 представлена інфографіка за кожним критерієм оцінювання про розвиток ВД в Україні у 2022 році.

Стрімкий розвиток сфери відкритих даних в Україні розпочався з 2015 року та залучився підтримкою міжнародних донорів, оскільки за допомогою ВД, через їх відкритість та прозорість інформації для всіх, набагато легше боротися з корупцією.

Україна здобула значних успіхів у відкритті суспільно необхідної інформації, яка є життєво важливою для держави та бізнесу. Бюджет громади, показники забруднення води та повітря, рух громадського транспорту, виплати лікарям, дороги, що внесені до переліку ремонтів, – ці та інші дані є у володінні держави та є корисними для громадян.

РОЗДІЛ 2. ОГЛЯД ТЕХНОЛОГІЙ І ДОСТУПУ ДО ВІДКРИТИХ ДАНИХ

2.1. Огляд та обґрунтування вибору технологій створення веб-додатків

«Веб-додаток – це клієнт-серверна програма, в якій клієнт взаємодіє з веб-сервером за допомогою браузера. Логіка веб-додатка розподілена між сервером та клієнтом, зберігання даних здійснюється переважно на сервері, обмін інформацією відбувається по мережі. Однією з переваг такого підходу є той факт, що клієнти не залежать від конкретної операційної системи користувача, тому веб-додатки є міжплатформними сервісами» [30].

Різниця між веб-сайтом і веб-додатком. Здавалося б, що веб-додаток і веб-сайт зовсім схожі поняття, але слід зазначити, що вони виконують різні функції і відрізняються компонентами розробки.

Детальніше про відмінності між веб-додатком і веб-сайтом наведено нижче [43]:

- веб-додаток може взаємодіяти з користувачем, тоді як веб-сайт відображає лише інформацію;
- веб-сайти служать лише метою доставки інформації у вигляді тексту, музики чи відео. Однак веб-додаток може взаємодіяти з користувачем і давати кілька результатів на основі операцій;
- веб-додаток має багато функцій та вирішує багато проблем, тоді як веб-сайт тільки відображає статичну сторінку, на якій відображається певна інформація;
- веб-додаток може бути як частиною веб-сайту, так і окремим додатком;
- веб-додаток здебільшого підключено до бази даних.

Технології розробки веб-додатків. HTML, CSS та JavaScript – це найпопулярніша зв'язка технологій для створення сайтів. Близько 90% усіх сайтів працює саме завдяки цьому набору технологій. Кожна з цих

технологій має різне призначення, цілі та функції. Але більшу цінність вони становлять, коли працюють разом, а не окремо.

«HTML (Hypertext Markup Language) – це мова гіпертекстової розмітки. Вона визначає зміст та структуру веб-контенту. Ця розмітка створюється за допомогою тегів, які складаються з імені елемента оточеного "<" та ">" символами. Наприклад, основний тег сторінки `html` пишеться в такий спосіб – `<html>`. Будь-яка сторінка в інтернеті складається з безлічі тегів. Кожен із цих тегів грає певну важливу роль» [8].

Для прикладу буде використано будинок, щоб краще показати залежність використовуваних технологій (HTML, CSS, JavaScript). У будинка є його екстер'єр, тобто те, з чого зроблено будинок. Теги на сторінці можна розглядати як невелику цеглу, за допомогою якої збудовано будинок. Важливо, щоб ці цеглини акуратно і красиво були розкладені, інакше будинок буде криво виглядати, а може і взагалі він буде непридатний для використання. Те ж саме стосується і веб-сторінок – при складанні структури сторінки важливо приділити особливу увагу тегам.

Кожна веб-сторінка в Інтернеті має загальну структуру (рис. 2.1), яка складається із обов'язкових тегів.

```
<!DOCTYPE html>
<html lang="en">
<head>
|   <title>Заголовок</title>
</head>
<body>
|   <h1>Заголовок першого рівня</h1>
|   <p>Абзац</p>
</body>
</html>
```

Рис. 2.1. Загальна структура веб-сторінки

Будь-яка веб-сторінка починається з `<!DOCTYPE html>`. Цей тег дає браузеру зрозуміти, що сторінка написана за допомогою `html` останньої, п'ятої, версії.

За ним іде парний тег `<html></html>`. Це основний тег сторінки, який є контейнером для всього вмісту веб-сторінки, тому він обов'язково повинен бути присутнім і містити в собі 2 основних теги, це `head` і `body`.

Всередині парного тегу `<head></head>` необхідно написати заголовок сторінки (тег `title`), який відображається у вкладці браузера. Також у контейнері `<head></head>` зазвичай знаходяться різні мета-теги для пошукових систем, підключення різних файлів до сторінки (наприклад, стилі) тощо. У цій секції міститься інформація, яка важлива для сторінки, для її правильного виконання браузером, але ця інформація не відображається користувачеві на сторінці.

Всередині тега `<body></body>` знаходиться все, що має відображатися на сторінці. Це будь-які з існуючих тегів, текст, картинки, елементи роботи з даними і так далі. Все, що бачить користувач на сторінках інтернету, завжди знаходиться в тезі `body`.

У наведеному на рисунку 2.1 прикладі в тезі `body` знаходяться 2 елементи – тег `h1` і `p`. Тег `<h1></h1>` означає заголовок на сторінці, а тег `<p></p>` – абзац. Кожен `html` тег має своє призначення. До того ж, усі елементи мають стандартне форматування браузера, а це означає, що розмір тексту в заголовку за замовчуванням буде більшим, ніж в абзаці. Із таких тегів і складається сторінка, яку користувач бачить у браузері. Однак без графічного оформлення ці елементи зовсім не презентабельні, тому для опису зовнішнього вигляду використовують `CSS`.

`CSS` (`Cascading Style Sheets`) – це каскадні таблиці стилів, набір параметрів форматування, який застосовується до елементів документа, щоб змінити їх зовнішній вигляд [5].

Знову приклад з будинком. Після будівництва будинок виглядає зовсім не презентабельно, тому, щоб надати гарного вигляду, його розфарбовують. Під'їзд пофарбований в один колір, балкони в інший тощо. Це і є графічне оформлення. Те ж саме відбувається й зі сторінкою: без стилів елементи

мають лише стандартне оформлення браузера, але за допомогою стилів можна змінити на сторінці розмір тексту, колір, шрифт і так далі.

На рисунку 2.2 зображено стилізацію тега h1. Щоб стилізувати елемент сторінки, потрібно вказати сам тег, або ім'я класу чи ідентифікатор заданий цьому тегові тощо. Далі у фігурних дужках описуються стилі вказаного елемента. У цьому блоці можна встановити будь-які з наявних стилів і вони будуть застосовані до елемента h1. У даному прикладі для тега h1 замінюється стандартний чорний колір тексту на червоний.



```
<style>
  h1 {
    color: red;
  }
</style>
```

Рис. 2.2. Стилiзацiя елементiв сторiнки

Після побудови та стилізації сторінки задається її функціонал за допомогою JavaScript.

JavaScript – це кросплатформна, об’єктно-орієнтована мова програмування, скорочено JS. Спочатку його створили для того, щоб «оживити» веб-застосунки та веб-сайти, тобто, щоб елементи інтерфейсу (спливаючі вікна, анімації, кнопки тощо) реагували на дії користувачів. Однак зараз цю мову програмування застосовують не тільки для «оживлення» сторінок, але і на стороні сервера, для створення мобільних програм, веб-сервісів і так далі [10].

Повертаючись до прикладу з будинком, то JavaScript це ліфт, який доставляє користувачів на потрібний поверх. Людина заходить в ліфт і їй потрібно потрапити на конкретний поверх, вона натискає кнопку поверху і далі ліфт автоматично доставляє людину на потрібний поверх. Так само і на сторінці, користувач натискає кнопку, а JavaScript виконує потрібну дію. Звичайно, людина завжди може піднятися сходами, так само як і сайт може працювати без JavaScript, але, саме ліфт робить будинок кращим та

комфортнішим у користуванні, так само як і JavaScript покращує веб-сторінку.

JavaScript-скрипти дозволяють автоматично заповнювати форми, переформатовувати сторінки, приховувати небажаний вміст та вбудовувати бажаний для відображення певного вмісту у потрібний момент під час користування веб-сайтом чи веб-додатком, змінювати поведінку клієнтської частини веб-застосунків, додавати елементи керування на сторінку тощо.

Технології, що спрощують розробку та підтримку веб-додатків. Сьогодні існує багато бібліотек та фреймворків, які надають розробникам основу для написання додатків. Вони містять заздалегідь визначені та реалізовані базові програмні модулі, класи чи функції на основі яких реалізуються специфічні для проекту компоненти. Тим самим досягається не лише висока швидкість розробки, а й велика продуктивність та надійність рішень.

Використання цих технологій надає такі основні плюси:

- Рентабельність. Більшість бібліотек та фреймворків безкоштовні та мають відкритий вихідний код для використання. Оскільки це значно прискорює створення програми, то відповідно і зменшує ціну самого веб-додатка.
- Продуктивність. Значно покращується продуктивність і все тому, що вони максимально оптимізують процес. Простіше використовувати одну із існуючих технологій, яка оптимізована та має добре структуровані шаблони, ніж писати сотні рядків коду.
- Безпека. Трендові JavaScript-фреймворки та бібліотеки можуть похвалитися не лише підтримкою великої спільноти на GitHub, але й залізною системою безпеки.

Сьогодні найбільш популярними технологіями для JavaScript, що полегшують розробку та об'єднання різних компонентів програмного проекту є React.js, Angular.js та Vue.js.

«Основною їх складовою є компоненти. Як правило, вони отримують вхідні дані та змінюють поведінку програми на їх основі. Візуально це проявляється у вигляді зміни інтерфейсу користувача.

Особливості компонент:

- компоненти розділені: частина відповідає за інтерфейс користувача, а частина – за поведінку;
- одна і та ж сама частина коду може відповідати як за створення елемента інтерфейсу, так і за його поведінку;
- інтерфейс користувача та поведінка також можуть бути комбінованими» [19, с. 446].

React і Vue багато в чому схожі, вони обидва використовують Virtual DOM, надають компонентну структуру, сторонні питання виносяться до додаткових бібліотек. Через те, що React і Vue здебільшого зосереджені на створенні інтерфейсу користувача, а Angular – на додатках, останній складніший для початківців. Робота з React та Vue стає важчою у великих додатках.

React перевершує Vue у різноманітності екосистеми та достатку доступних користувацьких засобів відтворення елементів на сторінці. У React абсолютно все – це JavaScript, тоді як Vue більшою мірою ґрунтується на класичних веб-технологіях, наприклад, будь-який валідний HTML також буде валідним шаблоном Vue.

У порівнянні з Angular обидва інші фреймворки – і React, і Vue – виявляють більшу гнучкість. Але в цьому і є перевага Angular: не доводиться вигадувати свою реалізацію, потрібно лише дотримуватися уже існуючих правил, що спрощує роботу над розробкою великих додатків.

2.2. Поняття API – використання, розробка та інтеграція в АІС

Поза наші очі програми, додатки та веб-сервіси безперервно спілкуються між собою за допомогою API.

API (Application Programming Interface – програмний інтерфейс додатку, або інтерфейс програмування додатків) – спеціальний протокол для взаємодії комп’ютерних програм, який дозволяє використовувати функції, класи, процедури, константи та структуру однієї програми всередині іншої, запитувати дані з іншої системи [23].

Інакше кажучи, інтерфейс виступає у якості «прошарку» між додатками А та Б, на якому відбуваються процеси, що дозволяють цим додаткам здійснювати обмін інформацією між собою та виконувати функції, пов’язані з обома сторонами, приховуючи «внутрішню будову» програм (рис. 2.3). Щоразу, коли ми використовуємо програму, наприклад Facebook, надсилаємо миттєве повідомлення або перевіряємо погоду на своєму смартфоні, ми використовуємо API.

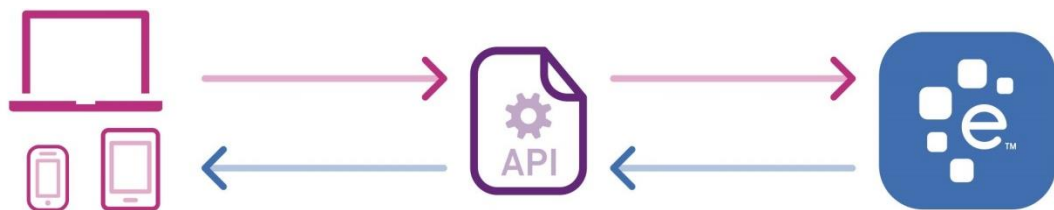


Рис. 2.3. Схема простого API [24]

Таким чином, API дозволяє одній програмі використовувати дані та функції інших програм, служб або операційних систем, водночас зберігаючи їх безпеку та контроль. Розробники можуть розширювати функціональність свого продукту та пов’язувати його з чужими розробками. Тобто, якщо розробник хоче створити додаток для електронної комерції, він може зосередитись конкретно на розробці свого продукту, а для виконання у ній платежів, білінгу, автентифікації тощо, може використовувати уже готові API різних програм та не витрачати на це зайвий час.

Найбільш використовуваними є API за наступними напрямками (рис. 2.4):

- API для роботи з документами, що завантажені у браузер. Явним прикладом є DOM (Document Object Model) API, який надає можливість

працювати з HTML і CSS, а саме: створювати, видаляти та змінювати HTML, динамічно змінювати вид сторінки та інше. Кожне спливаюче вікно на сторінці чи контент, що з'являється «на ходу» – усе це завдяки DOM [9].

- API, що отримують дані від сервера, часто застосовуються, щоб оновити невеликі частини веб-сторінки. Завдяки такому підходу підвищується продуктивність та швидкодія сайту, оскільки не потрібно перезавантажувати повністю всю сторінку, коли необхідно лише оновити список товарів або нових доступних історій [9].

- API для роботи з графікою також широко підтримуються браузерами.

- Аудіо та Відео API для роботи з мультимедіа. Наприклад, створити власний інтерфейс користувача для програвання аудіо/відео, виведення на екран субтитрів, записувати відео з веб-камери для обробки в canvas або для передачі на інший комп'ютер у відео-конференції тощо [9].

- API пристроїв – в основному, сюди відносяться API, що обробляють та зчитують дані з сучасних пристроїв зручним для роботи веб-додатків способом (наприклад, API Геолокації) [9].

- API зберігання даних на стороні клієнта набувають все більшого поширення у веб-браузерах, Це є дуже корисним, якщо потрібно створити додаток, який буде зберігати свій стан між перезавантаженнями сторінки чи працювати, навіть коли пристрій не знаходиться в мережі [9].



Рис. 2.4. API як засіб інтеграції додатків [9]

Список опцій, доступних під час роботи з тим чи іншим API, залежить від розробників. Виділяють три основні пункти, які описують роботу інтерфейсу та методи взаємодії з ним:

- Процес, що може бути виконаний програмою за допомогою API.
- Дані, котрі необхідно надіслати інтерфейсу для виконання функції.
- Дані, які програма отримає після обробки за допомогою API.

Фактично, маємо приховану функцію або набір прихованих функцій, що здійснюють обробку та видачу даних (рис. 2.5). Даний процес сховано шляхом інкапсуляції.

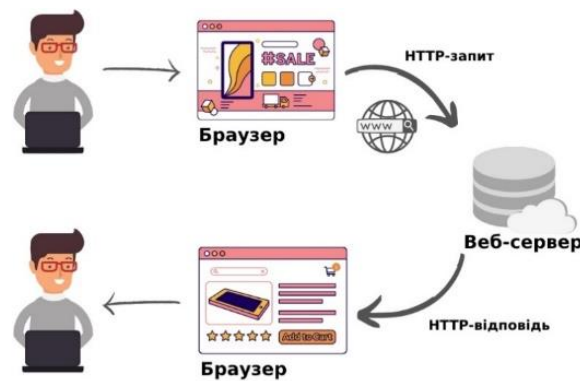


Рис. 2.5. Приклад HTTP-запиту до сервера

Виділяють чотири основні типи API, що використовуються найчастіше: відкриті, партнерські, приватні, композитні.

Відкриті API або, як їх ще називають, публічні чи внутрішні API, є загальнодоступними для всіх. Як наслідок, відкриті API зазвичай мають відносно низькі заходи автентифікації та авторизації і часто обмежені в ресурсах, якими вони спільно користуються.

Партнерські API, це спеціальні API, що надаються компаніями лише їхнім партнерам, хто має з ними ділові стосунки. Доступ обмежено авторизованими клієнтами з офіційними ліцензіями, тому заходи безпеки, як правило, сильніші з партнерськими API, ніж із загальнодоступними API.

Приватні або внутрішні API використовуються всередині компанії, якій вони належать і призначені для спільного використання ресурсів

розробниками, що працюють у ній, а також для оптимізації передачі даних між командами та системами. Використання API для внутрішньої передачі даних вважається більш ефективним, безпечним і доступним для відстеження.

Композитні або складні API об'єднують кілька API, що дозволяє розробникам об'єднувати виклики або запити й отримувати єдину відповідь від різних кінцевих точок одного API або програм, служб чи джерел даних.

2.3. Технологія REST – використання, розробка та інтеграція в АІС

REST (REpresentational State Transfer) – архітектурний стиль проектування API та взаємодії компонентів розподіленого додатка в мережі. Іншими словами, REST – це набір правил того, як розробнику організувати написання коду серверної програми, щоб усі системи легко обмінювалися даними і програму можна було масштабувати. Сервіси REST API взаємодіють за протоколом HTTP.

REST API використовується всюди, де користувачеві веб-додатку чи веб-сайту потрібно надавати дані з сервера (рис. 2.6).

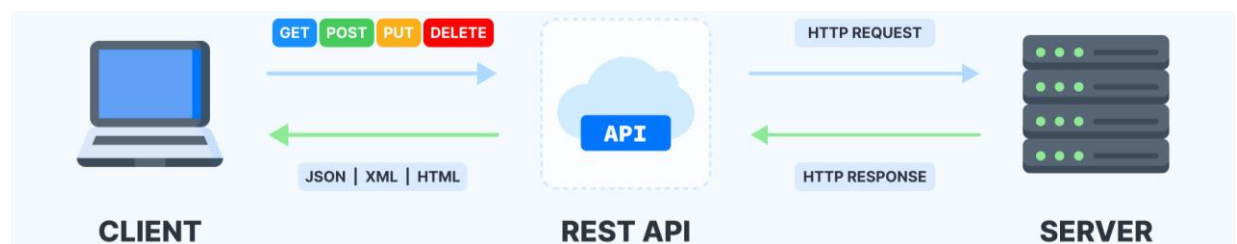


Рис. 2.6. Модель REST API [28]

У REST API виділяють три основні компоненти: клієнт, сервер та ресурс.

Клієнт – веб-додаток або сайт, запущений на пристрої користувача, який ініціює комунікацію з сервером.

Сервер надає API в якості доступу до своїх даних та функцій.

Ресурс – це будь-який контент, такий як текст, картинка, відео тощо, що може бути переданий клієнту сервером.

Якщо під час розробки API дотримуються 6 архітектурних обмежень, що описують, як саме сервер має обробляти запити та відповідати на них, то API є RESTful. Сюди входять наступні обмеження, необхідні для досягнення оптимізації системи:

1. клієнт-серверна архітектура;
2. stateless (без збереження стану);
3. кешування;
4. одноманітність інтерфейсу;
5. багаторівнева система;
6. код на вимогу.

Клієнт-серверна архітектура. Концепція клієнт-серверної архітектури полягає у тому, щоб розділити певні зони відповідальності між клієнтом та сервером (рис. 2.7).

Наприклад, система розділяється на клієнта (мобільний додаток), який відповідає за інтерфейс, відображення інформації та реалізацію лише функціональної взаємодії з сервером, а також на сервер, що реалізує у собі логіку зберігання та обробки даних, складні взаємодії з суміжними системами та інше.



Рис. 2.7. Клієнт-серверна архітектура

Також у сервера може бути наявна база даних (рис. 2.8). У цьому випадку пара «сервер – БД» також буде мати клієнт-серверну архітектуру. Тільки тепер у якості серверу буде виступати БД, а сам сервер буде клієнтом.

Клієнт-серверна архітектура надає:

- масштабованість. Наприклад, сервер витримує певне навантаження і до нього можна підключити певну кількість клієнтів. У випадку, якщо сервер не витримує навантаження чи потрібно збільшити кількість клієнтів, що можуть бути одночасно підключеними, то необхідно лише збільшити кількість серверів, між якими вони будуть розподілятися.

- простота підтримки. Якщо потрібно змінити логіку обробки інформації на сервері, то такі зміни виконуються на стороні сервера і їх не потрібно вносити до кожного клієнта.

Звісно, є й мінуси. У випадку з клієнт-серверною архітектурою існує єдина точка відмови у вигляді сервера. Якщо відмовить один єдиний сервер, а додаткових серверів немає, то система буде деякий час непрацездатною, поки проблеми з сервером не будуть вирішені.

Також можливе збільшення навантаження, оскільки частина логіки винесена з клієнта на сервер. Оскільки на стороні клієнта буде здійснюватися менше самостійних дій, то, відповідно, зросте кількість запитів між клієнтом та сервером.

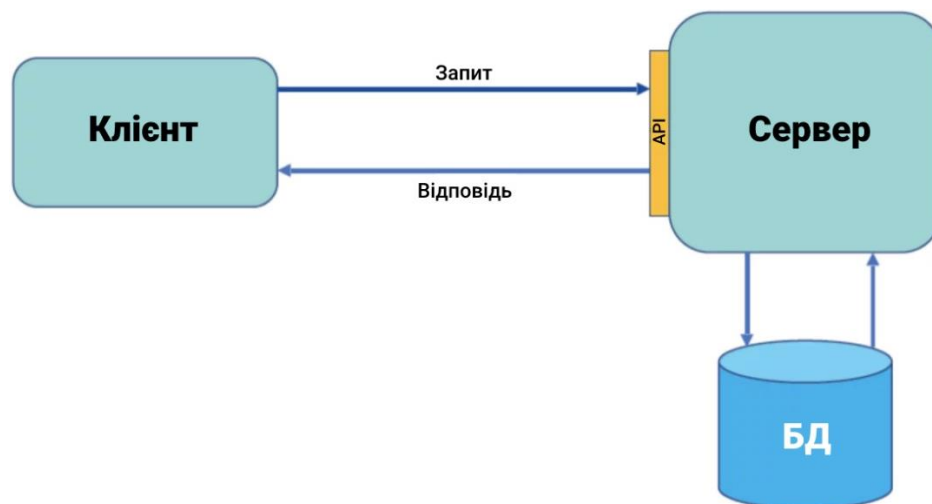


Рис. 2.8. Трирівнева архітектура

Stateless. Суть даного принципу полягає в тому, що сервер не повинен зберігати інформацію про поточну сесію з клієнтом. Він повинен у кожному запиті отримувати всю інформацію для обробки, тобто інформація про поточну сесію повинна повністю зберігатися у клієнта.

Переваги даного принципу:

- масштабованість сервера;
- зменшення часу обробки запиту;
- простота підтримки;
- можливість використовувати кешування.

Завдяки тому, що серверу не потрібно зберігати інформацію про стан клієнта, звільняються додаткові ресурси сервера, завдяки чому він може обслуговувати більшу кількість клієнтів одночасно.

Також, даний підхід може утворити наступні проблеми:

- ускладнення логіки клієнта (саме на стороні клієнта потрібно зберігати всю інформацію про стан, про допустимі дії, про неприпустимі дії та подібні речі).
- збільшення навантаження на мережу (щоразу потрібно передавати всю інформацію, весь контекст. Таким чином, більше інформації передається мережею).

Кешування. Даний принцип говорить про те, що у даних для запиту потрібно вказувати чи можна кешувати дані (зберігати у спеціальному буфері для частих запитів). Кешування є потрібним і корисним, коли сервер часто запитують про надання однакової інформації. Наприклад, кешування активно використовується на веб-ресурсах, до яких відбувається багато звернень (сайти новин чи соціальні мережі).

Плюси кешування:

- Зменшується кількість мережевих взаємодій.
- Зменшення навантаження на системи (не навантажуються додатковими запитами).

У деяких випадках однакових звернень буде не дуже багато. У такому разі кешувати дані немає сенсу.

До того ж потрібно враховувати, якщо надаються певні дані, що вже зберігалися раніше, то важливо пам'ятати, що такі дані вже можуть бути застарілими. У деяких випадках це може бути прийнятним, у той час як в

інших – абсолютно неприпустимим. Відповідно, чи варто використовувати кешування – завжди потрібно обмірковувати на конкретному прикладі.

Одноманітність інтерфейсу. Ще одне з обмежень REST, згідно з яким усі дані повинні запитуватись через одну URL-адресу стандартними протоколами, наприклад, HTTP. Це спрощує архітектуру сайту або програми та робить взаємодію з сервером зрозумілішим.

Завдяки даному принципу клієнтський додаток отримує набагато більше інформації та свободи дій, логіка клієнта стає більш гнучкою.

Проте є і зворотний бік медалі – хоч логіка на стороні клієнта і стає гнучкішою, але вона сильно ускладнюється. Також може ускладнюватись логіка на сервері для правильного формулювання відповідей на запити. Фактично відповідальність за дії, які робить клієнт, передається на його ж бік. Послаблюється контроль валідності здійснюваних операцій на стороні сервера.

Багаторівнева система. У багаторівневій системі посередники, такі як проксі-сервери, роутери, балансувальники можуть розміщуватися між клієнтом і сервером, використовуючи одноманітний інтерфейс мережі. І те, яким шляхом запит проходить від клієнта до сервера, ми часто не можемо знати.

Концепція багаторівневої архітектури полягає у тому, що ні клієнт, ні сервер нічого не мають знати про те, як відбувається ланцюжок викликів далі своїх прямих сусідів.

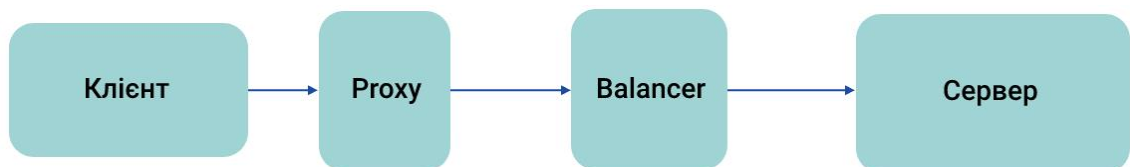


Рис. 2.9. Багаторівнева архітектура

На схемі, яка представлена на рисунку 2.9, знання балансувальника про учасників саме цього ланцюжка викликів повинні закінчуватися проху-сервером зліва та сервером праворуч. Про клієнта він уже нічого не знає.

Якщо змінюється поведінка проху-сервера (балансувальника, роутера тощо), це не повинно спричинити зміни для клієнтської програми або для сервера. Це дає змогу змінювати загальну архітектуру без доопрацювань на стороні клієнта чи сервера.

Однією з переваг багаторівневої системи є те, що посередники можуть перехоплювати трафік клієнт-сервера для певних цілей, наприклад, для кешування.

Мінуси:

- збільшується кількість учасників та викликів у мережі і, відповідно, збільшується мережеве навантаження, ніж якби відбувалося з'єднання один раз від клієнта до сервера безпосередньо;
- збільшується час отримання відповіді через появу додаткових учасників.

Код на вимогу. REST дозволяє розширювати по мірі необхідності функціональність клієнта завантажуючи необхідний код з сервера у вигляді аплетів чи скриптів та виконуючи його (рис. 2.10).

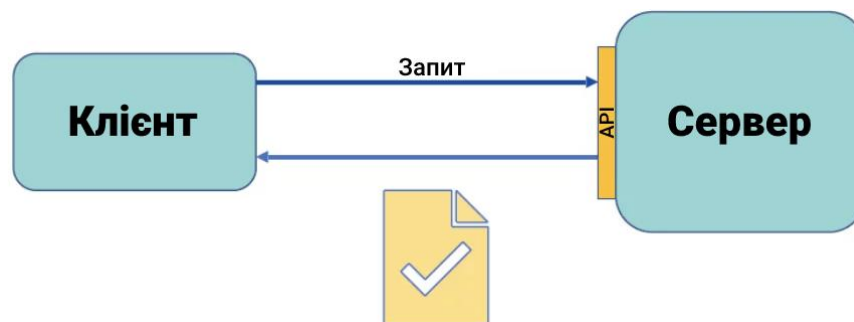


Рис. 2.10. Модель архітектури, що реалізує принцип «Код за запитом»

Наприклад, у якості клієнта буде звичайний веб-браузер. Клієнт надсилає певний запит і чекає відповіді – сторінку з певним інтерактивом (наприклад, повинен з'являтися феєрверк у тому місці, де користувач клацне кнопкою мишки).

Код на вимогу дозволяє клієнтові стати більш гнучким. Якщо знадобиться змінити колір феєрверку, то не потрібно вносити змін на стороні

клієнта, достатньо зробити це на сервері, а потім передавати клієнту. Приклад такої мови – JavaScript.

Принцип роботи REST API. REST API зазвичай реалізують за допомогою протоколу передачі гіпертексту HTTP, який інформує сервер про те, що потрібно зробити з ресурсом: створити, оновити, отримати (прочитати), видалити записи в ньому. Виділяють чотири методи (рис. 2.11), які описують, що потрібно робити з ресурсом [29]:

- POST – створити нові дані на сервері;
- GET – отримати дані;
- PUT – оновити існуючі дані ресурсу;
- DELETE – видалити дані з ресурсу.

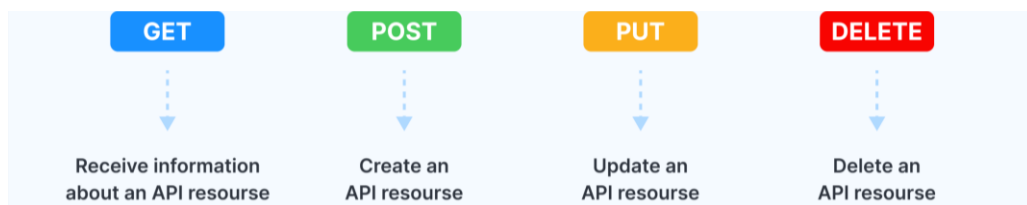


Рис. 2.11. HTTP-методи [28]

HTTP структура запиту (request) виглядає наступним чином [2]:

- HTTP-метод. Вказується необхідний метод (GET, POST, PUT, DELETE) залежно від операції, яку клієнту потрібно виконати з ресурсом. Найбільш використовуваними є метод GET, коли клієнт хоче отримати якісь дані з сервера та метод POST, щоб відправити дані (значення з HTML-форми) на сервер для їх збереження. Інші методи також є необхідними в певних випадках, але використовуються не так часто.

- Шлях до ресурсу. Вказується URL-адреса потрібного ресурсу – стандартизований спосіб запису адреси ресурсу в мережі Інтернет.
- Версія HTTP-протоколу.
- Заголовки, що надають серверу додаткову інформацію (характеризують тіло повідомлення, параметри передачі та містять інші додаткові відомості).

– Тіло для деяких методів, таких як POST, що містить дані, які відправляється на сервер.

На запит завжди є повідомлення з відповіддю (response) і для цієї відповіді HTTP також визначає свою структуру [2]:

- Версію HTTP-протоколу.
- HTTP код стану, завдяки якому клієнт розуміє про успішність запиту або причину невдачі і залежно від результату розуміє, що потрібно робити далі.
- Повідомлення стану – коротко описується код стану.
- HTTP заголовки, подібно до заголовків у запитах.
- Тіло ресурсу, який пересилається.

Для створення веб-додатку було взято API державного веб-порталу OpenBudget, за допомогою якого можна отримати дані у форматі JSON про фінансові надходження до державного або місцевого бюджету. Для цього потрібно задати необхідні параметри перед відправкою GET-запиту, такі як рік, місяць «з якого» та місяць «по який» необхідно отримати дані, а також вид бюджету. На рисунку 2.12 зображено URL-адресу, на яку буде відправлятися запит для отримання даних за вказаними параметрами.

<https://openbudget.gov.ua/api/reports/income/details/JSON?fundType=TOTAL&budgetType=NATIONAL&year=2022&monthFrom=1&monthTo=11>

budgetType - вид бюджету (державний або місцевий)

year - рік, за який потрібно надати дані

monthFrom - нижня межа періоду (місяць), за який потрібно надати дані

monthTo - верхня межа періоду (місяць), за який потрібно надати дані

Рис. 2.12. URL-адреса із заданими параметрами для отримання з OpenBudget даних про державний або місцевий бюджет

Також можна отримати дані про надходження до бюджету конкретної області, відправивши GET-запит на URL-адресу зображену на рисунку 2.13.

<https://openbudget.gov.ua/api/localBudgets/incomesLocal/JSON?fundType=TOTAL&codeBudget='260000000000'&year=2022&periodFrom=1&periodTo=11>

codeBudget - код місцевого бюджету (області)

year - рік, за який потрібно надати дані

periodFrom - нижня межа періоду (місяць), за який потрібно надати дані

periodTo - верхня межа періоду (місяць), за який потрібно надати дані

Рис. 2.13. URL-адреса із заданими параметрами для отримання з

OpenBudget даних про місцевий бюджет конкретної області

Отже, для створення різних програм, додатків чи веб-сервісів та отримання даних від сервера чи іншого додатку, використовуються різні технології, кожна з яких відповідає за реалізацію певного функціоналу.

РОЗДІЛ 3. РОЗРОБКА ВЕБ-ДОДАТКУ ДЛЯ АНАЛІЗУ ВІДКРИТИХ ДАНИХ ПОДАТКОВИХ НАДХОДЖЕНЬ

3.1. Основна мета та функції веб-додатку

Основна мета веб-додатку полягає у тому, щоб надати можливість отримувати та аналізувати відкриті дані податкових надходжень до бюджету України, представляючи їх у вигляді таблиці з можливістю фільтрувати відображувані дані необхідні для аналізу користувачеві, здійснювати пошук даних по таблиці та можливістю зберігати відредаговані дані у форматах CSV, XML, XLSX, HTML, JSON для подальшого їх використання. Представлення даних у вигляді діаграм та графіків для їх кращого сприйняття та аналізу.

Отже, виходячи з мети додатку, він має виконувати такі основні функції:

- *Завантаження та обробка даних, їх відображення.* Вказавши рік та з якого і по який місяць потрібно надати дані про надходження до бюджету, за допомогою GET-запиту через API будуть отримані дані у форматі json. Після цього вони будуть оброблені та приведені до вигляду необхідного для виконання подальших завдань та відображені у вигляді таблиці.

- *Налаштування відображуваних даних.* Оскільки дані відображуються у вигляді таблиці, то для їх аналізу було додано можливість відображати та приховувати необхідні стовпці та рядки даних. Також додано поле пошуку, яке дозволяє швидко знайти необхідні користувачеві дані у таблиці.

- *Завантаження відфільтрованих у таблиці даних на ПК.* Після аналізу та фільтрації даних у таблиці, їх можна зберегти у таких форматах, як CSV, XML, XLSX, JSON і навіть у вигляді HTML-файлу, який буде містити дані у звичайній html-таблиці, що дасть змогу вмонтувати її на інший сайт, якщо це потрібно.

– *Представлення даних у вигляді діаграм.* Для кращого сприйняття та аналізу даних про фінансові надходження до бюджету у роботі для прикладу формуються наступні діаграми:

- виконання бюджету;
- виконання бюджету за видами надходжень;
- структура бюджету.

Функціонально є можливість формувати й інші графіки податкових надходжень.

3.2. Формування та програмування компонентів веб-додатку

У другому розділі вже були описані технології створення веб-додатків. Отже, для розробки веб-додатку для аналізу відкритих даних податкових надходжень були використані як загальні технології: HTML – виступає у якості скелету додатку, CSS – зовнішній вигляд додатку та JS – відповідає за його функціонал, так і допоміжні, що спрощують розробку і допомагають виконати деякі поставлені завдання: React, ChartJS, Axios, SheetJS CE, FileSaver.js.

SheetJS Community Edition пропонує рішення з відкритим вихідним кодом для отримання корисних даних майже з будь-якої складної електронної таблиці та створення нових електронних таблиць, які працюватимуть як із застарілим, так і з сучасним програмним забезпеченням [18].

FileSaver.js – це рішення для збереження файлів на стороні клієнта, яке ідеально підходить для веб-програм, які генерують файли на клієнті [7].

Chart.js – популярна бібліотека, яку можна використовувати для створення різних типів діаграм. Вона дозволяє змішувати різні типи діаграм і з легкістю відображати дані за датою, часом, логарифмічною чи спеціальною шкалою. Бібліотека також містить готові анімації, які можна застосувати під час зміни даних або оновлення кольорів [4].

Axios – це бібліотека з відкритим вихідним кодом, що дає змогу робити HTTP-запити для отримання та відображення даних з API. Вона надає наступні методи для роботи з API: GET, POST, PUT та DELETE [27].

«React – це декларативна, ефективна і гнучка JavaScript-бібліотека для створення користувацьких інтерфейсів» [33]. Вона дозволяє збирати складний UI з маленьких ізольованих шматочків коду, які називаються компонентами.

Компонент – функція, яка приймає на вході якісь властивості (як властивості html тегу) і повертає JSX-розмітку [33]. Компонента не може напряду змінювати властивості, що їй передані, але може їх змінювати через callback функції. Назва компоненту починається завжди з великої літери.

JSX – це спеціальне розширення файлів (яке зазвичай мають компоненти), що дозволяє писати HTML-код у JS-файлі, що значно спрощує керування логікою та інтерфейсом програми, яка розробляється [33].

Використовуючи можливість React поділяти веб-додаток на окремі компоненти, які будуть виконувати свої окремі завдання, створений проект також було поділено на наступні компоненти, у які передаються лише необхідні дані для їх роботи:

- Upload – компонент для отримання даних, у якому вказується рік, місяці з якого та по який необхідно отримати дані, вид бюджету і відправляється запит через API для отримання цих даних.
- Table – відповідає за створення таблиці.
- TableSettings – блок з налаштуваннями таблиці, що містить кнопки «Вибрати всі», «Очистити всі» та «Підтвердити», які є спільними як для налаштування стовпців, так і рядків. За відображення списку налаштувань рядків чи стовпців відповідають компоненти TableRowsSettings та TableColumnsSettings.
- TableRowsSettings – відображає список рядків для налаштування.
- TableColumnsSettings – відображає список стовпців для налаштування.
- Search – компонент, що відповідає за функціонування поля пошуку.

- Download – відповідає за збереження на ПК відфільтрованих у таблиці даних у одному з запропонованих форматів.
- Infographics – відповідає за усі відображувані для аналізу діаграми додатку.
- ChartTax – побудова та функціонал діаграми «Структура бюджету».
- PlanDone – побудова діаграми «Виконання бюджету».
- PlanDoneByTypes – побудова діаграми «Виконання бюджету за видами надходжень».

Також веб-додаток буде поділено на три сторінки «Головна», «Вибірка даних» та «Діаграми».

3.3. Розробка інтерфейсу та відображення даних

Отримання даних. Робота з API. Додаток розроблявся на основі відкритих даних про надходження до бюджету України, що взяті з державного веб-порталу Open budget. Для отримання даних здійснюється GET-запит до API веб-порталу, у якому потрібно буде передати рік, місяць «з якого» та місяць «по який» необхідно отримати дані, а також вид бюджету.

Для реалізації даного функціоналу, були створені поля зі списком, який з'являється при натисканні на поле, де можна вибрати доступний вид бюджету, рік та період для отримання даних (рис. 3.1). Отримати дані можна починаючи з 2018 року. За замовчуванням встановлюється поточний рік і період вказується від першого місяця року і до останнього місяця за який уже відомі дані на момент використання додатку. Поточний рік та останній місяць, за який відомі дані, визначаються автоматично після запуску веб-додатку.

Коли задано необхідні параметри, щоб отримати дані для аналізу, потрібно натиснути кнопку «Отримати дані», після чого через API дані будуть отримані, оброблені та відображені.

Рис. 3.1. Компонент Upload

Обробка вхідних даних. Дані з порталу завантажуються у форматі JSON, який складається з масиву об'єктів даних, що мають структуру зображену у Додатку Б на рисунку Б1.

Доходи бюджету класифікуються за розділами. Кожен розділ поділяється на рівні, які в свою чергу також можуть поділятися на рівні. Поділ на рівні здійснюється на основі кодів бюджетної класифікації доходів, який мають всі надходження (Додаток В рисунок В1).

Для розробки частини функціоналу додатку, такої як редагування відображуваних рядків даних чи подання даних у вигляді діаграми, знадобляться дані у вигляді подібного «дерева» з рівнями. Щоб привести дані до подібного виду, була створена рекурсивна функція, яка перетворює масив даних на об'єкт даних з рівнями, у якому в якості властивості об'єкту вказується бюджетний код класифікації фінансових надходжень, а значенням для нього задається об'єкт, що містить необхідну у подальшому для розробки додатку інформацію про даний вид надходжень, а також коди, що являються підрівнями даного рівня, які у свою чергу містять також всю необхідну інформацію про себе та список кодів, які виступають їхніми підрівнями (Додаток Б рисунок Б2).

Як було показано у Додатку Б на рисунку Б1, дані надходять у вигляді масива об'єктів. Кожен із них у таблиці буде виступати у якості рядка. У свою чергу кожен об'єкт містить дані (ключ та значення), які будуть відображатися у вигляді стовпців (рис. 3.2).

```
{
  "fundType": "TOTAL",
  "incomeCode": "10000000",
  "incomeCodeName": "Податкові надходження",
  "yearBeginPlan": 1204081760600,
  "yearCorrectionPlan": 1192488540300,
  "periodCorrectionPlan": null,
  "periodDone": 603290190872.11,
  "deviationToPeriodCorrectionPlan": null,
  "percentDoneToPeriodCorrectionPlan": null,
  "percentDoneToYearCorrectionPlan": 50.59
},
{
  "fundType": "TOTAL",
  "incomeCode": "11000000",
  "incomeCodeName": "Податки на доходи, податки на прибуток, податки на збільшення ринкової вартості",
  "yearBeginPlan": 319843711600,
  "yearCorrectionPlan": 308250491300,
  "periodCorrectionPlan": null,
  "periodDone": 178018523680.86,
  "deviationToPeriodCorrectionPlan": null,
  "percentDoneToPeriodCorrectionPlan": null,
  "percentDoneToYearCorrectionPlan": 57.75
},
}
```



fundType	incomeCode	incomeCodeName	yearBeginPlan	yearCorrectionPlan	periodCorrectionPlan	periodDone	deviationToPeriodCorrectionPlan	percentDoneToPeriodCorrectionPlan	percentDoneToYearCorrectionPlan
TOTAL	10000000	Податкові надходження	1204081760600	1192488540300		603290190872.11			50.59
TOTAL	11000000	Податки на доходи, податки на прибуток, податки на збільшення ринкової вартості	319843711600	308250491300		178018523680.86			57.75

Рис. 3.2. Представлення об'єктів даних та їх ключів із значеннями у вигляді рядків та стовпців

Створення таблиці. Після отримання даних через API та після їх обробки, відбувається побудова таблиці, яка здійснюється наступним чином (рис. 3.3):

1. За допомогою функції `filter` перебираються всі об'єкти з масиву даних і кожен об'єкт перевіряється на те, чи є його код бюджетної класифікації у змінній `showFields.rows`, яка містить коди лише тих об'єктів даних, які потрібно відобразити. Якщо об'єкт проходить перевірку, то за допомогою функції `map` він перетворюється на рядок таблиці (тег `<tr></tr>`).

2. Кожен рядок таблиці складається зі стовпців. Тому далі перебираються всі властивості об'єкту даних і перевіряються на їх наявність у змінній `showFields.columns`, після чого властивості, що пройшли перевірку, перетворюються за допомогою функції `map` на стовпці таблиці (тег `<td></td>`).

```

{searchedData.filter(item => showFields.rows.has(item['incomeCode'])).map((item, index) => {
  return <>
    <tr className="table__row" key={`row_${item}_${index}`}>
      {columnNames.filter((item, index) => showFields.columns.has(index)).map((field, index) => {
        return <td className="table__column" key={`column_${field}_${index}`}>{Object.keys(item).includes(field) ? item[field] : null}</td>
      })}
    </tr>
  </>
})}

```

Рис. 3.3. Код побудови рядків та стовпців таблиці

Створення налаштувань. Не всі поля даних можуть бути потрібні для аналізу. Наприклад, відповідно до потреб користувача для аналізу даних потрібні будуть лише дані з 3 стовпців із 10. Тому було додано можливість налаштовувати таблицю, а саме, дати змогу користувачеві вибрати необхідні стовпці та рядки даних для аналізу і відображати тільки їх у таблиці.

З правої сторони від таблиці знаходяться її налаштування, що мають два режими: налаштування рядків та налаштування стовпців. За замовчуванням встановлений режим редагування стовпців.

Відповідає за перемикання режимів змінна `typeSettings` типу `Object`, яка має дві властивості з булевими значеннями `true` або `false` (рис. 3.4). Цими властивостями є `columns` та `rows`. Якщо для `columns` встановлено `true`, то для `rows` буде встановлено `false` і, відповідно, буде відображатися режим редагування стовпців. І навпаки, якщо для `columns` встановлено `false`, то для `rows` буде `true` і відображатиметься режим для редагування рядків. Одночасно для обох властивостей не може бути встановлено `true` або `false`.

```

let [typeSettings, setTypeSettings] = React.useState({
  columns: true,
  rows: false,
})

```

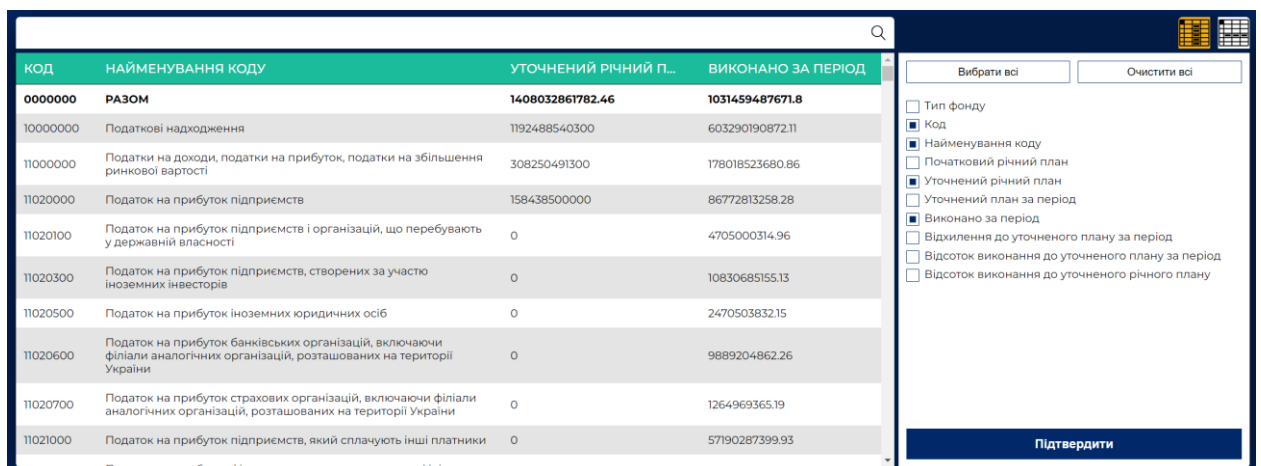
Рис. 3.4. Змінна, що відповідає за відображення типу налаштувань

Налаштування колонок. Тут відображається список всіх колонок таблиці (їх найменування). Біля кожного елементу списку є `checkbox` і якщо він активний, то стовпець буде відображений (рис. 3.5). При натисканні на найменування буде змінюватися стан `checkbox` між активний / неактивний. За

замовчуванням всі елементи списку мають активний checkbox, а отже всі стовпці відображаються.

Для зручності також були додані кнопки «Вибрати всі» та «Очистити всі», де перша зробить активними всі елементи списку, а інша – неактивними.

Щоб застосувати налаштування до таблиці, потрібно натиснути кнопку «Підтвердити».



КОД	НАЙМЕНУВАННЯ КОДУ	УТОЧНЕНИЙ РІЧНИЙ П...	ВИКОНАНО ЗА ПЕРІОД
0000000	РАЗОМ	1408032861782.46	1031459487671.8
10000000	Податкові надходження	1192488540300	603290190872.11
11000000	Податки на доходи, податки на прибуток, податки на збільшення ринкової вартості	308250491300	178018523680.86
11020000	Податок на прибуток підприємств	158438500000	86772813258.28
11020100	Податок на прибуток підприємств і організацій, що перебувають у державній власності	0	4705000314.96
11020300	Податок на прибуток підприємств, створених за участю іноземних інвесторів	0	10830685155.13
11020500	Податок на прибуток іноземних юридичних осіб	0	2470503832.15
11020600	Податок на прибуток банківських організацій, включаючи філіали аналогічних організацій, розташованих на території України	0	9889204862.26
11020700	Податок на прибуток страхових організацій, включаючи філіали аналогічних організацій, розташованих на території України	0	1264969365.19
11021000	Податок на прибуток підприємств, який сплачують інші платники	0	57190287399.93

- ☐ Тип фонду
- ☒ Код
- ☒ Найменування коду
- ☐ Початковий річний план
- ☒ Уточнений річний план
- ☐ Уточнений план за період
- ☒ Виконано за період
- ☐ Відхилення до уточненого плану за період
- ☐ Відсоток виконання до уточненого плану за період
- ☐ Відсоток виконання до уточненого річного плану

Рис. 3.5. Налаштування відображуваних стовпців таблиці

Налаштування рядків. Як було згадано раніше, надходження поділяються за кодом бюджетної класифікації. Тому налаштування рядків, що необхідно відобразити для аналізу, відбуватиметься за цим кодом. Для цього створюється список кодів бюджетної класифікації та, для зручності користування, вказується найменування надходжень біля коду. Коди відображаються у вигляді «дерева», де кожна гілка може мати чотири рівні.

У налаштуваннях рядків також кожен елемент списку має checkbox, наявні кнопки «Вибрати всі», «Очистити всі» та «Підтвердити», які виконують всі ті ж самі функції, але для рядків. Єдину відмінність має checkbox. Якщо на нього натиснути за допомогою комбінації клавіш ctrl + click, то активними або неактивними, стане не тільки вибраний код (рівень), а також і всі його дочірні елементи (підрівні), якщо такі є (рис. 3.6).

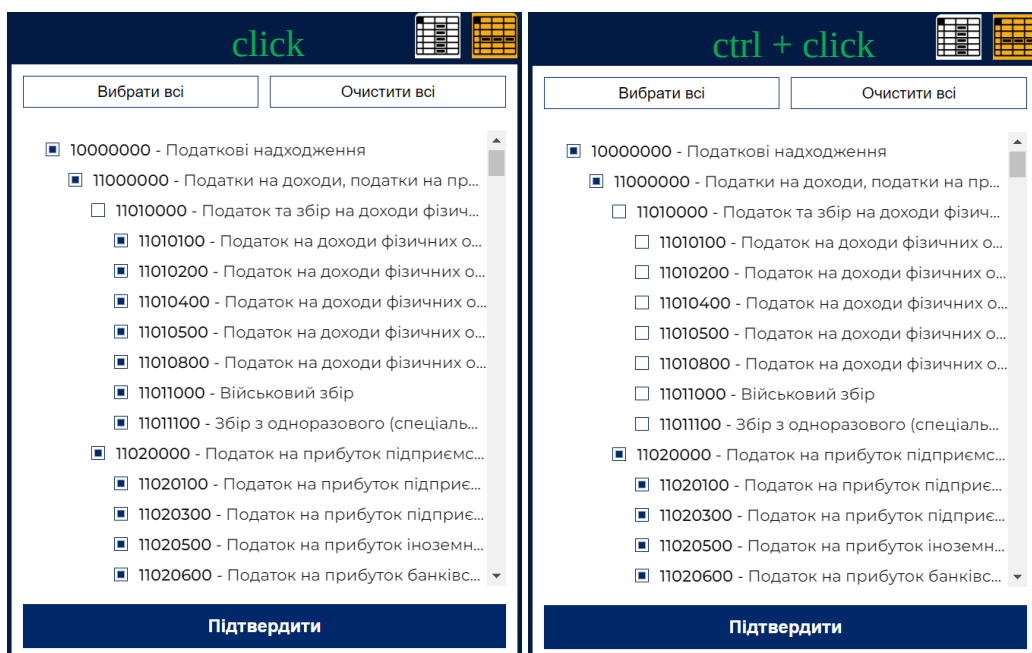


Рис. 3.6. Вибір необхідних для відображення рядків у таблиці

Поле пошуку. Оскільки даних у таблиці багато, то додано також поле пошуку, яке допоможе швидко знайти необхідні дані у таблиці. Слід звернути увагу, що пошук здійснюється та видає результати лише по вибраних полях у налаштуваннях. Також результат роботи поля пошуку ніяк не впливає на кінцевий результат даних, що будуть завантажуватись на ПК. Тобто, дані завантажуватися будуть лише ті, що вказані у налаштуваннях, без врахування змін таблиці за допомогою поля пошуку.

Збереження даних на ПК. Після аналізу та обробки даних користувач може зберегти їх на свій ПК в одному з представлених форматів: CSV, XML, XLSX, HTML, JSON.

У результаті з реалізованим вищеописаним функціоналом та заданим йому зовнішнім виглядом за допомогою CSS отримаємо частину реалізованого веб-додатку «Вибірка даних», що представлено на рисунку 3.7. Режим налаштування рядків можна побачити на рисунку 3.6.

Тип фонду	Код	Найменування коду	Початковий річний план	Уточнений річний план
	0000000	РАЗОМ	1553029918600	1666213327000.65
TOTAL	10000000	Податкові надходження	1204081760600	1204081760600
TOTAL	11000000	Податки на доходи, податки на прибуток, податки на збільшення ринкової вартості	319843711600	319843711600
TOTAL	11010000	Податок та збір на доходи фізичних осіб	149811991300	149811991300
TOTAL	11010100	Податок на доходи фізичних осіб, що сплачується податковими агентами, із доходів платника податку у вигляді заробітної плати	0	0
TOTAL	11010200	Податок на доходи фізичних осіб з грошового забезпечення, грошових винагород та інших виплат, одержаних військовослужбовцями та особами рядового і начальницького складу, що сплачується податковими агентами	0	0
TOTAL	11010400	Податок на доходи фізичних осіб, що сплачується податковими агентами, із доходів платника податку інших ніж заробітна плата	0	0
TOTAL	11010500	Податок на доходи фізичних осіб, що сплачується фізичними особами за результатами річного	0	0

Рис. 3.7. Результат відображених даних у таблиці завантажених через API

Створення діаграм. Для представлення даних у вигляді діаграми чи графіку використовувалася бібліотека ChartJS, яка дозволяє будувати різні діаграми, гістограми, графіки і т.д. Оскільки додаток створюється за допомогою React, то для роботи з діаграмами використовується спеціальна бібліотека, що налаштована під нього – react-chartjs-2.

ChartJS пропонує багато різних видів діаграм, графіків та їх комбінацій. Щоб побудувати діаграми необхідного типу у React використовується компонент з назвою цього типу. Наприклад, `<Doughnut />` – кругла діаграма або `<Bar />` – стовпчаста діаграма і т.д.

У параметри компонента передається вся необхідна інформація для налаштування та побудови діаграми. Так, вказуються фактичні числові дані, на основі яких буде побудована діаграма, колір стовпців чи секторів на діаграмі, легенда даних, на якій відображається інформація про кожен стовпець чи сектор діаграми, вказується інформація, що буде відображатись при наведенні на них і так далі. Кожна діаграма також має власні параметри, за допомогою яких можна керувати її виглядом: вказувати розмір діаграм, шрифт, його розмір і колір, відступи між елементами діаграми тощо.

Діаграма «Виконання бюджету». На даній діаграмі буде представлено відсоток виконання уточненого річного плану по надходженню коштів до бюджету України, сума в гривнях, що надійшла та сума, якої не вистачає до виконання плану.

Для цієї діаграми використовується компонент Doughnut із згаданої раніше бібліотеки, оскільки це кругова діаграма, яка корисна тоді, коли потрібно порівняти один конкретний параметр або набір даних, що і підходить для вирішення поставленого завдання.

Дані, що будуть передані у компонент, беруться з першого елементу масиву даних (`data[0]`), оскільки він містить загальні (сумарні) відомості по всіх надходженнях. Отже, у компонент у якості фактичних даних передається масив із двох значень, де перше – це загальна сума надходжень, а друге значення – різниця між уточненим річним планом та виконаним планом, тобто, скільки залишилось до виконання уточненого річного плану. Також для `labels` передається масив `chartLabel` із найменуваннями секцій діаграми (`["Виконано за період", "Залишилось до уточненого річного плану"]`).

Дана діаграма має бути у вигляді не повного кола, а половини, тобто 180 градусів, а також потрібно буде повернути її на 90 градусів, щоб її положення було горизонтальним. Це також вказується у параметрах компонента.

Передати можна також певні налаштування для діаграми у параметр `option`. Оскільки має відображатися лише півколо діаграми, то у налаштуваннях потрібно вказати, що ширина буде більша удвічі за висоту, щоб не було пустого місця, яке залишиться під іншу половину кола діаграми, яка не відображатиметься. Також потрібно відмінити відображення стандартної легенди секторів діаграми, оскільки створюватиметься своя власна. Легенда розшифровує назви рядків даних за якими було побудовано діаграму.

Компонент із вищенаведеними параметрами, що передаються в нього, зображено на рисунку 3.8.

```

<Doughnut id='plan-done-id'
  data={
    labels: chartLabel,
    datasets: [
      {
        data: [data[0]['periodDone'], data[0]['yearCorrectionPlan'] - data[0]['periodDone']],
        backgroundColor: [
          '#13aa75',
          '#5a5a5a',
        ],
        borderWidth: 0,
        rotation: -90, circumference: 180, cutout: "60%"
      },
    ],
  }
  options={{
    aspectRatio: 2,
    plugins: {
      legend: {
        display: false,
      },
    },
  }}
  ref={chartRef}
/>

```

Рис. 3.8. Компонент Doughnut для побудови діаграми

З правої сторони біля діаграми була додана власна легенда стосовно секторів діаграми з більшою кількістю інформації про виконання бюджету, ніж це показувала легенда за замовчуванням, яку було приховано. Діаграма «Виконання бюджету» зображена на рисунку 3.9.

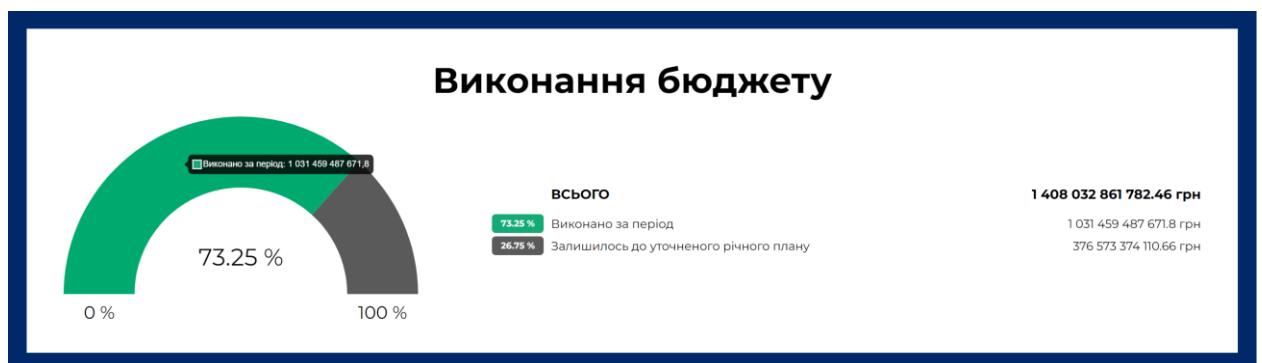


Рис. 3.9. Діаграма «Виконання бюджету»

Діаграма «Структура бюджету». Діаграма показує який відсоток загальних надходжень становить кожен вид надходжень на кожному рівні, що дає змогу аналізувати його більш детально та мати представлення того, які з надходжень на кожному рівні мають найбільший внесок у виконання річного плану надходжень до бюджету держави.

Дана діаграма також створюється за допомогою компонента Doughnut (круглої діаграми), яка також корисна у ситуаціях, коли потрібно показати пропорцію, в якій щось розподілено між різними об'єктами. Це саме те, що потрібно, щоб показати який відсоток від загальної суми надходжень до бюджету становить той чи інший вид надходжень. Для її побудови у параметри компонента передаються всі необхідні дані, такі як масив значень, необхідних для побудови секторів діаграми та масив з їх описом, масив кольорів для цих секторів, також приховується відображення стандартної легенди, оскільки буде відображатися своя власна, вказується, як має реагувати діаграма при наведенні курсора на сектор діаграми чи натисканні на нього тощо.

З правої сторони біля діаграми була також додана легенда діаграми (опис її секторів), яка показує відсоток та суму кожного виду надходжень від загальної суми надходжень (рис. 3.10).

Також під назвою діаграми вказується назва рівня, щоб було зрозуміло, який саме вид надходжень (рівень) зараз аналізується. За замовчуванням відображається перший рівень, який має назву «Доходи».



Рис. 3.10. Діаграма «Структура бюджету»

Користувач може взаємодіяти з даною діаграмою для проведення більш детального аналізу. Натиснувши на один із секторів діаграми або на

відповідний йому елемент у легенді справа відбувається перехід на рівень нижче по вибраному секторі. Як показано на рисунку 3.11, на початковій діаграмі з відображенням доходів першого рівня було вибрано «Податкові надходження» і по цій категорії надходжень відбувся перехід на нижчий рівень. Також змінилася й назва рівню на «Податкові надходження», щоб було зрозуміло, який саме вид надходжень аналізується зараз.

Тобто, з діаграми можна дізнатись, що загальна сума податкових надходжень за вибраний період – 603млрд., що становить 58,49% від загальної суми усіх надходжень до бюджету. Натиснувши на сектор «Податкові надходження» відбудеться перехід на один рівень нижче. Тут можна побачити, що податкові надходження складаються з 5 інших видів надходжень та яку суму і який відсоток податкових надходжень становить кожен із них.



Рис. 3.11. Діаграма «Структура бюджету». Перехід на нижній рівень податкових надходжень

Оскільки є можливість переходити на рівень нижче по тому чи іншому видові надходжень, то має бути й можливість повернутись на один чи одразу на декілька рівнів назад. За збереження даних щодо послідовності рівнів та інформацію про них відповідає змінна breadcrumbs. На основі даних з цієї змінної і відображається назва поточного рівня над діаграмою. Також, при

наведенні курсору на назву поточного рівня під нею з'явиться блок із рівнями від першого до поточного, де можна перейти на рівень чи декілька рівнів вище натиснувши на нього (рис. 3.12).

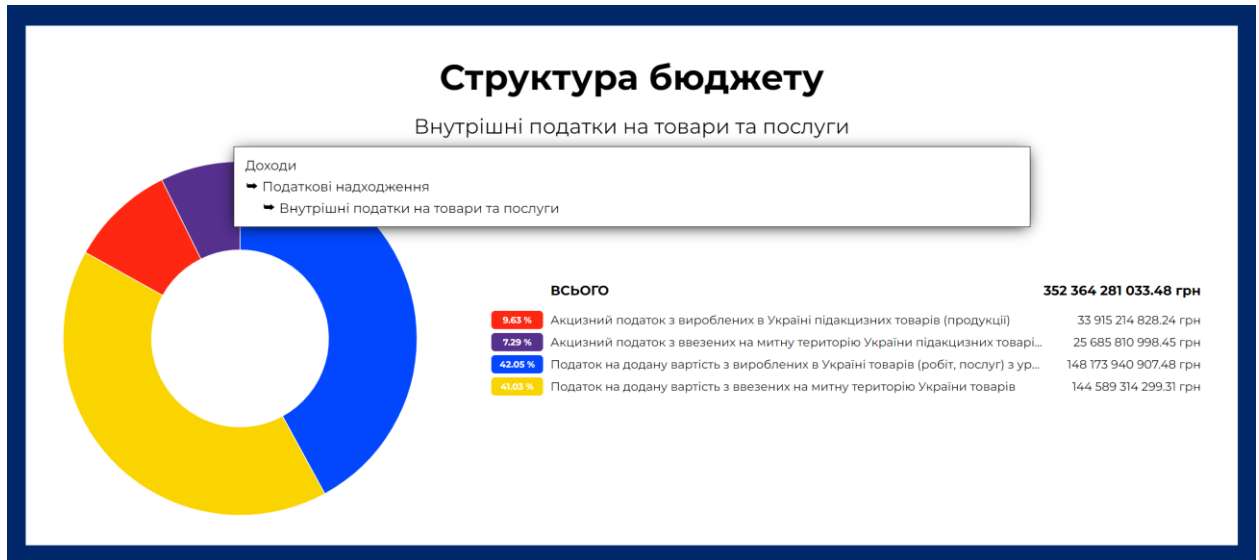


Рис. 3.12. Відображення «дерева» рівнів надходжень від верхнього рівня до поточного

Діаграма «Виконання бюджету за видами надходжень». Ця діаграма показує початковий план та фактичне його виконання по кожному з видів надходжень першого рівня.

Побудова даної діаграми створюється за допомогою компонента Bar, тобто стовпчастої діаграми, яка використовується для відображення тенденцій даних і порівняння кількох наборів даних поруч. У якості параметрів у компонент буде передано два набори даних:

- масив числових значень уточненого плану надходжень;
- масив числових значень виконання початкового плану надходжень (поточна сума надходжень по кожному виду).

Сума одних видів надходжень може вимірюватись мільярдами гривень, в той час як інша лише мільйонами. Через таку велику різницю стовпці деяких надходжень на діаграмі можуть бути просто непомітні, оскільки за замовчуванням встановлений лінійний тип діаграми і міститься шкала з постійним кроком, який встановлюється автоматично, залежно від

найменшого та найбільшого значень даних, зображених на діаграмі. Тому у параметрі, що відповідає за додаткові налаштування діаграми, потрібно задати для шкали діаграми логарифмічний тип (рис. 3.13).

```
<Bar data={
  labels: data.filter(obj => codes["00000000"]['codes'][obj['incomeCode']]).map(obj => obj['incomeCodeName']),
  datasets: [
    {
      label: renamedColumnNames['yearBeginPlan'],
      data: data.filter(obj => codes["00000000"]['codes'][obj['incomeCode']]).map(obj => obj['yearBeginPlan']),
      backgroundColor: 'rgba(2, 71, 254, 0.7)',
    },
    {
      label: renamedColumnNames['periodDone'],
      data: data.filter(obj => codes["00000000"]['codes'][obj['incomeCode']]).map(obj => obj['periodDone']),
      backgroundColor: 'rgba(250, 212, 0, 0.7)',
    },
  ],
})
options = {
  maintainAspectRatio: true,
  scales: {
    y: {
      type: 'logarithmic',
    },
  },
}
} />
```

Рис. 3.13. Компонент Bar для побудови діаграми

Результат діаграми представлено на рисунку 3.14. У даному випадку легенда залишилася за замовчуванням, оскільки потрібен тільки опис назви рядів даних, за якими було побудовано діаграму.



Рис. 3.14. Діаграма «Виконання бюджету за видами надходжень»

У результаті створений додаток може отримувати дані через API та обробляти, відображати їх як у вигляді таблиці з можливістю фільтрації та подальшого їх збереження на ПК у одному з представлених форматів, так і у вигляді діаграм (з однією можна взаємодіяти) для кращого аналізу.

ВИСНОВКИ

У ході дослідження теми дипломного проекту було проаналізовано та підтверджено актуальність використання відкритих даних, важливість їх місця у сьогоденні та можливостей, що вони надають.

Представлено результати дослідження стану відкритих даних, наведені нормативні документи, що регламентують їх формування та наведено порівняльний аналіз розвитку відкритих даних країн ЄС та в Україні, що вказують на широкі можливості застосування різних технологій для їх використання та аналізу.

Розглянуто та систематизовано веб-портали, що накопичують і надають відкриті дані, веб-додатки, що створюються на їх основі та проблеми, які вони вирішують.

Проведено дослідження надходжень до бюджету держави за бюджетною класифікацією та ресурси, що надають інформацію про надходження у формі відкритих даних.

Результати дослідження способів отримання відкритих даних за допомогою REST API підтверджують можливість їх практичного застосування для розробки спеціалізованих застосунків. Розроблений додаток показує таку можливість і у даній реалізації надає можливість редагувати відображувані дані для аналізу, а саме, обирати необхідні рядки та стовпці даних податкових надходжень для відображення. Наявне поле пошуку для пошуку даних по таблиці.

Показано, що для глибшого аналізу засобами JS дані можна відображати у вигляді діаграм виконання запланованого річного бюджету як в цілому, так і окремо по видам податків. Додаток дає можливість аналізувати відсоток кожного із виду надходжень від загальної суми надходжень, упорядковуючи від найбільшого до найменшого значення і навпаки.

Додаток надає можливість аналізувати кожен із видів надходжень за рівнями бюджетної класифікації, що дає змогу дізнатися більш детально з яких надходжень складається той чи інший вид податку і який відсоток вони становлять (наприклад «Податки на доходи, прибуток та збільшення ринкової вартості» складаються із «Податок на доходи фізичних осіб», «Податок на прибуток підприємств» тощо).

Отже, можна сформулювати основний висновок, що створення спеціалізованих веб-додатків для доступу до відкритих даних податкових надходжень на основі API державного веб-порталу OpenBudget відкриває нові можливості для аналітичної діяльності різних наукових та освітніх установ з можливістю застосування технологій машинного навчання та штучного інтелекту. Такі веб-додатки за допомогою REST API дозволять отримувати дані у форматі JSON і відображати їх у різних графічних форматах.

У подальшій роботі над додатком можна приділити увагу розширенню функціональності у частині графічного представлення інформації, використанню методів машинного навчання, штучного інтелекту для прогнозування податків, класифікації регіонів за податковими ризиками, оцінювання економічного потенціалу регіонів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 4 Types of APIs All Marketers Should Know. URL: <https://blog.hubspot.com/website/types-of-apis> (дата звернення: 18.11.2022).
2. An overview of HTTP? URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview> (дата звернення: 18.11.2022).
3. Axios Documentation. URL: <https://axios-http.com/docs/intro> (дата звернення: 18.11.2022).
4. Chart.js Documentation. URL: <https://www.chartjs.org/docs/latest/> (дата звернення: 18.11.2022).
5. CSS: Cascading Style Sheets. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS> (дата звернення: 18.11.2022).
6. Eurostat – European statistics. URL: https://ec.europa.eu/info/departments/eurostat-european-statistics_en (дата звернення: 18.11.2022).
7. FileSaver.js Documentation. URL: <https://github.com/eligrey/FileSaver.js> (дата звернення: 18.11.2022).
8. HTML: HyperText Markup Language. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML> (дата звернення: 18.11.2022).
9. Introduction to web APIs. URL: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Introduction (дата звернення: 18.11.2022).
10. JavaScript. What is JavaScript? URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction> (дата звернення: 18.11.2022).
11. Open data maturity report 2020. URL: https://data.europa.eu/sites/default/files/country-factsheet_ukraine_2020.pdf (дата звернення: 18.11.2022).
12. Open data maturity report 2021. URL: https://data.europa.eu/sites/default/files/country-factsheet_ukraine_2021.pdf (дата звернення: 18.11.2022).

13. Open data maturity report 2022. URL: https://data.europa.eu/sites/default/files/country-factsheet_ukraine_2022.pdf (дата звернення: 10.01.2023).
14. OpenBudget. Про портал. URL: https://openbudget.gov.ua/about-portal_(дата звернення: 18.11.2022).
15. Opendatabot. URL: <https://opendatabot.ua> (дата звернення: 18.11.2022).
16. Open Data Monitor. URL: <https://opendatamonitor.eu/frontend/web/index.php?r=site%2Fabout> (дата звернення: 18.11.2022).
17. React-chartjs-2 Documentation. URL: <https://react-chartjs-2.js.org/> (дата звернення: 18.11.2022).
18. SheetJS Community Edition Documentation. URL: <https://docs.sheetjs.com/docs/> (дата звернення: 18.11.2022).
19. The III International scientific and practical conference “Development of modern science, experience and trends” (October 11-14, 2022) International Science Group, Boston, USA. 2022. 480 p. URL: <https://isg-konf.com/development-of-modern-science-experience-and-trends/> (дата звернення: 18.11.2022).
20. The importance of open data. URL: <https://www.weforum.org/agenda/2016/02/the-importance-of-open-data> (дата звернення: 18.11.2022).
21. The official portal for European data. URL: <https://data.europa.eu/en> (дата звернення: 18.11.2022).
22. Types of APIs & Popular REST API Protocol. URL: <https://stoplight.io/api-types> (дата звернення: 18.11.2022).
23. What is an API? (Application Programming Interface). URL: <https://www.mulesoft.com/resources/api/what-is-an-api> (дата звернення: 18.11.2022).
24. What is an API? URL: <https://blog.axway.com/learning-center/apis/basics/what-is-an-api> (дата звернення: 18.11.2022).

25. What is an API? URL: <https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces> (дата звернення: 18.11.2022).
26. What is API: Definition, Types, Specifications, Documentation. URL: <https://www.altexsoft.com/blog/engineering/what-is-api-definition-types-specifications-documentation/> (дата звернення: 18.11.2022).
27. What is Axios? URL: <https://axios-http.com/docs/intro> (дата звернення: 18.11.2022).
28. What Is REST API and How It Differs From Other Types? URL: <https://appmaster.io/blog/what-rest-api-and-how-it-differs-other-types> (дата звернення: 18.11.2022).
29. What is RESTful API? URL: https://aws.amazon.com/what-is/restful-api/?nc1=h_ls (дата звернення: 18.11.2022).
30. Архітектура Web-додатків. URL: <https://webkonspect.com/?room=group&id=7&labelid=129> (дата звернення: 18.11.2022).
31. ВЕРХОВНА РАДА УКРАЇНИ. Офіційний веб-портал парламенту України. URL: <https://data.rada.gov.ua/open/main/index> (дата звернення: 18.11.2022).
32. Відкриті дані в Україні: досягнення, переваги, застереження. URL: <https://cedos.org.ua/researches/vidkryti-dani-v-ukrayini-dosyagnennya-perevagy-zasterezhennya/> (дата звернення: 18.11.2022).
33. Документація React. URL: <https://uk.reactjs.org/docs/getting-started.html> (дата звернення: 18.11.2022).
34. ЄДИНИЙ ВЕБПОРТАЛ ВИКОРИСТАННЯ ПУБЛІЧНИХ КОШТІВ. URL: <https://spending.gov.ua/new/about-portal> (дата звернення: 18.11.2022).
35. Єдиний державний веб-портал відкритих даних. Про портал. URL: <https://data.gov.ua/pages/about> (дата звернення: 18.11.2022).
36. Зелена книга «Політика відкритих даних». URL: <https://regulation.gov.ua/book/145-zelena-kniga-politika-vidkritih-danih> (дата звернення: 18.11.2022).

37. Методичні рекомендації підготовлені Державним агентством з питань електронного урядування України з залученням експертів програми «Прозорість та підзвітність у державному управлінні та послугах». URL: <https://data.gov.ua/uploads/files/2018-08-11-104353.234698Recomendation.pdf> (дата звернення: 18.11.2022).

38. Мінцифри: Відкриті дані – можливість легко моніторити державні видатки. URL: <https://www.kmu.gov.ua/news/mincifri-vidkriti-dani-mozhlivist-legko-monitoriti-derzhavni-vidatki> (дата звернення: 18.11.2022).

39. Портал реєстру проектів МФО. URL: <https://proifi.gov.ua/about-portal> (дата звернення: 18.11.2022).

40. Про внесення змін до Положення про набори даних, які підлягають оприлюдненню у формі відкритих даних : Постанова КМ від 20.12.2017 р. № 1100 : станом на 20 грудня 2017 р. URL: <https://zakon.rada.gov.ua/laws/show/1100-2017-%D0%BF#Text> (дата звернення: 18.11.2022).

41. Про доступ до публічної інформації : Закон України від 13.01.2011 р. № 2939-VI : станом на 27 січ. 2022 р. URL: <https://zakon.rada.gov.ua/laws/show/2939-17#Text> (дата звернення: 18.11.2022).

42. Про затвердження Положення про набори даних, які підлягають оприлюдненню у формі відкритих даних : Постанова КМ від 21.10.2015 р. № 835 : станом на 11 лютого 2022 р. URL: <https://zakon.rada.gov.ua/laws/show/835-2015-%D0%BF#Text> (дата звернення: 18.11.2022).

43. Що таке веб додаток? Різниця між сайтом, веб-додатком, SPA і PWA. URL: <https://webcase.com.ua/uk/blog/cho-takoe-web-prilozhenie-vse-vidy/#f1> (дата звернення: 18.11.2022).

ДОДАТКИ

Додаток А

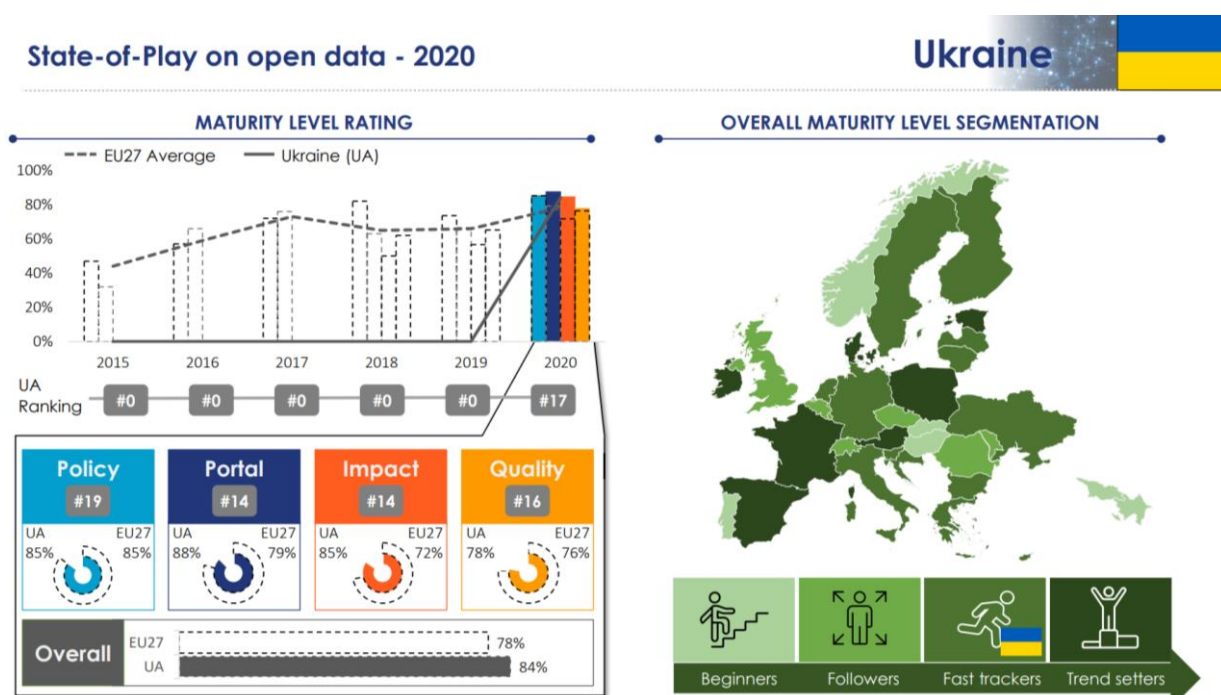


Рис. А1. Порівняння розвитку сфери відкритих даних України та ЄС у 2020 році [11]

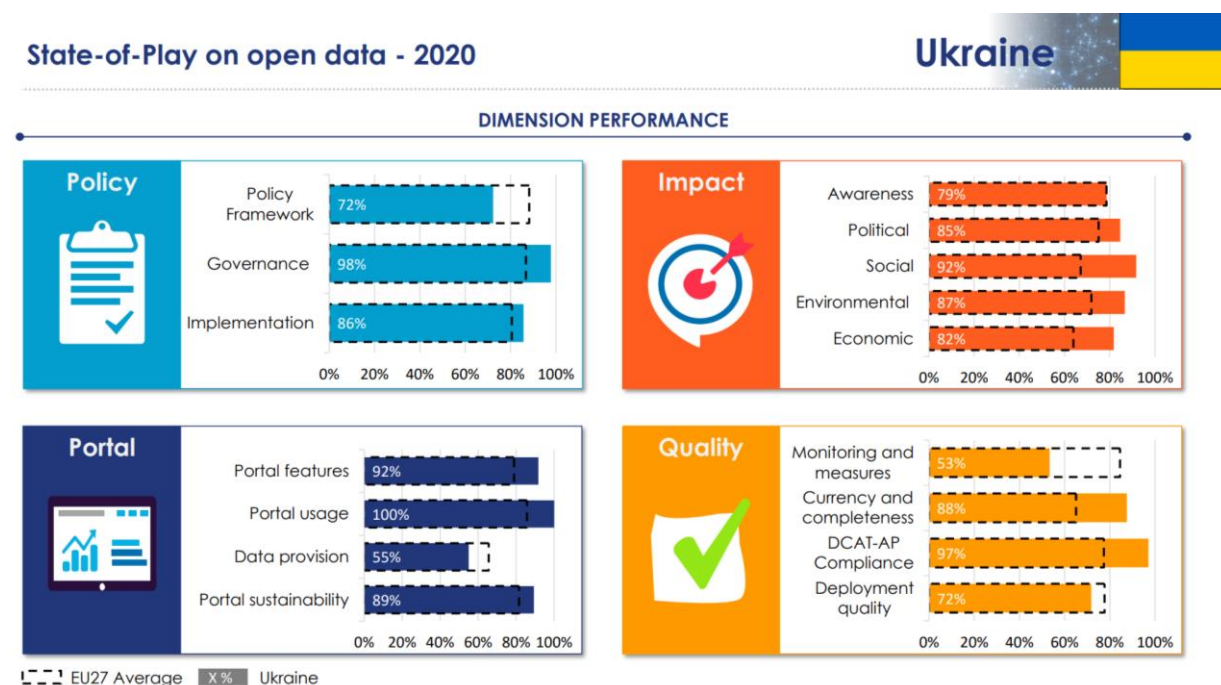


Рис. А2. Рівень розвитку відкритих даних України у 2020 році [11]

Продовження додатка А

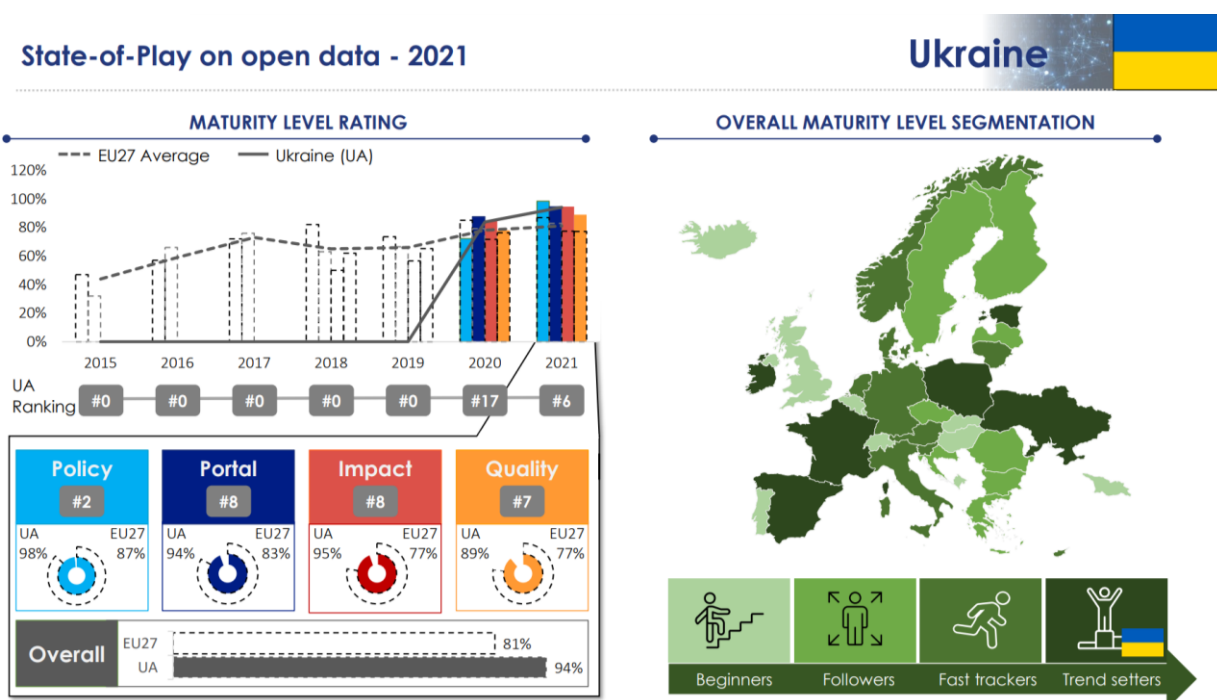


Рис. А3. Порівняння розвитку сфери відкритих даних України та ЄС у 2021 році [12]

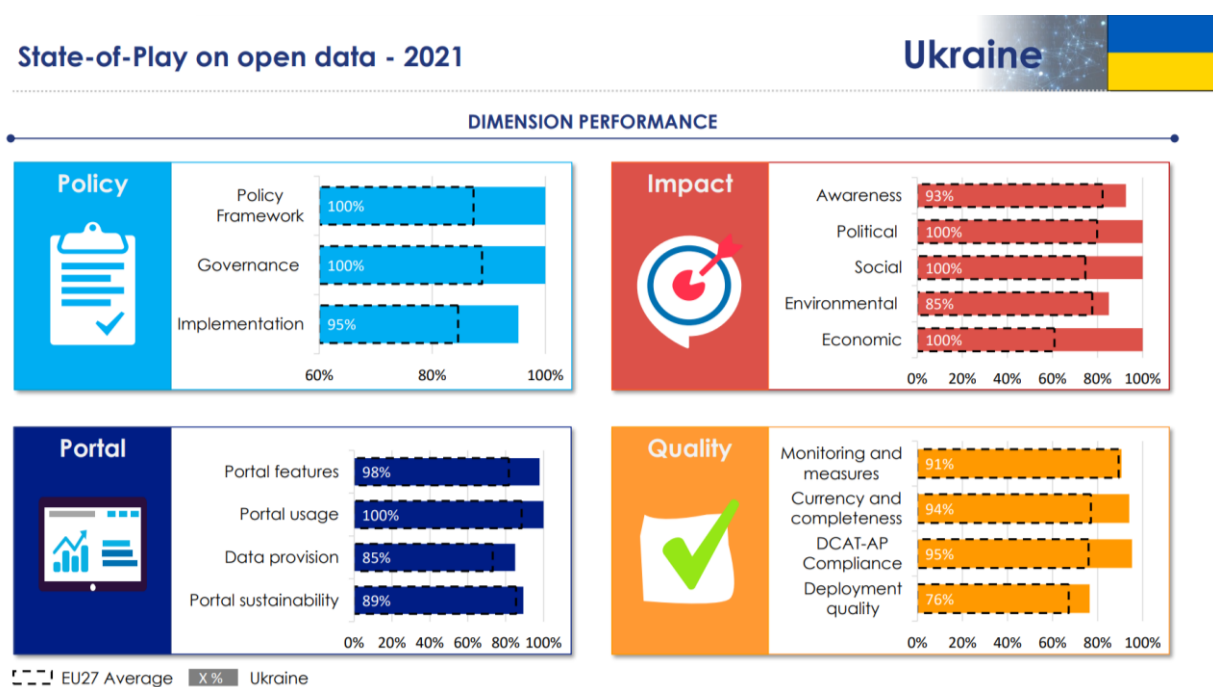


Рис. А4. Рівень розвитку відкритих даних України у 2021 році [12]

Продовження додатка А

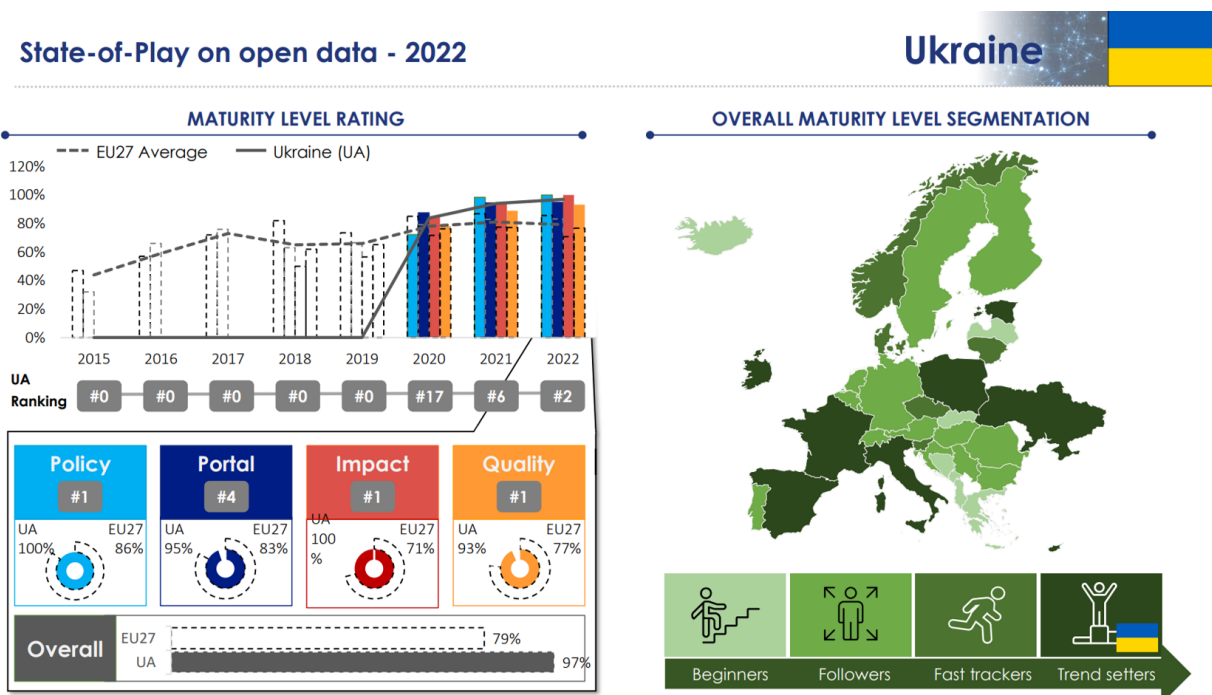


Рис. А5. Порівняння розвитку сфери відкритих даних України та ЄС у 2022 році [13]

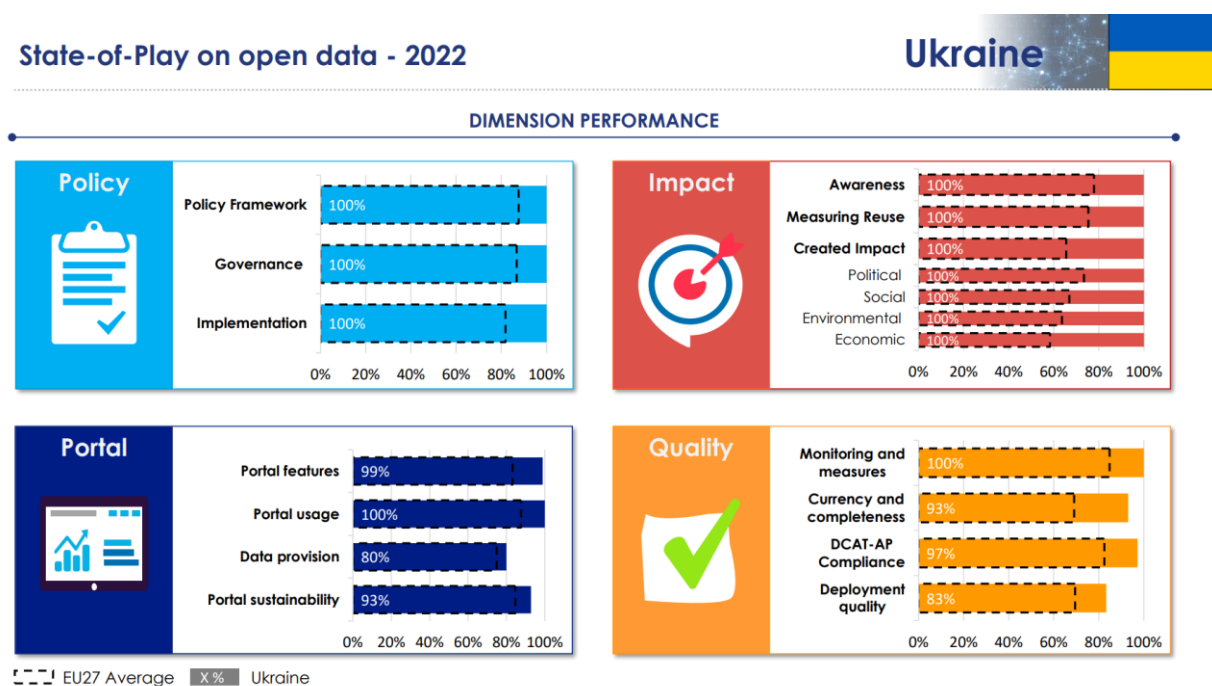


Рис. А6. Рівень розвитку відкритих даних України у 2022 році [13]

```

{
  "fundType": "TOTAL",
  "incomeCode": "10000000",
  "incomeCodeName": "Податкові надходження",
  "yearBeginPlan": 1192488540300,
  "yearCorrectionPlan": 1192488540300,
  "periodCorrectionPlan": null,
  "periodDone": 509725640994.09,
  "deviationToPeriodCorrectionPlan": null,
  "percentDoneToPeriodCorrectionPlan": null,
  "percentDoneToYearCorrectionPlan": 42.74
},
{
  "fundType": "TOTAL",
  "incomeCode": "11000000",
  "incomeCodeName": "Податки на доходи, податки на прибуток, податки на збільшення ринкової вартості",
  "yearBeginPlan": 308250491300,
  "yearCorrectionPlan": 308250491300,
  "periodCorrectionPlan": null,
  "periodDone": 147908028202.86,
  "deviationToPeriodCorrectionPlan": null,
  "percentDoneToPeriodCorrectionPlan": null,
  "percentDoneToYearCorrectionPlan": 47.98
},
{
  "fundType": "TOTAL",
  "incomeCode": "11010000",
  "incomeCodeName": "Податок та збір на доходи фізичних осіб",
  "yearBeginPlan": 149811991300,
  "yearCorrectionPlan": 149811991300,
  "periodCorrectionPlan": null,
  "periodDone": 78508598602.99,
  "deviationToPeriodCorrectionPlan": null,
  "percentDoneToPeriodCorrectionPlan": null,
  "percentDoneToYearCorrectionPlan": 52.4
},

```

Рис. Б1. Масив об'єктів даних та їх будова (отримано через API веб-порталу Open Budget)

```

{
  "10000000": {
    "codes": {
      "11000000": {
        "codes": {
          "11010000": {
            "codes": null,
            "yearBeginPlan": 149811991300,
            "yearCorrectionPlan": 149811991300,
            "periodDone": 91245710422.58,
            "percentDoneToYearCorrectionPlan": 60.9
          }
        },
        "yearBeginPlan": 319843711600,
        "yearCorrectionPlan": 308250491300,
        "periodDone": 178018523680.86,
        "percentDoneToYearCorrectionPlan": 57.75
      },
      "13000000": {
        "codes": {
          "13010000": {
            "codes": null,
            "yearBeginPlan": 862900000,
            "yearCorrectionPlan": 862900000,
            "periodDone": 368413816.88,
            "percentDoneToYearCorrectionPlan": 42.69
          }
        },
        "yearBeginPlan": 71245751200,
        "yearCorrectionPlan": 71245751200,
        "periodDone": 57160603061.19,
        "percentDoneToYearCorrectionPlan": 80.23
      }
    },
    "yearBeginPlan": 1204081760600,
    "yearCorrectionPlan": 1192488540300,
    "periodDone": 603290190872.11,
    "percentDoneToYearCorrectionPlan": 50.59
  }
}

```

Рис. Б2. Перетворений масив об'єктів даних у «дерево» з рівнями за кодами бюджетної класифікації (отримано автором)

Додаток В

- 10000000** – Податкові надходження
 - 11000000** – Податки на доходи, податки на прибуток, податки на збільшення ринкової...
 - 11010000** – Податок та збір на доходи фізичних осіб
 - 11010100** – Податок на доходи фізичних осіб, що сплачується податковими...
 - ...
 - ...
 - 13000000** – Рентна плата та плата за використання інших природних ресурсів
 - 13010000** – Рентна плата за спеціальне використання лісових ресурсів
 - 13010100** – ... лісових ресурсів в частині деревини, заготовленої в порядку рубок...
 - ...
 - ...
 - ...
- 20000000** – Неподаткові надходження
 - 21000000** – Доходи від власності та підприємницької діяльності
 - 21010000** – Частина чистого прибутку (доходу) державних або комунальних...
 - 21010100** – Частина чистого прибутку (доходу) господарських організацій...
 - ...
 - ...
 - ...
- 50000000** – Цільові фонди
 - 50070000** – Надходження до Фонду соціального захисту осіб з інвалідністю

Рис. В1. Поділ доходів до бюджету на рівні за кодом
бюджетної класифікації

ЛІСТИНГ ОСНОВНИХ ФУНКЦІЙ

Завантаження даних через API

```
const [selectedDataParam, setSelectedDataParam] = React.useState({}); // У змінній зберігаються
встановлені значення для полів Рік / Період / Вид бюджету / Область, необхідні для отримання
даних через API

// Відправка GET-запиту відповідно до заданих користувачем параметрів
function getData(e) {
  setIsLoading(false); // З'явиться вікно Loading, поки буде виконуватися запит для отримання даних
  if(e) e.preventDefault();
  if(selectedDataParam.budgetType === "LOCAL" && selectedDataParam.budgetCode !==
  "0000000000") {
    // Отримання даних про місцеві надходження конкретної області
    axios.get(`https://openbudget.gov.ua/api/localBudgets/incomesLocal/JSON?year=${selectedDataPar
    am.year}&periodTo=${selectedDataParam.toMonth}&periodType=MONTH&codeBudget=${selectedData
    Param.budgetCode}&periodFrom=${selectedDataParam.fromMonth}&fundType=TOTAL`)
    .then(request => {
      // Збереження отриманих даних у змінну data
      setData(request.data.map(obj => {
        let newObj = {}
        // У об'єктах, що містять інформацію про місцеві надходження певної області, відрізняються
        найменування ключів від ключів об'єктів, котрі отримуємо в результаті запиту на отримання
        інформації про загальні надходження, хоча інформація та ж сама. Тому замінюємо найменування
        таких полів для подальшого функціонування додатку
        Object.entries(obj).forEach(([key, value]) => {
          switch (key) {
            case 'totalDone':
              newObj["periodDone"] = value;
              break;
            case 'yearBudgetPlan':
              newObj["yearCorrectionPlan"] = value;
              break;
            case 'percentDone':
              newObj["percentDoneToYearCorrectionPlan"] = value;
              break;
            default:
              newObj[key] = value;
              break;
          }
        })
      }
    ));
    setIsLoading(true); // Зникне вікно Loading, після отримання даних
  })
  .catch(error => console.log(error));
} else {
```

Продовження додатка Г

```

// Отримання даних про надходження до державного бюджету або про загальні надходження до
місцевого бюджету
axios.get(`https://openbudget.gov.ua/api/reports/income/details/JSON?budgetType=${selectedDataP
aram.budgetType}&fundType=TOTAL&year=${selectedDataParam.year}&monthTo=${selectedDataPara
m.toMonth}&monthFrom=${selectedDataParam.fromMonth}`)
  .then(request => {
    setData(request.data); // Збереження отриманих даних у змінну data
    setIsLoading(true); // Зникне вікно Loading, після отримання даних
  })
  .catch(error => console.log(error));
}
}

```

Перетворення даних до вигляду дерева з рівнями за кодом бюджетної класифікації доходів

```

const [data, setData] = React.useState([]); // Завантажені через API дані
const [codes, setCodes] = React.useState({}); // Перетворений масив об'єктів даних до вигляду
об'єкта з рівнями
// Функція для перетворення даних, що надійшли у вигляді масиву до вигляду дерева з рівнями за
кодами бюджетної класифікації, подібно до вигляду у Додатку В. Додаток Б рисунок Б2 результат
роботи даної функції
function convertingDataToTree(data) {
  if (data.length > 0) {
    let codesTree = {}; // Змінна, у яку будуть формуватися дані у вигляді "дерева" з рівнями
    податкових надходжень
    // Утворення основних гілок. Основні види надходжень 1 рівня, з яких складається державний
    бюджет
    let index = 1;
    while (index < data.length) {
      let code = data[index]['incomeCode'];
      let codeLevel;
      if (code.slice(1, -1) == 0) {
        codeLevel = code.slice(0, 1);
      } else if (code.slice(2, -1) == 0) {
        codeLevel = code.slice(0, 2);
      } else if (code.slice(4, -1) == 0){
        codeLevel = code.slice(0, 4);
      } else {
        codeLevel = code.slice(0, 6);
      }
      codesTree[code] = {};
      codesTree[code]['codes'] = {};
      codesTree[code]['incomeCodeName'] = data[index]['incomeCodeName'];
      codesTree[code]['yearCorrectionPlan'] = data[index]['yearCorrectionPlan'];
      codesTree[code]['percentDoneToYearCorrectionPlan'] =
data[index]['percentDoneToYearCorrectionPlan'];
      codesTree[code]['periodDone'] = data[index]['periodDone'];
    }
  }
}

```

Продовження додатка Г

```

[codesTree[code]['codes'], index] = createCodesTree(data, index, codeLevel);
}
// Збереження у змінну codes перетворених функцією convertingDataToTree даних до вигляду
// дерева з рівнями
setCodes({
  [data[0]['incomeCode']]: {
    'codes': codesTree,
    'incomeCodeName': data[0]['incomeCodeName'],
    'yearCorrectionPlan': data[0]['yearCorrectionPlan'],
    'percentDoneToYearCorrectionPlan': data[0]['percentDoneToYearCorrectionPlan'],
    'periodDone': data[0]['periodDone'],
  }
});
// Рекурсивна функція, яка поділяє кожну з основних гілок 1 рівня на підрівні
function createCodesTree(data, prevID, prevCodeLevel) {
  let currentID = prevID+1;
  if (!data[currentID]) return [null, currentID];
  let code = data[currentID]['incomeCode'];
  let codeLevel = code.slice(0, prevCodeLevel.length);
  let currentCodeTree = {};
  if (prevCodeLevel === codeLevel){
    while (prevCodeLevel === codeLevel) {
      currentCodeTree[code] = {};
      [currentCodeTree[code]['codes'], index] = createCodesTree(data, currentID, code.slice(0,
codeLevel.length === 1 ? codeLevel.length+1 : codeLevel.length+2));
      currentCodeTree[code]['incomeCodeName'] = data[currentID]['incomeCodeName']
      currentCodeTree[code]['yearCorrectionPlan'] = data[currentID]['yearCorrectionPlan']
      currentCodeTree[code]['percentDoneToYearCorrectionPlan'] =
data[currentID]['percentDoneToYearCorrectionPlan']
      currentCodeTree[code]['periodDone'] = data[currentID]['periodDone']
      currentID = index;
      if (!data[currentID]) return [currentCodeTree, currentID];
      code = data[currentID]['incomeCode'];
      codeLevel = code.slice(0, prevCodeLevel.length);
    }
    return [currentCodeTree, index];
  } else {
    return [null, prevID+1];
  }
}
}
}

```

Завантаження відредагованих даних на ПК у вибраному форматі

```

const [showFields, setShowFields] = React.useState({columns: new Set(), rows: new Set({})}); //
Колонки та рядки вибрані в налаштуваннях таблиці для відображення
const [columnNames, setColumnNames] = React.useState([]); //Перелік усіх стовпців, що можуть бути

```

Продовження додатка Г

```

const fileTypes = ['xlsx', 'xls', 'csv', 'json', 'html']; // Можливі формати завантаження даних на ПК
let [activeFileType, setActiveFileType] = React.useState(fileTypes[0]); // У змінній зберігається
вибраний тип файлу для завантаження. За замовчуванням .xlsx

// Функція для завантаження відфільтрованих у таблиці даних у вибраному форматі на ПК
користувача
function download() {
  let wb = XLSX.utils.book_new();
  wb.SheetNames.push("Data");
  let ws_data = [];

  // Фільтрація даних для збереження відповідно до встановлених налаштувань рядків та стовпців
  let checkedFilds = columnNames.filter((item, index) => showFields.columns.has(index));
  data.filter(obj => showFields.rows.has(obj["incomeCode"])).forEach((item, index) => {
    let itemKeys = Object.keys(item);
    let rowData;
    if (activeFileType === 'json') {
      rowData = {};
      checkedFilds.forEach((field, index) => {
        rowData[field] = (itemKeys.includes(field) ? item[field] : '-')
      })
    } else {
      index === 0 && ws_data.push(checkedFilds);
      rowData = [];
      checkedFilds.forEach((field, index) => {
        rowData.push(itemKeys.includes(field) ? (typeof(item[field]) === 'number' ? String(item[field]) :
item[field]) : '-')
      })
    }
    ws_data.push(rowData)
  })

  // Формування файлу з даними та завантаження його
  if (activeFileType === 'json') {
    saveAs(new Blob([JSON.stringify(ws_data, null, 2)], {type: "application/json;charset=" +
document.characterSet}), "data.json")
  } else {
    let ws = XLSX.utils.aoa_to_sheet(ws_data);
    wb.Sheets["Data"] = ws;
    let wbout = XLSX.write(wb, { bookType: activeFileType, type: 'binary' });
    function s2ab(s) {
      let buf = new ArrayBuffer(s.length);
      let view = new Uint8Array(buf);
      for (let i = 0; i < s.length; i++) view[i] = s.charCodeAt(i) & 0xFF;
      return buf;
    }
    saveAs(new Blob([s2ab(wbout)], { type: "application/octet-stream" }), `Data.${activeFileType}`);
  }
}

```

ВІДГУК

наукового керівника на випускну кваліфікаційну (дипломну) роботу
з галузі знань 12 «Інформаційні технології» за спеціальністю 122
«Комп'ютерні науки», освітньо-професійної програми «Інформаційні
управляючі системи та технології в економіці», виконану студентом групи
КМІз-21-1

Мандриком Владиславом Володимировичем

на тему

**«Розробка web-додатку для аналізу відкритих даних податкових
надходжень»**

Актуальність теми. Доступ до актуальної інформації про податкові надходження є завжди досить важливим завданням і процесом у здійсненні наукових досліджень, використання таких даних у освітній діяльності. Враховуючи значний прогрес України у напрямі відкриття даних державних установ, зокрема Міністерства фінансів, дослідження інфраструктури відкритих даних, зокрема доступ до даних податкових надходжень, технологій їх використання є актуальним завданням.

Наукова новизна і практична значущість результатів. Отримані у роботі практичні результати можуть бути використані у науковій діяльності освітніх та дослідницьких установ, в органах місцевого самоврядування для швидкого отримання актуальних даних про надходження до бюджету держави, здійснення їх базового графічного та статистичного аналізу. Отримані дані можна використовувати для більш глибокого аналізу засобами інтелектуального аналізу та машинного навчання.

Автором розроблений веб-додаток з функціями відображення результатів податкових надходжень на платформі ReactJS, який може отримувати відкриті дані надходжень до бюджету держави з державного веб-порталу OpenBudget через API та обробляти, відображати їх як у вигляді таблиці з можливістю фільтрації та подальшого їх збереження на ПК у одному з представлених форматів, так і у вигляді діаграм (з однією можна взаємодіяти) для кращого аналізу.

Апробація результатів виконана на вигляді тез на Міжнародній науково-практичній конференції «Development of modern science, experience and trends», що проходила 11-14 жовтня 2022 р. у Бостоні, США.

Зауваження до роботи. Зустрічаються неякісні рисунки із нерозбірливими написами і незначні недотримання вимог оформлення роботи. Проте зазначені недоліки не применшують теоретичні і практичну значимість роботи.

Загальний висновок. За змістом, науковою новизною та практичною значимістю результатів робота відповідає вимогам до кваліфікаційних робіт випускників магістерського рівня, рекомендується до захисту перед Державною екзаменаційною комісією і заслуговує оцінки «**відмінно**».

Автору роботи **Мандрику Владиславу Володимировичу** може бути присвоєно ступінь магістра з галузі знань 12 «Інформаційні технології» спеціальності 122 «Комп'ютерні науки».

Науковий керівник

к.е.н, доцент кафедри комп'ютерних
та інформаційних технологій і систем



О.В. Редич

РЕЦЕНЗІЯ

на випускню кваліфікаційну роботу здобувача вищої освіти другого (магістерського) рівня з галузі знань 12 «Інформаційні технології» спеціальності 122 «Комп'ютерні науки» освітньо-професійної програми «Інформаційні управляючі системи і технології в економіці»,

студента групи КМІз-21-1

Державного податкового університету

Мандрика Владислава Володимировича на тему
«Розробка web-додатку для аналізу відкритих даних податкових надходжень»

Актуальність теми. Автор обґрунтовує вибір теми важливістю швидкого отримання актуальних даних про податкові надходження для здійснення наукових досліджень, використання їх в освітній діяльності. Відкриття даних державних установ для широкого загалу, зокрема Міністерства фінансів, дозволяє розробляти веб-додатки для доступу до таких Відкритих Даних, застосовувати сучасні технології їх оброблення та аналізу. Тому дослідження таких технологій та їх використання є актуальним заданням.

Наукова новизна. Автором доведено можливість застосування технологій доступу та аналізу відкритих даних з використанням Application Programming Interface (API) та технології REST і реалізації на їх основі веб-додатку з використанням даних про надходження до бюджету України openbudget.gov.ua. Предметом дослідження є технології розробки веб-додатків на базі Java Script, фреймворку ReactJS, API та технології REST.

Якість проведеного аналізу проблеми, текстового та табличного матеріалу

У ході дослідження теми дипломного проекту було проаналізовано та підтверджено актуальність використання відкритих даних, важливість їх місця у сьогоденні та можливостей, що вони надають. Результати дослідження стану відкритих даних, наведені нормативні документи, що регламентують їх формування та порівняльний аналіз розвитку відкритих даних країн ЄС та в Україні вказують на широкі можливості застосування різних технологій для їх використання та аналізу. Графічний матеріал якісно розширює сприйняття текстової інформації.

Обґрунтованість пропозицій і висновків та їх практична цінність

Додаток надає можливість отримувати офіційні відкриті дані із сайту OpenBudget та аналізувати кожен із видів надходжень за рівнями бюджетної класифікації.

Заслужують на увагу висновки автора, проте те, що можливість створення спеціалізованих веб-додатків для доступу до відкритих даних податкових надходжень на основі API державного веб-порталу OpenBudget відкриває нові можливості для аналітичної діяльності різних наукових та освітніх установ з можливістю застосування технологій машинного навчання та штучного інтелекту.

Рекомендації та зауваження. У подальшій роботі над додатком можна приділити увагу розширенню функціональності у частині графічного представлення інформації, використанню методів машинного навчання, штучного інтелекту для прогнозування податків, класифікації регіонів за податковими ризиками, оцінювання економічного потенціалу регіонів.

На деяких рисунках занадто дрібні написи.

Загальний висновок і оцінка роботи. За змістом, науковою новизною та практичною значимістю результатів робота відповідає вимогам до кваліфікаційних робіт випускників магістерського рівня, рекомендується до захисту перед Державною екзаменаційною комісією і заслуговує оцінки «відмінно».

Автору роботи **Мандрику Владиславу Володимировичу** може бути присвоєно ступінь магістра з галузі знань 12 «Інформаційні технології» спеціальності 122 «Комп'ютерні науки».

Рецензент :

Завідувач лабораторії відкритих інновацій
Київського академічного університету
к.т.н.



Володимир НОЧВАЙ

«30» січня 2023 р.

Висновок кафедри про випускну кваліфікаційну (дипломну) роботу

Випускна кваліфікаційна (дипломна) робота здобувача вищої освіти

Мандрика Владислава Володимировича

після попереднього захисту розглянута на засіданні кафедри, протокол від «12» 01 2023 р. № 6 і допущена до захисту перед екзаменаційною комісією.

Завідувач кафедри



М.М. Філоненко

Керівник
освітньо-професійної програми

А.Ю. Горбовий