

У. Ю. Дзелендзяк<sup>1</sup>, В. В. Самотий<sup>2, 3</sup>, В. О. Палюшок<sup>1</sup>

<sup>1</sup> Національний університет “Львівська політехніка”,  
кафедра комп’ютеризованих систем автоматки

<sup>2</sup> Львівський державний університет безпеки життєдіяльності,  
кафедра управління інформаційною безпекою

<sup>3</sup> Краківська Політехніка ім. Тадеуша Костюшко,  
кафедра автоматки та інформаційних технологій

## РОЗРОБЛЕННЯ ІНТЕРАКТИВНОГО ВЕБ-ДОДАТКА З ВИКОРИСТАННЯМ НЕРЕЛЯЦІЙНОЇ БАЗИ ДАНИХ

© Дзелендзяк У. Ю., Самотий В. В., Палюшок В. О., 2019

Розроблено інтерактивний веб-додаток, який складається з бази даних, сховища даних, прогресивного мобільного додатку та основного веб-інтерфейсу. Він дає можливість створювати, відправляти та опрацьовувати заявки користувачів, а також переглядати, змінювати, видаляти та переносити елементи, управляти додатковими функціями, що є недоступними для звичайного користувача. Ще однією важливою особливістю розробленого додатка є можливість роботи offline.

**Ключові слова:** нереляційна база даних, сховище даних, інтерактивний веб-додаток, прогресивний мобільний додаток, веб інтерфейс, хмарні технології.

The interactive web application which consist of database, data warehouses, progressive mobile application and main web interface has been developed. It gives an opportunity create, send and the other proceed users' requests and also review, change, delete and transfer component elements, manage additional functions which are not available for simple user. The other important specification of developed app is ability to work offline.

**Keywords:** non-relational database, data warehouse, interactive web application, progressive mobile application, web interface, cloud technologies.

### Вступ

Історією вже став час, коли веб-сайти склалися із статичного контенту і декількох cgi або java скриптів для обробки форм. Зараз від веб-додатків вимагається значно більше інтерактивності. Кількість різноманітних технологій і підходів для веб-розробок нині здається просто гігантською, а напрям їх розвитку – непередбаченим [2]. Інтерактивні веб-додатки є програмними системами та продуктами, які виконують операції, процедури та дії, під які вони адаптовані: починаючи з простого відображення даних і закінчуючи складними інтерактивними системами обміну даними та системами штучного інтелекту. При цьому вони можуть активно взаємодіяти з користувачем, широко застосовуються у всіх сферах надання послуг, управління інформацією і ресурсами. Знаходять застосування в інтернет-банкінгу та адмініструванні даних. Оскільки сьогодні більшість задач виконують через інтернет, то такі системи користуються значним попитом.

### Аналіз публікацій

Сьогодні частіше використовують різноманітні додатки, що генерують великі потоки даних, які будуть корисними лише тоді, якщо база даних зможе підтримувати необхідну

швидкість їх обробки. Ці дані, як правило, незмінні, великі за обсягом, впорядковані за часом і агреговані для доступу. Щоб обробити всю цю інформацію, потрібно створити величезну кількість сховищ різного призначення. Сучасні інформаційні системи характеризуються величезними обсягами збережених даних, складною організацією, необхідністю задовольняти різноманітні вимоги численних користувачів. Основою інформаційної системи є бази даних [4]. Якщо аналізувати реляційні та нереляційні бази даних, то очевидно, що NoSQL бази даних вирішують певні проблеми баз даних SQL, зокрема послаблюючи вимоги до узгодженості та доступності [6].

До переваг нереляційних баз даних можна віднести наступне: зберігання великих обсягів погано структурованих даних; швидша обробка та краща масштабованість даних, ніж у реляційних базах даних; використання дешевих кластерів товарних серверів для управління операціями та даними [8]. Недоліки NoSQL баз даних: погана узгодженість даних; відсутність стандартизації [2]. Незважаючи на ці недоліки, нереляційні бази даних все більше будуть застосовуватись при розробленні веб-додатків завдяки здатності до масштабування, що робить їх вигіднішими для розробників, а перевага у швидкодії дозволяє підтримувати необхідну швидкість обробки потоків даних та вчасно реагувати на збої у системі [4].



Рис. 1. Діаграма рівнів

### Структура інтерактивного веб-додатка

Інтерактивний веб-додаток, який представлений у вигляді притулку для тварин, складається з бази даних, сховища даних, прогресивного мобільного додатку та основного веб інтерфейсу. Він містить основні чотири рівні доступу: представлення, бізнес-логіку, рівень доступу до інформації та базу даних. Діаграму рівнів зображено на рис. 1.

Для розроблення інтерактивного веб-додатку було використано Google Firebase [5]. База даних у Firebase – це БД класу NoSQL (нереляційна), яка дає можливість зберігати та синхронізувати дані між скористувачами в реальному часі. Типом бази даних є сховище “ключ-значення”. Це сховище, яке використовує ключ для отримання доступу до значення. Такий підхід знімає всі обмеження реляційної моделі (недостатня продуктивність, трудомістке горизонтальне масштабування) і полегшує можливості збереження та доступу до даних. Такі дані можуть являти собою одиночний об’єкт JSON і разом з тим коректно відповідати на запити по полях.

У базі даних використовують неструктурований підхід, тобто структуру створюють «на льоту», тим самим знімаючи обмеження жорстких зв’язків і пропонуючи різні типи доступу до специфічних значень. Базу даних системи Firebase та її структуру наведено на рис. 2.

Розроблена база являє собою JSON об’єкт, який складається з ключів-значень. Наприклад, Animals містить об’єкт dogs, що складається з об’єктів, ключами яких є унікальні ідентифікатори, що генеруються базою

при додаванні нового об'єкта. Значеннями кожного з ключів є властивості, які містять інформацію відповідного змісту. Однією з властивостей є `url`, що містить посилання на місце збереження фото у сховищі даних Firebase.

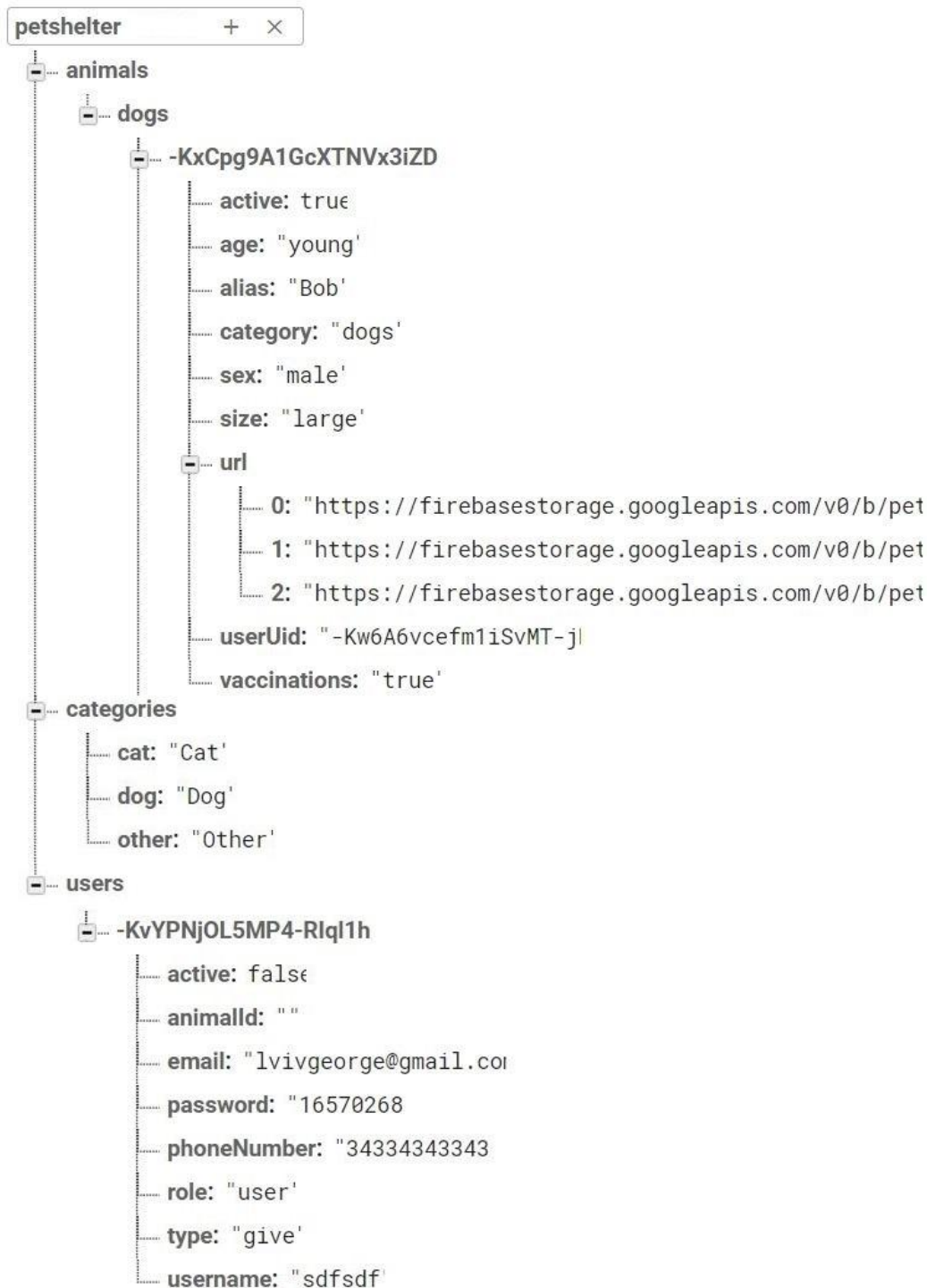


Рис. 2. Структура бази даних системи Firebase

На рис. 2 зображено структуру бази даних. Вона містить 3 об'єкти: `animals` (тварини), `categories` (категорії), `users` (користувачі). Розглянемо детальніше ці об'єкти та їх вкладення.

На рис. 3 наведено структуру об'єкта `animals`. Об'єкт `animals` може містити безліч вкладених об'єктів з назвами категорій тварин, що утримуються в притулку. Візьмемо за приклад вкладений об'єкт `dogs`. `Dogs` складається з об'єктів, ключами яких є унікальний ідентифікатор (`id`), що автоматично генерується Firebase при додаванні об'єкта. По суті, кожен об'єкт, ключем якого є такий `id`, містить властивості, що певним чином описують тварину.



Рис. 3. Структура об'єкта `animals`

На рис. 4 наведено структуру об'єкта `categories`. Він може містити безліч вкладень. У цьому випадку він містить три властивості з назвами категорій: `cat`, `dog`, `other`.



Рис. 4 Структура об'єкту `categories`

Структуру об'єкта `users` наведено на рис. 5. Об'єкт `users` складається з об'єктів, ключами яких є унікальний ідентифікатор (`id`), згенерований Firebase. Цей `id` може збігатися з `userId` з об'єкта `animals.id`. У такому випадку це двосторонній зв'язок. Він зв'яже тварину і користувача, котрий забронював чи залишив її в притулку. Це потрібно для побудови взаємозв'язків між об'єктами та ефективного управління елементами додатка.

У розробленому веб-додатку використовується сховище даних Firebase Cloud Storage. Хмарні технології надають доступ до особистих файлів з будь-якого комп'ютера, що має доступ в Інтернет. Ця технологія дозволяє значно ефективніше керувати, обробляти та працювати з інформацією завдяки централізації цієї самої інформації. Сховище даних (рис. 6) – це об'єкт з даними, який містить `Animals` та `Categories`. Кожна з категорій має дочірні підкатегорії зі значеннями, які можна

отримати за присвоєними унікальними ключами. Схожим чином можна записати нові об'єкти та додати їм властивості, які будуть доступні за новими згенерованими ключами.

Іншою важливою особливістю Firebase є сервіс авторизації (Authentication), який дає змогу створювати акаунти, використовуючи різні сервіси, авторизуватись у додатку та управляти акаунтами. Також у Firebase Authentication є методи, які дозволяють авторизовувати користувача, надсилаючи повідомлення на його телефон. Крім того, цей сервіс може керувати надсиланням повідомлень із скиданням паролю на пошту користувача. Це розширює можливості та покращує умови реєстрації для користувачів, оскільки кожен може вибирати найзручніший спосіб для себе. Реєстрація є швидкою та безпечною. Наступним кроком йде авторизація користувача в додатку. Цей процес теж захищений. Після введення користувачем логіну та паролю відбувається запит до серверу. Якщо той знаходить у себе такі дані (логін та пароль), то повертає одноразовий, зашифрований код (token), завдяки якому користувач потрапляє в додаток і може користуватись усіма доступними для його акаунту функціями.

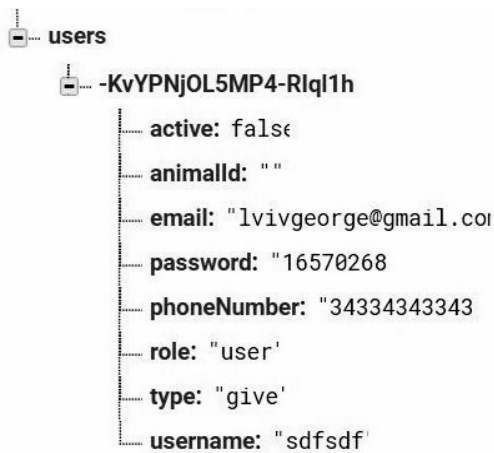


Рис. 5. Структура об'єкта "users"



Рис. 6. Сховище даних

Розроблений інтерактивний веб-додаток містить прогресивний мобільний додаток (PWA), який дає можливість сайту взаємодіяти з користувачем як додаток [7]. Тобто користувач може встановлювати його на будь-який гаджет, отримувати сповіщення і працювати з ним, навіть без інтернету. PWA додаток пропонуватиме встановити його на ваш пристрій вже після другого візиту на сайт. Перевагою PWA є можливість запускати його, використовуючи той самий код, що й для сайту, і не потрібно створювати окремий нативний додаток.

Веб-додаток розроблявся в інтегрованому середовищі для розробки (IDE) – Visual Studio Code. Було створено його структуру для використання зв'язки ReactJS-Redux (рис. 7):

Структура веб-додатка складається з папок із вкладеннями та службових файлів на верхньому рівні. Папка "build" призначена для збереження службових файлів, призначених для побудови та запуску веб-додатка на локальній машині. Також ця папка містить дуже важливий для розроблення файл – webpack.config.js. Webpack Module Bundler – це інструмент для розроблення, який дає можливість зібрати всі .js файли в будь-яку потрібну вам кількість пакетів (бандлів) і впорядкувати їх. Основна ціль Webpack – спростити процес розроблення. Тепер не потрібно писати весь код в одному файлі чи компоувати файли у різні конфігурації для розроблення девелоперами чи для відправлення на продакшн – webpack обробляє всі ці зміни. Хоча існують альтернативи Webpack такі, як Browserify, Webpack все ще залишається найпопулярнішим інструментом на ринку. Його перевагою є велика спільнота розробників, які підтримують та покращують його.

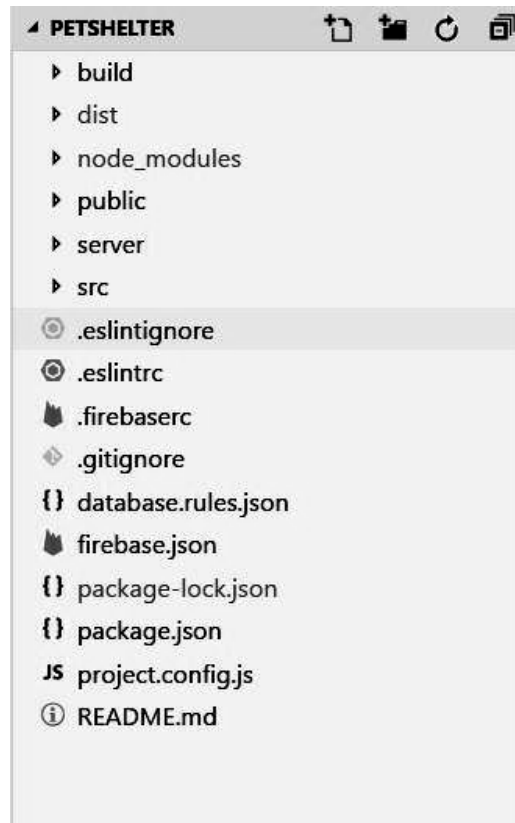


Рис. 7. Структура веб-додатка в середовищі розробки

Папка “dist” призначена для збереження скомпільованого проекту, готового для відправлення на продакшн-сервер. Папка “node\_modules” містить інструменти для розроблення та складання проекту. При відправленні на сервер відправка цієї папки ігнорується. Папка “src” містить всі розроблені компоненти та їхні складові.

Також на верхньому рівні знаходяться технічні та службові файли, які відповідають за встановлення основних функцій бази даних, налаштування пакетів для розроблення та файлів підтримки додатка.

Іншою важливою особливістю розробленого веб-додатка є можливість роботи offline [1]. При першому вході на сайт чи додаток дані зберігаються та кешуються, щоб бути доступними для використання при відсутності інтернету. PWA підтримується і продовжує імплементуватись усіма браузерами. Він є доступний для становлення на усіх пристроях, крім Apple. При цьому, навіть без встановлення, PWA може виконувати свої функції так само, як на сайті.

При розробленні візуального інтерфейсу інтернет-додатка його було максимально оптимізовано так, щоб отримати простий та зрозумілий інтерфейс для користувачів. Для запуску додатка використовується браузер Google Chrome. Головна сторінка додатка може подаватись у десктопному та мобільному виглядах. Практично усі сторінки, які бачить користувач, мають навігацію. Важливо звернути увагу на те, що навігація на десктопній та мобільній версіях відрізняються між собою. У мобільній версії навігація реалізована так званим “бургер-меню” і має два стани: закритий – у вигляді трьох горизонтальних смужок у лівому верхньому куті та відкритий – у вигляді виїжджаючого бічного блоку. Це реалізовано для кращої оптимізації простору на меншому розмірі дисплею. Розроблений інтернет-додаток дає користувачу такі можливості: переглядати та фільтрувати інформацію; відправляти заявку, щоб залишити чи отримати тварину; редагувати дані, пов’язані з тваринами; зберігати, оновлювати та видаляти дані; опрацювати заявки.

## Висновок

Розроблений інтерактивний веб-додаток можна використовувати у різних сферах залежно від заповнення бази даних. Крім того, він є дуже гнучким, оскільки може працювати та швидко підлаштовуватись під різні операційні системи: десктопні (Windows, Linux, MacOS) та мобільні (IOS, Android). Для розроблення додатка було використано такі інструменти, як ReactJS та Redux в ролі користувацького інтерфейсу і Google Firebase в ролі бази даних та сховища даних. Зваживши всі переваги та недоліки нереляційної бази даних, було прийнято рішення обрати саме її. Це дозволило ефективніше використовувати базу у разі збільшення кількості даних та простіше маніпулювати ними. Для забезпечення функціонування додатка за будь-якого сценарію було додано функціонал PWA для можливості роботи без інтернету. Крім цього, PWA дозволяє збільшити конверсію, кількість користувачів та забезпечити зручність використання веб-додатків на мобільних пристроях.

## Список літератури

1. Алекс Бэнкс, Ева Порселло. *React и Redux. Функциональная веб-разработка*. СПб.: Питер, 2018. 336 с.
2. Берковський В. В., Радіонов С. О. Аналіз методів створення інтерактивних веб-додатків // *Системи озброєння і військова техніка*. 2015, № 2(42), С. 61–64.
3. Кравчук О. А. Огляд та аналіз архітектури прогресивних веб-додатків // *Вісник Хмельницького національного університету*, № 5, 2017 (253) – С. 132–136.
4. Швець М. Ю., Заруба Д. С., Хохлов Ю. В. Порівняння SQL та NoSQL баз даних. / *Вчені записки ТНУ імені В. І. Вернадського. Серія: Технічні науки*. Т. 29 (68) ч. 2. № 6, 2018 С. 21–25.
5. Housseem Yahiaoui. *Firestore Cookbook: Over 70 recipes to help you create real-time web and mobile applications with Firestore*. 1st Edition. Birmingham: Packt Publishing, 2017. – 288 p.
6. J. S. van der Veen, B. van der Vaaij, and R. J. Meijer. Sensor data storage performance: SQL or NoSQL, physical or virtual, in *Proceedings of the IEEE 5<sup>th</sup> International Conference on Cloud Computing, June 2012* pp. 431–438.
7. Tal Ater. *Building Progressive Web Apps: Bringing the Power of Native to the Browser*. Sebastopol: “O’Reilly Media, Inc.”, 2017. 288 p.
8. Wodehouse C. *SQL vs. NoSQL Databases: What’s the Difference?* / URL: [www.upwork.com/hiring/data/sql-vs-nosql-databases-whats-the-difference/](http://www.upwork.com/hiring/data/sql-vs-nosql-databases-whats-the-difference/) (дата звернення 12.11.2019).

## References

1. Aleks Bэнks, Eva Porsello. *React y Redux. Funktsionalnaia veb-razrabotka*. Sankt-Peterburh: Piter, 2018. – 336 s.
2. Berkovskyi V. V., Radionov S. O. *Analiz metodiv stvorennia interaktyvnykh veb-dodatkov* // *Systemy ozbroiennia i viiskova tekhnika*, No. 2(42), 2015, S. 61–64.
3. Kravchuk O.A. *Ohliad ta analiz arkhitektury prohresyvnykh veb-dodatkov* // *Visnyk Khmelnytskoho natsionalnoho universytetu*, No. 5, 2017 (253). S. 132–136.
4. Shvets M. Iu., Zaruba D. S., Khokhlov Yu. V. *Porivniannia SQL ta NoSQL baz danykh*. / *Vcheni zapysky TNU imeni V.I. Vernadskoho. Serii: Tekhnichni nauky*. T. 29 (68) ch. 2. No. 6, 2018, S. 21–25.
5. Housseem Yahiaoui. *Firestore Cookbook: Over 70 recipes to help you create real-time web and mobile applications with Firestore*. 1st Edition. – Birmingham: Packt Publishing, 2017. 288 p.
6. J. S. van der Veen, B. van der Vaaij, and R. J. Meijer. Sensor data storage performance: SQL or NoSQL, physical or virtual, in *Proceedings of the IEEE 5<sup>th</sup> International Conference on Cloud Computing, June 2012* pp. 431–438.
7. Tal Ater. *Building Progressive Web Apps: Bringing the Power of Native to the Browser*. – Sebastopol: “O’Reilly Media, Inc.”, 2017. 288 p.
8. Wodehouse C. *SQL vs. NoSQL Databases: What’s the Difference?* / URL: [www.upwork.com/hiring/data/sql-vs-nosql-databases-whats-the-difference/](http://www.upwork.com/hiring/data/sql-vs-nosql-databases-whats-the-difference/) (data zvernennia 12.11.2019).