

competitiveness through scientometrics (edited by Francisco J. Cantú-Ortiz) Boca Raton, FL : Taylor & Francis, 2018. P. 59-80.

2. Martín-Martín, A., Thelwall, M., Orduna-Malea, E., & López-Cózar, E. D.. Google Scholar, Microsoft Academic, Scopus, Dimensions, Web of Science, and OpenCitations' COCI: a multidisciplinary comparison of coverage via citations. *Scientometrics*, Vol. 126, pages871–906(2021).
3. Harzing, A.-W., & Alakangas, S. (2016). Google Scholar, Scopus and the Web of Science: A longitudinal and cross-disciplinary comparison. *Scientometrics*, 106(2), 787–804.
4. Rahdari, B., Brusilovsky, P., Babichenko, D., Littleton, E. B., Patel, R., Fawcett, J., & Blum, Z. (2020). Grapevine: A profile-based exploratory search and recommendation system for finding research advisors. *Proceedings of the Association for Information Science and Technology*, 57(1), e271.
5. Saad-Falcon, J., Shaikh, O., Wang, Z. J., Wright, A. P., Richardson, S., & Chau, D. H. (2020). PeopleMap: Visualization Tool for Mapping Out Researchers using Natural Language Processing. *arXiv preprint arXiv:2006.06105*.
6. Rosen-Zvi, M., Griffiths, T., Steyvers, M., Smith, P., The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence* (2004), AUAI Press, 487--494.
7. D. Blei, A. Ng. and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993-1022. 2003.
8. Jian Jin, Qian Geng, Haikun Mou, Chong Chen. (2018) Author–Subject–Topic model for Reviewer Recommendation, *JIS-Journal of Information Science*, SAGE, 1-16.

УДК 004.05

Присіч А.В., студент 4 курсу
спеціальності 122 «Комп'ютерні науки»
Римар П.В., старший викладач кафедри
комп'ютерних наук та
інформаційних технологій

РОЗРОБКА СЕРВЕРНОЇ ЧАСТИНИ ДЛЯ НАВЧАЛЬНОЇ СИСТЕМИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Клієнт-серверна архітектура постає домінуючою технологією при створенні розподілених мережових програм, забезпечує взаємодію та обмін даними між розробленими програмними продуктами. Сервери і клієнти такої архітектури є незалежними один від одного і функціонують паралельно. Відсутність жорсткої прив'язки між клієнтами та серверами обумовлює можливість ситуації, коли один сервер одночасно обробляє запити від різних клієнтів чи клієнт звертається по чергові до різних серверів.

До переваг клієнт-серверної архітектури відносять зменшення навантаження на клієнтські місця, що, в свою чергу, призводить до зменшення вимог до апаратно-програмного забезпечення 4 клієнтів та зниження вартісних показників системи в цілому.

У своїй роботі у якості платформи для реалізації серверного функціоналу я обрав Firebase. Firebase - це одне з BaaS-рішень (Backend as a Service), яке дає розробнику масу можливостей. Firebase Storage забезпечує надійне завантаження і вивантаження файлів для додатка. Хмарне зберігання файлів відео, аудіо або будь-якого іншого типу підтримується Google Cloud Storage. Вміст хмарного сховища надійно захищене власною системою безпеки.

Налагодження роботи серверної частини включає:

1. Підключення бази даних (Cloud Firestore)
 - a. Написання правил за якими клієнт зможе зчитувати дані з бази
 - b. Створення коректної структури даних для коректної взаємодії з клієнтом
 - c. Створення моделей даних
 - d. Підключення функціоналу до проекту та перевірка коректної роботи
2. Підключення сервісу для аутентифікації користувачів
 - a. Обговорення та обрання методу аутентифікації (через телефон\емейл\соц. мережі\ тощо.)
 - b. Підключення функціоналу до проєкту та перевірка коректної роботи
3. Підключення сервісу для зберігання файлів (Cloud Storage)
 - a. Написання правил за якими клієнт зможе зчитувати дані з бази
 - b. Підключення функціоналу до проєкту та перевірка коректної роботи

Це і сервер, і база даних, і хостинг, і аутентифікація в одній платформі. Так, Firebase Realtime Database надає розробникам API, який синхронізує дані додатки між клієнтами і зберігає їх в хмарному сховищі.

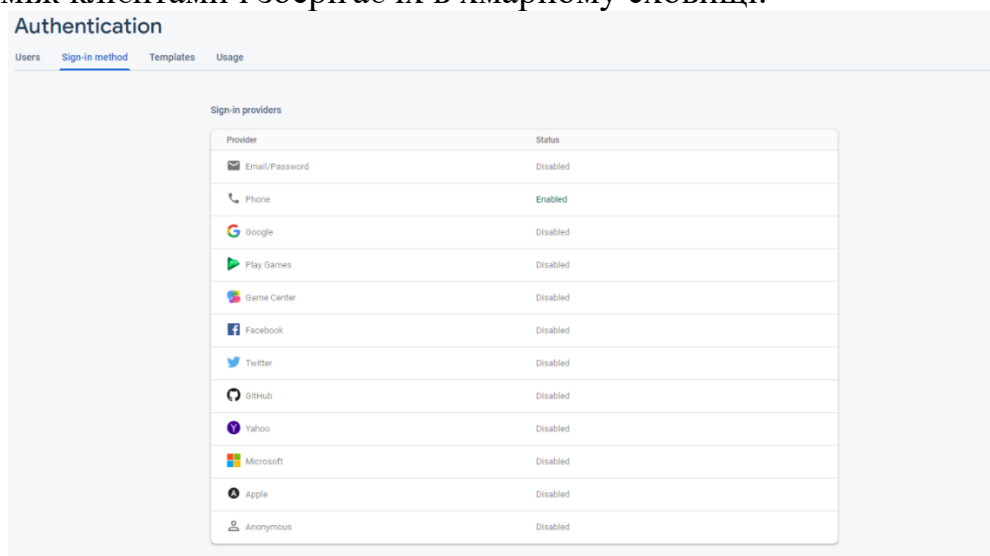


Рисунок 1 – Специфікація

Додаток підключається до бази даних через WebSocket, який відповідає за синхронізацію даних протягом усього сеансу.

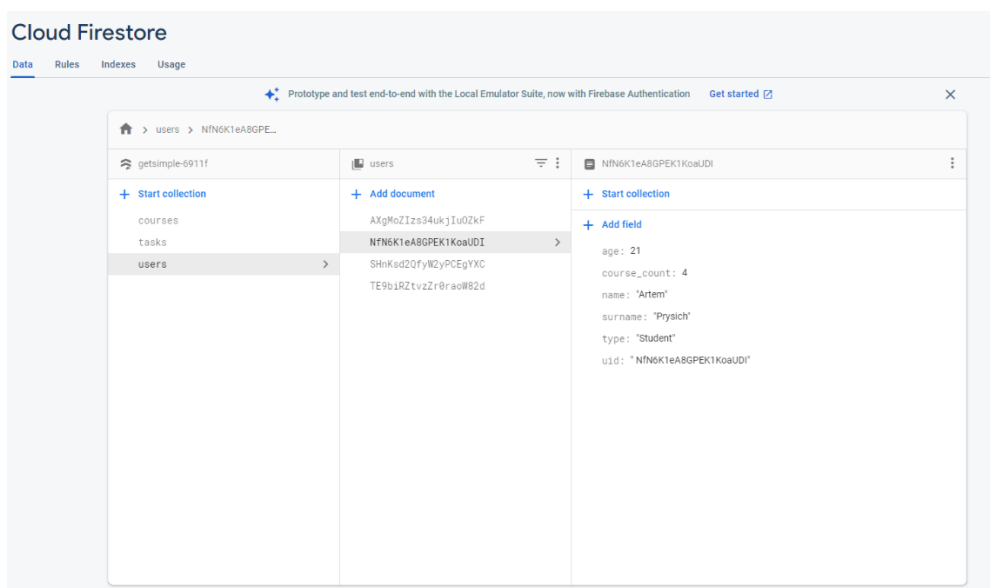


Рисунок 2 – Підключення до бази даних

Серверна частина додатку «Get Simple» була розроблена за допомогою Flutter SDK, та найзручніших і найтехнологічніших інструментів, які доступні на сьогодні в pub.dev та Firebase. Завдяки технологіям Firebase додаток включає в себе весь необхідний функціонал із потужним рівнем захисту даних від розробників платформи Google. Ця платформа надала змогу швидко інтегрувати функціонал аутентифікації, хмарової бази даних, і відкинула необхідність розкривати свій власний сервер для поширення додатку у мережі інтернет.

Список використаної літератури

1. Mobile App Usage Surged 40% During COVID-19 Pandemic. URL: <https://www.appannie.com/en/insights/market-data/mobile-app-usage-surged-40-during-covid-19-pandemic/>
2. 8 причин, щоб полюбити Figma. URL: <https://artjoker.ua/ru/blog/8-prichin-chtoby-polyubit-figma-tak-zhe-silno-kak-my/>
3. Введення в мову програмування Dart. URL: <https://metanit.com/dart/tutorial/1.1.php>
4. Dart documentation. URL: <https://dart.dev/guides>

УДК 323.17

*Прямухін Н. В. д.е.н, професор
кафедри політології та державного
управління*

УПРАВЛІННЯ СОЦІАЛЬНО-ПОЛІТИЧНИМИ РИЗИКАМИ В УМОВАХ ГЛОБАЛІЗАЦІЇ

Донецький національний університет імені Василя Стуса