Enterprises and Organizations of Ukraine, address, full name and name of the legal entity [3]. This can be done with the help of ready-made software products, such as YouControl and Opendatabot [3; 4]. Instructions for their use are available on the state web application diia.data.gov.ua.

The development of an app to support and monitor volunteering meets the current needs and challenges in the volunteer sector, especially during the war in Ukraine, and can be used to improve the performance of organizations and engage more people in charitable activities. The use of modern technologies and tools will make working with the application convenient and accessible to all stakeholders.

**REFERENCES**


1. Аналіз українського волонтерства на основі методології нових соціальних рухів. [Електронний ресурс] //niss.gov.ua Режим доступу: https://niss.gov.ua/news/komentari-ekspertiv/analiz-ukrayinskoho-volonterstva-na-osnovi-metodolohiyi-novykh-sotsialnykh
2. What Drives Volunteers to Accept a Digital Platform That Supports NGO Projects? [Електронний ресурс] // www.frontiersin.org Режим доступу: https://www.frontiersin.org/articles/10.3389/fpsyg.2020.00429/full
3. Відкриті дані та благодійність: як перевіряти волонтерів та благодійні організації? [Електронний ресурс] // https://diia.data.gov.ua. Режим доступу: https://diia.data.gov.ua/info-center/vidkriti-dani-ta-blagodijnist-yak-pereviryati-volonteriv-ta-blagodijni-organizaciyi
4. Перевірка псевдо благодійників: як не віддати гроші аферистам.[Електронний ресурс] // https://youcontrol.com.ua. Режим доступу: https://youcontrol.com.ua/articles/perevirka-psevdo-blahodiynykiv/
5. The Role of Technology in Facilitating Volunteer Engagement [Електронний ресурс] // https://www.firstlaregional.com. Режим доступу: https://www.firstlaregional.com/the-role-of-technology-in-facilitating-volunteer-engagement/



D. Hruzin, O. Lytvynov, O. Hurko


## METHODS FOR OPTIMIZING THE LOADING AND UPDATING OF WEB PAGES USING CLOUD TECHNOLOGIES

An integral aspect of contemporary systems using cloud technologies, crucial for determining system performance and efficiency, lies in the utilization of caching methods and technologies. For systems delivering static information to end-users, technologies like Static Site Generation (SSG) [4] and Server-side Rendering (SSR) [3] are employed in constructing the client side. SSG is typically suited for pages with infrequent content changes, as modifying content on one page necessitates

the rebuilding of the entire site. To cache SSR application pages, it is recommended to utilize a multi-level cache [5] and a cache proxy server [6]. Upon a client's page request, the CDN service receives the request, and if a not-expired version of the page is available, it is delivered to the client. If the page is absent or its time-to-live has expired, the request is redirected to the cache proxy server, which possesses a larger cache size and longer time-to-live for pages. In case the cache proxy server lacks a not-expired version, the page is requested from the server, rendered, sent to the client, and updated on the cache proxy server. An illustration of such an approach is the implementation of Next.js Serverless [2].

However, due to the weak caching model employed by Next.js Serverless, delivering up-to-date data to end-users often incurs a substantial time delay, negatively impacting user interaction.

This scenario can be exemplified in a healthcare system aiming to provide users with public profiles of doctors, their practices, and reviews. Changes occur when content managers modify doctor profiles or practices, or when patients add reviews. Utilizing a Cache Proxy Server with a weak caching model partially mitigates the issue by introducing a second level of managed cache. Nevertheless, some pages may remain unchanged for extended periods, while others undergo constant updates. But the cache time-to-live is set to the same value for all pages, which leads to extra updates of rare-changed pages and delays in data delivery for frequently updated pages. This work is dedicated to addressing this challenge.

A proposed solution involves a flexible and efficient approach for generating and caching static pages using a cache proxy server. Invalidation for data updates by the cache proxy server occurs upon each change in the system's state. Similar to the Cache Proxy Server approach, we suggest rendering and storing pages in a static repository but with a strong caching model, pre-storing all possible pages in advance. Data freshness in the storage is ensured by regenerating pages with each system change affecting them. Handlers subscribed to events about changes in the system's data determine affected pages and invoke the page renderer service to update these pages in the cache proxy server. After successful rendering and saving, a request for invalidating the CDN cache is sent. This approach integrates seamlessly with event-source systems [1], dispatching events about system changes after each update.

When a user's browser sends a page retrieval request, it goes to the CDN, which fetches data from the repository and serves the pre-rendered page to the client.

This approach significantly accelerates the delivery of up-to-date data by immediately invalidating the cache after a system change. In other words, pages regularly receiving new content will be updated regularly, while those without recent changes will not undergo a rerender.

Results from the conducted experiment demonstrate that the proposed approach effectively addresses the identified problem without compromising performance compared to the Next.js Serverless approach.

**REFERENCES**


1. Chamberlain, N. Applying Domain Driven Design with CQRS and Event Sourcing, 2017 [Electronic resource]. – Access mode: https://buildplease.com/products/fpc/
2. Running Next.js applications with Serverless services on AWS. URL: https://serverlessland.com/repos/nextjs-serverless-architecture.
3. Sun Y. (2019).Server-Side Rendering: Building Reliable, High-Performance Web Apps Using Elm-Inspired Architecture, Event Pub-Sub, and Components. Practical Application Development with AppRun,191-217. DOI:10.1007/978-1-4842-4069-4_9.
4. Vepsäläinen J., Vuorimaa P.(2022).Bridging Static Site Generation with the Dynamic Web. In: Di Noia, T., Ko, IY., Schedl, M., Ardito, C. (Ed.) *Web Engineering. ICWE 2022. Lecture Notes in Computer Science*, vol. 13362. DOI: 10.1007/978-3-031-09917-5_32. ISBN: 978-3-031-09916-8.
5. Vilas J.F., Pazos-Arias J.J., Vilas A.F. (2006). Optimizing Web Services Performance Using Cache. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 713-720. DOI: 10.20965/jaciii.2006.p0713.
6. Wang J. A Scalable Efficient Robust Adaptive (SERA) Architecture for the Next Generation of Web Service, 1999. URL: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=2b811e56c98c22ad3bf03267e6c9e6ba9f54fde0



N. Hryhorenko, S. Karpenko, N. Kondratiuk, O. Posudiievska


## DEVELOPMENT OF TECHNOLOGIES FOR NON-ALCOHOL AND LOW-ALCOHOL BEVERAGES CONTAINING FRUIT JUICES AND VITAMIN AND MINERAL COMPLEXES

Recent research from the National Institute for Health and Care Research shows an increase in the consumption of non-alcohol and low-alcohol drinks by the adults. However, there should also be mentioned a negative trend of the high cost of such drinks being followed by the increase in health inequalities if people from poorer families can't afford them (according to the data of 2022) [4].