

## РОЗРОБЛЕННЯ КЛІЄНТСЬКОГО ВЕБЗАСТОСУНКУ ДЛЯ ХМАРНИХ СИСТЕМ СКІНЧЕННО-ЕЛЕМЕНТНОГО АНАЛІЗУ

У роботі розглянуто підхід до створення клієнтського вебзастосунок системи скінченно-елементного аналізу з відкритим початковим кодом, призначеного для використання у хмарному середовищі та віддаленого доступу користувачів. Показано, що розвиток сучасного машинобудування неможливий без широкого впровадження систем автоматизованого проектування, важливим завданням яких є моделювання фізичних характеристик об'єктів із застосуванням методу скінченних елементів. Висока вартість відповідного пропрієтарного програмного забезпечення та обмеженість обчислювальних ресурсів стимулюють розроблення хмарних сервісів із відкритим кодом, які надають можливість проведення складних інженерних розрахунків без необхідності використання потужних локальних комп'ютерів.

Запропоновано архітектуру фронтенд-застосунок, який реалізує інтерфейс користувача для створення, редагування та відображення даних для скінченно-елементних розрахунків у зручній інтерактивній формі. Застосунок розроблено на основі бібліотеки React, що забезпечує масштабованість, багатоплатформність і високу швидкодію. У системі реалізовано три основні режими роботи: створення нової задачі, завантаження існуючої та перегляд результатів, отриманих із хмарного серверу. Передача даних між фронтендом і бекендом здійснюється у форматі JSON, що гарантує сумісність із різними обчислювальними системами. Для візуалізації тривимірних моделей застосовано технологію WebGL, яка дає змогу відображати геометрію та результати розрахунку без установаження спеціалізованого програмного забезпечення.

Розроблений вебінтерфейс підтримує завдання складних граничних умов, неоднорідних фізичних властивостей і навантажень у табличній формі, що забезпечує гнучкість під час підготовки розрахункових даних та зменшує ймовірність помилок користувача. Запропонований підхід дає змогу розділити функції графічного інтерфейсу та обчислювального ядра, що створює можливість інтеграції з різними серверними рішеннями та формує основу для побудови відкритої та доступної масштабованої хмарної системи скінченно-елементного аналізу.

**Ключові слова:** метод скінченних елементів, хмарні обчислення, вебзастосунок, фронтенд, мікросервісна архітектура, React, WebGL.

## DEVELOPMENT OF A CLIENT-SIDE WEB APPLICATION FOR CLOUD-BASED FINITE ELEMENT ANALYSIS

The paper presents an approach to developing a client-side web application for a finite element analysis (FEA) system with open-source code, designed for operation in a cloud environment and to provide remote user access. It is demonstrated that the advancement of modern mechanical engineering is impossible without the widespread adoption of computer-aided design (CAD) systems, a key task of which is modeling the physical characteristics of objects using the finite element method (FEM). The high cost of proprietary software solutions and the limited availability of computational resources stimulate the development of open-source cloud services that enable complex engineering calculations without the need to use powerful local computers.

The architecture of a frontend application is proposed, which implements a user interface for creating, editing, and visualizing data for finite element calculations in a convenient interactive form. The application is developed using the React library, which ensures scalability, cross-platform compatibility, and high performance. The system supports three main operating modes: creating a new problem, loading an existing one, and viewing results obtained from a cloud server. Data exchange between the frontend and backend is performed in JSON format, which guarantees compatibility with various computational systems. WebGL technology is used to visualize three-dimensional models, allowing geometry and calculation results to be displayed without installing specialized software.

The developed web interface supports the specification of complex boundary conditions, non-homogeneous physical properties, and loads in tabular form, which ensures flexibility during the preparation of calculation data and reduces the likelihood of user errors. The proposed approach enables the separation of graphical interface functions from the computational core, creating opportunities for integration with various server-side solutions and forming a foundation for building an open, accessible, and scalable cloud-based finite element analysis system.

**Key words:** finite element method, cloud computing, web application, frontend, microservice architecture, React, WebGL.

### Постановка проблеми

Оскільки проприетарні системи скінченно-елементного аналізу сьогодні є дуже дорогими, актуальним є завдання розроблення відповідного ПЗ із відкритим початковим кодом, яке б могло працювати як хмарний сервіс. Це завдання може бути поділено на два окремі складники: створення фронтенд (frontend) і бекенд (backend). Фронтенд у цьому контексті – це вебзастосунок, який допомагає користувачу задавати параметри скінченно-елементного розрахунку (дискретну модель, фізичні параметри, крайові умови тощо), що потім передаються бекенд-серверу, що працює у хмарі, для обробки. Бекенд безпосередньо виконує усі скінченно-елементні розрахунки і надсилає отриманий результат фронтенд-застосунку для аналізу користувачеві. Серед відомих систем скінченно-елементного аналізу, що працюють у хмарі, можна, наприклад, відзначити SPARSELAB [1], SimScale [2] та ін. [3].

Незважаючи на існування такого ПЗ, завдання створення нових хмарних сервісів для скінченно-елементного аналізу є актуальним, оскільки постійно змінюються можливості сучасних комп'ютерів та вимоги до таких програм. У даній статі розглянуто створення frontend-застосунку з відкритим початковим кодом для автоматизації використання МСЕ у хмарному середовищі.

### Аналіз останніх досліджень і публікацій

Розвиток машинобудування неможливий без широкого використання систем автоматизованого проєктування (САПР), які використовуються для вирішення трьох основних завдань:

- автоматизації підготовки проєктної документації (створення креслень, технічних завдань, проєктів тощо);
- моделювання функціональних характеристик об'єктів проєктування (наприклад, аналіз міцності та довговічності будівельних споруд);
- автоматизації розв'язання задач, що важко формалізуються (складення розкладів; трасування електронних друкованих плат і тому подібне) [4].

Комп'ютерне моделювання фізичних властивостей об'єктів проєктування є одним із найважливіших завдань САПР, оскільки дає змогу істотно пришвидшити й здешевити процес розроблення нової техніки за рахунок часткової заміни натурних фізичних випробувань дослідних зразків комп'ютерною симуляцією.

Одним із найбільш поширених на практиці підходів до моделювання складних фізичних процесів є застосування методу скінченних елементів (МСЕ) [5]. Сьогодні для його використання розроблено велику кількість програмних систем скінченно-елементного аналізу, серед яких можна виділити Ansys [6], MSC Nastran [7], COMSOL Multiphysics [8], FreeFEM [9], OpenFEM [10] та ін. [11; 12].

Незважаючи на велику кількість таких програм у зв'язку з постійним ускладненням проблем, що постають перед машинобудуванням, зростають і вимоги як до наявного програмного забезпечення (ПЗ), так і до обчислювальних систем, на яких воно запускається. Сьогодні потужні комп'ютери є дуже дорогими й складними в обслуговуванні, тому багатьом компаніям стає не вигідно купувати й обслуговувати власні комп'ютерні системи, що призвело до формування нової бізнес-моделі: користувачі ділять обчислювальні потужності віддалених комп'ютерів, які належать третій стороні. Така технологія називається «хмарою», а створення хмарних сервісів та відповідного ПЗ, що підтримують розподіл доступу до обчислювальних ресурсів з оплатою за використання, є актуальним завданням.

Основна ідея хмарної технології полягає у тому, що замість придбання комп'ютерної техніки (або ПЗ) користувач фактично бере його в оренду на певний час. Такий підхід дає такі переваги: замість великих капіталовкладень у відповідне обладнання компанія просто купує час на комп'ютері, що належить третій стороні, і, таким чином, позбавляє себе проблем, пов'язаних із його масштабуванням, обслуговуванням та оновленням. Компанія може придбати безпосередньо обчислювальний ресурс та використовувати його для запуску власного ПЗ, але все частіше нині орендується не лише hardware, а й software. Це називається «програмне забезпечення як послуга» (SaaS – software as a service).

### Мета дослідження

Метою роботи є розроблення архітектури та програмної реалізації клієнтського вебзастосування з відкритим початковим кодом для системи скінченно-елементного аналізу, адаптованого для роботи у хмарному середовищі.

Об'єктом дослідження є процес розроблення спеціалізованого програмного забезпечення для систем скінченно-елементного аналізу.

Предметом дослідження є архітектура та засоби реалізації клієнтського вебзастосування системи скінченно-елементного аналізу, призначеного для роботи у хмарному середовищі.

### Виклад основного матеріалу дослідження

**Програмна реалізація клієнтського вебзастосування системи скінченно-елементного аналізу.** Практична реалізація фронтенд-застосування передусім базується на виборі певного фреймворку (framework) для веброботи, тобто готового рішення, що дає змогу використовувати ефективну стандартну архітектуру проєкту, що перевірена на практиці. У даній роботі для створення графічного інтерфейсу користувача було використано бібліотеку мови програмування JavaScript із відкритим початковим кодом React [13], яка є багатоплатформною, масштабованою і високопродуктивною.

Загальний інтерфейс створеного вебзастосування наведено на рис. 1.

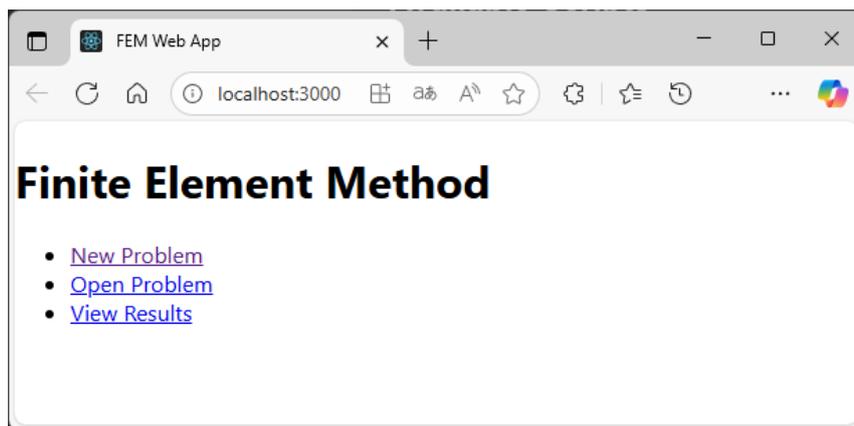


Рис. 1. Фронтенд системи скінченно-елементного аналізу

На початковому екрані клієнтського вебзастосування користувач може вибрати три режими роботи: 1) New Problem – створення нової задачі; 2) Open Problem – завантаження раніше збереженої задачі; 3) View Results – аналіз результатів розрахунку, завантажених із вебсервера, що працює у хмарі. Якщо користувач вибере перший пункт меню, то відкриється сторінка, що містить поля для введення таких параметрів розрахунку, як дискретна модель вихідної області; кількість потоків виконання, що використовуються для розрахунку; точність розрахунку; пружні параметри; крайові умови тощо (рис. 2). Під час завантаження дискретної моделі користувач може її переглянути, вибравши відповідний режим. Для відображення тривимірних графічних сцен (рис. 3) у браузері використовується технологія WebGL [14].

Також слід зазначити, що для завдання складних неоднорідних фізичних характеристик об'єкта розрахунку або граничних умов та навантажень, що діють на нього, підтримується можливість введення відповідних виразів у табличній формі (рис. 4).

Після введення всіх даних користувач, натискаючи на кнопку Calculate, ініціює створення JSON-файлу, що містить усі задані ним параметри розрахунку і якій надсилається серверу у хмарі для обробки.

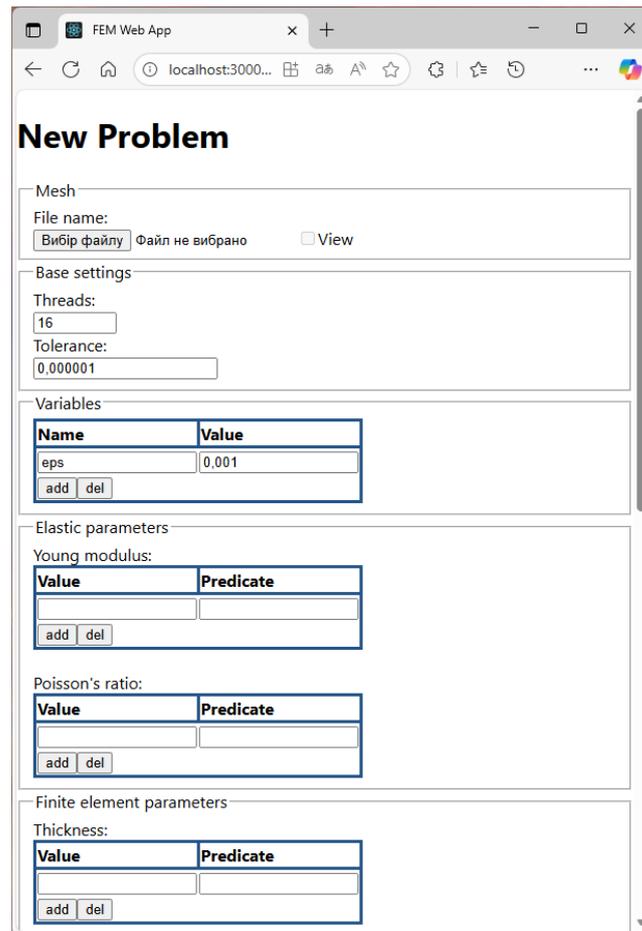


Рис. 2. Графічний інтерфейс користувача для створення нового розрахунку

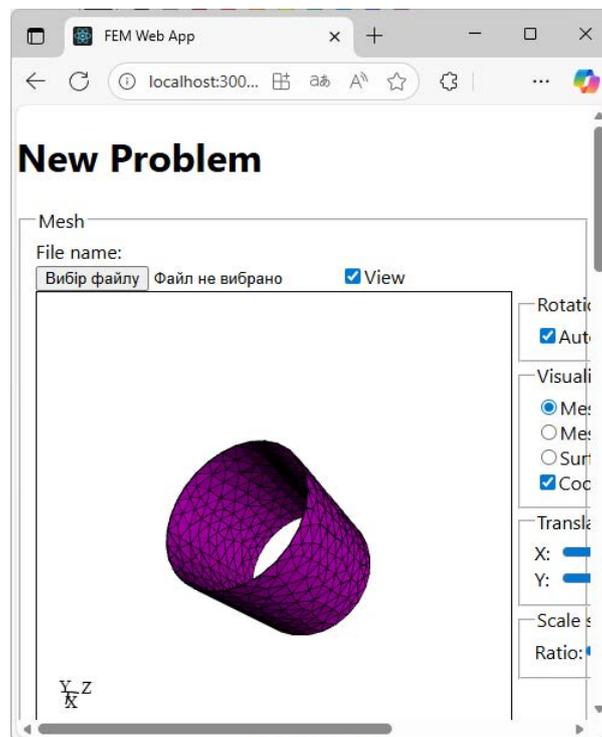


Рис. 3. Візуалізації дискретної моделі у браузері засобами WebGL

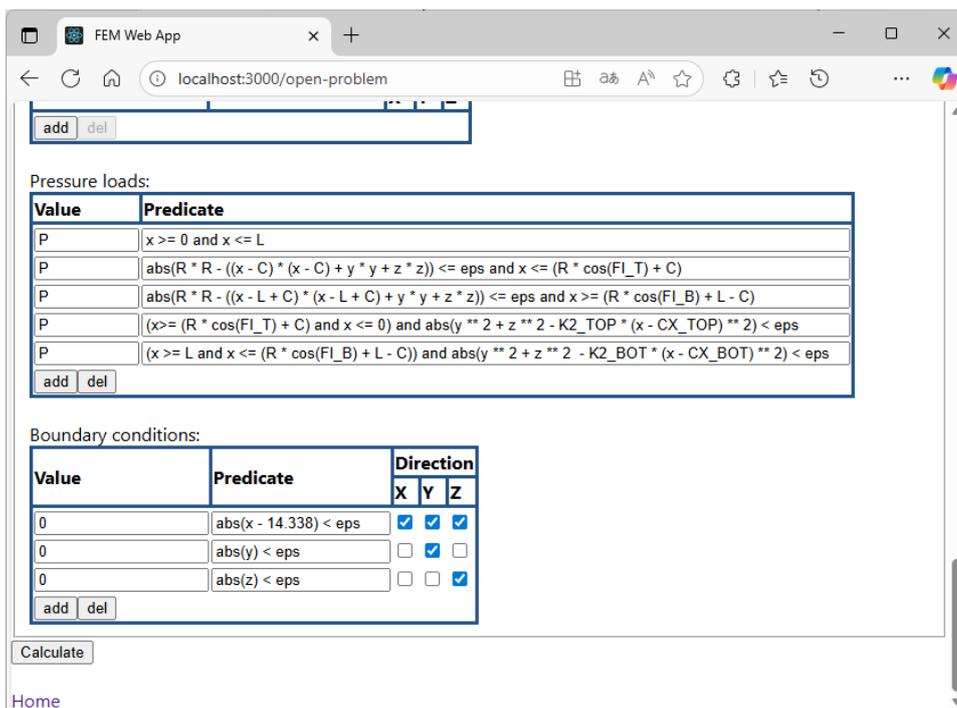


Рис. 4. Приклад завдання складних навантажень, що діють на об'єкт розрахунку

Така ідеологія взаємодії між фронтенд- і бекенд-застосунками дає змогу відокремити реалізацію графічного інтерфейсу користувача від розрахункової частини, що дає можливість використовувати різні сервери (за узгодження формату обміну інформацією).

### Висновки

У роботі представлено підхід до створення клієнтського вебзастосунку системи скінченно-елементного аналізу з відкритим початковим кодом, призначеного для роботи у хмарному середовищі. Показано, що застосування хмарних технологій дає змогу істотно спростити доступ користувачів до обчислювальних ресурсів і підвищити ефективність виконання інженерних розрахунків. Реалізація фронтенд-застосунку на базі бібліотеки React забезпечує багатоплатформність, гнучкість інтерфейсу та можливість масштабування проєкту.

Створений вебінтерфейс дає можливість користувачу формувати вхідні дані для скінченно-елементного розрахунку, задавати фізичні параметри, граничні умови та типи навантажень, а також здійснювати візуалізацію геометричної та дискретної моделей без необхідності встановлення спеціалізованого ПЗ. Передача даних у форматі JSON забезпечує сумісність між клієнтською частиною та обчислювальним сервером, що дає змогу гнучко поєднувати різні реалізації бекенд-систем.

Таким чином, розроблений прототип фронтенд-застосунку може бути основою для побудови повноцінної хмарної системи скінченно-елементного аналізу. Подальші дослідження плануються спрямувати на розширення функціональних можливостей інтерфейсу, інтеграцію з існуючими бібліотеками скінченно-елементного розрахунку та оптимізацію процесу взаємодії між клієнтською і серверною частинами системи.

### Список використаної літератури

1. Finite Element Analysis in the Cloud. URL: <https://sparselab.com/> (дата звернення: 27.10.2025).
2. Structural Mechanics Simulation. SimScale. URL: <https://www.simscale.com/> (дата звернення: 27.10.2025).
3. Finite Element Analysis with Cloud-Based High-Performance Computing. URL: <https://surl.li/htvkqi> (дата звернення: 27.10.2025).

4. Introduction to CAD Systems. URL: <https://surl.li/indifs> (дата звернення: 01.10.2025).
5. Zienkiewicz O.C., Taylor R.L., Zhu J.Z. The Finite Element Method: Its Basis and Fundamentals. Sixth edition. Butterworth-Heinemann, 2016. 753 p.
6. Engineering Simulation & 3D Design Software. Ansys. URL: <https://www.ansys.com/> (дата звернення: 01.10.2025).
7. MSC Nastran – Multidisciplinary Structural Analysis. URL: <http://surl.li/schezo> (дата звернення: 08.10.2025).
8. COMSOL Multiphysics ® Modelling Software. URL: <https://www.comsol.com/> (дата звернення: 08.10.2025).
9. FreeFEM – An open-source PDE Solver using the Finite Element Method. URL: <https://freefem.org/> (дата звернення: 23.10.2025).
10. OpenFEMLibrary. URL: <https://www.sdtools.com/research/openfem/> (дата звернення: 25.10.2025).
11. Best Open-Source Finite Element Analysis Software. URL: <http://surl.li/vmblnr> (дата звернення: 25.10.2025).
12. Finite element analysis software. URL: <http://surl.li/ukbjyp> (дата звернення: 07.11.2025).
13. React. URL: <https://react.dev/> (дата звернення: 28.10.2025).
14. WebGL Fundamentals. URL: <https://webglfundamentals.org/> (дата звернення: 27.10.2025).

### References

1. Sparselab. Finite element analysis in the cloud. Retrieved from <https://sparselab.com/> [in English].
2. SimScale. Structural mechanics simulation. Retrieved from <https://www.simscale.com/> [in English].
3. Finite element analysis with cloud-based high-performance computing. Retrieved from <https://surl.li/htvkqi> [in English].
4. Introduction to CAD Systems. Retrieved from <https://surl.li/indifs> [in English].
5. Zienkiewicz, O.C., Taylor, R.L., & Zhu, J.Z. (2016). The finite element method: Its basis and fundamentals (6th ed.). Butterworth-Heinemann [in English].
6. Ansys. Engineering simulation & 3D design software. Retrieved from <https://www.ansys.com/> [in English].
7. MSC Nastran – Multidisciplinary Structural Analysis. Retrieved from <http://surl.li/schezo> [in English].
8. COMSOL. COMSOL Multiphysics ® modelling software. Retrieved from <https://www.comsol.com/> [in English].
9. FreeFEM. FreeFEM – An open-source PDE solver using the finite element method. Retrieved from <https://freefem.org/> [in English].
10. OpenFEM Library. Retrieved from <https://www.sdtools.com/research/openfem/> [in English].
11. Best open-source finite element analysis software. Retrieved from <http://surl.li/vmblnr> [in English].
12. Finite element analysis software. Retrieved from <http://surl.li/ukbjyp> [in English].
13. React. Retrieved from <https://react.dev/> [in English].
14. WebGL Fundamentals. Retrieved from <https://webglfundamentals.org/> [in English].

Лісняк Андрій Олександрович – к.ф.-м.н., доцент, завідувач кафедри програмної інженерії Запорізького національного університету. E-mail: [a.lisnyak@znu.edu.ua](mailto:a.lisnyak@znu.edu.ua), ORCID: 0000-0001-9669-7858.

Просьяна Дар'я Ігорівна – викладач кафедри комп'ютерних наук Запорізького національного університету. E-mail: [prsjana.daria.i@gmail.com](mailto:prsjana.daria.i@gmail.com), ORCID: 0000-0002-3188-7942.

Lisniak Andrii Oleksandrovysh – Candidate of Physical and Mathematical Sciences, Associate Professor, Head of the Department of Software Engineering of the Zaporizhzhia National University. E-mail: [a.lisnyak@znu.edu.ua](mailto:a.lisnyak@znu.edu.ua), ORCID: 0000-0001-9669-7858.

Prosjana Daria Ihorivna – Lecturer at the Department of Computer Science of the Zaporizhzhia National University. E-mail: [prsjana.daria.i@gmail.com](mailto:prsjana.daria.i@gmail.com), ORCID: 0000-0002-3188-7942.

Отримано: 03.11.2025

Рекомендовано: 09.12.2025

Опубліковано: 30.12.2025

