

## АНОТАЦІЯ

Кваліфікаційна робота включає пояснювальну записку (?с., ? рис., ? додатки).

Об'єкт розробки – Клієнт-серверний веб додаток для управління даними пацієнтів лікарями, перегляду статистики та адміністрування системи.

Розроблена система дозволяє:

- Зареєстрованим користувачам системи (далі Лікар) дозволяється створювати персональну сторінку для кожного пацієнта закріпленим за лікарем, заповняти дані пацієнта на його сторінці. Додавати таку інформацію як діагноз захворювання, стан пацієнта, термін хвороби, ліки від хвороби, симптоми, опис захворювання, загострення хвороби, заносити пацієнта до глобальної статистики системи, переглядати статистику. Додавати до системної бази даних інформацію про нові захворювання та хвороби.
- Незареєстрованим користувачам системи (далі Пацієнт) дозволяється переглядати персональну сторінку створену Лікарем, задавати питання до Лікаря у вигляді емейл листування, переглядати статистику системи. Доступ до персональної сторінки надається через захищений URL з токеном доступу надісланим на емейл Пацієнта.
- Адміністраторам системи дозволяється редагувати будь-яку інформацію та персональні сторінки пацієнтів, видаляти, блокувати та створювати користувачів. Переглядати статистику пацієнтів, лікарів.

Система реалізована з використанням монолітно REST серверу та клієнта для веб перегляду. В процесі розробки була використана мова програмування JavaScript.

Серверна частина реалізована на платформі Node.js з використанням фреймворку Nest.js.

Клієнт частина реалізована з використанням фреймворку React.

Розробка виконувалась в інтегрованому середовищі WebStorm.

Система має автоматизоване розгортання додатку завдяки технології Docker.

Для організації даних була використана документно-орієнтована база даних MongoDB та JavaScript драйвер Mongoose.

В ході виконання дипломного проекту:

- проведено аналіз існуючих рішень в мережі Інтернет;
- проведено аналіз відкритих технологій для задоволення визначених потреб;
- розроблено архітектуру системи.

Використання цієї системи дозволить задовольнити потреби пацієнтів та співробітників лікарень, а використані архітектурні рішення дозволять легко підтримувати та розширювати проект, постійно задовольняючи потреби бізнесу.

Приведені необхідні схеми та документація, підведені підсумки щодо проведеної роботи.

Ключові слова: пацієнт, лікар, моноліт, JavaScript, Node.js, Nest.js, React, REST API, MongoDB, Mongoose, Docker, SOLID.

## ANNOTATION

Qualification work includes an explanatory note (?p., ? fig., ? appendices).

The object of development is a client-server web application for managing patient data by doctors, viewing statistics and system administration.

The developed system allows:

- Registered users of the system (hereinafter referred to as the Doctor) are allowed to create a personal page for each patient assigned to the doctor, to fill in the patient's data on his page. Add information such as disease diagnosis, patient condition, disease duration, disease medication, symptoms, disease description, disease exacerbation, enter patient into global system statistics, view statistics. Add information about new diseases and illnesses to the system database.
- Unregistered users of the system (hereinafter referred to as the Patient) are allowed to view the personal page created by the Doctor, ask questions to the Doctor in the form of e-mail correspondence, view the statistics of the system. Access to the personal page is provided through a secure URL with an access token sent to the Patient's email.
- System administrators are allowed to edit any information and personal pages of patients, delete, block and create users. View statistics of patients and doctors

The system is implemented using a monolithic REST server and client for web browsing. The JavaScript programming language was used in the development process.

The server part is implemented on the Node.js platform using the Nest.js framework.

The client part is implemented using the React framework.

The development was performed in the WebStorm integrated environment.

The system has automated application deployment thanks to Docker technology.

A document-oriented MongoDB database and a Mongoose JavaScript driver were used to organize the data.

During the implementation of the diploma project:

- analysis of existing solutions on the Internet;
- the analysis of open technologies for satisfaction of the certain needs is carried out;
- system architecture is developed.

Using this system will meet the needs of patients and hospital staff, and the architectural solutions used will allow you to easily maintain and expand the project, constantly meeting the needs of the business. Necessary schemes and documentation are resulted, summarized concerning the carried-out work.

Keywords: Patient, Doctor, Monolith, JavaScript, Node.js, Nest.js, React, REST API, MongoDB, Mongoose, Docker, SOLID.