

Одеський національний університет імені І.І.Мечникова

(повне найменування вищого навчального закладу)

Факультет математики, фізики та інформаційних технологій

(повне найменування інституту/факультету)

Кафедра теоретичної механіки

(повна назва кафедри)

## Дипломна робота

на здобуття ступеня вищої освіти «магістр»

на тему: «Розробка та проектування архітектури  
високонавантаженого WEB-додатку»  
«Design and architecture of a high-load web application»

Виконав: студент денної форми навчання  
спеціальності 113 Прикладна математика

Марченко Анастасія Владиславівна

Керівник Асланов Сергій Костянтинович

(науковий ступінь, вчене звання, прізвище та ініціали, підпис)

Рецензент Косирева Ліаліна Анатоліївна

(науковий ступінь, вчене звання, прізвище та ініціали, підпис)

Рекомендовано до захисту:

Захищено на засіданні ЕК №

Протокол засідання кафедри

протокол № \_\_\_\_\_ від \_\_\_\_\_ 2019 р.

№ \_\_\_\_\_ від \_\_\_\_\_ 2019 р.

Оцінка \_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_  
(за національною шкалою, шкалою ECTS, бали)

Завідувач кафедри

Голова ЕК

\_\_\_\_\_  
(підпис)

Асланов С.К.

\_\_\_\_\_  
(підпис)

Лещенко Д.Д.

Одеса – 2019

## ЗМІСТ

Вступ .....	3
1. Постанова завдання.....	4
2. Теоретичні відомості.....	6
2.1. Проектування мікросервісних додатків .....	6
2.2. Технологія побудови серверних додатків.....	11
2.3. Архітектура реляційних баз даних.....	16
3. Програмна реалізація веб-додатка .....	21
3.1. Розробка серверної частини з допомогою Node.js .....	22
3.2. Проектування бази даних .....	36
3.3. Розробка клієнтського додатка на JavaScript.....	38
4. Демонстрація роботи веб-додатка та інструкція по його використанню ..	46
Висновок.....	56
Література.....	57
Додаток .....	58

## ВСТУП

З розвитком веб-технологій на них переносять все більше і більше нетривіальних завдань, для вирішення яких доводиться використовувати різні підходи розробки та проектування. З кінця 2015-го року почали набирати популярність мікросервісні системи для вирішення завдань обчислення і зберігання даних з боку серверів, і SPA-підхід при розробці користувальницьких інтерфейсів.

Мікросервіси — інтерпретація сервіс-орієнтованої архітектури (SOA), яка використовується для створення розподілених програмних систем. Всі сервіси в цій архітектурі, як і в SOA, працюють в рамках однієї мережі для виконання однієї спільної справи [2].

Односторінковий додаток (single page application, SPA) — це веб-додаток або веб-сайт, який використовує єдиний HTML-документ як оболонку для всіх веб-сторінок і організуючий взаємодія з користувачем через динамічно підгружаєми HTML, CSS, JavaScript, зазвичай за допомогою AJAX.

Обидва даних підходу об'єднані в єдину систему і доповнюють один одного вступаючи в архітектурний симбіот. Дана колаборація останнім часом стала підходом для розробки, який є основним і універсальним для будь-якого типу систем. Майже будь-яка з існуючих завдань може бути інтерпретована у вигляді мікросервісів.

Однією з неприємних сторін даного підходу вважають дорожнечу в використанні ресурсів, тому що кожен з сервісів іноді вимагає окремого сервера. Але в контексті невеликих проектів, даний підхід себе виправдовує наявністю доступних способів віртуалізації.

## РОЗДІЛ І.

### ПОСТАНОВКА ЗАВДАННЯ

Розробити повноцінну веб-платформу для здобувачів, яка надає можливість анонімного пошуку пропозицій по роботі, пошуку релевантних кандидатів з використанням фільтрів і анонімної оцінки (оцінювання).

При розробці даної програми необхідно реалізувати наступну функціональність:

1. Можливість реєстрації двох типів аккаунтів - аккаунт для кандидата і аккаунт для роботодавця.
2. Для кандидата повинна бути реалізована функціональність для створення особистого портфолію, куди входять є опис досвіду роботи, можливість додати коротку інформацію про себе і список специфікацій.
3. Для рекрутера / компанії реалізувати можливість вказівки інформації про себе / компанії, регіон і короткого опису статусу компанії.
4. Реалізувати можливість перегляду і фільтрації анонімного списку кандидатів для аккаунта типу рекрутер / компанія.
5. Повинна бути спроектована можливість оцінки аккаунта кандидата з боку аккаунта рекрутера / компанії.
6. Можливість комунікації між акаунтами у вигляді чату з опцією групового чату (один акаунт кандидата - кілька акаунтів рекрутерів / компаній). В якості додаткової опції повинна бути реалізована функціональність обміну файлами.
7. Передбачити можливість ініціації чату тільки з боку аккаунта кандидата.
8. Для аккаунта кандидата - можливість перегляду оцінок профілю з опцією фільтрації.
9. Для обох типів акаунтів реалізувати список контактів.

Необхідно:

1. Додаток реалізувати у вигляді веб-додатка з використанням реляційної бази даних PostgreSQL. Серверну частину розробити з використанням технології NodeJS, клієнтську програму - з використанням фреймворку React.js, що реалізує технологію SPA.
2. При розробці платформи використати методологію побудови мікросервісних додатків.

## ВИСНОВОК

В результаті виконання даної роботи проведено дослідження на тему "Розробка високонавантажених призначених для користувача додатків, заснованих на мікросервісній архітектурі". Розроблена система з закладеними принципами горизонтальної масштабованості за допомогою інструменту контейнерної віртуалізації docker. В ході виконання роботи спроектована реляційна база даних на PostgreSQL з можливостями гнучкої розширюваності в майбутньому. Також розроблено повноцінний користувальницький додаток з функціональністю динамічного підвантаження. Це код, який запускається на стороні клієнта. У призначеному для користувача додатку розроблена архітектура сховища даних на основі імутабельного паттерна Flux, розроблені механізми мутації даних в кожному з модулів сховища, проведена інтеграція сховища зі сторонніми бібліотеками.

В ході виконання роботи побудована екосистема сервісів з використанням підходу контейнерної інкапсуляції і внутрішньо-мережевої взаємодії. При розробці сервісів, які відповідають за серверну частину, використана платформа Node.js та підписочна система черг для спілкування між сервісами. Також розроблений сервіс міграцій з використанням бібліотеки db-migrate для проведення безпечних маніпуляцій над базою даних.

Результатом роботи стала контейнеризована система, яка готова до інсталяції на будь-якій з існуючих платформ і подальшої експлуатації.

## ЛИТЕРАТУРА

1. Sam Newman. Building Microservices.: O'Reilly Media, 2015.
2. [https://ru.wikipedia.org/wiki/Микросервисная\\_архитектура](https://ru.wikipedia.org/wiki/Микросервисная_архитектура)
3. Michael T. Fisher, Martin L. Abbott. The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise, Second Edition.: Addison-Wesley Professional, 2015.
4. [https://developer.mozilla.org/ru/docs/Learn/Server-side/First\\_steps/Introduction](https://developer.mozilla.org/ru/docs/Learn/Server-side/First_steps/Introduction)
5. <https://ru.wikipedia.org/wiki/Docker>
6. Adrian Mouat. Using Docker.: O'Reilly Media, 2015.
7. <https://expressjs.com/ru/guide/using-middlewares.html>
8. C. J. Date. An Introduction to Database Systems (8th Edition). Pearson, 2003.
9. <https://ru.vuejs.org/v2/guide>