Лабораторна робота №5

# Робота з роутами

Кішкін Олександр.

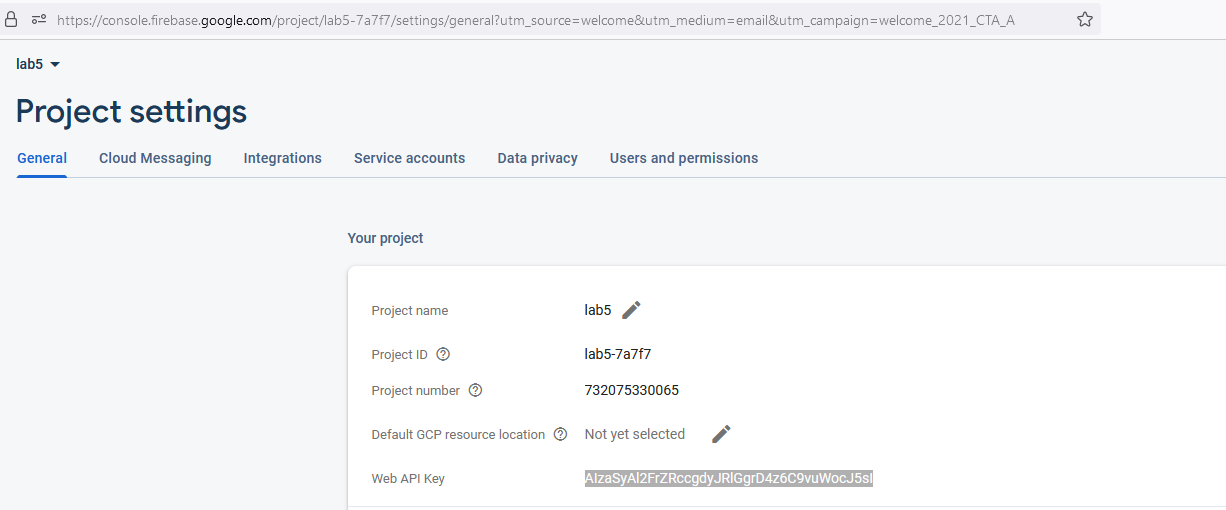
8.1212

### **Крок 1. Налаштування vue-router**

1. Ознайомлення з офіційною бібліотекою маршрутизації vue-router
2. Додати vue-router до проекту з попередньої роботи.
3. Створити сторінку "todo" для додатку з попередньої роботи
4. Створити сторінку "about", яка буде містити інформацію про автора додатку.

### **Крок 2. Налаштування проєкту Firebase Authentication**

1. **Створіть проект Firebase**: - Зайдіть на [Firebase Console](https://console.firebase.google.com/). - Натисніть "Додати проект" або оберіть існуючий проект, якщо він вже створений у вашому обліковому записі Google.
2. **Створіть додаток**: - Додаток може бути мобільним, веб-додатком або іншим. Оберіть Виберіть веб-додаток.
3. **Додайте Firebase до проекту**: - Завантажте конфігураційні файли - Додайтк залажності відповідно до рекомендацій firebase.
4. **Активуйте Firebase Authentication**: - У Firebase Console оберіть "Authentication" в меню та виберіть "Sign-in method". - Активуйте методи автентифікації, які вам потрібні (Обераємо Email/Пароль).
5. **Налаштування автентифікації**: - Налаштуйте параметри, такі як мінімальна довжина паролю, обов'язковість підтвердження електронної пошти і т. д.

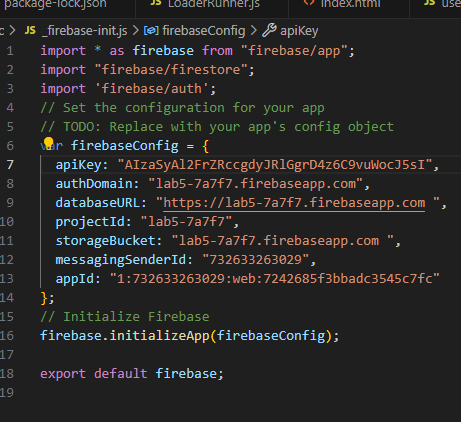


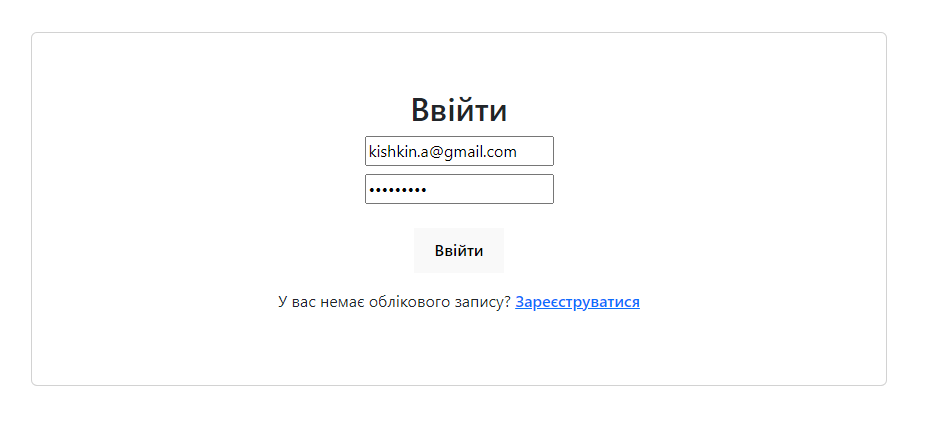
### **Крок 3. Стоворення роутів авторизації та реєстрації**

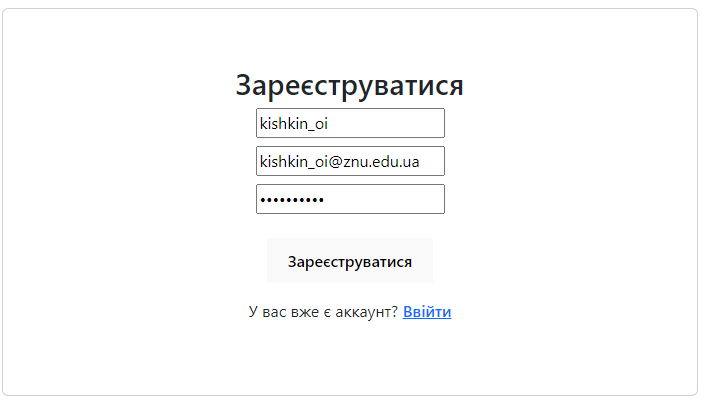
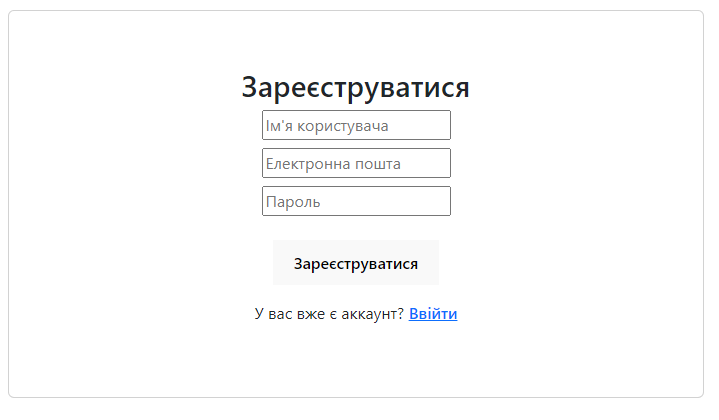
1. **Додайте сторінки авторизації (login) та реєстрації (register)** - Створіть відповідні компоненти та відповідні шляхи для них. - Реалізуйте логіку авторизації та реєстрації з використанням firebase auth. - Після успішної авторизації
2. перенаправляйте користувача на сторінку "todo". - На сторінці "todo" реалізуйте
3. вихід з системи та відображення поточного користувача. - Виконати клонування та запуск проекту: <https://bitbucket.org/munspel/ex6-firebase-auth/src/master/>
4. Спираючись на приклад сторінки авторизації реалізувати сторінку реестрації нових користувачів.
5. Для нових користувачів реалізувати перевірку веріфікації email.
6. Заборонити доступ не верифікованих користувачів.
7. Реалізувати можливість повторної відправки повідомлення для верифікації email.
8. Налаштуйте хук beforeEach для перевірки авторизації користувача. - Якщо користувач не авторизований, перенаправляйте його на сторінку авторизації. - Якщо користувач авторизований, перенаправляйте його на сторінку "todo".
9. Для нових користувачів реалізувати перевірку веріфікації email.
10. Заборонити доступ не верифікованих користувачів.
11. Реалізувати можливість повтоної відпрвки повідомлення для верифікації email.

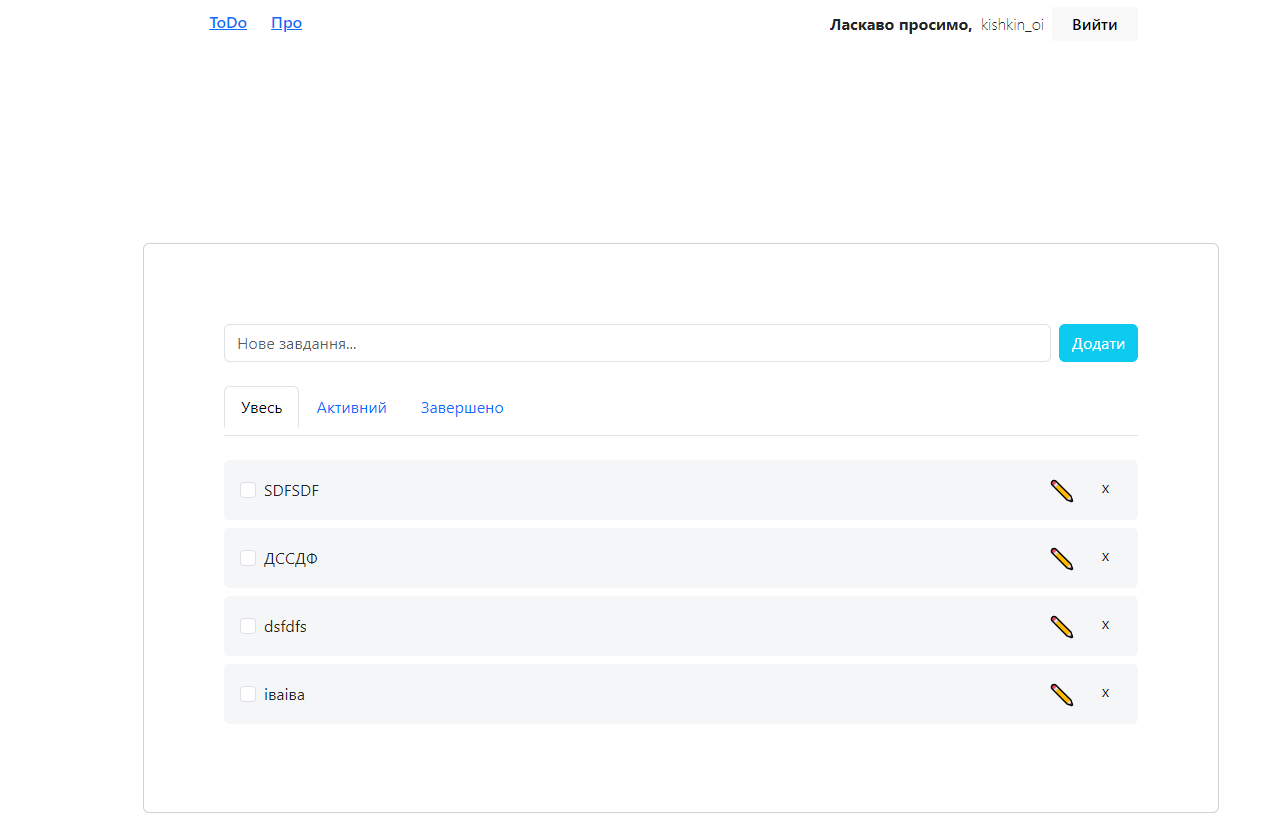
**NOTE:**

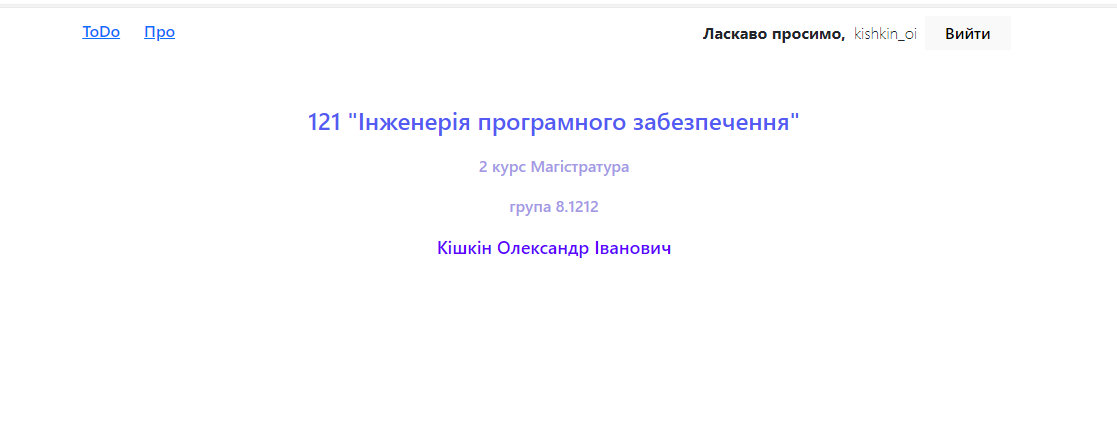
У якості прикладу можна використати проєкт <https://bitbucket.org/munspel/ex6-firebase-auth/src/master/>











**https://github.com/KishkinA/mjsf-mag-lab5**

## **Контрольні запитання:**

* **Що таке та для чого потрібен Vue Router, які є аналоги?**

Vue Router - це офіційна бібліотека для маршрутизації в додатках, побудованих на фреймворку Vue.js. Вона дозволяє створювати одно сторінкові додатки (SPA - Single Page Applications) та керувати різними URL-шляхами в додатку, відображаючи відповідні компоненти Vue на сторінці. Основна мета Vue Router - це допомогти організувати навігацію між різними частинами додатка, які можуть представляти різні розділи, сторінки або вкладки.

За допомогою Vue Router ми можемо виконувати такі завдання:

1. Навігація між різними сторінками вашого додатку без перезавантаження всієї сторінки браузера.
2. Визначення структури URL-шляхів та маршрутів для відповідних компонентів.
3. Передача параметрів в URL та їх отримання в компонентах.
4. Вбудована підтримка динамічних маршрутів.
5. Захист певних маршрутів від доступу без авторизації.
6. Інтеграція з іншими бібліотеками та функціями Vue.js для створення потужних SPA.

Аналогами Vue Router для маршрутизації в односторінкових додатках є такі бібліотеки:

1. **React Router**: Якщо ви працюєте з React, то React Router - це аналог Vue Router для React додатків. Він надає схожий функціонал і способи налаштування маршрутів.
2. **Angular Router**: Якщо ви використовуєте Angular, то Angular Router - це вбудований маршрутизатор для створення SPA в додатках на Angular.
3. **SvelteKit Routing**: Для додатків на Svelte і SvelteKit також є бібліотеки для маршрутизації, такі як SvelteKit Routing, які дозволяють вам керувати маршрутами.

Кожна з цих бібліотек має свої особливості та синтаксис, але основна ідея однакова - допомогти вам керувати маршрутами та навігацією в вашому одно сторінковому додатку. Вибір бібліотеки залежить від фреймворку або бібліотеки, яку ви використовуєте, та ваших конкретних потреб.

* **Які існують UI бібліотеки для Vue та для чого вони потрібні?**

Існує багато UI бібліотек для Vue.js, які допомагають розробникам створювати користувацький інтерфейс (UI) для своїх додатків. Ці бібліотеки надають готові компоненти, стилі та функціональність для швидкої та зручної розробки інтерфейсу. Ось кілька популярних UI бібліотек для Vue та їхнє призначення:

1. **Vuetify**: Vuetify - це розширена бібліотека з багатьма готовими компонентами та стилями відповідно до Material Design. Вона надає розробникам доступ до широкого спектру UI-елементів, таких як кнопки, карти, таблиці, форми, тощо. Vuetify є однією з найпопулярніших бібліотек для створення інтерфейсу Vue.
2. **Element UI**: Element UI - це інша популярна бібліотека, яка надає компоненти та стилі для створення сучасних веб-додатків. Ця бібліотека має велику спільноту користувачів і включає в себе компоненти, такі як таблиці, модальні вікна, меню, форми, тощо.
3. **Quasar Framework**: Quasar - це компрехенсивний фреймворк для Vue, який не тільки надає готові компоненти, але також дозволяє розробляти мобільні додатки, десктоп-додатки та веб-додатки з використанням єдиного коду. Quasar має велику кількість готових компонентів та допомагає вам створювати додатки з кросплатформеною підтримкою.
4. **Bootstrap-Vue**: Bootstrap-Vue - це інтеграція Bootstrap 4 для Vue.js. Вона дозволяє вам використовувати Bootstrap компоненти разом з Vue, і вона надає легку інтеграцію між двома бібліотеками.
5. **Ant Design Vue**: Ant Design - це бібліотека компонентів, створена на основі Ant Design для React. Ant Design Vue надає схожий дизайн та компоненти для Vue.js додатків.
6. **Tailwind CSS і VueTailwind**: Tailwind CSS - це CSS-фреймворк, а VueTailwind надає інтеграцію Tailwind CSS з Vue.js, дозволяючи вам використовувати стилі і компоненти Tailwind разом з Vue.

Ці бібліотеки роблять розробку веб-додатків на Vue.js більш ефективною та швидкою, оскільки вони надають готові компоненти та стилі, які можна легко налаштовувати та використовувати в своєму додатку. Ви можете вибрати ту, яка найкраще відповідає потребам вашого проекту та дизайну.

* Як реалізується механізм авторизації/аутентифікації?

У Vue.js, авторизація або аутентифікація може бути реалізована двома основними методами: за допомогою механізму JWT (JSON Web Tokens) або сесії. Зазвичай цей процес включає наступні етапи:

1. Користувач вводить свої облікові дані.
2. Інформація про користувача перевіряється на сервері для підтвердження ідентифікації.
3. Якщо ідентифікація успішна, сервер генерує токен (JWT), який надсилається на клієнтську сторону.
4. Клієнтська сторона зберігає цей токен та використовує його для доступу до захищених ресурсів.

Цей процес дозволяє користувачам успішно аутентифікувати себе та отримувати доступ до ресурсів після підтвердження їхньої ідентифікації на сервері.

* **Як створювати та працювати з динамічними маршрутами?**

Динамічні маршрути в Vue Router дають можливість створювати маршрути на льоту, використовуючи дані з сервера або введені користувачем. Для створення динамічних маршрутів використовуються динамічні сегменти шляху, які позначаються двокрапками у шаблоні маршруту. Наприклад, маршрут /user/:id дозволяє створювати маршрути з різними значеннями id. Це особливо корисно для сторінок, які потребують відображення динамічних даних або різних ідентифікаторів. Для реалізації цього механізму, ви можете використовувати параметри маршруту та отримувати їх у компоненті, використовуючи this.$route.params.