

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ І ПРИКЛАДНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**КУРСОВА РОБОТА**

Тема: «Розробка Web-додатку для створення і зберігання нотаток»»

**Виконав:**

студент 3 курсу

денної форми навчання

спеціальності: 122 Комп'ютерні науки

освітньої програми «Сучасні інформаційні  
технології та програмування»

Яковенчук Владислав Володимирович

**Керівник:**

Бабаков Р.М

**Оцінка:**

\_\_\_\_\_ (балів)

\_\_\_\_\_ (ECTS)

\_\_\_\_\_ (національна)

**Члени комісії:**

_____	_____
(підпис)	П.І.Б.

_____	_____
(підпис)	П.І.Б.

_____	_____
(підпис)	П.І.Б.

ВСТУП.....	5
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ ТА ОГЛЯД ВЖЕ ІСНУЮЧИХ АНАЛОГІВ	
.....	4
Висновок до розділу 1. ....	6
РОЗДІЛ 2. ВИБІР ІНСТРУМЕНТІВ ДЛЯ РОЗРОБКИ .....	7
2.1 JavaScript.....	7
2.2 MERN.....	8
2.3 REACT.JS.....	9
2.4 NODE.JS .....	11
2.5 EXPRESS.JS .....	12
2.6 MONGODB .....	12
2.7 GIT.....	13
Висновок до розділу 2.....	15
РОЗДІЛ 3. Розробка Full-stack Web – додатку «TO.DOO»	
.....	16
3.1 Створення Backend .....	16
3.2 Створення Frontend.....	17
3.2 Створення дизайну .....	18
Висновок до розділу 3. ....	20
ВИСНОВКИ .....	21
СПИСОК ЛІТЕРАТУРИ .....	22

## ВСТУП

У сучасному світі інформація є найціннішим глобальним ресурсом, вона постійно ускладнюється, змінюється. Інформація є базою для прийняття будь-яких рішень.

Сучасним людям доводиться працювати в режимі мультизадачності, та все встигати. Кожного дня в нас з'являються нові задачі, які нам потрібно вирішити.

В такому режимі – постійного потоку інформації, потрібно мати змогу зосередитись на чомусь дійсно важливому та потрібному. На жаль, це не завжди можливо, адже наш мозок має властивість видаляти, не потрібну, на його думку інформацію з нашої пам'яті, для того, аби краще всмоктувати нову. Тому в нас час, як ніколи раніше, нам потрібні інструменти, для того, щоб ефективніше користуватись своїм часом, та вхідною інформацією.

Одним з таких інструментів є task manager, це програми які дозволяють записувати нотатки, вказувати пріоритети, правильно розподіляти наш час. В даній курсовій роботі як вирішив створити саме такий додаток – TO.DOO.

Мета даної курсової роботи полягає у:

- Огляді уже існуючих інструментів;
- Ознайомленні з новими інструментами розробки;
- Створення веб-додатку для кращого керування часом

В ході розробки належить розібратись із побудовою Fullstack Web-додатку на основі каркасу MERN. Вивчення мови програмування JavaScript, та фреймворків на основі даної мови.

Робота складається із 3 розділів, 13 рисунків, 9 літературних джерел, загальний обсяг роботи 22 сторінок.

## **РОЗДІЛ 1**

### **ПОСТАНОВКА ЗАДАЧІ ТА ОГЛЯД ІСНУЮЧИХ TODO – LIST ДОДАТКІВ**

Даний розділ присвячений постановці задачі, а також виборі та опису всіх необхідних інструментів та технологій для розробки todo – list додатку

#### **1.1 Постановка задачі**

Необхідно розробити Web - додаток для створення та зберігання нотаток.

Додаток має містити наступні ключові моменти:

- a) Зручний та інтуїтивно зрозумілий інтерфейс
- b) Швидке створення нотаток
- c) Можливість вказувати пріоритет задачі, відмічати виконані задачі, вказувати автора (виконавця)
- d) Редагування задач (зміна опису, пріоритету, стану),
- e) Підключення до бази даних
- f) Адаптивний, для різних платформ, дизайн
- g) Зв'язок з розробником

Вся інформація про нотатки зберігається в базі даних.

#### **1.2 Огляд вже існуючих аналогів**

Є велика кількість програм типу todo.

Найвідоміші з них:

- a) Google Keep
- b) Microsoft Office OneNote
- c) Evernote
- d) Any.do

Оберемо для огляду додаток, з найбільш схожим інтерфейсом – Any.do



Рисунок 1.1 – Логотип

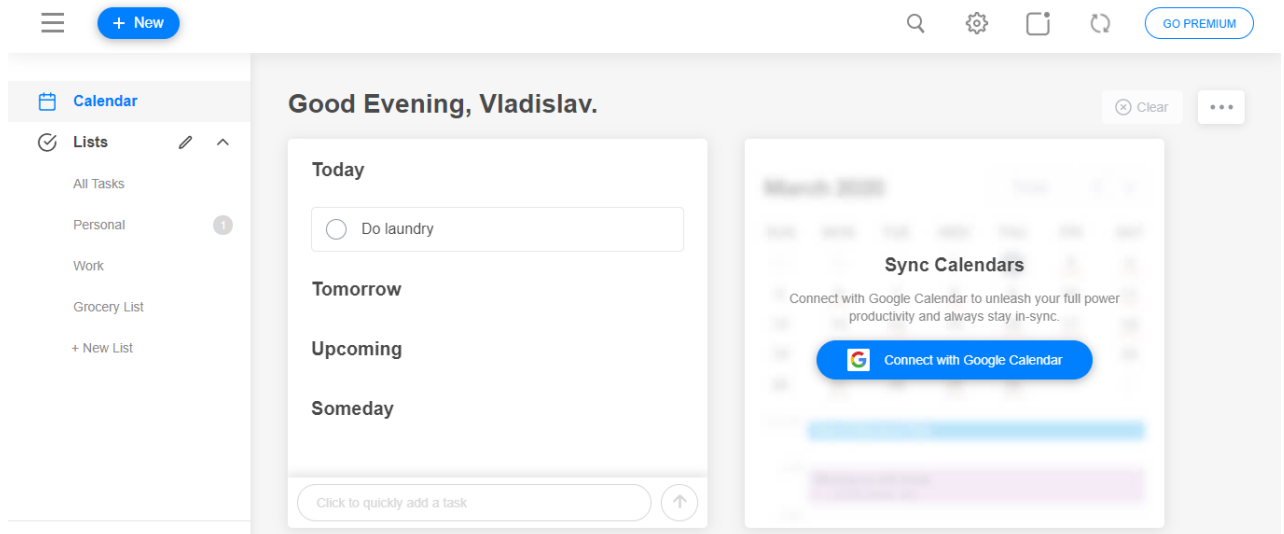


Рисунок 1.2 – Сайт Any.do

Any.do – сервіс, створений в 2011 році, який призначений для створення і зберігання нотаток.

Переваги даного додатку:

- Дизайн
  - Сучасний та інтуїтивно зрозумілий інтерфейс
- Синхронізація

- Миттєва синхронізація ваших нотаток на інших платформах, на яких доступний даний додаток (Web, Android, IOS)
- Є можливість синхронізувати додаток з Google Calendar
- Безкоштовність
  - У додатку доступні платні функції, але більшість функцій безкоштовна
- Функціональність
  - Додаток покриває більшість потреб звичайного користувача
- Стабільність
  - Даний додаток знаходиться на стадії Beta тестування, однак за час його користування я не помітив жодної проблеми
- Кроссплатформеність
  - Існують версії додатку для: Android, IOS, Chrome app, Apple Watch, Windows, та інтеграція з Gmail

## **Висновок до розділу 1**

У даному розділі було розглянуто постановку задачі курсової роботи, окреслено приблизний функціонал майбутнього додатку, а також проведений огляд вже існуючого аналога. Наступний розділ буде присвячений вибору інструментів для розробки.

## Розділ 2

# Вибір та опис інструментів для розробки та розбір функціоналу інструментів

Даний розділ присвячений вибору інструментів для розробки. Зараз існує багато інструментів для створення такого типу додатків. В цьому розділі буде описано які інструменти використовуються, та чому саме вони.

### 2.1 Мова програмування JavaScript

JavaScript – динамічна, високорівнева, об’єктно орієнтована мова програмування. Наряду з HTML та CSS є одною з основних технологій WWW та забезпечує інтерактивні веб-сторінки. Велика кількість веб-сайтів використовують дану JS для поведінки веб-сторінок на стороні клієнта. Всі основні браузерери мають спеціальний механізм для виконання JS коду

JavaScript є домінуючою мовою сценаріїв на стороні клієнта в інтернеті, 95% веб-сайтів використовують його для цієї цілі. Сценарії вбудовуються в документи HTML або взаємодіють з DOM.

Приклади скриптової поведінки:

- Завантаження нового вмісту сторінки без перезавантаження
- Анімація елементів
- Інтерактивний контент
- Перевірка вхідних значень форм
- Передача інформації про поведінку користувача для аналітики

JavaScript відомий великою кількістю бібліотек (фреймворків)

Фреймворк (каркас) – програмна платформа, яка визначає структуру програмної системи. Спрощує розробку та об’єднання різних компонентів великого програмного проекту.

Основною різницею між фреймворком та бібліотекою є те, що бібліотека використовується в програмному продукті як набір підпрограм близької функціональності, не впливає на архітектуру програмного продукту і не накладає на неї ніяких обмежень. Фреймворк диктує свої правила побудови архітектури додатку, встановлюючи на початковому етапі розробки поведінку за замовченням – каркас, який потрібно розширювати та змінювати згідно з вказаними вимогами. Також, на відміну від бібліотеки, яка об'єднує в собі набір близької функціональності, фреймворк може містити велику кількість різних по тематиці бібліотек.

Більшість веб сайтів використовують сторонні бібліотеки JavaScript

- створення односторінкових веб-застосунків (React, AngularJS, Vue.js);
- програмування на стороні сервера (Node.js);
- стаціонарних застосунків (Electron, NW.js);
- мобільних застосунків (React Native, Cordova);

## 2.2 MERN



Рисунок 2.1 - MERN

MERN (MongoDB, Express.js, React.js, Node.js) — набір серверного програмного забезпечення, який використовується для веб-розробки. Всі компоненти даного набору(стеку) підтримують програмування на JavaScript, серверна та клієнтська частина MERN-додатків реалізовується даній мові програмування

Серверна частина додатку реалізується на:

Node.js – для створення серверу



Express.js – для маршрутизації запитів

Клієнтська частина реалізується окремо на бібліотеці React.js

База даних - MongoDB

Для комунікації використовується прикладний програмний інтерфейс – API

Основною перевагою даного стеку є те, що весь додаток написаний на мові програмування JavaScript, як на стороні клієнта, так і на стороні серверу

### 2.3 React – JavaScript бібліотека

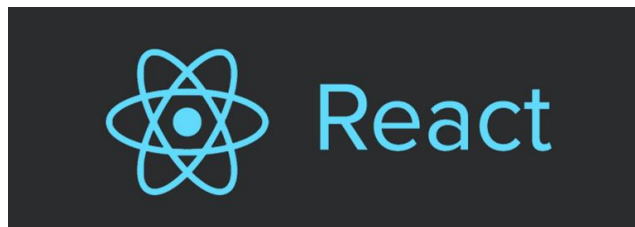


Рисунок 2.2 - React

React – відкрита JavaScript бібліотека для створення інтерфейсів користувача, яка створена з метою вирішувати проблеми часткового оновлення вмісту веб-сторінки, з якими найчастіше стикаються в розробці односторінкових додатків. Розроблена та підтримується Facebook, Instagram, та спільнотою розробників.

React дозволяє розробникам створювати великі веб застосунки, які використовують дані з динамічним змістом, без перезавантаження сторінки. Мета React в тому, щоб бути швидким, простим та масштабованим. Він працює тільки з користувацьким інтерфейсом у додатках. Це відповідає *view* у шаблоні модель – вид – контролер (MVC), і може бути використаний у поєднанні з іншими JavaScript бібліотеками. Найчастіше React, як бібліотеку інтерфейсу використовують разом з Redux.

Особливості React

React використовує віртуальний DOM, а не користується виключно DOM браузера. Це дозволяє йому визначати які частини DOM змінилися, порівняно з збереженою версією віртуального DOM, і таким чином визначити, як найефективніше оновити браузерний DOM. Таким чином здається, що веб-сторінка повністю вся, але бібліотека самостійно вирішує які компоненти потрібно оновити.

Компоненти React зазвичай написані на JSX. Код написаний на JSX компілюється у виклики методів бібліотеки React

Основними перевагами React є:

- Легкий у вивченні
- Віртуальний DOM
- Добре працює при високих навантаженнях
- Велике ком'юніті, та зрозуміла документація
- Добре підходить для командної розробки

Деякі недоліки:

- Відсутність підтримки старих браузерів
- Якщо ваш додаток не має велику кількість динамічних сторінок, вам доведеться писати багато лишнього коду

Для стилізації клієнтської частини додатку, я використовував CSS, та бібліотеку Bootstrap

CSS — це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду. Самі ж сторінки написані мовами розмітки даних.

Bootstrap — це безкоштовний набір інструментів з відкритим кодом, призначений для створення веб-сайтів та веб-додатків, який

містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript. Він спрощує розробку динамічних веб-сайтів і веб-додатків.

## 2.4 Node.js



Рисунок 2.3 – Node.js

Node.js – програмна платформа, яка дозволяє перетворити JavaScript з вузькоспеціалізованої мови в мову загального призначення. Додає можливість взаємодіяти з пристроями вводу – виводу за допомогою власного API (написаного на C++), підключати бібліотеки, написані на різних мовах. Використовується частіше на сервері, виконуючи роль веб-серверу.

Node.js побудований на базі V8, основна особливість якого компіляція коду безпосередньо у власний машинний код, мінаючи стадію проміжного байт-коду

До переваг даної платформи можна віднести:

- Використовується JavaScript
- Бібліотека має широкий стартовий набір можливостей
- Велика кількість зовнішніх бібліотек і готових модулів, використання пакетного менеджера NPM, дозволяє постійно розвивати екосистему Node
- Постійний розвиток

Node.js використовується в таких великих компаніях, як: Reddit, Medium, Wikipedia, Yahoo

## 2.5 Express.js



Рисунок 2.4 – Express.js

Express.js – фреймворк для Node.js. Спроектований для створення веб-додатку та API. По своїй суті, він є стандартним каркасом для Node.js

Фреймворк забезпечує такі механізми

- Написання обробників для запитів
- Встановлює загальні параметри веб-додатків
- Інтеграція механізмів рендерингу view, для генерування відповідей, за допомогою вставки в шаблон

Express надає методи, для визначення конкретного HTTP запиту (GET, POST) та URL шаблону, також для визначення шаблону, який використати для виводу відповіді

## 2.6 MongoDB



Рисунок 2.5 – Express.js

MongoDB – документоорієнтована система управління базами даних, з відкритим кодом, не вимагаючи описування схеми таблиць. Класифікується як NoSQL BD, використовує JSON-подібні документи і схему БД. Написана на мові програмування C++. Використовується у веб розробці

Особливості:

- Зберігає дані в гнучких JSON-подібних документа

- Є розподіленою базою даних, завдяки цьому має високу доступність та легке масштабування
- Динамічні запити

Запис в MongoDB – це документ, який представляє собою структуру даних, вона складається з пар полів та значень, схожа за структурою на JSON. Значення полів можуть включати в себе другі документи, масиви і масиви документів.

Приклад:

```
{  
name: "John",  
age: "20",  
status: "A",  
groups: [ "news", "sports" ]  
}
```

Переваги використання документів:

- Документи відповідають рідним типам даних в різних мовах програмування
- Динамічна схема підтримує плавний поліморфізм

## 2.7 Система контролю версій Git



Рисунок 2.6 - GIT

Git – це розподілена система контролю версій для відстежування змін в коді під час розробки ПЗ. Він призначений для координації роботи програмістів, але може використовуватись для відстежування змін в будь-якому наборі

файлів. Його цілі включають швидкість, цілісність даних та підтримку нелінійних робочих процесів.

Система спроектована як набір програм, спеціально розроблених з врахуванням їхнього використання у скриптах. Це дозволяє зручно створювати спеціалізовані системи управління на базі Git, або користувацькі інтерфейси. Наприклад, Gotio є одним з прикладів фронтенда до репозиторіїв Git.

Система має ряд користувацьких інтерфейсів: наприклад, `gitk` та `git-gui` розповсюджуються з самим Git

Віддалений доступ до репозиторіїв Git забезпечується `git-daemon`, SSH або HTTP сервером.

Принципи роботи:

Локальні операції – більшість дій можна виконувати на локальній файловій системі без використання інтернету. Вся історія змін зберігається локально і при необхідності вивантажується у віддалений репозиторій.

Цілісність даних – Git в своїй базі зберігає все по хешам файлів. Як хешуюча функція використовується SHA-1. Перед кожним збереженням файлів, система обчислює хеш файлу, і він стає індексом файлу в Git. Використовуючи даний метод система легко відслідковує зміни в файлах.

Гілки (branches) - це розмежування від основної лінії розробки. Git дозволяє створити декілька гілок і перемикатись між ними. Це корисно, оскільки дозволяє працювати декільком розробникам над своїм функціоналом не заважаючи іншим і не псуючи основну гілку. За замовченням Git створює гілку з назвою `master`.

Зливання та перебазовування даних – Git підтримує два способи інтеграції змін з гілки в гілку: `merge`(зливання) та `rebase` (перебазовування). Основна різниця полягає в тому, що `rebase` запам'ятовує фіксації, у вигляді патчів, `merge` ж, зливає дві гілки в одну.

## **Висновок до розділу 2**

В цьому розділі було оглянуто усі технології які було використано для розробки веб-додатку та охарактеризовано детально усі можливості , які були корисні для розробки. У наступному розділі буде розглянуто сам процес розробки веб-додатку.

## Розділ 3

### Розробка Full-stack Web – додатку «TO.DOO»

В даному розділі буде описано процес створення веб-додатку, створення дизайну інтерфейсу, описано взаємодії різних частин додатку.

#### 3.1 Створення Backend

В першу чергу потрібно визначити які можливості будуть реалізовані в нашому додатку. Візьмемо за основу CRUD – операції (create, read, update, delete)

Для початку потрібно встановити пакети NPM, такі як Express.js – для створення та читання запитів, mongoose – для взаємодії з базою даних MongoDB, body-parser – для обробки JSON запитів.

Далі ініціалізуємо сервер, та виділимо порт.

Створимо роути для наших операцій, та підключимо базу даних.

Після цього створимо модель задачі, яку ми будемо зберігати в базі даних, та відтворювати на сервері.

```
let Todo = new Schema({
  todo_description: {
    type: String
  },
  todo_responsible: {
    type: String
  },
  todo_priority: {
    type: String
  },
  todo_completed: {
    type: Boolean
  }
});
```

Після цього перейдемо до створення клієнта, нашого веб додатку.

#### 3.2 Створення Frontend

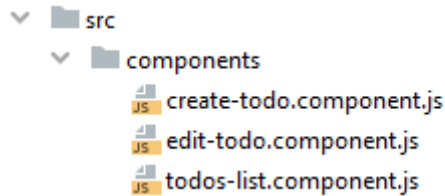
Клієнтську частину ми будемо будувати за допомогою React.js.



React використовує JSX це розширення синтаксису JavaScript, JSX створює React елементи.

Елемент описує те, що ви хочете бачити на екрані. На відміну від DOM-елементів, елементи React – звичайні об’єкти, легкі для створення.

Створимо основні елементи нашого додатку:



create-todo.component.js – елемент, який дозволяє створювати нотатки.

edit-todo.component.js – редагувати.

todos-list.component.js – відображення нотаток на головному екрані.

Після цього імпортуємо наші елементи в головний файл – App.js.

```
import TodosList from './components/todos-list.component';  
import CreateTodo from './components/create-todo.component';  
import EditTodo from './components/edit-todo.component';
```

Та відображаємо їх на сторінці. Далі можемо перейти до створення дизайну.

### 3.3 Створення дизайну

За допомогою програми Figma створюємо логотип нашого додатку.



Рисунок 3.1 - Логотип

Дизайн ми створюємо за допомогою CSS шаблонізатора – Bootstrap.

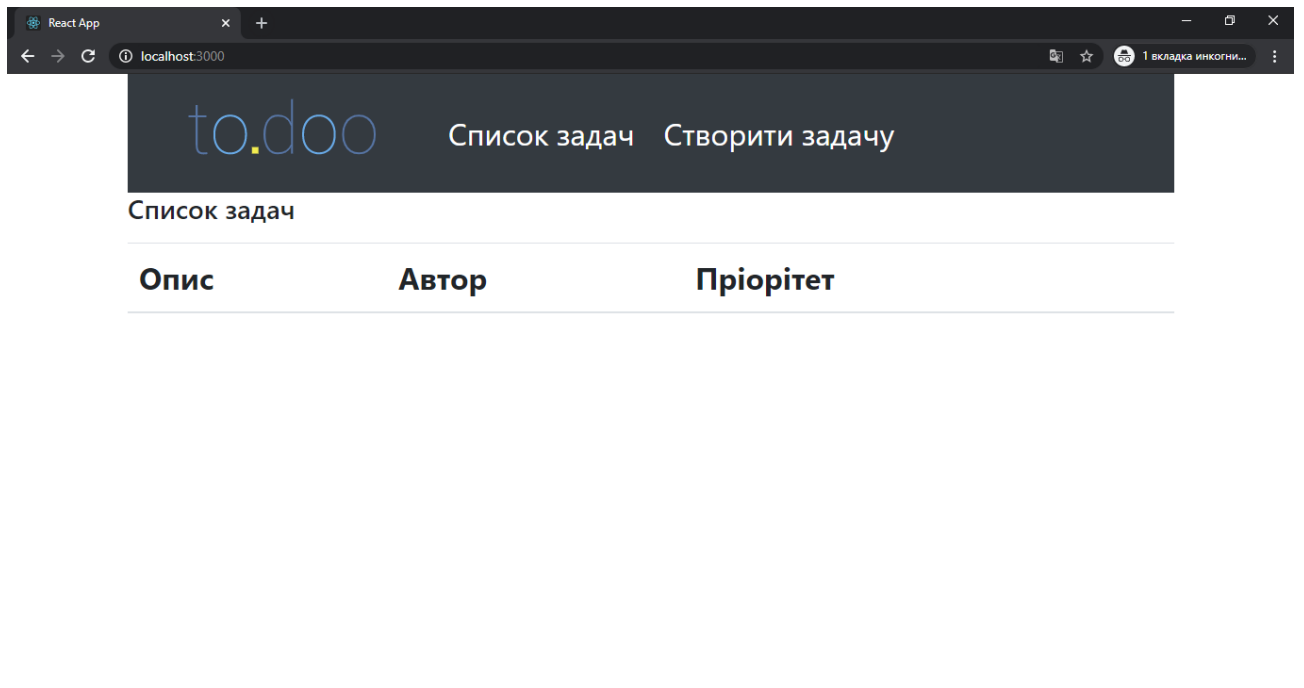


Рисунок 3.2 – Головний екран

Головний екран сайту, поки без запланованих задач.

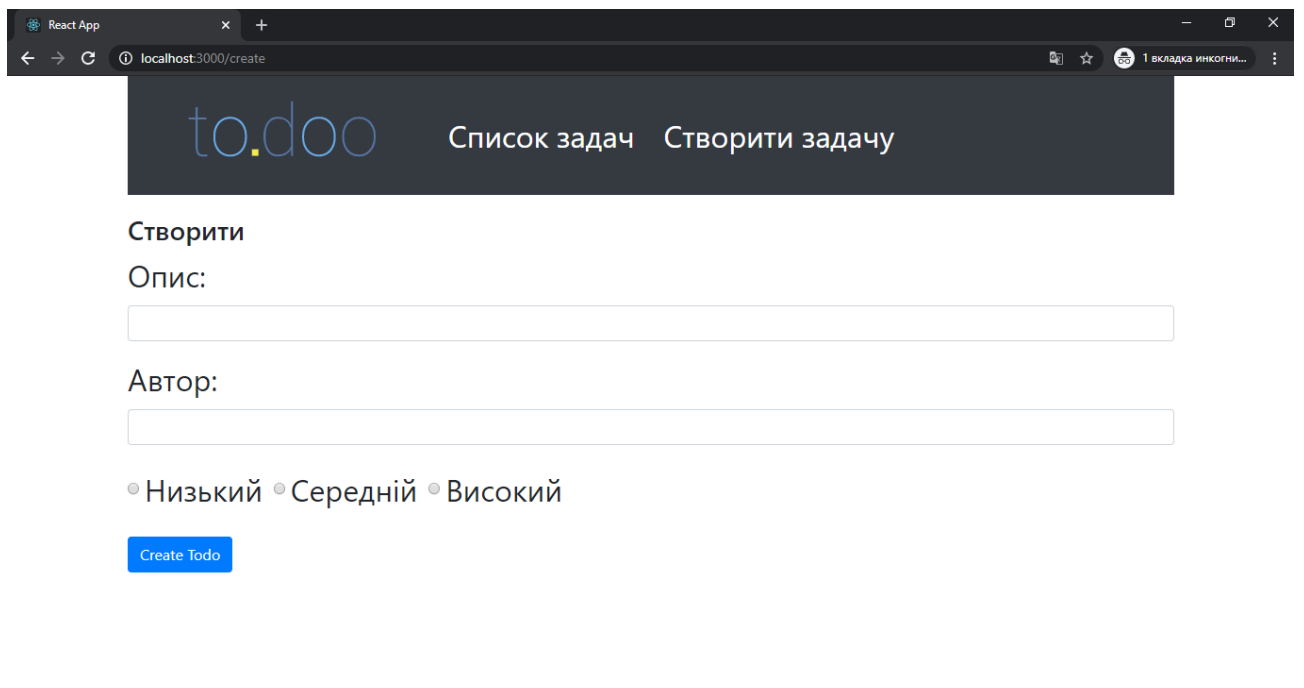


Рисунок 3.3 – Створення задачі

Вікно створення задачі. Можемо описати задачу, автора і пріоритет.

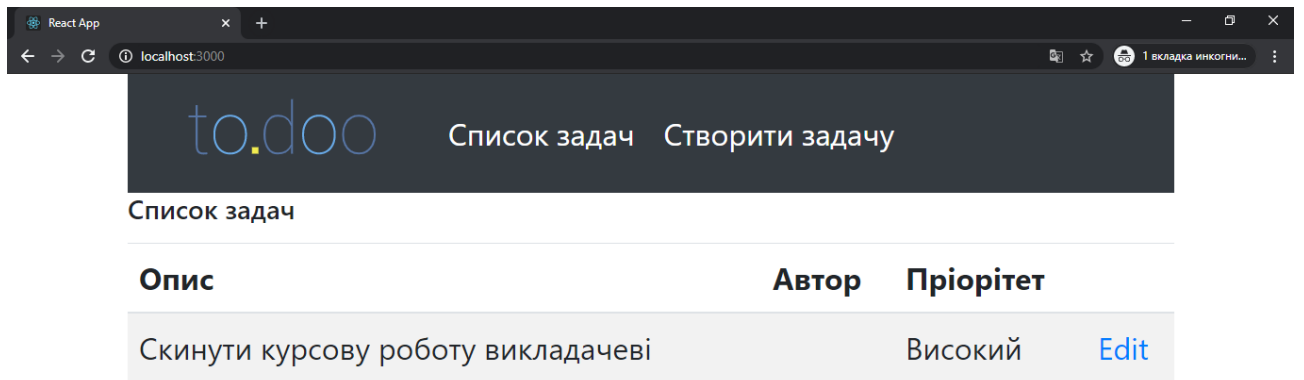


Рисунок 3.4 – Список задач

### Список задач уже з наявним завданням

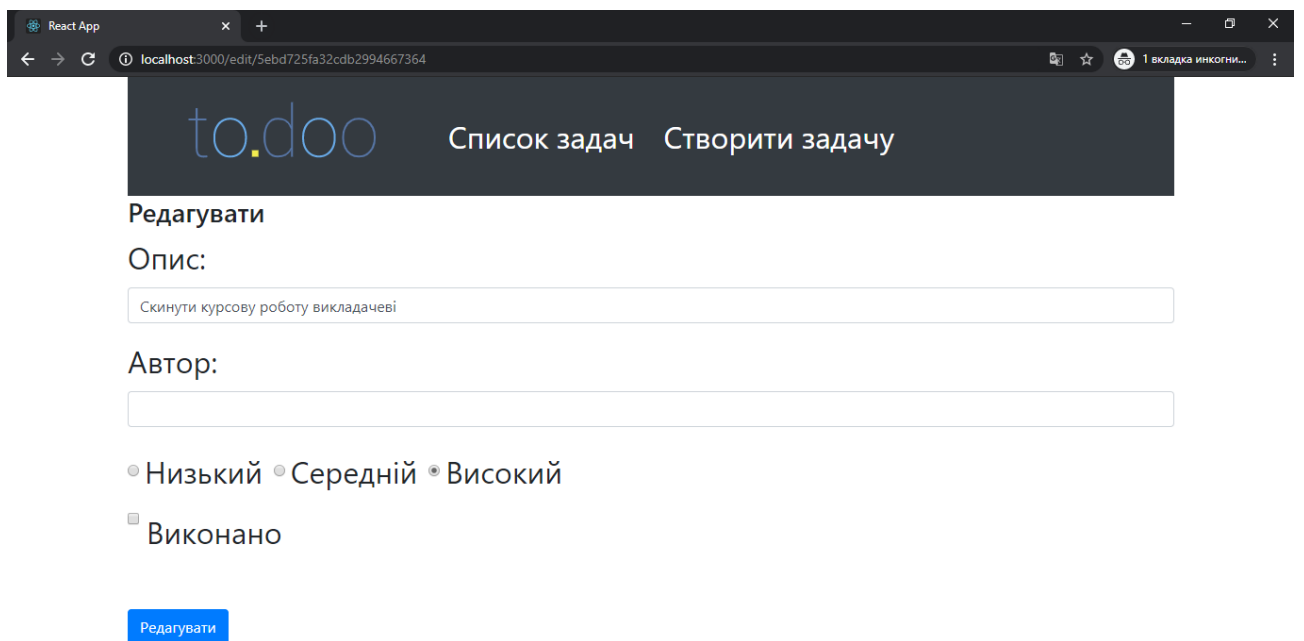
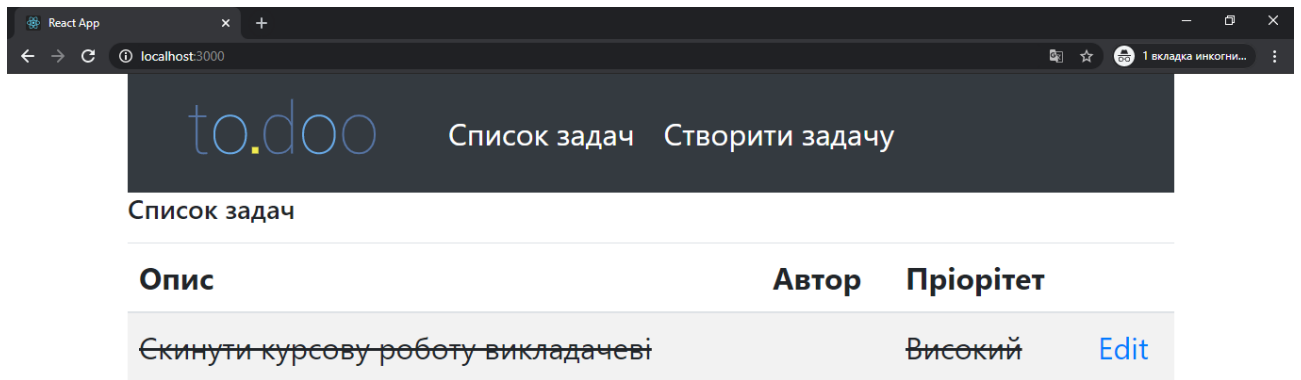


Рисунок 3.5 – Редагування

Вікно редагування. Можливо змінити опис, автора, пріоритет та статус виконання.



Виконане завдання.

### Висновок до розділу 3

В даному розділі було описано процес створення додатку, серверну та клієнтську частину, дизайн. Додано фото інтерфейсу.

## ВИСНОВОК

В ході виконання курсової роботи була детально вивчена тема розробки веб-додатків за допомогою стеку MERN. Використано React для розробки frontend, Node.js для створення серверу, Express.js для побудови запитів, MongoDB як базу даних.

Було розглянуто схожий за типом додаток і перераховано його переваги, деякі з яких я спробував реалізувати в своїй курсовій роботі.

Основним етапом в створенні даного додатку було вивчення мови програмування JavaScript, адже вона є основною в даному стекові.

Даний стек технологій можна використовувати для створення різних за об'ємом веб-додатків. Він може легко масштабуватися при потребі, та не є складним у вивченні завдяки детальній документації.

## СПИСОК ЛІТЕРАТУРИ

1. Додаток Any.do - <https://www.any.do/>
2. MERN - <https://habr.com/ru/company/piter/blog/458096/>
3. Документація Node.js - <https://nodejs.org/uk/docs/>
4. JavaScript - <https://learn.javascript.ru/>
5. Express.js - <https://expressjs.com/ru/>
6. React - <https://reactwarriors.teachable.com/courses/749153/lectures/13564344>
7. React - <https://uk.reactjs.org/>
8. Bootstrap - <https://bootstrap-4.ru/docs/4.3.1/getting-started/introduction/>
9. React - <https://learn.javascript.ru/screencast/react>