

ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
(повне найменування вищого навчального закладу)
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ
(повне найменування інституту, назва факультету (відділення))
КАФЕДРА ПРОГРАМНИХ ЗАСОБІВ І ТЕХНОЛОГІЙ
(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до кваліфікаційної роботи

бакалавра

(освітньо-кваліфікаційний рівень)

на тему: «Розробка фронтенд-інтерфейсу для
ефективного управління завданнями»

Виконав: здобувач 4 курсу, групи 2ПРс

напряму підготовки (спеціальності)

121 «Інженерія програмного забезпечення»

(шифр і назва напряму підготовки, спеціальності)

Тимошук В.І.

(прізвище та ініціали)

Керівник

Захарченко Р.М.

(прізвище та ініціали)

Рецензент

Григорова А.А.

(прізвище та ініціали)

Хмельницький – 2025

ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Інститут, факультет, відділення Інформаційних технологій та дизайну
Кафедра, циклова комісія Програмних засобів і технологій
Освітньо-кваліфікаційний рівень бакалавр
ОПП Програмна інженерія
(шифр і назва)
Спеціальність 121 – Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри
програмних засобів і технологій
к.т.н., доцент О.Є. Огнєва
«__» _____ 2025 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Тимощуку Владиславу Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) «Розробка фронтенд-інтерфейсу для ефективного управління завданнями»»

керівник проекту (роботи) к.т.н., доцент Захарченко Раїса Миколаївна

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «_20_» січня 2025 року №_99-с_

2. Строк подання студентом проекту(роботи) 19.06.2025

3. Вихідні дані до проекту (роботи) _____

Аналіз предметної області, Огляд існуючих рішень, Тестування системи. Висновки

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
Дослідження та аналіз предметної області, проектування фронтенд-інтерфейсу,
програмна реалізація, тестування та оцінка ефективності.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Комп'ютерна презентація

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 05.11.2024 р

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів атестаційного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Отримання завдання	05.11.2024	Виконано
2	Розробка технічного завдання	01.12.2024 – 15.12.2024	Виконано
3	Проектування інтерфейсу	16.12.2024 – 15.01.2025	Виконано
4	Розробка фронтенд-компонентів	16.01.2025 – 22.03.2025	Виконано
5	Оптимізація та доопрацювання	23.03.2025 – 10.04.2025	Виконано
6	Тестування та налагодження	11.04.2025 – 25.04.2025	Виконано
7	Інтеграція з API	26.04.2025 – 15.05.2025	Виконано
8	Підготовка документації	16.05.2025 – 24.05.2025	Виконано
9	Попередній захист і фіналізація	26.05.2025	Виконано
10	Захист кваліфікаційної роботи	19.06.2025	Виконано

Здобувач

(підпис)

Тимощук В.І.

(прізвище та ініціали)

Керівник проекту (роботи)

(підпис)

Захарченко Р.М.

(прізвище та ініціали)

РЕФЕРАТ

Кваліфікаційна робота бакалавра: 97 сторінок, 55 рисунків, 1 додаток, 30 джерел, 2 таблиці.

Мета роботи – створити зручний вебінтерфейс для планування та управління завданнями, який допоможе користувачам ефективно організувати свою роботу. Проєкт зосереджений на розробці вебдодатку, що забезпечує швидкий доступ до завдань, їх сортування та відстеження через інтуїтивний інтерфейс.

Об'єкт дослідження – системи управління завданнями у вебдодатках.

Предмет дослідження – створення фронтенд-інтерфейсу на базі Next.js для організації та управління завданнями.

Робота включає проєктування та кодування компонентів інтерфейсу, зокрема календар для планування із фільтрами за статусом завдань (виконані/невиконані) та категоріями (відображення завдань лише вибраної категорії), списки завдань, статистику виконання та проєкти для спільного доступу кількох користувачів до завдань.

Методи дослідження – вивчення та порівняння існуючих систем управління завданнями (Trello, Asana) для визначення їхніх особливостей і недоліків; аналіз літератури з розробки вебінтерфейсів на Next.js; створення прототипів інтерфейсу в Figma для оцінки зручності; кодування компонентів із використанням Next.js; тестування функціональності, адаптивності та продуктивності на різних пристроях.

Результати роботи:

1. Створено вебінтерфейс на Next.js для ефективного управління завданнями.
2. Реалізовано компоненти: календар із фільтрами за статусом (виконані/невиконані) та категоріями, списки завдань, модуль статистики виконання, проєкти для командної роботи.

3. Забезпечено адаптивність інтерфейсу для коректної роботи на смартфонах, планшетах і ПК.
4. Інтегровано API для синхронізації даних із сервером.
5. Проведено тестування, яке підтвердило зручність і стабільність інтерфейсу.

Новизна роботи: розроблено фронтенд-інтерфейс на базі Next.js із використанням інструментів штучного інтелекту для прискорення створення компонентів, а також реалізовано зручну систему фільтрації завдань за категоріями та статусом, що підвищує ефективність управління задачами для індивідуальних і командних проєктів.

Ключові слова: веб-інтерфейс, управління завданнями, Next.js, адаптивний дизайн, React, компонент, API-інтеграція, штучний інтелект, тестування.

АНОТАЦІЯ

Кваліфікаційна робота бакалавра складається з вступу, чотирьох розділів, висновків, списку використаних джерел та додатків.

Перший розділ «Аналіз предметної області» включає огляд систем управління завданнями, аналіз методів розробки фронтенд-інтерфейсів, проблематику їх створення та існуючі методи оптимізації [24]. Проведено порівняння популярних систем (Trello, Asana), визначено основні виклики у створенні зручних інтерфейсів для управління завданнями та розглянуто сучасні технології веброзробки.

Другий розділ «Проектування фронтенд-інтерфейсу» охоплює вимоги до програмного продукту, розробку інтерфейсу користувача та побудову прототипу додатка. Сформовано функціональні вимоги до веб-додатку, зокрема до календаря з фільтрами за статусом і категоріями, списків завдань, профілю, статистики, проектів і поп-апів. У Figma створено прототип інтерфейсу для оцінки зручності [7].

Третій розділ «Програмна реалізація» присвячено розробці веб-додатку на базі Next.js. Описано технологічний стек і інструменти, реалізацію компонентів інтерфейсу (календар із фільтрами, списки завдань, профіль користувача, статистику виконання, проекти для командної роботи, поп-апи) та інтеграцію з API для обміну даними. Використання інструментів штучного інтелекту дозволило прискорити розробку компонентів.

Четвертий розділ «Тестування та оцінка ефективності» містить налаштування та розгортання веб-додатку, ручне тестування ПЗ, оцінку зручності інтерфейсу та перспективи розвитку. Проведено тестування функціональності всіх компонентів, а також адаптивності на різних пристроях. Оцінено зручність інтерфейсу для користувачів, підтверджено відповідність вимогам і запропоновано ідеї для подальшого вдосконалення.

Робота демонструє створення зручного веб-додатку для ефективного управління завданнями, який відповідає сучасним стандартам фронтенд-розробки та потребам користувачів.

ABSTRACT

The bachelor's qualification work consists of an introduction, four chapters, conclusions, a list of references, and appendices.

The first chapter, "Analysis of the Subject Area," provides an overview of task management systems, examines methods for developing frontend interfaces, identifies challenges in their creation, and explores existing optimization techniques. A comparison of popular systems (Trello, Asana) was conducted, key challenges in designing user-friendly task management interfaces were identified, and modern web development technologies were reviewed.

The second chapter, "Frontend Interface Design," focuses on defining software requirements, designing the user interface, and creating a prototype of the application. Functional requirements for the web application were established, including a calendar with status and category filters, task lists, user profile, statistics, project management, and pop-ups. A prototype was developed in Figma to evaluate usability.

The third chapter, "Software Implementation," details the development of the web application using Next.js. It describes the technology stack, tools, implementation of interface components (calendar with filters, task lists, user profile, performance statistics, team project features, and pop-ups), and integration with an API for data exchange. The use of artificial intelligence tools accelerated the development of components.

The fourth chapter, "Testing and Performance Evaluation," covers the setup and deployment of the web application, manual software testing, usability evaluation, and future development prospects. The functionality of all components and adaptability across various devices were tested. The interface usability was assessed, confirming compliance with requirements and proposing ideas for further improvement.

The work demonstrates the creation of a user-friendly web application for efficient task management, meeting modern frontend development standards and user needs.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	11
ВСТУП	12
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	Ошибка! Закладка не определена.
1.1. Огляд систем управління завданнями	Ошибка! Закладка не определена.
1.2. Аналіз методів розробки фронтенд-інтерфейсів для веб-додатків	Ошибка! Закладка не определена.
1.3. Проблеми фронтенд-інтерфейсів для управління завданнями	Ошибка! Закладка не определена.
1.4. Існуючі моделі та методи оптимізації фронтенд-інтерфейсів	Ошибка! Закладка не определена.
Висновки до розділу 1	30
РОЗДІЛ 2. ПРОЄКТУВАННЯ ФРОНТЕНД-ІНТЕРФЕЙСУ	Ошибка! Закладка не определена.
2.1. Вимоги до програмного продукту та його структури.....	Ошибка! Закладка не определена.
2.2. Розробка інтерфейсу користувача	Ошибка! Закладка не определена.
2.3. Побудова прототипу додатка.....	42
Висновки до розділу 2	Ошибка! Закладка не определена.
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ	Ошибка! Закладка не определена.
3.1. Опис технологічного стеку та інструментів	Ошибка! Закладка не определена.
3.2. Реалізація компонентів інтерфейсу.....	52
3.3. Інтеграція з API для обміну даними.....	75
Висновки до розділу 3	83

	11
РОЗДІЛ 4. ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ.....	85
4.1. Налаштування та розгортання веб-додатку	85
4.2. Ручне тестування ПЗ.....	88
4.3. Оцінка зручності інтерфейсу для користувачів.....	96
4.4. Перспективи розвитку	98
Висновки до розділу 4	99
ВИСНОВКИ.....	100
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	102
ДОДАТОК А.....	104

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ПЗ – Програмне забезпечення;

Next.js — популярний фреймворк для розробки веб-додатків на базі React.

HTML — HyperText Markup Language, мова розмітки, що використовується для створення структури веб-сторінок.

CSS — Cascading Style Sheets, мова стилізації, яка визначає зовнішній вигляд і форматування елементів на веб-сторінках.

JS — JavaScript, мова програмування, що додає інтерактивність і динамічність веб-додаткам.

UI/UX — User Interface/User Experience, термін, що об'єднує дизайн інтерфейсу (UI) та досвід користувача (UX), спрямований на створення зручного й привабливого продукту.

Web-design — веб-дизайн, процес створення візуальної частини веб-сайтів або додатків, включаючи макети, кольори та структуру.

VSCode — Visual Studio Code, потужний редактор коду з відкритим кодом, популярний серед розробників для написання та налагодження програм.

SSR — Server-Side Rendering, технологія рендерингу веб-сторінок на сервері перед відправленням їх користувачу, що покращує швидкість завантаження.

SSG — Static Site Generation, метод генерації статичних HTML-сторінок на етапі збірки, що підвищує продуктивність сайтів.

LCP — Largest Contentful Paint, метрика продуктивності веб-сторінок, що вимірює час завантаження найбільшого видимого елемента, впливаючи на враження користувача.

CRUD — Create, Read, Update, Delete, базовий набір операцій для роботи з даними в додатках (створення, читання, оновлення, видалення).

ВСТУП

Сучасні системи управління завданнями відіграють важливу роль у підвищенні продуктивності як індивідуальних користувачів, так і команд. Зручний вебінтерфейс дозволяє спростити планування, відстеження та організацію завдань, що є особливо актуальним в умовах швидкого ритму життя та зростання попиту на цифрові інструменти. Існуючі платформи, такі як Trello, Asana чи Todoist, пропонують широкий функціонал, однак часто мають обмеження у гнучкості, зручності чи адаптивності до специфічних потреб користувачів. Розробка фронтенд-інтерфейсу, який поєднує інтуїтивний дизайн, швидкодію та сучасні технології, є важливим завданням для забезпечення ефективного управління завданнями.

Фронтенд-розробка, зокрема з використанням фреймворку Next.js, відкриває можливості для створення швидких і адаптивних вебдодатків [11]. Проте створення зручного інтерфейсу пов'язане з низкою викликів: забезпечення простоти використання, швидкого доступу до функцій, адаптивності для різних пристроїв, а також інтеграції з API для обміну даними. У кваліфікаційній роботі розглянуто підходи до вирішення цих проблем шляхом створення вебдодатку з календарем, що підтримує фільтрацію за статусом (виконані/невиконані) та категоріями, списками

завдань, профілем користувача, статистикою виконання, підтримкою командних проектів і сповіщеннями про наближення термінів [15].

Мета дослідження полягає у створенні зручного фронтенд-інтерфейсу на базі Next.js для ефективного управління завданнями, що забезпечує інтуїтивне планування, фільтрацію та відстеження завдань, а також підтримує командну взаємодію.

Для досягнення мети поставлено такі задачі:

- провести аналіз існуючих систем управління завданнями для виявлення їхніх переваг і недоліків;
- дослідити сучасні методи розробки фронтенд-інтерфейсів і проблеми, пов'язані з їх створенням;
- сформулювати вимоги до вебдодатку, включаючи функціонал календаря, фільтрів, профілю, статистики, проектів і сповіщень;
- розробити прототип інтерфейсу у Figma для оцінки зручності дизайну;
- реалізувати компоненти інтерфейсу, такі як календар із фільтрами, списки завдань, профіль, статистику, проекти та поп-апи з нагадуваннями, на базі Next.js;
- провести тестування функціональності та адаптивності вебдодатку на різних пристроях;
- оцінити зручність інтерфейсу для користувачів і запропонувати напрями для його вдосконалення.

Об'єкт дослідження — системи управління завданнями у вебдодатках, які забезпечують планування, організацію та відстеження завдань через зручний інтерфейс.

Предмет дослідження — створення фронтенд-інтерфейсу на базі Next.js для управління завданнями, що включає календар із фільтрами за статусом і категоріями, списки завдань, профіль користувача, статистику виконання, підтримку командних проектів і поп-апи з нагадуваннями.

Перед початком розробки було проведено детальний аналіз предметної області. Досліджено потреби користувачів, зокрема необхідність швидкого

доступу до завдань, фільтрації за категоріями чи статусом, перегляду статистики виконання та організації командної роботи. Визначено вимоги до інтерфейсу: простота, адаптивність для різних пристроїв, швидке завантаження сторінок. Особливу увагу приділено можливостям Next.js для створення ефективних вебдодатків і використанню інструментів штучного інтелекту для прискорення розробки компонентів.

Основними джерелами інформації стали:

- Технічне завдання на розробку вебдодатку.
- Аналіз аналогів, таких як Trello, Asana, Todoist.
- Технічна документація Next.js та інших бібліотек для фронтенд-розробки.
- Відгуки користувачів щодо зручності інтерфейсів подібних систем.
- Публікації про сучасні підходи до веброзробки та застосування ШІ.

На основі аналізу сформовано технічне завдання, яке включає опис функціоналу (календар, фільтри, профіль, статистика, проекти, поп-апи), архітектуру додатку, перелік компонентів і вимоги до продуктивності. Визначено технології розробки, зокрема Next.js для реалізації інтерфейсу та Figma для створення прототипів.

Наукова новизна роботи полягає у розробці фронтенд-інтерфейсу, який поєднує зручний календар із фільтрами за статусом і категоріями, підтримку командних проектів і сповіщення про дедлайни, з використанням Next.js та інструментів ШІ для оптимізації процесу створення компонентів. Ефективність запропонованого рішення підтверджена тестуванням і порівнянням із аналогами.

Практичне значення результатів роботи полягає в можливості використання вебдодатку для організації завдань у різних сферах: навчанні, професійній діяльності, особистих проектах чи командній співпраці. Додаток сприяє підвищенню продуктивності завдяки зручному інтерфейсу та швидкому доступу до функцій.

Теоретичне значення роботи полягає в аналізі сучасних методів фронтенд-розробки, дослідженні проблем створення зручних інтерфейсів і вивченні можливостей застосування ШІ у веброзробці.

Апробація результатів. Основні положення роботи, зокрема підходи до розробки фронтенд-інтерфейсу та використання ШІ, було представлено науковому керівнику. Результати роботи обговорено з науковим керівником і отримано позитивні відгуки щодо їхньої практичної цінності.