

УДК 004.738.5

Гринчишин Я. - ст. гр. КН-321

*Відокремлений структурний підрозділ "Тернопільський фаховий коледж
ТНТУ імені Івана Пулюя"*

ОПТИМІЗАЦІЯ ПРОДУКТИВНОСТІ ВЕБ-ДОДАТКІВ ЗА ДОПОМОГОЮ REACT TA LAZY LOADING

Наукові керівники: спеціаліст вищої категорії Лісовий В.М.,
спеціаліст вищої категорії Слободян Р.О., к.т.н., спеціаліст вищої категорії
Ляпандра А.С.

Hrynychshyn Y.

*Separate Structural Subdivision «Ternopil Professional College of Ternopil Ivan
Puluj National Technical University»"*

OPTIMIZATION OF WEB APPLICATION PERFORMANCE USING REACT AND LAZY LOADING

Supervisors: Lisovyi V., Slobodian R., Lyapandra A.

Ключові слова: React, lazy loading, продуктивність

Keywords: React, lazy loading, performance

Сучасні веб-додатки дедалі частіше стикаються з вимогами до швидкості завантаження та ефективного використання ресурсів. Бібліотека React, побудована на JavaScript, дозволяє розробляти інтерактивні інтерфейси, однак великі проекти можуть мати проблеми з продуктивністю через об'ємний початковий бандл. У цій роботі досліджено методи оптимізації за допомогою техніки lazy loading, яка передбачає відкладене завантаження компонентів. Основним інструментом є React.lazy(), що дозволяє динамічно імпортувати модулі лише тоді, коли вони потрібні користувачу. Наприклад, у додатку з кількома сторінками (профіль, налаштування, головна) компоненти профілю та налаштувань завантажуються асинхронно. Це зменшує розмір початкового бандлу на 20-30%, залежно від складності проекту.

Для забезпечення плавного переходу між станами використовується компонент Suspense, який відображає тимчасовий вміст (наприклад, спінер) під час завантаження. Таке рішення покращує користувацький досвід, адже сторінка стає доступною швидше. У тестах із додатком на 10 компонентів час першого рендерингу скоротився на 15%.

Додаткову оптимізацію забезпечує інтеграція з CSS. Наприклад, анімації та стилі для відкладених компонентів завантажуються разом із ними, що знижує навантаження на основний потік браузера [4]. У порівнянні з традиційним підходом, де всі стилі завантажуються одразу, це дає економію до 10% часу обробки. Також була розглянута проблематика повторного рендерингу. Використання мемоїзації через React.memo() дозволяє уникнути зайвих обчислень для статичних компонентів, що особливо корисно в додатках із великою кількістю даних. У результаті продуктивність зростає ще на 5-7%.

Нарешті, аналіз реальних проєктів показав, що lazy loading ефективний лише за умови правильного розподілу компонентів. Надмірне дроблення може призвести до збільшення кількості запитів, тому оптимальним є групування логічно пов'язаних модулів. Цей підхід протестовано на прототипі системи керування задачами з позитивними результатами.