**Engineer**

# Побудувала тести для Go, SQL і браузера

Побудувала багаторівневий набір автоматизованих тестів — 981 тест Go, 543 фронтендні та браузерні специфікації, 494 поведінкові тести SQL — з мутаційним тестуванням, property-based тестами та обов'язковою перевіркою доступності.

2025

## 981

тестів Go поряд із 543 браузерними та 494 SQL

**Ситуація.** Платформа, що тримає свою бізнес-логіку в базі даних, має проблему з тестуванням, якої немає в більшості проєктів. Логіка написана не тією мовою, з якою добре працює тестовий фреймворк, — вона в SQL, захована всередині функцій, а SQL — саме той код, який зазвичай лишається без тестів, бо тестувати його незручно. Якщо зверху додати API на Go та фронтенд на Next.js, «важливі частини покрито» непомітно перетворюється на «покрито ті частини, які було легко покрити».

**Завдання.** Поведінку кожного рівня треба було перевіряти там, де вона насправді живе, а не перевіряти все ззовні через браузер.

**Дія.** Чотири рівні отримали чотири види тестів. В API на Go — 981 тестова функція у 310 файлах. У фронтенді — 543 специфікації Vitest і Playwright, зокрема 48 файлів наскрізних тестів і 30 браузерних специфікацій. У базі даних — 494 файли поведінкових тестів: 116 876 рядків SQL, що роблять перевірки через 6 654 викинуті винятки, — тож збережену функцію тестують у самій базі даних, а не крізь три рівні застосунку над нею. Над усім цим — тести, які перевіряють самі тести: мутаційне тестування Stryker навмисно ламає рядок коду й падає, якщо цього ніхто не помітив, а fast-check генерує вхідні дані, які нікому не спало на думку записати. Гейт на axe-core перевіряє, що немає жодного порушення WCAG 2.0 і 2.1 рівнів A та

AA, тож проблема з доступністю зупиняє збірку, а не стає знахідкою аудиту через кілька місяців. Пороги покриття рухаються лише вгору. Увесь набір запускається як 10 задач у workflow на 612 рядків.

**Результат.** Змінювати щось структурне вже не страшно, а тільки це й рятує кодову базу такого розміру від закам'яніння. Чесно кажучи, платити доводиться часом: набір повільний, кожна зміна сплачує йому свою данину, а за такого розміру він і сам потребує догляду. Натомість він дає змогу й далі рухатися швидко, а це варте більше за хвилини, які він забирає.

### Застосуйте цей досвід у своєму проєкті

Використайте «Побудувала тести для Go, SQL і браузера» як відправну точку для схожого завдання у вашій організації. Послуги, що стоять за цією роботою, доступні й для вашого проєкту. Та частина, якої користувач не бачить: де зберігаються дані, які діють правила і як інші системи можуть із нею взаємодіяти.

**Переглянути наші послуги** Зв'язатися з нами →

### Дізнайтеся, що привело до результату

Прочитайте про ситуацію, рішення та результат цього прикладу. Ми можемо обговорити, що відповідає вашим умовам, а що потребує іншого підходу. Побудувала тести для Go, SQL і браузера — 981 тестів Go поряд із 543 браузерними та 494 SQL

**Переглянути наші послуги** Зв'язатися з нами →

## Хочете тести, які справді виявляють помилки?

Обговоріть свій проєкт з інженером, відповідальним за виконання. Обсяг, основу розрахунку вартості та передачу результатів узгоджуємо письмово до початку робіт.

Роль у цій роботі: Інженер з тестування та якості. Тут задіяні Backend-розробка, Frontend-розробка, PostgreSQL, DevOps, Автоматизація та CI/CD, Надійність і резервне копіювання та Тестування та QA. І, до речі, ми добре вміємо Backend- та API-розробка, DevOps та автоматизація CI/CD та Технічне лідерство та консалтинг.

**Зв'язатися з нами** Ознайомтеся з нашими послугами

Одна частина **довшого переліку** — і він увесь на цьому сайті.

Кожен запис написано однаково: ситуація, завдання, що ми зробили і що змінилося.

**Переглянути весь перелік**

Визначте потрібне покращення

Опишіть робочий процес, наявні інтерфейси та користувачів, яким потрібна зміна. Ми можемо визначити перевірюваний результат і спосіб передачі, який дасть вашій команді змогу підтримувати його.

**Дивитися, скільки це коштує** Зв'язатися з нами →

## Хочете тести, які справді виявляють помилки?

Запитайте про рішення в роботі «Побудувала тести для Go, SQL і браузера». Ви говоритимете з людьми, які можуть пояснити роботу й оцінити ваш проєкт.

**Переглянути наші послуги**

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

**Дбайливо зроблено з любов'ю — D.**

**CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N**

**Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.**

**Надруковано з https://engineer.company/uk/portfolio/built-a-test-suite-across-go-sql-and-browser/**