

Projekt Snaply

Narzędzie do Zaawansowanego Przechwytywania Ekranu

Snaply to zaawansowane narzędzie do wykonywania zrzutów ekranu, które stworzyłem, aby umożliwić łatwe przechwytywanie zawartości zarówno z urządzeń mobilnych, jak i desktopowych. Aplikacja oferuje szereg funkcji usprawniających pracę programistów, testerów i specjalistów UX/UI.

O Projekcie Snaply

Snaply powstał jako odpowiedź na potrzebę efektywnego dokumentowania interfejsów użytkownika podczas procesu rozwoju oprogramowania. Narzędzie umożliwia automatyczne wykonywanie zrzutów ekranu na różnych urządzeniach, w różnych rozdzielczościach i stanach interfejsu. Jest to nieoceniona pomoc zarówno w fazie projektowej, jak i testowej każdego projektu cyfrowego.

Rozwiązane Problemy

1. Czasochłonny proces ręcznego tworzenia zrzutów ekranu 2. Trudność w przechwytywaniu spójnych zrzutów na różnych urządzeniach 3. Problemy z dokumentowaniem dynamicznych elementów interfejsu 4. Brak narzędzia łączącego przechwytywanie z analizą wizualną 5. Potrzeba szybkiego wykrywania zmian w interfejsie między wersjami

Technologie Wykorzystane w Projekcie

• Python jako główny język programowania • Selenium WebDriver do automatyzacji przeglądarek • Appium do testowania aplikacji mobilnych • OpenCV do analizy obrazu i wykrywania różnic • Flask do stworzenia interfejsu webowego • SQLite do przechowywania metadanych zrzutów • Docker do zapewnienia spójnego środowiska wykonawczego

Kluczowe Funkcjonalności

• Równoczesne wykonywanie zrzutów ekranu na wielu urządzeniach • Automatyczne wykrywanie zmian wizualnych między wersjami • Przechwytywanie pełnej strony z przewijaniem • Izolowanie określonych elementów interfejsu • Harmonogramowanie sesji przechwytywania • Eksport rezultatów w różnych formatach (PNG, PDF, HTML) • Integracja z systemami CI/CD

Wyniki i Efekty

Wdrożenie Snaply przyniosło wymierne korzyści:

- Redukcja czasu poświęconego na dokumentację o 75%
- Wcześniejsze wykrywanie błędów wizualnych w interfejsie
- Usprawnienie procesu akceptacji zmian przez klientów
- Lepszy wgląd w spójność UI na różnych urządzeniach
- Automatyzacja raportowania różnic wizualnych
- Przyspieszenie procesu testowania regresyjnego

Wnioski i Plany Rozwojowe

Stworzenie Snaply było fascynującym wyzwaniem technicznym, które połączyło moje umiejętności w zakresie automatyzacji i przetwarzania obrazu. Projekt jest stale rozwijany, a w planach są:

1. Wdrożenie sztucznej inteligencji do wykrywania istotnych zmian
2. Rozszerzenie obsługi na urządzenia IoT i SmartTV
3. Dodanie funkcji nagrywania interakcji użytkownika
4. Stworzenie rozszerzenia do popularnych środowisk IDE
5. Implementacja chmurowej wersji usługi

Przykłady Zastosowań

Snaply znajduje zastosowanie w wielu scenariuszach zawodowych:

- Testowanie regresyjne interfejsów - automatyczne wykrywanie nieplanowanych zmian
- Dokumentacja produktu - tworzenie zrzutów do instrukcji i materiałów marketingowych
- Kontrola jakości - weryfikacja spójności wizualnej na różnych urządzeniach
- Rozwój responsywnych stron - testowanie wyglądu przy różnych rozdzielczościach
- Współpraca z klientem - łatwe dokumentowanie błędów i propozycji zmian
- Archiwizacja wersji produktu - śledzenie ewolucji interfejsu w czasie

Umiejętności Rozwinięte Podczas Projektu

Umiejętność	Zastosowanie w projekcie
Automatyzacja testów	Tworzenie skryptów do automatycznego przechwytywania ekranu
Przetwarzanie obrazu	Analiza i porównywanie zrzutów ekranu
Frontend/Backend	Budowa interfejsu użytkownika i silnika aplikacji
DevOps	Konteneryzacja i ciągła integracja narzędzia
UX Design	Projektowanie intuicyjnego interfejsu użytkownika

Podsumowanie

Projekt Snaply pokazuje, jak automatyzacja może znacząco usprawnić proces rozwoju oprogramowania i kontroli jakości. Narzędzie to pozwala zaoszczędzić czas, obniżyć koszty i podnieść jakość produktów cyfrowych dzięki precyzyjnemu i systematycznemu podejściu do dokumentacji wizualnej. Jestem dumny, że stworzyłem rozwiązanie, które rozwiązuje realne problemy deweloperów i zespołów QA. Snaply reprezentuje moje podejście do inżynierii oprogramowania: identyfikacja potrzeby, opracowanie

efektywnego rozwiązania i ciągłe doskonalenie.