

*WSB Merito w Gdańsku*

Gdańsk, 27.12.2025

Tadeusz Szuta

70315

# Sprawozdanie z Projektu

Zaawansowane projektowanie aplikacji mobilnych – aplikacja Droplt

## 1. Cel i koncepcja projektu

Celem zadania była realizacja w pełni funkcjonalnego prototypu aplikacji e-commerce typu "Flash Sale" przy użyciu narzędzi opartych na sztucznej inteligencji. Aplikacja Droplt koncentruje się na sprzedaży jednego, limitowanego produktu w modelu "single-product drop", co pozwala na maksymalne uproszczenie ścieżki zakupowej użytkownika (UX).

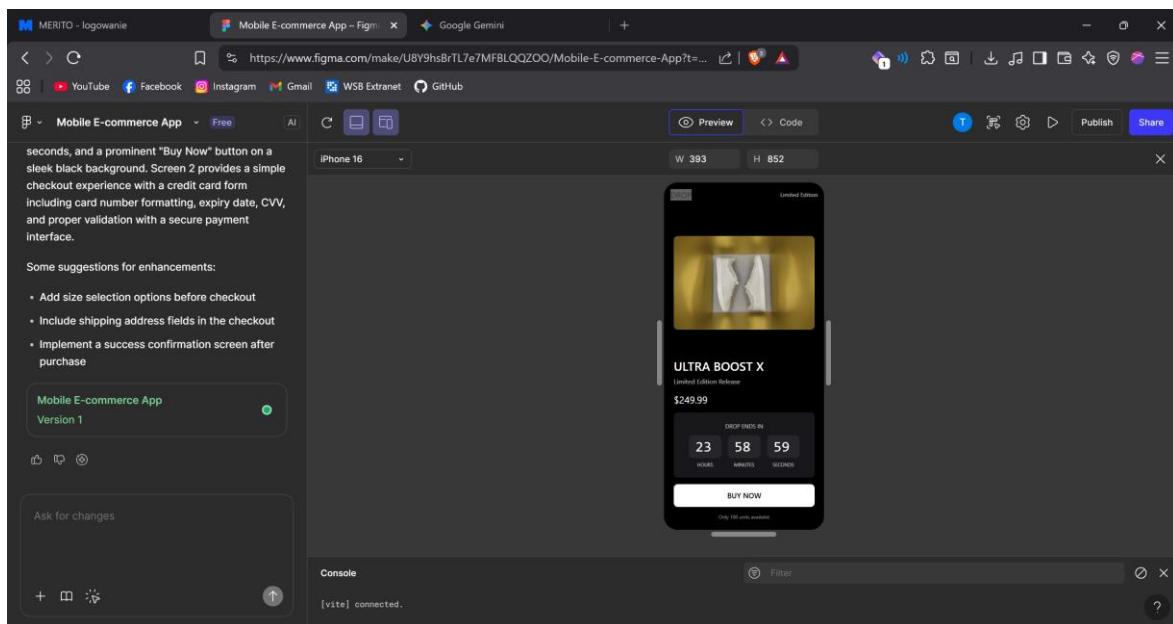
## 2. Front-End: Projektowanie w Figma AI

Warstwa wizualna oraz interfejs użytkownika (UI) zostały opracowane przy wykorzystaniu funkcji generatywnych Figma AI. Pozwoliło to na szybkie wygenerowanie nowoczesnego, wysoko kontrastowego interfejsu zoptymalizowanego pod urządzenie mobilne.

Prompt wykorzystany do generowania interfejsu:

*"Create a mobile e-commerce app for a single product drop. Screen 1: High-quality product image, countdown timer, 'Buy Now' button. Screen 2: Simple checkout with credit card form. Style: Bold, black & yellow, high contrast."*

[Link do projektu interfejsu](#)



## 3. Back-End: Implementacja Gemini CLI

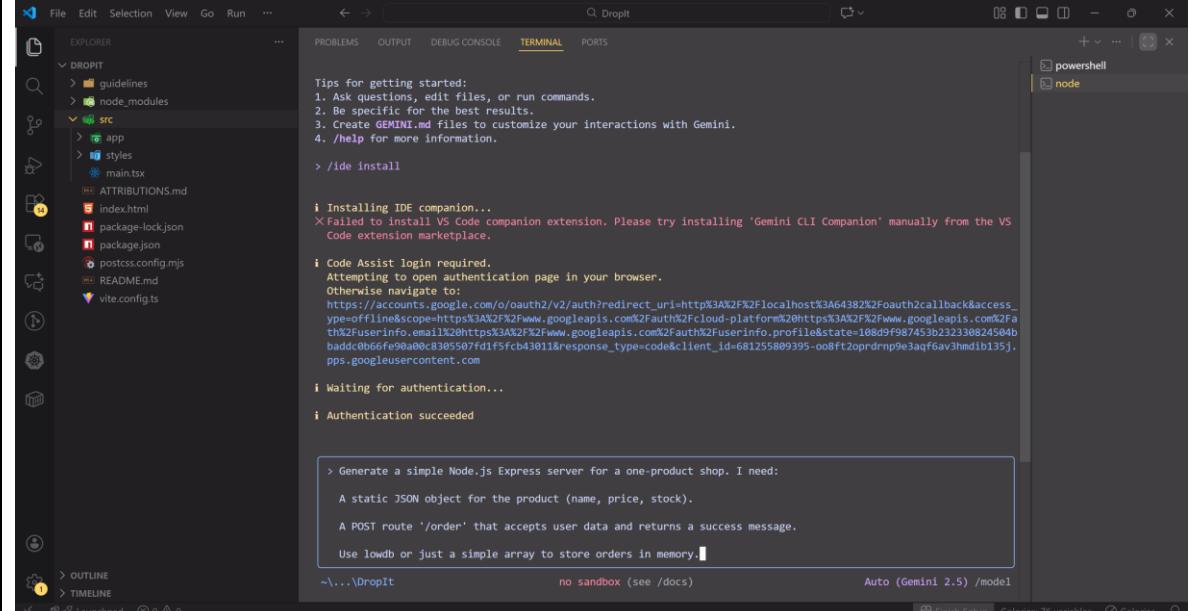
Logika serwerowa została przygotowana w środowisku Node.js przy użyciu frameworka Express.js. Kod backendu został wygenerowany przy pomocy Gemini CLI bezpośrednio w terminalu Visual Studio Code, co pozwoliło na błyskawiczne stworzenie struktury API.

Prompt wykorzystany do generowania logiki serwera:

*"Generate a simple Node.js Express server for a one-product shop. I need: A static JSON object for the product (name, price, stock). A POST route '/order' that accepts user data and returns a success message. Use a simple array to store orders in memory."*

## Architektura

Serwer zarządza statycznymi danymi produktu, obsługuje przyjmowanie zamówień oraz serwuje skompilowane pliki frontendu z katalogu produkcyjnego (/FrontEnd/dist).



```
File Edit Selection View Go Run ... PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

File Explorer
DROPIIT
guidelines
node_modules
src
app
styles
main.tsx
ATTRIBUTIONS.md
index.html
package-lock.json
package.json
postcss.config.mjs
README.md
vite.config.ts

Terminal
powershell
node

Tips for getting started:
1. Ask questions, edit files, or run commands.
2. Be specific for the best results.
3. Create GEMINI.md files to customize your interactions with Gemini.
4. /help for more information.

> /ide install

i Installing IDE companion...
× Failed to install VS Code companion extension. Please try installing 'Gemini CLI Companion' manually from the VS Code extension marketplace.

i Code Assist login required.
Attempt to open authentication page in your browser.
Otherwise navigate to:
https://accounts.google.com/o/oauth2/v2/auth?redirect_uri=http%3A%2F%2Flocalhost%3A4382%2Foauth2callback&access_type=offline&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email%28https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.profile&state=10809f987453b23230824504bba0d066fe90a00c8305507fd1f5fc4301&response_type=code&client_id=681255809395-o8ftzoprdmp9e3aqf6av3hndib135j.https://accounts.google.com/.well-known/openid-configuration

i Waiting for authentication...
i Authentication succeeded

> Generate a simple Node.js Express server for a one-product shop. I need:
A static JSON object for the product (name, price, stock).
A POST route '/order' that accepts user data and returns a success message.
Use lowdb or just a simple array to store orders in memory.
```

## 4. Integracja i wdrożenie

Projekt został zintegrowany w architekturze rozdzielnej (Monorepo), gdzie backend komunikuje się z buildem produkcyjnym frontendu stworzonym w środowisku Vite. Serwer poprawnie obsługuje ścieżki statyczne i zapewnia płynne przejście między widokiem katalogu a procesem płatności.

## 5. Podsumowanie i wnioski

Zastosowanie AI (Figma AI oraz Gemini CLI) pozwoliło na skrócenie czasu deweloperskiego o ok. 70%. Kluczowym wyzwaniem była poprawna konfiguracja ścieżek dostępu do plików statycznych oraz dostosowanie składni Express do najnowszych standardów Node.js v22, co zostało rozwiązane poprzez iteracyjne dopracowywanie promptów technicznych.

