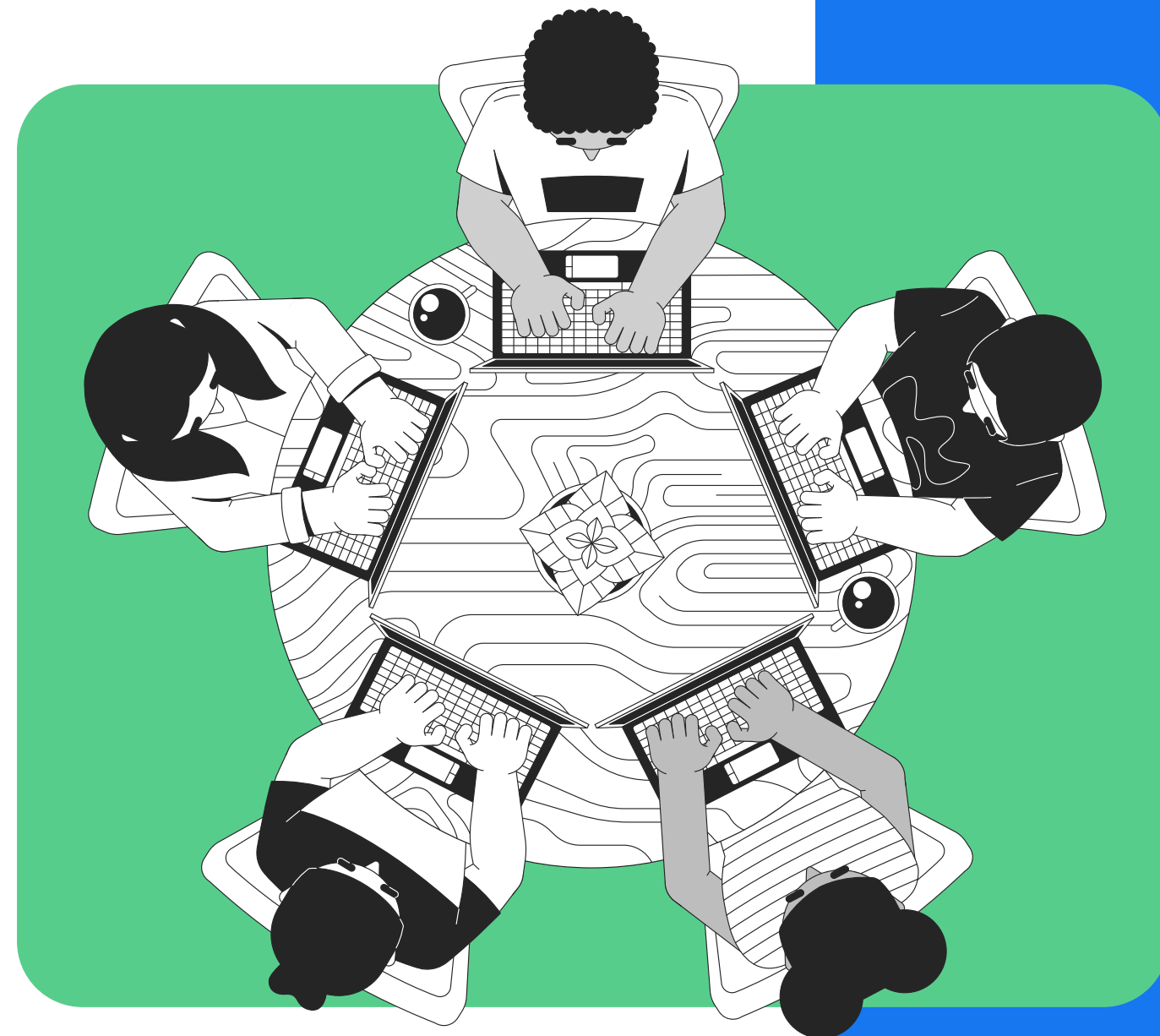


# SCHOOL < HACK DAY /> AKADEMIA

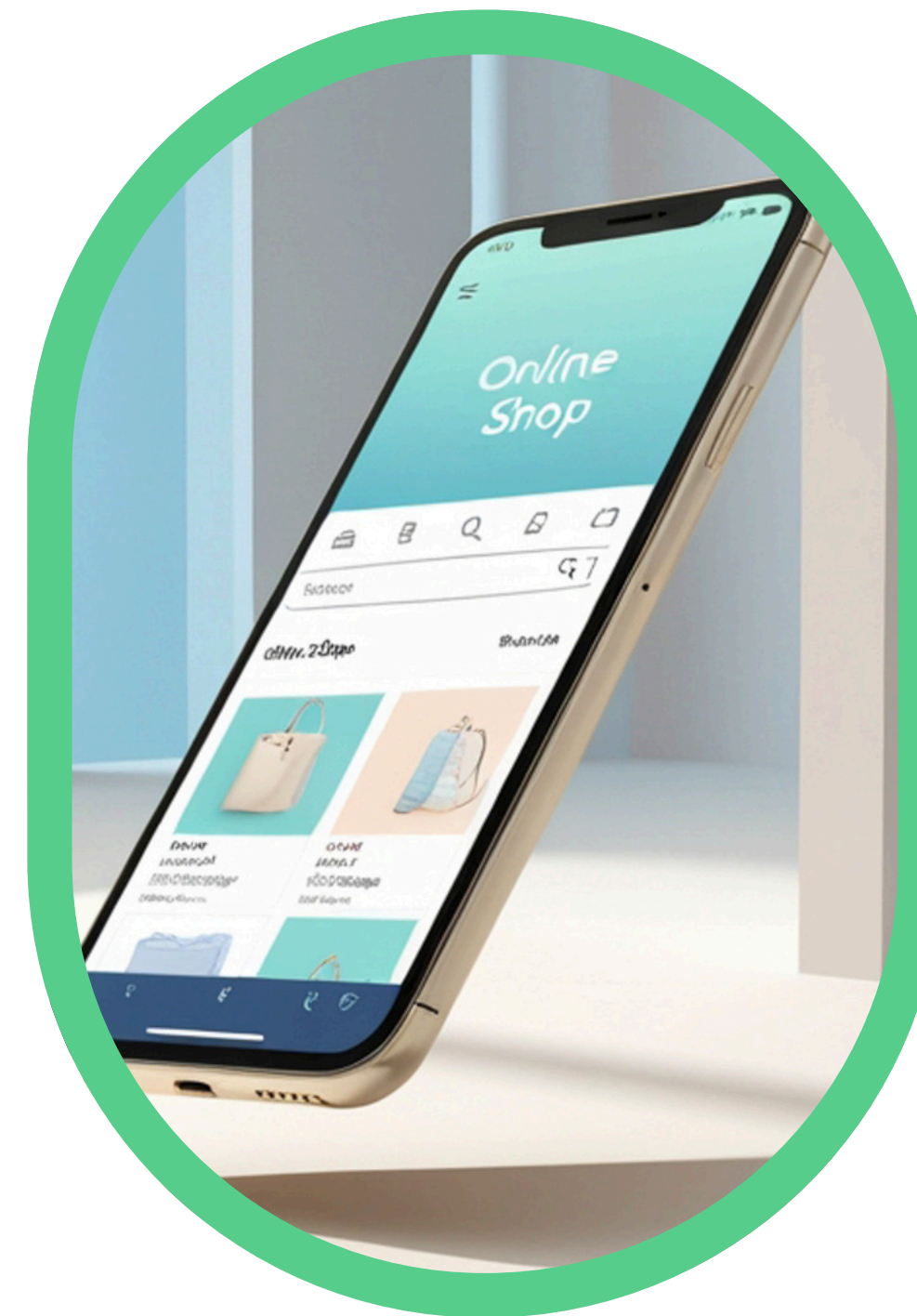
**Vibe Coding w praktyce  
Jak stworzyć klikalną  
aplikację z pomocą AI?**



# Co chcemy uzyskać?

## Cechy aplikacji na hackaton

- Mieć efekt “**wow**”
- Atrakcyjna wizualnie - czysta, schludna
- Kluczowa funkcja powinna działać - **PoC**
- Prosta na wielu poziomach:
  - stack technologiczny
  - architektura
  - jednolite środowisko pracy zespołu
  - dostęp do poprzednich wersji aplikacji
  - bez rozbudowanych wdrożeń





# Technologia apki

- Frontend: **HTML + Javascript (vanilla)**
- Backend jako usługa: <https://supabase.com>
- Repozytorium: <https://github.com>
- Środowisko: **GitHub Pages**
  
- Inne przydatne biblioteki i narzędzia:
  - <https://peerjs.com> - WebRTC
  - <https://pixijs.com> - renderer 2D
  - <https://tailwindcss.com> - komponenty UI
  - <https://openai.com/api/> - do AI :)
  - <https://socket.io> + <https://render.com> - do bardziej zaawansowanych backendów



# Narzędzia

- Laptop z MacOS / Windows / Linux
- Zainstalowany package manager:
  - MacOS → <https://brew.sh>
  - Windows → <https://chocolatey.org>
- Oprogramowanie:
  - <https://git-scm.com> (kontrola wersji)
  - <https://nodejs.org> (walidacja kodu)
  - <https://code.visualstudio.com> (IDE - opcja)
  - <https://github.com/features/copilot/cli> (asystent Vibe-codingu)
  - <https://www.python.org> (operacje na plikach)
  - dodatki: **curl wget jq**
  - dowolna przeglądarka



# Stwórzmy aplikację...

01

## Specyfikacja wymagań

Określamy wymagania funkcjonalne - "Co?", oraz niefunkcjonalne - "Jak?".

02

## Pierwsza wersja

Spisane wymagania przekazujemy Copilotowi, który przygotowuje pierwszą iterację apki

03

## Vibe-coding

Za pomocą języka naturalnego dostosowujemy rozwiązanie do naszych potrzeb, oraz aktualizujemy dokumentację

# Specyfikacja



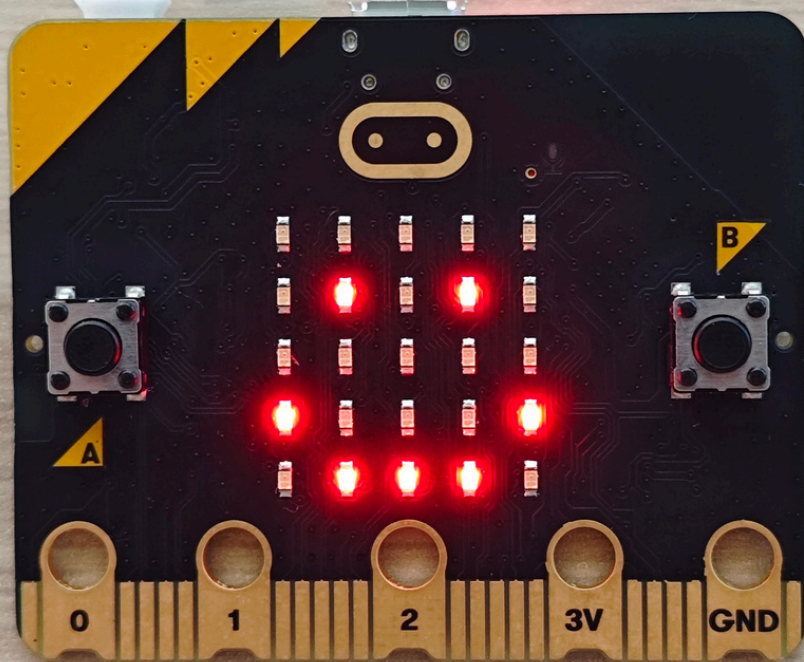
## Wymagania funkcjonalne

W pliku .md opisujemy szczegółowo wszystkie funkcje aplikacji, walidacje, interfejs. Lepszy opis = większa szansa na działającą aplikację już na starcie.

## Wymagania niefunkcjonalne

Specyfikujemy aspekty wydajności, layout katalogów, pożądane biblioteki, typy obsługiwanych urządzeń itp.

# Pierwsza wersja



## Zlecamy produkcję

W tym momencie mamy już “wsad” dla Copilota do implementacji pierwszej wersji aplikacji.

To krok, który może zająć sporo czasu, w dodatku efekty prawdopodobnie nie będą idealne. Na tym etapie powinniśmy mieć możliwość manualnego przetestowania działania aplikacji, możemy też zacommitować zmiany do repozytorium.

# Vibe-coding



## Dopracowujemy kolejne wersje

Wprowadzamy poprawki “rozmawiając” z Copilotem. Zlecamy Copilotowi, żeby dbał o aktualność naszej specyfikacji. Commitujemy wszystkie zmiany do repozytorium, żeby w razie potrzeby się cofnąć (copilot może robić to za nas).

# Wdrożenie aplikacji



## GitHub Pages

Używając tego mechanizmu, każda zmiana w repozytorium staje się widoczna publicznie (po kilku minutach). Wdrożenia powiązanych usług (np. Supabase), robimy niezależnie w asyście Copilota.

# Live demo

## Zbudujemy jedną aplikację od “0”

Przejdziemy wszystkie kroki opisane na poprzednich slajdach, by otrzymać w pełni funkcjonalną, dostępną publicznie aplikację.

## Omówimy dwie inne...

Pokażę Wam dwie aplikacje zbudowane za pomocą podobnych technik.



# Linki do repozytoriów aplikacji



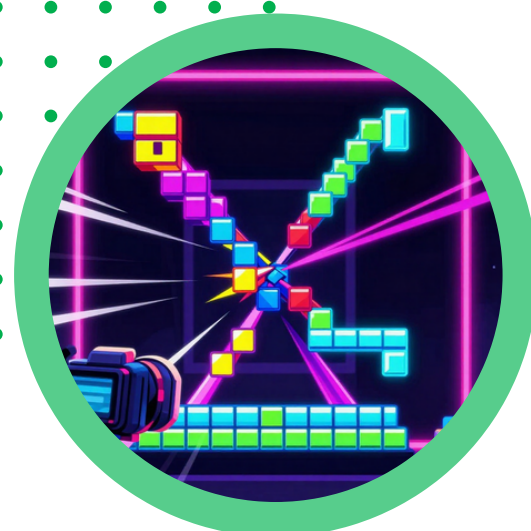
## Animal Farm

[GITHUB.COM/SCHOOLHACKDAY/ANIMAL-FARM](https://github.com/SCHOOLHACKDAY/ANIMAL-FARM)



## Virtual-Tennis

[GITHUB.COM/MILEJKO/VIRTUAL-TENNIS](https://github.com/MILEJKO/VIRTUAL-TENNIS)



## Arkanoid

[GITHUB.COM/MILEJKO/ARKANOID](https://github.com/MILEJKO/ARKANOID)



## Wasza apka!

CZEKAMY NA LINK :)

# Do zobaczenia wkrótce!

Zaczynamy kodowanie już 21 kwietnia

SchoolHackDay@sanoma.com



sanoma learning

