

FAILY TEAM

Narodziny LEGENDY



THE #1 INTERNATIONAL
BESTSELLER

Spis treści

1	Wstęp	2
2	Wodzu	3
2.1	Commit zerowy i narodziny legendy	3
2.2	Baza danych jak plastelina	3
2.3	Wielka Refaktoryzacja	4
2.4	Kryzys i papierosowa mądrość	5
3	Chimek: Mistrz odkładania i władca animców	6
4	Gabi: Ostatni bastion zdrowego rozsądku	8
5	Epilog – Kod, który przetrwa	11

1 Wstęp

Mówią, że gdy Laravel powstawał, gdzieś w czeluściach GitHuba rodziła się inna legenda. Człowiek, którego nikt nie wybrał, ale wszyscy słuchali. Człowiek, który mruknięciem potrafił naprawić błąd krytyczny. Człowiek z głosem tak stanowczym, że nawet najbardziej uparte bugi ustępowały pod jego spojrzeniem.

Mówią, że miał na imię Mati, ale wszyscy znali go pod innym mianem – Wódz.

Jak powstała ta legenda? Skąd się wziął? Dlaczego był wodzem? Nikt nie wiedział. Nikt nie pytał. Bo niektóre rzeczy po prostu są i nikt nie próbuje ich kwestionować, podobnie jak nikt nie kwestionuje, dlaczego wszystkie produkty w projekcie Faily działają – bo Wódz nad nimi czuwa.

2 Wodzu

2.1 Commit zerowy i narodziny legendy

Historia zaczęła się niepozornie. Dokładne daty są nieznane, archiwa milczą, ale starsze developerki wspominają, że kiedyś – w zamierzchłych czasach – pojawił się commit zerowy. Jeden z tych pierwszych, założycielskich commitów, które ustanawiają projekt.

```
1  commit a3f721b4d5e6c8a9e2f1d0b7c654a321e9f8d7b6
2  Author: Wódz <wodz@faily.io>
3  Date: Tue Sep 12 03:47:21 2022 +0200
4
5  Initial commit. Framework MVC, routing, ORM i auth w jednym. Bo tak.
6  - Dodano Laravel
7  - Skonfigurowano podstawową strukturę
8  - Zintegrowano Dockera
9  - Dodano PostgreSQL
10 - Skonfigurowano pgAdmin
11 - Napisano 347 testów
12 - Zoptymalizowano cache
13
14 Nikt tego nie sprawdzi, ale działa.
```

Legenda głosi, że ten commit powstał o 3:47 nad ranem, podczas gdy w tle leciał Kapitan Bomba, a Wódz popijał już piątą kawę. Commit był tak obszerny, że GitHub przez chwilę zastanawiał się, czy to nie atak DDoS.

Gdy następnego dnia zespół zobaczył to monstrum i próbował zrozumieć, jak jedna osoba mogła napisać tyle kodu w jedną noc, Wódz powiedział tylko:

“To tylko załączek. Wieczorem dopiszę resztę.”

I tak rozpoczęła się era Wodza w projekcie Faily – era, która zmieniła życie wszystkich zaangażowanych. Szczególnie Chimka, który musiał teraz nadążyć z frontendem za backendowym potworem, którego stworzył Wódz.

Gdy ktoś zapytał, kto wybrał Wodza na lidera projektu, otrzymał jedynie wzruszenie ramion i legendarną odpowiedź:

“Nikt nie wybiera Wodza. Wódz po prostu jest.”

2.2 Baza danych jak plastelina

PostgreSQL był dla Wodza jak plastelina – mógł go formować wedle uznania. Podczas gdy inni deweloperzy spędzali godziny na analizowaniu struktury bazy danych, Wódz podchodził do tego z pazurem i pewnością siebie mistrza kung-fu.

- “Potrzebujemy relacji wiele do wielu?” – zapytała Gabi.
- “Potrzebujemy tego, co ja powiem, że potrzebujemy” – odpowiedział Wódz.

Wódz nie uznawał tradycyjnych rozwiązań. Migracje, które pisał, były dziełami sztuki:

```
1 // Fragment migracji autorstwa Wódza
2 public function up() {
3   Schema::create('events', function (Blueprint $table) {
4     $table->id();
5     $table->string('name');
6     $table->text('description');
7     $table->timestamp('starts_at');
8     $table->timestamp('ends_at');
9     $table->foreignId('location_id')->constrained();
10    $table->foreignId('creator_id')->constrained('users');
11    $table->json('custom_fields')->nullable();
12    $table->timestamps();
13    $table->softDeletes();
14
15    // Tylko Wódz wie, po co te indeksy
16    $table->index(['starts_at', 'ends_at']);
17    $table->index(['location_id', 'starts_at']);
18    $table->fulltext(['name', 'description']);
19
20    // Kiedyś to się przyda, zobaczycie
21    $table->unique(['creator_id', 'name', 'starts_at']);
22  });
23
24  // Trochę triggerów dla zabawy
25  DB::unprepared('
26  CREATE TRIGGER update_search_vector
27  BEFORE INSERT OR UPDATE ON events
28  FOR EACH ROW EXECUTE FUNCTION update_event_search_vector();
29  ');
30
31  // Komentarz dla potomnych
32  DB::statement("COMMENT ON TABLE events IS 'Tu się dzieje magia. Nie
33  dotyczyć bez konsultacji z Wodzem.'");
```

Gdy ktoś pytał, dlaczego potrzebujemy tak złożonej struktury dla prostej aplikacji do znajdowania wydarzeń, Wódz odpowiadał z błyskiem w oku:

“Bo któregoś dnia będziemy mieli milion użytkowników i wtedy podziękujecie mi za te indeksy.”

Nikt nie śmiał kwestionować tej wizji. Nawet jeśli statystyki wskazywały, że aplikacja będzie miała maksymalnie kilkuset użytkowników. Wódz patrzył dalej, widział więcej, planował z wyprzedzeniem, które wydawało się absurdalne – ale nikt nie protestował.

2.3 Wielka Refaktoryzacja

Zdarzały się dni spokojne. Dni, gdy Wódz milczał, a reszta zespołu oddychała z ulgą, pracując nad swoimi zadaniami. Wydawało się, że czas płynie wolniej, że deadline jest daleko, że może tym razem...

Aż do pewnego poranka, gdy wszyscy zobaczyli powiadomienie:

```
1    commit f8e92a1b7c3d6e5f4a2b0c9d8e7f6a5b4c3d2e1
2    Author: Wódz <wodz@faily.io>
3    Date: Wed Oct 18 04:23:15 2022 +0200
4
5    Wielka refaktoryzacja, bo stary kod śmierdzał
6    - Przepisano wszystkie kontrolery
7    - Zmieniono strukturę katalogów
8    - Wprowadzono wzorzec Repository
9    - Dodano Service Layer
10   - Zaimplementowano Value Objects
11   - Dodano Command Bus
12   - Przepisano ORM
13   - Zmieniono konwencje nazewnicze
14
15   Musiałem. Kapitan Bomba kazał. NIE MERGOWAĆ bez review, ale i tak
    zmerguję sam.
```

Zespół zamarł. Przerazenie malowało się na twarzach wszystkich. Chimek zbladł, widząc powiadomienia o konfliktach w 147 plikach. Gabi spojrzała na swój kod integrujący backend z frontendem – kod, który jeszcze wczoraj działał, a dziś był tylko wspomnieniem.

Wódz wszedł do biura ze spokojem człowieka, który właśnie uratował świat przed zagładą.

“Musiałem to zrobić. Stary kod był nieelegancki”

W następnych dniach zespół pracował po godzinach, dostosowując cały frontend do nowej architektury, ucząc się nowych wzorców i konwencji. Chimek płakał nad swoimi komponentami, Gabi przepisywała adaptory API.

Ale w głębi duszy wszyscy wiedzieli, że kod faktycznie stał się lepszy. Czystszy. Bardziej elegancki. Łatwiejszy w utrzymaniu.

“Bo Wódz wiedział lepiej. Zawsze wiedział.”

2.4 Kryzys i papierosowa mądrość

Nadszedł dzień prezentacji. Miesiące pracy, tysiące linii kodu, setki commitów – wszystko miało się złożyć w działającą aplikację do znajdowania wydarzeń dla podobnych sobie osób.

Chimek siedział przy komputerze, wciąż poprawiając drobne detale CSS. Gabi testowała interakcje, upewniając się, że połączenie między frontendem a backendem działa. Wszyscy byli spięci.

I wtedy nastąpiła katastrofa.

Platforma się zawiesiła. Na ekranie wyświetlił się błąd:

```
1    Fatal error: Uncaught Exception: SQLSTATE[HY000]: General error: 1
    no such function: query_with_magic_parameters() in /var/www/html/app/
    Services/EventService.php:142
```

Cisza. Przerazenie. Chimek i Gabi spojrzeli po sobie z paniką w oczach. Wódz spojrział na ekran. Jego twarz była nieczytelna. Bez słowa wstał od biurka i wyszedł na zewnątrz.

Wszyscy wiedzieli, co to znaczy – Wódz szedł na papierosa. A gdy sprawy były naprawdę poważne, do papierosa dołączało się piwo.

Minuty mijały. Zespół nerwowo przeglądał logi, próbując znaleźć źródło błędu, ale bezskutecznie. W końcu Wódz wrócił, w dłoni ściskając niedopałek, w drugiej – otwarte, zimne piwo. Zapach tytoniu i gorycz chmielu unosiły się wokół niego jak aura nieomyślności.

“Dajcie mi klawiaturę”

Jedna linijka kodu

Jego palce zawisły nad klawiszami na ułamek sekundy. Potem, z precyzją chirurga, wpisał jedną linijkę:

```
1 // Zamień magiczną funkcję na prawdziwe zapytanie
2 $events = DB::select($query, $params);
```

Zatwierdził zmianę. Commit, push.

“Odświeżcie”

Chimek kliknął F5.

Aplikacja ożyła. Wydarzenia łądowały się błyskawicznie, interfejs migotał płynnymi animacjami, a baza danych milczała – posłuszna.

Zespół wybuchnął śmiechem, ulgą, niedowierzaniem.

– “To... to wszystko?!” – wyjąkał Chimek.

– “Czasem wystarczy jedna linijka. Ale żeby ją napisać, trzeba wiedzieć, która to linijka.”

I tak oto legenda Wodza sięgnęła zenitu. Bo prawdziwi liderzy nie potrzebują tysiąca słów – wystarczy im jedno, dobrze umieszczone, zapytanie SQL. A gdy presja rośnie, wychodzą na papierosa. Czasem na piwo.

I zawsze wracają z rozwiązaniem.

3 Chimek: Mistrz odkładania i władca animców

Mówią, że gdy Bóg rozdawał talenty do inicjalizacji repozytoriów, Chimek stał w kolejce po zdolności artystyczne. I trzeba przyznać – dostał ich w nadmiarze. Jego oko do kolorów, jego wyczucie proporcji, jego zdolność do umieszczenia każdego elementu interfejsu w idealnym miejscu – wszystko to sprawiało, że projekty Chimka wyglądały jak dzieła sztuki.

Szkoda tylko, że nigdy nie były na czas.

– “Chimek, jak tam wygląd strony głównej? Deadline był wczoraj” – zapytał kiedyś Wódz, podchodząc do biurka naszego mistrza designu.

Chimek podniósł wzrok znad monitora, na którym widniała najnowsza odcinka “Attack on Titan”.

– “To już jutro. Muszę tylko dopracować cienie pod przyciskami i spacing w nawigacji”.

Wódz westchnął, wiedząc dobrze, że “jutro” w języku Chimka oznacza “w ciągu najbliższych siedmiu dni roboczych, ale raczej bliżej siódmego”.

Historia projektu Faily była usiana takimi “jutrami” Chimka. Legenda głosi, że gdy kiedyś faktycznie dostarczył coś na “jutro”, Gabi o mało nie zemdliała z wrażenia, a Wódz wyszedł na podwójnego papierosa.

Ale największym grzechem Chimka nie było nawet opóźnianie terminów. Było nim jego upodobanie do bootowania projektu z CDN:

```
1 <!-- Fragment kodu autorstwa Chimka -->
2 <head>
3 <title>Faily - znajdź swoje wydarzenia</title>
4 <!-- 17 linijek importów z CDN zamiast npm -->
5 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/
bootstrap@5.2.0/dist/css/bootstrap.min.css">
6 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.9.1/font/bootstrap-icons.css">
7 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/animate.
css@4.1.1/animate.min.css">
8 <script src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.
min.js"></script>
9 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/
bootstrap.bundle.min.js"></script>
10 <script src="https://cdn.jsdelivr.net/npm/vue@3.2.37/dist/vue.global
.js"></script>
11 <!-- I tak dalej, i tak dalej... -->
12 </head>
```

Gdy Wódz pierwszy raz zobaczył ten kod, jego oczy rozszerzyły się w przerażeniu. Chimek, niewzruszony, sączył swoją herbatę z kubka z nadrukiem jakiegoś anime, na którym młoda dziewczyna o niebieskich włosach trzymała miecz dwa razy większy od swojego ciała.

– “Chimek, czemu nie używasz npm?” – zapytał Wódz głosem, którym normalnie pytał o to, czemu komuś umarł kot.

– “Tak jest szybciej. Nie muszę nic instalować” – odparł Chimek z niewinnym uśmiechem.

Gabi, która siedziała obok, przewróciła oczami tak mocno, że prawie zobaczyła swój mózg.

– “Ale przecież mamy package.json, webpack, całą konfigurację...” – zaczęła, ale Chimek już jej nie słuchał. Na jego ekranie rozgrywała się epicka bitwa, która najwyraźniej była ważniejsza niż jakiegokolwiek argumenty o zarządzaniu zależnościami.

Miłość Chimka do animców była legendarna w biurze. Jego monitor był stale podzielony na dwie części: po lewej kod, po prawej – najnowszy odcinek czegoś, co zawierało wielkie miecze, kolorowe włosy i krzyki o mocy przyjaźni. Nikt nie wiedział, jak udawało mu się w ogóle cokolwiek zaprojektować w takich warunkach, ale faktem było, że gdy już dostarczał skończony design – był on zachwycający.

Problem polegał na tym, że często design ten był dostarczany po terminie i zawierał tajemnicze odniesienia do seriali, których nikt inny nie oglądał:

```
1  /* CSS autorstwa Chimka */
2  .eren-mode .button-attack {
3  transform: scale(1.3);
4  box-shadow: 0 0 15px rgba(255, 0, 0, 0.7);
5  transition: all 0.5s cubic-bezier(0.68, -0.55, 0.27, 1.55);
6  }
7
8  /* Tryb Sasuke - aktywuje się w nocy */
9  @media (prefers-color-scheme: dark) {
10 .navigation {
11 background: linear-gradient(to right, #2c3e50, #000000);
12 /* Jak oczy Sasuke gdy używa Sharingana */
13 }
14 }
15
16 /* Inspirowane sceną z odcinka 342 */
17 .loader::after {
18 content: "Omae wa mou shindeiru";
19 opacity: 0;
20 /* Będzie widoczne tylko przez debuggera - niespodzianka dla nastę
21 pnego developera */
22 }
```

Kiedy któregoś dnia Gabi odkrył, że główny przycisk na stronie wydarzenia ma klasę `.hokage-button`, niemal doznał apopleksji.

– “Co to w ogóle jest hokage?” – zapytała, wpatrując się w kod z mieszaniną irytacji i zagubienia.

Chimek oświecił ją 45-minutowym wykładem o systemie politycznym w wiosce ukrytej w liściach, podczas gdy deadline projektu zbliżał się nieubłaganie. Wódz milczał, sącząc powoli kawę, a jego oczy mówiły: “Zabiję cię, ale najpierw dokończ ten pieprzony design.”

I tak mijały dni w projekcie Faily. Chimek żyjący we własnym świecie animowanych przygód, dostarczający boskie designy – zawsze za późno i zawsze wypełnione tajemniczymi referencjami do postaci, których nikt poza nim nie znał.

Ale nikt nie mógł zaprzeczyć, że gdy już skończył pracę, efekt zapierał dech w piersiach.

– “To już jutro” – mawiał Chimek z uśmiechem, a wszyscy wiedzieli, że to kłamstwo. Ale wszyscy też wiedzieli, że warto poczekać.

4 Gabi: Ostatni bastion zdrowego rozsądku

Gdyby istniała nagroda za zachowanie poczytalności w obliczu chaosu, Gabi z pewnością byłaby jej zdobywczynią. Jako frontend developer w projekcie Faily musiała znosić z jednej strony architektoniczne szaleństwa Wodza, a z drugiej – artystyczne wizje Chimka, które nadchodziły zawsze po terminie i wymagały znajomości trzech sezonów “Naruto”, by je w pełni zrozumieć.

Ale Gabi przetrwała. Więcej – zachowała zdrowie psychiczne. Jej bronią był pragmatyzm, sarkazm i stado chomiczków, których zdjęcia zdobiły jej biurko niczym talizman przeciwko szaleństwu.

Podczas gdy Wódz wychodził na papierosa po każdym większym kryzysie, a Chimek uciekał w świat animowanych bitew, Gabi po prostu wzdychała, poprawiała okulary i zabierała się do pracy.

– “Ok, więc potrzebuję komunikować się z backendem przez ten nowy pattern repository-service-manager-validator-observer-strategy-chain-of-responsibility, który Wódz zaimplementował o czwartej nad ranem?” – zapytała kiedyś, czytając dokumentację, która bardziej przypominała powieść fantasy niż specyfikację techniczną.

– “Tak, to bardzo eleganckie rozwiązanie” – odparł Wódz z dumą.

– “I mam to połączyć z frontendem Chimka, w którym główny komponent nazywa się GokuSuperSaiyanForm?”

Wódz skinął głową, nie widząc w tym nic dziwnego. Chimek, siedzący obok, uśmiechnął się szeroko, jakby nazwa ta była szczytem profesjonalizmu.

Gabi zamknęła oczy, policzyła w myślach do dziesięciu, po czym otworzyła szufladę biurka i pogłaskała małe zdjęcie chomika w hamaku.

– “Dobra. Zrobię to. Ale jeśli jeszcze raz zmienicie strukturę API bez uprzedzenia, zaczynam projekt od nowa. W jQuery”.

Ta groźba zawsze działała. Wódz bladł na samo wspomnienie jQuery, a Chimek – choć prawdopodobnie nie miałby nic przeciwko nostalgicznemu powrotowi do 2010 roku – wiedział, że oznaczałoby to przeprojektowanie całego interfejsu.

```
1 // Fragment kodu Gabi, adapter łączący dzieła Wodza i Chimka
2 export class EventServiceAdapter {
3   // Komentarz dla potomnych: Wybaczcie mi to, co tu stworzyłam
4   async fetchEvents(params) {
5     try {
6       // Tłumaczenie parametrów Chimka na język Wodza
7       const wodzParams = this.translateToWodzLanguage(params);
8
9       // Wywołanie jednego z 17 endpointów, które robią prawie to
10      samo
11       // Ale Wódz upiera się, że one się "semantycznie różnią"
12       const response = await this.getCorrectEndpointBasedOnMoon(
13         wodzParams);
14
15       // Tłumaczenie odpowiedzi Wodza na język Chimka
16       const chimekResponse = this.translateToChimekLanguage(
17         response);
18
19       // Modyfikacja odpowiedzi tak, żeby działały animacje Chimka
20       // (bo inaczej obrazi się i pójdzie oglądać Naruto)
21       return this.addRequiredGokuPowerLevel(chimekResponse);
22     } catch (error) {
23       // Log dla mnie, żebym wiedziała, co się znowu popsło
24       console.error("Wszystko się posypało, znowu:", error);
25
26       // Uspokajająca wiadomość dla użytkownika
27       return {
28         error: "Coś poszło nie tak",
29         realError: "Wódz i Chimek znowu się nie dogadali, a
30         ja muszę to naprawiać",
31         // Tego i tak nikt nie zobaczy, ale pomaga mi
32         zachować zdrowie psychiczne
33         secretMessage: "Kiedyś to wszystko rzucę i założę
34         hodowlę chomików"
35       };
36     }
37   }
38 }
```

```
32 // 150 linii kodu łątającego komunikację między backendem a
33 frontendem
34 // ...
35 }
```

Sposób, w jaki Gabi radziła sobie z chaosem, był godny podziwu. Jej kod był jak szwajcarski scyzoryk – przygotowany na każdą ewentualność, każdą zmianę API, każdy wybryk Wódza czy artystyczną wizję Chimka. Miała adaptery do adapterów, fallbacki do fallbacków i cały system early warning, który wykrywał, kiedy Wódz zaczynał przepisywać architekturę – zwykle około drugiej w nocy, po czwartej kawie.

– “Dlaczego zawsze masz działający kod, nawet gdy wszystko inne się wali?” – zapytał ją kiedyś Chimek, szczerze zdziwiony.

Gabi spojrzała na niego znad monitora, na którym widniał skomplikowany diagram przedstawiający przepływ danych między dziesiątkami komponentów.

– “Bo w przeciwieństwie do was, ja nie mam czasu na dramatyczne gesty. Nie wychodzę na papierosa, żeby rozwiązać kryzys. Nie oglądam anime zamiast dotrzymywać terminów. Po prostu piszę kod, który działa”.

Jej biurko było oazą porządku w chaosie projektu. Podczas gdy przestrzeń Wódza przypominała laboratorium szalonego naukowca, a stanowisko Chimka – sklep z gadżetami anime, biurko Gabi było minimalistyczne, funkcjonalne i – co najważniejsze – ozdobione kolekcją małych figurek i zdjęć chomików.

– “Co jest takiego wyjątkowego w chomikach?” – zapytał kiedyś Wódz, obserwując, jak Gabi ustawia nową figurkę na klawiaturze.

– “Są małe. Są puszyste. Nie wymyślają skomplikowanych architektur o trzeciej nad ranem. Nie nazywają przycisków imionami postaci z anime. Po prostu... są”.

Wódz zamyślił się, jakby ta prosta odpowiedź zawierała głęboką mądrość, której nie potrafił pojąć. Chimek, który akurat przechodził obok, zatrzymał się i z powagą stwierdził:

– “Chomiki są jak czysty, semantyczny HTML. Proste, funkcjonalne, robią swoje bez zbędnego skomplikowania”.

Gabi spojrzała na nich obu z mieszaniną zaskoczenia i podejrzliwości, zastanawiając się, czy przypadkiem nie żartują. Ale w ich oczach widziała autentyczne zrozumienie.

I może właśnie na tym polegała magia projektu Faily – każdy z nich, na swój sposób, dążył do tej samej prostoty i elegancji. Wódz poprzez skomplikowane architektury, które ostatecznie miały uczynić świat prostszym. Chimek przez artystyczne wizje, które czyniły interfejs przyjemniejszym. A Gabi – poprzez pragmatyczne podejście, które sprawiało, że wszystko po prostu działało.

Razem tworzyli nieprawdopodobną, dysfunkcyjną, ale skuteczną całość.

I być może dlatego, mimo wszystkich kryzysów, opóźnień i chaotycznych decyzji, projekt Faily działał. Działał lepiej niż powinien.

Wódz, Chimek i Gabi. Backend, design i frontend. Papierosy, anime i chomiki.

A na samym końcu – aplikacja, która jakoś, wbrew wszelkiej logice, stawała się coraz lepsza z każdym dziwnym commitem.

5 Epilog – Kod, który przetrwa

Z czasem projekt Faily zyskał popularność. I choć na pierwszy rzut oka była to po prostu platforma do znajdowania eventów, to dla zespołu stała się czymś więcej – dowodem na to, że nawet najbardziej skomplikowane systemy mogą działać, jeśli mają odpowiedniego strażnika.

Chimek nadal zmieniał CSS i wygląd aplikacji, nigdy w pełni nie zadowolony z efektu końcowego. Gabi wciąż łączyła frontend z backendem, tłumacząc jednym, czego chcą drudzy.

A Wódz? Wódz nadal pisał kod, który czasem wydawał się nadmiernie skomplikowany, ale zawsze – ale to zawsze – działał. I gdy tylko pojawiał się kryzys, wychodził na papierosa. Czasem na piwo.

Mówi się, że gdy otwierasz Faily i widzisz, jak szybko ładują się wydarzenia, gdzieś w kodzie wywołuje się funkcja napisana przez Wodza, która szepce do bazy danych:

“Działaj. Po prostu działaj.”

I może na tym właśnie polega magia backendu – tworzenie fundamentów, na których inni mogą budować, nawet jeśli nie zawsze rozumieją, jak te fundamenty działają.

Frontend i backend. Design i logika. Chimek i Gabi.

A u podstaw tego wszystkiego – Wódz, który sprawił, że słowo “Faily” nigdy nie odnosiło się do działania systemu.