

# Master Your Local Development Environment for AI adoption

YEONG SHENG



STANLY



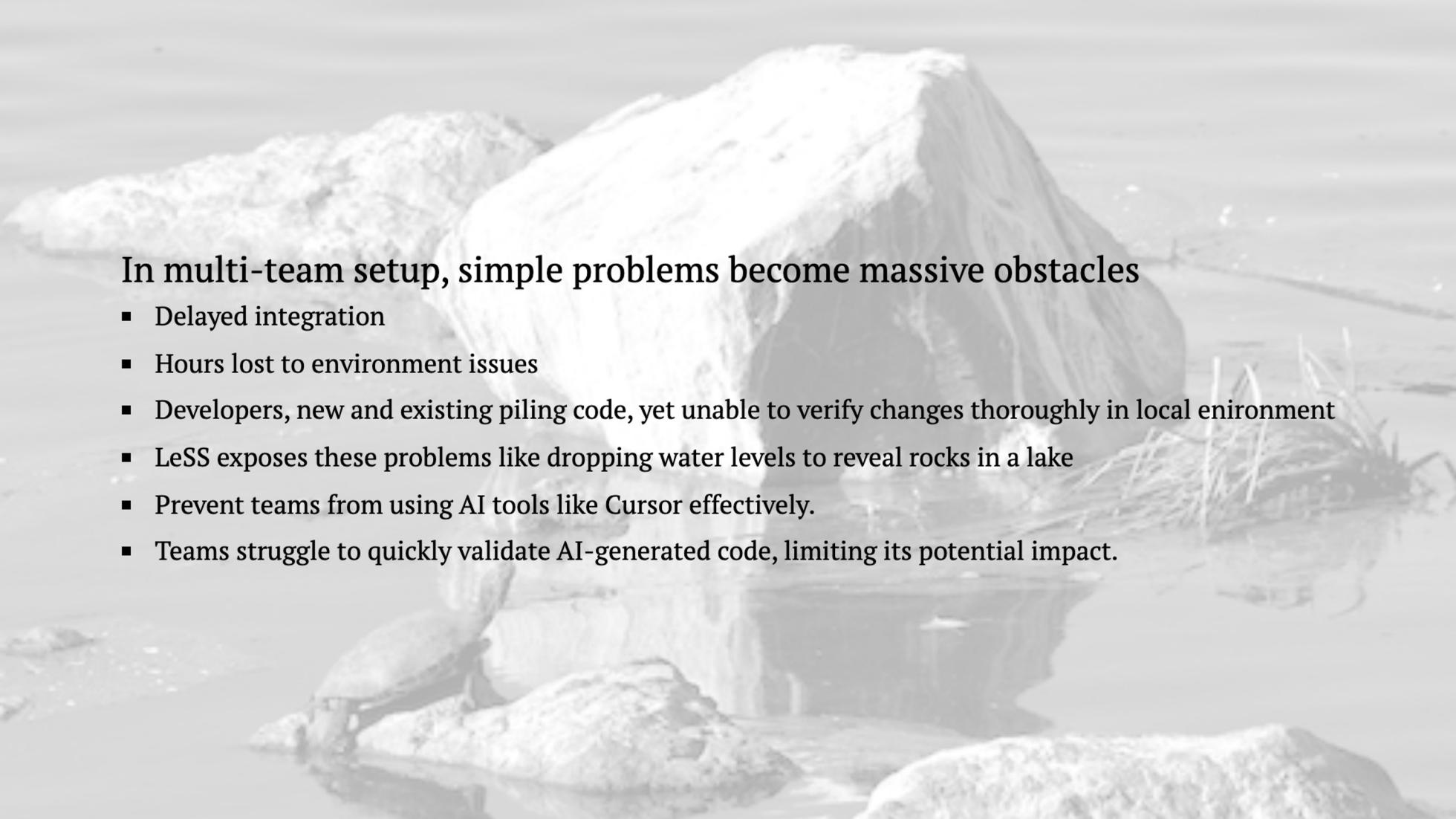
LeSS

The Lake and Rocks Metaphor (LeSS - Queueing Theory)

Press Space for next page →

**"Works on my machine"**

Isn't funny when it blocks the entire team/s

A black and white photograph of a majestic mountain range. In the foreground, a large, rocky mountain peak rises, its slopes covered in patches of snow and ice. Behind it, more snow-capped peaks are visible, creating a sense of depth and scale. The sky above is clear and bright, suggesting a sunny day.

## In multi-team setup, simple problems become massive obstacles

- Delayed integration
- Hours lost to environment issues
- Developers, new and existing piling code, yet unable to verify changes thoroughly in local environment
- LeSS exposes these problems like dropping water levels to reveal rocks in a lake
- Prevent teams from using AI tools like Cursor effectively.
- Teams struggle to quickly validate AI-generated code, limiting its potential impact.

# Demo Repository



[lunchbox\\_api](#)

## API Service used for demo: lunchbox\_api

mix phx.routes

Compiling 1 file (.ex)

Generated lunchbox\_api app

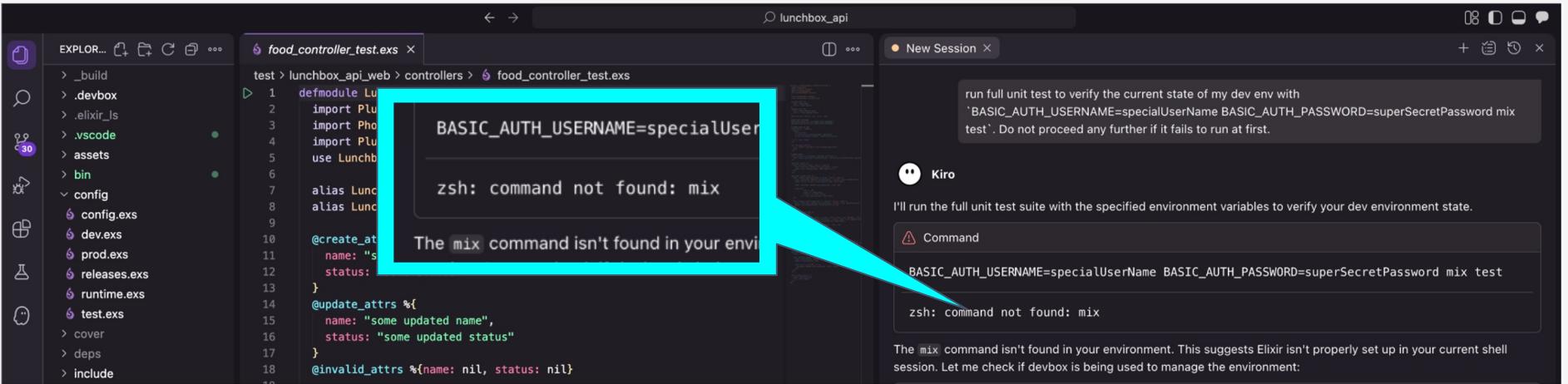
```
page_path GET   /
food_path GET   /api/v1/foods
food_path GET.  /api/v1/foods/:id
food_path POST  /api/v1/foods
food_path PATCH /api/v1/foods/:id
food_path PUT   /api/v1/foods/:id
food_path DELETE /api/v1/foods/:id
```

```
localhost:4000/api/v1/foods
Pretty print 
{
  "data": [
    {
      "id": 4,
      "name": "Apple",
      "status": "fresh"
    },
    {
      "id": 5,
      "name": "Golden Kiwi",
      "status": "sweet"
    }
  ]
}
```

```
LunchboxApiWeb.PageLive :index
LunchboxApiWeb.FoodController :index
LunchboxApiWeb.FoodController :show
LunchboxApiWeb.FoodController :create
LunchboxApiWeb.FoodController :update
LunchboxApiWeb.FoodController :update
LunchboxApiWeb.FoodController :delete
```

```
http localhost:4000/api/v1/foods --auth specialUserName:superSecretPassword
HTTP/1.1 200 OK
cache-control: max-age=0, private, must-revalidate
content-length: 980
Content-Type: application/json; charset=utf-8
date: Wed, 06 Aug 2025 05:02:34 GMT
server: Cowboy
x-request-id: GFKU8Qd0is0sEh0AAAhB

{
  "data": [
    {
      "id": 4,
      "name": "Apple",
      "status": "fresh"
    },
    {
      "id": 5,
      "name": "Golden Kiwi",
      "status": "sweet"
    }
  ]
}
```



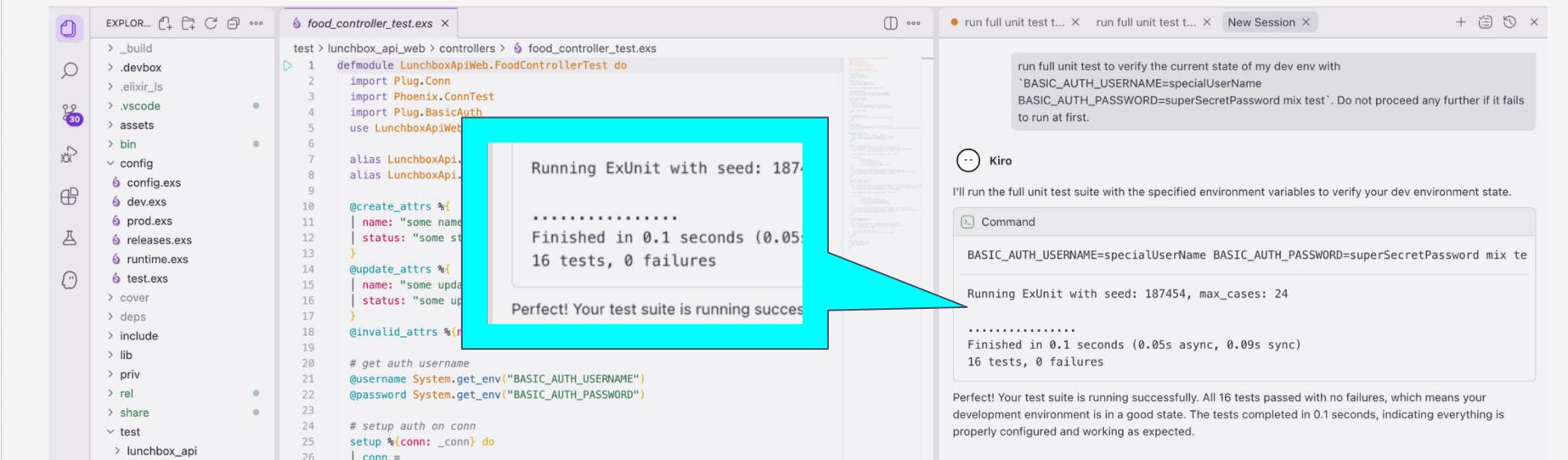
`BASIC_AUTH_USERNAME=specialUser`

`zsh: command not found: mix`

The `mix` command isn't found in your environment.

This screenshot shows a terminal window within a code editor interface. The terminal is running a command that requires the `mix` command, which is not found in the current environment. A large cyan box highlights the error message "The mix command isn't found in your environment".

```
test > lunchbox_api_web > controllers > food_controller_test.exs
1 defmodule LunchboxApiWeb.FoodControllerTest do
2   import Plug.Conn
3   import Phoenix.ConnTest
4   import Plug.BasicAuth
5   use LunchboxApiWeb
6
7   alias LunchboxApiWeb
8   alias LunchboxApiWeb.FoodControllerTest
9
10 @create_attrs %{
11   | name: "some name"
12   | status: "some status"
13 }
14 @update_attrs %{
15   | name: "some updated name",
16   | status: "some updated status"
17 }
18 @invalidAttrs %{name: nil, status: nil}
```



Running ExUnit with seed: 187454, max\_cases: 24

.....

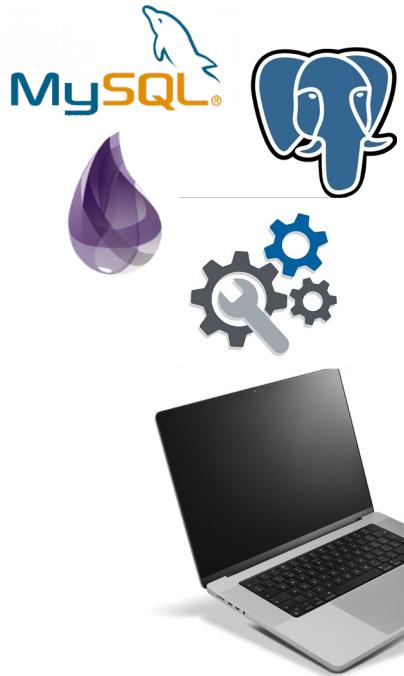
Finished in 0.1 seconds (0.05s async, 0.09s sync)

16 tests, 0 failures

Perfect! Your test suite is running successfully.

This screenshot shows a terminal window displaying the results of an ExUnit test suite. The tests were run with a seed of 187454 and a maximum of 24 cases. All 16 tests passed with no failures. A large cyan box highlights the success message "Perfect! Your test suite is running successfully".

```
test > lunchbox_api_web > controllers > food_controller_test.exs
1 defmodule LunchboxApiWeb.FoodControllerTest do
2   import Plug.Conn
3   import Phoenix.ConnTest
4   import Plug.BasicAuth
5   use LunchboxApiWeb
6
7   alias LunchboxApiWeb
8   alias LunchboxApiWeb.FoodControllerTest
9
10 @create_attrs %{
11   | name: "some name"
12   | status: "some status"
13 }
14 @update_attrs %{
15   | name: "some updated name",
16   | status: "some updated status"
17 }
18 @invalidAttrs %{name: nil, status: nil}
19
20 # get auth username
21 @username System.get_env("BASIC_AUTH_USERNAME")
22 @password System.get_env("BASIC_AUTH_PASSWORD")
23
24 # setup auth on conn
25 setup %{conn: _conn} do
26   | conn =
```

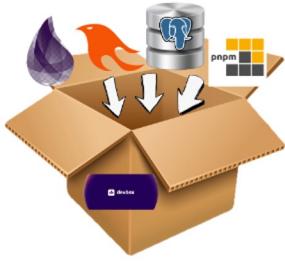


Unlock the full potential of AI integration in your development workflow.  
Achieve a tight, full-cycle feedback loop, with rapid iteration.

## Is it even possible to take complete control of your local environment?

- databases, messaging, and dependencies?
- repeatable, composable, clean-slate DB migrations?
- time-travel testing, and bending your software product system to your will for effective testing & verification?

# { DEMO }



 NixOS Nix with NixPkgs  
 Jetify Devbox

```
{
  "packages": ["postgresql@17.5", "beam27Packages.elixir@1.18.4", "nodejs@22.14.0"],
  "shell": {
    "init_hook": ["cd assets && pnpm --frozen-lockfile recursive install"],
    "scripts": {
      "test": ["BASIC_AUTH_USERNAME=specialUserName BASIC_AUTH_PASSWORD=superSecretPassword mix test"],
      "init-db": [
        "export PGDATA=$PWD/.devbox/virtenv/postgresql/data",
        "rm -rf $PWD/.devbox/virtenv/postgresql",
        "mkdir -p $PGDATA",
        "initdb --username=postgres -D $PGDATA"
      ]
    }
  }
}
```

# Key Outcomes & Takeaways

# Key Outcomes & Takeaways

The Problem

# Key Outcomes & Takeaways

## The Problem

- Integration delays ripple across teams, creating massive obstacles in multi-team setups

# Key Outcomes & Takeaways

## The Problem

- Integration delays ripple across teams, creating massive obstacles in multi-team setups
- Hours lost to "works on my machine" issues that block entire development workflows

# Key Outcomes & Takeaways

## The Problem

- Integration delays ripple across teams, creating massive obstacles in multi-team setups
- Hours lost to "works on my machine" issues that block entire development workflows
- AI tools become unreliable without proper local verification capabilities

# Key Outcomes & Takeaways

## The Problem

- Integration delays ripple across teams, creating massive obstacles in multi-team setups
- Hours lost to "works on my machine" issues that block entire development workflows
- AI tools become unreliable without proper local verification capabilities
- Blocked productivity affects both human developers and AI-assisted development

# Key Outcomes & Takeaways

## The Problem

- Integration delays ripple across teams, creating massive obstacles in multi-team setups
- Hours lost to "works on my machine" issues that block entire development workflows
- AI tools become unreliable without proper local verification capabilities
- Blocked productivity affects both human developers and AI-assisted development
- AI-generated code is only as valuable as your ability to test it locally

# Key Outcomes & Takeaways

## The Problem

- Integration delays ripple across teams, creating massive obstacles in multi-team setups
- Hours lost to "works on my machine" issues that block entire development workflows
- AI tools become unreliable without proper local verification capabilities
- Blocked productivity affects both human developers and AI-assisted development
- AI-generated code is only as valuable as your ability to test it locally
- Without a clean, reproducible environment, you can't trust AI suggestions

# Outcome

- **Recognize the true cost** of broken local environments - it's expensive and blocks both human and AI productivity
- **Master your local development environment** - service dependencies, & runtime for instant code verification
- **Unlock AI tool potential** - use tools like Cursor productively with reliable local verification to validate generated code
- **Build improvement roadmaps** - guide your teams toward better integration practices and effective AI adoption

## The Bottom Line

- AI-generated code is only as valuable as your ability to test it locally
- Without a clean, reproducible environment, you can't trust AI suggestions
- Fast feedback loops are essential for both human and AI-assisted development success

# Learn More

Lunchbox API Demo Code · Nix · Jetify Devbox

---

Powered by  Sliderv