Tech Workshop #7

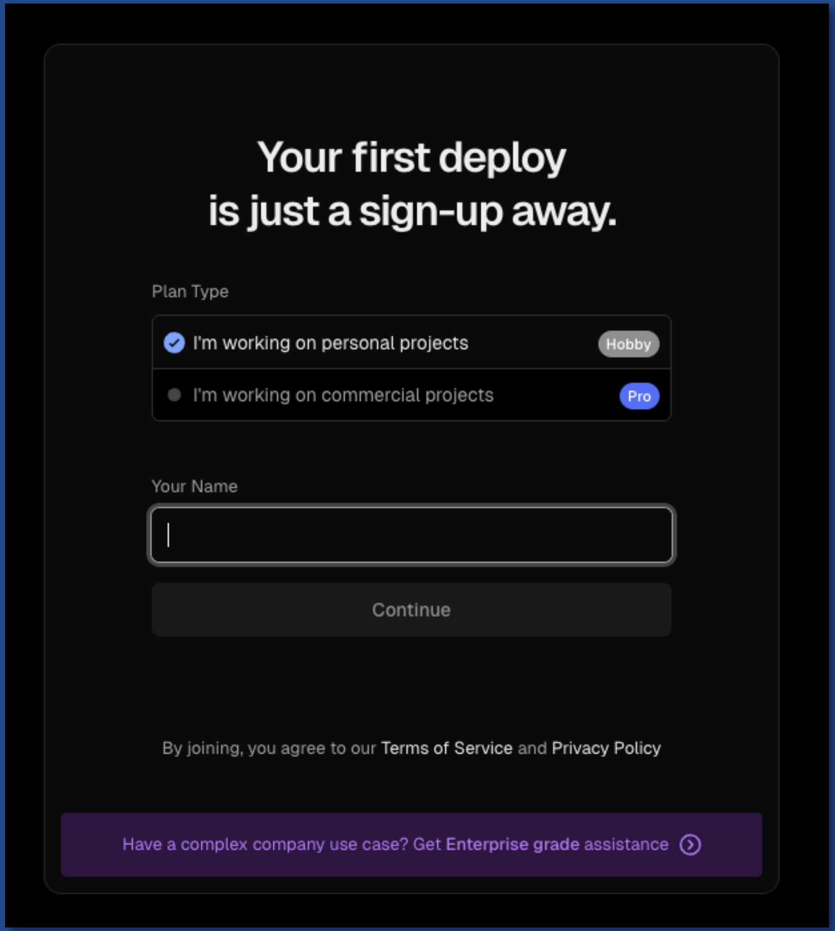
Braden Guliano

Director of Technical Development

Kappa Theta Pi

Before we start...

Prerequisites


Your first deploy is just a sign-up away.

Plan Type

[✓ I'm working on personal projects] Hobby

I'm working on commercial projects Pro

Your Name

By joining, you agree to our Terms of Service and Privacy Policy

Have a complex company use case? Get Enterprise grade assistance

Vercel account

Vercel

- What is it?
  - Cloud platform for deploying web apps
  - Vercel built Next.js, so compatibility right out of the gate
  - Handles hosting, builds, domains, etc.
- But why?
  - Instant deploys
  - Serverless functions (`/api`)
  - Edge network (fast loading)
  - Dynamic content (not static like GH pages)
  - Built-in environment var management

Neon

- What is it?
  - Serverless, Postgres database (SQL)
  - Built for modern cloud apps
  - Fully compatible with Vercel
- But why?
  - Serverless compute (pay-as-you-go)
  - Branching (like GH for databases)
  - Connection pooling

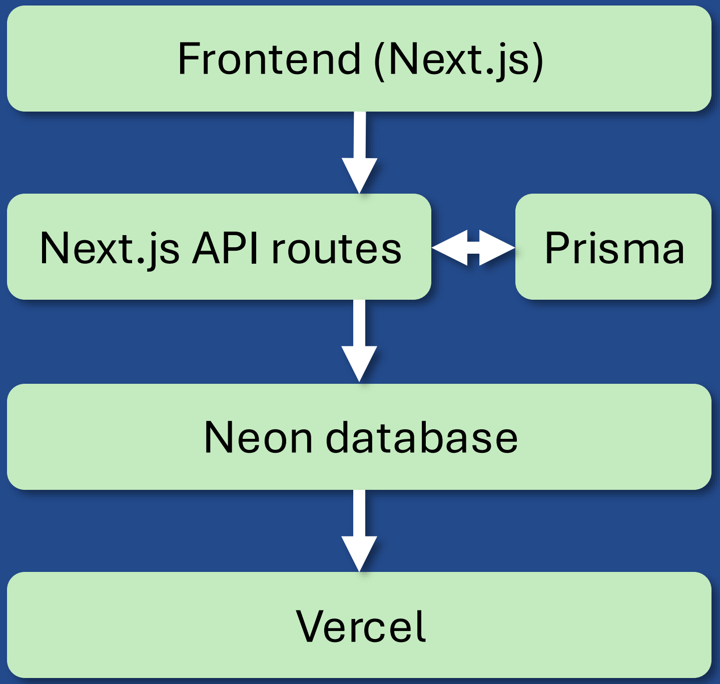
Prisma

- What is it?
  - Object-Relational Mapper (ORM) for working with databases
  - Translates TypeScript into SQL queries (no manual SQL required)
- But why?
  - Works seamlessly with Vercel and Neon
  - Autocomplete and type-safe querying
  - Migrations for version control

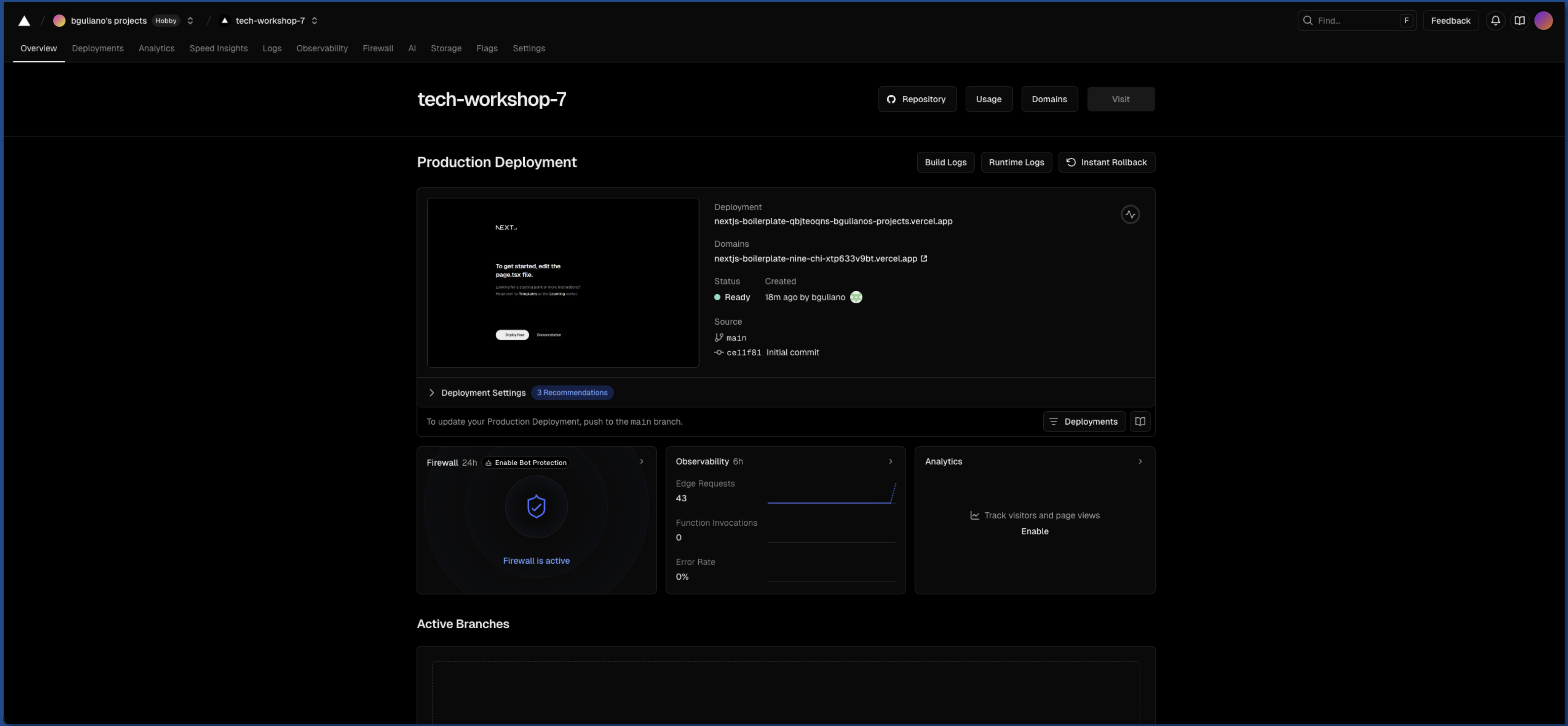
```
// Create a new todo
await prisma.todo.create({
  data: { title: 'Learn Prisma' }
})

// Get all todos
const todos = await prisma.todo.findMany()
```

How do they all work together?

1. You deploy your app to Vercel
2. You connect Neon through Vercel Integrations
3. Vercel automatically sets up a DATABASE_URL env var
4. You build custom API routes that use Prisma to talk to Neon



Vercel + Neon setup



tech-workshop-7

Production Deployment

Deployment
nextjs-boilerplate-qbjteoqns-bgulianos-projects.vercel.app

Domains
nextjs-boilerplate-nine-chi-xtp633v9bt.vercel.app

Status: ● Ready
Created: 18m ago by bguliano

Source
`main`
`ce11f81` Initial commit

Deployment Settings 3 Recommendations

To update your Production Deployment, push to the `main` branch.

Firewall 24h

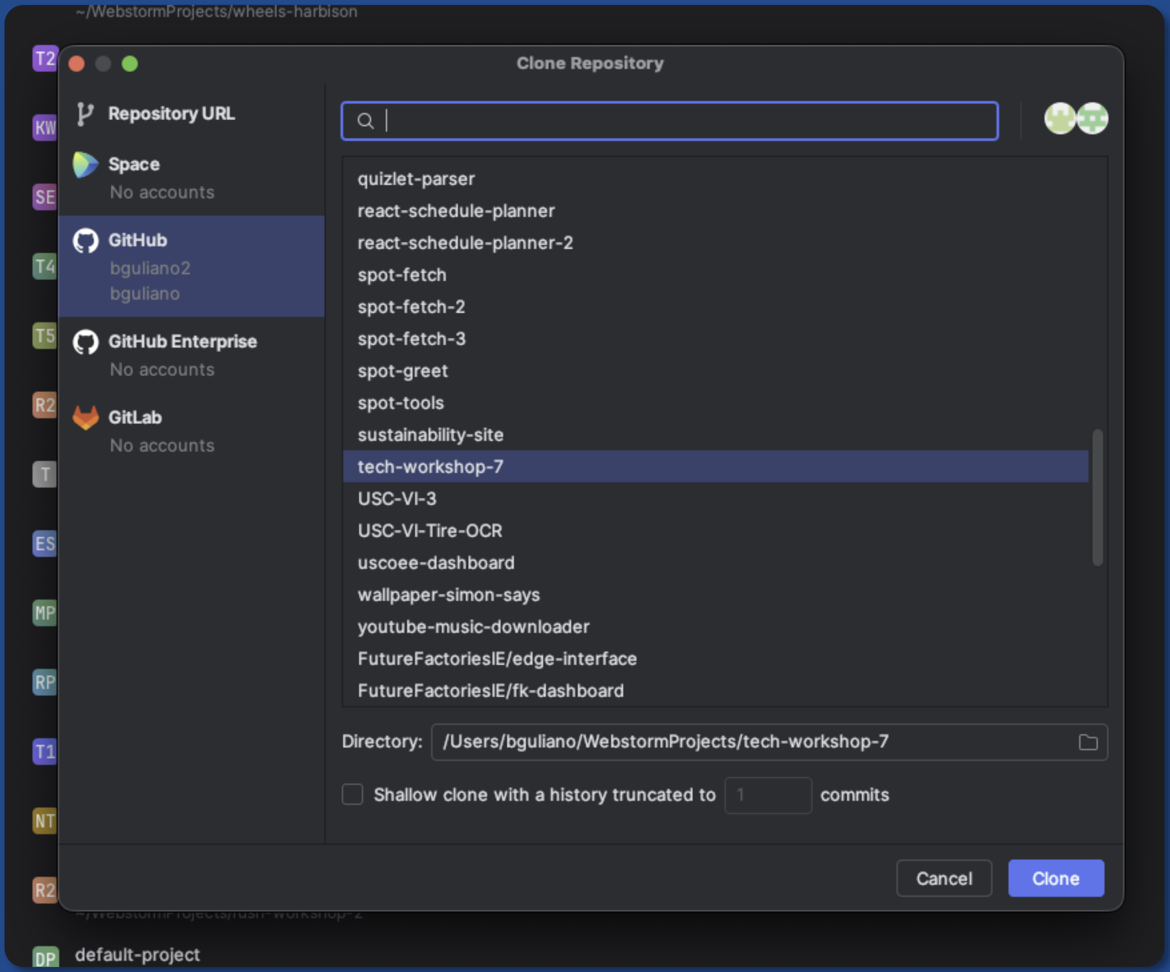
Firewall is active

Observability 6h
Edge Requests 43
Function Invocations 0
Error Rate 0%

Analytics
Track visitors and page views

Active Branches

Vercel integration



~/WebstormProjects/wheels-harbison

Clone Repository

quizlet-parser
react-schedule-planner
react-schedule-planner-2
spot-fetch
spot-fetch-2
spot-fetch-3
spot-greet
spot-tools
sustainability-site
tech-workshop-7
USC-VI-3
USC-VI-Tire-OCR
uscoee-dashboard
wallpaper-simon-says
youtube-music-downloader
FutureFactoriesIE/edge-interface
FutureFactoriesIE/fk-dashboard

Directory: /Users/bguliano/WebstormProjects/tech-workshop-7

☐ Shallow clone with a history truncated to 1 commits

default-project

Setup

```
npm install -g vercel
```

```
vercel link
```

```
vercel env pull .env.development.local
```

```
npm install @neondatabase/serverless @prisma/client prisma
```

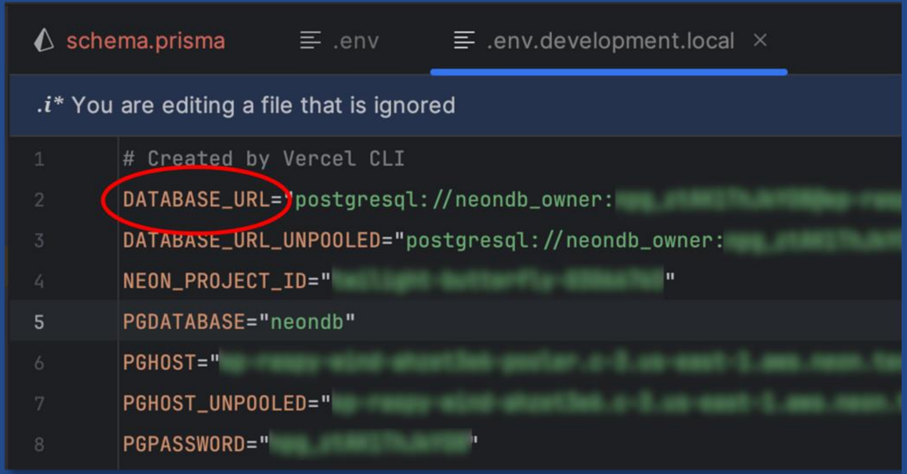
```
npx prisma init
```

Setup

```
vercel env pull .env.development.local
```

DO NOT COMMIT ENV FILES!

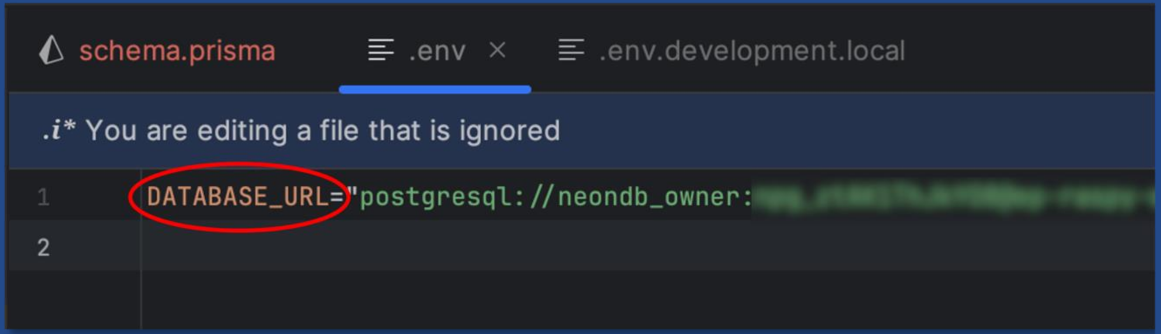
Setup



.i* You are editing a file that is ignored

```
# Created by Vercel CLI
DATABASE_URL="postgresql://neondb_owner:[redacted]
DATABASE_URL_UNPOOLED="postgresql://neondb_owner:[redacted]
NEON_PROJECT_ID="[redacted]"
PGDATABASE="neondb"
PGHOST="[redacted]
PGHOST_UNPOOLED="[redacted]
PGPASSWORD="[redacted]"
```

Copy DATABASE_URL env var from `.env.development.local` to `.env`



.i* You are editing a file that is ignored

```
DATABASE_URL="postgresql://neondb_owner:[redacted]

```

Prisma init output

```
Next steps:

1. Install `dotenv`, and add `import "dotenv/config";` to your `prisma.config.ts` file to
load environment variables from `.env`.

2. Run prisma dev to start a local Prisma Postgres server.

3. Define models in the schema.prisma file.

4. Run prisma migrate dev to migrate your local Prisma Postgres database.

5. Tip: Explore how you can extend the ORM with scalable connection pooling, global
caching, and a managed serverless Postgres database. Read: https://pris.ly/cli/beyond-orm
```

Prisma init output

```
Next steps:

~~1. Install `dotenv`, and add `import "dotenv/config";` to your `prisma.config.ts` file to
load environment variables from `.env`.~~

~~2. Run prisma dev to start a local Prisma Postgres server.~~

3. Define models in the schema.prisma file.

4. Run prisma migrate dev to migrate your local Prisma Postgres database.

~~5. Tip: Explore how you can extend the ORM with scalable connection pooling, global
caching, and a managed serverless Postgres database. Read: https://pris.ly/cli/beyond-orm~~
```

What is a migration?

A **migration** is a set of instructions that tell your database **how to change its structure** (tables, columns, relations) to match your current Prisma schema.

```
4. Run prisma migrate dev to migrate your
local Prisma Postgres database.
```

```
npx prisma migrate dev --name init
```

What is a migration?

```
npx prisma migrate dev --name init
```

```
model Todo {
  id        String   @id @default(cuid())
  title     String
  completed Boolean  @default(false)
  createdAt DateTime @default(now())
}
```

```
-- CreateTable
CREATE TABLE "Todo" (
    "id" TEXT NOT NULL,
    "title" TEXT NOT NULL,
    "completed" BOOLEAN NOT NULL DEFAULT false,
    "createdAt" TIMESTAMP(3) NOT NULL DEFAULT CURRENT_TIMESTAMP,

    CONSTRAINT "Todo_pkey" PRIMARY KEY ("id")
);
```

Prisma migrate

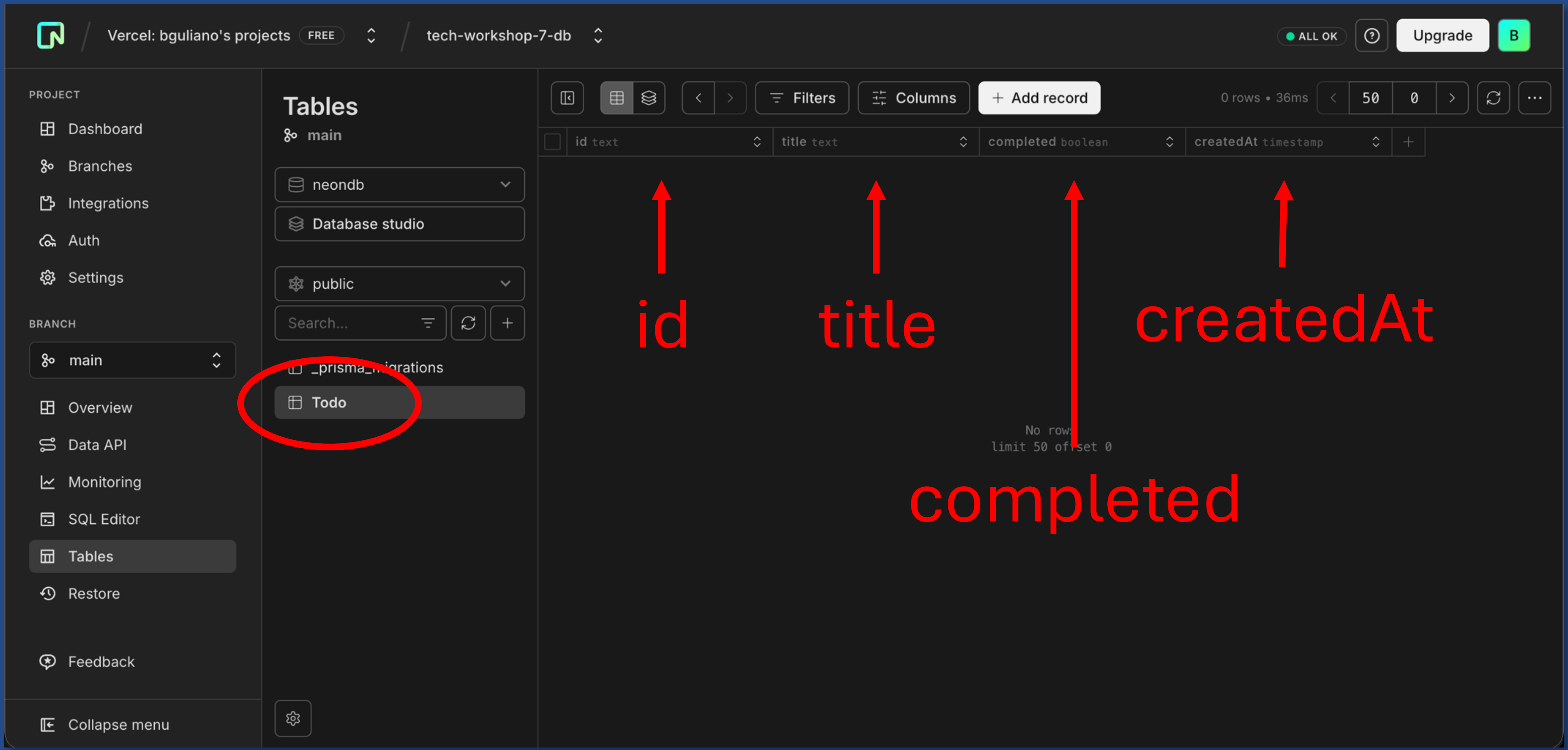
- Looks at your models in `schema.prisma`
- Creates a SQL migration to match your Neon database
- Runs that migration on your Neon instance
- Updates your local Prisma history in `prisma/migrations/`

```
4. Run prisma migrate dev to migrate your
local Prisma Postgres database.
```

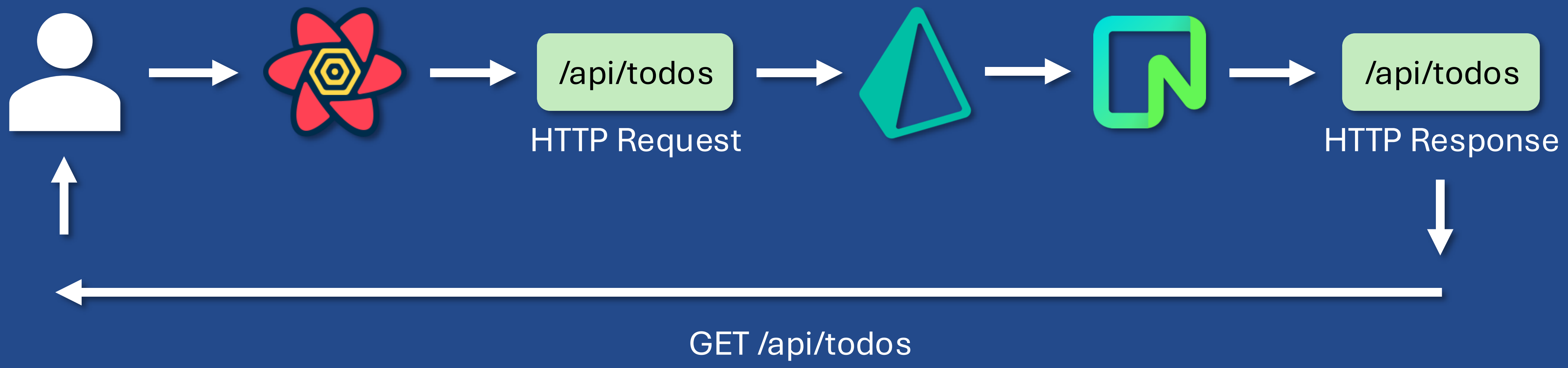
```
npx prisma migrate dev --name init
```

```
Your database is now in sync with your
schema.
```

Prisma migrate



API routing

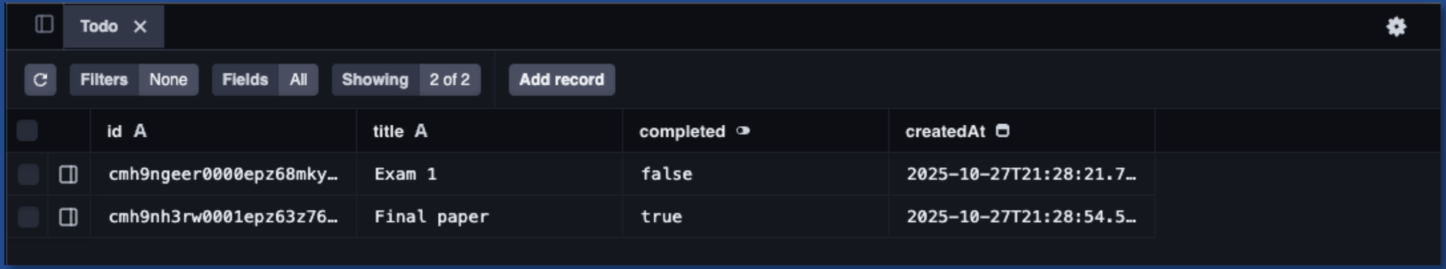

/api/todos
HTTP Request
/api/todos
HTTP Response
GET /api/todos

UI connection

```
npm install @tanstack/react-query
```

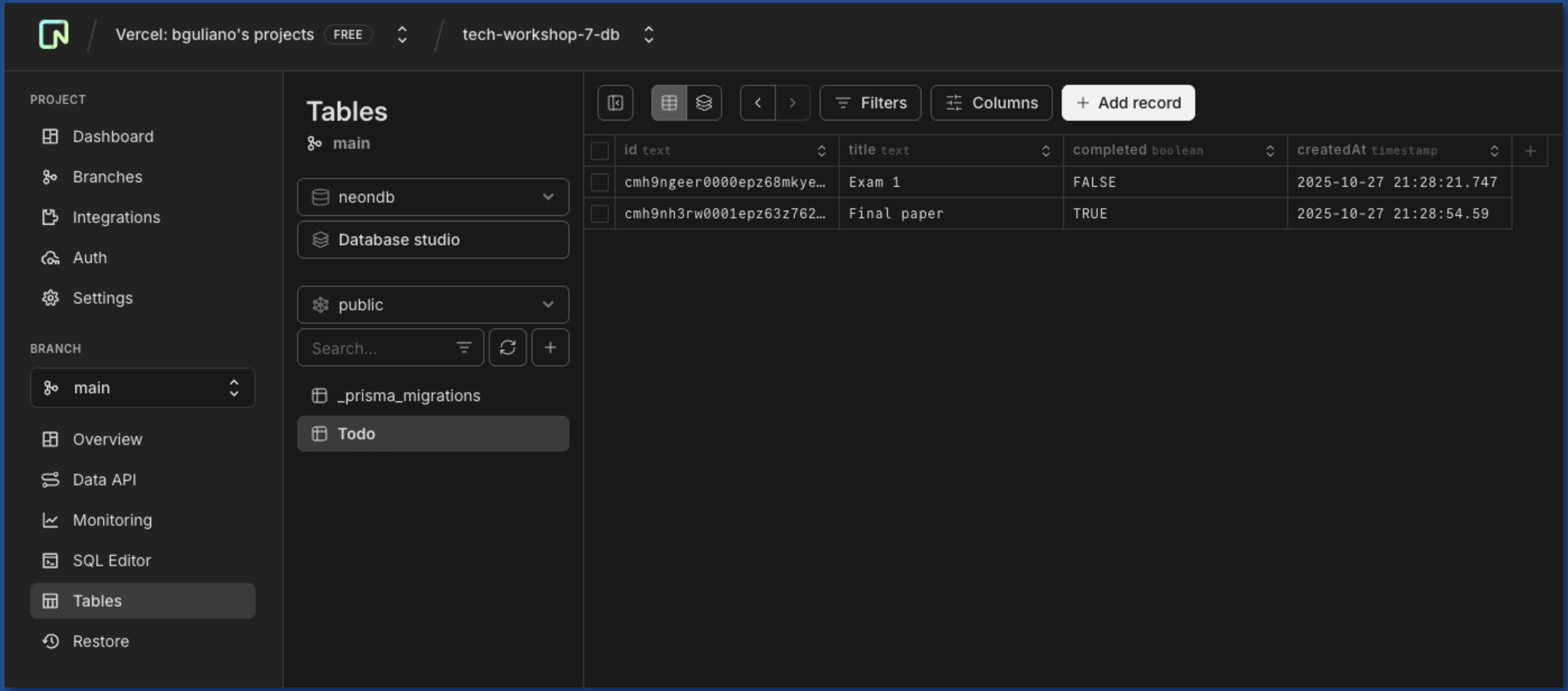
Local Prisma studio

```
npx prisma studio
```



id	title	completed	createdAt
cmh9ngeer0000epz68mky…	Exam 1	false	2025-10-27T21:28:21.7…
cmh9nh3rw0001epz63z76…	Final paper	true	2025-10-27T21:28:54.5…

Neon cloud storage



id text	title text	completed boolean	createdAt timestamp
cmh9ngeer0000epz68mkye…	Exam 1	FALSE	2025-10-27 21:28:21.747
cmh9nh3rw0001epz63z762…	Final paper	TRUE	2025-10-27 21:28:54.59

How does this work in production?

- Yes, the DATABASE_URL env var is needed to talk to the Neon database
- But…in production, Vercel automatically (and securely) injects it into the deployed backend environment
- All API functions run on Vercel's servers, not on the client's device

Questions?

ΚΘΠ