Build Your First Full Stack App With Next.js

Presented by Jeremie Bornais

Sample code available at:

github.com/jere-mie/nextjs-workshop

A Little About Me

- Full Stack Software Developer at Assent

- Co-Founder of WinHacks and BorderHacks

- Former Research & Teaching assistant at UWindsor

- Former President of the UWindsor Computer Science Society

- Former Project Lead at GDSC

- 10+ Hackathon participant, mentor, speaker, organizer

jeremie.bornais.ca

github.com/jere-mie

linkedin.com/in/jeremie-bornais

Agenda

Introduction to Next.js

What is it? Why use it? When to use it?

Creating a Next.js Project

How to create a Next.js project and run it locally

Next.js Fundamentals

Overview of the default Next.js project structure, important files and concepts.

Interactive Demo

Building a simple full-stack Next.js app from scratch

What is Next.js?

A React framework for building **full-stack web applications** with:

- 🔄 Server and Client Components (best of both worlds)

- 📁 File-based routing (App Router)

- ⚡ Built-in optimization (images, fonts, code splitting)

- 🔌 API routes for serverless backend

- 📦 Built-in support for TypeScript, CSS modules, Tailwind

Why Next.js?

- Production-ready framework (no configuration hell)

- Better performance with server-side rendering & static generation

- Simplified full-stack development in one codebase

- Large ecosystem and community support

When to use it?

- Full-stack web applications ✅

- Blogs with dynamic content ✅

- E-commerce sites ✅

- Static sites (overkill, but works) ❌

Creating a Next.js Project

Using create-next-app (recommended):

```
npx create-next-app@latest --empty --yes campus-cravings
cd campus-cravings
npm run dev
```

What you get:

TypeScript support, Tailwind CSS, ESLint configuration, Next.js App Router enabled by default, Ready-to-use project structure

Visit http://localhost:3000 to see your app! 🎉

Next.js Fundamentals

Project Structure

- `app/` - Main application code (pages, layouts, components)
- `actions/` - Server Actions (server functions callable from Client Components)
- `public/` - Static assets (images, fonts, etc.)
- `components/` - Reusable React components
- `.env` - Environment variables (e.g. database connection strings)
- `globals.css` - Global styles (imported in `app/layout.tsx`)
- lots of other config files (tsconfig.json, next.config.ts, etc.). Don't worry about these for now, you'll know when you need to use them!

File-based Routing (App Router)

- Files in `app/` directory automatically become routes

- `app/page.tsx` → `/` (home page)

- `app/about/page.tsx` → `/about`

- `app/layout.tsx` → Wraps all pages (metadata, navbar, etc.). Put things like navbars, footers, and global styles here.

- `app/api/ratings/route.ts` → API routes (backend endpoints for `/api/ratings`)

Server vs Client Components

- **Server Components** (default): Run on server, great for databases, secrets. No access to browser APIs or interactivity.

- **Client Components**: Requires `'use client'` directive, run in browser, handle interactivity. No access to server-only features like databases or secrets.

- **API Routes**: `app/api/` folder for backend endpoints

- **Server Actions**: Server functions that can be called from Client Components (e.g. form submissions)

Tech Stack (for this workshop)

- **Framework**: Next.js with TypeScript

- **Styling**: Tailwind CSS (utility-first CSS)

- **Database**: SQLite with Drizzle ORM

- **Forms & Actions**: Server Actions for form submissions

- **Deployment**: Ready for Vercel, AWS, etc. (For db, you can use something like Turso for serverless SQLite)

Deployment Options

When it comes to deploying your Next.js app, you need to think about two different things: **where to host the app itself** and **where to host the database**. Here are some good options for both:
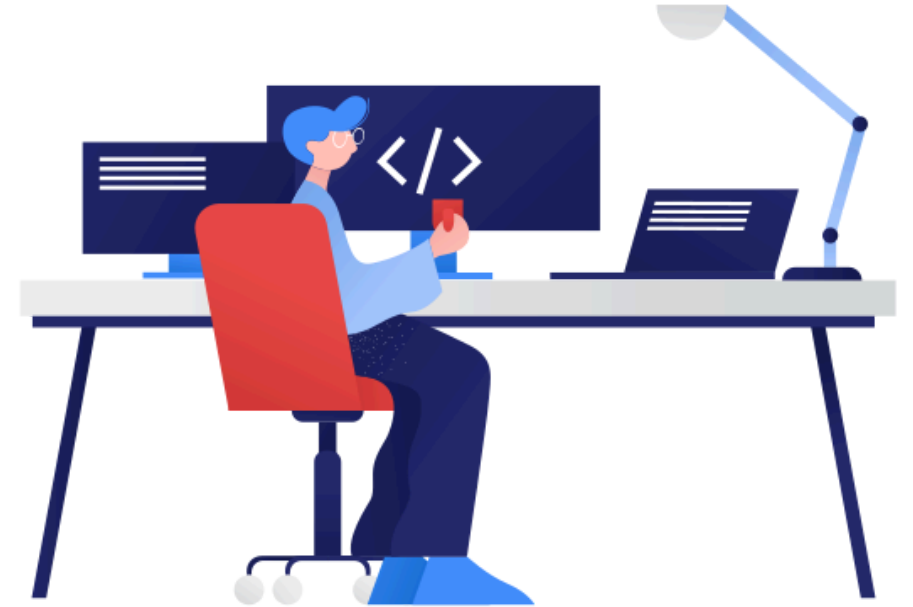
For Your App:

Vercel: Seamless Next.js deployment, serverless functions for API routes, easy environment variable management, generous free tier available for non-commercial projects. https://vercel.com/

For Your Database:

Turso: Serverless SQLite database, easy integration with Next.js, very generous free tier available. https://turso.tech/

Interactive Demo

Let's make something!

Questions?

Ask away!

THANK YOU FOR JOINING!

I hope you learned something new 😀

Remember, the source code and these slides can be found here:

github.com/jere-mie/nextjs-workshop