Helixaa Academy

Next.js Complete Course

Zero to Full-Stack Next.js Developer (10-Week Comprehensive Program)

www.academy.helixaa.com

Overview

This course takes a student from zero web development knowledge to building and deploying a full production-grade Next.js application. Since students don't know React yet, the first two weeks build the necessary JavaScript and React foundation before moving into Next.js itself.
Final Outcome: Every student builds and deploys a full-stack Next.js app (e.g., a blog/e-commerce/dashboard platform) with authentication, database, and live deployment.

Learning Objectives

- **Explain core web concepts** : how the client-server model, HTTP, and rendering strategies (SSR, SSG, ISR, CSR) work together in a modern web application.
- **Write modern JavaScript (ES6+) confidently** : including arrow functions, destructuring, array methods, and asynchronous code with async/await.
- **Build UI with React** : using components, props, state, hooks (useState, useEffect, useRef, useContext), and conditional/list rendering.
- **Structure applications using the Next.js App Router** : including nested layouts, dynamic routes, loading/error states, and navigation.
- Differentiate and correctly use Server Components vs Client Components, understanding what runs on the server vs the browser and why it matters for performance.
- **Fetch and cache data effectively** : using fetch() with Next.js caching/revalidation strategies, and choosing between Server Components, Server Actions, and API routes.
- **Style production-ready UIs** : using Tailwind CSS, CSS Modules, and component libraries to build responsive, accessible interfaces.
- **Handle forms and mutations safely** : using Server Actions, input validation (Zod), and proper error/loading states.
- **Design and connect a real database** : using Prisma ORM with PostgreSQL, including schema design and relational data (e.g., posts and comments).
- **Implement authentication and authorization** : including login/signup, OAuth, protected routes, and role-based access control.
- **Optimize applications for performance and SEO** : using next/image, next/font, metadata APIs, sitemaps, and caching/revalidation techniques.
- **Deploy a full-stack Next.js application to production** : using Vercel, environment variables, and a live production database.
- **Apply Git/GitHub workflows** : committing, pushing, and managing code throughout a real project lifecycle.
- **Independently plan, build, and present a capstone project** : demonstrating the ability to take a full-stack Next.js application from idea to live deployment.

Program Information

Estimated Time

10 Weeks · 100 Live Hours
20 Intensive Sessions 7:00 PM – 12:00 AM per session

Skill Level

Beginner

Medium

Blend of English and Sinhala

Session Delivery

Live Delivered
*Recordings available

Required Software

- Node.js & npm
- Visual Studio Code

VS Code Extensions

1. *ES7+ React/Redux/React-Native snippets*
2. *Tailwind CSS IntelliSense*
3. *Prisma*
4. *GitLens*

- Git & GitHub Account
- PostgreSQL
- Prisma CLI
- Vercel Account (free tire)

System Requirements:

- OS: Windows / macOS / Linux
- RAM: 8GB minimum (16GB better, especially DB + dev server run)
- Internet connection (stable) — npm packages, deployment, cloud DB access

The length of this program is an estimation of total hours. Live sessions are 5 hours each, held on Saturdays and Sundays. Prerequisites including HTML, CSS, and Responsive Design fundamentals are delivered via pre-recorded LMS modules before live sessions begin.

Module 1 LIVE

Web Fundamentals Refresher (Week 1)

Goal: Make sure every student is on the same page before touching React/Next.js

Lesson 1.1 — How the Web Works

- Client vs server, HTTP requests/responses, browsers, DNS
- What is a "framework" vs a "library"
- Where Next.js fits in the modern web ecosystem

Lesson 1.2 — HTML & CSS Essentials

- Semantic HTML, forms, tags recap
- CSS box model, Flexbox, Grid (quick practical recap)
- Responsive design basics

Lesson 1.3 — Modern JavaScript (ES6+) Crash Course

- let/const, arrow functions, template literals
- Array methods: map, filter, reduce
- Destructuring, spread/rest operators
- Promises, async/await
- ES Modules (import/export)

Lesson 1.4 — Node.js & Package Managers

- What is Node.js, installing Node
- npm/yarn/pnpm basics, package.json
- Running scripts, installing dependencies

Assignment 1: *Build a small static HTML/CSS/JS page (e.g., a personal profile card) using ES6+ syntax.*

Module 2 LIVE

React Fundamentals (Week 2)

Goal: Understand React deeply before Next.js abstracts things away

Lesson 2.1 — Introduction to React

- Why React exists, Virtual DOM concept
- JSX syntax rules
- Creating a React app (Vite) for practice

Lesson 2.2 — Components & Props

- Functional components
- Passing and typing props
- Component composition

Lesson 2.3 — State & Events

- useState hook
- Handling forms and events
- Conditional rendering, list rendering with key

Lesson 2.4 — Hooks Deep Dive

- useEffect (side effects, cleanup)
- useRef, useContext
- Custom hooks (intro)

Assignment 2: *Build a To-Do List app in plain React (CRUD in local state).*

Module 3 LIVE

Introduction to Next.js (Week 3)

Goal: Understand what Next.js adds on top of React

Lesson 3.1 — Why Next.js?

- SPA vs SSR vs SSG vs ISR — the rendering strategies explained
- Next.js vs Create React App vs Vite
- Setting up a new Next.js project (create-next-app)
- Project structure walkthrough

Lesson 3.2 — App Router Basics

- File-based routing (app/ directory)
- Pages, layouts, nested routes
- page.tsx, layout.tsx, loading.tsx, error.tsx, not-found.tsx

Lesson 3.3 — Navigation

- <Link> component vs <a> tag
- useRouter, usePathname, useSearchParams
- Programmatic navigation

Assignment 3: *Convert the To-Do app from Module 2 into a multi-page Next.js app with navigation.*

Module 4 LIVE

Server & Client Components (Week 4)

Goal: Master the biggest mental shift Next.js introduces

Lesson 4.1 — Server Components vs Client Components

- What runs where, and why it matters
- The "use client" directive
- Rules and limitations of each

Lesson 4.2 — Data Fetching in Server Components

- fetch() with caching strategies (force-cache, no-store, revalidate)
- Fetching directly inside components (async components)
- Parallel vs sequential data fetching

Lesson 4.3 — Dynamic Routes & Params

- [id], [...slug], [[...slug]] patterns
- generateStaticParams for SSG
- generateMetadata for SEO per page

Assignment 4: *Build a blog listing + dynamic blog detail page pulling data from a public API.*

Module 5 LIVE

Styling in Next.js (Week 5)

Goal: Master the biggest mental shift Next.js introduces

Lesson 5.1 — Styling Options Overview

- CSS Modules
- Global styles
- Styled-jsx (built-in)

Lesson 5.2 — Tailwind CSS with Next.js

- Setup and configuration
- Utility-first workflow, responsive design
- Building reusable UI patterns

Lesson 5.3 — Component Libraries

- Introduction to shadcn/ui (or similar)
- When to build custom vs use a library

Assignment 5: *Redesign the blog project fully with Tailwind CSS + a component library.*

Module 6 LIVE

Forms, Server Actions & Mutations (Week 6)

Lesson 6.1 — Server Actions

- What are Server Actions and why they matter
- Creating and calling a Server Action from a form
- Progressive enhancement benefits

Lesson 6.2 — Forms in Next.js

- Controlled forms with client components
- Validation (manual + with Zod)
- Handling errors and loading states (useFormStatus, useActionState)

Lesson 6.3 — API Routes (Route Handlers)

- Creating REST-style endpoints in app/api/
- GET, POST, PUT, DELETE handlers
- When to use API routes vs Server Actions

Assignment 6: *Add a "Create Post" and "Delete Post" feature using Server Actions.*

Module 7 LIVE

Database Integration (Week 7)

Lesson 7.1 — Connecting a Database

- Choosing a database (PostgreSQL/MongoDB) for the course project
- Setting up Prisma ORM (or Drizzle)
- Schema design basics

Lesson 7.2 — CRUD with a Real Database

- Replacing mock/API data with real DB queries
- Server Components querying the DB directly
- Handling relations (e.g., posts → comments)

Lesson 7.3 — Data Validation & Error Handling

- Zod schemas for input validation
- Try/catch patterns, user-friendly error messages

Assignment 7: *Migrate the blog project to use a real database with Prisma.*

Module 8 LIVE

Authentication & Authorization (Week 8)

Lesson 8.1 — Auth Concepts

- Sessions vs JWT, cookies basics
- Introduction to NextAuth.js / Auth.js

Lesson 8.2 — Implementing Authentication

- Email/password + OAuth (Google/GitHub) login
- Protecting routes (middleware)
- Role-based access (admin vs normal user)

Lesson 8.3 — User-specific Data

- Showing only the logged-in user's data
- Profile pages, session management

Assignment 8: *Add login/signup and make posts belong to specific logged-in users.*

Module 9 LIVE

Optimization, SEO & Deployment (Week 9)

Lesson 9.1 — Performance Optimization

- next/image, next/font optimization
- Code splitting, lazy loading (dynamic())
- Caching strategies revisited, revalidatePath/revalidateTag

Lesson 9.2 — SEO Fundamentals

- Metadata API, Open Graph tags
- sitemap.xml, robots.txt generation
- Structured data basics

Lesson 9.3 — Deployment

- Deploying to Vercel (step-by-step)
- Environment variables, production DB setup
- Custom domains, monitoring basics

Assignment 9: *Deploy the full project live on Vercel with a working production database.*

Module 10 LIVE

Capstone Project (Week 10)

Lesson 10.1 — Project Planning

- Choosing a capstone idea (e-commerce store / job board / social app / SaaS dashboard)
- Planning features, database schema, page structure

Lesson 10.2–10.4 — Guided Build Sessions

- Instructor-guided build time with checkpoints
- Code reviews and feedback sessions

Lesson 10.5 — Final Presentation

- Each student presents their deployed capstone project
- Peer feedback + certificate of completion

Final Deliverable: *A fully deployed, full-stack Next.js application with auth, database, and clean UI — ready to be added to a portfolio.*