EDVANTA

Ignite your growth, inspire your mind

12-WEEK PROGRAM

Full Stack Web Development

A Practical, Industry-Aligned 3-Month Course for Early Professionals & Career Switchers

This program is designed to take you from foundational concepts to job-ready skills in full stack web development. Over 12 structured weeks, you'll combine recorded weekend sessions with live 1:1 mentoring to build real projects, master industry tools, and align your skills with what employers actually need.

🎯 Who It's For

Early professionals (0–4 yrs), graduate students, and career switchers entering the web dev industry

🗓️ Duration

3 Months · 12 Weeks · Weekend recorded sessions + live mentoring

⚙️ Approach

70–80% core concepts, 10–15% industry use cases, 100% practical & project-based

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

WHY THIS COURSE

Why Full Stack Development Matters in Today's Industry

Full stack developers are among the most in-demand roles in tech. Companies — from early-stage startups to Fortune 500 enterprises — prefer engineers who can own both the frontend and backend of a product. This reduces handoff friction, speeds up delivery, and makes small teams dramatically more capable.

The modern web has become the universal platform for business. SaaS products, e-commerce, fintech tools, healthcare portals, and internal enterprise dashboards are all built on web technologies. Knowing the full stack means you can ship end-to-end features independently — a skill that commands premium salaries and strong career leverage.

- Average U.S. salary for full stack developers: **$110K–$140K**
- One of the top 5 most-posted tech job titles globally
- High overlap with adjacent roles: DevOps, mobile, cloud
- Startup founders frequently begin as full stack developers

Industry Signal

According to the U.S. Bureau of Labor Statistics, web developer employment is projected to grow **16% through 2032** — much faster than average. The shift to cloud-native, API-first architectures has made full stack fluency a baseline expectation at most product companies.

What Employers Want

- React or Vue on the frontend
- Node.js or Python on the backend
- REST / GraphQL API design
- SQL + NoSQL database experience
- Git, CI/CD, and cloud deployment

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

COURSE ARCHITECTURE

3-Phase Learning Journey

The curriculum is structured in three progressive phases, each building on the last. Every phase combines concept mastery with applied practice — so you're never just reading about development, you're doing it.

1

Phase 1 · Weeks 1–4

Foundation

HTML, CSS, JavaScript, Git, command line, responsive design, and web fundamentals

2

Phase 2 · Weeks 5–8

Intermediate

React, Node.js, Express, REST APIs, databases (SQL + NoSQL), and authentication

3

Phase 3 · Weeks 9–12

Advanced & Application

System design, cloud deployment, CI/CD, performance, security, and capstone project

Each week includes: a **recorded concept session** (weekend), a **live mentoring session** (doubt clearing, code review, feedback), 5–6 core topics, a hands-on exercise, and a weekly mini-assignment.

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

TOOLS & STACK

Technology Stack Overview

This course is built around the most widely used and employer-requested tools in the industry. Every technology below is chosen because it appears consistently in job descriptions and real production systems — not just tutorials.

Frontend

- HTML5, CSS3, JavaScript (ES6+)
- React.js (hooks, context, router)
- Tailwind CSS / Bootstrap
- Figma for UI reference

Backend

- Node.js + Express.js
- REST API design patterns
- JWT & OAuth authentication
- Python/Flask (intro)

Databases

- PostgreSQL (relational/SQL)
- MongoDB (NoSQL/document)
- Prisma ORM
- Redis (caching basics)

DevOps & Deployment

- Git + GitHub (version control)
- Docker (containers)
- AWS / Vercel / Render
- GitHub Actions (CI/CD)

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

PHASE 1 · FOUNDATION

Week 1 — Web Fundamentals & Developer Setup

Learning Outcome: Set up a professional dev environment, understand how the web works, and write your first structured, styled webpage.

1

How the Web Works

DNS, HTTP/HTTPS, client-server model, request-response cycle, status codes

2

Dev Environment Setup

VS Code, terminal basics, Node.js install, browser DevTools, extensions

3

HTML5 Structure & Semantics

Tags, attributes, semantic HTML (header, main, article), forms, accessibility basics

4

CSS Foundations

Selectors, box model, cascade, specificity, units (px, rem, %), colors

5

CSS Layout Basics

Flexbox, positioning, display properties, simple responsive behavior

6

Git & Version Control

git init, add, commit, push, pull, branching basics, GitHub account setup

Mini Project: Build and deploy a personal profile page with sections for bio, skills, and contact info. Push to a public GitHub repo.

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

PHASE 1 · FOUNDATION

Week 2 — JavaScript Core Concepts

Learning Outcome: Write functional JavaScript for the browser — understand syntax, control flow, DOM manipulation, and event handling.

1 JS Syntax & Data Types

Variables (let/const/var), strings, numbers, booleans, arrays, objects, typeof

2 Functions & Scope

Function declarations, arrow functions, closures, hoisting, lexical scope

3 Control Flow & Loops

if/else, switch, ternary, for/while loops, break/continue, early returns

1 DOM Manipulation

querySelector, createElement, appendChild, innerHTML vs. textContent, classList

2 Events & Listeners

addEventListener, event object, bubbling vs. capturing, form events, keyboard events

3 ES6+ Modern Syntax

Destructuring, spread/rest, template literals, optional chaining, nullish coalescing

Mini Project: Interactive to-do list app — add, check off, and delete tasks. All DOM-based, no frameworks yet.

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

PHASE 1 · FOUNDATION

Week 3 — Async JavaScript & APIs

Learning Outcome: Understand asynchronous programming in JavaScript and confidently fetch and display data from real external APIs.

01

Callbacks & the Event Loop

How JS handles async, call stack, task queue, microtasks, setTimeout, setInterval

02

Promises

.then()/.catch()/.finally(), Promise.all(), Promise.race(), chaining, error propagation

03

Async / Await

async functions, await keyword, try/catch for async errors, cleaner syntax patterns

04

Fetch API & REST Basics

fetch(), HTTP verbs (GET/POST/PUT/DELETE), headers, response parsing, JSON handling

05

Working with Public APIs

API keys, query params, reading API docs, displaying fetched data in the DOM

06

Error Handling Patterns

Network failures, status code checks, user-facing error messages, loading states

Mini Project: Build a weather dashboard using the OpenWeatherMap API — search by city, display temperature, conditions, and a 5-day forecast.

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

PHASE 1 · FOUNDATION

Week 4 — Responsive Design & CSS Advanced

Learning Outcome: Build fully responsive, accessible, visually polished layouts using modern CSS techniques that work across all device sizes.

CSS Grid Layout

grid-template-columns/rows, fr units, grid-area, auto-fit/auto-fill, complex two-dimensional layouts

Advanced Flexbox

Nested flex containers, flex-grow/shrink/basis, order, align-self, real layout patterns

Responsive Design Principles

Mobile-first approach, breakpoints, media queries, viewport meta tag, fluid typography

CSS Variables & Custom Properties

:root variables, theming with custom properties, dynamic values, dark mode basics

Animations & Transitions

transition property, @keyframes, transform, animation-timing-function, performance considerations

Tailwind CSS Introduction

Utility-first philosophy, setup, common classes, responsive prefixes, customizing config

Phase 1 Assignment: Responsive multi-page portfolio website — home, about, projects, and contact pages. Mobile-first. Deployed to GitHub Pages or Netlify.

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

PHASE 2 · INTERMEDIATE

Week 5 — React.js Foundations

Learning Outcome: Build component-based UIs in React — understand the mental model shift from vanilla JS, and use state and props fluently.

Core React Concepts

- **JSX** — HTML-like syntax inside JavaScript
- **Components** — functional components, reusability
- **Props** — passing data between components, prop types
- **State** — useState hook, re-rendering, controlled inputs
- **Conditional Rendering** — &&, ternary, early returns
- **Lists & Keys** — .map(), unique key prop, rendering data arrays

React Ecosystem & Tooling

- Vite for fast project scaffolding
- React DevTools browser extension
- Component folder structure best practices
- Styling in React: CSS modules, Tailwind integration

Hooks Deep Dive

- useEffect — side effects, lifecycle equivalents
- useRef — DOM access and mutable values
- Dependency arrays and cleanup functions

Mini Project: Movie search app using the OMDB API — search, display cards, add favorites with local state.

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

PHASE 2 · INTERMEDIATE

Week 6 — Advanced React & State Management

Learning Outcome: Manage complex application state, build multi-page apps with routing, and apply performance optimization patterns used in production.

Context API

createContext, useContext, global state patterns, avoiding prop drilling in medium-scale apps

useReducer Hook

Reducer pattern, dispatch/actions, when to choose over useState, combining with Context

React Router v6

BrowserRouter, Route, Link, useParams, useNavigate, nested routes, protected routes

Custom Hooks

Extracting reusable logic, useFetch, useLocalStorage, useDebounce — industry naming conventions

Performance Optimization

React.memo, useMemo, useCallback, lazy loading with Suspense, avoiding unnecessary re-renders

Form Handling

Controlled vs. uncontrolled forms, React Hook Form library, validation, error messages, submission

Mini Project: E-commerce product listing with cart — add/remove items, cart total, persistent cart using localStorage.

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

PHASE 2 · INTERMEDIATE

Week 7 — Node.js, Express & REST APIs

Learning Outcome: Build a fully functional REST API from scratch — understand the backend runtime, HTTP routing, middleware, and RESTful design conventions.

01

Node.js Runtime

V8 engine, CommonJS modules (require/exports), ES modules, npm ecosystem, package.json, scripts

02

Express.js Core

Setting up a server, routing (GET/POST/PUT/DELETE), req/res objects, sending JSON, status codes

03

Middleware

Built-in middleware (express.json), third-party (cors, morgan, helmet), custom middleware, error handlers

01

RESTful API Design

Resource naming, HTTP verbs, status codes (200/201/400/401/404/500), versioning (/api/v1/)

02

Environment Variables

dotenv, .env files, .gitignore best practices, separating dev/prod config, process.env

03

API Testing with Postman

Creating collections, testing endpoints, environment variables in Postman, writing test scripts

Mini Project: Build a Books API — full CRUD endpoints for a book catalog, in-memory data store, tested with Postman.

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

PHASE 2 · INTERMEDIATE

Week 8 — Databases: SQL, NoSQL & Authentication

Learning Outcome: Connect your Express API to real databases, model data correctly for both relational and document stores, and implement secure user authentication.

1

PostgreSQL & SQL Foundations

Tables, relationships, primary/foreign keys, JOINs, indexes, CRUD with raw SQL, pg library

2

Prisma ORM

Schema definition, migrations, CRUD via Prisma client, relations, seeding, type safety

3

MongoDB & Mongoose

Document model, collections, schemas, CRUD operations, embedded vs. referenced documents

4

Authentication Fundamentals

Hashing passwords with bcrypt, JWT structure (header/payload/signature), signing and verifying tokens

5

Auth Middleware & Routes

Registration, login, protected routes, middleware to verify JWT, role-based access control basics

6

Session vs. Token Auth

Cookies vs. JWT, stateless vs. stateful auth, refresh tokens, OAuth2 overview (Google login)

Phase 2 Assignment: Full stack task manager — React frontend + Express backend + PostgreSQL + JWT auth. Users can register, login, and manage their own tasks.

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

PHASE 3 · ADVANCED

Week 9 — Advanced Backend: Architecture & Design Patterns

Learning Outcome: Architect scalable, maintainable backend systems using patterns that real engineering teams use in production codebases.

MVC Architecture

Separating Models, Controllers, and Routes; folder structure; keeping controllers thin; service layer pattern

Input Validation & Security

Zod/Joi for schema validation, sanitizing inputs, rate limiting with express-rate-limit, CORS policy

File Uploads & Cloud Storage

Multer for handling multipart forms, uploading images to AWS S3 or Cloudinary, serving via CDN

Background Jobs & Queues

Use cases (email, notifications), BullMQ basics, Redis as queue store, job retry strategies

GraphQL Introduction

GraphQL vs. REST, schema/resolvers/queries/mutations, Apollo Server, when to choose GraphQL

Logging & Monitoring

Winston/Pino for structured logging, log levels, Morgan for HTTP logs, intro to Sentry for error tracking

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

PHASE 3 · ADVANCED

Week 10 — DevOps, Cloud Deployment & CI/CD

Learning Outcome: Deploy full stack applications to the cloud with automated pipelines, containerization, and monitoring — the way it's actually done at real companies.

Containerization with Docker

- Writing Dockerfiles for Node.js apps
- docker build, run, exec, ps commands
- Docker Compose for multi-container apps (app + DB)
- Environment variables in Docker

Cloud Deployment Options

- **Vercel / Netlify** — frontend deployment, edge functions
- **Render / Railway** — backend + database hosting for projects
- **AWS EC2 + RDS** — intro to cloud infrastructure
- Environment management in production

CI/CD with GitHub Actions

- Workflows, triggers, jobs, and steps
- Automated testing on every push
- Deploy-on-merge to main branch
- Secrets management in GitHub

Domain, HTTPS & DNS

- Custom domain setup with Namecheap/GoDaddy
- SSL/TLS certificates via Let's Encrypt
- Nginx as reverse proxy basics

Monitoring & Uptime

- UptimeRobot for availability alerts
- Basic metrics: response time, error rate
- Intro to Datadog / New Relic dashboards

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

PHASE 3 · ADVANCED

Week 11 — Performance, Security & System Design Basics

Learning Outcome: Build apps that are fast, secure, and scalable — apply optimization and security techniques that are expected at the junior-to-mid level in any serious engineering team.

Frontend Performance

Code splitting, lazy loading, image optimization (WebP, responsive images), Lighthouse audits, Core Web Vitals (LCP, FID, CLS)

Backend Performance

Database query optimization, N+1 problem, indexing strategies, Redis caching, connection pooling, pagination

Web Security Essentials

OWASP Top 10 (SQL injection, XSS, CSRF), input sanitization, HTTPS enforcement, secure headers with Helmet.js

API Security

JWT best practices, refresh token rotation, brute-force protection, API key management, sensitive data exposure

System Design Fundamentals

Scalability concepts (horizontal vs. vertical), load balancing, CDN usage, stateless services, microservices vs. monolith trade-offs

Testing Practices

Unit tests (Jest/Vitest), integration tests, API testing (Supertest), test coverage, TDD intro, testing React components

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

PHASE 3 · ADVANCED

Week 12 — Capstone Project & Career Readiness

Learning Outcome: Deliver a production-quality full stack application that demonstrates end-to-end mastery — and prepare your portfolio and professional presence for the job market.

Capstone Project Requirements

- Full stack app: React frontend + Node/Express backend
- Database: PostgreSQL or MongoDB (student's choice)
- Auth: JWT-protected routes, role-based access
- Deployed to cloud with custom domain
- CI/CD pipeline via GitHub Actions
- README with setup instructions and screenshots

Career Readiness Checklist

- ✅ GitHub profile — pinned repos, green contribution graph
- ✅ Portfolio website live and linked
- ✅ LinkedIn — projects section with live links
- ✅ Resume — tailored for full stack roles
- ✅ 2–3 strong portfolio projects with descriptions
- ✅ LeetCode — 30+ easy/medium problems solved

Presentation / Demo Expectations

- 5-minute recorded walkthrough of the live app
- Code walkthrough: one frontend + one backend module
- Challenges faced and how you solved them
- What you would improve with more time

Final Deliverable: Submit GitHub repo + live app URL + 5-min demo video. Peer review one classmate's project with structured feedback using a provided rubric.

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

INDUSTRY EXAMPLE 1

Building a SaaS Dashboard for a Fintech Startup

A seed-stage fintech startup needed an internal analytics dashboard for their operations team — showing transaction volumes, fraud alerts, and customer account summaries in real time. The founding engineer had to ship it in 3 weeks with a two-person team.

🏢 Business Problem

Operations staff were manually pulling data from CSVs and spreadsheets daily. No real-time visibility, high error rate, 3–4 hours of daily manual work per analyst

🛠️ Stack Applied

React + Recharts (frontend), Node.js + Express (API), PostgreSQL (transactions DB), JWT auth, deployed on AWS + Nginx

📈 Business Impact

Reduced manual reporting time by **87%**, enabled same-day fraud detection (vs. 48-hr lag), and became a core feature in Series A pitch deck

---

Role of Early Professionals

The junior developer owned the **frontend dashboard components** (charts, filters, table with pagination), worked directly with the backend engineer on API contracts, wrote integration tests, and managed the deployment pipeline. This is exactly the type of end-to-end ownership this course prepares you for.

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

INDUSTRY EXAMPLE 2

E-Commerce Platform Migration for a D2C Brand

A direct-to-consumer apparel brand with $8M in annual revenue needed to migrate off a legacy Shopify setup to a headless commerce architecture — giving their team more flexibility over the buying experience and a 40% improvement in page load speed.

The Problem

Their Shopify theme was slow (3.8s LCP), hard to customize, and couldn't support the new subscription model they planned to launch. The CTO wanted a decoupled frontend that the web team could iterate on independently of the backend catalog.

Solution Architecture

- **Frontend:** Next.js (React) with Tailwind CSS
- **Backend:** Node.js API layer + Shopify Storefront API
- **Database:** PostgreSQL for subscriptions, Redis for cart sessions
- **Deployment:** Vercel for frontend, Railway for API
- **CI/CD:** GitHub Actions with automated Lighthouse checks

Measurable Business Impact

- Page load time: 3.8s → **1.2s** (68% improvement)
- Conversion rate increased by **22%** post-launch
- Subscription feature launched **on schedule** in 6 weeks
- Dev team velocity doubled with decoupled architecture

Junior Dev Contributions

- Built product listing and PDP (product detail page) components
- Implemented cart context with useReducer + localStorage persistence
- Wrote API integration layer for Shopify Storefront queries
- Set up performance monitoring and Lighthouse CI

EDVANTA

INDUSTRY EXAMPLE 3

Healthcare Patient Portal — Enterprise Full Stack Project

A regional hospital network contracted a mid-sized dev agency to build a patient-facing web portal — allowing patients to view records, book appointments, and message their care team securely. The project had strict HIPAA compliance requirements.

Business Challenge

Phone-based scheduling generated 1,200+ calls/day to the front desk. Patient records were siloed across three legacy systems with no unified interface for patients or staff.

Technical Approach

React SPA with role-based views (patient vs. admin), Express API with strict auth middleware, PostgreSQL with row-level security, all hosted on HIPAA-eligible AWS infrastructure (EC2, RDS, S3)

Security & Compliance

End-to-end HTTPS, JWT + refresh tokens, audit logging for every data access event, input sanitization against injection attacks, penetration testing pre-launch

Business Outcome

Phone calls reduced by **61%** in 90 days. Patient satisfaction scores improved by **34%**. Portal onboarded 18,000 patients in the first 3 months. Staff reported 2 fewer FTEs needed for scheduling.

Early professional role: Junior developers handled appointment booking UI components, form validation, API integration for the scheduling service, and contributed to the automated testing suite — shipping production code in a regulated environment from day one.

EDVANTA

Ignite your growth, inspire your mind

CAREER ALIGNMENT

Job Roles Aligned to This Course

Completing this program positions you for a wide range of roles in the current job market. Below is an honest map of where you can realistically land — and what each role typically demands.

Junior Full Stack Developer

The primary target role. React + Node + DB + deployment. Usually at startups or product companies. Expect feature ownership from week one.

Frontend Developer (React)

Specializes in React, state management, performance, and UI implementation. Common at larger teams with a dedicated frontend guild.

Backend / API Developer

Focuses on Node.js services, REST/GraphQL APIs, database design, and auth. Common in companies with microservices or separate frontend teams.

Software Engineer (Generalist)

Broad scope at startups — you may touch infra, mobile (React Native), backend, and frontend in the same sprint. High ownership, fast learning.

Associate DevOps / Platform Engineer

For learners who enjoyed the CI/CD and Docker modules. Entry point to cloud engineering roles at AWS/GCP/Azure partner firms.

Technical Consultant / Solutions Engineer

Combines coding with client communication. Common at SaaS companies and dev agencies. Requires both technical depth and presentation skills.

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

CAREER ALIGNMENT

Skills Progression: Beginner → Job Ready

Use this map to honestly assess where you are and what you need to close the gap. Each level is a realistic checkpoint in your development journey.

1

Beginner (Weeks 1–2)

HTML/CSS/JS basics, Git, static pages, simple DOM manipulation

2

Developing (Weeks 3–4)

Async JS, API calls, responsive layouts, first deployed project on Netlify

3

Intermediate (Weeks 5–6)

React components, hooks, routing, state management, component architecture

4

Capable (Weeks 7–8)

REST APIs with Node/Express, SQL + NoSQL, JWT auth, full stack projects

5

Advanced (Weeks 9–11)

System design basics, Docker, CI/CD, performance, security, testing

6

Job Ready (Week 12+)

Capstone project deployed, portfolio live, resume tailored, interview prep active

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

ROADMAP

Your Roadmap After Completing This Course

Finishing this program is the beginning, not the end. Here's a clear, prioritized plan for the 90 days after graduation — designed to accelerate your first hire.

01

Days 1–30: Consolidate & Ship

Polish your capstone. Add one more portfolio project from a real brief (open-source contribution or freelance). Write a case study for each project on LinkedIn.

02

Days 31–60: Apply & Practice

Apply to 10–15 roles/week. Solve 3–5 LeetCode problems daily (focus: arrays, strings, hashmaps). Do 2 mock interviews per week on Pramp or Interviewing.io.

03

Days 61–90: Specialize & Network

Pick a specialization track (Next.js, TypeScript, AWS, React Native). Attend 2 meetups or virtual events. Reach out to 5 professionals per week on LinkedIn.

Recommended Next Skills to Learn

- **TypeScript** — mandatory at most mid-size+ companies
- **Next.js** — SSR, SSG, App Router, Vercel deployment
- **AWS Certified Cloud Practitioner** — entry-level cloud cert
- **Docker + Kubernetes basics** — container orchestration
- **React Native** — expand to mobile with existing React skills

Stay Current

- Follow: Josh W. Comeau, Kent C. Dodds, Fireship.io
- Read: JavaScript Weekly, Node Weekly newsletters
- Contribute to open source on GitHub (good first issue)
- Build in public — tweet/post your progress weekly

EDVANTA

EDVANTA

Ignite your growth, inspire your mind

SUMMARY

Key Takeaways

Here's what makes this program different — and what you should carry with you through every week of the course.

🏗️ Build, Don't Just Learn

Every week ends with a working project or feature. By graduation, you'll have 6–8 shipped projects in your portfolio — not certificates, not tutorials. Real code, live on the internet.

🎯 Industry-First Curriculum

Every technology in this course was chosen because it appears on job descriptions. We skip legacy tools and teach the stack that employers expect from junior hires in 2024–2025.

🔄 Full Ownership Mindset

You'll own features end-to-end — from database schema to UI interaction to cloud deployment. This is what "full stack" actually means at work, and it's the confidence companies are hiring for.

🤝 Mentoring, Not Just Videos

Live 1:1 sessions every week mean you're never stuck for more than a few days. Real feedback on your code, real answers to your questions, and a network you'll use for years.

> The gap between learning to code and getting hired isn't talent — it's the combination of **shipped projects, industry-relevant skills, and the confidence to own a problem end-to-end.** This course is designed to close that gap in 12 weeks.

EDVANTA