

Duration

16-20 Weeks

AI-Powered MERN Stack Developer Program



Swipe right to explore more



About Us

Knovista Learning is a leading career development platform in Kochi, offering hands-on MERN Stack Development internship programs with 100% placement support across the technology and software development industry.

We empower learners through expert mentorship and real-world projects—including specialized training in **MongoDB, Express.js, React, and Node.js.**

Our comprehensive MERN Stack curriculum is designed to equip students with in-demand full stack development skills, enabling them to build dynamic, scalable web applications.

Through practical exposure, **industry-relevant projects, and continuous guidance**, we help our interns gain strong technical expertise and enterprise-level development experience—preparing them for successful careers in modern web development.

Award-Winning Developers.

Ready To Build

Our MERN Stack training program is designed to create industry-ready full stack developers equipped to build modern, scalable web applications. We focus on transforming learners into professionals who can handle real-world development challenges using practical workflows and live project experience.

With a strong emphasis on hands-on learning, system design thinking, and AI-driven development practices,

we ensure you gain the skills required to work confidently in real production environments from day one.



Trademarks of Our **MERN Developers**



No Beginner Struggles



Strong Frontend Skills with React & Tailwind



Robust Backend Development with Node.js & Express



Real-World Project Development Experience



REST API Design & Integration Expertise



Database Management with MongoDB



Authentication & Security Implementation (JWT, OAuth)



Deployment Tools and AI Tools (Vercel, Render, Cloud, Copilot & APIs)

Who Can Learn MERN Stack Development

- ✓ Diploma Students (IT Stream)
- ✓ BCA / BSc Computer Science / BSc IT Students
- ✓ BE / BTech (Any Branch) Graduates
- ✓ MCA / MSc IT Students
- ✓ Working Professionals (IT & Non-IT)
- ✓ Web Designers & Frontend Developers
- ✓ Career Switchers Interested in Software Development



Minimum system Requirements



What you Will Learn

1. CORE JAVASCRIPT & TYPESCRIPT

JavaScript fundamentals, ES6+, TypeScript basics, types, interfaces, type safety

2. ADVANCED TYPESCRIPT

Generics, utility types, type inference, modules, scalable project structure

3. DATABASE (MONGODB)

CRUD operations, schema design, indexing, MongoDB Atlas, Mongoose

4. FRONT-END TECHNOLOGIES

HTML, CSS, Tailwind CSS, responsive design, modern UI practices

5. REACT JS WITH TYPESCRIPT

Components, props & state typing, hooks, context API, reusable architecture

6. NODE JS & EXPRESS (TS)

Backend development, REST architecture, middleware, routing, MVC pattern with TypeScript

What you Will Learn

7. REST API DEVELOPMENT

API design, authentication (JWT), validation, error handling, API integration

8. FULL STACK INTEGRATION

Connecting frontend & backend, data flow, state management, real-world apps

9. DEPLOYMENT & DEVOPS BASICS

Git & GitHub, Docker basics, environment setup, CI/CD, cloud deployment

Opportunities In the industry

↘ Job opportunities in MERN Stack



- **Frontend Developer (React)**

- **Backend Developer (Node.js / Express)**

- **Full Stack Developer (MERN)**

- **Software Engineer**

- **Web Developer**

- **JavaScript Developer**

- **React Developer**

- **Node.js Developer**

- **API Developer**

- **UI Developer**

- **Application Developer**

- **Junior Developer / Trainee Developer**

- **Freelance Web Developer**

- **Full stack plus AI Developer**

- **Remote Developer**

COURSE OVERVIEW

Program Information

Duration 16–20 Weeks	Modules 8 Core Modules	Format Online / Hybrid	Level Beginner → Advanced
Prerequisites Basic computer literacy	Instructor Knovista Lead Instructor	Platform learn.knovista.dev	GitHub GitHub Classroom

MODULE SCHEDULE

8-Module Roadmap

#	Module Title	Duration	Key Focus Areas
01	The AI Architect's Toolkit	1–2 Wks	Environment, Git, AI Mindset
02	Frontend Foundations (UI + React Basics)	3–4 Wks	HTML, CSS, Tailwind, React
03	The Data Vault (MongoDB)	2 Wks	MongoDB, Mongoose, Schemas
04	The Brains of the App (Node + Express)	2–3 Wks	APIs, Routing, Middleware
05	Connecting the Dots (Full Stack)	1–2 Wks	Axios, CORS, Data Flow
06	Authentication & Security	1–2 Wks	JWT, bcrypt, OAuth Basics
07	Advanced Frontend & Real-Time	2 Wks	Redux, Socket.io, Uploads
08	Debugging, AI Integration & Deployment	1–2 Wks	AI APIs, Deploy, Ship

RESOURCES

Required Materials & Tools

Software (Free)

1. Node.js LTS
2. Git
3. VS Code
4. Postman
5. MongoDB Compass

VS Code Extensions

1. GitHub Copilot
2. Prettier
3. ESLint

COURSE OVERVIEW

Program Information

Duration 16–20 Weeks	Modules 8 Core Modules	Format Online / Hybrid	Level Beginner → Advanced
Prerequisites Basic computer literacy	Instructor Knovista Lead Instructor	Platform learn.knovista.dev	GitHub GitHub Classroom

MODULE SCHEDULE

8-Module Roadmap

#	Module Title	Duration	Key Focus Areas
01	The AI Architect's Toolkit	1–2 Wks	Environment, Git, AI Mindset
02	Frontend Foundations (UI + React Basics)	3–4 Wks	HTML, CSS, Tailwind, React
03	The Data Vault (MongoDB)	2 Wks	MongoDB, Mongoose, Schemas
04	The Brains of the App (Node + Express)	2–3 Wks	APIs, Routing, Middleware
05	Connecting the Dots (Full Stack)	1–2 Wks	Axios, CORS, Data Flow
06	Authentication & Security	1–2 Wks	JWT, bcrypt, OAuth Basics
07	Advanced Frontend & Real-Time	2 Wks	Redux, Socket.io, Uploads
08	Debugging, AI Integration & Deployment	1–2 Wks	AI APIs, Deploy, Ship

RESOURCES

Required Materials & Tools

Software (Free)

1. Node.js LTS
2. Git
3. VS Code
4. Postman
5. MongoDB Compass

VS Code Extensions

1. GitHub Copilot
2. Prettier
3. ESLint

COURSE OVERVIEW

Program Information

Duration 16–20 Weeks	Modules 8 Core Modules	Format Online / Hybrid	Level Beginner → Advanced
Prerequisites Basic computer literacy	Instructor Knovista Lead Instructor	Platform learn.knovista.dev	GitHub GitHub Classroom

MODULE SCHEDULE

8-Module Roadmap

#	Module Title	Duration	Key Focus Areas
01	The AI Architect's Toolkit	1–2 Wks	Environment, Git, AI Mindset
02	Frontend Foundations (UI + React Basics)	3–4 Wks	HTML, CSS, Tailwind, React
03	The Data Vault (MongoDB)	2 Wks	MongoDB, Mongoose, Schemas
04	The Brains of the App (Node + Express)	2–3 Wks	APIs, Routing, Middleware
05	Connecting the Dots (Full Stack)	1–2 Wks	Axios, CORS, Data Flow
06	Authentication & Security	1–2 Wks	JWT, bcrypt, OAuth Basics
07	Advanced Frontend & Real-Time	2 Wks	Redux, Socket.io, Uploads
08	Debugging, AI Integration & Deployment	1–2 Wks	AI APIs, Deploy, Ship

RESOURCES

Required Materials & Tools

Software (Free)

1. Node.js LTS
2. Git
3. VS Code
4. Postman
5. MongoDB Compass

VS Code Extensions

1. GitHub Copilot
2. Prettier
3. ESLint

COURSE OVERVIEW

Program Information

Duration 16–20 Weeks	Modules 8 Core Modules	Format Online / Hybrid	Level Beginner → Advanced
Prerequisites Basic computer literacy	Instructor Knovista Lead Instructor	Platform learn.knovista.dev	GitHub GitHub Classroom

MODULE SCHEDULE

8-Module Roadmap

#	Module Title	Duration	Key Focus Areas
01	The AI Architect's Toolkit	1–2 Wks	Environment, Git, AI Mindset
02	Frontend Foundations (UI + React Basics)	3–4 Wks	HTML, CSS, Tailwind, React
03	The Data Vault (MongoDB)	2 Wks	MongoDB, Mongoose, Schemas
04	The Brains of the App (Node + Express)	2–3 Wks	APIs, Routing, Middleware
05	Connecting the Dots (Full Stack)	1–2 Wks	Axios, CORS, Data Flow
06	Authentication & Security	1–2 Wks	JWT, bcrypt, OAuth Basics
07	Advanced Frontend & Real-Time	2 Wks	Redux, Socket.io, Uploads
08	Debugging, AI Integration & Deployment	1–2 Wks	AI APIs, Deploy, Ship

RESOURCES

Required Materials & Tools

Software (Free)

1. Node.js LTS
2. Git
3. VS Code
4. Postman
5. MongoDB Compass

VS Code Extensions

1. GitHub Copilot
2. Prettier
3. ESLint

Cloud Accounts (Free Tier)

1. GitHub
2. MongoDB Atlas
3. Vercel
4. Render

Reference Docs

1. MDN Web Docs
2. react.dev
3. expressjs.com
4. mongoosejs.com

This syllabus is subject to revision. Updates communicated via the course LMS and Discord.

MODULE DETAILS

Complete Module Breakdown

01

The AI Architect's Toolkit

FOCUS

Environment setup, full-stack system understanding, and developing the English-to-Code AI mindset.

CONTEXT

Before writing any code, students learn how all MERN layers communicate and how to leverage AI as a thinking partner — not just a code generator.

TOPICS

Core Concepts

1. What is Full Stack Development
2. **MERN Architecture**
 1. Frontend (React)
 2. Backend (Node + Express)
 3. Database (MongoDB)
3. Client-Server Architecture & Request-Response Lifecycle
4. Introduction to Version Control Systems (VCS)

Environment Setup

1. Install Node.js, Git, VS Code
2. **VS Code Extensions**
 1. GitHub Copilot
 2. Prettier
 3. ESLint

Git & GitHub

1. git init, clone, add, commit, push, pull
2. Branching, merging, repository management

AI COPILOT SKILLS

1. Use @workspace to understand project structure
2. Generate project ideas, folder structure, README
3. Prompt: 'Explain how this project works step by step'
4. Convert idea to technical breakdown

PROJECT TASK

1. Generate a Project Blueprint with feature list
2. Define folder structure and write README.md

Goal — Understand how all parts of MERN communicate, without writing code.

02

Modern Frontend & Type-Safe Development Foundations

FOCUS

Master JavaScript fundamentals, TypeScript type safety, and modern web UI architecture.

CONTEXT

JavaScript is the foundation of all MERN development — React, Node, and Express all depend on it. TypeScript adds powerful type safety across your entire full-stack application. Together, they form the backbone of professional frontend development.

TOPICS

JavaScript (Core Logic)

1. Variables (var, let, const) & scope management
2. Data types & type coercion
3. **Functions**
 1. Arrow functions
 2. Callbacks & closures
4. **Objects & Arrays**
 1. Deep understanding
 2. Methods & iteration
5. **ES6+ Features**
 1. Destructuring
 2. Spread/rest operators
 3. Template literals
6. DOM basics & manipulation (essential for React understanding)
7. **Async JavaScript**
 1. Promises
 2. async/await
 3. Fetch API
8. **Modules**
 1. import/export
 2. CommonJS vs ESM

TypeScript (Core + MERN Integration)

1. **TypeScript Fundamentals**
 1. Types, interfaces, type aliases
 2. Functions & object typing
 3. Union & optional types
 4. Generics basics
2. **TypeScript with Node.js & Express**
 1. Typed request/response objects
 2. Middleware typing
 3. API route structure with types
3. **MongoDB & Mongoose with TypeScript**
 1. Schema typing
 2. Model interfaces
 3. Document types
4. **React + TypeScript (Frontend Integration)**

1. Props & state typing
2. Event handler typing
3. Hooks typing (useState, useEffect, useContext)
4. Component type definitions

HTML (Complete)

1. Document structure, semantic tags, headings, paragraphs
2. Links, images, lists, tables, forms & input validation
3. Iframe, media (audio/video), SVG basics
4. Meta tags, SEO basics, Accessibility (ARIA)

CSS (Complete)

1. Selectors, box model, typography, colors & backgrounds
2. Positioning, Flexbox, Grid
3. Transitions, animations, media queries

Tailwind CSS

1. Utility-first styling & layout utilities
2. Spacing, typography, responsive design
3. Reusable components, custom configuration

React Basics

1. React setup with Vite, JSX
2. Functional components, Props, useState
3. Event handling, conditional rendering
4. Lists & keys

AI COPILOT SKILLS

1. Generate type definitions from code snippets
2. Refactor JavaScript to TypeScript
3. Generate UI from natural-language prompts
4. Convert HTML to React components with type safety
5. Convert CSS to Tailwind utilities
6. Generate reusable, typed component templates

PROJECT TASK

1. Build type-safe JavaScript utilities & helper functions
2. Convert a plain JavaScript file to fully typed TypeScript
3. Build a responsive landing page with proper typing
4. Create a dashboard layout with Navbar + Sidebar using TypeScript
5. Build reusable Task card components with type-safe Props

Goal — Master JavaScript & TypeScript fundamentals to write type-safe, maintainable code across your full-stack application.

03

The Data Vault (MongoDB)

FOCUS

Designing and managing application data structures correctly from the start.

CONTEXT

Bad data design leads to broken applications. Students must think in data structures before writing any backend logic.

TOPICS**Core Concepts**

1. NoSQL vs SQL
2. Collections & Documents
3. Schema design principles

MongoDB

1. Installation / MongoDB Atlas setup
2. Database & collections
3. CRUD operations, data types, querying
4. Sorting, limiting, indexing
5. Aggregation pipeline

Mongoose

1. Schema definition & Models
2. Validation
3. Relationships (references & embedded documents)

AI COPILOT SKILLS

1. Prompt AI for schema design suggestions
2. Generate queries from natural language
3. Explain data relationships in plain English

PROJECT TASK

1. Create User, Project, and Task schemas
2. Set up MongoDB Atlas cluster
3. Connect the database to the application

Goal — Structure data so the system can scale and remain maintainable.

04**The Brains of the App (Node.js & Express)****FOCUS**

Building backend logic and RESTful APIs that power the application.

CONTEXT

The backend handles all requests, processes logic, and communicates with the database — it is the decision-making brain of the application.

TOPICS**Node.js**

1. Node architecture & event loop
2. Modules & NPM
3. File system, Streams & buffers

Express.js

1. Express setup & Routing
2. HTTP methods (GET, POST, PUT, DELETE)
3. Middleware, Request & response handling
4. Static files

API Development

1. REST API design principles
2. CRUD operations & status codes
3. Error handling

Advanced

1. Async handlers, Form data
2. File uploads with Multer
3. API testing with Postman

AI COPILOT SKILLS

1. Generate server boilerplate instantly
2. Create complete API routes from descriptions
3. Debug backend errors with AI explanations

PROJECT TASK

1. Build CRUD APIs for Task management

Goal — Understand how the backend processes requests and communicates with data.

05

Connecting the Dots (Full Stack Integration)

FOCUS

Wiring Frontend, Backend, and Database together into a unified application.

CONTEXT

This is where most beginners struggle — understanding data flow across all three layers and keeping state synchronised in real time.

TOPICS

Topics

1. Fetch API & Axios for HTTP requests
2. useEffect for data fetching lifecycle
3. Handling loading states & errors gracefully
4. CORS (Cross-Origin Resource Sharing) configuration
5. JSON data flow end-to-end

AI COPILOT SKILLS

1. Debug API connection issues
2. Fix CORS errors automatically
3. Explain data flow from UI to DB

PROJECT TASK

1. Connect React frontend to Express backend
2. Connect backend to MongoDB
3. Display live data in the UI

Goal — Build a fully connected, end-to-end functional application.

06**Authentication & Security****FOCUS**

Implementing robust user identity management and protecting application routes.

CONTEXT

Every real-world application requires user identity and protection. Security is not optional — it is a core feature that must be built in from the start.

TOPICS**Topics**

1. Authentication vs Authorization
2. JWT (JSON Web Token) authentication flow
3. Password hashing with bcrypt
4. Login & signup flow design
5. Protected routes & middleware guards
6. Sessions, cookies, and OAuth basics

AI COPILOT SKILLS

1. Generate complete authentication flows
2. Debug JWT token issues

PROJECT TASK

1. Implement Signup and Login with JWT
2. Add protected route middleware

Goal — Secure your application properly before deploying to production.

07**Advanced Frontend & Real-Time Features****FOCUS**

Handling application complexity, performance, and real-time communication.

CONTEXT

The application now needs to handle real users at real scale. This module separates a demo project from a genuine product.

TOPICS**Advanced React**

1. useEffect deep dive

- 2. Custom hooks
- 3. Forms (controlled & uncontrolled)
- 4. Context API
- 5. Redux (global state management)
- 6. useMemo & useCallback for performance

Advanced Features

- 1. Real-time communication with Socket.io
- 2. File uploads
- 3. API optimisation strategies

AI COPILOT SKILLS

- 1. Debug performance issues and re-renders
- 2. Optimise component rendering

PROJECT TASK

- 1. Add real-time updates via Socket.io
- 2. Implement file uploads
- 3. Integrate global state with Redux

Goal — Build a scalable and optimised frontend ready for production.

08

Debugging, AI Integration & Deployment

FOCUS

Finalising, AI-enhancing, and deploying the application to the cloud.

CONTEXT

This module transforms your project into a real-world product — fully debugged, AI-enhanced, and live on the internet for real users.

TOPICS

Debugging

- 1. Error handling strategies & logging
- 2. Performance tuning and profiling

AI Integration

- 1. OpenAI / Gemini API integration
- 2. AI task suggestions and content summarisation

Deployment

- 1. Environment variables & build configuration
- 2. **Cloud Hosting**
 - 1. Frontend: Vercel / Netlify
 - 2. Backend: Render
 - 3. Database: MongoDB Atlas
- 3. AWS fundamentals overview

AI COPILOT SKILLS

1. Use /fix for automated debugging
2. Use /explain to understand error root causes
3. Generate deployment configuration files

PROJECT TASK

1. Deploy the complete application to cloud platforms
2. Add AI-powered features: task suggestions & summaries

Goal — Ship a production-ready, AI-powered MERN Stack application.