

*\*\*Each module usually has 10-12 videos.*

## Mission 0:

# Welcome to the Next Level Course

Module 0	Welcome
Module 0.5	<b>Orientation: Course Overview</b>  How to use the website, how the group works, our assignments, modules, know about conceptual sessions and many more.

## Mission 1:

# Be A Typescript Technocrat

Module 1	<b>Getting Started with TypeScript – Basics &amp; Setup</b>  This module covers the foundation by exploring typeScript setup, basic data types, array, objects, functions, optional types, literal types, spread operator, rest operator, destructuring, type alias, union & intersection types and many more.
Module 2	<b>Advanced TypeScript – Deep Dive Into Types</b>  This module is designed to level up your typeScript skills by exploring types, interfaces, type assertion, type guards, generics, constants,

	asynchronous typescript, conditional, mapped and utility types with real-world use cases and more.
Module 3	<b>Object-Oriented TypeScript – Master OOP Principles</b>  This module explores object-oriented programming in TypeScript, covering object, class, access modifiers, getter, setter, statics, polymorphism, inheritance, abstraction, encapsulation, and more.
Module 4	Assignment 1

## Mission 2:

# Be A RDBMS Rockstar

Module 5	<b>Database Foundations &amp; Relational Thinking</b>  This module introduces database basics—understanding data, information, and DBMS. Explore relational models, table structure, and key types like super, primary, foreign, and composite keys. Learn top-down design techniques, relationship types, and cardinality. Finally, use tools to create your first ER diagram..
Module 6	<b>Database Normalization &amp; ER Modeling</b>  Learn to structure databases efficiently using normalization. Understand anomalies, functional dependency, and apply 1NF, 2NF, and 3NF. Resolve many-to-many relationships with junction tables and complete your ER diagram. Get started with PostgreSQL, install it, and explore data flow using PSQL.

Module 7	<b>Case Study Solutions, Database Management and Data Types</b>  Install PostgreSQL, set up your environment, and understand how PostgreSQL works within real-world applications. Explore the data flow, psql, and visual tools like pgAdmin & Valentina Studio.
Module 8	<b>PostgreSQL Fundamentals and Advanced Data Manipulation Techniques</b>  Learn to use ALTER to modify tables in PostgreSQL and set up in VS Code. Master SELECT queries, column aliasing, and result ordering. Filter data with WHERE, logical and comparison operators. Explore scalar/aggregate functions, NOT, NULL handling, and operators like IN, BETWEEN, LIKE, and ILIKE. Understand pagination, data deletion, and the UPDATE operator.
Module 9	<b>Managing Dates, Relationships And Understanding JOIN in PostgreSQL</b>  This module covers PostgreSQL topics like date handling, grouping and filtering with GROUP BY and HAVING, and constructing relationships with foreign key constraints. Learn about referential integrity during insertion and deletion, and explore various joins—INNER, LEFT, RIGHT, FULL, CROSS, and NATURAL. End with practical query exercises.
Module 10	<b>Advanced Query Techniques and Database Objects in PostgreSQL</b>  This module covers advanced PostgreSQL topics, including subqueries, views, functions, and stored procedures. Learn to implement triggers and indexing techniques to optimize query performance and improve database efficiency.
Module 11	Assignment 2

## Mission 3:

# Be A Prisma Express Expert

Module 12	<b>Explore The Fundamentals of Node.js</b>  Understand the core of Node.js, its power for backend development, and explore essential modules like fs, http, path, and more.
Module 13	<b>Mastering The Foundation Of Express</b>  Learn to create routes, handle requests and responses, use middleware, and structure your server for scalable backend applications.
Module 14	<b>Prisma ORM Fundamentals</b>  Learn the essentials, from installation and schema setup to mastering the Prisma workflow. Understand data model mapping, and explore powerful tools like Prisma Client, Prisma Migrations, and Prisma Studio for seamless database management.
Module 15	<b>Prisma Relation Queries and Filtering</b>  Dive into Prisma relationships! Learn how to handle relation queries, apply filters, and implement advanced techniques like pagination, sorting, and dynamic filtering to manage and retrieve data with precision.
Module 16	<b>Prisma Advanced Patterns</b>  Master advanced Prisma patterns like upsert operations, unique constraints, nested writes, and aggregation. Optimize your queries using

	raw SQL, and fine-tune database migrations with seamless rollback capabilities for improved performance and reliability.
Module 17	<b>Advanced CRUD with Typescript, Express &amp; Prisma</b>  Leverage TypeScript, Express, and Prisma to build powerful CRUD operations. Manage transactions with ease, boost performance, handle complex database tasks efficiently, and enable detailed logging for better monitoring and debugging.
Module 18	Assignment 3

## Mission 4:

# Be A Redux Reaper

Module 19	<b>Fundamentals of State Management</b>  Basics of state management and its importance in building scalable apps, props drilling issues, roadblocks of React state management, understanding the benefits of global state management and more.
Module 20	<b>Introduction to Redux</b>  Discover Redux and its core components—actions, dispatch, reducers, and store. Understand how it helps manage global state in complex applications.
Module 21	<b>Hands-on Redux Toolkit</b>

	Get hands-on with Redux Toolkit to perform CRUD operations, manage middlewares, and sync state with local storage.
Module 22	<b>Fundamentals of RTK Query</b>  RTK Query for efficient data fetching, error handling, caching, and optimistic updates, dependent queries, real-time data through polling and subscription.
Module 23	Assignment 4

## Mission 5:

# Be An Industry Standard Backend Brainiac

## Project Details:

The PH Tour Management System is a full-stack platform designed for customers, and admins, built with Typescript, Node.js, Express, Prisma, PostgreSQL, TailwindCSS & Redux.

It offers features like role based authentication with JWT, Access token & Refresh token, Google OAuth, and OTP verification. Users can manage profiles, tour packages, and bookings, with dynamic pricing and real-time seat availability. Payments are integrated through Stripe and SSLCommerz, while smart search, sorting, filtering, pagination etc. and AI recommendations enhance the user experience. Admins can manage reviews, send notifications, and track revenue with

detailed reports. The system leverages tools like Passport.js ,Winston for logging, Redis for caching and real-time updates via Socket.IO for seamless communication and efficient tour management.

Module 24	Industry Standard Project 1 : Initiation and Analysis
Module 25	Industry Standard Project 1 : Backend (Part 1)
Module 26	Industry Standard Project 1 : Backend (Part 2)
Module 27	Industry Standard Project 1 : Backend (Part 3)
Module 28	Industry Standard Project 1 : Backend (Part 4)
Module 29	Industry Standard Project 1 : Backend (Part 5)
Module 30	Industry Standard Project 1 : Backend (Part 6)
Module 31	Industry Standard Project 1 : Backend (Part 7)
Module 32	Assignment 5

## Mission 6:

# Be An Industry Standard Frontend Futurist

Module 33	Industry Standard Project 1 : Frontend (Part 1)
Module 34	Industry Standard Project 1 : Frontend (Part 2)
Module 35	Industry Standard Project 1 : Frontend (Part 3)
Module 36	Industry Standard Project 1 : Frontend (Part 4)
Module 37	Industry Standard Project 1 : Frontend (Part 5)
Module 38	Industry Standard Project 1 : Frontend (Part 6)
Module 39	Industry Standard Project 1 : Frontend (Part 7)
Module 40	Assignment 6



## Mission 7:

# Be A Next.js Mongoose Magician

Module 41	<b>In-Depth Exploration of MongoDB Queries</b>  Master MongoDB queries using tools like MongoDB Compass and NoSQLBooster. Insert and retrieve data with insert, insertOne, find, and findOne. Filter fields with operators like \$eq, \$ne, \$gt, \$lt, \$gte, \$lte, \$in, \$nin, \$and, and \$or. Use \$exists, \$type, and \$size for deep filtering, and \$all, \$elemMatch for arrays. Update data with \$set, \$addToSet, \$push, and clean with \$unset, \$pop, \$pull. Learn advanced querying and deletion techniques with delete and drop.
Module 42	<b>Mastering MongoDB Aggregation &amp; Indexing</b>  Explore MongoDB's Aggregation Framework with stages like \$match, \$project, \$addFields, \$group, and \$sum to filter, shape, and group data. Work with arrays using \$unwind, sort and limit results with \$sort and \$limit, and join collections with \$lookup. Learn advanced stages like \$facet and \$bucket, and optimize queries with indexing techniques such as COLLSCAN vs IXSCAN, compound, and text indexes..
Module 43	<b>Explore Mongoose CRUD Operation</b>  Learn data validation in MongoDB with Mongoose, using custom validation methods, instance and static methods, and middlewares. Explore schema-based validation with packages like Validator, Joi, and Zod. Discover new ways to delete data, apply query middlewares, and use Mongoose virtuals for computed fields.

Module 44	<b>Fundamentals of Next.js</b>  Explore why Next.js stands out over React for production apps. Learn pre-rendering vs. client-side rendering, create your first app, and understand routing from basic to dynamic and nested. Dive into React Server Components, navigation methods, layout-based group routing, and built-in CSS module support.
Module 45	<b>Next.js Core Concepts Part-1</b>  Master Next.js optimization with the Image component, custom fonts, and SEO-friendly metadata. Understand pre-rendering through SSG, SSR, and ISR. Set up JSON Server to explore real-world data fetching. Learn advanced strategies for SSR, handle global errors, and manage active links for a better user experience.
Module 46	<b>Next.js Core Concepts Part-2</b>  Learn caching, revalidation, and dynamic data fetching using SSR and SSG. Create blogs with form components and Server Actions. Generate dynamic metadata and build RESTful APIs using route handlers. Finally, explore client-side data fetching and loading states with RTK Query.
Module 47	<b>Next.js Authentication System</b>  Integrate NextAuth for authentication. Implement GitHub and Google login, manage sessions, and secure routes with middleware. Handle user registration and login using Server Actions with a custom server. Finally, build and deploy your app on Vercel.
Module 48	<b>Assignment 7 Release</b>

## Mission 8:

# Be A Next.js Project Perfectionist

## Project Details:

NextMart is a full-stack e-commerce platform where users can either create shops or purchase products. It's built with Next.js, TypeScript, Node.js, Express, MongoDB, Mongoose, TailwindCSS, Redux, and ShadCN UI.

It features JWT-based authentication, refresh tokens, Google ReCAPTCHA, and Zod for form validation. User sessions are managed using cookies, Context API, and auto-refreshing tokens. Users can manage profiles, create shops, upload product images, and handle dynamic product data with secure route protection via middleware.

The frontend uses Next.js route groups for personalized dashboards. The home page includes Hero Section, Categories, and Featured Products, while product browsing supports pagination, detailed views, and a Redux-powered cart for shipping, payment, and optional coupon codes.

NextMart delivers a seamless, scalable shopping experience tailored for both sellers and buyers under a unified user system.

Module 49	Industry Standard Project 2 : Part 1
Module 50	Industry Standard Project 2 : Part 2
Module 51	Industry Standard Project 2 : Part 3
Module 52	Industry Standard Project 2 : Part 4

Module 53	Industry Standard Project 2 : Part 5
Module 54	Industry Standard Project 2 : Part 6
Module 55	Assignment 8

## Reward Courses

### Be A Testing Titan

Module 1	<b>Getting Started with Unit And Integration Tests</b>  Learn the basics of testing and TDD. Compare Jest and Vitest, write your first unit test, and follow best practices like the AAA pattern. Explore testing errors, integration tests, and asynchronous code. Understand testing hooks, mocks, and spies for better test management.
Module 2	<b>Module 2: Advanced Mocking and Testing React Apps</b>  Dive into advanced mocking techniques and learn how to mock modules globally. Understand what to test in React apps and explore different testing environments, including DOM testing. Get started with React Testing Library (RTL), testing components step-by-step, and efficiently querying elements with methods like getByRole() and queryBy().
Module 3	<b>Module 3: Debugging React Tests &amp; REST API Testing with Jest and Supertest</b>  Learn to use the findBy() method and explore the testing playground. Debug React tests with data-testid and test user events like clicks and typing. Understand how to spy on functions and test React with Redux. Transition to REST API testing with Jest and Supertest, set up tests in an

	Express app, and use MongoDB Memory Server to simulate MongoDB for efficient testing.
--	---

## Be A GraphQL Guru

Module 1	<b>Getting Started with GraphQL</b>  Set up a GraphQL project, define schemas and resolvers, and retrieve data like product by ID. Learn to structure files and establish relationships, such as one-to-one (product-category) and one-to-many (category-products, product-reviews). Wrap up with a summary.
Module 2	<b>Building Blog App (Part-I)</b>  Analyze project requirements, initialize the Apollo Server, and integrate Prisma with data models. Define GraphQL schemas, implement mutation queries, and retrieve user data. Learn secure password hashing, JWT token generation, and user login with mutation queries. Focus on data validation, file structure organization, and managing environment variables, concluding the project setup.
Module 3	<b>Building Blog App (Part-II)</b>  Structure files and define context parameters. Create and update posts with mutation queries in multiple steps, including deletion. Implement post publishing, retrieving post data, and fetching profile data by ID. Establish relationships between posts and authors, managing user-post connections and profile-user relationships.
Module 4	<b>Building Blog App (Part-III)</b>  Identify the N+1 problem and boost performance using DataLoader to batch and cache queries. Link the client to the server with Apollo Client and display post data in the UI. Manage user registration and login by

	storing tokens in local storage. Finally, wrap up the app development process, ensuring all features work efficiently.
--	--

## Be A Cloud-Container Craftsman

Module 1	<b>Introduction to Docker</b>  Get started with Docker by understanding what it is and why it's used. Learn the difference between virtual machines and containers. Set up Docker on Windows, Mac, or Linux, explore Docker tools, and configure VS Code to run your first container.
Module 2	<b>Docker Images &amp; Containers</b>  Understand the difference between images and containers. Use pre-built images and write your own Dockerfile. Build custom images, run containers, and explore attached vs detached modes. Learn to manage, tag, push, and pull Docker images with DockerHub.
Module 3	<b>Docker Volumes And Managing Data</b>  Learn how to manage app data using Docker volumes. Understand data categories, use named volumes and bind mounts, and combine different volume types. Work with .env and .dockerignore files, and run containers directly from VS Code.
Module 4	<b>Docker Networks And Cross Container Communication</b>  Understand how containers communicate—with the web, the host machine, and each other. Explore real-world cases, analyze app

	communication flow, and set up Docker networks. Learn how Docker handles IP addresses and manages networks effectively.
Module 5	<b>Building Multi-Container Application With Docker</b>  Learn to build a full-stack app using multiple containers. Dockerize MongoDB, Node.js, and Next.js apps. Run and connect them, add bind mounts, persist MongoDB data, and secure your database container.
Module 6	<b>Multi-Container Orchestration with Docker Compose &amp; Utility Containers</b>  Learn to use Docker Compose to manage multi-container applications. Create a Docker Compose file with services for MongoDB, Node-Express, and NextJS. Explore Docker utility containers and build a Node-Express project from scratch using these containers for a streamlined development process.
Module 7	<b>Getting Started With Cloud Computing</b>  Understand what cloud computing is and why it's important. Explore AWS deployment models, services, and global infrastructure. Set up your AWS account, manage budgets, and learn IAM fundamentals. Get hands-on with user roles, policies, and dive into AWS S3 for scalable object storage.
Module 8	<b>AWS S3 Deep Dive &amp; File Upload</b>  Learn to control S3 object access with custom policies. Host static websites, explore storage classes, and manage versioning and lifecycle rules. Generate access keys and upload files programmatically using Node.js, Express, and the latest AWS SDK. Wrap up with a look at pre-signed URLs for secure file access.
Module 9	<b>Mastering Linux Essentials: A Beginner's Guide</b>  Learn the basics of Linux, focusing on Ubuntu containers, shell commands, and essential system commands. Understand man pages, command options, and arguments. Master file system navigation, file creation,

	deletion, and manipulation. Get familiar with package managers, installing Nano and Vim, and using Vim for file management.
Module 10	<p><b>Mastering Linux Essentials: Permissions, Processes, And Utilities</b></p> <p>Explore the less, head, and tail commands for file viewing. Learn about file permissions, user creation, and controlling access with chmod. Understand the root user and sudo command for elevated permissions. Use the chown command to change file and folder ownership. Get a quick overview of the grep command and master process monitoring with ps, top, and htop.</p>
Module 11	<p><b>EC2 Essentials: Build, Deploy, And Scale With AWS</b></p> <p>Learn what EC2 is and how it works. Create your first EC2 instance on AWS and explore the Free Tier. Understand SSH access with .pem key and security groups. Set up Node.js on EC2, clone projects, and install dependencies. Run your first Node.js server on EC2 and access it online. Explore PM2 for managing Node.js processes and configure it for multiple apps with auto-start on reboot. Understand EC2 lifecycle, Elastic IP, and work with Elastic Block Store (EBS).</p>
Module 12	<p><b>Server Deployment With NGINX And EC2</b></p> <p>Start with SSH and check service status. Learn how to generate SSH keys and connect to a remote server. Understand ports, NGINX, and proxy types. Install EC2 and set up NGINX. Explore the NGINX default page and configuration. Customize NGINX configuration files for your needs. Host a static website using NGINX and set up Node.js with PM2 for your app. Configure NGINX as a reverse proxy and connect your server to a Namecheap domain. Add CNAME and SSL for secure website deployment.</p>
Module 13	<p><b>Automating CI/CD With GitHub Actions</b></p> <p>Learn about GitHub Actions, its purpose, and why it's essential for automation. Explore key workflow points and write your first workflow. Understand YAML file structure and run your first workflow. Create another workflow and dive into the execution environment. Use</p>



	third-party actions to copy code and install Node.js. Handle job failures and add multiple jobs to a workflow. Learn to run jobs sequentially and explore events and types in GitHub Actions for better automation control.
Module 14	<b>Hosting An Application With GitHub Actions (CI/CD)</b>  This module guides you through the process of hosting an application using GitHub Actions and CI/CD. It's divided into multiple parts, each covering different stages of the hosting process. From initial setup to deploying and configuring your application, you will learn how to automate hosting tasks effectively using GitHub Actions. The module concludes with final thoughts and a wrap-up, ensuring you understand how to apply CI/CD for continuous deployment of your application.