**Typescript:**

* Introduction to TypeScript
* Setting up a TypeScript Project
* TypeScript Variables and Data Types
* Type Annotations and Inference
* Functions in TypeScript
* Type Guards/ Type Narrowing
* Arrays and Tuples
* Classes and Interfaces
* Modules and Namespace
* Generics in TypeScript
* Generics with Interface
* Extending Interface Using Generics
* Advanced Type Inference and Union Types
* Conditional Types in TypeScript
* Mapped Types in TypeScript
* Type Guards and Type Assertions
* Intersection Types and Discriminated Unions
* Type Aliases and Utility Types
* Keyof and Lookup Types

**Mongoose:**

* Introduction to Mongoose and MongoDB
* Creating a Mongoose schema
* Defining Mongoose models for CRUD operations
* Data validation with Mongoose
* Understanding Mongoose middleware
* Querying with Mongoose, including aggregation, sorting, filtering, and pagination
* Populating referenced documents with Mongoose
* Integrating Mongoose with Express for building RESTful APIs
* Building RESTful APIs with Mongoose and Express
* Handling relationships between data models in Mongoose and Express
* Advanced querying with Mongoose, including aggregate pipelines, and geospatial queries
* Using Mongoose virtuals
* Handling concurrency with Mongoose
* Optimizing performance with indexing
* Introduction to Mongoose aggregation
* Aggregation pipeline operators
* Aggregating data with Mongoose

**Build Backend of a Production Grade Project Using Typescript , Express , Mongoose and Rest API**

**Redux**

* React's state and reactivity.
* React data flow and its limitations.
* Global state management.
* Redux's inner working.
* Redux Store, Action and Dispatch.
* React Redux and Redux Thunk.
* Redux Toolkit.
* RTK Query.
* Action types and Action Creators.
* Redux DevTool.
* Redux middlewares and custom middleware.
* CRUD operation using Thunk.
* Handling async task using createAsync Thunk in Reduxt Toolkit.
* Handling data mutation and CRUD operation using RTK Query.
* Caching
* Revalidating
* Comprehensive Guide to Testing JavaScript Applications using Jest, Vitest, RTL, and Cypress

**Building the Frontend of a Production-Grade Project Using React, Redux, and TypeScript**

**Testing**

* **Getting Started with Testing**
* Overview of why testing is important
* Different types of tests (unit, integration, end-to-end, etc.)
* Overview of popular testing frameworks and libraries
* Installing and setting up testing frameworks (e.g. Vitest, Jest)
* Jest vs Vitest
* Configuring testing environment
* Course project setup
* Introduction to unit testing
* Writing your first unit test with Vitest
* AAA(Arrange, Act, Sort) Pattern
* **Unit & Integration Tests**
* Introduction to TDD
* How to use TDD with JavaScript
* The red-green-refactor cycle
* Best practices for writing and organizing tests
* Test with multiple assertions
* Writing test for thrown errors and error messages
* Introduction to integration testing
* Writing integration tests
* Overview of code coverage
* **Advanced Testing**
* Testing Asynchronous Code With or without callback
* Getting started with testing hooks
* Using Testing Hooks (beforeEach, beforeAll, afterEach, afterAll)
* Concurrency and concurrent tests
* Introducing Spies & Mocks
* Working with Spies
* Working with Mocks
* Mocking Front-end Libraries
* Test-specific Mocking Functionality
* Introduction to Dom Testing
* Testing Dom Functionality
* **React Testing Library and Cypress**
* Getting Started with React Testing Library
* Introduction to User Event in React Testing
* Testing React Components with react-testing-library
* Testing React Hooks
* Mocking API Requests in React Testing (MSW)
* How to Test Async Code with React Testing Library
* Advanced React Testing Techniques with react-testing-library
* Testing Redux with react-testing-library
* Debugging Tests and Testing Playground
* Getting Started with End-to-End Testing in React Apps with Cypress
* Setting up Cypress for E2E Testing in a React App
* Writing Your First Cypress Test for a React App
* How to Test User Interactions in a React App with Cypress
* **Api Testing**
* Getting Started with Testing Express APIs
* Writing Unit Tests for Express APIs with Jest
* Integration Testing Express APIs with Supertest
* Writing Test Suites for CRUD Operations in Express APIs
* Mocking Dependencies in Express API Tests
* Advanced Testing Techniques for Express APIs with Jest and Supertest
* Tips and Tricks for Testing Express APIs

**GraphQL**

* GraphQL Schema
* GraphQL Supported types
* Resolvers
* Queries
* Relating Data ( one to many, many to one, many to many and one to one)
* The context param
* Filtering and Input type
* Mutations
* Error Handling
* Authentication and authorization
* Data Loaders and The N+1 Problem
* Connecting To Client

**Convert The Project Using GraphQL with Best Practices and Implementation Strategies**

**Database Management System:**

* Introduction to Database Management Systems (DBMS) and Different Database Models
* Database Design: Relational Diagrams and Modeling
* Relational Database Concepts: Normalization, Joins, and Indexing
* SQL Basics: Syntax, Data Types, and Querying with phpMyAdmin
* Postgres: Installation, Configuration, Basic to Advanced Features
* Integrating Postgres with Node API: Using Raw Driver and Best Practices
* Query Optimization: Techniques and Tools for Improving Database Performance
* Production-Grade ORM with TypeScript: Using Prisma for Type-Safe Database Access

Final Project: Implementing a Database-Driven Backend Web Application Using Node.js, Postgres, and Prisma

**NextJS:**

* Let’s start our journey with Next.js
* Why Next.js? The Competitive Edge of Next.js over React.js
* Exploring the World of Next.js Project Creation and its Folder Structure
* Clear Concept of Pre-Rendering and Client-Side Rendering
* Key Features of Next.js
* Routing System
  + File-Based Routing
  + Nested Routing & Catch All Routes
  + Dynamic Routes
  + Shallow Routing
* Layout
  + Basic Layout
  + Nested Layout
* Navigating Using Link Component and Programmatically
* Custom 404 Error Page
* Difference Between Components and Pages
* Image Component
* Using Head Component for Better SEO
* Adding CSS Component Library
* Data Fetching
  + Data Fetching using getStaticProps()
  + How getStaticProps() Works Under the Hood
  + Dynamic Data Fetching using getStaticPaths()
  + How getStaticPaths() Works Under the Hood
  + Concept of Incremental Static Regeneration (ISR)
  + Data fetching using getServerSideProps()
  + How getServerSideProps() Works Under the Hood
  + Client-Side Data Fetching using React Query
* Basic Idea on API Routes
* Use of ENV
* Authentication
  + Introduction of NextAuth
  + Implement Google Authentication System
  + Get Logged-In User from useSession
  + Implement GitHub Authentication System
  + Protect a Route from Unauthorized Users Using Middleware
  + Create a Login Form for Email Password Authentication
  + Implement Email Password-Based Authentication
  + Protect Routes using Session-Based Authentication
* Deployment
  + Build and Deploy Next.js Project in Vercel

Final Project: Implementing a Frontend Application Using NextJS

**AWS & DigitalOcean, Docker, NGINX, and CI/CD**

* Linux Command Line Interface (CLI) Fundamentals
* Linux Filesystem Essentials
* TCP/IP Protocol and Basic Networking Concepts
* Nginx Web Server Configuration and Administration
* Load Balancing using Niginix
* Docker Containerization and Deployment
* Continuous Integration/Continuous Deployment (CI/CD) with GitHub Actions
* Fullstack Deployment on DigitalOcean/Any VPS
* AWS
  + AWS Overview: Introduction to AWS, Console and Dashboard Navigation
  + AWS Identity and Access Management (IAM): User and Role Management, Policies and Permissions
  + AWS S3: Object Storage Essentials and Hosting a Static Website, Uploading Files
  + AWS EC2 and Nginx: Hosting Basic Website and Backend on EC2 Instances
  + SSL on Domain: Setting up SSL Certificates and Encryption on AWS Route 53
  + Application Load Balancing
  + AWS CloudFront: Content Delivery Network (CDN) Configuration and Management

Bonus: React Native