



Internship Program 2024

Mastering Back-end Development

Week 1: Foundational Knowledge

Day 1	
Topics	<ul style="list-style-type: none">• Introductory Session, Client-server Architecture, HTTP vs HTTPS• JavaScript Basics: Variables, Constants, Data Types
Learn	<ul style="list-style-type: none">• Search "Client-server architecture overview MDN"• Search "JavaScript Basics MDN"
Tasks	<ul style="list-style-type: none">• Draw a simple client-server architecture diagram.• Write a brief explanation (200 words) on how HTTPS improves security over HTTP.• Create a program to calculate the area of a circle using variables.• Write a program to take user input for name & age, then display a greeting message.

Day 2	
Topics	<ul style="list-style-type: none">• JavaScript: Arrays, Objects, Hash Maps and Sets• JavaScript Functions and Types
Learn	<ul style="list-style-type: none">• Search "JavaScript array methods tutorial"• FreeCodeCamp JavaScript Course
Tasks	<ul style="list-style-type: none">• Write a program to store student names & scores, then calculate the average score.• Create a function that converts temperatures between Celsius and Fahrenheit.• Write a function that returns only unique elements from an array.

Day 3	
Topics	Object and Array Manipulation, Closure
Learn	Search "JavaScript Closures Explained"
Tasks	<ul style="list-style-type: none">• Write a program using a closure to track button clicks.• Create a function that counts the occurrences of each character in a string.

Day 4	
Topics	Promises, Callbacks and Async Functions
Learn	Search "Async JavaScript Tutorial"
Tasks	<ul style="list-style-type: none">• Create a function that fetches data from a public API using fetch and handles it with promises.• Write an async function that simulates a delayed response and handles it using await.

Day 5	
Topics	<ul style="list-style-type: none">• ECMAScript Evolution: ES6 to ES13• ES6+ Features: let, const, async/await
Learn	<ul style="list-style-type: none">• Search "Modern JavaScript Features ES6+ Overview"• Look for "JavaScript ES6 Tutorial"
Tasks	<ul style="list-style-type: none">• Refactor the temperature conversion program to use ES6+ features like arrow functions.• Write a summary of useful ES8+ features for backend development.• Use <code>async/await</code> and <code>Object.entries()</code> to iterate over data and display key-value pairs.

Day 6	
Topics	Review and Practice Day
Learn	Review all previous materials
Tasks	<p>Student Records Manager (Mini Project) Build a JavaScript-based system for managing student records.</p> <p>Task Breakdown</p> <ul style="list-style-type: none">• JavaScript Functions: Write functions to add new students, calculate the average score and find the highest and lowest scores among students.• Object & Arrays Manipulation: Use objects to store student records and arrays to manage the list of scores.• Closures: Create a closure that keeps track of the total number of students added.• Async Functions: Use <code>async/await</code> to simulate fetching student data from a local JSON file.• ES6+ Features: Refactor functions using arrow functions, template literals, <code>let/const</code>, and the spread operator for managing student records.• Project Output: Display the list of students and their average scores in a formatted way using <code>console.log()</code>.• Documentation: Write a brief summary of the project, the purpose of each function, and any challenges faced.

Week 2: TypeScript Basics

Day 1	
Topics	Introduction to TypeScript
Learn	"TypeScript Handbook" by Microsoft
Tasks	<ul style="list-style-type: none">Setup a basic TypeScript project using TSC (TypeScript Compiler).Understand type annotations and compile a simple TypeScript file into JavaScript.
Day 2	
Topics	The Annotations and Basic Types
Learn	"TypeScript for Beginners" tutorial
Tasks	<ul style="list-style-type: none">Convert JavaScript functions to TypeScript with type annotations for variables, parameters, and return types.Create a simple calculator function using type safety.
Day 3	
Topics	Interfaces and Classes
Learn	Search "TypeScript Interfaces and Classes"
Tasks	<ul style="list-style-type: none">Define interfaces for a student object with properties like name, age, and scores.Create a class for managing student records and implement methods for adding and retrieving student data.
Day 4	
Topics	Advanced Types: Enums, Union Types, Generics
Learn	Search "TypeScript Advanced" Tutorial
Tasks	<ul style="list-style-type: none">Use enums to categorize students by grade level (e.g., freshman, sophomore).Implement a generic function to filter students by various criteria (e.g., passing students).
Day 5	
Topics	TypeScript with ES6+ Features
Learn	"Modern JavaScript with TypeScript" Guide
Tasks	<ul style="list-style-type: none">Refactor functions to use arrow functions, destructuring, and template literals in TypeScript.Use <code>let</code>, <code>const</code>, and the spread operator in a TypeScript context.
Day 6	
Topics	TypeScript Practice
Learn	Review and Internal Exercises
Tasks	Build a simple TypeScript-based script that manages student data, uses classes, and implements a basic CRUD (Create, Read, Update, Delete) functionality.

Week 3: NestJS Fundamentals

Day 1	
Topics	Introduction to NestJS
Learn	"NestJS Official Documentation"
Tasks	<ul style="list-style-type: none">Set up a basic NestJS project using the Nest CLI.Explore the structure of a NestJS application, including modules, controllers, and services.Create a "Hello World" API with a simple controller.
Day 2	
Topics	NestJS Modules and Controllers
Learn	"NestJS Modules and Controllers Tutorial"
Tasks	<ul style="list-style-type: none">Create a new module for managing student data.Implement a controller with basic routes (GET, POST) for handling student data.Use <code>@Controller</code> and <code>@Get</code>, <code>@Post</code> decorators.

Day 3	
Topics	NestJS Services and Dependency Injection
Learn	"Understanding NestJS Services and DI"
Tasks	<ul style="list-style-type: none"> • Create a service to handle business logic for student records. • Inject the service into the controller using NestJS's dependency injection system. • Understand the <code>@Injectable</code> decorator.

Day 4	
Topics	RESTful API with NestJS
Learn	"Building REST APIs with NestJS"
Tasks	<ul style="list-style-type: none"> • Expand the controller to include all CRUD operations (GET, POST, PUT, DELETE) for managing student records. • Implement routes for adding, updating, and deleting students.

Day 5	
Topics	Input Validation with DTOs
Learn	"NestJS DTOs and Validation Pipe"
Tasks	<ul style="list-style-type: none"> • Use Data Transfer Objects (DTOs) with <code>class-validator</code> to validate incoming data. • Create DTOs for adding and updating student data. • Implement validation and test error handling.

Day 6	
Topics	Error Handling and Custom Exceptions
Learn	"NestJS Exception Filters Tutorial"
Tasks	<ul style="list-style-type: none"> • Implement basic error handling using NestJS's built-in exception filters. • Create a custom exception filter for handling validation errors gracefully.

Week 4: Building RESTful APIs with NestJS

Day 1	
Topics	Middleware in NestJS
Learn	"NestJS Middleware Tutorial"
Tasks	<ul style="list-style-type: none"> • Create a custom middleware to log request details like method, URL, and timestamp. • Apply the middleware globally and to specific routes. • Understand the difference between middleware and guards.

Day 2	
Topics	NestJS Guards and Interceptors
Learn	"Understanding Guards and Interceptors in NestJS"
Tasks	<ul style="list-style-type: none"> • Create a guard that restricts access to specific routes based on user roles. • Use interceptors to transform outgoing responses. • Implement a logging interceptor for request duration.

Day 3	
Topics	Building a RESTful API Service
Learn	"Building REST APIs with NestJS"
Tasks	<ul style="list-style-type: none"> • Develop a RESTful API for a library management system with CRUD routes for books. • Use services to handle business logic. • Implement pagination for listing books.

Day 4	
Topics	Using Pipes for Data Transformation
Learn	"NestJS Pipes and Data Transformation"
Tasks	<ul style="list-style-type: none"> Create custom pipes to transform and validate input data. Use built-in validation pipes to sanitize data before saving it to the database. Apply pipes to specific DTOs and routes.

Day 5	
Topics	Error Handling and Custom Responses
Learn	"NestJS Custom Exception Handling"
Tasks	<ul style="list-style-type: none"> Create a custom exception filter for handling specific errors (e.g., <code>NotFoundException</code>). Customize HTTP responses using NestJS's <code>HttpException</code> class. Implement a global exception filter.

Day 6	
Topics	Testing RESTful APIs
Learn	"Testing NestJS Applications"
Tasks	<ul style="list-style-type: none"> Write unit tests for controllers and services using Jest. Mock services to isolate tests for different parts of the API. Test API endpoints using Postman or similar tools.

Week 5: Introduction MongoDB

Day 1	
Topics	<ul style="list-style-type: none"> SQL vs NoSQL Databases Introduction to MongoDB
Learn	<ul style="list-style-type: none"> Search "Difference between SQL and NoSQL Databases" "MongoDB Official Documentation"
Tasks	<ul style="list-style-type: none"> Write a comparison between SQL and NoSQL databases. Set up MongoDB locally or using MongoDB Atlas.

Day 2	
Topics	MongoDB Ecosystem and Working with Collections and Documents
Learn	Search "MongoDB Collections and Documents Guide"
Tasks	Create a <code>users</code> Collection and perform CRUD operations (Create, Read, Update, Delete) using <code>mongo shell</code> or MongoDB compass

Day 3	
Topics	<ul style="list-style-type: none"> Mongoose Basics Schemas and Models
Learn	Search "Getting Started with Mongoose"
Tasks	Define a <code>user</code> schema with Mongoose and create models for managing user data

Day 4	
Topics	Querying and Aggregation
Learn	"MongoDB Aggregation Framework Guide"
Tasks	Write queries to filter, sort and aggregate data in the <code>users</code> collection.

Day 5	
Topics	Indexing and Performance Optimization
Learn	Search " MongoDB Indexing for Performance"
Tasks	<ul style="list-style-type: none"> Create indexes on fields to optimize query performance. Measure query performance before and after indexing.

Week 6: NestJS with MongoDB Integration

Day 1	
Topics	Introduction to MongoDB with NestJS
Learn	"NestJS MongoDB Integration Tutorial"
Tasks	<ul style="list-style-type: none">Set up a MongoDB database using MongoDB Atlas and connect it to a NestJS application using Mongoose.Understand Mongoose schemas, models, and how they map to MongoDB collections.Create a <code>Book</code> schema with properties like <code>title</code>, <code>author</code>, and <code>publishedDate</code>.

Day 2	
Topics	Defining Schemas and Relationships
Learn	"Mongoose Relationships and NestJS"
Tasks	<ul style="list-style-type: none">Define relationships between schemas (e.g., <code>Book</code> and <code>Author</code>).Implement <code>ref</code> in schemas and use <code>populate</code> to retrieve related data.Create a service to manage book and author data with methods for adding, retrieving, and updating records.

Day 3	
Topics	CRUD Operations with Mongoose
Learn	"CRUD with Mongoose in NestJS"
Tasks	<ul style="list-style-type: none">Implement CRUD operations for books and authors using Mongoose methods like <code>find</code>, <code>save</code>, <code>update</code>, <code>delete</code>.Write services that interact with MongoDB for managing data.Create API endpoints to handle these operations in the controllers.

Day 4	
Topics	MongoDB Indexing and Aggregation
Learn	"MongoDB Indexing and Aggregation"
Tasks	<ul style="list-style-type: none">Create indexes for fields like <code>title</code> and <code>author</code> to optimize search queries.Use aggregation pipelines to analyze data (e.g., total books by genre).Implement a search functionality using MongoDB's text indexes.

Day 5	
Topics	Advanced Querying and Transactions
Learn	"Advanced Mongoose Queries with NestJS"
Tasks	<ul style="list-style-type: none">Use Mongoose's <code>aggregate</code> function to perform complex queries (e.g., find all books by a specific author within a date range).Implement transactions using MongoDB's <code>session</code> for atomic operations across multiple collections.

Day 6	
Topics	Testing Database Interactions
Learn	"Testing Mongoose in NestJS"
Tasks	<ul style="list-style-type: none">Write tests for services that interact with the MongoDB database.Mock MongoDB interactions using an in-memory database like <code>mongo-memory-server</code> for testing.Test CRUD operations and validation logic using Jest.

Week 7: Third-Party API Integrations with NestJS

Day 1	
Topics	<ul style="list-style-type: none">Introduction to Third-Party APIsOverview of Twilio, Stripe, and other popular APIs
Learn	<ul style="list-style-type: none">Search "What are Third-Party APIs?"Twilio and Stripe Official Documentation"Using External Libraries in NestJS"
Tasks	<ul style="list-style-type: none">Write a summary of the benefits and challenges of using third-party APIs with NestJS.Explore the Twilio and Stripe documentation and note key concepts like API keys and authentication.

Day 2	
Topics	Integrating Twilio for SMS & Voice in NestJS
Learn	"Twilio Node.js Quickstart Guide" Search "Sending SMS with Twilio in NestJS" "Creating NestJS Services"
Tasks	<ul style="list-style-type: none"> Set up a Twilio account and create a NestJS service that sends SMS messages. Extend the service to include functionality for making voice calls using Twilio's Voice API.

Day 3	
Topics	<ul style="list-style-type: none"> Stripe Payments with NestJS Handling Webhooks
Learn	<ul style="list-style-type: none"> "Stripe Integration with Node.js and NestJS" Search "Handling Webhooks in NestJS"
Tasks	<ul style="list-style-type: none"> Create a NestJS module for handling Stripe payments and integrate a simple payment flow. Set up a NestJS controller to handle Stripe webhooks for payment success and failure events.

Day 4	
Topics	Error Handling & Rate Limiting with NestJS
Learn	<ul style="list-style-type: none"> Search "NestJS Middleware for Error Handling" "Rate Limiting with NestJS and Express"
Tasks	<ul style="list-style-type: none"> Implement error handling for Twilio and Stripe services using NestJS filters. Add a rate-limiting middleware to prevent exceeding API request limits.

Day 5	
Topics	Project: Combining APIs in NestJS
Learn	Review "Building Multi-Service Integrations with NestJS"
Tasks	<ul style="list-style-type: none"> Build a NestJS application that uses Twilio for sending notifications and Stripe for processing payments. Create a scenario where a user makes a purchase and receives an SMS confirmation using the combined services.

Final Project Proposal and Implementation

Project Brainstorming and Proposal
Internal guidance and mentor support
<ul style="list-style-type: none"> Brainstorm project ideas that integrate concepts learned throughout the internship (e.g., a library management system, real-time chat app). Write a proposal that outlines the project scope, goals, and technical requirements. Discuss the proposal with a mentor for feedback.