



Back-End Web Development

Building the future of education with technology

3 Months (2 Live Sessions Weekly)

Instructor

David Sampson

School of Engineering

Email

instructor@elitepaths.com

Module #3

Course Name: BWD

Live Sessions

- Saturday 3-6PM
- Sunday 5-7PM

Course overview

Who Am I?

A meticulously crafted 12-week journey designed to turn learners from novices into job-ready developers. Experience hands-on, practical learning for real-world application that covers essential web technologies, modern back-end frameworks, and key languages like Javascript and Node.js. As a back-end developer, you are like the electrician of a website, ensuring the essential connections behind the scenes for proper functionality and contributing vitally to the success of digital platforms.

Methods

- Lectures, demonstrations, and readings
- Assignments in the live class and assignments to be done as homework outside of class
- Group and individual presentations & critiques
- Attendance and class participation will be major contributing factors in determining final grades. Late projects receive a letter grade reduction.

Why Back-End Web Development?

The relevance of the programme is highlighted by its alignment with the growing industry demand for skilled back-end developers, addressing the rising complexity of businesses and applications. With an industry-standard capstone project, learners receive expert mentorship from experienced software engineers and programme alumni, fostering a supportive learning environment and providing valuable industry insights.

Our Uniqueness

The programme emphasises practical application through industry-standard project-based learning, enabling learners to apply theoretical knowledge directly to real-world scenarios for a deep understanding of concepts.

Learners join a vibrant community of like-minded learners and professionals, fostering connections that support their learning journey and open doors to career opportunities.

Upon completion, learners emerge not just as proficient front-end developers but as versatile professionals ready to seize a spectrum of opportunities, amplifying the overall value of their investment in the programme.

Why elitePath?

Project-based learning approach - This approach involves providing participants with hands-on experience by working on real world projects.

Participants can work in teams to design and build web applications, develop marketing campaigns or create multimedia content, while receiving guidance and support from expert instructors

You will become a well-rounded professional by sharpening your strong points, developing growth areas, and connecting your career path with a purposeful mission.

Objectives

To teach students how to build scalable and secure server-side applications and APIs. Basic proficiency in back-end languages like Node.js and TypeScript.js. Database management with SQL (e.g., MySQL) and NoSQL (e.g., MongoDB) databases. REST API, Authentication, and other back-end concepts. How to build a basic, functional web application from the ground up. To problem-solve based on real-world case studies.

Key Areas

Students will become proficient in backend technologies, including server-side scripting, database design and management, and API development. They will learn to build robust backend systems that handle data storage, processing, and retrieval efficiently, ensuring the reliability and performance of web applications.

Your Responsibilities

- Maintain good communication with instructor regularly throughout the course
- Demonstrate your understanding of principles learned through coursework
- Be an active participant and team member in class, in lab and in discussions
- Accept constructive criticism and show a concerted effort to respond to the feedback
- Execute your work to a professional level of craftsmanship
- Manage your time well, both in and out of class
- Respect yourself, your work, instructors, your classmates, the discipline, and company.

Grades & attendance

Grades

50% project work / 30% Weekly assignments / 20% attendance/participation

Grades will be posted on our Google Classroom

A

80-100 Excellent 4.0

B

70 - 79 Above Average 3.0

C

60 - 69 Average / Satisfactory 2.0

D

50 - 59 Minimum Passing Grade 1.0

F

00 - 49 Fail 0

Grading

Monthly Scorecard:

Monthly assessments based on assignments, participation, and project outcomes, providing feedback on performance and areas for improvement.

Attendance

It is the responsibility of all students to attend their scheduled classes regularly and punctually in order to promote their progress and to maintain conditions conducive to effective learning.

Absences, for whatever reason, do not relieve students of their responsibility for fulfilling normal requirements in any course. In particular, it is the student's responsibility to make individual arrangements in advance of missing class due to personal obligations such as religious holidays or work obligations, etc., in order that he or she may meet his or her obligations without penalty for missing class.

Work Submissions

The instructor may request to retain digital and/or hard copies of student work for display, school promotion, program accreditation and academic dissemination.

This may include gallery exhibitions, promotional brochures, websites, Open House presentations, conference presentations, program accreditation and academic publications. Students retain full ownership of the work (unless a different agreement has been discussed as part of the course) and proper authorship credit will be given to students whose work is highlighted. Please talk to your community manager if you have any questions or concerns about this policy.

Week 1

Month 1: Introduction to Web Development and JavaScript

Fundamentals of Web Development

- Daily Tasks:**
- Explore the basics of web development, including HTML, CSS, and JavaScript.
 - Learn about client-server architecture and the role of backend development.

- Weekly Assignments:**
- Create simple web pages using HTML and CSS for structure and styling.
 - Practice JavaScript coding exercises to understand variables, data types, and control flow.

- Live Sessions:**
- Session 1: Introduction to Web Development (Saturday 3-6pm)
 - Session 2: Getting Started with JavaScript (Sunday 5-7pm)

Week 2

JavaScript Programming

Daily Tasks:

- Dive deeper into JavaScript programming concepts, including functions, arrays, and objects.
- Learn about asynchronous programming and event handling in JavaScript.

- Weekly Assignments:**
- Write JavaScript code to solve programming challenges and exercises.
 - Develop interactive web applications using JavaScript for DOM manipulation and user interaction.

Live Sessions:

- Session 3: Advanced JavaScript Concepts (Saturday 3-6pm)
- Session 4: Asynchronous Programming in JavaScript (Sunday 5-7pm)

Week 3

Introduction to Node.js

Daily Tasks:

- Understand the basics of Node.js and its role as a server-side runtime environment.
- Learn how to set up and configure a Node.js development environment.

- Weekly Assignments:**
- Create simple Node.js applications to handle HTTP requests and responses.
 - Explore Node.js modules and npm packages for extending functionality.

Live Sessions:

- Session 5: Getting Started with Node.js (Saturday 3-6pm)
- Session 6: Building Web Servers with Node.js (Sunday 5-7pm)

Week 4

Building APIs with TypeScript.js

Daily Tasks:

- Explore the Express.js framework for building RESTful APIs and web applications.
- Learn about routing, middleware, and error handling in TypeScript.js.

- Weekly Assignments:**
- Design and implement RESTful APIs using TypeScript.js for CRUD operations.
 - Test and debug TypeScript.js applications to ensure functionality and reliability.

Live Sessions:

- Session 7: Introduction to TypeScript.js (Saturday 3-6pm)
- Session 8: Building APIs with TypeScript.js (Sunday 5-7pm)

Week 5

Month 2: Advanced Backend Development

Working with Databases

Daily Tasks:

- Understand the basics of databases and their role in web development.
- Learn about relational databases (SQL) and NoSQL databases (MongoDB).

- Weekly Assignments:**
- Set up and configure a MongoDB database for a Node.js application.
 - Perform CRUD operations (Create, Read, Update, Delete) using MongoDB and Mongoose.

Live Sessions:

- Session 9: Introduction to Databases (Saturday 3-6pm)
- Session 10: Working with MongoDB and Mongoose (Sunday 5-7pm)

Week 6

Authentication and Authorization

Daily Tasks:

- Explore authentication and authorization mechanisms for securing web applications.
- Learn about JSON Web Tokens (JWT) and session-based authentication.

- Weekly Assignments:**
- Implement user authentication and authorization using JWT in a Node.js application.
 - Secure routes and resources based on user roles and permissions.

Live Sessions:

- Session 11: User Authentication Strategies (Saturday 3-6pm)
- Session 12: Implementing Authorization in Node.js (Sunday 5-7pm)

Week 7

Error Handling and Logging

Daily Tasks:

- Understand the importance of error handling and logging in web development.
- Learn about error handling techniques and best practices in Node.js applications.

Weekly Assignments:

- Implement error handling middleware and logging functionality in a Node.js application.
- Test error scenarios and analyze logs to troubleshoot issues and improve reliability.

Live Sessions:

- Session 13: Error Handling in Node.js (Saturday 3-6pm)
- Session 14: Logging and Debugging Techniques (Sunday 5-7pm)

Week 8

Testing and Deployment

Daily Tasks:

- Explore testing methodologies and tools for Node.js applications.
- Learn about deployment strategies and best practices for deploying Node.js applications.

Weekly Assignments:

- Write unit tests and integration tests for a Node.js application using testing frameworks like Mocha and Chai.
- Deploy a Node.js application to a hosting platform such as Heroku or AWS.

Live Sessions:

- Session 15: Introduction to Testing in Node.js (Saturday 3-6pm)
- Session 16: Deploying Node.js Applications (Sunday 5-7pm)

Week 9

Month 3: Advanced Topics and Capstone Project

Advanced Node.js Concepts

Daily Tasks:

- Dive deeper into advanced Node.js topics such as streams, child processes, and performance optimization.
- Learn how to leverage Node.js for real-time communication and serverless computing.

Weekly Assignments:

- Explore advanced Node.js modules and features for specific use cases and applications.
- Optimize and scale a Node.js application for performance and scalability.

Live Sessions:

- Session 17: Advanced Node.js Concepts (Saturday 3-6pm)
- Session 18: Real-Time Communication with Node.js (Sunday 5-7pm)

Week 10

Building Microservices

Daily Tasks:

- Understand the concept of microservices architecture and its advantages in web development.
- Learn how to design, develop, and deploy microservices using Node.js.

Weekly Assignments:

- Design and implement microservices for a distributed application using Node.js and Express.js.
- Integrate and orchestrate microservices to work together seamlessly.

Live Sessions:

- Session 19: Introduction to Microservices Architecture (Saturday 3-6pm)
- Session 20: Building and Deploying Microservices with Node.js (Sunday 5-7pm)

Week 11

Scalability and Performance Tuning

Daily Tasks:

- Explore strategies for scaling Node.js applications to handle increased load and traffic.
- Learn techniques for performance tuning and optimization in Node.js applications.

Weekly Assignments:

- Perform load testing and benchmarking on a Node.js application to identify performance bottlenecks.
- Implement optimizations such as caching, compression, and clustering to improve scalability and efficiency.

Live Sessions:

- Session 21: Scaling Node.js Applications (Saturday 3-6pm)
- Session 22: Performance Tuning Strategies (Sunday 5-7pm)

Week 12