



CODING CLUB MIET

Curriculum for Web Development

Curriculum Overview :

This curriculum will take the learners through the concepts of web development right from the basics to intermediate and then advanced. Later “Creative Projects” will be assigned that can be completed in small groups or individually. Working in small groups is sometimes required by state standards. These projects can be scaled up or down to meet available class times.

The course curriculum is divided into two parts:

Front-End development

Back-End development

Front-End development:

- Introduction to Front-End Engineer Career Path

Welcome to the Front-End Engineer Career Path!

- Overview of Web Development

Be introduced to the field of front-end web development.

- Fundamentals of HTML

Learn to build a structure for a website using HTML.

- Fundamentals of CSS

Learn how to apply styles to HTML documents using CSS.

- **Developing Websites Locally**

Get acquainted with text editors and Chrome DevTools to develop websites locally.

- **Deploying Websites**

Learn how to deploy websites using GitHub Pages and the command line.

- **Improved Styling with CSS**

Explore intermediate topics in CSS to customize styles and implement navigation elements.

- **Fundamentals of Web Design**

Explore the best user interface (UI) design practices to implement CSS.

- **Making a Website Responsive**

Learn how to utilize responsive web design practices using CSS grid, flexbox, and media queries.

- **JavaScript Syntax, Part I**

Use JavaScript to create basic programs that can store and manipulate various types of data.

- **JavaScript Syntax, Part II**

Explore JavaScript to create complex programs using arrays, loops, objects, and iterators.

- **Building Interactive Websites**

Implement JavaScript to add interactive experiences to a website.

- **Making A Website Accessible**

Get acquainted with the best accessibility (a11y) practices to ensure that your websites are usable to everyone.

- **CSS Transitions and Animation**

Learn how to create visually dynamic websites using CSS transitions and animations.

- **Git and GitHub, Part I**

Explore Git and GitHub to version control your programs.

- **Portfolio Project: Personal Portfolio Website**

Bring together what you have learned in the previous lessons and build a project off of Codecademy.

- **JavaScript Syntax, Part III**

Level up your JavaScript by learning how to implement classes, modules, and error handling.

- **JavaScript Testing**

Using Mocha and Chai, learn various JavaScript testing methodologies, such as Test-Driven Development (TDD).

- **Async JavaScript and HTTP Requests**

Learn about APIs (Application Programming Interfaces). Working with APIs will enable you to work with data stored on remote servers.

- **Web Apps**

Be introduced to web applications and learn about single-page applications (SPAs) and how they are different from static websites.

Back-End development :

- **Welcome to the Back-End Engineer Path**

The first steps to your new career as a back-end software engineer.

- **Setting Up a Dev Environment**

Learn about software engineering and the tools developers use to create websites.

- **Web Development Fundamentals**

Learn about web development and create your first web page on the internet.

- **JavaScript Syntax, Part I**

Use JavaScript to create programs that can store and manipulate data.

- **JavaScript Syntax, Part II**

Create more complex programs using arrays, loops, objects, and iterators.

- **Building Interactive Websites with JavaScript**

Implement JavaScript to add interactive experiences to a website.

- **JavaScript Syntax, Part III**

Learn intermediate JavaScript by implementing classes, modules, and error handling.

- **Git and Github, Part I**

Explore Git and GitHub to version control your programs.

- **Portfolio Project: Mixed Messages**

Build a Node.js console app that generates random messages each time a user runs the program and version of your project with Git/GitHub.

- **Basics of Back-End Development**

Learn about back-end development and programming servers.

- **Async JavaScript and HTTP Requests**

Learn about APIs (Application Programming Interfaces). Working with APIs will enable you to work with data stored on remote servers.

- **Build a Back-End with Express.js**

Learn how to create back-end servers and APIs in JavaScript using the popular Express.js framework.

- **Git and Github, Part II**

Learn how to use Git and GitHub to collaborate efficiently with developers.

- **Portfolio Project: Budget I**

Use Node and Express.js to build an API that allows users to create and manage a personal budget.

- **Database Basics**

Get acquainted with databases and how they help store, retrieve, and manipulate data.

- **Working with Databases**

Craft more sophisticated SQL queries and calculations to build data-intensive applications.

- **Designing Relational Databases**

Learn how to design relational databases that you can then implement in PostgreSQL.

- **Advanced PostgreSQL**

Learn about database performance and techniques for efficiently accessing data and maintaining optimal performance.

- **Connecting a Database to a Server**

Learn how to integrate a PostgreSQL database into your applications.

- **Deploying a Server**

Learn how to build an API with Node, Express, and PostgreSQL and deploy it to Heroku.

- **JavaScript Testing**

Understand the fundamentals of Test Driven Development (TDD) and how you can apply its principles when developing back-ends.

- **Software Design Principles**

Start designing more complex systems with techniques such as design patterns, SOLID principles, and UML.

- **API Development with Swagger and OpenAPI**

Learn how to develop APIs using Swagger tooling and the OpenAPI specification.

- **Portfolio Project: Budget II**

Take your budget app one step further and add a new layer to the project by setting up a database.

- **Web Security Fundamentals**

Learn intermediate concepts in web security including prevention techniques for common threats.

- **User Authentication and Authorization**

Learn how to protect your resources with authorization and authentication techniques.

- **Data Security**

Learn to secure your data using Transport Layer Security, Role-Based Access Control, and more!

- **Common Attacks on Web Applications**

Explore common threats that web applications face and how to mitigate them.

- **Fundamentals of Operating Systems**

Learn about operating systems by deep diving into each of their main functionalities.

- **Caching and CDNs**

Learn about how content is stored on different servers with caching and CDNs.

- Scalability

Learn different strategies for scaling a software system.

- DevOps Fundamentals

Learn the fundamentals of DevOps, a culture supported by practices and tools.