

Node Js კურსის სილაბუსი

Foundations

1. The Internet and Web Technologies Introduction

- What is the Internet and how does it work
- What is WWW, servers and clients
- Transport Control Protocol
- Application protocols
- FrontEnd VS Backend Development

2. HTML/CSS Basics

- Introduction To HTML
- Introduction To CSS
- HTML Documentation Standards, tags
- CSS: Colors and Typography
- CSS: Layout and Positioning
- Responsive Design and Accessibility

3. Javascript Fundamentals

- Hello, world!
- Code structure
- The modern mode, "use strict"
- Variables
- Data types
- Type Conversions
- Basic operators, Maths
- JS Random
- Comparisons
- Conditional operators: if, '?'
- Logical operators
- Loops: while and for
- The "switch" statement
- Functions
- Function expressions
- Arrow functions, the basics
- JavaScript specials
- Debugging in Chrome

- Coding Style
- Comments
- Arrays
- Array methods
- Date and time
- JS Math
- Function setInterval() and setTimeout()

4. Javascript In Browser

- What is DOM?
- DOM Elements
- DOM Methods
- DOM HTML
- DOM CSS
- DOM Animations
- DOM HTML Elements Events
- DOM Event Listener
- DOM Navigation
- DOM HTML Elements Collections

5. Advanced Javascript

- What is OOP?
- Objects
- Object methods
- The “this” keyword
- Accessing Object methods
- Object copying, references
- JS classes
- What is a class
- Static Methods
- Inheritance
- Extends and super() keywords
- Getters and Setters
- What is JSON
- Use cases for JSON
- JSON methods
- Error handling try ... catch
- Custom errors, extending error
- HTML rendering
- What is an API

- Request types(Get, post, put, delete)
- What is fetch API request
- How to make fetch API request

6. NodeJs Basics

- Globals
- Nodemon

6.1 Introduction to NodeJs

- Understanding the role of Node.js in web development
- benefits of using JavaScript on the server

6.3 NodeJs Modules

- Local modules
- Third Party Modules

Core Modules

- **File System (fs)**: Reading and writing files, working with directories.
- **HTTP**: Creating a basic HTTP server, handling requests and responses.
- **Path**: Working with file and directory paths
- **OS (Operating System)**: Provides information about the operating system.
- **Events**: Implementing and handling custom events
- **Stream**: Reading and writing data with streams

6.4 Asynchronous Programming

- Understanding the **Event Loop** and **Non-blocking I/O Model**
- Working with **Callbacks** to Handle **Asynchronous** Operations
- Using **Promises** for Better Error Handling and **Asynchronous** Flow Control
- Utilizing `async/await` to Write Clean and Readable **Asynchronous** Code
- Working with the **EventEmitter**

6.5 HTTP

- HTTP Request/Response Cycle
- HTTP Messages
- HTTP - Basics
- HTTP - Headers
- HTTP - Request Object
- HTTP - HTML File

6.6 Building a Simple HTTP Web Server

- Creating an **HTTP** server using the **HTTP** module.

- Handling different types of **HTTP** requests (**GET**, **POST**, etc.).
- Serving static files and understanding **routing**.

6.7 Node Package Manager (NPM)

- What is NPM
- Installing Packages Locally
- Installing package globally
- Adding dependency in package Json

7. Express Basics

- What is express ?
- JSON - Basics
- Params
- Query String
- Middlewares
- App.use
- Multiple middleware functions
- Express router

Methods

- GET
- POST
- PUT
- DELETE

8. Working with Databases

- What is a database ?
- Relational VS Document databases
- Connecting Express.js to database
- Performing CRUD operations with databases
- Using Object-Relational Mapping (ORM) libraries

9. Working on RESTful APIs

- Designing RESTful APIs
- Handling HTTP methods (GET, POST, PUT, DELETE)
- Validating and sanitizing user input
- Error Handling and Responses
- Testing and Debugging

10. Authentication and Authorization

- What is authentication and authorization theory
- Implementing **JWT TOKEN** based authentication
- Securing routes with middleware

11. Deployment and Hosting

- Preparing the application for deployment
- Deploying Express.js apps to cloud platforms
- Performance optimization and best practices

12. Typescript

- What is typescript?
- Upsides and downsides of typescript
- What are Typescript features?