**Buổi 1: Giới thiệu về Node.js và JavaScript cơ bản I**

Giới thiệu về Node.js

Cài đặt môi trường Node.js

Giới thiệu về JavaScript (JS) cơ bản

Kiểu dữ liệu trong JS

Toán tử trong JS

Sử dụng biến và hằng số trong JS

Sử dụng console log để debug

Cú pháp điều kiện if - else trong JS

Sử dụng vòng lặp for và while trong JS

Tạo và sử dụng function cơ bản trong JS

**Buổi 2: JavaScript cơ bản II**

Phân biệt var, let, và const

Làm việc với Array trong JS

Sử dụng Array methods như map, filter, reduce

Giới thiệu về Object trong JS

Truy cập và thay đổi thuộc tính trong Object

Sử dụng vòng lặp for...in và for...of

Giới thiệu về JS Object methods

Tạo và sử dụng Class trong JS

Sử dụng module http để tạo server đơn giản

Sử dụng NPM (Node Package Manager)

**Buổi 3: JavaScript nâng cao I**

Giới thiệu về Callback, Promise, và Async/Await

Sử dụng callback trong JS

Sử dụng promise trong JS

Sử dụng async/await trong JS

Xử lý lỗi trong promise với try/catch

Xử lý bất đồng bộ trong JS

Giới thiệu về Event Loop

Sử dụng setTimeout và setInterval

Thao tác với file trong Node.js bằng module fs

Sử dụng module path và os

**Buổi 4: Bắt đầu với Express.js I**

Giới thiệu về Express.js

Cài đặt và cấu trúc dự án Express

Tạo và cấu hình router trong Express

Xử lý HTTP Request & Response

Giới thiệu và sử dụng Middleware cơ bản trong Express

Sử dụng Middleware để xử lý lỗi

Sử dụng Middleware cho authentication

Sử dụng Middleware cho logging

Quản lý Session và Cookie

Xử lý File Upload trong Express

**Buổi 5: Express.js II và Giới thiệu về MySQL**

Sử dụng Template Engine trong Express (EJS, Pug)

Render dữ liệu động vào template

Sử dụng layout và partial trong EJS

Sử dụng form để gửi dữ liệu POST

Xử lý dữ liệu form trên server bằng middleware body-parser

Giới thiệu về MySQL

Cài đặt MySQL và phần mềm quản lý MySQL (MySQL Workbench, phpMyAdmin)

Tạo cơ sở dữ liệu và bảng trong MySQL

Thao tác cơ bản với MySQL: INSERT, SELECT, UPDATE, DELETE

Làm việc với Node.js và MySQL bằng thư viện mysql của Node.js

**Buổi 6: Làm việc với MySQL trong Node.js I**

Giới thiệu về MySQL

Cài đặt MySQL và MySQL Workbench

Sử dụng module mysql để kết nối với MySQL từ Node.js

Thực hiện truy vấn cơ bản trong MySQL (SELECT, INSERT)

Tạo, đọc, cập nhật, và xoá dữ liệu từ MySQL (CRUD)

Quản lý transaction trong MySQL

Tạo và sử dụng Stored Procedure

Tạo và sử dụng Function

Làm việc với INDEX để tăng tốc độ truy vấn

Thực hành: Tích hợp MySQL vào dự án Express

**Buổi 7: Làm việc với MySQL trong Node.js II và Sequelize**

Quản lý database và table trong MySQL

Tạo và sử dụng JOIN trong MySQL

Tạo và sử dụng Subquery và UNION trong MySQL

Tạo và sử dụng Trigger trong MySQL

Giới thiệu về ORM (Object-Relational Mapping) và Sequelize

Cài đặt và cấu hình Sequelize

Tạo model và migration trong Sequelize

Tạo, đọc, cập nhật, và xoá dữ liệu với Sequelize

Tạo và sử dụng association trong Sequelize

Tối ưu hóa câu query với Sequelize

**Buổi 8-9: Làm việc với MongoDB trong Node.js**

Giới thiệu về NoSQL và MongoDB

Cài đặt và sử dụng MongoDB

Sử dụng module mongodb để kết nối với MongoDB

Tạo, đọc, cập nhật, và xoá dữ liệu từ MongoDB (CRUD)

Quản lý transaction trong MongoDB

Làm việc với Indexed, Aggregation, và Replication trong MongoDB

Sử dụng thư viện Mongoose để làm việc với MongoDB

Tích hợp MongoDB vào dự án Express

Thiết kế schema và quan hệ dữ liệu trong MongoDB

Sử dụng GridFS để lưu trữ file trong MongoDB

**Buổi 10-12: Xây dựng và tối ưu hóa API với Express, MySQL, và MongoDB**

Lập kế hoạch và thiết kế API

Tạo và xác thực API với Express

Tích hợp MySQL và MongoDB vào API

Quản lý lỗi API

Sử dụng pagination, sorting, và filtering cho API

Sử dụng caching để tăng tốc API

Bảo mật API

Test API với Postman

Sử dụng load testing để kiểm tra hiệu năng API

Deploy API lên server

**Buổi 13-15: Authentication và Authorization trong Node.js**

Giới thiệu về Authentication và Authorization

Sử dụng sessions để xác thực người dùng

Sử dụng cookies để xác thực người dùng

Sử dụng JWT (JSON Web Token) để xác thực người dùng

Tạo và xác minh JWT trong Express

Sử dụng Passport.js để xác thực người dùng

Tích hợp Google và Facebook login với Passport.js

Sử dụng OAuth 2.0 với Passport.js

Xác thực và phân quyền trong API

Thực hành: Tích hợp JWT vào dự án

**Buổi 5: Làm việc với File và Tạo PDF trên Server (real-time)**

Tìm hiểu về FileSystem module trong Node.js

Đọc và ghi dữ liệu vào file

Sử dụng Stream để đọc và ghi file lớn

Xử lý file đính kèm từ client

Tạo file PDF động trên server

Sử dụng library để tạo và chỉnh sửa PDF

Gửi file từ server về client

Xử lý các lỗi liên quan đến file

Tìm hiểu về MIME type

Tạo module riêng trong Node.js

**Xử lý file upload trong Express.js**

Sử dụng middleware để xử lý file upload

Lưu trữ file upload lên server

Hiển thị file đã upload lên trang web

Xử lý file download trong Express.js

Thiết lập header để cho phép download file

Bảo mật file trên server

Tạo link tải file động

Xử lý lỗi khi upload và download file

Tối ưu hoá thời gian upload và download

**Làm việc với Sessions và Cookies**

Hiểu về sessions và cookies

Sử dụng cookies trong Express.js

Tạo và sử dụng sessions trong Express.js

Lưu trữ sessions vào cơ sở dữ liệu

Quản lý sessions và cookies cho các yêu cầu khác nhau

Sử dụng sessions và cookies để xác thực người dùng

Bảo mật cho sessions và cookies

Sử dụng JWT (Json Web Tokens)

Thực hiện quên mật khẩu và đặt lại mật khẩu qua email

Xử lý các vấn đề liên quan đến sessions và cookies

Buổi 11: Xác thực và Phân quyền Người dùng

**Hiểu về xác thực người dùng**

Tạo chức năng đăng ký, đăng nhập, đăng xuất

Mã hóa mật khẩu người dùng

Sử dụng sessions để quản lý trạng thái đăng nhập

Hiểu về phân quyền người dùng

Thực hiện phân quyền cho người dùng trong ứng dụng

Xác thực người dùng với các yêu cầu khác nhau

Xử lý lỗi khi xác thực và phân quyền

Xác thực thông qua các dịch vụ khác (Facebook, Google)

Bảo mật thông tin người dùng

Hiểu về xác thực trong REST APIs

Tạo token cho người dùng

Xác thực token trong các request

Sử dụng JWT (Json Web Tokens) cho xác

Xác thực token trong các request

Sử dụng JWT (Json Web Tokens) cho xác thực

Refresh token và cách làm việc của nó

Xử lý lỗi liên quan đến xác thực

Bảo mật token

**Tải file trong REST APIs**

Hiểu về tải file trong REST APIs

Sử dụng middleware để xử lý file upload

Lưu trữ file upload và thay đổi thông tin trong cơ sở dữ liệu

Download file từ REST APIs

Xử lý lỗi khi tải file

Bảo mật cho file

Tải nhiều file cùng một lúc

Xử lý file upload với cơ sở dữ liệu NoSQL

Tối ưu hoá quá trình tải file

Sử dụng Cloud Storage (ví dụ: AWS S3) để lưu trữ file

**Buổi 22-24: GraphQL trong Node.js**

Giới thiệu về GraphQL

So sánh REST và GraphQL

Cài đặt và cấu hình GraphQL

Tạo schema và resolver cho GraphQL

Sử dụng query và mutation trong GraphQL

Sử dụng variables và inputs trong GraphQL

Làm việc với nested data trong GraphQL

Sử dụng subscription trong GraphQL

Tích hợp GraphQL vào Express

Thực hành: Tích hợp GraphQL vào dự án

**Gửi Email**

Hiểu về gửi email trong ứng dụng web

Cấu hình để gửi email từ Node.js

Tạo nội dung email động

Gửi email đơn và hàng loạt

Xử lý lỗi khi gửi email

Sử dụng template cho email

Gửi email với các tệp đính kèm

Xác thực email người dùng

Gửi email thông báo cho người dùng

Theo dõi tình trạng gửi email

**Xử lý Thanh toán với Stripe.js**

Giới thiệu về Stripe và cách sử dụng Stripe

Cài đặt và cấu hình Stripe.js

Tạo form thanh toán với Stripe

Xử lý thanh toán với Stripe

Xác thực thông tin thanh toán

Xử lý lỗi thanh toán

Lưu trữ thông tin thanh toán

Xác nhận thanh toán qua email

Bảo mật thông tin thanh toán

Thực hiện thanh toán định kỳ

**Xây dựng ứng dụng Realtime với Node.js và Websockets**

Hiểu về ứng dụng Realtime và Websockets

Cài đặt và cấu hình Websockets cho Node.js

Gửi và nhận dữ liệu thông qua Websockets

Xây dựng ứng dụng chat đơn giản với Websockets

Xử lý lỗi khi làm việc với Websockets

Sử dụng Websockets trong ứng dụng đa người dùng

Bảo mật cho Websockets

Tối ưu hóa hiệu năng khi làm việc với Websockets

Sử dụng Websockets với REST APIs và GraphQL APIs

Deploy ứng dụng sử dụng Websockets

Sử dụng TypeScript với Node.js

Giới thiệu về TypeScript và lợi ích khi sử dụng TypeScript

Cài đặt và cấu hình TypeScript cho Node.js

Sử dụng TypeScript cho các chức năng cơ bản trong Node.js

TypeScript với Express.js

TypeScript với MongoDB và Mongoose

Xử lý lỗi và Debug khi sử dụng TypeScript

TypeScript với REST APIs và GraphQL APIs

TypeScript với Websockets

Sử dụng các tính năng nâng cao của TypeScript

Triển khai ứng dụng sử dụng TypeScript

**Khám phá Deno.js**

Giới thiệu về Deno.js

Cài đặt và cấu hình Deno.js

Xây dựng ứng dụng đơn giản với Deno.js

Sử dụng mô-đun bên ngoài trong Deno.js

Xử lý file và cơ sở dữ liệu trong Deno.js

Xây dựng REST APIs và GraphQL APIs với Deno.js

Xử lý lỗi và Debug trong Deno.js

Sử dụng TypeScript trong Deno.js

Tối ưu hóa hiệu năng khi làm việc với Deno.js

Triển khai ứng dụng Deno.js

Buổi 27: Dự án thực hành - Xây dựng ứng dụng mua sắm trực tuyến

**Phân tích và thiết kế ứng dụng mua sắm trực tuyến**

Xây dựng cơ sở dữ liệu cho ứng dụng

Xây dựng chức năng đăng ký, đăng nhập và quản lý tài khoản người dùng

Xây dựng chức năng hiển thị, tìm kiếm và lọc sản phẩm

Xây dựng giỏ hàng và chức năng mua hàng

Xây dựng chức năng thanh toán

Xây dựng chức năng quản lý đơn hàng cho người dùng

Xây dựng REST APIs cho ứng dụng

Kiểm thử và triển khai ứng dụng

Thảo luận và cải thiện ứng dụng

**Dự án thực hành - Xây dựng ứng dụng Blog**

Phân tích và thiết kế ứng dụng Blog

Xây dựng cơ sở dữ liệu cho ứng dụng

Xây dựng chức năng đăng ký, đăng nhập và quản lý tài khoản người dùng

Xây dựng chức năng tạo, hiển thị, chỉnh

Xây dựng chức năng tạo, hiển thị, chỉnh sửa, và xoá bài viết

Xây dựng chức năng bình luận và thích bài viết

Xây dựng chức năng tìm kiếm và lọc bài viết

Xây dựng REST APIs cho ứng dụng

Xây dựng ứng dụng realtime với Websockets cho chức năng bình luận

Kiểm thử và triển khai ứng dụng

Thảo luận và cải thiện ứng dụng

**Buổi 25-27: Deploy và Maintain Node.js App**

Giới thiệu về các cách deploy Node.js app

Deploy app lên Heroku

Deploy app lên AWS

Deploy app lên Google Cloud

Sử dụng PM2 để quản lý app

Sử dụng Nginx như reverse proxy

Giới thiệu về CI/CD (Continuous Integration/Continuous Deployment)

Tạo pipeline CI/CD với Jenkins/GitHub Actions

Quản lý phiên bản code với Git

Cấu hình và sử dụng Docker để đóng gói ứng dụng

Monitoring ứng dụng Node.js với các công cụ như New Relic, Datadog

**Buổi 28-30: Advanced Node.js và Thực hành dự án cuối khóa**

Giới thiệu về Worker Threads trong Node.js

Sử dụng Worker Threads để tăng hiệu năng ứng dụng

Tạo và sử dụng Event Emitter trong Node.js

Tìm hiểu về Streams trong Node.js

Sử dụng Buffer và Binary Data trong Node.js

Giới thiệu về Clustering và Child Process

Thực hành Clustering để tăng hiệu năng ứng dụng

Dự án cuối khóa: Xây dựng ứng dụng web hoàn chỉnh với Express, MySQL/MongoDB, GraphQL, và Socket.io

Review dự án và nhận xét

Kết thúc khóa học và hướng dẫn tiếp theo.

Với nội dung này, bạn sẽ có được một cái nhìn toàn diện về Node.js từ cơ bản đến nâng cao, bao gồm việc làm việc với Express, MySQL, MongoDB, GraphQL và các chủ đề khác. Mong rằng nó sẽ giúp bạn đạt được mục tiêu học tập của mình.