Lesson Plan: Level 6 (Intermediate Web Development)

Table of Content

- Table of Content
  - Real Life Application
  - Tools
  - Content Overview
  - Chapter 1: Introduction to Node
    - Concepts to Cover
    - Lessons to Cover
    - Chapter Challenge
  - Chapter 2: Advanced Node
    - Concepts to Cover
    - Lessons to Cover
    - Chapter Challenge
  - Chapter 3: API Development using ExpressJS, NestJS and TypeORM
    - Concepts to Cover
    - Lessons to Cover
    - Chapter Challenge

Real Life Application

- Web Sites
- Web Applications
- Web APIs
- Agile Development Approach
- https://dev.to/devbookmark/400-free-resources-to-learn-to-code-in-2021-mern-design-mean-fullstack-2e1e

Tools

- Visual Studio Code

- Google Chrome
- Node.js
- Postman
- Heroku
- NestJS CLI
- Node.js for Professionals

Content Overview

- What is Node
- What is the Node Package Manager
- Create NPM project
- What are Promises?
- What is Agile Development
- What is Reactive Programming?
- What is an API?
- How do we access an API?
- What is Swagger?
- What is Typescript?

Chapter 1: Introduction to Node

Concepts to Cover

- Introduction to node
- Setting up the environment for node development
- Introduction to Agile Development
- CRUD functionality
- Process

- Modules
- Reactive Programming
- Promises

Lessons to Cover

Introduction to node.js

Introduction to Agile Development

Setting Up node.js

Node Through The Eye Of The Environment

Promises

Reactive Programming; core concepts of the reactive approach to application development

Chapter Challenge

Create a Math package, that will collect all the fizz buzz numbers based on the range

- The functions must all be observables
- I must be able to import the package, and use the functions

Chapter 2: Advanced Node

Concepts to Cover

- HTTP programming, via the AXIOS framework
- Swagger Interpretation
- Typescript; Adding types to the javascript application
- API development using Express.js

Lessons to Cover

The AXIOS Framework

Typescript

Chapter Challenge

Create an application that will consume the information, based on the user location. As soon as the application loads, the app should obtain the day, and the location, from the browser, and retrieve the weather information, and display it in HTML

Chapter 3: API Development using ExpressJS, NestJS and TypeORM

Concepts to Cover

- Express.js
- TypeORM
- NestJS
- Deploy to Heroku

Lessons to Cover

Express JS

NestJS

Databases

TypeORM on NestJS

Swagger

Heroku

Chapter Challenge

Create an API for your own choosing. the API must include:

- A swagger document
- Must store your information in a database
- Must be deployed on Heroku, and the source code must be stored on Github