Syllabus for the course of

FULL STACK DEVELOPMENT - (MERN Stack)

(with 1 month extensive hands on Internship)

About this course

This course will serve as a comprehensive introduction to various topics in Software Development. This course is a proper blend of theory and the practical hands on session for each and every concept. Throughout the course participants will work on a complete end to end tech stack in order to implement the concepts learned during the course.

Program Highlights

- 120+ Hours of Live Lectures
- 60+ Hours of Live Handson Sessions
- 100+ Hours of Assignment and Projects
- 3+ Mini Projects
- 2+ Major Projects
- 1 Capstone Project

Technology Stack & Tools Used

Learning Outcomes

At the end of the course participants should be able to -

- Build an extensive backend API with Java SpringBoot.
- Protecting routes/endpoints with JWT (JSON Web Tokens)
- Extensive API testing with Postman.
- Integrating React with our backend in an elegant way, creating a great workflow.
- Building our frontend using React to work with backend API's
- Creating the build scripts and securing the keys before deployment
- Get handson experience on Git, GitHub, GitHub Pages & Heroku

Prerequisites

There are no prerequisites required for the course. But a little knowledge of programming and HTML, CSS and JavaScript is a benefit for the candidate.

Schedule

Topic	Total hours
HTML, CSS and Introduction to Object Oriented JavaScript – ES6, Classes, JS tips and tricks	30
Introduction to React Library – CDN, script and npm packages, Components, functional and class based	40
NodeJS + Express - Designing backend, handling routes, handling errors and integrating No-SQL MongoDB for storing data	40
Databases – Creating a DB using MongoDB & learning various CRUD operations and filters.	15

Front End Development and Version Control

1. **Introduction**
    a. Download and install Visual Studio Code (VS Code)
    b. Overview of Full Stack Development Program

2. **HTML**
    a. Inline elements
    b. Block elements
    c. Element Attributes
    d. HTML Forms and Form Elements
    e. Type of Inputs
    f. Buttons
    g. Anchor tags
    h. Images
    i. Video
    j. Meta tags
    k. Miscellaneous Tags (code, blink, table, span, etc.)

3. **CSS**
    a. Cascading
    b. CSS Selectors
    c. CSS Attributes
    d. Cascading Algorithms
    e. Padding
    f. Margin
    g. Floating
    h. Inline and inline-block
    i. Border
    j. Box Model
    k. Bootstrap

4. **JavaScript**
    a. JS Programming Fundamentals (ES6)
    b. DOM Access and DOM Manipulations
    c. DOM Events
    d. JavaScript Classes & OOPs
    e. JavaScript Functions & JavaScript Arrow Functions
    f. Higher Order Array Methods - forEach, map, filter and reduce
    g. JavaScript await, Callbacks & Promises
    h. JavaScript tips and tricks

5. **AJAX**
   a. Introduction to Client-Server Architecture
   b. Response Notations - XML & JSON
   c. Introduction to APIs
   d. Testing APIs with Postman
   e. Introduction to asynchronous communication
   f. Inserting APIs with Front End

6. **React**
   a. Creating React App
   b. Folder Structure
   c. Understanding React Components
   d. Adding components
   e. Listening to events
   f. Understanding JSX
   g. Creating functional components
   h. Working with props & states
   i. State Management using Redux
   j. Getting user inputs
   k. Handling Events
   l. Adding forms
   m. Routing in React
   n. Third Party APIs

7. **Version Control**
   a. Introduction to Version Control
   b. Introduction to Git and GitHub
   c. Why Git
   d. Centralized and Distributed Version Control System
   e. Installing Git
   f. Git Basics
   g. Forking
   h. Cloning
   i. Making changes to Local Repositories
   j. Committing
   k. Branching
   l. Collaborating with Multiple Developers

8. **Project on Front End Development**

Back End Development, Database and Deployment

1. **Introduction to Back-End Development**
   a. Introduction to Client-Server Architecture
   b. Understanding the need and complications
   c. Understanding the use and need for RESTful APIs

2. **NodeJS**
   a. Introduction to NodeJS, a JavaScript runtime environment
   b. Installing NodeJS
   c. Terminal / Command Prompt Basics
   d. The in-built basics modules of NodeJS

3. **Express - (i)**
   a. Introduction to Express, a NodeJS Framework
   b. Creating servers using Express
   c. Express & NodeJS as a JavaScript compiler
   d. Understanding the requests, responses and request headers
   e. Creating routes using Express (GET, POST, PUT, DELETE)
   f. Installing Nodemon package (the NodeJS package)
   g. Responding to request with JSON application type
   h. Installing the BodyParser package (the NodeJS package)

4. **MongoDB**
   a. Introducing to MongoDB as a database
   b. Difference between SQL and NoSQL databases
   c. Installing MongoDB into local system using MongoDB Atlas
   d. Creating MongoDB clusters on MongoDB servers
   e. Creating database in MongoDB
   f. Understanding documents in MongoDB
   g. Inserting documents into database
   h. Searching documents using queries
   i. Deleting documents from the database
   j. Updating documents on the basis of certain constraints

5. **Express - (ii)**
   a. Exploring challenges in the existing setup
   b. Installing Mongoose package (the MongoDB package)
   c. Connecting database to the back-end application
   d. Using mongoose for storing data in the database
   e. Using mongoose for retrieving data from the database
   f. Using mongoose for updating the database

      g.   Using mongoose for deleting data from the database

6. **Combining the Front-End & BackEnd**
   a. Understanding the utility of a Full-Stack application
   b. Integrating the NodeJS + Express application to a SPA
   c. Exploring various integration methods - Fetch, AJAX & Axios
   d. Understanding the complications and debugging

7. **Deployment**
   a. Deploying the NodeJS + Express back-end to Heroku
   b. Integrating the exposed back-end to the SPA
   c. Integrating the SPA to GitHub Pages a full-stack application