MERN Stack Development

Course Outline:

This course will cover the following topics (each topic can include subtopics):

- HTML, CSS and JavaScript (Prior knowledge required)
- Advance Javascript ES6
- React Project scaffolding
- React functional and class components
- React parent to child and child to parent communications
- React Redux
- React Hooks
- React Context Api
- React Routers
- React Material UI
- React project with fake JSON server
- Nodejs
- Npm package registry understanding with package.json
- Nodejs HTTP server
- Express Introduction
- Express server
- Creation of rest apis
- Validation with YUP or JOI validator
- Usage of Lodash
- Mail configuration with SMTP nodemail
- Job with queues
- Schedule Commands with cron jobs
- File uploads with Multer
- File upload with GridFS
- Rest APIs (Modern Technique for Web applications)
- Graphql Apis (Modern Technique for Web applications)
- Mongodb
- Mongodb CRUD queries, Lookup queries, aggregation pipeline
- Mogodb with reference and embedded relations
- Mongodb indexing

- Mongodb with Mongoose ORM
- Mongodb schema creation with mongoose
- Apis Integration with Frontend React project
- Api calls with Axios and fetch on frontend
- Dotenv on frontend react and backend nodejs express
- React store creation with redux persistent
- React service Layer for Apis
- Completion of React Project with Backend Express JSON apis
- Git and GitHUB
- Lifecycle of git
- Git Useful commands
- Git branches
- Git remote and local repo
- Deployment of Project on Heroku / Deployment of Project on Ubuntu Server

Week 1-2 (Web Pre-requisite and Advance Javascript):

- HTML, CSS and JavaScript Intro
- Advance Javascript ES6
    - Let VS const
    - Arrow function VS Normal function
    - Class inheritance
    - Default and Named imports
    - This keyword scope
    - Asynchronous javascript VS Synchronous javascript
    - Fetch API
    - Promises and Callbacks
    - Callback Hell Problem
    - Async await
    - Exception handling with promises
    - OOP in javascript

- React Project Scaffolding
- Basic Component Creation
- React Class and Functional Components
- React Parent to child and child to parent communication
- React imports

Week 3-4 (Advance React):

- React Hooks
- React Redux and store configuration
- React context API
- React Routers
- React with Material UI
- React Project with Fake JSON server

Week 5 (Nodejs With Express):

- Intro to Nodejs
- Nodejs HTTP server
- Express server
- Creation of GET, POST, PUT PATCH Apis
- Use lodash to manage the collection of objects
- POSTMAN test cases and automation
- POSTMAN environment variables
- POSTMAN collection
- Express middlewares
- Express routing
- Dotenv in the project for environment variables
- REST APIs

Week 6-7 (Mongodb database in Express):

- Into to mongodb
- Mongo compass installation with mongo

- Mongodb CRUD queries
- Mongodb Relations
- Mongodb Pipeline
- Mongodb with mongoose
- Connection of mongodb in express app
- Storing Data in Mongodb
- CRUD of Todos/Users Endpoints and persist data in mongodb using Mongoose ORM.

Week 8-9 (APIs authentication and authorization):

- Creation of a Signup route with mail invitation
- Creation of login route
- Forgot and reset password endpoints
- Creation of other endpoints
- Validation of API request with YUP/JOI validator
- Usage of JWT authentication
- Express auth middleware

Week 10 (File uploads, Scheduling, Jobs, Exception handling):

- File upload with Multer
- File upload with Grid FS
- Public Url for file upload
- Base64 File uploads
- Jobs and queues
- Cronjob with scheduling
- Exception handling and logs creation
- Removal of logs with cronjob
- Webhook for exception in Slack channel

Week 11 (Frontend and Backend Integration):

- Integration of APIs created in nodejs express with frontend

- Creation of REDUX store to manage global state
- Protected routes with AUTH guard
- Frontend User authentication and authorization
- Creation of service layer for JSON based APIs
- Usage of toast for showing unhandled error
- Usage of Axios and fetch apis
- Redux persistent for store state perseverance

Week 12 (Deployment of Project on Heroku):

- Use Git for version control
- Git lifecycle + git branches for development and production
- Adding Production environment variables
- Push code to the remote repo on Github to manage changes and history
- Bring code to the production server either on Heroku or on a fresh Ubuntu Server.
- Finalize and Test the production code.