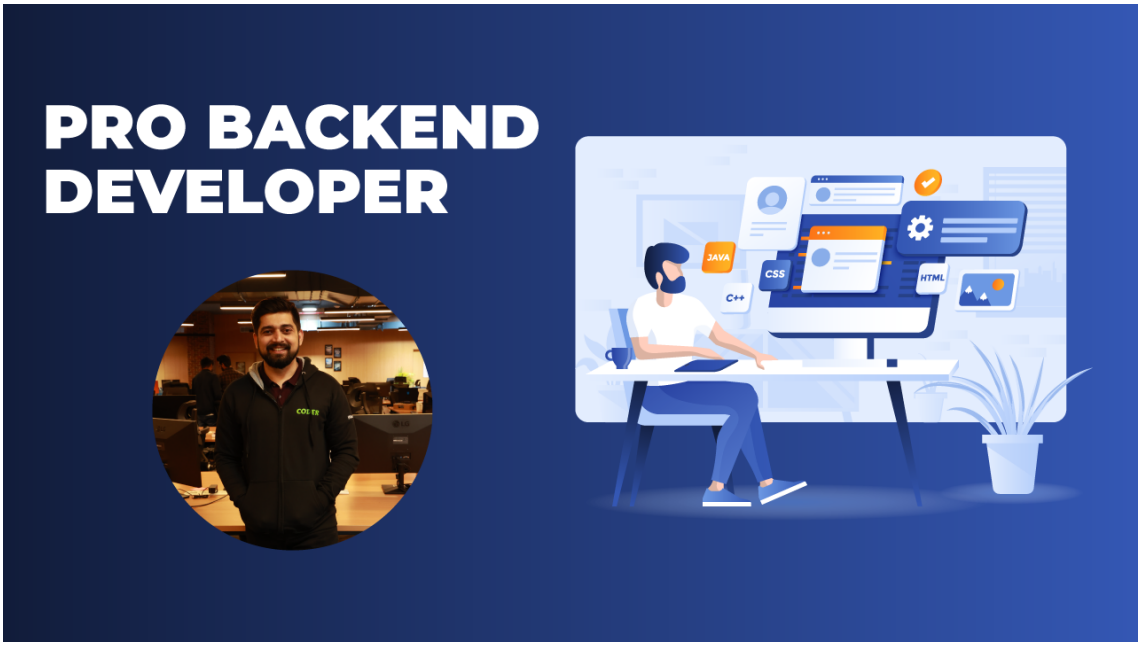
Welcome to ineuron.ai



Pro Backend Developer

Description:

This course is tiled pro for a reason. In this practical hands-on course, you will learn how to build complex backend applications that can be used for any web or mobile application. Your REST API will be in production with docs, social logins, images, authentications, mail and, much more. This is a true pro backend course.

Start Date:

Doubt Clear Time:

Course Time:

Features:

Course material

Course resources

On demand recorded videos

Practical exercises

Quizzes

Assignments

Course completion certificate

What we learn:

MongoDB

Heroku Cloud

Swagger

Authentication

File,image and form handling

MORGAN and razorpay

Configs and imports

Controllers and routes

Requirements:

System with minimum i3 processor or better

At least 4 GB of RAM

Working internet connection

Dedication to learn

Instructor:

Name:

Hitesh Choudhary

Description:

I like to make videos related to code and tech in my free time. I also lead a few tech teams in startups, help in hiring talent for companies. I am also on a part time traveller, with 31 countries

checked off so far!

>Getting started:

>>Goal of this course and instructions

>>Tools for backend developer

>>Mongodb MAC install

>>Mongodb WIN install

>> MongoDB in cloud - Atlas

>> Mongo GUI - compass

>Take it up to Heroku - Production:

>>Things you need to deploy on Heroku

>>Plan your application

>>Types of web request

>>Framework - Express, Koa, Hapi

>>Starting with package JSON file

>>Your first express app

>>Request Response and Status code

>>All social routes

>>Handle the date situation

>>Parameters and bugs in route

>>Pushing app to HEROKU

>>Debug social app in production

>Swagger Docs:

>>What is swagger and api docs

>>Nodemon ext and YAML docs

>>Authentication token for swagger docs

>>Docs for HTTP methods swagger

>>A new documentation centric project

>>Setup information - swagger

>>Authentication and Authorization - swagger

>>String based GET request - swagger

>>handling objects - swagger

>>handling array in Swagger docs

>>Sending data in URL - swagger

>>managing request body in swagger

>>handle url query in swagger

>>handling images in swagger

>>handling header tokens in swagger

>Authentication:

>>What we have done till section 3 - backend

>>Hiding secrets in backend

>>Picking up a database for backend

>>Why we need mongoose - ODM

>>Pro db modeling tools

>>Creating model for auth system

>>Creating basic structure for auth system

>>Creating user schema and dotenv

>>Registering a user in auth system

>>Database connection in auth system

>>What is a middleware

>>Handling password situation

>>What is JWT and creating token

>>Register route in auth app

>>Login flow for auth app

>>Web vs Mobile

>>Writing custom middleware

>>Setting up secure cookies

>File, image and form handling:

>>Why people face issue in image upload

>>Cloudinary and EJS

>>How GET works and postman issues

>>Using template engines

>>Biggest confusion in front end forms

>>Handling images in forms

>>Handling images in forms part 2

>>upload image to cloudinary or other providers

>>Handling multiple files and uploading them

>Theory and Razorpay:

>>File structure for production app

>>Getting a logger - MORGAN

>>Error handler and Promises

>>Sending emails using nodemailer

>>Why mongoose docs are important

>>Razorpay project

>>Razorpay front end integration

>Big Ecommerce app starts:

>>Project requirement

>>User modeling and file structure

>>Product model discussion

>>Order Model discussion

>>How forgot password feature work

>>Functions in user model and hooks

>Basic Config and imports:

>>Getting files and folders ready

>>Preparing basic express app

>>Routes and controllers in dummy

>>Injecting docs and middleware

>>Custom error handlers

>>The big Promise

>User model and signup:

>>Creating a user model and validator

>>password encryption and mongoose prototypes

>>Validating the password

>>creating JWT tokens

>>forgot password and crypto hashing

>>User routes and postman

>>Signup a user and cookies

>>Database connection

>>Testing the user signup with postman

>>Handling image upload

>>Testing photo upload and user signup

>>yes, we know about postman files

>User controllers and routes:

>>Login route and controller

>>logout controller and route

>>Send email from node

>>Forgot password controller

>>Reset password controller and routes

>>Middleware - injecting information

>>User dashboard controller and routes

>>Update the password for a user

>>Updating the user profile

>>User, admin, manager and more roles

>>Manager only routes

>>Admin get a single user

>>Admin can update any user

>>Admin can delete a user now

>Working on Product Model:

>>Product middleware setup for routes

>>Product Model and refs

>>A long talk on URL replace and mongo operators

>>Creating a product

>>Where clause in search

>>Where clause Pager

>>Aggregation filer in Where Clause

>>Get all products with WHERE and pager

>>Debugging and testing of product add and get

>More routes in Products:

>>Single product route

>>Update the product with photos

>>Delete a product and minor bug

>>Testing and debugging

>>Add a review

>>Delete a review and requested routes

>>Configure routes for reviews

>Razorpay and Stripe:

>>Stripe Docs

>>Stripe controllers

>>Razorpay payments and order

>>Setup payment routes

>Processing Orders:

>>Order model in action

>>Creating an order and BSON

>>Testing create order and routes

>>Populate fields in order

>>Order of routes is important

>>Updating the stock

>>Delete order and push to git

>>Pushing code to production server

>OAuth and Social Logins:

>>Social login foundation and demo app

>>Consent screen and API keys

>>Why passport js

>>Package installation

>>Home routes and EJS

>>Preparing routes for login

>>Showing consent screen of google

>>Getting information and email from google

>>Moving google data to database

>>Serialize and deserialize user

>>Protect the Home