

Welcome to ineuron.ai



## MERN Stack Bootcamp

### **Description:**

Learn complete web stack programming with React and Node by doing it the way a full-stack professional would do it. Learn how to create whole web apps from start to finish with one of the most trending tech stacks available. You can learn how to create data-driven applications, as well as how to test, protect, and deploy your code, in this thorough study path. This course necessitates a basic understanding of HTML, CSS, and JavaScript.

### **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:**

# File structure and response codes

# Express Server & Database Modelling

# Preparing Database Models

# Database Connectivity

### **Requirements:**

# System with Internet Connection

# Interest to learn

# Dedication

### **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!

#### **>Introduction to Full Stack:**

- >>Getting started as full stack developer
- >>Installing node and mongo
- >>File structure and response codes
- >>Let me help you to think about a project
- >>Designing the architecture of our project

## **>Express Server & Database**

### **Modelling:**

- >>Getting started with express
- >>Reading express documentation and assignment
- >>Why everyone loves nodemon ?

### **>Preparing Database Models:**

- >>Installing from exercise files
- >>What is mongoose ?
- >>Creating a basic schema for User
- >>Your first virtual
- >>What the UUID ?
- >>Create schema methods
- >>Writing complete User schema

## **>Creating Schema with Mongoose:**

- >>Creating category schema

>>Creating product - Tshirt Schema

>>Order page is complex

## **>Database Connectivity:**

>>MongoDB connection and other DB talk

>>Environment variables in process

## **>Middleware and Initial Routes:**

>>What is a middleware ?

>>Most common middlewares

>>Router in express

>>How to use controllers ?

>>What is RoboMongo ?

>>Use Postman for post routes request

>>Fixing models bug and postman

>>Saving a user to database

>>What about data validation ?

>>Setting up validation in routes

>>Setting up validation results with custom messages

## **>Authentication      Routes      &**

### **Tokens:**

>>What are tokens and how to create them ?

>>Creating signin route

- >>Sending tokens in cookie for user
- >>Learn to do testing with Postman
- >>What is Bearer and signout route ?
- >>How to protect a route ?
- >>How to write custom middleware ?
- >> Where we are so far

## **>Parameter & User Controllers:**

- >> What are PARAMS
- >> Get ID from param
- >> Handling user route
- >> Fixing a major bug
- >> An assignment for USERS
- >> Get all users assignment solution
- >> Learn to update the user info
- >> Testing the PUT route for user
- >> Using populate from other collection
- >> One more middleware to update purchases

## **>Summer & Winter Collection - Category:**

- >> Moving further for collections
- >> Get category from param
- >> Creating category and saving it

- >> Get all categories at once
- >> Update the collection
- >> Perform a delete operation
- >> Testing all routes with postman

## **>Adding Tshirt to our backend:**

- >> JSON vs FORM DATA
- >> Get product by ID
- >> Saving photo in mongo and tshirt assets
- >> Add restriction on product fields
- >> Long debugging at postman
- >> Get compressed image assets here
- >> Some optimization for binary data
- >> Update and delete of Tshirts
- >> Get all products
- >> Update your inventory
- >> This is an alternative to get all distinct categories

## **>Placing the Order:**

- >> Getting started with order
- >> Get Order by ID
- >> create an order
- >> Why ENUMS are important
- >> Update the order status

## **>Moving to Front End:**

>> Bootstrap crash course

>> React Router crash course

## **>Connecting to Frontend to Backend:**

>> Create a react application

>> Prepare the folder structure

>> Remove all errors first

>> How to add CSS in react app

>> Inject backend into frontend env

>> Take full advantage of base component

## **>Navigation & Authentication:**

>> Inject the navigation bar

>> Adding styling to navbar

>> Routes for signin and signup

>> Signup and Signin form

>> Authentication helper for react

>> Signup component talks to backend

>> Testing the signup route

>> Sign in logic

>> Sign in testing part 1

>> Sign in testing part 2

### **>Restricted Routes & Profile:**

>> Conditional rendering for signout

>> Restriction on Routes

>> Admin and user routes in front end

>> Protect the admin dashboard

>> Admin left side

>> Admin Right side

### **>Design an Admin Panel:**

>> Add category routes

>> Making category request to backend

>> finish creating categories

>> CRUD calls for product

>> Assignment and add product front end

>> useEffect and preload data

>> Fixing categories bug

>> Create a product and assignment

>> Delete a product from admin

>> update the Tshirt part 1

>> update the tshirt part 2

>>SMALL HELP IN ASSIGNMENT

### **>Purchase Card & Cart:**



- >> Creating a reusable card
- >> Conditional rendering of card elements
- >> Bring everything to home page
- >> Home page loads all tshirts
- >> Add tshirts to cart
- >> UI for Cart page
- >> Design a force component reload method
- >> Order helpers

## **>Integrating Stripe - Track1 -**

### **Optional:**

- >> Get the final amount for user
- >> Get a stripe in front end
- >> Writing backend for stripe payment
- >> Writing front end for stripe
- >> Add key and payment gateway is all set

## **>Integrating Paypal with card:**

- >> Read the payment gateway docs
- >> Setup backend from docs
- >> Some bug fixes and payment helpers
- >> we are finally processing all payments

## **>Empty card bugs & Optional**

## **Orders:**

>> Bug hunting and placing orders