

# Build and Deploy Full Stack MongoDB, Express, React, Node.js App



## Beginning MERN Stack: Build and Deploy a Full Stack MongoDB, Express, React, Node.js App by Greg Lim

★★★★☆ 4.4 out of 5

Language : English  
File size : 11338 KB  
Text-to-Speech : Enabled  
Screen Reader : Supported  
Enhanced typesetting : Enabled  
Print length : 161 pages  
Lending : Enabled



In this tutorial, we will build and deploy a full stack MongoDB, Express, React, Node.js application. We will use MongoDB as our database, Express.js as our web framework, React.js as our front-end framework, and Node.js as our server-side runtime environment. We will also use Docker to containerize our application and deploy it to a cloud platform.

## Prerequisites

Before you begin, you will need the following:

\* A text editor or IDE \* Node.js and npm installed \* MongoDB installed \* Docker installed

## Creating a New Node.js Project

First, we will create a new Node.js project. Open your terminal or command prompt and run the following command:

```
mkdir my-app cd my-app npm init -y
```

This will create a new directory called `my-app` and initialize a new npm project.

## **Installing the Required Dependencies**

Next, we will install the required dependencies for our project. Run the following command in your terminal:

```
npm install express mongoose react react-dom
```

This will install the following dependencies:

\* express: A web framework for Node.js \* mongoose: A MongoDB object modeling tool for Node.js \* react: A front-end framework for building user interfaces \* react-dom: The DOM manipulation library for React

## **Creating a MongoDB Database**

Now, we will create a MongoDB database. Open your MongoDB shell and run the following command:

```
use my-app
```

This will create a new database called `my-app`.

## **Creating a Mongoose Schema**

Next, we will create a Mongoose schema for our database. A schema defines the structure of our data in MongoDB. Create a new file called `models/user.js` and add the following code:

```
javascript const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({ name: String, email: String,
password: String });

const User = mongoose.model('User', userSchema);

module.exports = User;
```

This schema defines a User model with the following properties:

- \* `name`: A string representing the user's name
- \* `email`: A string representing the user's email address
- \* `password`: A string representing the user's password

## **Creating an Express Server**

Now, we will create an Express server. Create a new file called `server.js` and add the following code:

```
javascript const express = require('express'); const mongoose =
require('mongoose'); const User = require('./models/user');

const app = express();

app.use(express.json());
```

```
app.get('/users', async (req, res) => { const users = await User.find();  
res.json(users); });
```

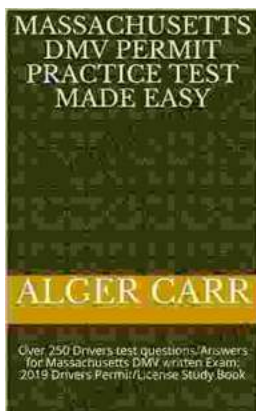
```
app.post('/users
```



## Beginning MERN Stack: Build and Deploy a Full Stack MongoDB, Express, React, Node.js App by Greg Lim

★★★★☆ 4.4 out of 5

Language : English  
File size : 11338 KB  
Text-to-Speech : Enabled  
Screen Reader : Supported  
Enhanced typesetting : Enabled  
Print length : 161 pages  
Lending : Enabled



## Ace Your Massachusetts DMV Written Exam: Over 250 Test Questions and Answers

Are you preparing to take the Massachusetts DMV written exam? If so, you're in luck! This article provides over 250 test questions and answers to help you...



## Off Balance: Dominique Moceanu's Inspiring Memoir

A Heartfelt Account of a Champion's Journey and Advocacy In her gripping memoir, "Off Balance," former Olympic gymnast and vocal advocate...