Mastering MEAN Stack

Build full stack applications using MongoDB, Express.js, Angular, and Node.js

Pinakin Ashok Chaubal

www.bpbonline.com

First published: 2023

Published by BPB Online
WeWork
119 Marylebone Road
London NW1 5PU

UK | UAE | INDIA | SINGAPORE

ISBN 978-93-55510-525

www.bpbonline.com

Dedicated to

My wife ***Manasi***
My sons ***Arnav*** *&* ***Yash***
&
My parents who supported me throughout
the journey of this book

About the Author

Pinakin Chaubal is a PMP, ISTQB Foundation, and HP0-M47 QTP11 certified IT professional with over 23 years in Performance testing with LoadRunner & JMeter. He has worked in the fields of Software Testing & Quality Automation for various projects with demonstrative success in applying test methodologies & QA processes, software defect tracking, and new deployment environments & automation with automated test tools & frameworks across multi-national companies. He has also gained global onsite exposure of 6 years in the USA and 8 months in Hong Kong & China.

About the Reviewer

Mohd Monis is an accomplished full-stack developer currently working as a technical lead, bringing over 8 years of invaluable industry experience. His expertise primarily lies in frontend development, where he excels in analyzing and synthesizing complex business problems.

He actively engages in fruitful conversations about emerging technology trends with his peers. Additionally, he has recently developed a keen interest in creating design mockups.

Acknowledgement

I want to express my deepest gratitude to my family and friends for their unwavering support and encouragement throughout this book's writing, especially my wife Manasi.

I am also grateful to BPB Publications for their guidance and expertise in bringing this book to fruition. It was a long journey of revising this book, with valuable participation and collaboration of reviewers, technical experts, and editors.

I would also like to acknowledge the valuable contributions of my colleagues and co-workers during many years working in the tech industry, who have taught me so much and provided valuable feedback on my work.

Finally, I would like to thank all the readers who have taken an interest in my book and for their support in making it a reality. Your encouragement has been invaluable.

Preface

Building full stack applications is a complex task that requires a comprehensive understanding of the latest technologies and programming languages. MongoDB, Express, Angular and Node.js are powerful tools that have become increasingly popular in the field of enterprise development.

This book is designed to provide a comprehensive guide to building enterprise applications using the MEAN stack. It covers a wide range of topics, including the basics of MongoDB, Angular and advanced concepts such as object-oriented programming, and the use of the Node.JS for building robust and scalable applications.

Throughout the book, you will learn to buld a MEAN stack for a blog site. The reader is taken step by step through the learning journey. The reader gets to know various concepts in application development and deployment.

This book is intended for developers who are new to building full stack applications. It is also helpful for experienced developers who want to expand their knowledge of these technologies and improve their skills in building robust and reliable applications.

With this book, you will gain the knowledge and skills to become a proficient developer in the field of full-stack development using the MEAN stack. I hope you will find this book informative and helpful.

Chapter 1: Fundamentals of Full Stack Development and the MEAN Stack - This chapter will focus on the concept of full stack development, it's importance and how the MEAN stack is a quick way to develop such applications. We will understand why the MEAN stack is so irresistible for developers. We will also see a glimpse of the application that we will be developing. Understand how the MEAN stack helps in fast, efficient and scalable development. We will understand the difference between the traditional multi-threaded web server and the single-threaded Node web server, learn what blocking/non-blocking code is, and understand the Express framework. We move on to MongoDB as the database for our stack and eventually we look at the Angular frontend framework which will be used in designing the face of our application. We will look at what TypeScript is and also look at the use of Bootstrap. We will be introducing Git as our version control system. We will take a look at AWS as our hosting service. We will then take a look at how the different MEAN components interact with each other. We will get introduced to Docker and Kubernetes and understand their role in deploying MEAN applications..

Chapter 2: Architectural Design of Our Sample Application - This chapter will focus on the various design considerations for a MEAN stack. We will explore various options available. We will start with a basic architecture and then move on to the desired Microservices architecture in which we can have separate docker containers for the Posts service, Comments service, the Express and Node.js framework and the database. This way our system will be fault tolerant and resilient.

Chapter 3: Installing the Components - In this chapter, we will be looking at the Node.js and Angular needed for our MEAN project. We will be installing all these on Windows, although the installation remains largely the same for any operating system. We will create the project folder and initialize it. We will install Angular CLI and then create a new Angular project within the parent folder. We will then install Angular material inside the Angular project folder. We will also look at the generated Angular project and package.json file. We will then check the installed dependencies in node_modules folder. Eventually, we will take a look at the generated app folder and the various files within it.

Chapter 4: Creation of the Frontend Using Angular - In this chapter, we will be building our frontend. This will be a bare-bones frontend which will be integrated with MongoDB and the server framework later. We will start with a bird's eye view of how angular works by customizing the current Angular application.

Chapter 5: Addition of Node.js and Ideas for Integration- This chapter will explore the backend server part of our MEAN framework which comprises of Node.js [Open-source, cross-platform backend JavaScript runtime environment that runs on V8 engine built for Chrome] Express JS [Backend web-application framework for Node.js]. We will start with the basics of building backend logic using Express and Node.js. In this process, we will make use of Postman[API platform for building and using APIs.] Postman simplifies the API lifecycle and streamlines collaboration so that better APIs can be created faster. We will also see two main HTTP verbs for building the backend code which is GET and POST.

Chapter 6: Handling Authorization - With this chapter, we start the development of the first piece of our framework which is the Authorization piece. We will be making use of Typescript throughout this book for the development of the backend part of our code. This chapter will incorporate certain best practices for the development of code. We will be using Google Cloud environment for developing the application, since many times due to creation of many Kubernetes pods, older systems may crash. One can only create a single node cluster on a local desktop or laptop. We can create a multi node cluster when we develop on Google Cloud. We will be creating a docker container for our Auth service and will also make use of Google cloud shell as our Integrated Development Environment.

Chapter 7: Creating the Posts Service and NATS Streaming Integration - In this chapter, we start with the introduction to the common module. As a part of the common module, we create a folder called common and move the Authentication related logic to this folder. We will create an NPM module of the common module. This common module can be included in all services of our project. We then move to the development of the Posts service. This is followed by an introduction to the NATS streaming server, and we create a simple Publisher and Listener which have a simple NATS implementation. We will see how to publish messages and listen to messages. The NATS Streaming code will be moved to the common module. We create a Posts service and integrate it with the NATS streaming server. We then test our integration with Postman

Chapter 8: Introducing Automated Testing - Here, we will be looking at how to incorporate automated testing in our framework using Supertest. Supertest is a library which will test all our APIs without the need to bring up a Kubernetes cluster. The Supertest library can be used by itself or with Mocha. In this book, we will focus on Supertest with the JEST framework.

Chapter 9: Integrating the Comments Service - In this chapter, we will look at the concepts behind building and integrating the comments service. Concepts such as sub-documents and references with regard to MongoDB will be introduced. We will see the pros and cons of using sub-documents and learn the advantages of references.

Chapter 10: Creating the Comments Service - In this chapter, we will look at the creation of the comments service and see how we can use Supertest to run automated tests to test our comments service.

Chapter 11: Implementing the Frontend - This chapter will wrap up things by implementing the front end. We will look at how to go about creating the frontend with Angular. We will also learn about the various pieces of an Angular application like components and services. We will see what the app component is and how to plug in the various components in the app component.

Code Bundle and Coloured Images

Please follow the link to download the
Code Bundle and the ***Coloured Images*** of the book:

https://rebrand.ly/tdi84xj

The code bundle for the book is also hosted on GitHub at **https://github.com/bpbpublications/Mastering-MEAN-Stack**. In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

https://discord.bpbonline.com

Table of Contents

CHAPTER 1
Fundamentals of Full Stack Development and the MEAN Stack

Welcome to the exciting world of full stack development. This book will be divided into three parts. The first part will make the user understand what full stack development is and what the MEAN stack is. The second part will explore the installation of the different MEAN stack components, the creation of the front-end using Angular, adding Node.js and connecting with Angular, and the addition of the MongoDB (NOSQL) database. This part will take care of the **Create, Read, and Delete** part of **CRUD**. The third part will cover the Update part of CRUD and routing, image upload, and pagination. The fourth part will cover adding authentication, authorization, error handling, optimization, and deployment. It is going to be a very exciting journey, so let's get started and understand what full stack development all is about. We will then understand the fundamentals of the MEAN stack.

This chapter will focus on the concept of full stack development, its importance and how the MEAN stack is a quick way to develop such applications. We will understand why the MEAN stack is so irresistible for developers. We will also see a glimpse of the application that we will be developing. Understand how the MEAN stack helps in fast, efficient, and scalable development. We will understand the difference between the traditional multi-threaded web server and the single-threaded Node web server, learn what blocking/non-blocking code is, and understand the Express framework. We move on to MongoDB as the database for our stack and eventually, we look at the Angular frontend framework which will be used in designing the face of our application. We will look at what TypeScript is and look at the use of Bootstrap. We will introduce Git as our version control system. We will look at AWS as our hosting service. We will then look at how the different MEAN components interact with each other. We will get introduced to Docker and Kubernetes and understand their role in deploying MEAN applications.

Structure

This chapter will predominantly focus on the following topics:

- What is full stack development?
- Introduction to the MEAN stack
- Our sample application
- Introduction to Node.js
- Introduction to Express.js
- Introduction to MongoDB
- Introduction to Angular
- Introduction to Typescript and Bootstrap
- Introduction to Git as a version control system
- Interaction between components of the MEAN stack
- Introduction to Docker and Kubernetes as deployment tools

Objectives

After reading this chapter, the readers will understand what full stack development all is about. They will also get introduced to frameworks that support full stack development. They will understand what the MEAN stack is, and they will also get a glimpse of the sample application that we will be building throughout this book. They will learn about the underlying components of the MEAN stack and understand how the various components interact together. They will get introduced to Docker and Kubernetes and see how Docker simplified the build and deployment process.

What is full stack development?

Full stack development refers to the development of both frontend (client-side) and backend (server-side) parts of a web application. It is the art of designing complete web applications or websites. It comprises the development of frontend, backend, and database.

Full stack development can consist of the development of whole web applications using components like HTML, CSS, Java/JavaScript/Python, and a database of choice. JavaScript offers two major choices when it comes to designing web applications which are: The MEAN stack and the MERN stack. These stacks comprise frameworks and libraries.

Using full stack development, developers can design complete end-to-end applications. Such applications typically have 3 parts which are frontend, backend, and database. Let's see each of these in a nutshell. We will be expanding on each of these throughout the book.

Frontend

What we call the frontend here is the visible part of the web application or website which is responsible for providing user experience. The user interacts directly with this part of the application.

The frontend languages can be any one of the following:

- **HyperText Markup Language (HTML)**: HTML is used to design web pages using a markup language. HTML is a combination of Hypertext and Markup language. Hypertext is used to establish a link between the web pages. The text within the tag is defined by the Markup language and this defines the structure of a web page.
- **Cascading Style Sheets (CSS)**: CSS is a language that simplifies the process of making web pages attractive. CSS helps to apply styles to web pages and CSS aids to do this independent of the HTML for each page.
- **JavaScript**: JavaScript is a scripting language which is used to make the site interactive for the user.

The frontend frameworks and libraries can be any of the following:

- **Angular**: Angular is an open-source front-end framework, written in JavaScript, mainly used to develop **single-page web applications** (**SPAs**). It converts static HTML to dynamic HTML. It is available to be used and changed by anyone. It extends HTML attributes with directives where data is attached with HTML.
- **ReactJS**: React is a declarative JavaScript library created by Facebook for building user interfaces. ReactJS is an open-source, front-end library responsible for the front- end of the application.
- **Bootstrap**: Bootstrap is a free and open-source collection to create responsive websites and web applications. It is the most popular framework for

developing responsive websites which are very popular nowadays in the mobile world.

Backend

What we refer to as backend here is the server-side development of a web application or website which focuses on the functionality of the website. It is also responsible for managing the database through queries and APIs using client-side commands.

The backend languages can be any one of the following:

- **Java**: Java is one of the most popular and extensively used programming languages and platforms known for its high scalability. Moreover, Java components are easily available.
- **Python**: The Python programming language allows the developer to work quickly and integrate systems efficiently.
- **JavaScript**: JavaScript can be used as a back-end programming language in addition to its being a front-end programming language.

The backend frameworks and libraries can be any of the following:

- **Express**: Express.js, or Express, is a back-end web application framework for Node.js, which is free and open-source software under the MIT License. It is specifically designed for building web applications and APIs. It is one of the popular frameworks for node.js among developers.
- **Django**: Django is a high-level web framework built in Python that encourages rapid development and clean, pragmatic design. It takes care of many of the problems of web development so that the developer can focus on writing her app without starting from scratch. It's free and open-source software.
- **Spring**: The Spring framework is an application framework and inversion of control (IOC) container designed for the Java platform. The core features of the framework can be used by any Java application, but extensions are available for building web applications on top of the Java EE platform.

Database

A database is a collection of data that is interrelated which helps in easy and efficient retrieval, insertion, and deletion of data. Data in a database is organized in the form of tables, views, schemas, reports, and so on.

Few of the popular databases used in full stack development are MongoDB, Oracle, MySQL, etc. Let's take a brief look at these: