

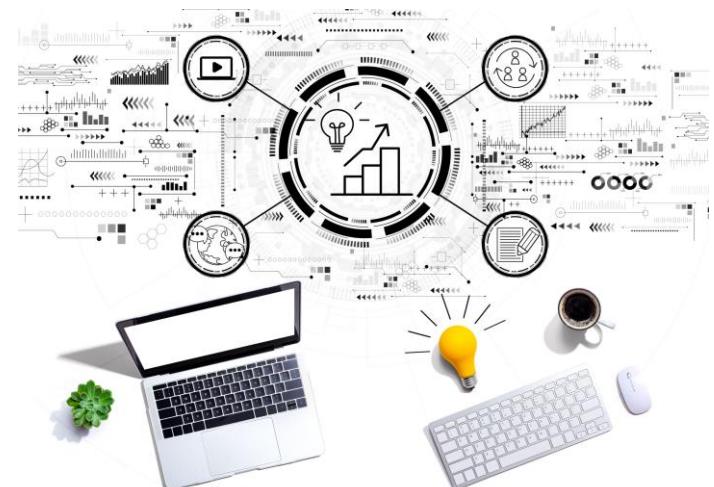
MSWD (MEA(R)N Stack Web Development)

Ajay Solanki,
Computer Science & Engineering



CHAPTER-1

Introduction To Web Development And The MEAN Stack



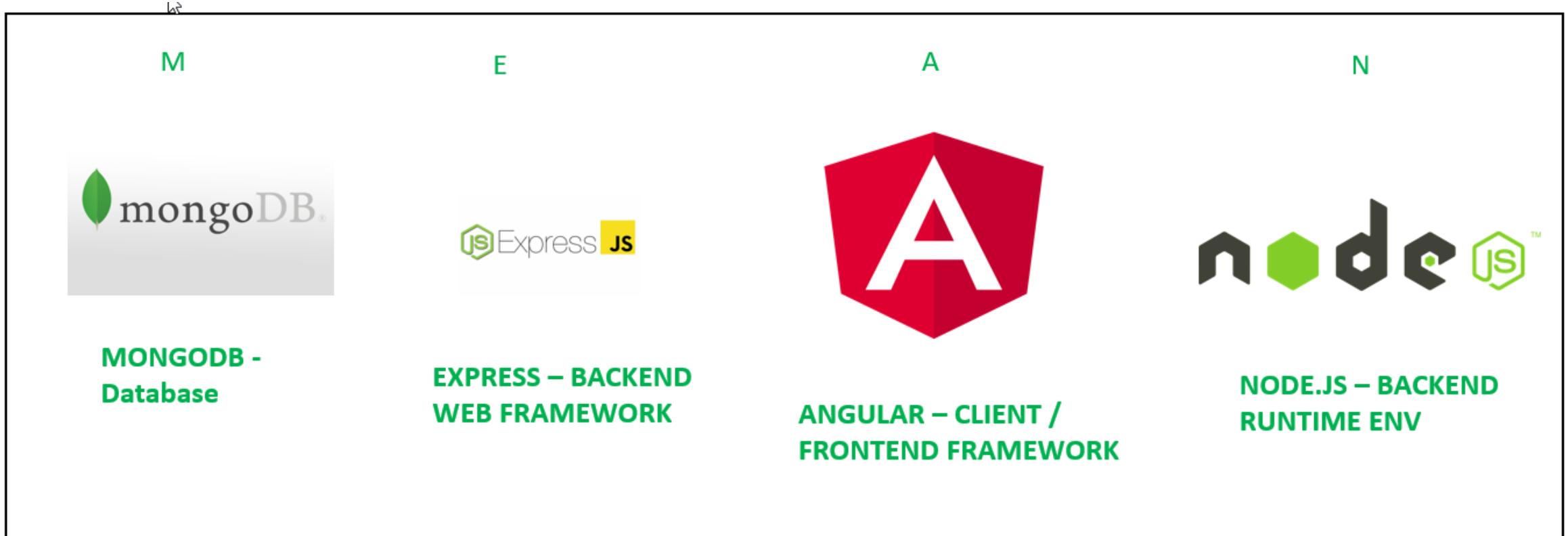
What we learn in this chapter ?

1. What is the MEAN stack?
2. How does the MEAN stack work?
3. MEAN stack components
4. Advantages of the MEAN stack
5. Disadvantages of the MEAN stack
6. When can the MEAN stack be used?
7. How secure is the MEAN stack?
8. Setting Up the Development Environment

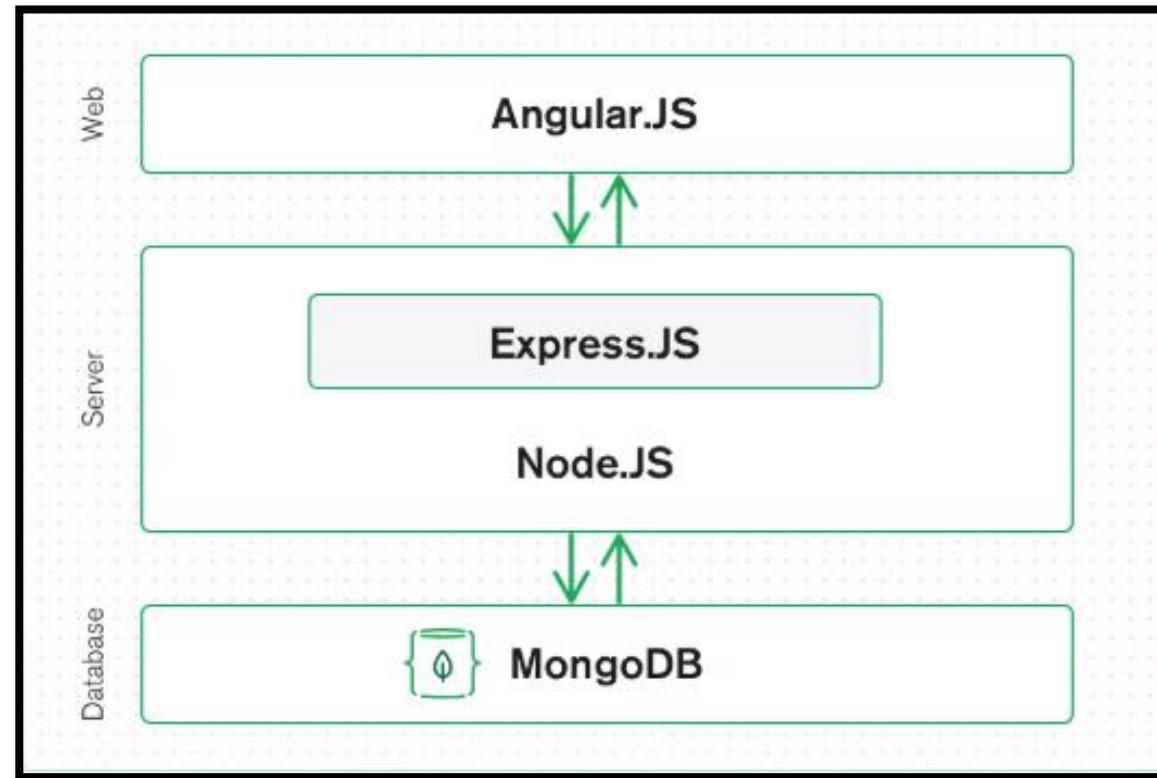
What is the MEAN stack?

- One of the most common examples of these "standardized building blocks" is called the MEAN stack. Discover what the MEAN stack is, how it works, the advantages and disadvantages of the MEAN stack, and how it's commonly used.
- The MEAN stack is a JavaScript-based framework for developing scalable web applications. The term MEAN is an acronym for MongoDB, Express, Angular, and Node — the four key technologies that make up the layers of the technology stack.
 - MongoDB: A NoSQL, object-oriented database designed for use with cloud applications
 - Express(.js): A web application framework for Node(.js) that supports interactions between the front end (e.g., the client side) and the database
 - Angular(.js): Often referred to as the “front end”; a client-side JavaScript framework used to create dynamic web applications to work with interactive user interfaces
 - Node(.js): The premier JavaScript web server used to build scalable network applications

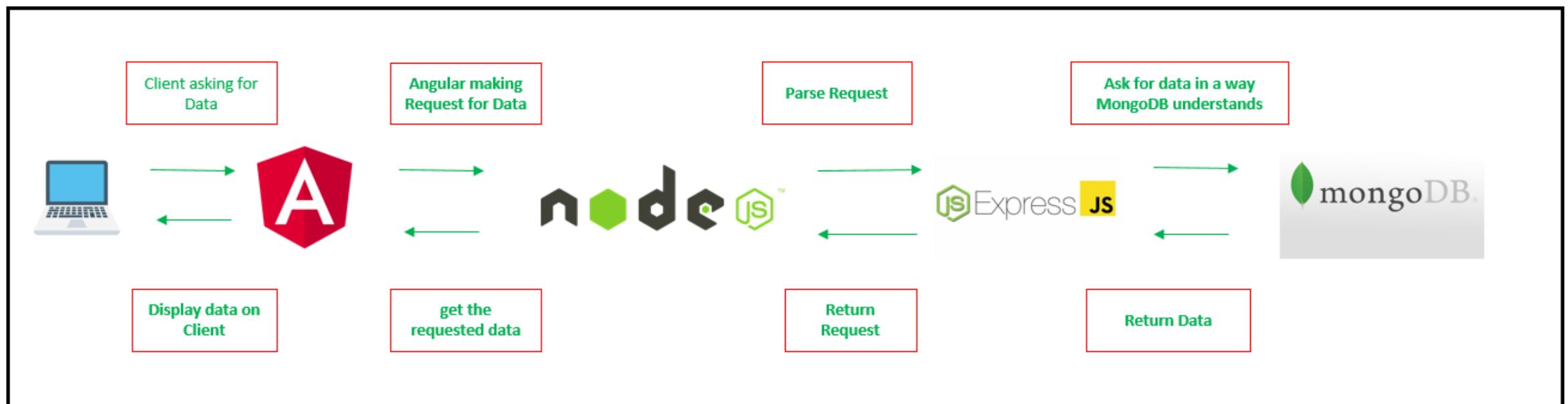
What is the MEAN stack?



How does the MEAN stack work?



How does the MEAN stack work?



1. Angular.js front end

- At the very top of the MEAN stack is Angular.js, the self-styled “JavaScript MVW Framework” (MVW stands for “Model View and Whatever”).
- Angular.js allows you to extend your HTML tags with metadata in order to create dynamic, interactive web experiences much more efficiently than, say, building them yourself with static HTML and JavaScript (or jQuery).
- Angular has all of the bells and whistles you’d expect from a front-end JavaScript framework, including form validation, localization, and communication with your back-end service.

MEAN stack components

1. Express.js and Node.js server tier

- The next level down is [Express.js](#), running on a Node.js server. Express.js calls itself a “fast, unopinionated, minimalist web framework for Node.js,” and that is indeed exactly what it is.
- Express.js has powerful models for URL routing (matching an incoming URL with a server function), and handling HTTP requests and responses. By making XML HTTP requests (XHRs), GETs, or POSTs from your Angular.js front end, you can connect to Express.js functions that power your application.
- Those functions, in turn, use MongoDB’s Node.js drivers, either via callbacks or promises, to access and update data in your MongoDB database.

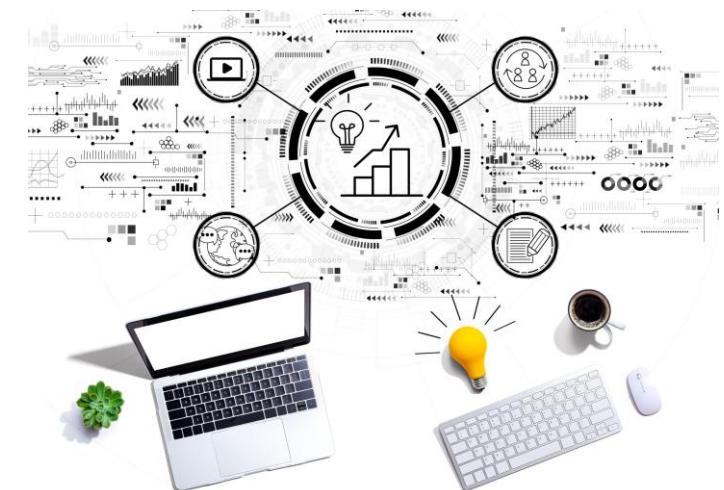
MEAN stack components

1. MongoDB database tier

- If your application stores any data (user profiles, content, comments, uploads, events, etc.), then you're going to want a database that's just as easy to work with as Angular, Express, and Node.
- That's where MongoDB comes in: JSON documents created in your Angular.js front end can be sent to the Express.js server, where they can be processed and (assuming they're valid) stored directly in MongoDB for later retrieval.
- If you want to easily get the best of MongoDB, you'll want to look at [MongoDB Atlas](#). This will give you built-in, full database security and cross-cloud scalability with the click of a button. [More on that later on this page.](#)

Advantages of the MEAN stack

- MEAN applications can be used in many ways with a cross-platform, write-once approach. While MEAN is particularly suited to real-time applications, particularly those running natively in the cloud and single-page (dynamic) web applications built in Angular.js, it can be used for other use cases, such as:
 - Workflow management tools.
 - News aggregation sites.
 - To-do and calendar applications.
 - Interactive forums.



Advantages of the MEAN stack

- There are many more uses for the MEAN stack, as well.
 - Since all the components are based on JavaScript and JSON, the integration between the components of the stack is intuitive and straightforward.
 - Additionally, the E and A of MEAN (Express and Angular) are two of the most popular and well-supported JavaScript frameworks for back-end and front-end development, respectively. Express makes routing and managing HTTP requests and responses super easy and includes great support for middleware to handle JSON endpoints and form posts. Angular is a powerful tool for building dynamic HTML pages that communicate with a back-end server.
 - Whether you're building a high-throughput API, a simple web application, or a microservice, MEAN is the ideal stack for building Node.js applications.
 - All of the MEAN stack components are open source in nature and therefore allow a generous, free-of-charge opportunity for developers.

Disadvantages of the MEAN stack

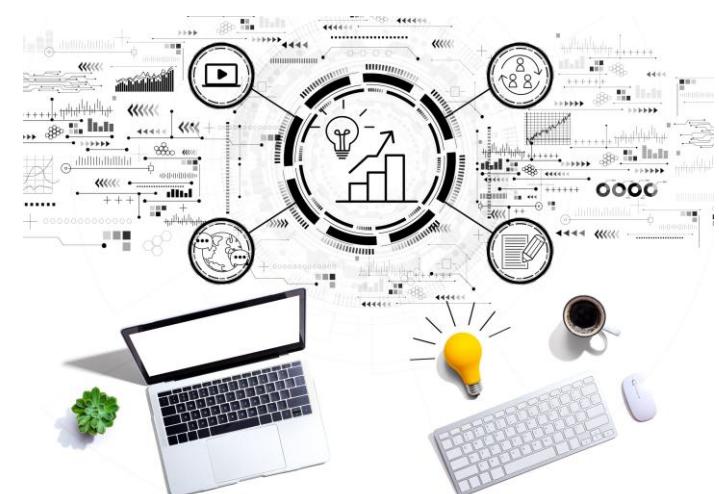
- JavaScript is a great modern language, but it wasn't initially designed to build back-end servers. Since the foundation of the MEAN stack is JavaScript, including the back-end server, it might come with concurrency and performance problems at scale due to JavaScript nature.
- Additionally, since the development opportunity is so rapid, business and server logic might suffer from poor isolation, making potential spaghetti code and bad practices a reality along the way.
- Finally, although there are many guides and tutorials out there, they generally will not include concrete JS coding guidelines appropriate for this stack. Therefore, something that worked really well for one application might surface issues for another.

When can the MEAN stack be used?

- MEAN follows the traditional three-tier stack pattern, including the display tier (Angular.js), application tier (Express.js and Node.js), and database tier (MongoDB).
- If you're building a JavaScript application, particularly in Node.js, then you should give MEAN a serious look.
- MongoDB stores data in a JSON-like format (BSON, a binary JSON extension), the MongoDB Query APIs are defined in JSON, and its command line interface (CLI) is a JavaScript interpreter. Not only is MongoDB essentially a JavaScript/JSON data store, but it's full of advanced features like indexing and querying deep into JSON documents, has powerful native Node.js drivers, and is designed for horizontal scale-out. It's even easier to develop apps in the cloud using [MongoDB Atlas](#), the cloud-native database as a service from the creators of MongoDB.
- Whether you're building a high-throughput API, a simple web application, or a microservice, MEAN is the ideal stack for building Node.js applications.

How secure is the MEAN stack?

- We recommend using the MEAN stack with MongoDB Atlas since Atlas has built-in credentials, a firewall, and end-to-end encryption, which is the best foundation for securing your MongoDB.
- Additionally, the MEAN stack has a concrete three-tier separation that, if used with best practices and correct network isolation, should prevent your end users from having access to the business logic and, moreover, to your database layer. Therefore, your application is by default designed to avoid malicious user interaction from putting your application at risk (query injection, code manipulation, port spoofing, etc.).



Setting Up the Development Environment

- Prerequisites
 - **Basic Knowledge:** HTML, CSS, JavaScript fundamentals.
 - Tools Required:
 - **Code Editor:** VS Code (recommended).
 - **Browser :** Chrome or Firefox for development and debugging.
 - **Package Manager:** npm (Node Package Manager).

Setting Up the Development Environment

- Installation Steps

1. Install Node.js

- Download from the [Node.js official website](#).
- Verify installation using node -v and npm -v in the terminal.

2. Install MongoDB

- Download MongoDB Community Server from [MongoDB official website](#).
- Set up a local MongoDB server or use a cloud database like MongoDB Atlas.

3. Install Angular CLI

- Run npm install -g @angular/cli to globally install Angular CLI.

4. Install Express.js

- Use npm install express in your project folder.

5. Setup Project Structure

- Create a new folder for your project.
- Use ng new for Angular frontend and npm init for backend initialization.

Setting Up the Development Environment

- Now check whether it was installed correctly using below

command
ng --version

```
Angular CLI: 8.3.21
Node: 12.14.0
OS: win32 x64
Angular:
...
Package          Version
@angular-devkit/architect    0.803.21
@angular-devkit/core         8.3.21
@angular-devkit/schematics   8.3.21
@schematics/angular          8.3.21
@schematics/update           0.803.21
rxjs                  6.4.0
```



Setting Up the Development Environment

Now, create a new project using below command:

```
ng new project_name
```

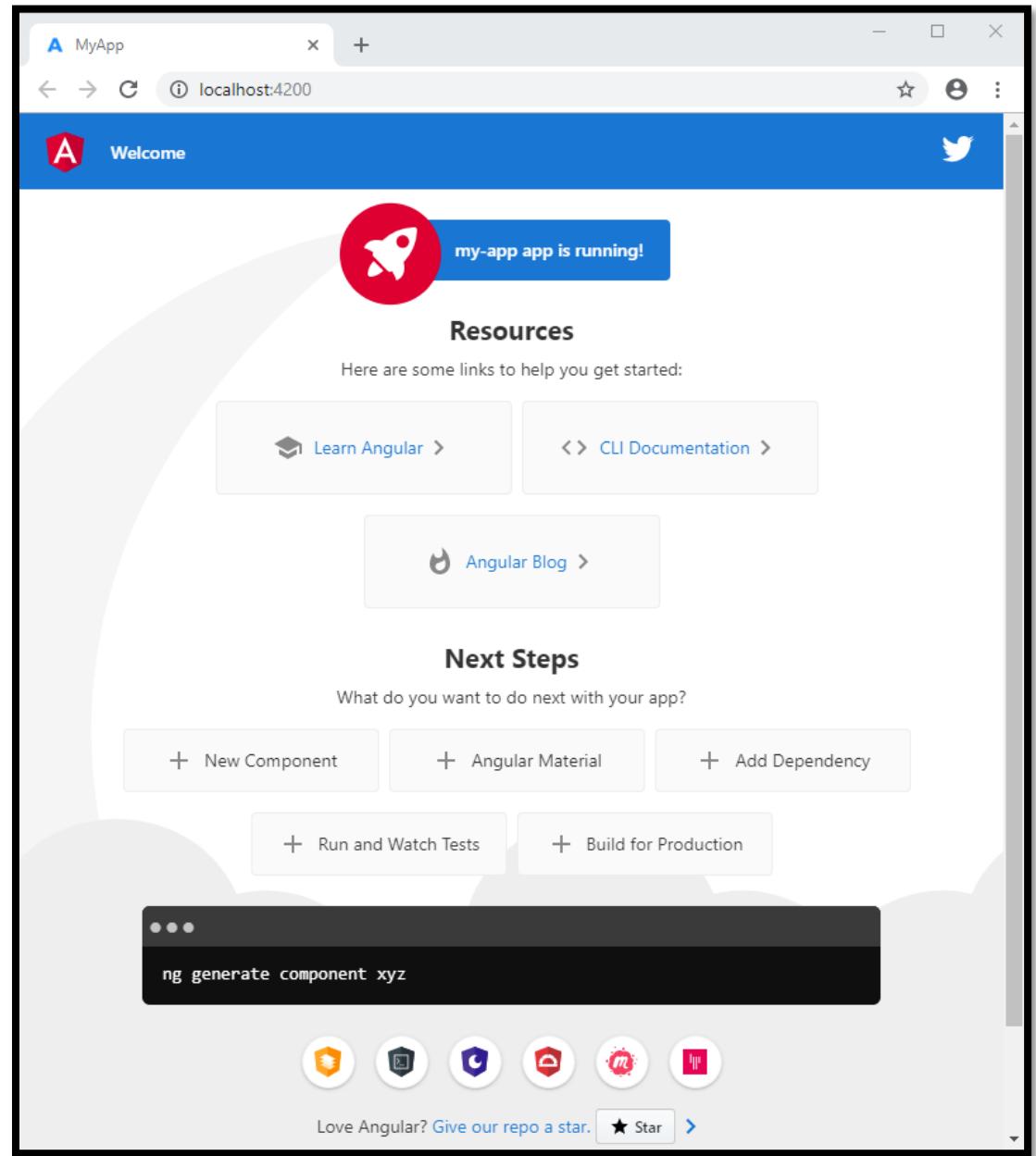
Go to project Directory using below command :

```
cd project_name
```

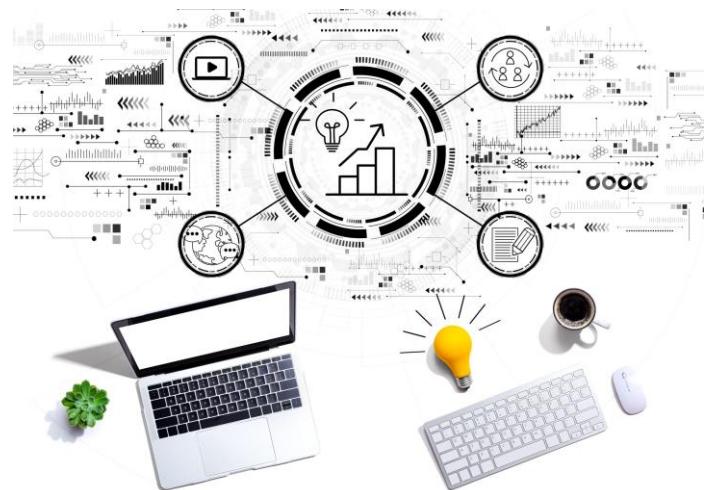
Start the Angular Application using below command:

```
ng serve
```

Application will start on <http://localhost:4200>,



THANK YOU 😊



DIGITAL LEARNING CONTENT



Parul® University



www.paruluniversity.ac.in