**Course Title:** MEAN STACK TECHNOLOGY -LAB

|  |
| --- |
| List of Experments |
| • Angular :  • Getting Started with Angular  • Introduction to Components  • Templates, Interpolation, and Directives  • Data Binding & Pipes  • More on Components  • Building Nested Components  • Forms  • Services and Dependency Injection  • Retrieving Data Using HTTP  Navigation and Routing Basics |
|  |
| • Node Js :  • Introduction  • Exploring language additions to the V8 JavaScript engine  • Understanding NodeJS  • HTTP and File System  • Buffers, Streams, and Events  • Using Express Framework  • Working with Models, Views, and Routes  • Database  • Working with MongoDB  • Working with RESTful web services  Angular With Node |

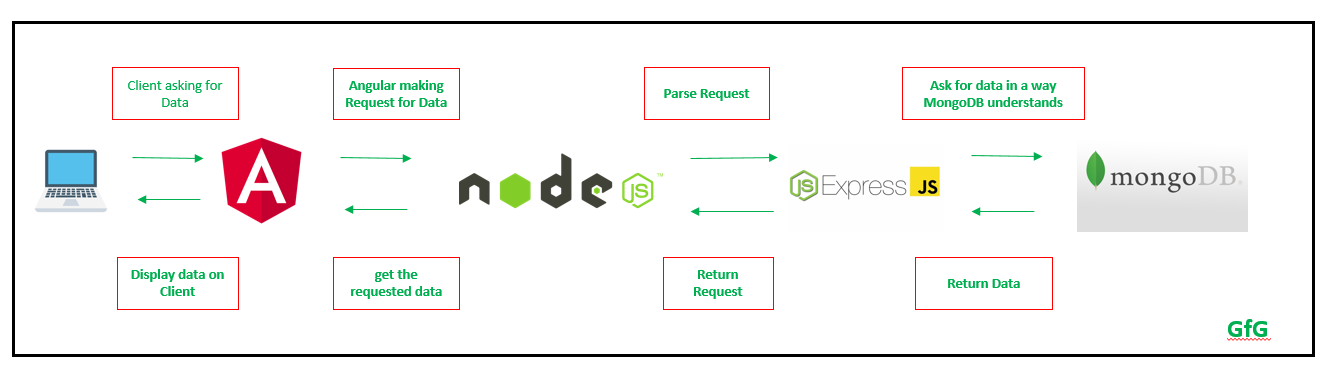
MEAN Stack is one of the most popular Technology Stack. It is used to develop a Full Stack Web Application. Although it is a Stack of different technologies, all of these are based on JavaScript language.

MEAN Stands for:

1. **M** – MongoDB
2. **E** – Express
3. **A** – Angular
4. **N** – Node.js

This stack leads to faster development as well as the deployment of the Web Application. Angular is Frontend Development Framework whereas Node.js, Express, and MongoDB are used for Backend development as shown in the below figure.

**Flow of Data in MEAN Stack Application:** Here, each module communicates with the others in order to have a flow of the data from Server/Backend to Client/Frontend.



**Getting Started with each Technology with examples:** The description of each Technology in this Stack as well as the links to learn them are given below:

**1. Node.js:** Node.js is used to write the Server Side Code in Javascript. One of the most important points is that it runs the JavaScript code outside the Browser. It is cross-platform and Open Source.

* Pre-requisites to learn Node.js- JavaScript/TypeScript
* Go to [Node.js Downloads](https://nodejs.org/en/download/) and click Download button to get the latest version and Install as per your Operating System.
* Verify whether it is installed correctly by checking the version:

node -v

If no version is obtained then it is not installed correctly.

* Check the version of npm (It is installed by default with node):

npm -v

* Create an index.js file inside your project folder and copy the following code to it:

|  |
| --- |
| **var** http = require("http");    http.createServer(**function** (request, response) {       response.writeHead(200, {'Content-Type': 'text/plain'});       // Send the response text as "Hello World"     response.end('Hello World\n');  }).listen(3100);    console.log('Server running at <http://127.0.0.1:3100/>'); |

* Now open terminal and execute the following command:

node index.js

* You will see on Terminal console a log which says:

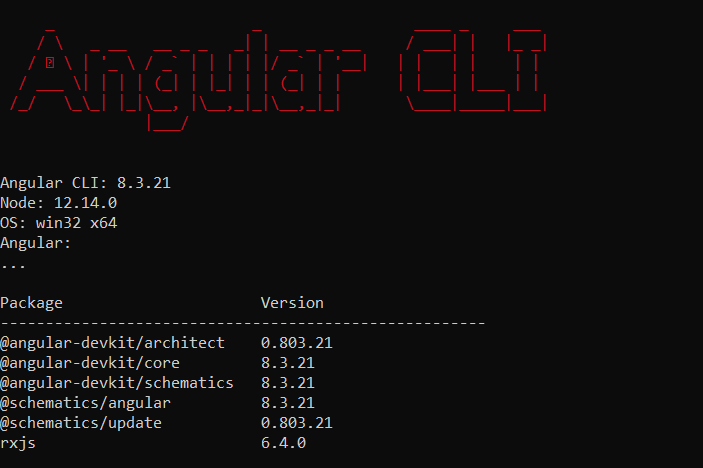
Server running at http://127.0.0.1:3100/

* Go to the browser and type the URL: http://127.0.0.1:3100/ you will see an output as below:  
  **2. AngularJS:** Angular is a Front-end Open Source Framework developed by Google Team. This framework is revised in such a way that backward compatibility is maintained (If there is any breaking change then Angular informs it very early). Angular projects are simple to create using Angular CLI (Command Line Interface) tool developed by the Angular team.
* Pre-requisites to learn Angular:
  1. TypeScript
  2. CSS Preprocessor
  3. Template Code (Angular Material, HTML 5, etc)
* Installing Angular CLI – Command Line Interface using npm (Node Package Manager)

npm install -g @angular/cli

* Now check whether it was installed correctly using below command:

ng --version

It should show something like:  


* Now, create a new project using below command:

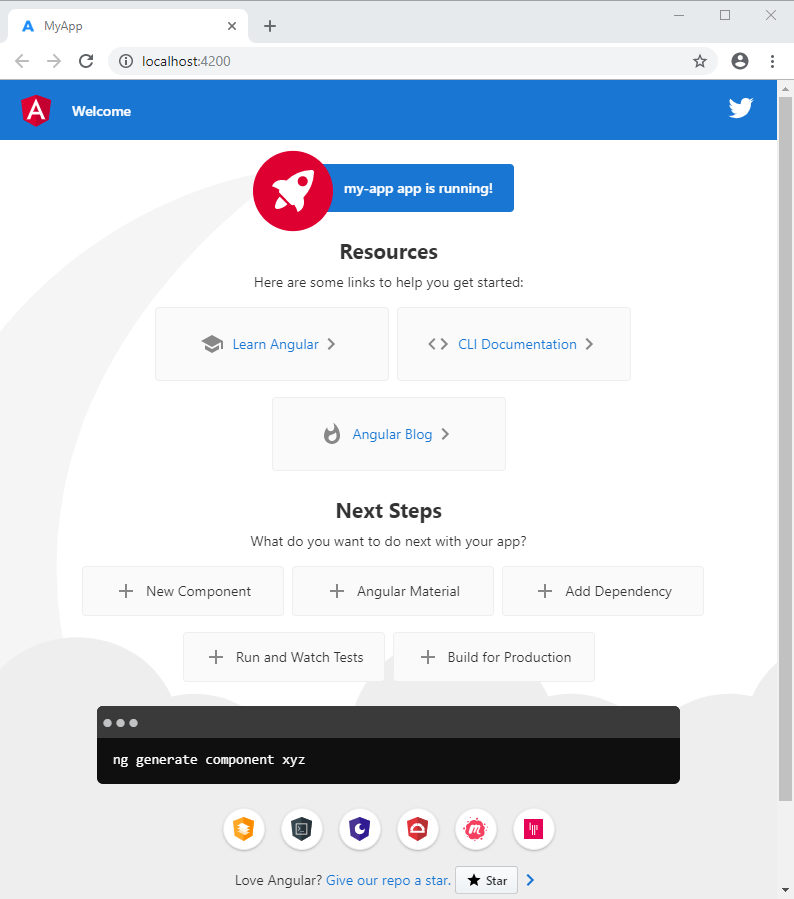
ng new project\_name

* 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, you will see the following:  
  
* Now make changes in app.component.html file and save the file, the application will reload automatically and corresponding changes will be reflected.

**3. MongoDB:** MongoDB is a NoSQL Database. It has JSON like documents. It is document oriented database.

* Pre-requisites to learn MongoDB:
  1. What is Database
  2. Disadvantages of SQL Database
* Creating a database:

use database\_name;

* Create a collection:

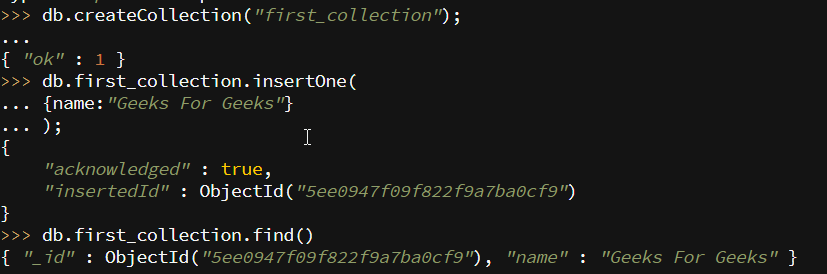
db.createCollection("first\_collection");

* Insert a record in the Collection:
* db.first\_collection.insertOne(
* {name:"Geeks For Geeks"}

);

* Print all the records in a collection:

db.first\_collection.find()



**4. ExpressJS:** Express is a web Framework build on Node.js and used to make API and to build Web Applications.

* Pre-requisites to learn Express:
  1. JavaScript/ TypeScript
  2. Node.js
* Initialize a Project by typing the following command on terminal:

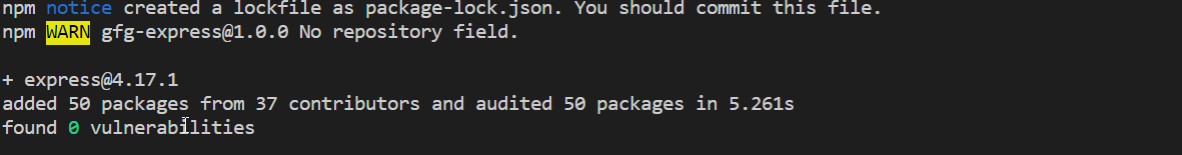
npm init

* It will ask some questions, Press enter in order to set all the default options. This will create package.json file as shown below:

|  |
| --- |
| {    "name": "gfg-express",    "version": "1.0.0",    "description": "Basic Express Node.js Application",    "main": "index.js",    "scripts": {      "test": "echo \"Error: no test specified\" && exit 1"    },    "author": "",    "license": "ISC",  } |

* Install the express using below command:

npm install express --save



* Now, the package.json file will be changed to add the dependencies as shown below:

|  |
| --- |
| {    "name": "gfg-express",    "version": "1.0.0",    "description": "Basic Express Node.js Application",    "main": "index.js",    "scripts": {      "test": "echo \"Error: no test specified\" && exit 1"    },    "author": "",    "license": "ISC",    "dependencies": {      "express": "^4.17.1"    }  } |

* Create index.js file and add the below code to it:

|  |
| --- |
| const express = require('express')  const app = express()  const PORT = 3000    app.get('/', (req, res) =>          res.send('Hello World!'))    app.listen(PORT, () => console.log(  `Example app listening at http://localhost:${PORT}`)) |

* Start the express server using below command:

node index.js

* Go to http://localhost:3000 to see the output as below:
* [Switch or Create a new MongoDB Database (tutorialsteacher.com)](https://www.tutorialsteacher.com/mongodb/create-mongodb-database)