Week-3 Web application Assignment

Explain in detail about MEAN Stack.

What is the MEAN stack?

The MEAN stack is a JavaScript-based framework for developing web applications. MEAN is named after MongoDB, Express, Angular, and Node, the four key technologies that make up the layers of the stack.

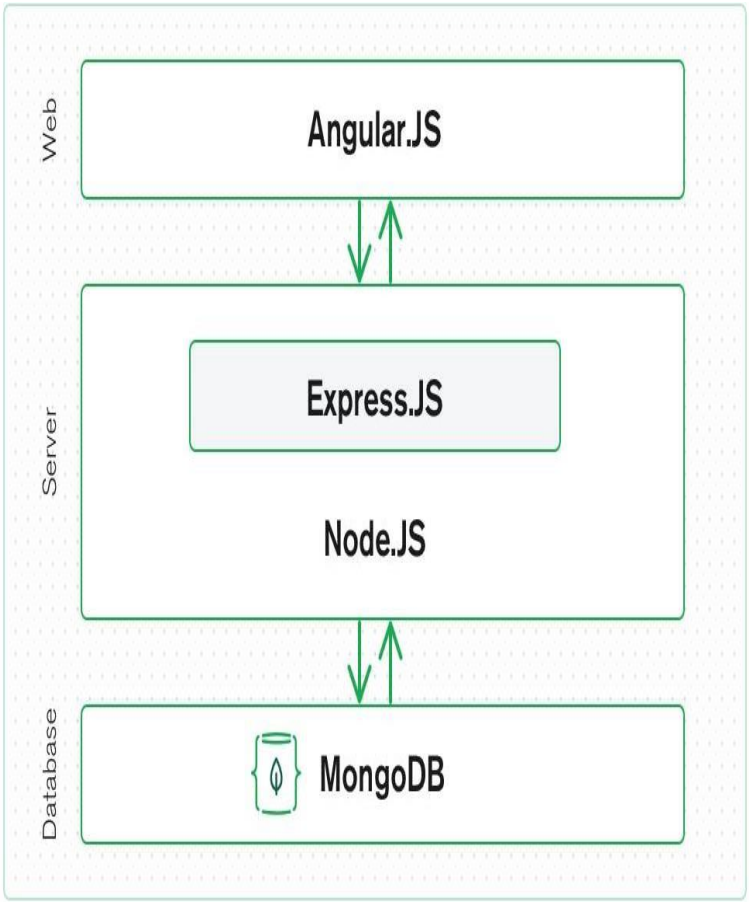
- MongoDB — document database
- Express(.js) — Node.js web framework
- Angular(.js) — a client-side JavaScript framework
- Node(.js) — the premier JavaScript web server

There are variations to the MEAN stack such as MERN (replacing Angular.js with React.js) and MEVN (using Vue.js). The MEAN stack is one of the most popular technology concepts for building web applications.

How does the MEAN stack work?

MEAN stack architecture

The MEAN architecture is designed to make building web applications in JavaScript and handling JSON incredibly easy.



MEAN stack components

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 powerfully 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.

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 using promises, to access and update data in your MongoDB database.

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.

Again, if you want to easily get the best of MongoDB, you'll want to look at MongoDB Atlas. This will allow 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
- Todo and calendar applications
- Interactive forums

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 Language (MQL) is 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.

Using MEAN stack with MongoDB Atlas

The MongoDB Node.js driver makes working with MongoDB from inside a Node.js script simple and intuitive for developers — saving developers time and increasing their productivity.

Next, you'll need a MongoDB database. The easiest way to get started with MongoDB is to create a free cluster in MongoDB Atlas, MongoDB's fully managed, multi-cloud document database as a service.

Atlas databases are easily deployed and scaled, providing you with a consistent URI to connect. See the official MongoDB documentation on connecting to a cluster.

Along the way, Atlas connections come with built-in username/password and TLS end-to-end encryption by default. Additionally, these connections allow you to utilize advanced MongoDB security features such as certificate/IAM authentication, LDAP, Encryption-at-rest, and Auditing with the click of a button.

Moreover, an Atlas project can utilize the Atlas App Services applications platform to easily integrate many authentication providers such as Google, Facebook, JWT, and custom authentication.

Scaling and managing Atlas is very easy; its biggest benefit is that it supports and secures the MEAN stack's most valuable layer: the data layer.