4 marks questions

Compare and contrast the use of MongoDB and MySQL in MEAN stack development.

MongoDB is an open-source database developed by MongoDB, Inc. MongoDB stores data in JSON-like documents that can vary in structure. It is a popular NoSQL database.	MySQL is a popular open-source relational database management system (RDBMS) that is developed, distributed and supported by Oracle Corporation.

How Data is Stored?

MongoDB	MySQL
In MongoDB, each individual records are stored as 'documents'.	In MySQL, each individual records are stored as 'rows' in a table.
MongoDB	MySQL
Documents belonging to a particular class or group as stored in a 'collection'. Example: collection of users.	A 'table' is used to store rows (records) of similar type.
MongoDB	MySQL
MongoDB is what is called a NoSQL database. This means that pre-defined structure for the incoming data can be defined and adhered to but also, if required different documents in a collection can have different structures. It has a dynamic schema.	MySQL as the name suggests uses Structured Query Language (SQL) for database access. The schema can not be changed. The inputs following the given schema are only entered.

SALIENT FEATURES

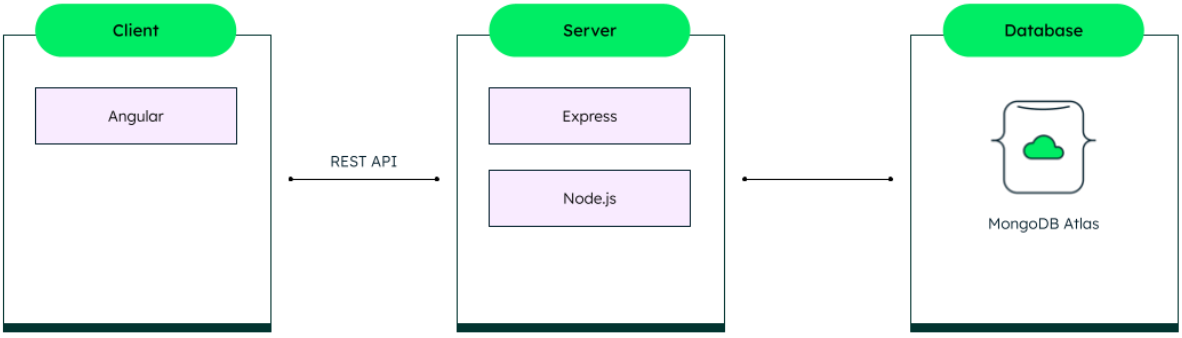
MongoDB	MySQL
MongoDB was designed with high availability and scalability in mind, and includes out-of-the-box replication and sharding.	MySQL concept does not allow efficient replication and sharding but in MySQL one can access associated data using joins which minimizes duplication.

How the MEAN stack components work together to create a web application?

MEAN is a technology stack used for building full stack applications. It's a combination of the following technologies:

- MongoDB—document database
- Express—a Node.js framework for building APIs
- Angular—front-end application framework
- Node.js—server-side JavaScript runtime environment

Applications built with the MEAN stack follow the client-server architecture. The client, built with Angular, can be a web application, a native mobile application, or a desktop application. The client communicates with the server through an API, which is built with Express. The server then manages the requests with the MongoDB database.



1. 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.

2. 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.

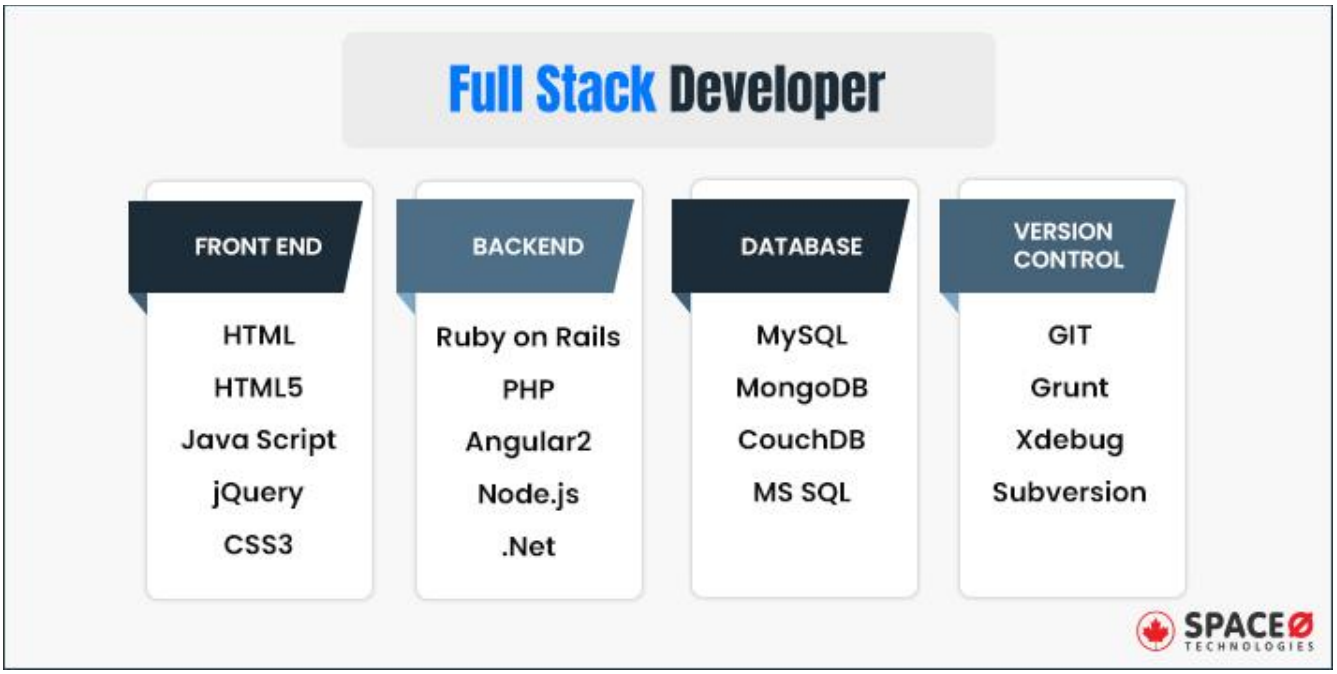
3. MongoDB: MongoDB is a NoSQL Database. It has JSON like documents. It is document oriented database.

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

Explain the concept of full-stack development and its advantages.

A full-stack developer means a developer who is skilled to work exclusively on the application's front-end development technologies, backend web development technologies, databases, DevOps, and

mobile apps. A full-stack developer deals with the technologies behind the entire application stack. Check this image for reference.



To be precise, a full-stack developer is a jack of all trades; works with JavaScript code as well as MySQL queries.

Full-stack engineers have a clear understanding of software architecture:

- Presentation layer: Deals with the front-end part of the application, i.e. user interfaces
- Business logic layer: Deals with the data validation, connectivity, server
- Database layer: Deals with the connectivity of the database from the application's front-end

In short, a full-stack programmer knows how to work on the client and server-side and what's happening in the app during the development process.

They can work on a web stack or native application stack depending on the client's requirements. Thus, a full stack web developer is almost like a T-shaped developer who has knowledge of the specific language and is sound with a range of generalized concepts.

The primary responsibility of a full-stack developer is to design user interactions on websites and platforms, write optimized code for mobile, and develop databases. First, let's look at the other responsibilities of full-stack developers.

Here is a list of full stack developer's responsibilities:

- Understand the requirements of the client and work with the development team and project managers to design a solution
- Manage databases, servers, and third-party apps
- Design and develop API
- Need to work with the entire software development team, including front-end, middleware, and back-end
- Build Proof-of-Concept (POC) to demonstrate the clients

- Ensures cross-platform optimization for mobile phones
- Ensures responsive design for all the platforms
- Works with the graphic designers for the web design features
- Understand software architecture concepts
- Test and debug the issues in the code
- From concept to implementation; they transform requirements into a finished product
- Build reusable code for future use
- Work with the creative team to design the innovative solution
- Ensures high quality of code on design and website
- Stay updated with the latest web applications and programming languages

What is Front-End Development?

Front-end development deals with the user interface of a website or application that includes, buttons, images, text, tables and graphs.

A front-end development is fully focused on "client-side" development of software, web and application that code and develop the front-end components. Front-end components mean UI design elements and functions that the end-user interacts with directly.

A front-end developer is in control of everything you see on the screen and works to refine and smooth out the user interface for a better user experience. Simply, front-end web development deals with the interface and designing of an application.

What is Back-End Development?

Back-end development is scripting, architecture, and writing code of an app that communicates with the database. The back-end web development process involves server-side application logic and integration, as opposed to frontend development, which focuses on customer-facing products and programs.

Back-end developers create code for an application that connects the database with another component of an application. Apparently, back-end developers takes the charge of writing logical functionalities of application.

Furthermore, check the technologies considered in full-stack web development.

1. Get Your Project Developed With a Limited Budget

   Rather than hire different professionals for the front-end and back-end separately, it is recommended to hire a full-stack developer. In addition, you can have fewer people on board as full-stack professionals easily handle the front-end and back-end of the website.

   Full-stack developers' knowledge and flexibility aid in the elimination of causes that contributes to cost overruns in the long term. This is one of the advantages of full-stack development that no business owners developing web applications can afford to ignore.

   Hiring developers and managing them separately needs time, money, and effort. So, full-stack developers reduce this hassle.

2. Get Your Project Delivered on Time

Full-stack developers often work as a team within a project. This makes it possible to share job responsibilities among the participants for creating web applications. Since full-stack developers are aware of different techniques and tools, they will spend less time discussing and helping everyone in the team.

Plus, skilled full-stack web developers make the client's website compliant with the current online safety standards and regulations. With the expertise in diverse tools and technologies, they are able to find performance bottlenecks in current web/mobile apps and quickly devise solutions.

Having a full-stack developer in the team streamlines the overall process of web development, which results in fast and timely delivery of the project.

3. Quick Troubleshooting Throughout The Project

   Several types of bugs and errors can occur while developing an app or website. And this can happen at any point in web/app development. Finding the source of such errors is time-consuming for most developers. This results in an increase in costs and delays in projects.

   However, the involvement of a full-stack developer improves the development procedure because such experienced full-stack developers quickly resolve the issues as they have a core understanding of the functionalities.

   Since full-stack developers understand various emerging technologies, they provide better support and maintenance in application development. These programmers are more capable of troubleshooting than most developers.

4. Utilize the Diverse Experience of Full Stack Developers

   Full-stack developers act as a bridge between the client and team; they can easily communicate the requirements and are skilled enough to identify the problems during the development. They even will quickly evaluate project specifications and create a viable blueprint and implementation stages.

   An experienced full stack professional has in-depth knowledge about web and mobile app development. In addition, full-stack web developers are knowledgeable about the complexities of network and application development.

   Their knowledge of the industry aids these developers in coming up with novel full-stack project ideas and overcoming unexpected challenges.

5. Get Scalable and Optimizable Code for Your Project

   Since full-stack web programmers are familiar with the most recent updates and innovations, the method of managing and upgrading the Minimum Viable Product is streamlined, increasing the likelihood of favourable feedback from future consumers.

As a result, full-stack programmers are able to implement the same method in their practice, allowing customers to benefit from getting access to the latest up-to-date technology. And the clients will get the best software applications developed with the latest tech stacks.

6. Having a Developer in Your Team Who Understands In and Out

   The full-stack web developer you recruit will be accountable for the whole project, from designing front-end architecture to backend development and database management. Full-stack developers and his team takes the charge of whole project's developing process.

   As a result, when there are any problems with the delivery period or product development, the full stack developer is kept responsible. So, there are no issues or blame-shifting between any members of team. The standard of projects handled by a full-stack web developer is certainly higher than projects handled by conventional developers.

   A full-stack engineer can take up the ownership of the entire project from beginning to end. If the client has asked for proof-of-concept (POC), they get involved in the middle of the project and are able to quickly build the application. Having a full-stack web developer project development adds beneficial it ensure to provide quality product.

7. Always Heading With Trends

   A professional full-stack web developer always keeps him/herself up to date with new technologies. For example, these developers are familiar with progressive web applications, blockchain technologies, machine learning, and artificial intelligence (AI) development.

   Since they are updated with the latest technology and frameworks, they can implement smart features to build responsive and interactive websites. Moreover, developers choose full-stack technologies depending on the trends of a web stack or native application stack.

   In addition, a full-stack developer can implement the whole design structure and be able to share innovative design solutions with the UI/UX team. Apart from web development, they maintain and optimize the existing system. Using such advanced technologies

   Custom meat processing software that helps to automate the entire operation from receiving to tracking orders through a single platform. By implementing this software organization's productivity surged by 60% and saved 70% time on generating reports.

8. Flexibility of Full Stack Developers to Move Between Front-end and Back-end

   A full-stack programmer offers a lot of flexibility because he knows both the client and server side of the application. Full-stack web developers know coding languages like Python, .NET, PHP, JavaScript, Node.js, and other latest web development frameworks. So, it becomes easy to switch between different tasks in a snap. Apart from front-end and back-end, full-stack engineers even understand DevOps and are adept at handling all three-tier architecture.

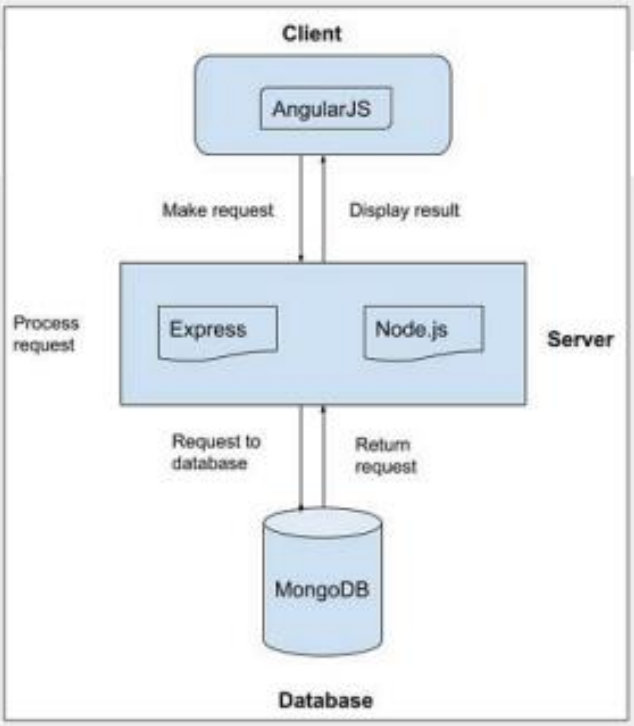
   In the presence of a full-stack developer, the backend developers do not need to meet front-end developers to illustrate the problem and vice versa. A full-stack mobile developer is aware of the outcome and issues during development. So, it brings

Explain Mean and its architecture in detail.

MEAN is an open source JavaScript framework, used for building dynamic websites and web applications. It includes following four building blocks to build an application.

- **MongoDB** − It is a document database, that stores data in flexible, JSON-like documents.
- **Express** − It is web application framework for Nodejs.
- **Node.js** − It is Web Server Platform. It provides rich library of various JavaScript modules which simplifies the development of web applications.
- **AngularJS** − It is a web frontend JavaScript framework. It allows creating dynamic, single page applications in a clean Model View Controller (MVC) way.

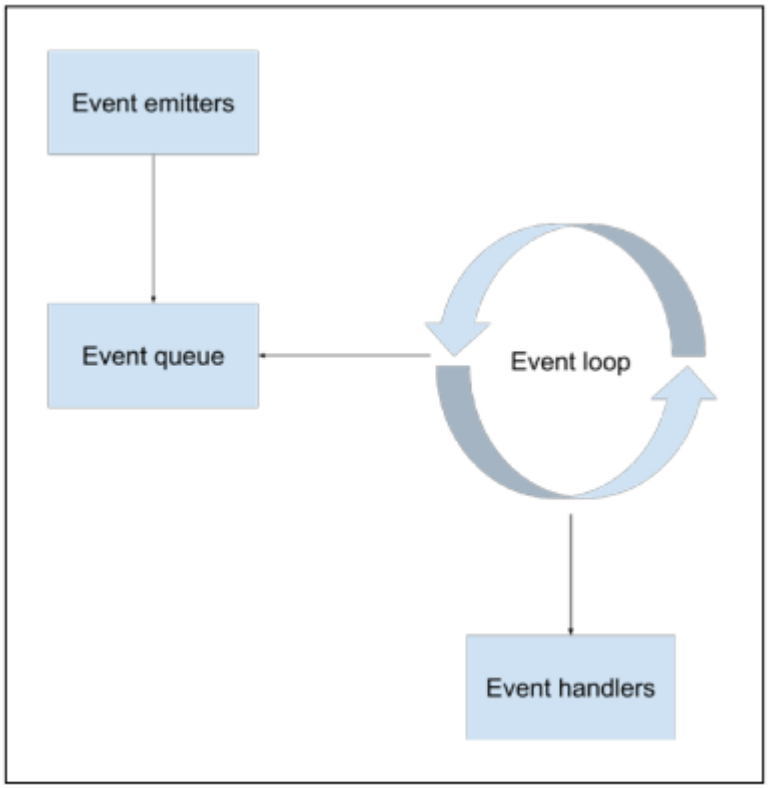
For more information on these, you can refer the overview chapter. The below diagram depicts architecture of MEAN stack application.



As shown in the above image, we have AngularJS as client side language which processes the request of a client.

- Whenever a user makes a request, it is first processed by AngularJS.
- Next, request enters second stage, where we have Node.js as server side language and *ExpressJS* as backend web framework.
- *Node.js* handles the client/server requests and *ExpressJS* makes request to the database.
- In the last stage, *MongoDB* (database) retrieves the data and sends the response to ExpressJS.
- ExpressJS returns the response to Nodejs and in turn to AngularJS and then displays the response to user.

Explain event-driven programming in Node.js.



The event loop cycle

Node.js uses the event-driven nature of JavaScript to support non-blocking operations in the platform, a feature that enables its excellent eficiency. JavaScript is an event-driven language, which means that you register code to speciic events, and that code will be executed once the event is emitted. This concept allows you to seamlessly execute asynchronous code without blocking the rest of the program from running.

vent-driven architecture will help you dramatically reduce the load on your server while leveraging JavaScript's asynchronous behavior in building your web application. This approach is made possible thanks to a simple design pattern, which is called closure by JavaScript developers. A function that listens for the triggering of an event is said to be an 'Observer'. It gets triggered when an event occurs. Node.js provides a range of events that are already in-built. These 'events' can be accessed via the 'events' module and the EventEmitter class. Most of the in-built modules of Node.js inherit from the EventEmitter class EventEmitter: The EventEmitter is a Node module that allows objects to communicate with one another. The core of Node's asynchronous event-driven architecture is EventEmitter. Many of Node's built-in modules inherit from EventEmitter.

The idea is simple – emitter objects send out named events, which trigger listeners that have already been registered. Hence, an emitter object has two key characteristics:

- Emitting name events: The signal that something has happened is called emitting an event. A status change in the emitting object is often the cause of this condition.
- Registering and unregistering listener functions: It refers to the binding and unbinding of the callback functions with their corresponding events.

Advantages of Event-Driven Programming:

- Flexibility: It is easier to alter sections of code as and when required.

- Suitability for graphical interfaces: It allows the user to select tools (like radio buttons etc.) directly from the toolbar
- Programming simplicity: It supports predictive coding, which improves the programmer's coding experience.
- Easy to find natural dividing lines: Natural dividing lines for unit testing infrastructure are easy to come by.
- A good way to model systems: Useful method for modeling systems that must be asynchronous and reactive.
- Allows for more interactive programs: It enables more interactive programming. Event-driven programming is used in almost all recent GUI apps.
- Using hardware interrupts: It can be accomplished via hardware interrupts, lowering the computer's power consumption.
- Allows sensors and other hardware: It makes it simple for sensors and other hardware to communicate with software.

Disadvantages of Event-Driven Programming:

- Complex: Simple programs become unnecessarily complex.
- Less logical and obvious: The flow of the program is usually less logical and more obvious
- Difficult to find error: Debugging an event-driven program is difficult
- Confusing: Too many forms in a program might be confusing and/or frustrating for the programmer.
- Tight coupling: The event schema will be tightly coupled with the consumers of the schema.
- Blocking: Complex blocking of operations.

What are Node modules, and how are they used in building Node.js web applications?

In simple terms, a module is a piece of reusable JavaScript code. It could be a .js file or a directory containing .js files. You can export the content of these files and use them in other files.
Modules help developers adhere to the DRY (Don't Repeat Yourself) principle in programming. They also help to break down complex logic into small, simple, and manageable chunks.

Types of Node Modules

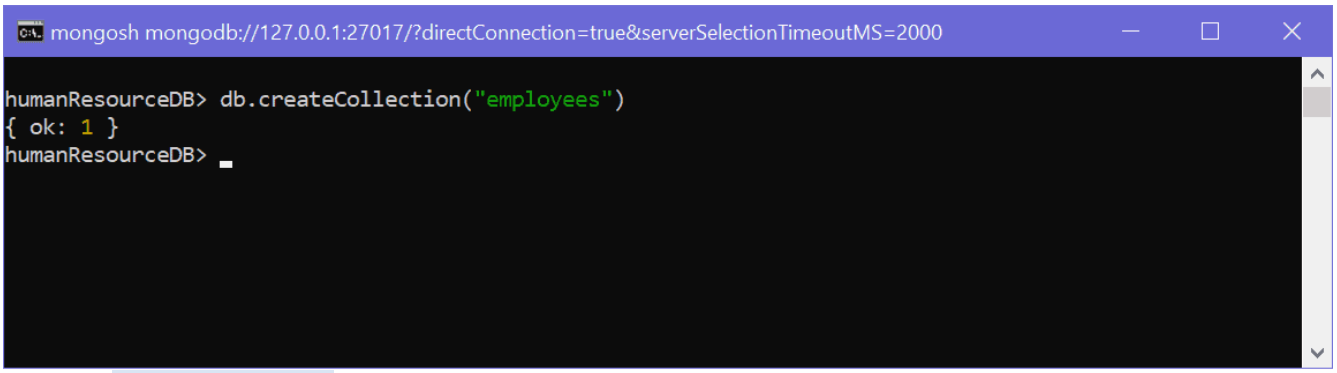
There are three main types of Node modules that you will work with as a Node.js developer. They include the following.

- Built-in modules
- Local modules
- Third-party modules

Explain the concept of collections in MongoDB.

A collection in MongoDB is similar to a table in RDBMS. MongoDB collections do not enforce schemas. Each MongoDB collection can have multiple documents. A document is equilant to row in a table in RDBMS.

To create a collection, use the db.createCollection() command. The following creates a new employees collection in the current database, which is humanResourceDB database created in the previous chapter.



Use the show collections commands to list all the collections in a database.

How to design a common MEAN stack architecture using a API built in Node.js, Express, and MongoDB?

Must follow mean blog of my

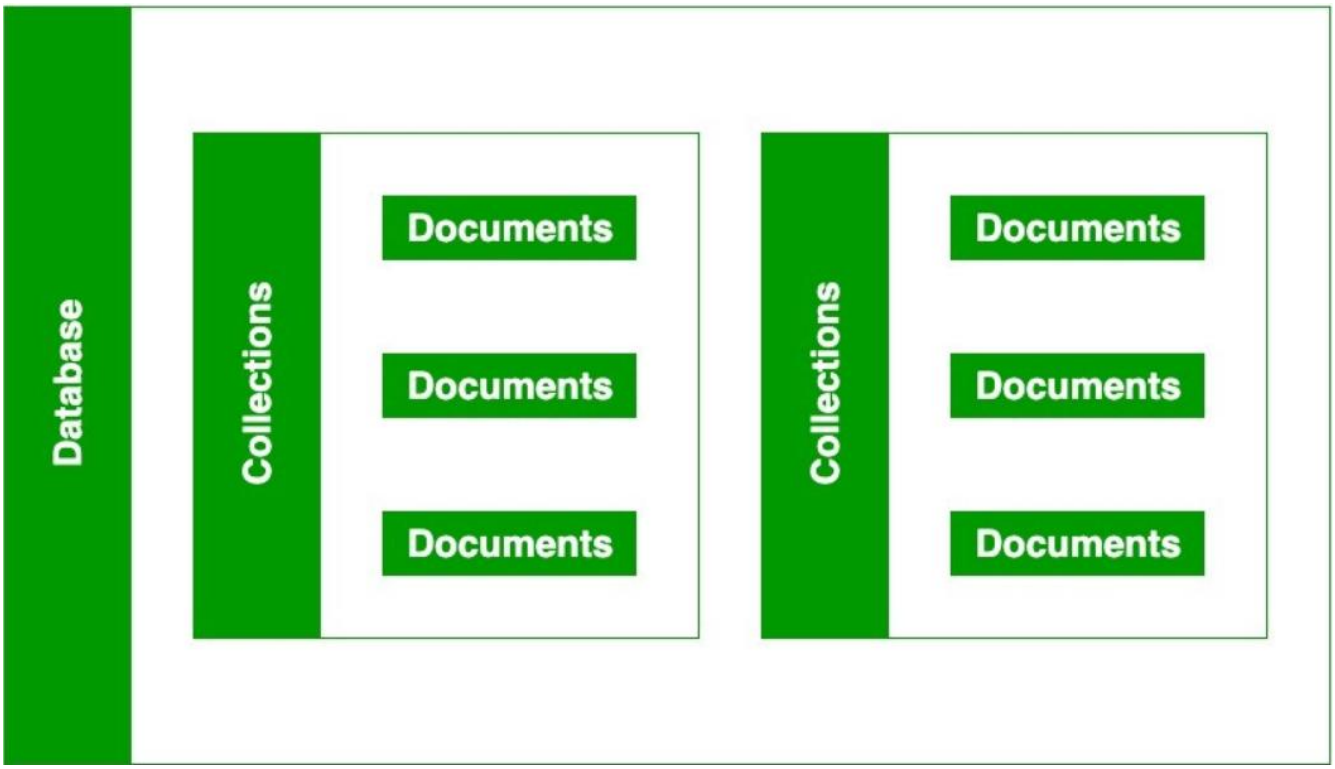
Discuss the role of Mongoose schemas in MongoDB development.

MongoDB collections A MongoDB collection is a list of MongoDB documents and is the equivalent of a relational database table. A collection is created when the irst document is being inserted. Unlike a table, a collection doesn't enforce any type of schema and can host different structured documents. To perform operations on a MongoDB collection, you'll need to use the collection methods. Let's create a posts collection and insert the irst post. In order to do this, execute the following command in the MongoDB shell: > db.posts.insert({"title":"First Post", "user": "bob"}) After executing the preceding command, it will automatically create the posts collection and insert the irst document. To retrieve the collection documents, execute the following command in the MongoDB shell: > db.posts.find()

This means that you have successfully created the posts collection and inserted your irst document. To show all available collections, issue the following command in the MongoDB shell: > show collections The MongoDB shell will output the list of available collections, which in your case are the posts collection and another collection called system.indexes, which holds the list of your database indexes. If you'd like to delete the posts collection, you will need to execute the drop() command as follows: > db.posts.drop() The shell will inform you that the collection was dropped, by responding with a true output

Connecting to your MongoDB instance was the irst step but the real magic of the Mongoose module is the ability to deine a document schema. As you already know, MongoDB uses collections to store multiple documents, which aren't required to have the same structure. However, when dealing with objects, it is sometime necessary for documents to be similar. Mongoose uses a Schema object to deine the document list of properties, each with its own type and constraints, to enforce the document structure. After specifying a schema, you will go on to deine a Model constructor that you'll use to create instances of MongoDB documents. In this section, you'll learn how to deine a user schema and model, and how to use a model instance to create, retrieve, and update user documents.

What is the difference between MongoDB database, collection, and document?



How do you install Express and create your first Express application?

Server.js file follow it in my mean blog