VELAGAPUDI RAMAKRISHNA SIDDHARTHA ENGINEERING COLLEGE
DEPARTMENT OF INFORMATION TECHNOLOGY

FULL STACK DEVELOPMENT

Presentation On

MEAN STACK TECHNOLOGIES

CODE : 20IT7607 B

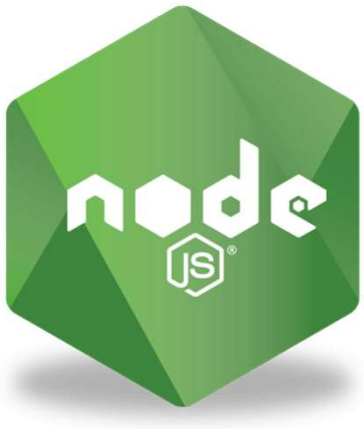
Presented By
MOHAMMAD RIZWANULLAH (208W1A1299)

Under The Guidance Of
Dr . M . Ashok Kumar, Assistant Professor, Ph.D

Technology : Node JS

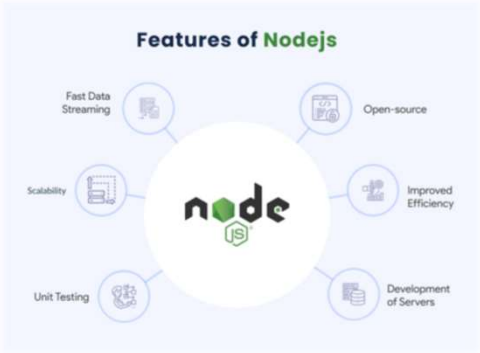
Introduction :

- Node.js is an open-source, server-side runtime environment that allows you to run JavaScript code on the server.
- It is built on the V8 JavaScript engine, which is the same engine used by Google Chrome.
- Node.js is event-driven and non-blocking, making it well-suited for building scalable and high-performance applications.
- It provides a rich set of built-in modules and libraries for various tasks, such as file I/O, networking, and more.
- Node.js is commonly used for building web applications, APIs, real-time applications, and server-side scripting.
- It uses a single-threaded event loop to handle asynchronous operations, which can lead to efficient resource utilization.



FEATURES :

- Node.js is designed to be non-blocking and event-driven, allowing it to handle many simultaneous connections without blocking the execution of other code. This is particularly useful for building real-time applications and handling I/O-intensive tasks.
- Node.js is built on the V8 JavaScript engine, which compiles JavaScript to machine code, making it extremely fast in execution.
- Node.js operates on a single-threaded event loop, which efficiently manages asynchronous I/O operations. While this single-threaded model can simplify programming, it can be complemented with worker threads for CPU-bound tasks.
- Node.js is cross-platform and can run on various operating systems, making it versatile and accessible for developers on different platforms.
- Node.js allows developers to use JavaScript on the server side, which can lead to code reusability and consistency, especially in full-stack web development.



CALLBACK In Node :

- Callbacks are a fundamental feature in Node.js that enables asynchronous programming. They allow you to define functions that are executed when a specific operation, such as reading a file or making an HTTP request, completes, ensuring non-blocking code execution.
- Callbacks are functions passed as arguments to other functions. They are typically executed once the asynchronous operation finishes, helping manage the flow of control in your code.
- While callbacks are essential, they can lead to callback hell or "Pyramid of Doom" when dealing with multiple asynchronous operations. Many developers use Promises or async/await to simplify asynchronous code and make it more readable.
- Node.js also provides an EventEmitter class that allows you to create and handle custom events. This can be a more flexible way to manage asynchronous events and communication in your application, often used in combination with callbacks.

Debugging :

- Debugging and testing are critical aspects of developing Node.js applications. Proper testing ensures that the application is free from bugs and errors, while debugging helps developers identify and fix issues in the code. In this section, we will discuss some best practices for debugging and testing Node.js applications.

- Node.js provides several built-in debugging tools that can be used to identify and fix errors in the code. The most commonly used debugging tool is the Node.js debugger, which can be accessed by running the Node.js application with the --inspect flag. This opens a debugging session in Chrome DevTools, allowing developers to set breakpoints, inspect variables, and step through the code.

Testing :

- Node.js applications can be tested using a variety of frameworks, including Mocha, Jasmine, and Jest. These frameworks provide a range of features, including test runners, assertion libraries, and mocking utilities.

- Node.js has a variety of testing frameworks to choose from, such as Mocha, Jest, and Jasmine. These frameworks provide a structured way to write and execute tests for your Node.js applications. They often include features for test suites, test cases, assertions, and reporting.

- To isolate and test specific components of your application, mocking and stubbing libraries like Sinon or Jest's mocking capabilities allow you to simulate external dependencies, such as databases or APIs. This helps ensure that your tests focus on the specific functionality you're testing without relying on the actual implementations of external services.

TECHNOLOGY : EXPRESS

Introduction :

- Start by clearly defining what "Express" means in the context of your presentation. If you're talking about Express.js, mention that it's a popular web application framework for Node.js. If it's about expressing ideas or concepts, explain the broader concept.
- Explain why Express is relevant. Highlight its significance in the field, such as how Express.js streamlines web development or how expressing ideas effectively can lead to better communication.
- If applicable, give a brief overview of the history and evolution of Express or the evolution of expressing ideas or concepts. This can help your audience understand the background and context.

Features :

- Express provides a simple way to route requests to different handlers.
- Express allows you to use middleware to add functionality to your application.
- Express can serve static files, such as images, CSS, and JavaScript files.
- Express can manage sessions, which allows you to store data about users across requests.
- Express provides a way to handle errors that occur in your application.
- Express is designed to be secure, making it a good choice for building applications that need to handle sensitive data.

```
var express = require('express');
var app = express();
app.get('/', function (req, res)
{
  res.send('Welcome to Express');
});
var server = app.listen(8000, function ()
{
  var host = server.address().address;
  var port = server.address().port;
  console.log('app listening at http://%s:%s',
  host, port);
});
```

Routing :

- Routing is made from the word route. It is used to determine the specific behavior of an application. It specifies how an application responds to a client request to a particular route, URI or path and a specific HTTP request method (GET, POST, etc.). It can handle different types of HTTP requests.
- Route paths, in combination with a request method, define the endpoints at which requests can be made. Route paths can be strings, string patterns, or regular expressions.
- Route handlers can be in the form of a function, an array of functions, or combinations of both.
- You can create chainable route handlers for a route path by using app.route(). Because the path is specified at a single location, creating modular routes is helpful, as is reducing redundancy and typos. For more information about routes

```
var express = require('express');
var app = express();
app.get('/', function (req, res) {
   res.send('Welcome to Student Portal!');
})
app.post('/', function (req, res) {
   res.send('I am Impossible! ');
})
app.delete('/del_student', function (req, res) {
   res.send('I am Deleted!');
})
var server = app.listen(8000, function () {
var host = server.address().address
  var port = server.address().port
  console.log("Example app listening at http://%s:%s",
  host, port)
})
```

Advantages :

- Express.js is a minimal and lightweight web framework that provides developers with a basic set of features for building web applications. It is designed to be flexible and modular, allowing developers to easily add new functionality as needed. Express.js does not include a lot of unnecessary features, making it fast and efficient.
- Express.js is easy to learn and use, making it an ideal choice for both novice and experienced developers. It provides developers with a simple and intuitive API for building web applications. Express.js is built on top of Node.js, which means that developers can leverage the power of the Node.js ecosystem and easily integrate with other Node.js modules.
- One of the key features of Express.js is its support for middleware. Middleware is a function that is executed for every HTTP request that is received by the server. Middleware can be used to perform a variety of tasks, such as logging, authentication, and error handling. Express.js provides developers with a wide range of built-in middleware functions, as well as the ability to create custom middleware.
- Express.js provides developers with a variety of options for working with databases. It has built-in support for popular databases, such as MongoDB and MySQL, as well as the ability to use other Node.js database modules. Developers can easily connect to databases, perform CRUD operations, and handle database transactions.

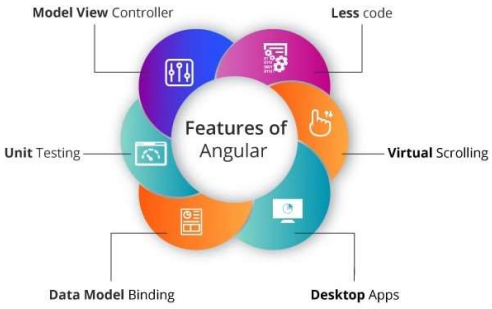
TECHNOLOGY : ANGULAR

Introduction :

- Angular is a TypeScript-based open-source full-stack web application framework.
- Instead of scope and controller, Angular uses hierarchy of components as its primary architectural characteristic.
- Angular uses the different expression syntax. It uses "[ ]" for property binding, and "( )" for event binding.
- Angular uses of Microsoft's TypeScript language, which provides Class-based Object Oriented Programming, Static Typing, Generics etc. which are the features of a server-side programming language.
- Angular supports dynamic loading of the page.

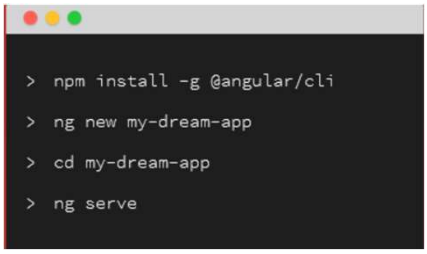
Features :

- Angular facilitates you to create desktop installed apps on different types of operating systems i.e. Windows, Mac or Linux by using the same Angular methods which we use for creating web and native apps.
- You can built native apps by using Angular with strategies from Cordova, Ionic, or NativeScript.
- Progressive web applications are the most common apps which are built with Angular. Angular provides modern web platform capabilities to deliver high performance, offline, and zero-step installation apps.
- Angular apps are fast and loads quickly with the new Component Router, which delivers automatic code-splitting so users only load code required to render the view they request.



Separation Of Responsibilities :

- Angular applications are typically organized into components,which are the building blocks of the user interface. Each component should have a well-defined responsibility, such as rendering a specific part of the UI or handling a specific feature.
- Keep the presentation logic in the component's template (HTML). Templates should focus on displaying data and user interactions and should not contain business logic. This separation makes it easier to maintain and test your UI.
- Angular services are used to encapsulate data and logic that can be shared amongmultiple components. Services are responsible for making API requests, handling data, andmanaging state. Separating these concerns into services promotes reusability and maintainability.
- Angular provides pipes for transforming and formatting data in templatesand directives for extending the behavior of HTML elements.



Advantages :

- ❑ Angular applications are highly performant, thanks to its use of a virtual DOM and its ability to optimize code execution.
- ❑ Angular includes a wide range of features that make it easy to create complex and sophisticated web applications.
- ❑ Angular has a large and active community that provides support and resources for developers.
- ❑ Angular is backed by Google, which means that it is well-maintained and will continue to be developed in the future.
- ❑ TypeScript is a superset of JavaScript that includes additional data types, such as classes, enums, and interfaces. This makes it easier to write code that is more secure, maintainable, and reusable.
- ❑ Angular's code is designed to be concise and easy to maintain. This is due in part to its use of TypeScript's data types and its built-in code linting and minification features.
- ❑ Angular applications are easy to test, thanks to its built-in testing framework. This framework provides a number of features that make it easy to write unit tests, integration tests, and end-to-end tests.

TECHNOLOGY : MONGO DB

Introduction :

- MongoDB is a No SQL database. It is an open-source, cross-platform, document-oriented database written in C++.
- MongoDB is an open-source document database
- MongoDB can handle dynamic data models and schema
- MongoDB has a rich query language
- If you have huge amount of data to be stored in tables, think of MongoDB before RDBMS databases. MongoDB has built-in solution for partitioning and sharding your database.
- Adding a new column in RDBMS is hard whereas MongoDB is schema-less. Adding a new field does not effect old documents and will be very easy.
- MongoDB allows a highly flexible and scalable document structure.

Features :

- ❖ MongoDB stores the main subject in the minimal number of documents and not by breaking it up into multiple relational structures like RDBMS.
- ❖ Without indexing, a database would have to scan every document of a collection to select those that match the query which would be inefficient. So, for efficient searching Indexing is a must and MongoDB uses it to process huge volumes of data in very less time.
- ❖ MongoDB increases the data availability with multiple copies of data on different servers. By providing redundancy, it protects the database from hardware failures. If one server goes down, the data can be retrieved easily from other active servers which also had the data stored on them.

Data Modelling :

- Design the schema according to user requirements.
- Do join on write operations, not on read operations.
- Combine objects that you want to use together into one document.
- Optimize your schema for more frequent use cases.
- Do complex aggregation in the schema.
- Replication provides redundancy and increases data availability.
- Data in MongoDB is schema-less, so there is no need to define a structure for the data before insertion.
- Data normalization refers to restructuring a database to improve data integrity and reduce data redundancy.
- Embedded data models are an example of a denormalized data model.

Advantages :

- MongoDB is a NoSQL database, which means it's cheaper and easier to maintain than other databases.
- MongoDB stores data in document format, which makes it flexible and able to handle large volumes of data.
- MongoDB is a document database in which one collection holds different documents. Number of fields, content and size of the document can differ from one document to another.
- Conversion/mapping of application objects to database objects not needed.
- Deep query-ability. MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL.
- MongoDB uses internal memory to store the working set, which enables faster access of data.
- Widely supported and code-native data access.
- It uses internal memory for storing working sets and this is the reason of its fast access.
- There may be difference between number of fields, content and size of the document from one to other.
- Its attributes allow an increase in data availability. It is also easy to access documents using indexing.
- MongoDB offers a simple query syntax that is much easier to grasp than SQL. It provides an expressive query language that users find helpful during development.

Conclusion

In conclusion, MEAN stack technologies offer a powerful and efficient framework for web application development. By combining MongoDB, Express.js, Angular, and Node.js, developers can create dynamic, responsive, and scalable applications that cater to modern user expectations. The MEAN stack's versatility, open-source nature, and widespread community support make it an excellent choice for both startups and established companies. With the ability to handle both server-side and client-side development, MEAN stack provides a unified platform that simplifies the development process. Its use of JavaScript across the stack streamlines communication and reduces the learning curve for developers. Furthermore, the support for real-time applications, cloud deployment, and strong performance capabilities further solidify the MEAN stack's status as a compelling technology stack in the web development world.

REFERENCES

- https://nodejs.org/en/learn
- https://angular.io/docs
- https://expressjs.com/en/starter/installing.html
- https://www.tutorialspoint.com/mongodb/mongodb_advantages.htm
- https://www.typescriptlang.org/docs/
- https://www.w3schools.com/angular/
- https://www.simplilearn.com/tutorials/angular-tutorial
- https://www.freecodecamp.org/news/learn-angular-full-course/
- https://www.w3schools.com/mongodb/
- https://www.codecademy.com/learn/learn-mongodb