

1) Explain routing in node js.

2) Explain package.json file with an example.

Ans:

- The package.json file is the **heart** of Node.js system.
- '**package.json**' is a file used in Node.js projects to define various properties of the project, such as **name**, **version**, **description**, **dependencies**, **scripts**, and **metadata**.
- It is a plain text file that contains a JSON object with **key-value** pairs that define various properties of the project.
- The '**package.json**' file can be created manually or using the **npm init** command in the terminal or command prompt.
- The '**package.json**' file is located in the root directory of a Node.js project.

```
{  
  "name": "my-app",  
  "version": "1.0.0",  
  "description": "My awesome Node.js app",  
  "main": "index.js",  
  "scripts": {  
    "start": "node index.js",  
    "test": "mocha tests/*.js"  
  },  
  "dependencies": {  
    "express": "^4.17.1",  
    "body-parser": "^1.19.0",  
    "mongoose": "^5.10.15"  
  },  
}
```

Each key-value pair in this example:

- **name:** The name of the project.

- **version:** The version of the project.
- **description:** A short description of the project.
- **main:** The entry point for the application.
- **scripts:** There are two scripts defined: start and test.
- **dependencies:** These are packages that are required for the application to run properly in production.

3) Explain CRUD operation in mongodb.

- CRUD stands for **Create, Read, Update, and Delete**.
- It is a set of basic operations that can be performed on any **persistent storage** system, including MongoDB.
- So these operations when performed on the database, help us to **view, search** or **modify** data and **delete** data in our database.

Explanation of each operation in MongoDB:

Create:

- **Create:** In MongoDB, the Create operation is used to **insert a new document** into a collection.
- **Syntax:**

```
db.collection.insertOne({name: "Kishan"})
```

- The **insertOne** method inserts a **single document** into the collection.
- There is also an **insertMany** method that can be used to **insert multiple documents** at once.

Read:

- **Read:** In MongoDB, the Read operation is used to **retrieve** one or more documents from a collection.
- **Syntax:**

```
db.collection.find({name: "Kishan"})
```

- **collection** is the name of the collection from which the documents should be retrieved.
- The **find method** returns a cursor that can be used to **match** over the results.

- There are also other methods, such as **findOne** and **count**, that can be used to **retrieve** a single document or the count of **matching documents**.

Update:

- **Update:** In MongoDB, the Update operation is used to **modify** one or more documents in a collection.
- **Syntax:**

```
db.collection.updateOne({name: 'Kishan'},
{$set: {'name': 'Kushal'}},
{upsert: true})
```

- **collection** is the name of the collection in which the documents should be updated.
- **update** is an update object that specifies the **modifications** to be made.
- The **updateMany** method updates the first document that **matches** the **filter**.

Delete:

- **Delete:** In MongoDB, the Delete operation is used to **remove** one or more documents from a collection.
- **Syntax:**

```
db.collection.deleteOne({name: 'Kushal'})
```

- **collection** is the name of the collection from which the documents should be removed.
- There is also a **deleteMany** method that can be used to **remove multiple** documents at once.

4) Explain the I/O cycle of nodejs.

- Node.js uses **non-blocking I/O**, the mechanism that allows you to have a **single thread** of execution **running your program**.
- It is **waiting** for I/O operations to **complete before moving on to the next operation**.
- Node.js can **continue to execute** other code while the I/O operation is being **processed**.

I/O cycle in Node.js:

- **Receive I/O request:** When Node.js receives an I/O request, such as **reading a file** or **making a network request**, it adds the request to its **event queue**.
- **Process pending events:** Node.js continuously processes events from its event **queue in a loop**, called the **event loop**.
- **Execute I/O operation:** When an I/O event is processed, Node.js will execute the associated I/O operation, such as **reading a file** from **disk** or **sending a network request**.
- **Return result or error:** Once the I/O operation is complete, Node.js will either return the **result of the operation** or an **error**, depending on whether the operation was **successful or not**.

This cycle is repeated for each I/O request that Node.js **receives**, allowing it to handle **multiple requests simultaneously without blocking the main thread**.

5)What is REST api.

- REST stands for Representational State Transfer.
- It's used to fetch or give some information from a web service.
- An application implementing a RESTful API will define one or more URL endpoints with a domain, port, path.
- In a RESTful API, resources are identified by URIs (Uniform Resource Identifiers), and the client can interact with these resources using standard HTTP methods such as GET, POST, PUT, DELETE, and PATCH.

Explanation of each method of a RESTful API:

- **GET:** Used to retrieve a representation of a resource.
- **POST:** Used to create a new resource.
- **PUT:** Used to update an existing resource or create a new resource if it doesn't exist.
- **DELETE:** Used to delete a resource.
- **PATCH:** Used to update a resource with a partial representation.

6)Write down the features of nodejs

- Node.js is an **open-source, cross-platform** JavaScript runtime environment.
- It allows developers to build **server-side application** in JavaScript, making it a popular choice for **building fast** and **scalable network applications**.

- Node.js uses the V8 JavaScript engine, **developed by Google**, which is also used in the **Chrome web browser**.
- Node.js applications are written in JavaScript, and can be **run** within the **Node.js runtime** on **OS X, Microsoft Windows, and Linux**.

Key features of Node.js:

- **Asynchronous I/O:** Node.js is built on a non-blocking I/O model, which allows it to **handle large numbers of concurrent connections** and perform I/O operations efficiently **without blocking the main thread**.
- **Very Fast:** Being built on **Google Chrome's V8 JavaScript Engine**, Node.js library is **very fast in code execution**.
- **Cross-platform:** Node.js can run on various operating systems, including **Windows, Linux, and macOS**.
- **Single threaded:** Node.js follows a **single threaded model** with **event looping**.
- **No Buffering:** Node.js applications **never buffer any data**. These applications simply output the data in **chunks**.
- **License:** Node.js is released under the MIT license.

7) Explain Flutter Framework architecture.

- Flutter is a **popular open-source mobile application** development framework that uses **Dart programming language**.
- The Flutter architecture is based on a **layered** and reactive design, which allows developers to build performant and **scalable mobile apps**.

Key components of the Flutter architecture:

- **Widgets:** Widgets are the building blocks of the Flutter UI. They are **lightweight**. **In Flutter, Everything is a widget**. Flutter provides a large set of pre-built widgets, **including buttons, text fields, lists, and more**.
- **Material Design:** Flutter provides two sets of widgets that implement the Material Design. This allows developers to create apps with a native look and feel on both Android and iOS platforms.
- **Rendering engine:** Flutter uses Skia, a **2D graphics rendering engine**, to render its UI. This allows for **fast and smooth animations and transitions**.
- **Platform channels:** Flutter allows for the adding of native features like **camera, geolocation, and sensors** into Flutter apps.

- **Dart runtime:** Flutter apps are **built using Dart programming language**, which is **compiled to native code** for iOS and Android platforms. Dart provides features like **garbage collection**, just-in-time (JIT) compilation, and ahead-of-time (AOT) compilation, **which improve the performance of Flutter apps**.

8)What is express framework? How to install express framework?

- Express is a popular web framework for Node.js that process of building web applications.
- It can be used to **design single-page, multi-page and hybrid web applications**.
- It provides a powerful set of features for **web and mobile applications** such as **routing, middleware, templates, and much more**.

To install Express framework, follow the steps below:

- Before installing, Express you need to download and install Node.js.
- If this is a brand new project, make sure to create a **package.json** first with the **npm init command**.
- **Installation is done using the npm install command:**

\$ npm install express

- **Start the server:**

\$ npm start

- **IF you can now run your Express application by running the following command in your terminal:**

node index.js

9)Explain Mongoddb Compass.

- MongoDB Compass is a **graphical user interface (GUI)** tool for working with MongoDB databases.
- Compass is **free to use** and source available, and can be **run on macOS, Windows, and Linux**.
- MongoDB Compass is a **powerful tool** for working with MongoDB databases, and can help developers have **more easily manage and analyze their data**.

- MongoDB Compass is a powerful GUI for **querying, aggregating, and analyzing** your MongoDB data in a **visual environment**.
- It perform operations like **filtering, and updating data, deleting data** without writing any code.

10)What is Flutter. What are the importance of Flutter.

- Flutter is an open-source mobile application development framework created by Google.
- It allows developers to build high-performance, visually attractive mobile applications for Android, iOS, and web platforms using a single codebase.
- Flutter uses the Dart programming language, which is also developed by Google.

key features and benefits of Flutter include:

- **Fast development:** Flutter provides a hot reload feature that allows developers to see changes in their app in real-time, making the development process faster and more efficient.
- **High performance:** Flutter's architecture is designed to deliver high performance, with features, fast startup times and reduced execution times.
- **Single codebase for multiple platforms:** Flutter allows developers to build apps for multiple platforms, such as Android, iOS, and web, using a single codebase, which reduces development time and cost.
- **Access to native features:** Flutter provides access to a wide range of native features, such as **camera, GPS, and storage**, allowing developers to build feature-rich and highly functional apps.

11)What is npm. Explain steps to install nodejs

- npm stands for "**Node Package Manager**".
- Open-source developers use npm to share software.
- npm is free to use.
- It is a package manager for the JavaScript programming language and is primarily used to install, share, and manage packages of code for Node.js.
- It is a popular runtime environment for running JavaScript on the server-side.

To install Node.js and npm steps:

- Go to the official Node.js website and download nodejs.
- Choose the any installer for your operating system and download it.
- Once the download is complete, run the installer.
- After installation, open your terminal or command prompt and type '**node -v**' to **check the version of Node.js** installed on your system. If you see the version number, it means that **Node.js is installed correctly**.

12) Explain angular file structure.

- Angular is a popular web application development framework that follows a specific file structure.
- The file structure of an Angular application is designed to help organize the code and make it easier to manage and maintain.

Angular file structure:

- **src:** This is the root directory of an Angular application. It contains all the source files for the application, including the HTML, CSS, TypeScript, and asset files.
- **app:** This is a subdirectory of the src directory and contains the main application module, components, services, pipes, and directives.
- **assets:** This directory contains static files such as images, fonts, and other resources that are used in the application.
- **environments:** This directory contains configuration files for different environments, such as development, production, and staging.
- **index.html:** This is the main HTML file for the application and is the entry point for the Angular application.
- **main.ts:** This is the main TypeScript file that bootstraps the application module.
- **styles.css:** This is the main stylesheet for the application and contains global styles that are applied to the entire application.
- **angular.json:** This is the configuration file for the Angular CLI (Command Line Interface) and contains settings for the application build process.
- **tsconfig.json:** This is the TypeScript configuration file that contains settings for the TypeScript compiler.
- **package.json:** This is the configuration file for the npm (Node Package Manager) and contains information about the dependencies and scripts for the application.

13) Explain the types of widgets with examples.

- In Flutter, widgets are the basic building blocks of the **user interface**. There are **two types** of widgets in Flutter:

Stateful Widget:

- **i) Stateful Widget:** Stateful widgets are those widgets that **have mutable state or data that can change over time**. These widgets are usually used for dynamic content that can be updated or changed by user interaction, such as **input fields, buttons, or animations**.

Examples of Stateful Widgets:

- **Text Field Widget:** Allows the user to input text.
- **Checkbox Widget:** Allows the user to select or deselect a checkbox.
- **Slider Widget:** Allows the user to select a value within a range by sliding a thumb along a track.
- **Progress Indicator Widget:** Shows the progress of a task.

Stateless Widget:

- **Stateless Widget:** Stateless widgets are those widgets that **have immutable state or data that never changes**. These widgets are usually used for static content that does not change over time, such as **text labels or icons**.

Examples of Stateless Widgets:

- **Text Widget:** Displays text on the screen.
- **Icon Widget:** Displays icons on the screen.
- **Image Widget:** Displays images on the screen.
- **Container Widget:** A basic rectangular widget that can be used to group other widgets together.

14) What do you mean by Single Page Application?

- A **Single Page Application (SPA)** is a web application that loads a **single HTML page** and dynamically content updates that page as the user interacts, **without requiring a page reload or navigation to a new page**.

- **SPA** is most use in many applicaions For example examples of single page applications like **Gmail, Calculator,Weather, Google Maps, Netflix, Pinterest, Paypal,etc.**
- In a SPA, a page refresh never occurs.