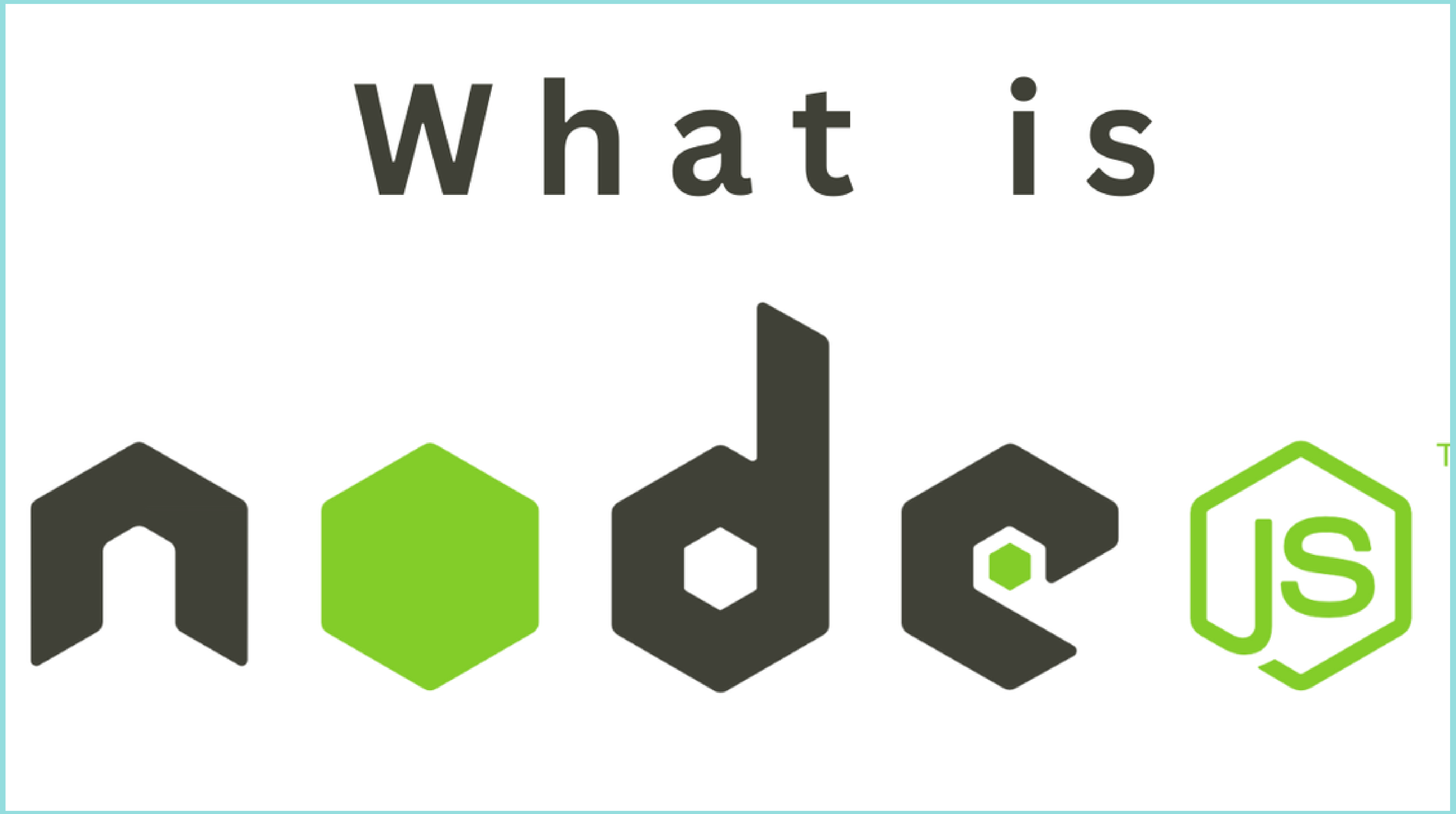
NODE.JS

Download the latest version of Node.js - https://nodejs.org/en/download

Date- 26/10/2023 Day- Thursday

Presented By- Mrunmayee Phadke

What is Node.js?

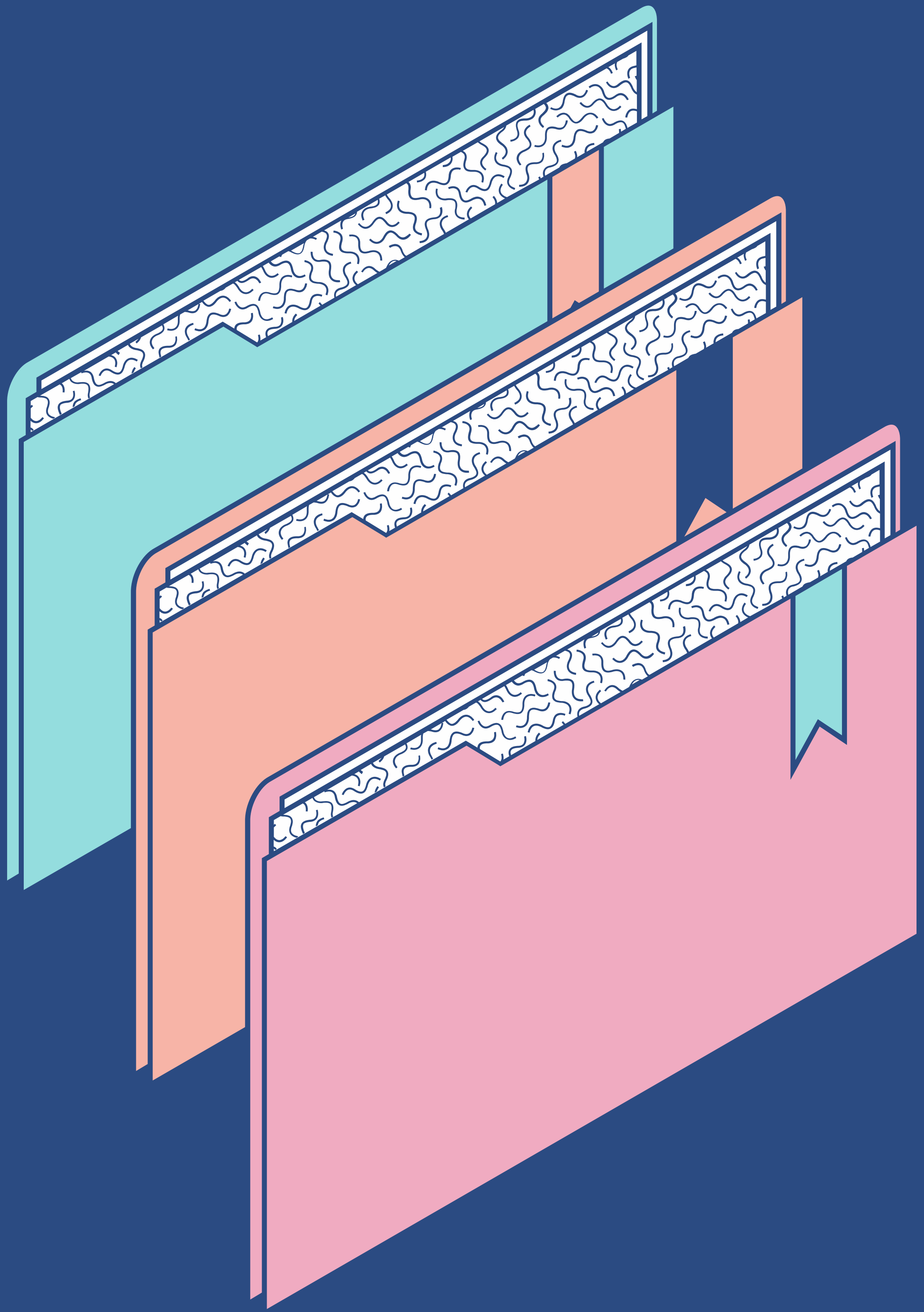


- JavaScript runtime: Node.js is an open-source JavaScript runtime built on Chrome's V8 JavaScript engine.

- Cross-platform: Node.js is designed to be cross-platform, allowing you to develop and deploy applications on various operating systems.

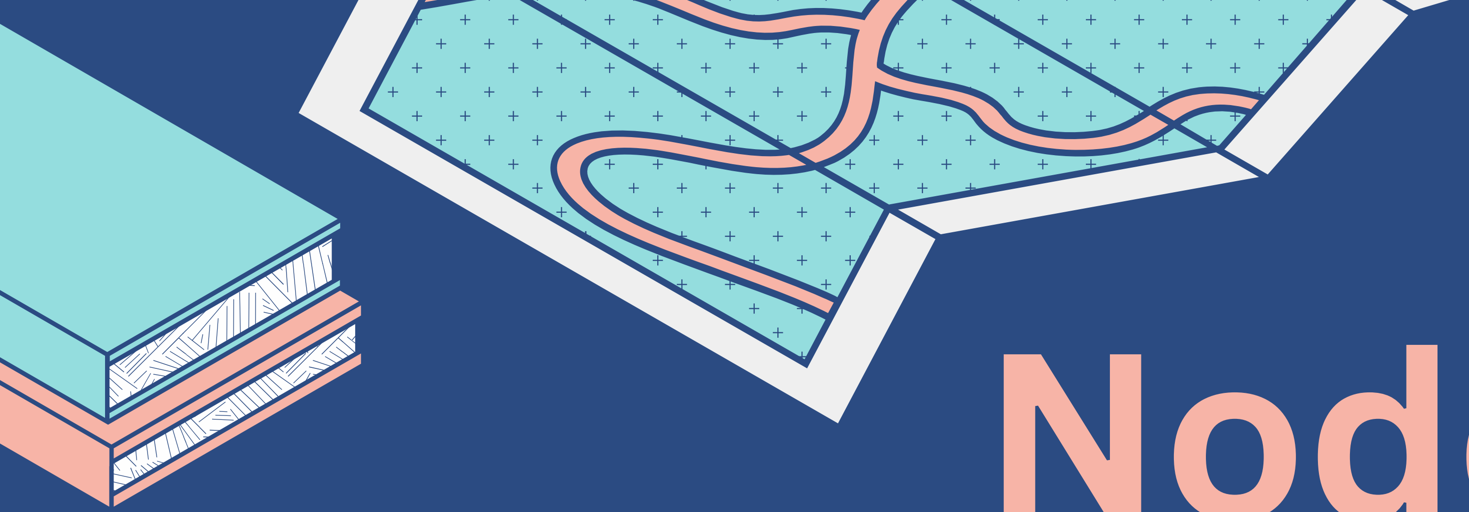
- Large and active community: Node.js has a thriving and supportive community of developers. This active community contributes to the growth of Node.js by creating libraries, frameworks, tools, and sharing best practices.

Downloading and Installing Node.js



Download the latest version of Node.js - https://nodejs.org/en/download

Verifying Installation using the terminal to check if Node.js is properly installed
Use command - node -v or node --version



Node.js Modules

Modules are like small units of code that can be reused and separated into individual files, making it easier to manage and maintain large applications.

- Modularity - Node.js promotes a modular code organization. Each module has specific functionality.
- Reusability - Modules can be reused across applications. Vital for efficient code development.
- Node.js provides a set of core modules that are built-in 'fs' (File System), 'http' (HTTP server/client functionality), and 'path' (file path manipulation)

Some Modules of Node.js

01 FS

The **fs** module provides file system-related operations, allowing you to read from and write to files, create directories, and perform other file system-related tasks.

02 http

The **http** module allows you to create HTTP servers and make HTTP requests. It provides the necessary functionality to interact with the HTTP protocol, handle requests, and send responses.

03 path

The **path** module provides utilities for working with file and directory paths. It helps in resolving and manipulating file paths, handling platform-specific path conventions, and more.

04 url

The **url** module provides utilities for working with URLs. It allows you to parse URLs, resolve relative URLs, and manipulate URL components like hostname, path, query parameters, etc.

NPM - Node Package Manager

- NPM which stands for Node Package Manager is a package manager for Node.js packages, or modules

- It is the default package manager for the JavaScript runtime environment Node.js.

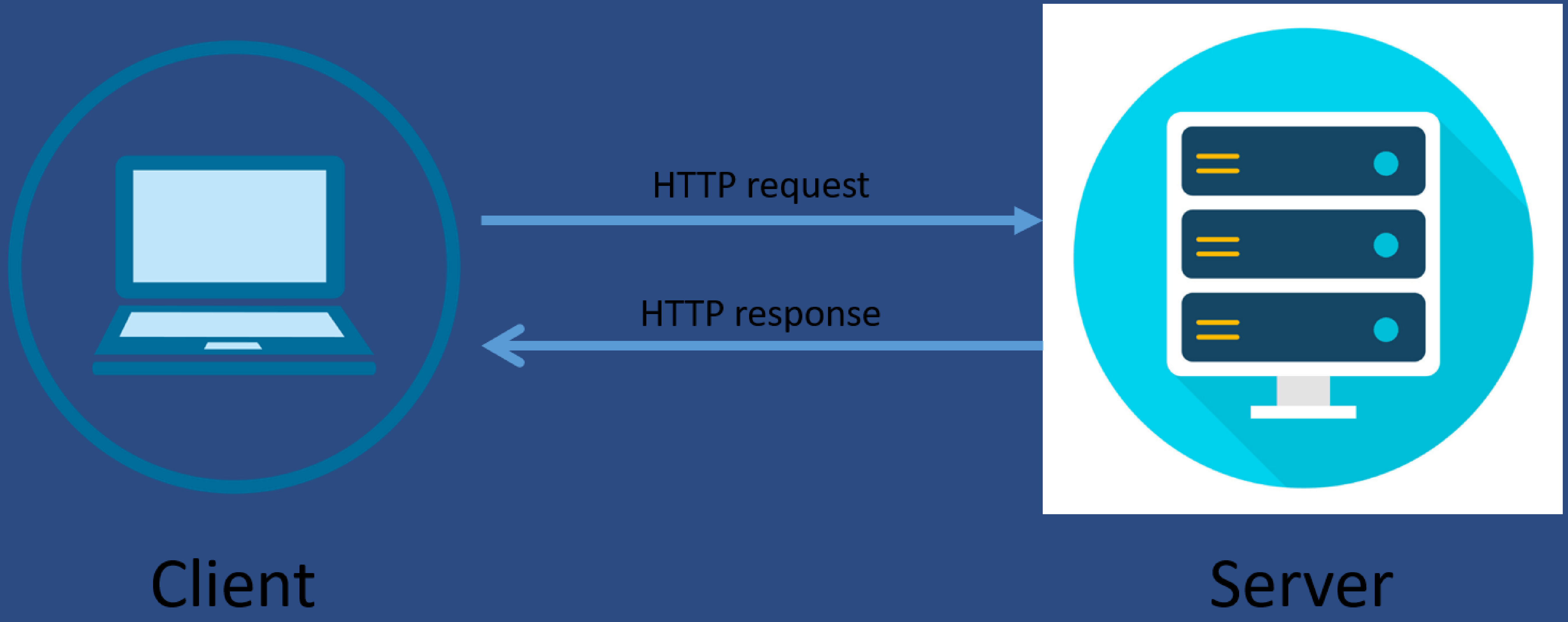
- It consists of a command line client, also called npm, and an online database of public and paid-for private packages, called the npm registry.

- Using that particular chunk handle(ID) the client can go to that specific chunkserver and read data from there.

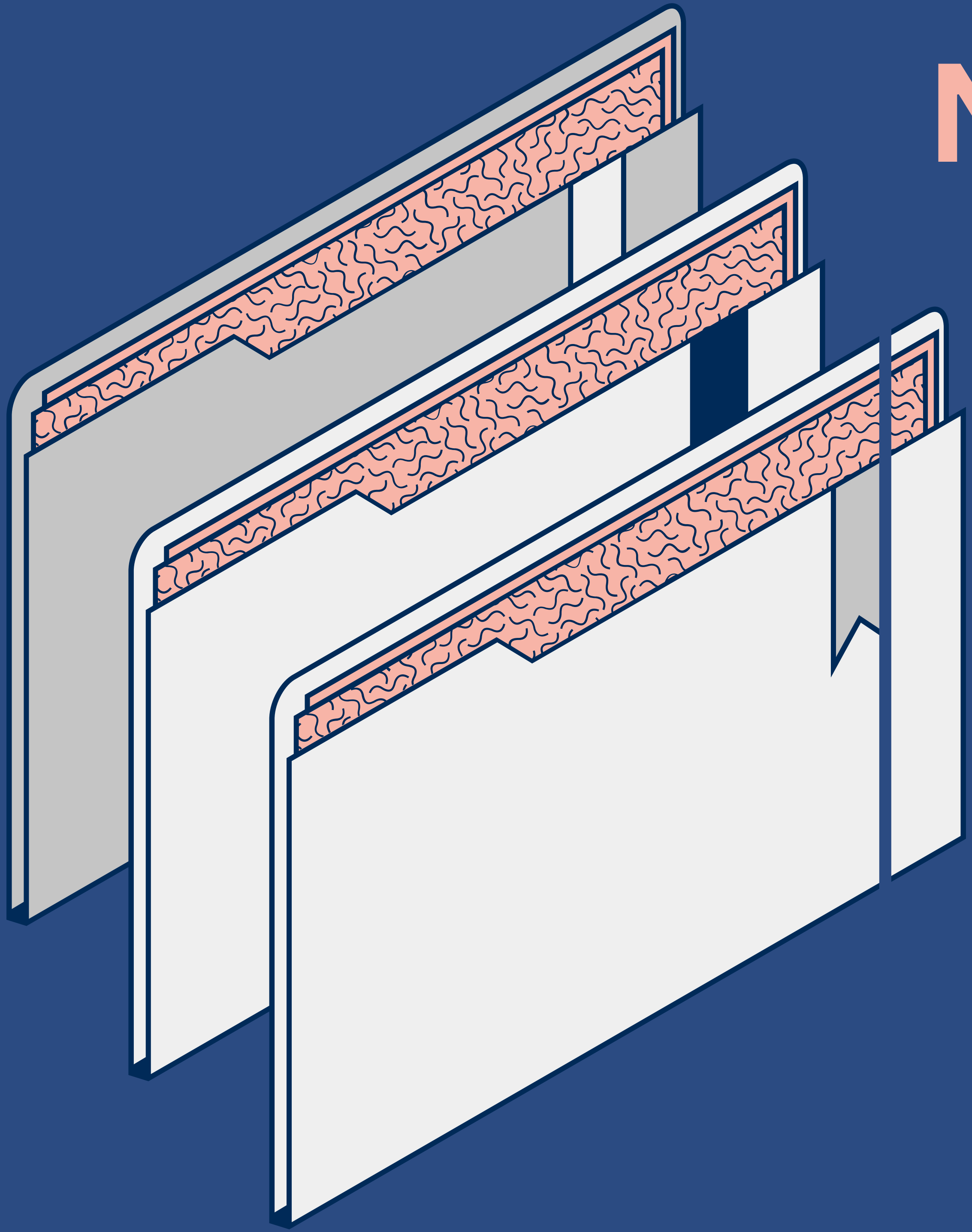
Some NPM commands -

- *npm init* :- Used to initialize a new Node.js project or package.
- *npm install <package name>* :- installs package locally.
- *npm update <package name> -g* :- Updates the package globally. Without "-g" it updates locally
- *npm uninstall <package name> -g* :- Uninstalls the previously downloaded package.
- *npm install <package name> --save* :- Another command to download a package.
- *npm help*- Gives a list of all commands that can be used.

Setting up a Basic Web Server



Working with the File System in Node.js



- fs' module allows us to read files synchronously and asynchronously.
- Synchronous reading: fs.readFileSync() blocks the event loop. Useful for simple scripts and command-line tools.
- Asynchronous reading: fs.readFile() is non-blocking, suitable for web servers and real-time applications

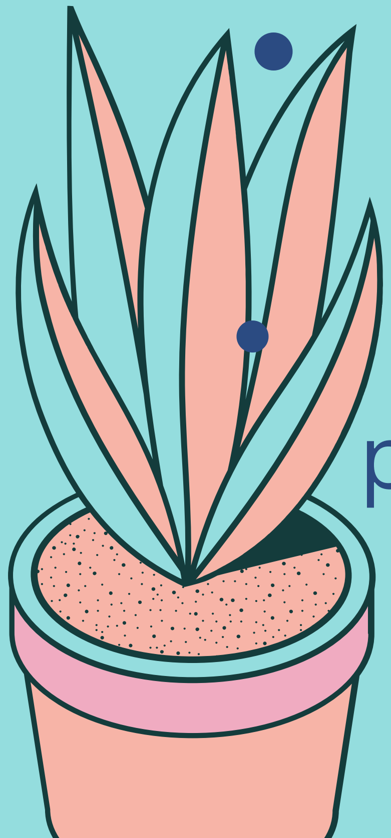
Express.js

- Express.js is a powerful web framework for building web applications and APIs.
- It simplifies server-side development by providing features like routing, middleware, and request/response handling.
- Create a simple RESTful API with Express.js.

Serving Static Resources in Express.js

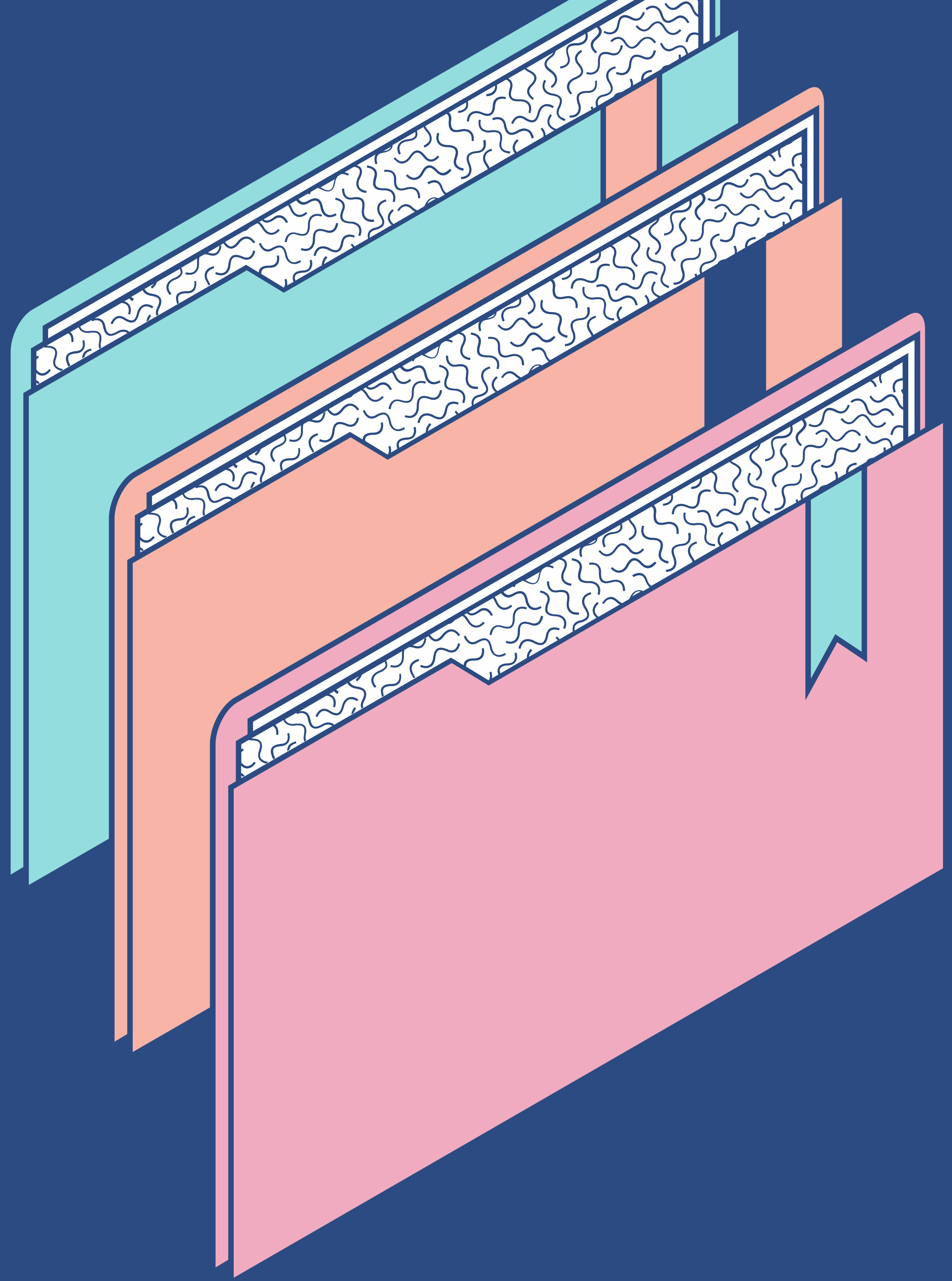
- Express.js provides a straightforward way to serve static files
- Static resources are: files that don't change on the server's end (e.g., HTML, CSS, JavaScript, images).

Why separating static and dynamic content is beneficial?
Faster load times: Browsers can cache static resources.
Easier maintenance: Separation makes code organization cleaner, Scalability.



- **app.use(express.static('public'));**

- express.static('public'): express.static is a built-in middleware function provided by Express.js for serving static files. In this code, it is configured to serve static files from the "public" directory.



DATABASE CONNECTIVITY

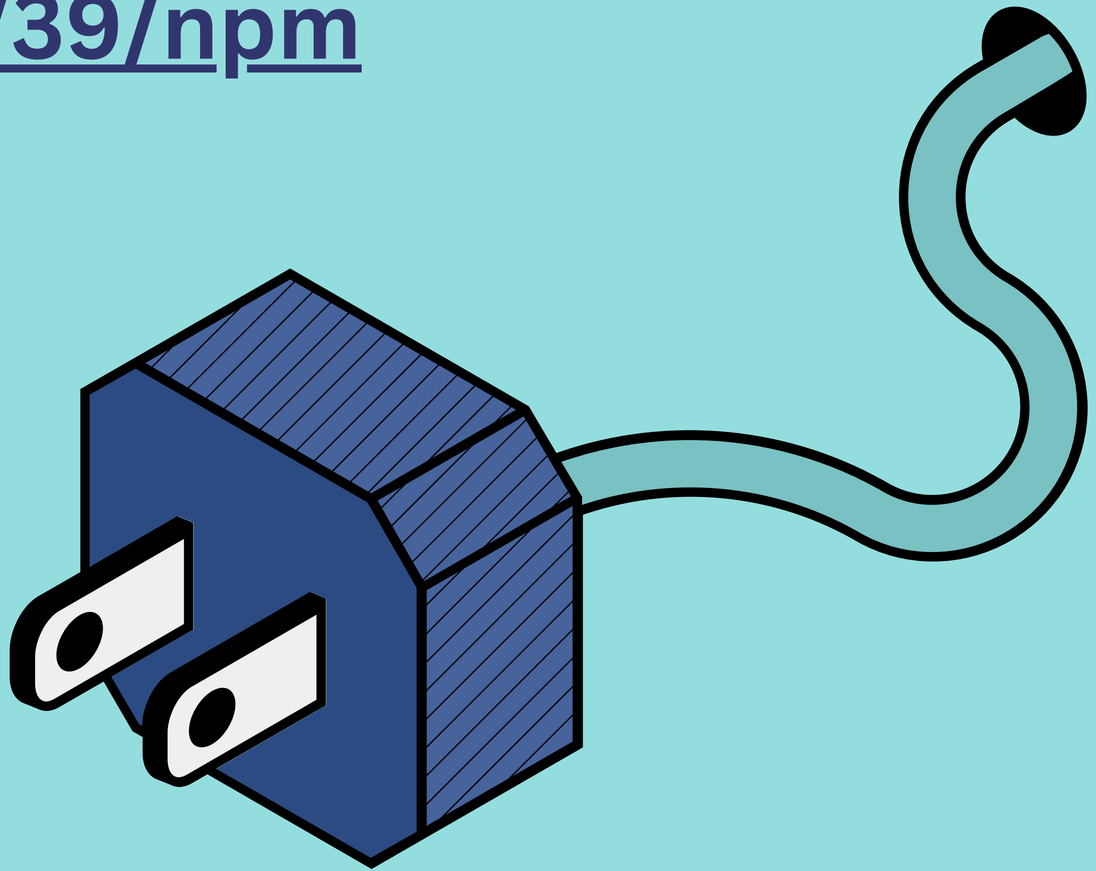
- Databases store and manage data critical for application functionality.
- There always a need for efficient and secure database connectivity.
- Types of databases: SQL (relational) and NoSQL (non-relational).
- Popular databases like MongoDB and SQL databases require specific libraries for connectivity.

REFERENCES

https://nodejs.org/en/knowledge/getting-started/npm/what-is-npm/

https://www.w3schools.com/nodejs/nodejs_npm.asp#:~:text=NPM%20is%20a%20package%20manager,when%20you%20install%20Node.js

https://www.bettercoder.io/job-interview-questions/c/39/npm



Thank You!

Happy Coding!