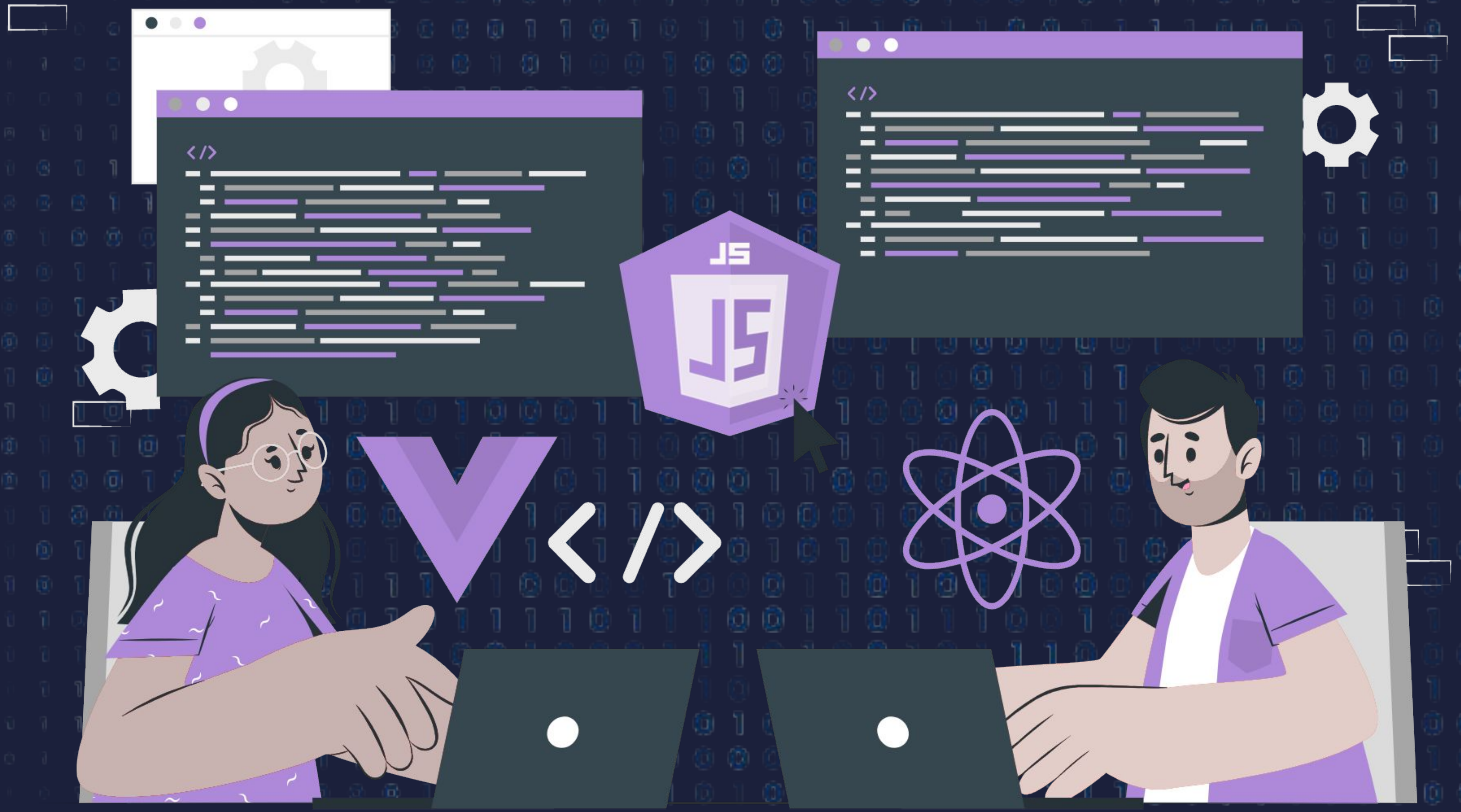
PW SKILLS

npm init and package.json



List of content:

1. npm
2. npm-init
3. Installing and uninstalling packages
4. Package.json
5. Creating package.json file

npm

- NPM (Node Package Manager) is a package manager for Node.js applications.
- It is a command-line utility that makes it easy to install, manage, and share packages of Node.js code.
- NPM is the default package manager for Node.js, and it allows developers to easily install and manage dependencies for their Node.js projects.
- NPM provides a central repository of packages, which developers can browse and search to find the packages they need. Packages can also be published to the repository, making it easy for other developers to discover and use them.

npm init

- The npm init command is used to initialize a new or existing npm package. It creates a package.json file that contains important information about the package, such as its name, version, dependencies, and other metadata.

  When an initializer is specified, it installs the corresponding npm package named

  create-<initializer> using npm-exec

Examples :

- To create a new React-based project using create-react-app:
- To generate a package.json file without any user input:
- To generate a plain old package.json using legacy init:
- To generate a package.json file without any user input:
- To create a new workspace within a project:
- To create a new React-based workspace within a project:

Installing packages in node

Uninstalling packages in node

package.json

- The package.json file serves as the centerpiece of the Node.js system and is considered the project's manifest file, containing crucial metadata information.
- Understanding and learning how to work with this file is a fundamental aspect of working with Node.js. It serves as the first step towards becoming proficient in Node.js development.
- To be more specific, the metadata information in the package.json file can be divided into two categories.
- The first category is identifying metadata properties, which includes details such as the project's name, current version, author, license, and project description.
- The second category is functional metadata properties, which includes values related to the project's functionality, such as the entry or starting point of the module, project dependencies, scripts being used, and repository links for the Node.js project.

package.json

package.json file can be created in two ways:

1. **Using npm init :** With this command, system expects user to fill the required information. It provides with default values which are editable by the user.

2. **Writing directly to file :** You can also directly write into file with all the required information and can include it in your Node project.

Example:

A demo package.json file with the required information.

SKILLS

THANK YOU