Multi Choice Type Question Michael is working on a large-scale React project with multiple subdirectories. He wants to organize his codebase by grouping related components together. Which directory structure option would be the most suitable for his project? Marks : 1 Negative Marks : 0 image Answer here Placing all components directly in the src directory Creating a separate directory for each component in the src directory Grouping components by feature or module in separate directories within the src directory Storing all components in the public directory

**ChatGPT said:**

The correct answer is:

**Grouping components by feature or module in separate directories within the src directory** ✅

**Explanation:**

* For large-scale React projects, organizing components by feature or module helps keep related files together, makes the project easier to navigate, and improves maintainability.
* Placing all components directly in src becomes messy as the number of components grows.
* Creating a separate directory for each component can be overkill unless components are very complex.
* Storing components in the public directory is incorrect because public is meant for static assets, not React components.

**You said:**

Multi Choice Type Question Olivia wants to understand the purpose of the node\_modules directory in a React project. What does it contain? Marks : 1 Negative Marks : 0 image Answer here It stores the project's source code. It holds configuration files for the build process. It contains external packages and dependencies It is used for routing and navigation

**ChatGPT said:**

The correct answer is:

**It contains external packages and dependencies** ✅

**Explanation:**

* The node\_modules directory stores all the third-party libraries and dependencies that your project requires, as specified in package.json.
* It does **not** contain your project’s source code, configuration files, or routing logic.

**You said:**

Multi Choice Type Question Michael wonders about the purpose of the .env file in a React project. What is its role? Marks : 1 Negative Marks : 0 image Answer here It stores environment variables used in the project It contains configuration settings for the build process It defines the routing structure of the application It specifies the project's dependencies

**ChatGPT said:**

The correct answer is:

**It stores environment variables used in the project** ✅

**Explanation:**

* The .env file is used to define environment-specific variables, such as API keys, endpoints, or feature flags, without hardcoding them into the source code.
* It does **not** define routing, dependencies, or general build configuration.

**You said:**

Multi Choice Type Question Emma is working on a Create React App project and wants to manage the app dependencies and scripts. Which file should Emma look into? Marks : 1 Negative Marks : 0 image Answer here package.json .gitignore node\_modules README.md

**ChatGPT said:**

The correct answer is:

**package.json** ✅

**Explanation:**

* package.json contains the list of project dependencies, scripts for running or building the app, and other metadata like project name and version.
* .gitignore specifies files to exclude from version control.
* node\_modules stores the installed dependencies, but does not manage them.
* README.md is for documentation.

**You said:**

Multi Choice Type Question David wants to import an external CSS file to apply styles to all components in his React project. Where should he import the CSS file to ensure it is applied globally? Marks : 1 Negative Marks : 0 image Answer here In the public/index.html file In the package.json file In the App.js file In the index.js file

**ChatGPT said:**

The correct answer is:

**In the index.js file** ✅

**Explanation:**

* Importing the CSS file in index.js ensures that the styles are applied globally to the entire React app.
* Importing in App.js would limit the scope to App and its children, which may work in some cases but is not truly global.
* public/index.html can include CSS via <link> tags, but using ES6 imports in index.js is the standard React approach.
* package.json does not handle CSS imports.

**You said:**

Multi Choice Type Question Emily is working on a React project and wants to locate the file that stores project-related metadata and dependencies. In which directory should she expect to find the package.json file? Marks : 1 Negative Marks : 0 image Answer here In the root directory of the project In the public directory where static assets are stored In the components directory where reusable UI components are stored In the src directory where the application code resides

**ChatGPT said:**

The correct answer is:

**In the root directory of the project** ✅

**Explanation:**

* package.json is always located in the root directory of a Node.js or React project.
* It stores project metadata, dependencies, and scripts for running or building the project.
* It is **not** found in public, components, or src directories.

**You said:**

Multi Choice Type Question Emily notices a directory named "build" in her React project. What is the purpose of this directory? Marks : 1 Negative Marks : 0 image Answer here It contains the final optimized and bundled version of the application It stores the development server configurations It holds the source code for the React components It is used for package management and dependency resolution

**ChatGPT said:**

The correct answer is:

**It contains the final optimized and bundled version of the application** ✅

**Explanation:**

* The build directory is generated when you run a production build (npm run build) in a React project.
* It contains optimized, minified, and bundled files ready for deployment.
* It does **not** store development configurations, source code, or manage dependencies.

**You said:**

Multi Choice Type Question David wonders about the purpose of the .babelrc file in a React project. What role does it play? Marks : 1 Negative Marks : 0 image Answer here It stores the project's dependencies It contains configuration settings for Babel It defines the project's build process It specifies the entry point of the React application

**ChatGPT said:**

The correct answer is:

**It contains configuration settings for Babel** ✅

**Explanation:**

* The .babelrc file is used to configure Babel, which transpiles modern JavaScript (ES6+) and JSX into code that can run in current browsers.
* It does **not** manage dependencies, define the build process, or specify the app’s entry point.

**You said:**

Multi Choice Type Question John wants to initialize a new React project with a basic package structure. Which command should he use? Marks : 1 Negative Marks : 0 image Answer here npm install react react-init my-app npm init react-app my-app create-react-app my-app

**ChatGPT said:**

The correct answer is:

**create-react-app my-app** ✅

**Explanation:**

* create-react-app is the official tool to quickly set up a new React project with a ready-to-use folder structure and configuration.
* npm install react only installs React in an existing project.
* react-init is not a standard command.
* npm init react-app my-app was used in older versions but is deprecated in favor of create-react-app.

**You said:**

Multi Choice Type Question Sophia is curious about the purpose of the package-lock.json file in a React project. What does it do? Marks : 1 Negative Marks : 0 image Answer here It stores the configuration settings for the project. It keeps track of the installed packages and their versions. It provides a summary of the project's dependencies It is used to specify the entry point of the React application

**ChatGPT said:**

The correct answer is:

**It keeps track of the installed packages and their versions** ✅

**Explanation:**

* package-lock.json ensures that the exact same versions of dependencies are installed across different environments, providing consistency and preventing version conflicts.
* It does **not** store general configuration settings, summarize dependencies, or define the entry point of the app.

**You said:**

Multi Choice Type Question David wants to add routing to his React project and is looking for a tool that simplifies the process. Which command should he use to install React Router in his project? Marks : 1 Negative Marks : 0 image Answer here npm add react-router-dom yarn install react-router create-react-app my-app --router npm install react-route-dom

**ChatGPT said:**

The correct answer is:

**npm install react-router-dom** ✅

**Explanation:**

* react-router-dom is the standard package for adding routing to React applications.
* npm add react-router-dom is not the correct npm syntax.
* yarn install react-router is incorrect because the package name is react-router-dom.
* create-react-app my-app --router does not exist; routing must be added after project creation.

**You said:**

Multi Choice Type Question You are tasked with building a dynamic user interface for a web application. Which of the following best describes React? Marks : 1 Negative Marks : 0 image Answer here A versatile programming language specifically designed for front-end development. A JavaScript library that provides a framework for server-side rendering. A powerful JavaScript library for building interactive and reusable user interfaces. A database management system that enables efficient data storage and retrieval

**ChatGPT said:**

The correct answer is:

**A powerful JavaScript library for building interactive and reusable user interfaces** ✅

**Explanation:**

* React is a JavaScript library focused on building dynamic and reusable UI components.
* It is **not** a programming language, a database system, or primarily a framework for server-side rendering (though it can be used with SSR via frameworks like Next.js).

**You said:**

Multi Choice Type Question Ram is planning to create a React project that utilizes TypeScript. Which template should Ram choose to set up the project easily? Marks : 1 Negative Marks : 0 image Answer here React-Typescript template TypeScript-React template Create React App TypeScript template React with TypeScript template

**ChatGPT said:**

The correct answer is:

**Create React App TypeScript template** ✅

**Explanation:**

* When using Create React App, you can set up a TypeScript project with the command:  
  npx create-react-app my-app --template typescript
* This template includes all necessary TypeScript configurations and typings.
* Other options listed are not standard or recognized templates.

**You said:**

Multi Choice Type Question Sophia is developing a React app using Create React App. Where should she primarily work and find the source code for her app? Marks : 1 Negative Marks : 0 image Answer here README.md node\_modules package.json src

**ChatGPT said:**

The correct answer is:

**src** ✅

**Explanation:**

* The src directory contains all the application’s source code, including components, styles, and other JavaScript/TypeScript files.
* README.md is for project documentation.
* node\_modules stores installed dependencies.
* package.json holds metadata, dependencies, and scripts, but not the source code itself.

**You said:**

Multi Choice Type Question Liam needs to store static assets such as images, svgs, and fonts for his React app in a Create React App project. Which folder should Liam utilize? Marks : 1 Negative Marks : 0 image Answer here README.md node\_modules package.json public

**ChatGPT said:**

The correct answer is:

**public** ✅

**Explanation:**

* The public folder is intended for static assets like images, fonts, and icons that need to be served directly without being processed by the build system.
* README.md is for documentation.
* node\_modules contains installed dependencies.
* package.json stores project metadata and scripts, not assets.

**You said:**

Multi Choice Type Question David prefers using npm for managing packages in his React projects. Which command should he use to install the Create React App globally on his machine? Marks : 1 Negative Marks : 0 image Answer here npm install -g create-react-app npm add create-react-app -g npm create-react-app -g npm global install create-react-app

**ChatGPT said:**

The correct answer is:

**npm install -g create-react-app** ✅

**Explanation:**

* The -g flag installs the package globally, allowing the create-react-app command to be used anywhere on the machine.
* Other options listed are not valid npm commands for global installation.

**You said:**

Multi Choice Type Question Lisa wants to start her React project using Create React App. Which command should she run in the terminal to start the project? Marks : 1 Negative Marks : 0 image Answer here npm run npm build npm start npm run dev

**ChatGPT said:**

The correct answer is:

**npm start** ✅

**Explanation:**

* In a Create React App project, npm start launches the development server and opens the app in the browser.
* npm run by itself does nothing without a specific script.
* npm build is incorrect; the correct command for production build is npm run build.
* npm run dev is not a default script in Create React App projects.

**You said:**

Multi Choice Type Question Ram wants to create a new React project using Create React App. He wants to name the project "my-website". Which command should Ram use? Marks : 1 Negative Marks : 0 image Answer here npm create-react-app my-website create-react-app my-website npx create-react-app my-website npx react-create-app my-website

**ChatGPT said:**

The correct answer is:

**npx create-react-app my-website** ✅

**Explanation:**

* npx create-react-app my-website is the recommended way to create a new React project without needing a global installation of create-react-app.
* create-react-app my-website requires a global installation.
* npm create-react-app my-website and npx react-create-app my-website are not valid commands.