The command, « npm start », by default ...

- ☐ installs all dependencies listed in the 'package.json' settings file of an NPM project
- [✓ executes the main script file of an NPM project mentioned in the 'package.json' settings file]
- ☐ runs unit tests on an NPM project
- ☐ sets up a Node.js project to start using NPM as the package manager

The Postman app is used to...

- ○ Access and log into a database and fetch the data.
- ○ Perform unit tests on the back-end code.
- ○ Log all incoming HTTP(S) requests to the back-end server's command-line.
- [✓ Streamline the issuing of HTTP(S) requests to back-end routes during development.]

Calling the next() function within a middleware function invocation will immediately...

- ☐ Send a response to the client
- ☐ Invoke the next middleware function
- [✓ Invoke the next route function, if no subsequent middleware functions exist]
- ☐ Use the 3rd-party middleware module named 'next' to log the incoming request data to the command-line in nicely-formatted color-coded output.

Middleware functions in Express.js are invoked...

- [✓ Before route functions are invoked]
- ○ After route functions are invoked

Which of the following are functions of the 'res' object passed to a route function? Look it up!

- [✓ cookie()]
- [✓ download()]
- [✓ json()]
- [✓ location()]
- [✓ redirect()]
- [✓ render()]
- [✓ send()]
- [✓ sendFile()]
- [✓ set()]
- [✓ status()]
- [✓ type()]

It is possible for Express to serve static files from a particular directory without needing to make routes for each file.

[✓ True]

False

Which of the following are true?

- [✓ The 3rd-party 'multer' middleware module is useful for streamlining the upload of file data passed to the back-end server via HTTP(S) POST requests.]
- [✓ The 3rd-party 'bodyParser' middleware module is useful for parsing data passed to the server along with HTTP(S) POST requests.]
- [✓ The 3rd-party 'axios' module is useful for streamlining how asynchronous HTTP requests are made from the back-end server to other external servers or APIs.]
- [✓ The 3rd-party 'dotenv' module is useful for loading environmental variables into the back-end, so sensitive credentials and global settings do not need to be hard-coded in the main back-end code files.]
- [✓ The 3rd-party 'passport' middleware module is useful for streamlining user authentication.]
- [✓ The 3rd-party 'mongoose' module is useful for simplifying how express.js-based code interacts with a MongoDB database.]
- [✓ The 3rd-party 'mocha' and 'chai' modules are useful for running unit tests on back-end code.]

The configuration file in Express.js-based apps is called...

- .config
- express.yml
- [✓ package.json]
- requirements.txt
- .project

The 3rd-party middleware, body-parser, makes all data sent to the server with an incoming HTTP(S) request into an easily accessible object called...

- [✓ req.body]
- req.bodyParser
- req.files
- req.params

Express-based projects should have a directory named 'node_modules' where the code of all dependencies is tracked by version control.

- [✓ True]
- False

When a client uploads a file to a server, it is best practice to ...

- Leave the filename as-is and simply save it to the server.
- [✓ Modify the filename to ensure its uniqueness - often by appending the time/date or a random number - before saving it to the server.]

The command, « npm install express --save »...

- ☐ Builds and executes the NPM project
- ☐ Installs the 3rd-party module named 'save'
- ☐ Installs all dependencies listed in the file named 'package.json' onto the local machine
- ✓ Installs 'express' onto the local machine
- ✓ Updates the file named 'package.json' to include 'express' as a dependency
- ☐ Updates the file named 'package.json' to include 'save' as a dependency

The command, « npm install » ...

- ☐ Builds and executes the NPM project
- ☐ Installs the 3rd-party module named 'save'
- ☑ Installs all dependencies listed in the file named 'package.json' onto the local machine
- ☐ Installs 'express' onto the local machine
- ☐ Updates the file named 'package.json' to include 'express' as a dependency
- ☐ Updates the file named 'package.json' to include 'save' as a dependency

Which type of incoming HTTP request(s) CANNOT be handled by an Express-based project and is better handled by a different web server framework?

- ☐ a request for a URL that does not match the name of an actual file or directory on the server
- ☐ A request with a parameter in the URL, such as the ID number of a record in a database or API
- ☐ a request that results in a plain text response
- ☐ a request that results in an HTML document response
- ☐ a request that results in a JSON object response
- ☐ A request that results in a JSON array response
- ☐ A request that results in an image response
- ☐ HTTP POST requests
- ☐ HTTP GET requests
- ☐ A request to upload one file
- ☐ A request to upload several files
- ☐ A request that results in the same response each time the same type of request is made
- ☐ A request that results in a different response each time the same type of request is made
- ☑ None - all of the above can be handled by a project using Express

The purpose of middleware functions in Express.js is to...

- [ ] ensure that the functionality of the prototype matches the functionality of the back-end production code
- [✓] modify the req or res objects that will eventually be passed to a route function
- [ ] update the user interface in response to a request
- [✓] perform some sort of 'side-effect' in response to a request that is not directly related to constructing the response

Which of the following are true?

- [✓] Express's 'use()' function is used to pass a middleware function for Express to call with each relevant incoming HTTP(S) request.
- [✓] Express's 'post()' function is used to pass a route function for Express to call with each relevant incoming HTTP(S) POST request.
- [✓] Express's 'get()' function is used to pass a route function for Express to call with each relevant incoming HTTP(S) GET request.
- [✓] Express's 'listen()' function is used to begin listening to a specific port for all incoming HTTP(S) requests.

The command, « npm init »...

- ○ Automatically installs all necessary 3rd-party modules for developing a back-end application.
- [✓ Executes a wizard that, when complete, generates the starting version of the package.json settings file.]
- ○ Initializes a github repository for the given Node.js project.
- ○ Informs Node.js's package manager that you wish to use the Express.js back-end framework in the code.

Express.js is a ...

- ○ front-end framework built in Node.js
- ○ front-end framework built in React.js
- ○ front-end framework built in browser-based Javascript
- [✓ back-end framework built in Node.js]
- ○ back-end framework built in React.js
- ○ back-end framework built in browser-based Javascript

Code in Express.js-based apps is written in ...

- ○ Browser-based Javascript
- [✓ Node.js-based Javascript]
- ○ React.js-flavored Javascript

The module named 'nodemon' saves developers time by ...

- ○ mocking external APIS so the server doesn't have to wait for responses
- [✓ stopping and restarting the server every time the code is changed]
- ○ monitors changes to a project's dependencies, and automatically installs and imports them as necessary
- ○ monitoring the server and automatically informing the developer when it is overloaded with requests
- ○ monitoring incoming HTTP(S) requests and automatically routing them to the appropriate code