**Javascript and their frameworks:**

Download node js.

Download nvm and npm setup and install it,

nvm –v (gives the version of nvm )

npm –v(gives the version of npm)

nvm install <version> (to install a version of nodejs)

nvm list(shows versions of nodejs installed)

nvm use <version>(to use a version of nodejs that u have installed)

**Electron Framework setup:**

npm install -g electron-forge (in c:users/vatsalnds directory) required only once

Now to make an electron app :

In cmd/terminal type :

electron-forge init myfirstapp

cd myfirstapp

(Now you can Open myfirstapp folder with VisualStudio now)

npm start (in terminal of visual studio or directly in cmd)

**Express Framework Setup:**

**Method 1:**

The -g switch installs the Express Generator globally on your machine so you can run it from anywhere.

Install the Express Generator by running the following from a terminal:

**npm install -g express-generator** (required once only)

We can now scaffold a new Express application called myExpressApp by running:

**express myExpressApp --view pug**

This creates a new folder called myExpressApp with the contents of your application. The --view pug parameters tell the generator to use the [pug](https://pugjs.org/api/getting-started.html) template engine.

To install all of the application's dependencies (again shipped as npm modules), go to the new folder and execute npm install:

**cd myExpressApp**

**npm install**

(Now you can Open myExpressApp folder with VisualStudio now)

At this point, we should test that our application runs. The generated Express application has a package.json file which includes a start script to run node ./bin/www. This will start the Node.js application running.

From a terminal in the Express application folder, run:

**npm start**

 To make our development process a lot easier, we will install a tool from npm, nodemon. This tool restarts our server as soon as we make a change in any of our files, otherwise we need to restart the server manually after each file modification. To install nodemon, use the following comman

npm install -g nodemon

**Method 2 (for not using pug template):**

The -g switch installs the Express Generator globally on your machine so you can run it from anywhere.

Install the Express Generator by running the following from a terminal:

**npm install -g express-generator** (required once only)

open cmd :

md myfirstexpressapp

cd myfirstexpressapp

npm init

npm install express –save

now open in visual studio

create a file named app.js or index.js whichever name suits you, I created app.js and write basic code

var express = require("express");

var app = express();

app.listen(3000, () => {

 console.log("Server running on port 3000");

});

In package.json do changes:

"main": "app.js",

"scripts": {

    "start": "node app.js",

    "test": "echo \"Error: no test specified\" && exit 1"

  }

To make our development process a lot easier, we will install a tool from npm, nodemon. This tool restarts our server as soon as we make a change in any of our files, otherwise we need to restart the server manually after each file modification. To install nodemon, use the following comman

npm install -g nodemon

**React Js Setup:**

You can now create a new React application by typing:

npx create-react-app my-app

where my-app is the name of the folder for your application. This may take a few minutes to create the React application and install its dependencies.

**Note**: If you've previously installed create-react-app globally via npm install -g create-react-app, we recommend you uninstall the package using npm uninstall -g create-react-app to ensure that npx always uses the latest version.

Let's quickly run our React application by navigating to the new folder and typing npm start to start the web server and open the application in a browser:

cd my-appnpm start

You should see the React logo and a link to "Learn React" on [http://localhost:3000](http://localhost:3000/) in your browser. We'll leave the web server running while we look at the application with VS Code.

To open your React application in VS Code, open another terminal or command prompt window, navigate to the my-app folder and type code .:

cd my-appcode .

**Angular js Setup:**

**https://code.visualstudio.com/docs/nodejs/angular-tutorial**