**Om Anant Khairnar**

**2019BTECS00078**

**SET Lab**

**Assingment N0 2**

**Topic : node.js**

For every Frameworks/IDEs/Softwares given above provide the answers for below questions

1. Original author

🡪

Node.js was written initially by [Ryan Dahl](https://en.wikipedia.org/wiki/Ryan_Dahl) in 2009,[[24]](https://en.wikipedia.org/wiki/Node.js#cite_note-training.com-24) about thirteen years after the introduction of the first

server-side JavaScript environment, [Netscape's](https://en.wikipedia.org/wiki/Netscape) LiveWire Pro Web.[[25]](https://en.wikipedia.org/wiki/Node.js#cite_note-25) The initial release supported only

Linux and Mac OS X. Its development and maintenance was led by Dahl and later sponsored

by [Joyent](https://en.wikipedia.org/wiki/Joyent" \o "Joyent).[[26]](https://en.wikipedia.org/wiki/Node.js#cite_note-Google_Groups_post_by_Ryan_Dahl_about_Joyent-26)

1. Developers

🡪

Developer of Node.js ius **OpenJS Foundation**

Node. js developers **deploy and maintain network applications**.

1. Initial release

🡪 May 27, 2009

1. Stable release

🡪 January 18, 2022

1. Preview release

🡪

Version LTS Date V8 npm NODE\_MODULE\_VERSION[1]

Node.js 17.5.0 2022-02-10 9.6.180.15 8.4.1 102

Node.js 17.4.0 2022-01-18 9.6.180.15 8.3.1 102

Node.js 17.3.1 2022-01-10 9.6.180.15 8.3.0 102

Node.js 17.3.0 2021-12-17 9.6.180.15 8.3.0 102

Node.js 17.2.0 2021-11-30 9.6.180.14 8.1.4 102

Node.js 17.1.0 2021-11-09 9.5.172.25 8.1.2 102

Node.js 17.0.1 2021-10-20 9.5.172.21 8.1.0 102

Node.js 17.0.0 2021-10-19 9.5.172.21 8.1.0 102

Node.js 16.14.0 Gallium 2022-02-08 9.4.146.24 8.3.1 93

Node.js 16.13.2 Gallium 2022-01-10 9.4.146.24 8.1.2 93

1. Repository (with cloud support )

🡪

github.com/nodejs/node

1. Written in (Languages)

🡪

**JavaScript is the only language that** Node.js supports natively, but many compile-to-JS

languages are available. As a result, Node.js applications can be written in CoffeeScript,

Dart, TypeScript, ClojureScript and others. Node.js is primarily used to build network

programs such as Web servers.

1. Operating System support

🡪

Node.js is officially supported on **Linux, macOS and Microsoft Windows 8.1 and Server**

**2012** (and later), with tier 2 support for SmartOS and IBM AIX and experimental support for

FreeBSD. OpenBSD also works, and LTS versions available for IBM i (AS/400).c

1. Platform ,portability

🡪

Node. js is an **open-source and cross-platform runtime environment** for executing JavaScript

code outside a browser. You need to remember that NodeJS is not a framework and it's not a

programming language

1. Available in (Total languages)

🡪

JavaScript is the only language that Node.js supports natively

1. List of languages supported

🡪

JavaScript is the only language that Node.js supports natively, but many compile-to-JS languages

are available. As a result, Node.js applications can be written in **CoffeeScript, Dart, TypeScript,**

**ClojureScript and others**.

1. Type (Programming tool, integrated development environment etc.)
2. Website

🡪

https://nodejs.org/en/

1. Features

🡪

**Features of Node.** **js**

* Asynchronous and Event Driven − All APIs of Node. js library are asynchronous, that is, non-blocking. ...
* Very Fast − Being built on Google Chrome's V8 JavaScript Engine, Node. js library is very fast in code execution.
* Single Threaded but Highly Scalable − Node. ...
* No Buffering − Node. ...
* License − Node.

1. Size (in MB, GB etc.)

🡪

If you were to switch to the full version of the “node:6.9. 5” image, however, you would save

approximately **550MB** of drive space by not duplicating the build tool installation. Yes, you need to

have one copy of the full image and all of it's layers, taking up 250MB of space when you build from

it

1. Privacy and Security

🡪

One way to enforce security on your data, server, and code is to **use HTTPS.** By default, Node. js sends content over HTTP. Using the HTTPS module, you can force communication to be done over a secure channel with the client.

**Node.** **js Security Practices**

* Use parameterized inputs to prevent injection attacks.
* Use multi-factor authentication to prevent automated attacks.
* Discard sensitive data after use.
* Patch old XML processors.
* Enforce access control on every request.
* Create fluid build pipelines for security patches.

1. Type of software (Open source/License)

🡪

**JavaScript runtime environment**

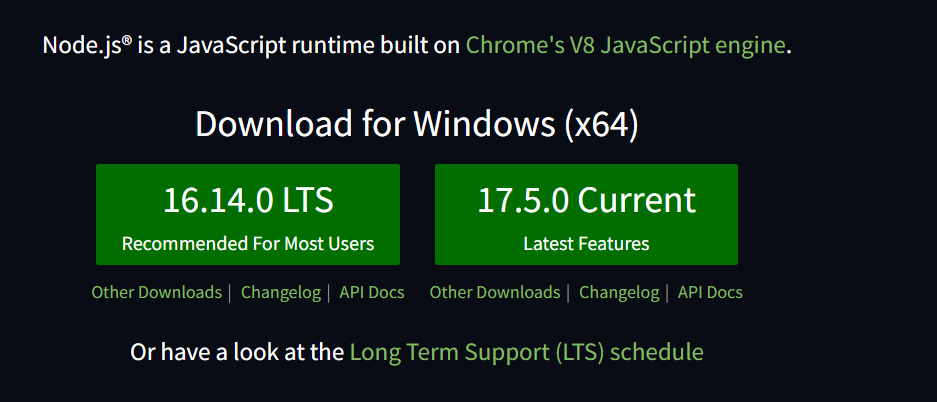
Node.js is an **open-source, cross-platform, back-end JavaScript runtime environment** that runs on the V8 engine and executes JavaScript code outside a web browser.

1. If License- Provide details.

🡪

1. Latest version

🡪



1. Cloud support (Yes/No)

🡪 yes

Google Cloud **lets you choose the best environment to run your Node**. js applications, with options for serverless, Kubernetes, VMs, or custom hardware

1. Applicability

🡪

Node. js is **primarily used for non-blocking, event-driven servers**, due to its single-threaded nature. It's used for traditional web sites and back-end API services, but was designed with real-time, push-based architectures in mind

1. Drawbacks (if any)

🡪

* Reduces performance when handling Heavy Computing Tasks.
* Node.js invites a lot of code changes due to Unstable API.
* Node.js Asynchronous Programming Model makes it difficult to maintain code.
* Choose Wisely – Lack of Library Support can Endanger your Code.

1.Implement linear regression problem using Google colab (Perform preprocessing, training and testing)

Dataset 2- <https://archive.ics.uci.edu/ml/datasets/Air+Quality>

**Collab Link :**

https://colab.research.google.com/notebooks/intro.ipynb?authuser=1#scrollTo=5TCFvhdXwo&line=3&uniqifier=1

**Code**

import matplotlib.pyplot as plt

import numpy as np

from sklearn import linear\_model

import pandas as pd

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.model\_selection import train\_test\_split

from google.colab import files

uploaded = files.upload()

df = pd.read\_csv('AirQualityUCI.csv')

df.head()

x=df.iloc[:,2].values

y=df.iloc[:,3].values

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 1/3,

random\_state = 0)

x\_test=x\_test.reshape(-1,1)

x\_train=x\_train.reshape(-1,1)

lin\_reg=linear\_model.LinearRegression()

lin\_reg.fit(x\_train,y\_train)

lin\_reg\_pred=lin\_reg.predict(x\_test)

print("Coefficients:\n",lin\_reg.coef\_)

print("Intercept:\n",lin\_reg.intercept\_)

print("Mean squared error: %.2f"

% mean\_squared\_error(y\_test, lin\_reg\_pred))

# Explained variance score: 1 is perfect prediction

print('Variance score: %.2f' % r2\_score(y\_test, lin\_reg\_pred))

plt.scatter(x\_test, y\_test, color = 'red')

plt.plot(x\_test, lin\_reg\_pred, color = 'blue')

plt.title('Temperature vs Humidity(Test set)')

plt.xlabel('Temperature')

plt.ylabel('Relative Humidity')

plt.show()

result :

