

### Practical No. 3

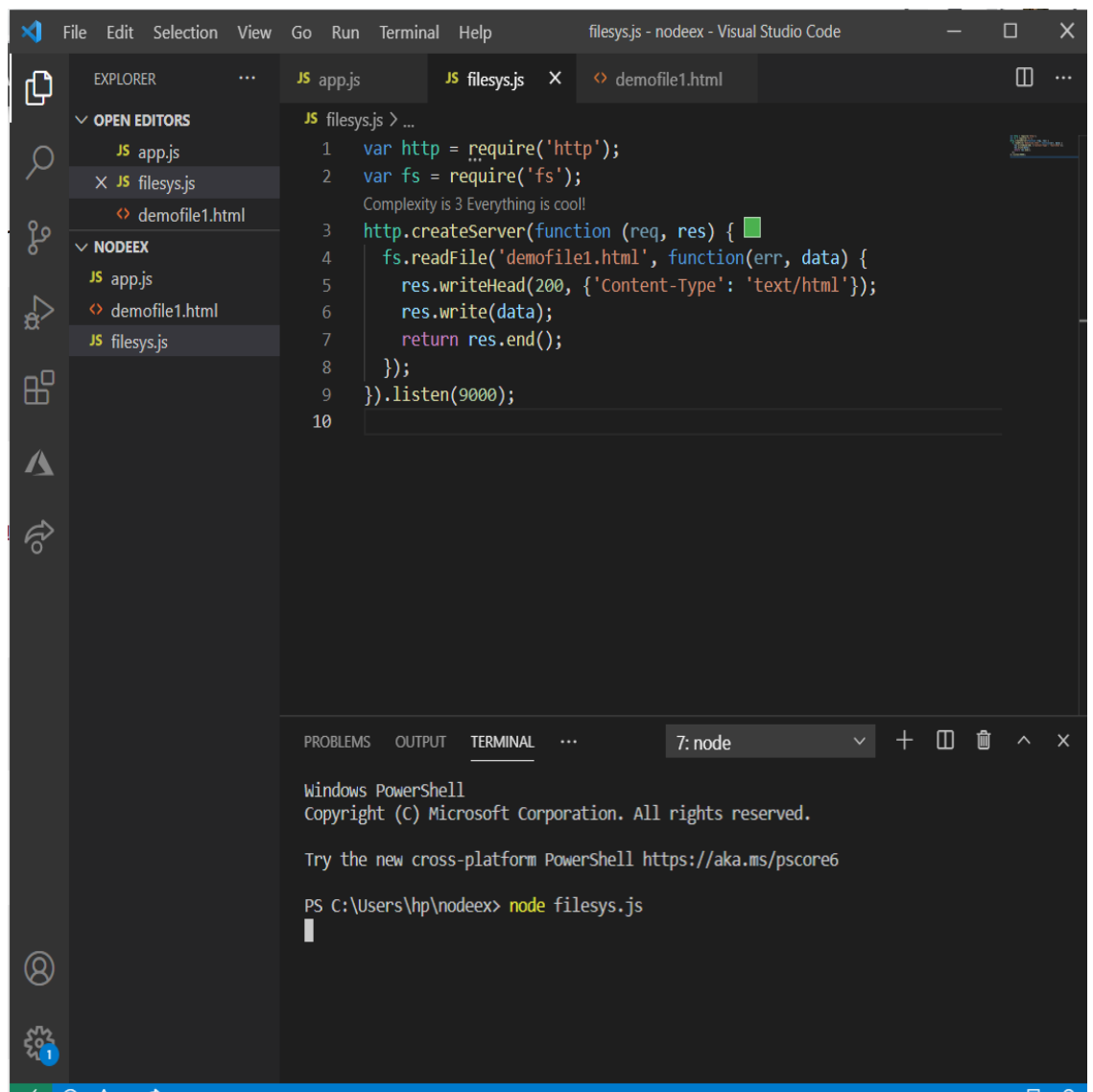
#### To use pre-built and user defined modules in Node.js Application.

NAME:POOJA VISHNU SHINDE

PRN NO: 2018BTECS00042

#### Problem Statement 1:

1. With the help of suitable examples use following Node.js pre-built modules in Node.js Application.
  - a. File System Module (fs)



The screenshot displays the Visual Studio Code interface with a project named 'nodeex'. The Explorer sidebar on the left shows the file structure with 'app.js', 'filessys.js', and 'demofile1.html'. The main editor window is open to 'filessys.js', which contains the following JavaScript code:

```
1 var http = require('http');
2 var fs = require('fs');
3
4 http.createServer(function (req, res) {
5     fs.readFile('demofile1.html', function(err, data) {
6         res.writeHead(200, {'Content-Type': 'text/html'});
7         res.write(data);
8         return res.end();
9     });
10 }).listen(9000);
```

Below the editor, the integrated terminal shows the command prompt for Windows PowerShell. The command 'node filessys.js' has been executed, and the output is visible:

```
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

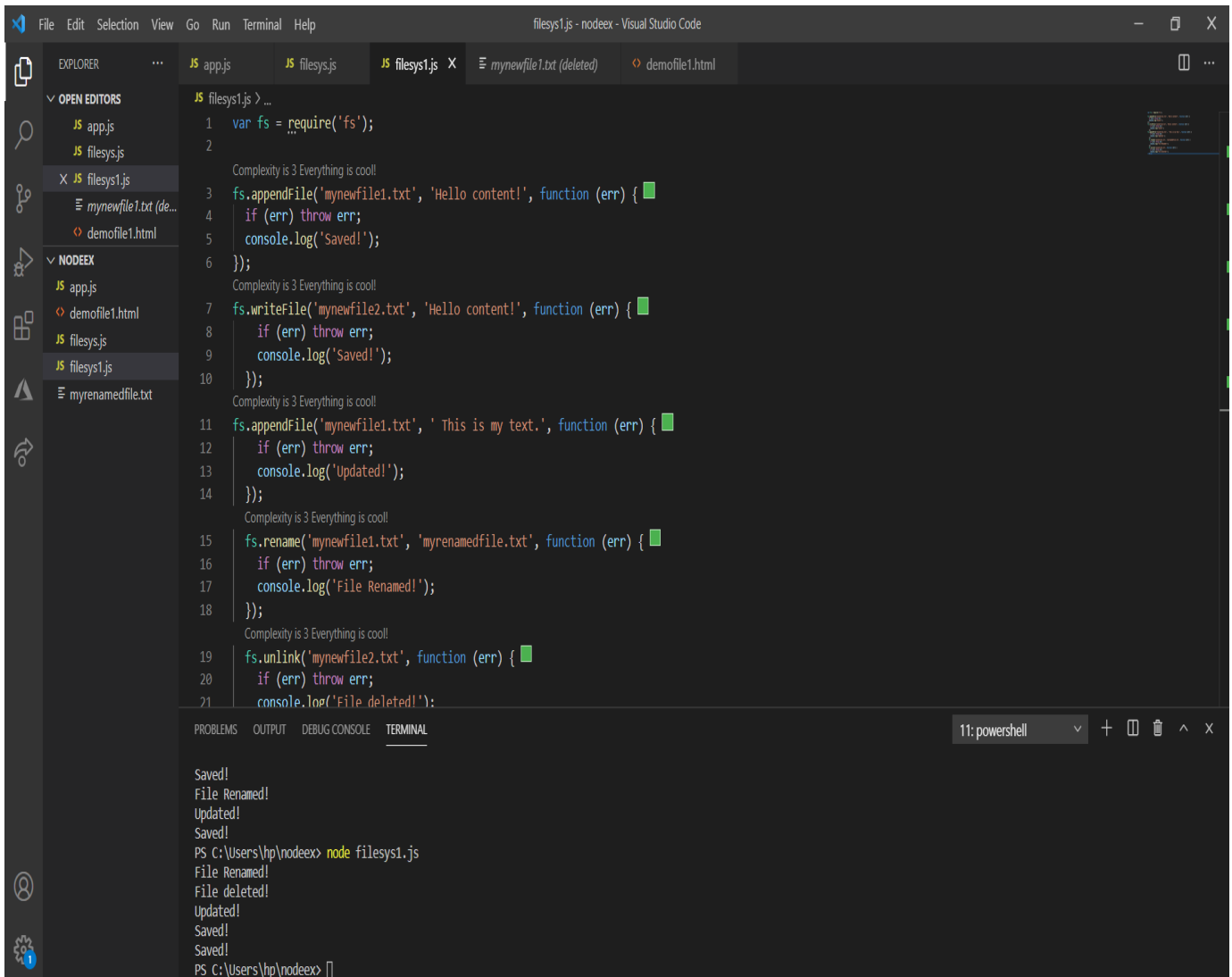
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\hp\nodeex> node filessys.js
```

← → ↻ ⓘ localhost:9000

# NODEJS

This is practical no.3



The screenshot displays the Visual Studio Code interface with a project named 'filesys1.js - nodeex'. The Explorer sidebar on the left shows the file structure: 'app.js', 'filesys.js', 'mynewfile1.txt (deleted)', and 'demoFile1.html'. The 'OPEN EDITORS' section shows 'filesys1.js' as the active file. The main editor area displays the code for 'filesys1.js', which includes file operations like creating, writing, appending, renaming, and deleting files. The code is as follows:

```
1 var fs = require('fs');
2
3 Complexity is 3 Everything is cool!
4 fs.appendFile('mynewfile1.txt', 'Hello content!', function (err) {
5   if (err) throw err;
6   console.log('Saved!');
7 });
8 Complexity is 3 Everything is cool!
9 fs.writeFile('mynewfile2.txt', 'Hello content!', function (err) {
10   if (err) throw err;
11   console.log('Saved!');
12 });
13 Complexity is 3 Everything is cool!
14 fs.appendFile('mynewfile1.txt', ' This is my text.', function (err) {
15   if (err) throw err;
16   console.log('Updated!');
17 });
18 Complexity is 3 Everything is cool!
19 fs.rename('mynewfile1.txt', 'myrenamedfile.txt', function (err) {
20   if (err) throw err;
21   console.log('File Renamed!');
22 });
23 Complexity is 3 Everything is cool!
24 fs.unlink('mynewfile2.txt', function (err) {
25   if (err) throw err;
26   console.log('File deleted!');
27 });
```

The TERMINAL panel at the bottom shows the output of the application, including 'Saved!', 'File Renamed!', 'Updated!', and 'File deleted!' messages. The command prompt shows the command 'node filesys1.js' being executed in the directory 'C:\Users\hp\nodeex'.

b. Secure HTTP Module (https)

```
Administrator: Command Prompt - choco install openssl
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\windows\system32>cd C:\Users\hp\nodeex

C:\Users\hp\nodeex>choco install openssl
Chocolatey v0.10.15
Installing the following packages:
openssl
By installing you accept licenses for the packages.
Progress: Downloading vcredist2015 14.0.24215.20170201... 100%
Progress: Downloading vcredist140 14.28.29913... 100%
Progress: Downloading chocolatey-core.extension 1.3.5.1... 100%
Progress: Downloading KB3033929 1.0.5... 100%
Progress: Downloading chocolatey-windowsupdate.extension 1.0.4... 100%
Progress: Downloading KB3035131 1.0.3... 100%
Progress: Downloading KB2919355 1.0.20160915... 100%
Progress: Downloading KB2919442 1.0.20160915... 100%
Progress: Downloading KB2999226 1.0.20181019... 100%
Progress: Downloading openssl 1.1.1.1000... 45%
```

```
Administrator: Command Prompt - choco install openssl

Installed/updated chocolatey-core extensions.
The install of chocolatey-core.extension was successful.
Software installed to 'C:\ProgramData\chocolatey\extensions\chocolatey-core'

chocolatey-windowsupdate.extension v1.0.4 [Approved]
chocolatey-windowsupdate.extension package files install completed. Performing other installation steps.
Installed/updated chocolatey-windowsupdate extensions.
The install of chocolatey-windowsupdate.extension was successful.
Software installed to 'C:\ProgramData\chocolatey\extensions\chocolatey-windowsupdate'

KB3035131 v1.0.3 [Approved]
kb3035131 package files install completed. Performing other installation steps.
The package KB3035131 wants to run 'ChocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): Y
```

C:\> Administrator: Command Prompt - choco install openssl

```
default install location if installer.

vcredist2015 v14.0.24215.20170201 [Approved]
vcredist2015 package files install completed. Performing other installation steps.
The install of vcredist2015 was successful.
Software install location not explicitly set, could be in package or
default install location if installer.

openssl v1.1.1.1000 [Approved]
openssl package files install completed. Performing other installation steps.
The package openssl wants to run 'chocolateyinstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): Y

Installing 64-bit openssl...
```

C:\Windows\system32>set

C:\> Administrator: Command Prompt

```
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): Y

Installing 64-bit openssl...
openssl has been installed.
PATH environment variable does not have C:\Program Files\OpenSSL-Win64\bin in it. Adding...
WARNING: OPENSSL_CONF has been set to C:\Program Files\OpenSSL-Win64\bin\openssl.cfg
openssl can be automatically uninstalled.
Environment Vars (like PATH) have changed. Close/reopen your shell to
see the changes (or in powershell/cmd.exe just type `refreshenv`).
The install of openssl was successful.
Software installed to 'C:\Program Files\OpenSSL-Win64\'

Chocolatey installed 10/10 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Installed:
- kb2919355 v1.0.20160915
- kb3033929 v1.0.5
- chocolatey-core.extension v1.3.5.1
- kb2999226 v1.0.20181019
- openssl v1.1.1.1000
- vcredist2015 v14.0.24215.20170201
- kb2919442 v1.0.20160915
- vcredist140 v14.28.29913
- kb3035131 v1.0.3
- chocolatey-windowsupdate.extension v1.0.4

C:\Users\hp\nodeex>
```

```
Select Command Prompt - openssl req -newkey rsa:2048 -nodes -keyout key.pem -x509 -days 365 -out certificate.p...
C:\Users\hp>cd nodeex
C:\Users\hp\nodeex>openssl req -newkey rsa:2048 -nodes -keyout key.pem -x509 -days 365 -out certificate.pem

Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
```

```
Command Prompt
C:\Users\hp>cd nodeex
C:\Users\hp\nodeex>openssl req -newkey rsa:2048 -nodes -keyout key.pem -x509 -days 365 -out certificate.pem

Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:Maharashtra
Locality Name (eg, city) []:Latur
Organization Name (eg, company) [Internet Widgits Pty Ltd]:WCE, sangli
Organizational Unit Name (eg, section) []:CSE Dept
Common Name (e.g. server FQDN or YOUR name) []:POOJA
Email Address []:pooja.shinde@walchandsangli.ac.in

C:\Users\hp\nodeex>
```

Name

- node\_modules
- nodejs-email
- p4
- app
- certificate.pem
- connection
- createdatabase
- createtable
- demofile1
- email
- filesys
- filesys1
- insertrecord
- key.pem
- myfirstmodule
- myrenamedfile

Command Prompt

```
Common Name (e.g. server FQDN or YOUR name) []:POOJA
Email Address []:pooja.shinde@walchandsangli.ac.in

C:\Users\hp\nodeex>openssl x509 -text -noout -in certificate.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            4a:16:48:e7:bc:88:33:9e:f7:3f:fb:de:2e:3a:50:9c:4b:ce:92:01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = IN, ST = Maharashtra, L = Latur, O = "WCE, sangli", OU = CSE Dept, CN = POOJA, emailAd
ress = pooja.shinde@walchandsangli.ac.in
        Validity
            Not Before: Mar 31 11:57:06 2021 GMT
            Not After : Mar 31 11:57:06 2022 GMT
        Subject: C = IN, ST = Maharashtra, L = Latur, O = "WCE, sangli", OU = CSE Dept, CN = POOJA, emailAd
dress = pooja.shinde@walchandsangli.ac.in
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public-Key: (2048 bit)
            Modulus:
                00:b8:c0:99:84:8d:c9:11:fe:7a:f8:65:d0:b5:ad:
                70:39:68:7c:41:75:98:83:22:63:a4:62:05:e4:69:
                67:72:85:a2:22:0f:c1:32:9f:dd:41:41:66:09:9d:
                88:23:ad:b3:46:dc:f2:93:41:ac:be:da:9c:9c:eb:
                bf:29:b8:41:84:cb:0c:04:9d:4d:db:5d:a1:97:23:
                84:69:c1:12:96:c4:3a:58:73:49:b8:5c:26:b1:f1:
                93:54:8d:9f:66:2d:1f:64:8e:c5:26:11:bf:5c:08:
                54:80:fd:53:34:e5:e5:b6:53:ec:f7:05:b8:bf:c3:
                1f:9a:ac:60:e2:e0:f2:1e:ee:30:8c:c1:88:3d:ad:
```

```
54:a1:eb:eb:23:3f:ec:e0:c3:9d:bd:68:4e:39:56:
21:1d
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Subject Key Identifier:
    63:88:BB:2F:F7:D1:FC:47:2D:9F:90:10:B5:C4:8E:1A:CB:DE:25:35
X509v3 Authority Key Identifier:
    keyid:63:88:BB:2F:F7:D1:FC:47:2D:9F:90:10:B5:C4:8E:1A:CB:DE:25:35

X509v3 Basic Constraints: critical
CA:TRUE
Signature Algorithm: sha256WithRSAEncryption
a9:e6:e8:b1:a6:79:65:ea:f4:a0:b4:a8:03:c7:3c:71:87:94:
d2:39:13:8b:48:76:9a:c8:fc:74:47:bb:e5:d4:ec:c9:a3:67:
69:11:f6:db:7e:ec:f9:5a:67:4e:e2:43:d9:7b:44:51:d8:57:
1c:9c:7d:26:03:63:40:71:72:ea:33:bb:6b:60:fb:c6:d4:7b:
0a:64:d7:e7:f8:35:61:75:70:31:7a:4f:d6:86:88:e1:74:c2:
37:84:b5:3a:a8:b8:1d:7a:12:7c:c4:c2:d7:b2:59:f8:33:4a:
9b:55:ee:d1:33:15:56:2a:6a:ee:74:ca:1e:0b:62:2a:56:d1:
dc:59:c6:c2:60:e3:25:bb:77:33:aa:28:55:fe:7d:5e:bf:02:
c7:6e:ae:3d:2c:b9:77:ad:1a:85:37:86:61:ee:55:b8:d7:fc:
ad:85:c2:10:d0:df:c3:fa:25:ee:25:d0:1a:a8:18:01:0c:b8:
c3:fc:11:a2:52:f6:57:5e:2a:91:ce:86:b6:a6:44:be:85:90:
19:49:e6:3b:86:04:5c:17:b9:e8:98:8f:97:e6:5d:b6:8c:d7:
6b:55:bb:24:89:c5:a8:fe:6a:77:5f:eb:af:a7:03:7e:75:4b:
09:88:9a:9e:fa:16:bb:b6:50:da:e8:ee:ed:00:d7:99:4c:77:
09:34:e7:e5
```

C:\Users\hp\nodeex>

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists files in the 'NODEEX' directory, including 'certificate.pem', 'connection.js', 'createdatabase.js', 'createtable.js', 'demoFile1.html', 'email.js', 'fileSYS.js', 'fileSYS1.js', 'insertrecord.js', 'key.pem', 'myfirstmodule.js', 'myrenamedfile.txt', 'package-lock.json', 'package.json', 'retrieve1.js', 'retrieve2.js', 'retrieve3.js', 'securehttp.js', and 'userdefined.js'. The 'securehttp.js' file is open in the editor, showing the following JavaScript code:

```
1 const https = require('https');
2 const fs = require('fs');
3
4 const options = {
5   key: fs.readFileSync('key.pem'),
6   cert: fs.readFileSync('certificate.pem')
7 };
8
9 https.createServer(options, function (req, res) {
10   res.writeHead(200);
11   res.end("hello world\n");
12 }).listen(8000);
13
```

The Terminal window at the bottom shows the command prompt for Windows PowerShell, displaying the command 'node securehttp.js' and a cursor.



### Your connection is not private

Attackers might be trying to steal your information from **localhost** (for example, passwords, messages, or credit cards). [Learn more](#)

NET::ERR\_CERT\_AUTHORITY\_INVALID



To get Chrome's highest level of security, [turn on enhanced protection](#)

Advanced

Back to safety

hello world



c. Nodemailer Module (nodemailer)

i. Install this module from npm

```
Command Prompt

C:\Users\hp>cd nodeex

C:\Users\hp\nodeex>choco install openssl
'choco' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\hp\nodeex>cd..

C:\Users\hp>node -v
v14.16.0

C:\Users\hp>npm -v
6.14.11

C:\Users\hp>npm install nodemailer
npm WARN saveError ENOENT: no such file or directory, open 'C:\Users\hp\package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open 'C:\Users\hp\package.json'
npm WARN hp No description
npm WARN hp No repository field.
npm WARN hp No README data
npm WARN hp No license field.

+ nodemailer@6.5.0
added 1 package from 1 contributor and audited 1 package in 6.389s
found 0 vulnerabilities

C:\Users\hp>
```

The screenshot shows the Visual Studio Code editor with the 'email.js' file open. The file contains the following code:

```
10  });
11  var mailOptions = {
12    from: 'shindep9294@gmail.com',
13    to: 'pooja.shinde@walchandsangli.ac.in',
14    subject: 'Sending Email using Node.js',
15    text: 'That was easy!'
16  };
17
18  Complexity is 3 Everything is cool!
19  transporter.sendMail(mailOptions, function(error, info){
20    if (error) {
21      console.log(error);
22    } else {
23      console.log('Email sent: ' + info.response);
24    }
25  });
```

The terminal at the bottom shows the command `node email.js` being executed, resulting in the output: `Email sent: 250 2.0.0 OK 1617079709 c11sm8213663pgk.83 - gsmtp`.

↩ Reply   ↩ Reply all   → Forward   📁 Archive   ⋮

## Sending Email using Node.js



**shindep9294@gmail.com** <shindep9294@gmail.com>

10:18



To: pooja.shinde@walchandsangli.ac.in

That was easy!

## Problem Statement 2:

1. Create a user defined module and use it in Node.js Application that uses HTTP server.

The screenshot displays the Visual Studio Code interface. The Explorer sidebar on the left shows the project structure with files like `app.js`, `filessys.js`, `filessys1.js`, `myfirstmodule.js`, `userdefined.js`, `mynewfile1.txt`, and `demofile1.html`. The main editor window shows the code in `userdefined.js`:

```
1 var http = require('http');
2 var dt = require('./myfirstmodule');
3
4 http.createServer(function (req, res) {
5     res.writeHead(200, {'Content-Type': 'text/html'});
6     res.write("The date and time are currently: " + dt.myDateTime());
7     res.end();
8 }).listen(4000);
```

Below the editor is a terminal window running a Windows PowerShell prompt. It shows the command `node userdefined.js` being executed. At the bottom of the image, a browser window shows the output of the server at `localhost:4000`.

The date and time are currently: Tue Mar 30 2021 09:03:37 GMT+0530 (India Standard Time)

