

1. Assignment number 1, nodejs installation and running server with pm2

1. Install node js and npm

node js and npm were installed with :

```
# sudo apt install nodejs  
# sudo apt install npm
```

```
manish@manish-intern:~$ sudo apt install nodejs
```

```
manish@manish-intern:~$ node -v  
v10.19.0  
manish@manish-intern:~$
```

```
manish@manish-intern:~$ npm --version  
6.14.4  
manish@manish-intern:~$
```

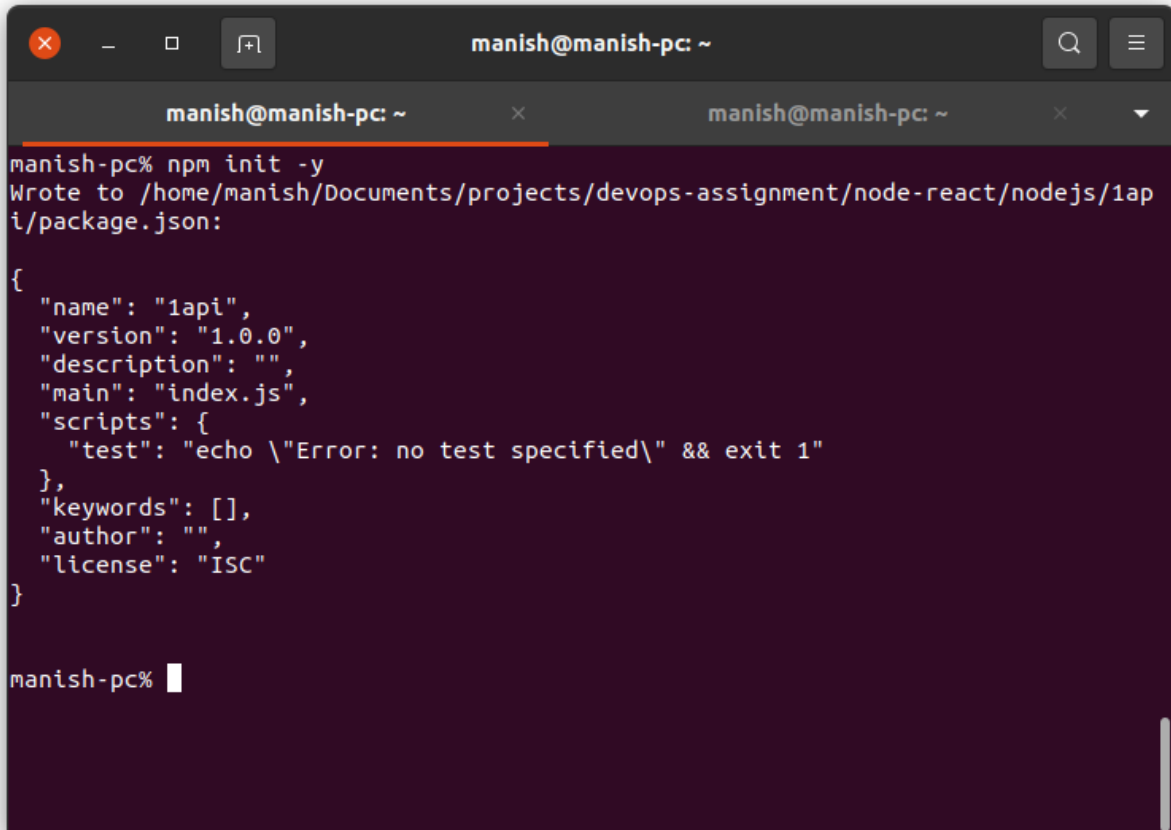
2. Create node projects 1api and 2api

Created 2 directories 1api and 2api and ran the following command respectfully in both of the directories:

mkdir 1api 2api

Created node projects with npm init command in both of the directories:

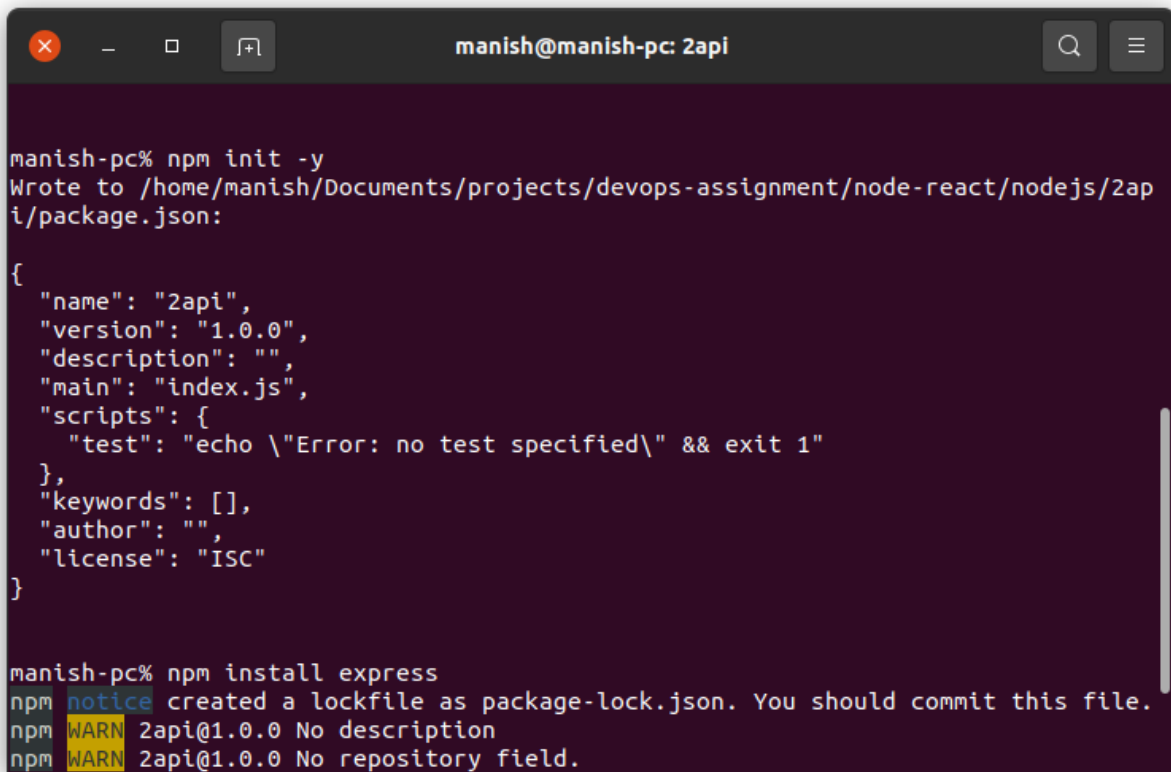
npm init -y

A screenshot of a terminal window with a dark purple background. The window title is "manish@manish-pc: ~". The terminal shows the command "manish-pc% npm init -y" being executed. The output indicates that a package.json file was written to the path "/home/manish/Documents/projects/devops-assignment/node-react/nodejs/1api/package.json:". The resulting JSON object is displayed, containing fields for name, version, description, main, scripts, keywords, author, and license. The "scripts" field includes a "test" script that echoes an error message and exits with code 1. The terminal prompt "manish-pc%" is visible at the bottom, followed by a cursor.

```
manish-pc% npm init -y
Wrote to /home/manish/Documents/projects/devops-assignment/node-react/nodejs/1api/package.json:

{
  "name": "1api",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

manish-pc% 
```

A terminal window titled 'manish@manish-pc: 2api' with standard window controls. It shows the execution of 'npm init -y' which creates a package.json file. The output of 'npm install express' shows a lockfile being created and two warnings about missing description and repository fields.

```
manish-pc% npm init -y
Wrote to /home/manish/Documents/projects/devops-assignment/node-react/nodejs/2api/package.json:

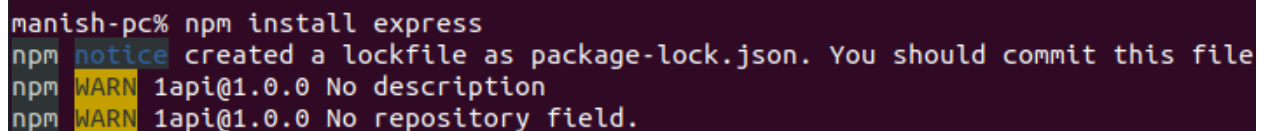
{
  "name": "2api",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

manish-pc% npm install express
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN 2api@1.0.0 No description
npm WARN 2api@1.0.0 No repository field.
```

3. Installed dependencies for api server:

Express js is used to run a node js app server.

npm install express

A terminal window showing the command 'npm install express' and its output, including a notice about a lockfile and two warnings about missing description and repository fields.

```
manish-pc% npm install express
npm notice created a lockfile as package-lock.json. You should commit this file
npm WARN 1api@1.0.0 No description
npm WARN 1api@1.0.0 No repository field.
```

4. Install pm2 in both projects:

pm2 was installed in both of the projects with the following command:

npm install pm2

```
manish@manish-pc: 2api
found 0 vulnerabilities

manish-pc% npm install pm2
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.3.2 (node_modules/chokidar/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
npm WARN 2api@1.0.0 No description
npm WARN 2api@1.0.0 No repository field.

+ pm2@5.1.2
added 200 packages from 190 contributors and audited 251 packages in 19.023s

10 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

manish-pc%
```

5. Setting the ports of node apps to 6080

Api 1 was configured to run on port 6080:

```
app.listen(6080, () => {  
  console.log(`Server listening on port 6080`);  
});
```

Api 2 was configured to run on port 7080:

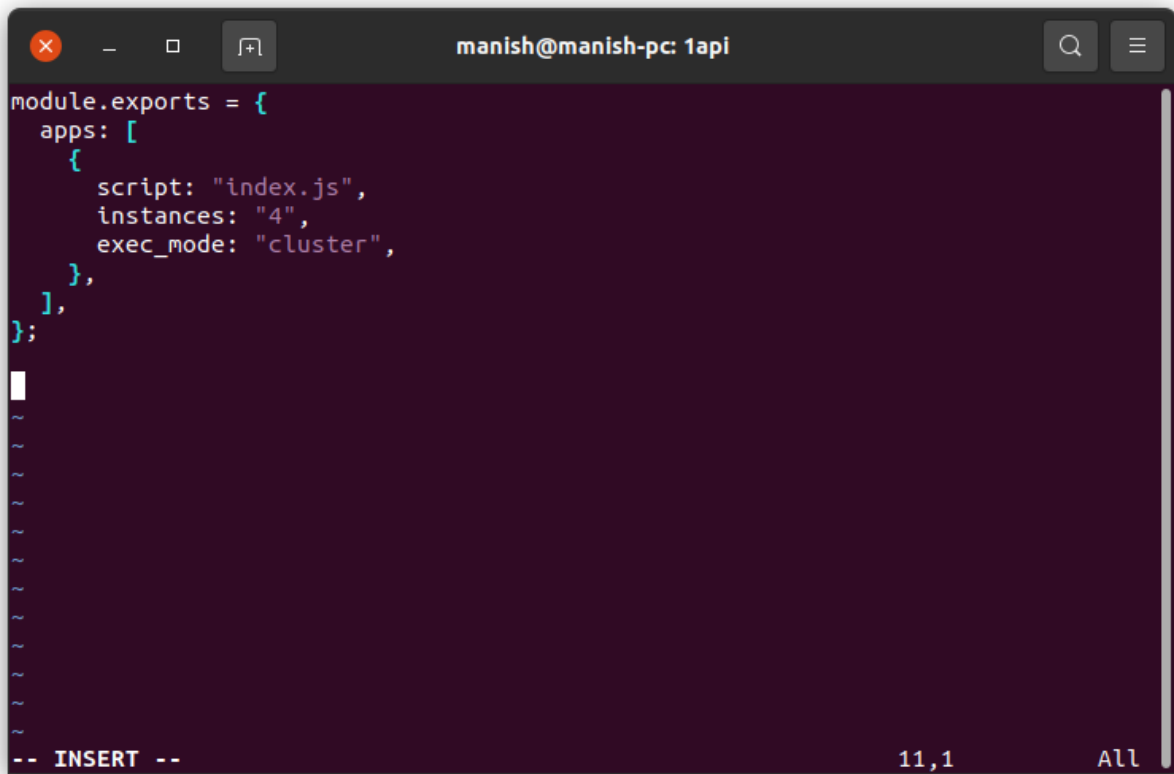
```
app.listen(7080, () => {  
  console.log(`Server listening on port 7080`);  
});
```

6. Create pm2 configuration file:

Created a configuration file **process.config.js** on both api projects containing the following parameters:

```
module.exports = {  
  apps: [  
    {  
      script: "index.js",  
      instances: "4",  
      exec_mode: "cluster",  
    },  
  ],  
};
```

Where: exec_mode= cluster indicates the pm2 to run our node js server on cluster mode and instances 4 means that we are running 4 processes.

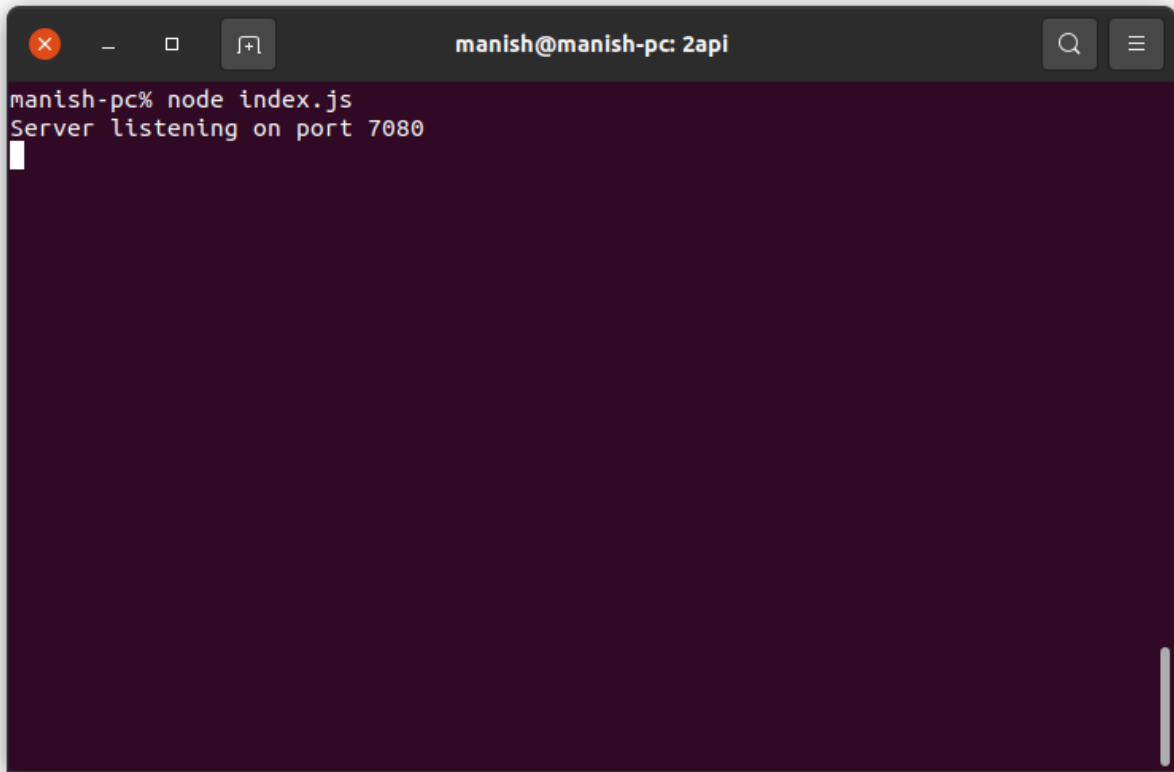


The image shows a code editor window with a dark theme. The title bar at the top reads "manish@manish-pc: 1api". The code is as follows:

```
module.exports = {  
  apps: [  
    {  
      script: "index.js",  
      instances: "4",  
      exec_mode: "cluster",  
    },  
  ],  
};
```

Below the code, there are several tilde (~) characters representing a list. At the bottom left, it says "-- INSERT --". At the bottom right, it shows "11,1" and "All".

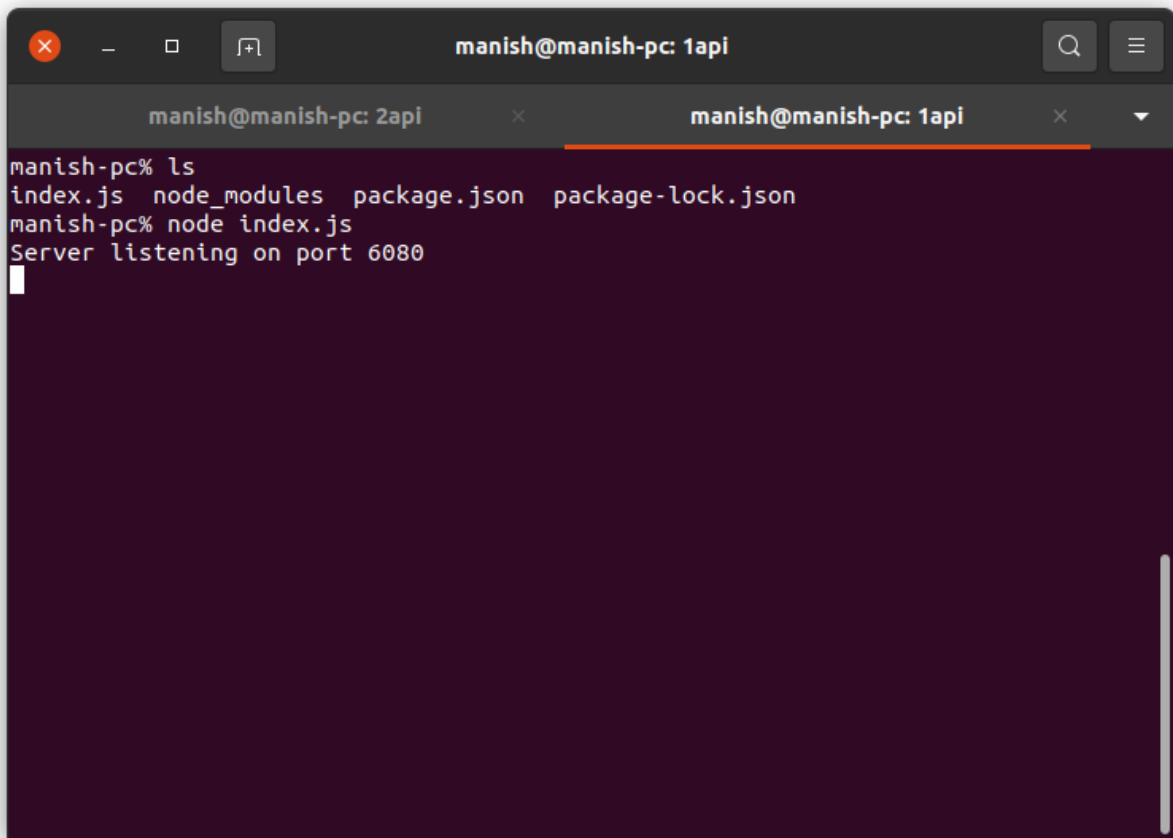
7. Run the apps with node command
Our app server code was tested with node command as :
node index.js



A terminal window titled "manish@manish-pc: Zapi" with standard macOS window controls. The terminal output shows the command "node index.js" being executed, resulting in the message "Server listening on port 7080". A white cursor is positioned on the line following the output. The terminal has a dark purple background and a vertical scrollbar on the right side.

```
manish-pc% node index.js
Server listening on port 7080

```

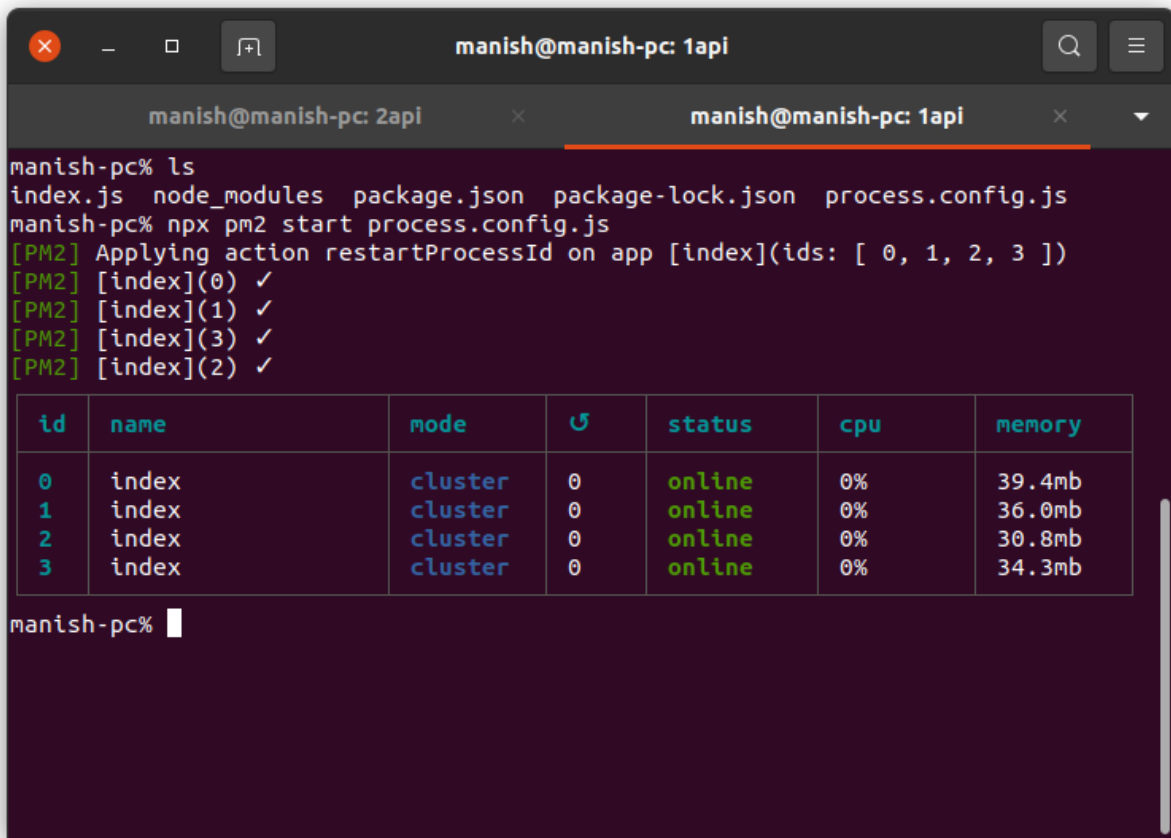
A terminal window with a dark background and light-colored text. The window title bar shows 'manish@manish-pc: 1api'. The terminal content shows the user running 'ls' to list files, then 'node index.js' to start a server. The output shows the server listening on port 6080.

```
manish@manish-pc: 1api
manish-pc% ls
index.js  node_modules  package.json  package-lock.json
manish-pc% node index.js
Server listening on port 6080
```


8. Run the apps with pm2 commands

Node server was started with pm2 with the following command:

```
# npx pm2 start process.config.js
```

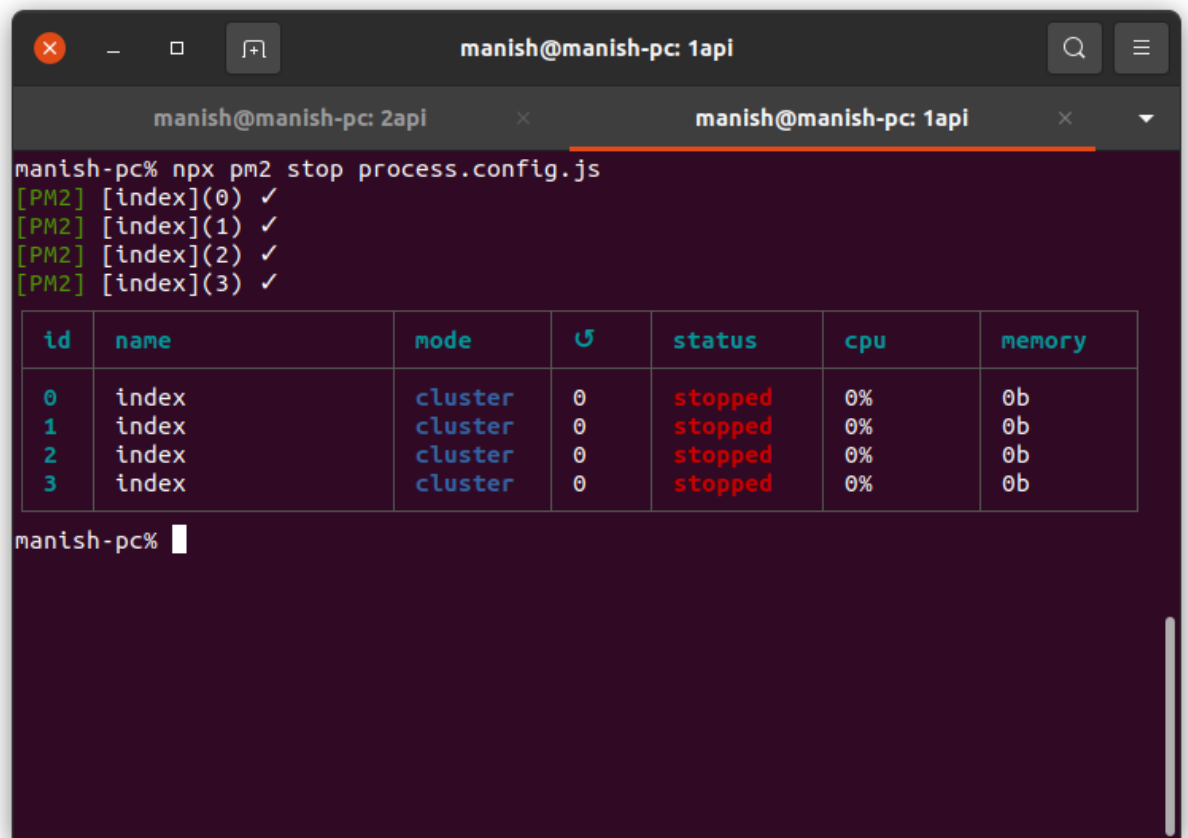

A terminal window titled 'manish@manish-pc: 1api' with two tabs. The active tab shows the following commands and output:
manish-pc% ls
index.js node_modules package.json package-lock.json process.config.js
manish-pc% npx pm2 start process.config.js
[PM2] Applying action restartProcessId on app [index](ids: [0, 1, 2, 3])
[PM2] [index](0) ✓
[PM2] [index](1) ✓
[PM2] [index](3) ✓
[PM2] [index](2) ✓
Below the output is a table showing the status of the four processes.

id	name	mode	🔄	status	cpu	memory
0	index	cluster	0	online	0%	39.4mb
1	index	cluster	0	online	0%	36.0mb
2	index	cluster	0	online	0%	30.8mb
3	index	cluster	0	online	0%	34.3mb

manish-pc% 

Node server was stopped with pm2 with the following command:

npx pm2 stop process.config.js

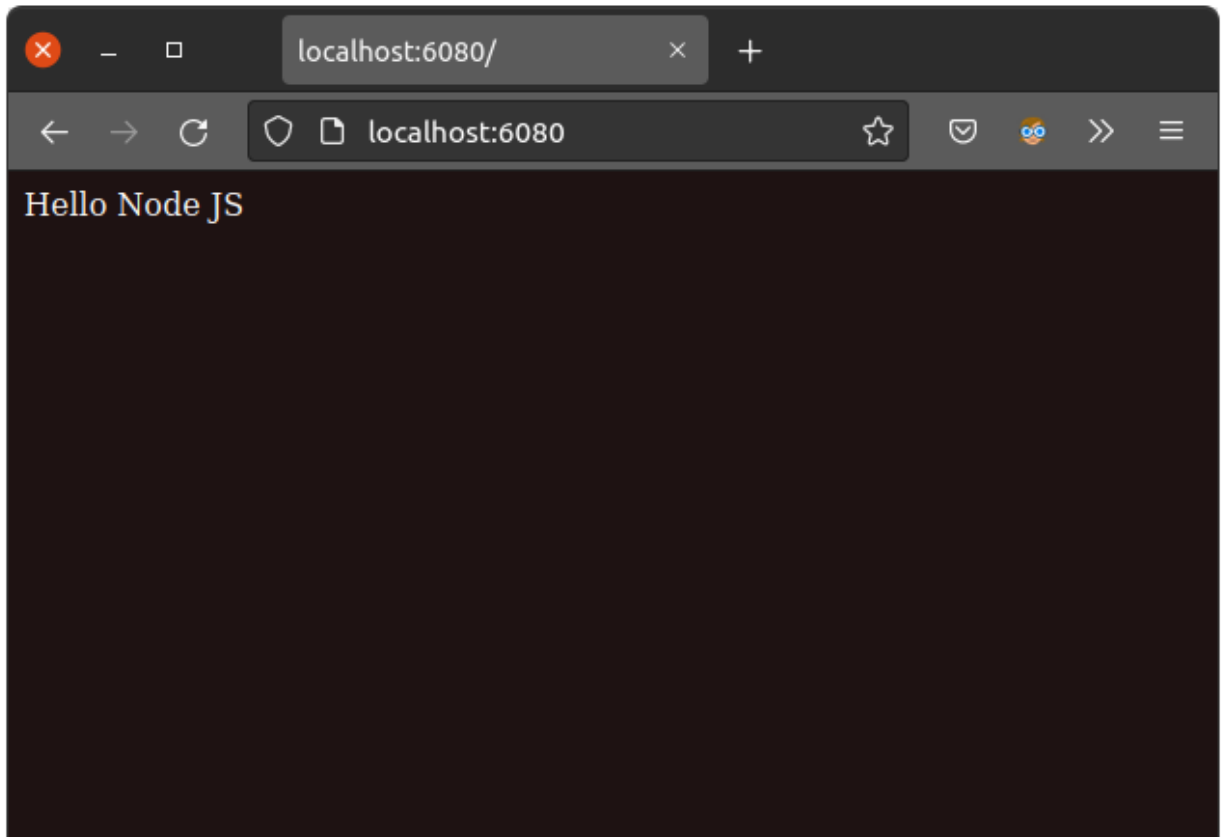


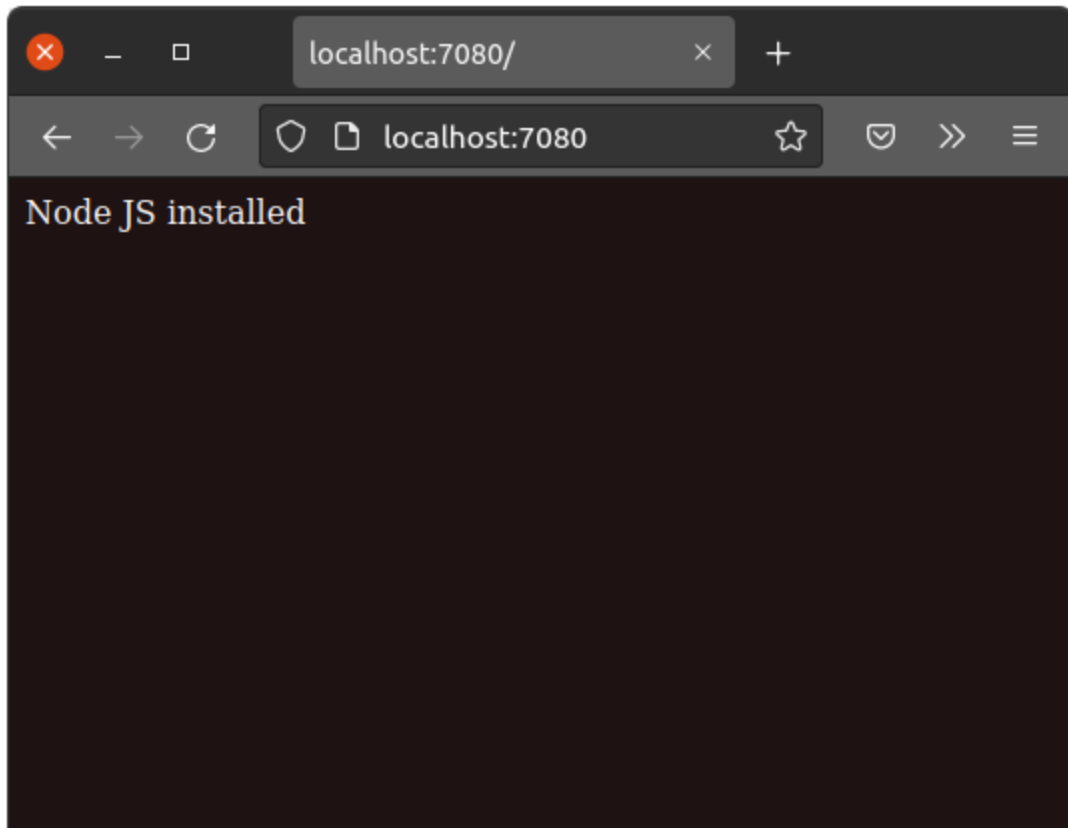
```
manish-pc% npx pm2 stop process.config.js
[PM2] [index](0) ✓
[PM2] [index](1) ✓
[PM2] [index](2) ✓
[PM2] [index](3) ✓
```

id	name	mode	↻	status	cpu	memory
0	index	cluster	0	stopped	0%	0b
1	index	cluster	0	stopped	0%	0b
2	index	cluster	0	stopped	0%	0b
3	index	cluster	0	stopped	0%	0b

```
manish-pc% 
```

9. Running the api on browser

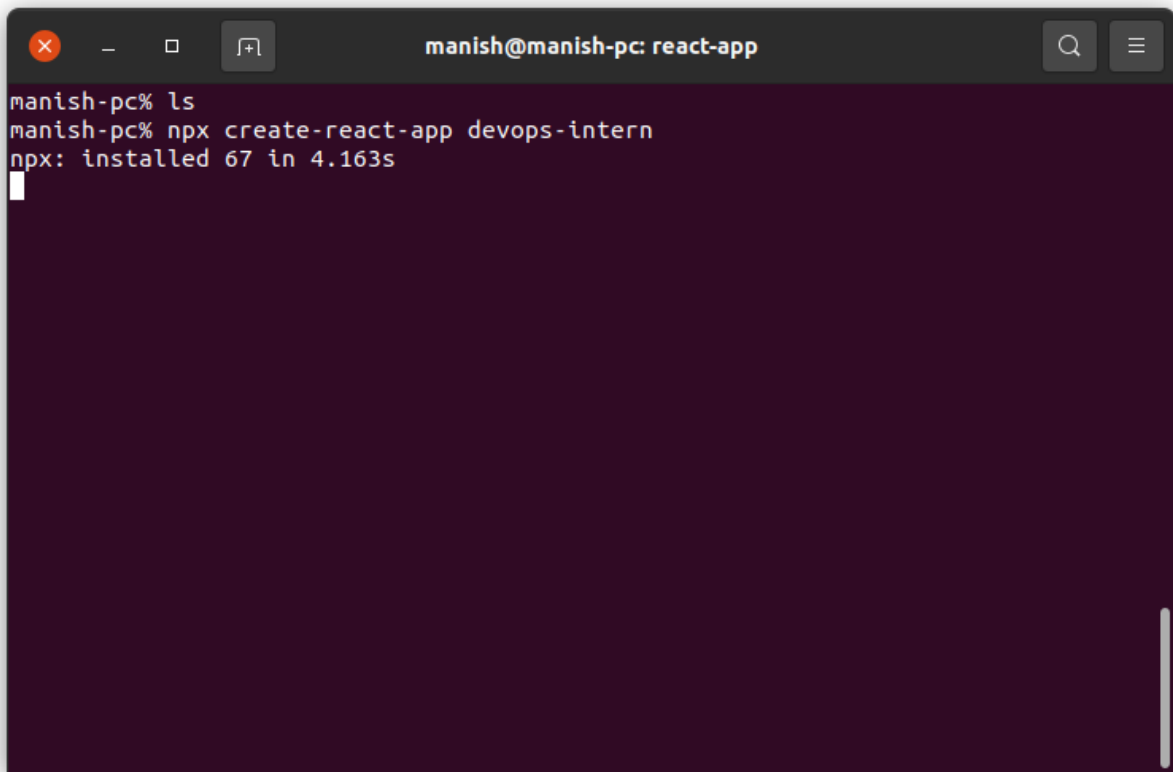




2. Assignment number 2, running react app

1. Create a new react with create-react-app:
A new react project named devops-intern was created using the create-react-app command as:

```
# npx create-react-app devops-intern
```

A terminal window titled "manish@manish-pc: react-app" with standard window controls. The terminal shows the following commands and output:

```
manish-pc% ls
manish-pc% npx create-react-app devops-intern
npx: installed 67 in 4.163s
```

2. Create an env file with PORT variable with value 3001:

React runs on port 3000 on default, to run it on a different port a .env file with PORT variable needs to be created.

PORT=3001

3. Run the react app

React app was run with the npm start command which is specified in the package.json of the root project's root directory.

npm start

```
manish@manish-pc: devops-intern

Compiled successfully!

You can now view devops-intern in the browser.

  Local:            http://localhost:3001
  On Your Network:  http://192.168.1.66:3001

Note that the development build is not optimized.
To create a production build, use npm run build.
```

