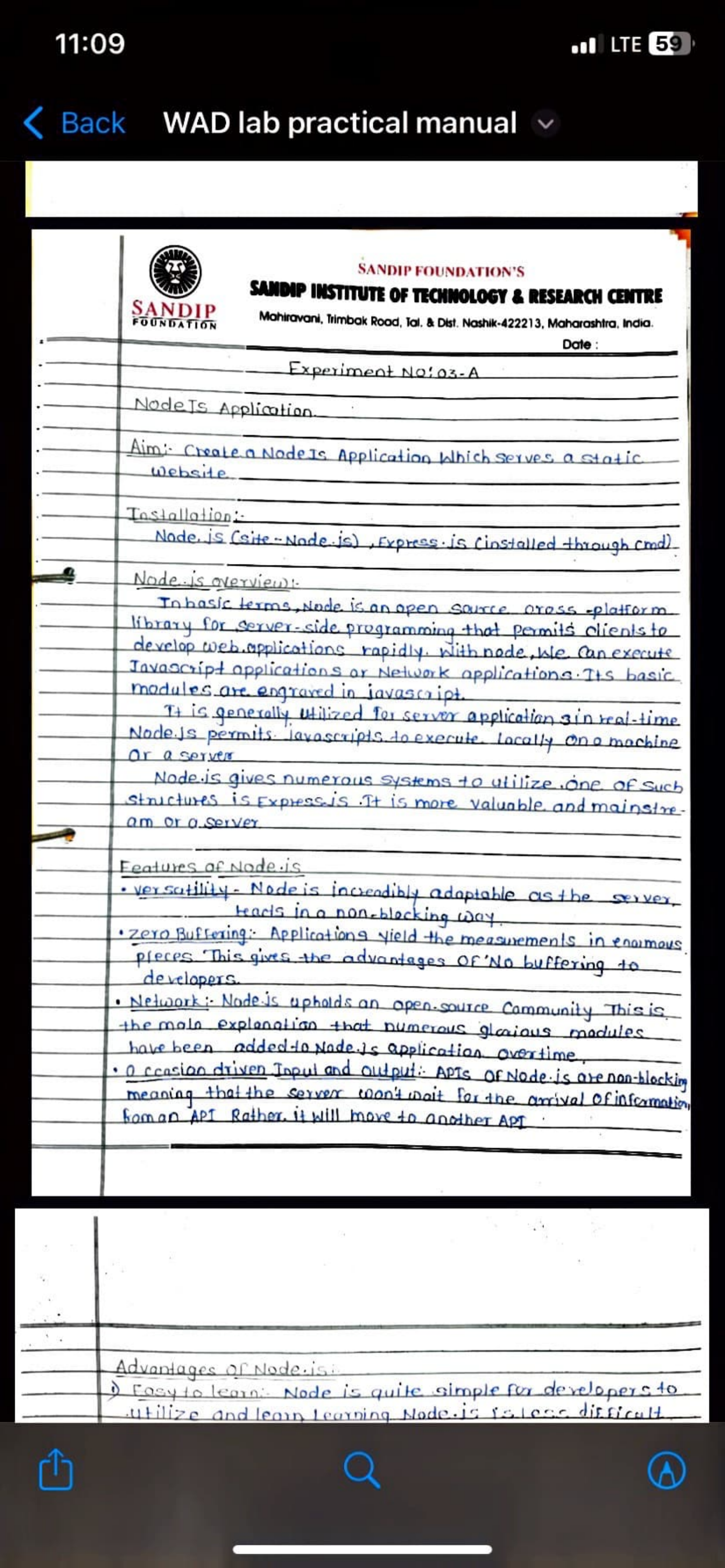


Date :

Experiment No:03-A

NodeJs Application

Aim:- Create a NodeJs Application Which serves a static website

Installation:-
Node.js (site - Node.js), Express.js (installed through cmd)

Node.js overview:-

In basic terms, Node is an open source cross -platform library for server-side programming that permits clients to develop web.applications rapidly. With node, We Can execute Javascript applications or Network applications. Its basic modules are engraved in javascript.

It is generally utilized for server application s in real-time Node.js permits javascripts to execute locally on a machine Or a server

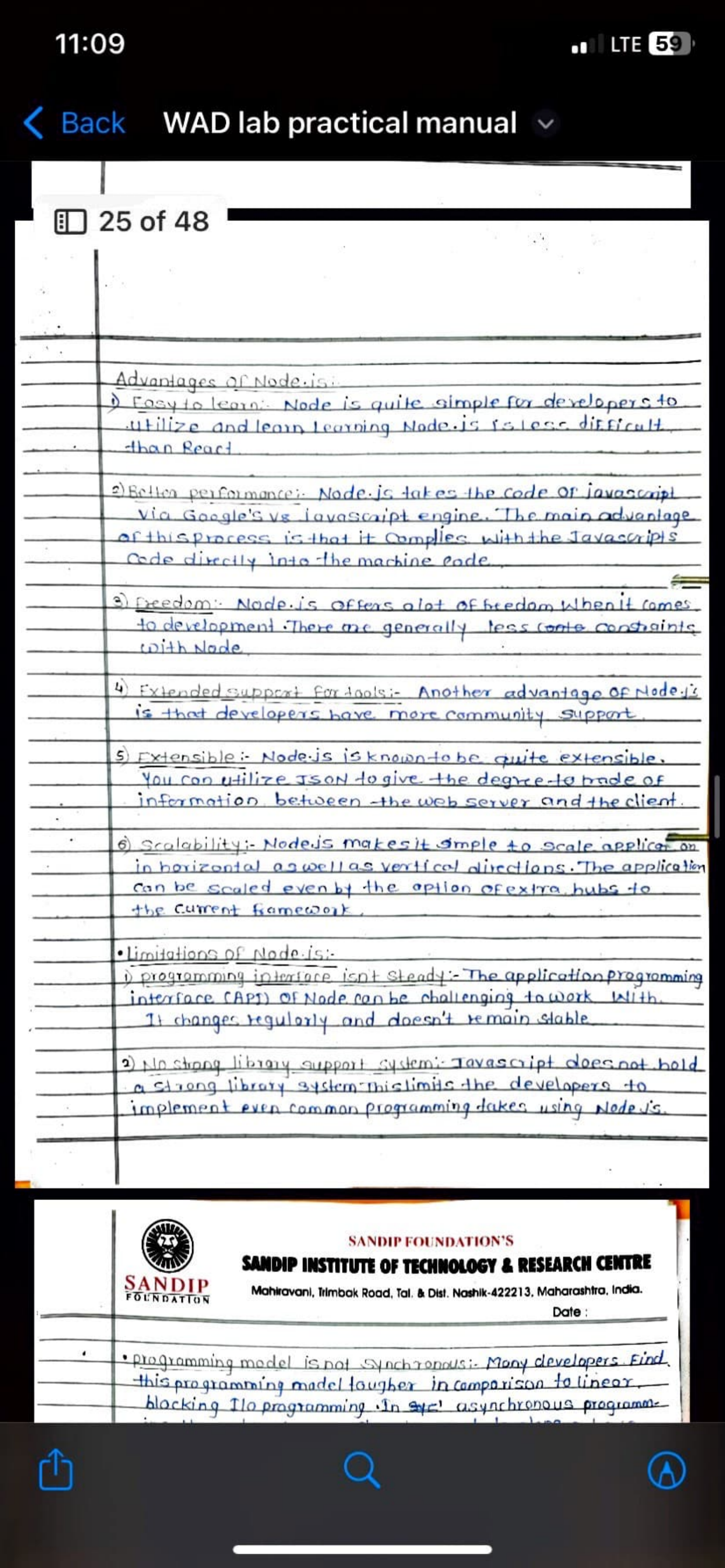
Node.js gives numerous systems to utilize. One of such structures is Express.js. It is more valuable and mainstream or a server

Features of Node.js

- versatility - Node is incredibly adaptable as the server reacts in a non-blocking way.
- Zero Buffering:- Applications yield the measurements in enormous pieces. This gives the advantages Of 'No buffering to developers.
- Network:- Node.js upholds an open-source Community This is the main explanation that numerous glorious modules have been added to Node.Js application overtime.
- O ccasion driven Input and output:- APIs Of Node.js are non-blocking meaning that the server won't wait for the arrival of information from an API Rather. it will move to another API

Advantages of Node.js:

1) Easy to learn:- Node is quite simple for developers to utilize and learn Learning Node.js is less difficult



Advantages of Node.is:

1) Easy to learn:- Node is quite simple for developers to utilize and learn learning Node.is is less difficult than React.

2) Better performance:- Node.js takes the code of javascript via Google's v8 javascript engine. The main advantage of this process is that it complies with the Javascripts code directly into the machine code.

3) Freedom:- Node.is offers alot of freedom when it comes to development. There are generally less ~~code~~ constraints with Node.

4) Extended support for tools:- Another advantage of Node.js is that developers have more community support.

5) Extensible:- Node.js is known to be quite extensible. You can utilize JSON to give the degree to trade of information between the web server and the client.

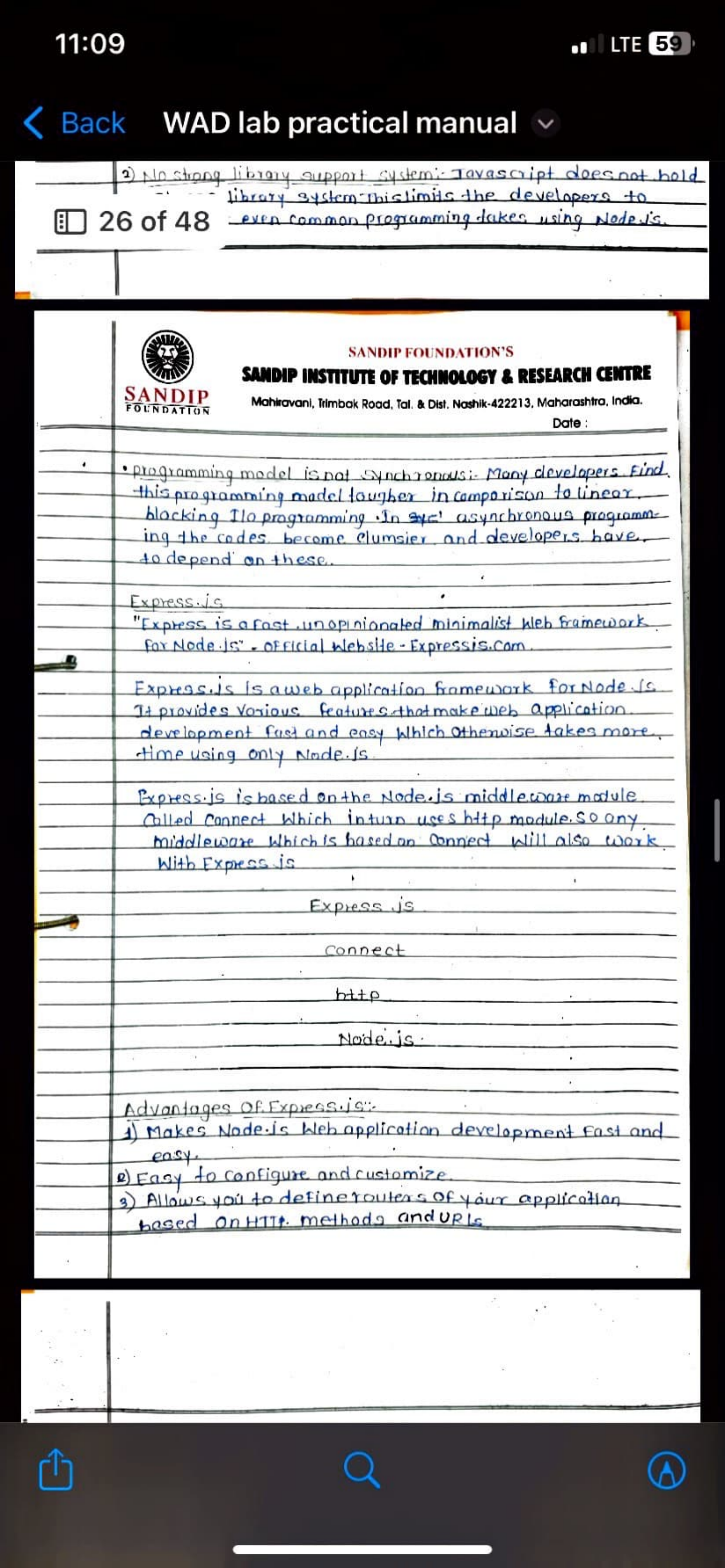
6) Scalability:- Node.js makes it simple to scale application in horizontal as well as vertical directions. The application can be scaled even by the option of extra hubs to the current framework.

- Limitations of Node.is:-

1) programming interface isn't steady:- The application programming interface (API) of Node can be challenging to work with. It changes regularly and doesn't remain stable.

2) No strong library support system:- Javascript does not hold a strong library system. This limits the developers to implement even common programming takes using Node.js.

- programming model is not synchronous:- Many developers find this programming model tougher in comparison to linear blocking I/o programming. In ~~syc~~ asynchronous programm-



2) No strong library support system:- Javascript does not hold library system. This limits the developers to even common programming takes using Node.js.

Date :

- programming model is not synchronous:- Many developers Find this programming model tougher in comparison to linear blocking I/o programming. In ~~syc~~' asynchronous programming the codes become clumsier and developers have to depend on these.

Express.js

"Express is a fast, unopinionated minimalist Web framework for Node.js" - Official Website - Expressjs.com.

Express.js is a web application framework for Node.js. It provides Various features that make web application development fast and easy Which otherwise takes more time using only Node.js

Express.js is based on the Node.js middleware module called Connect Which in turn uses http module. So any middleware Which is based on Connect Will also work With Express.js

Express.js

Connect

http

Node.js

Advantages Of Express.js:-

1) Makes Node.js Web application development Fast and easy.
2) Easy to configure and customize.
3) Allows you to define routers of your application based on HTTP methods and URLs.

1) Makes Node.js web application development fast and easy.
2) Easy to configure and customize.
3) Allows you to define routers of your application based on HTTP methods and URLs.

4) Includes various middleware modules which you can use to perform additional tasks on request and response.
5) Easy to integrate with different template engines like Jade Vash EJS etc.
6) Allows you to define an error handling middleware.
7) Easy to serve static files are resources of your applicatio
8) Allows you to Create REST API server.
9) Easy to connect with databases such as MongoDB, Redis, MySQL.

Steps:-
1. Install Node.js
2. Setting up express.js
3. Structuring files
4. Creating your express server
5. Servicing your static files.
6. Building your web page.
7. Running your project.

Open Node.js command terminal & run the following in your terminal -

Setting up express.js
1. create a new directory for your project : mkdir your-project -name
2) change into your new directory : cd your-project-name
3) Initialize a new node project with defaults. This will set a package.json file to access your dependencies : npminit -y
4) Create your entry file index.js This is where you will store your Express server. If you are working on Linux, you can run : touch index.js. if you are working on windows, you can edit in VS Code
5) Install Express as a dependency : npm Install express --save

on windows: you can edit in VS Code

5) Install Express as a dependency : npm Install express --save

6) Edit Packagejson within your package.json update your start script to include node and your Index.js file.

Let express-static-file-tutorial Is your project name

Package.json:-

```
{
  "name":"express-static-File-tutorial",
  "version": "1.0.0",
  "description": "",
  "main": "index.js".
  "scripts": {
   "start": "node.index.js" // change start value as node
                               index.js
                            // This Will ~~also~~ allow You to use the
  },                         npm start command in your
  "keywords": [],            terminal to launch Your express
  "author": "paul Holliday".   server
  "license": "MIT"}
```

Structuring Your Files:-

To store your files on the client-side, create a public directory and include an index.html file. express-static file-tutorial

```
|- index.js
|- public
  |- index.html
```

Creating Your Express Server

Edit index.js file

Index.js.

```
const express = require('express');
const app = express();
const PORT = 3000;
```

app.use(express.static('public')); //represents application

```
const express = require('express');
Const app = express();
st PORT = 3000;
```

```
app.use(express.static('public')); //represents application
                    is serving static
                //webpage in public directory
app. get ('/'.(req.res) => {
res.send ("Hello world!");
 });
app.listen(PORT.()=> console.log(server listening on port
           : ${PORT}'));
```

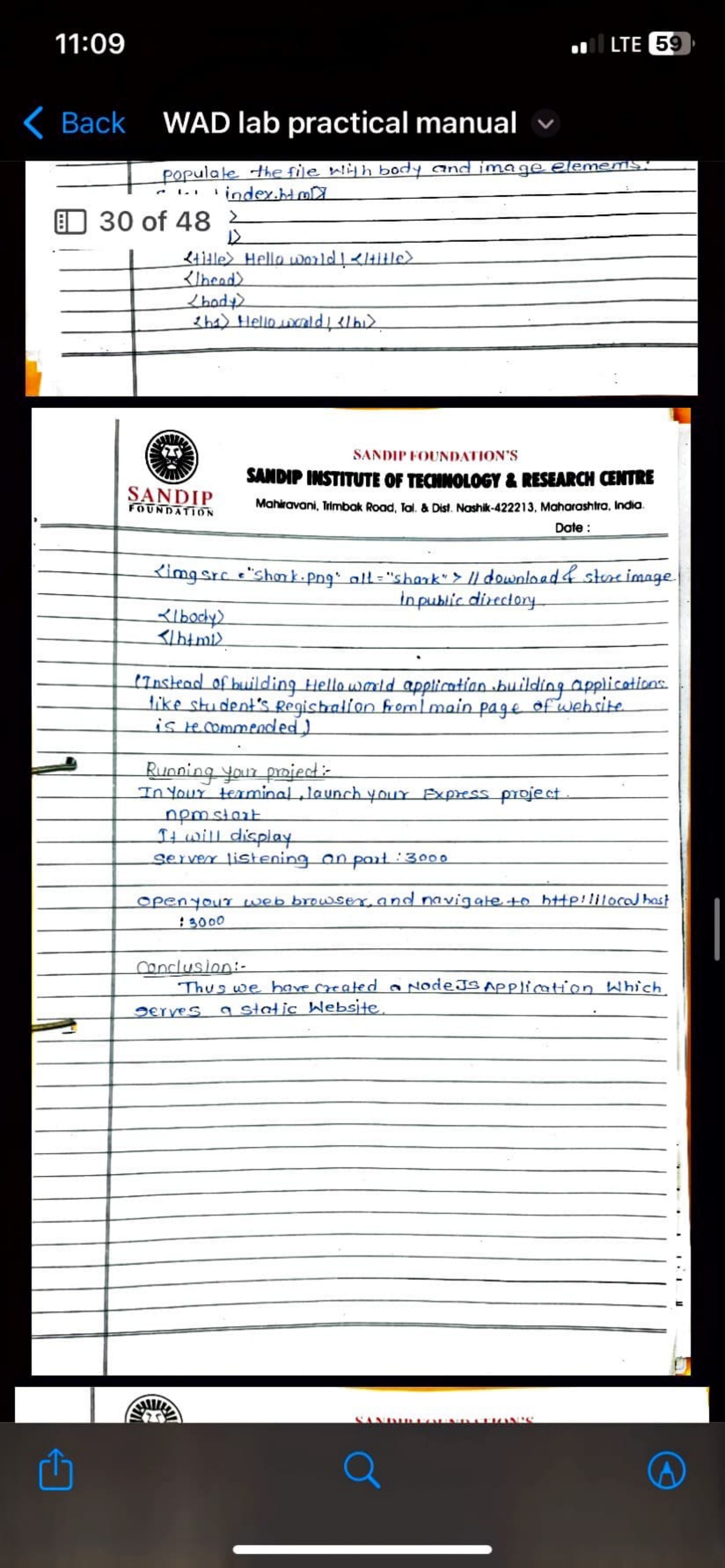
First of all, import the Express.is module
In above example, we imported Express.is module using require() function. The express module returns a function This function returns an object which can be used to Configure Express application (app in the above example)

- The app object includes methods for routing HTTP requests. configuring middleware, rendering HTML, views and registering a template engine.
- The app.listen() function creates the Node.js web server at the specified host and port. It is identical to Node's http.server.listen() method. Instead of Get(), post(), put() and delete() methods can be used.

Buil

- Building Your web page - client side
  Navigate to your index.html file In the public directory. populate the file with body and image elements:

```
[ label index.html]
  <html>
  <head>
  <title> Hello world! </title>
  </head>
  <body>
  <h1> Hello world! </h1>
```



populate the file with body and image elements:
index.html

```
>
1>
<title> Hello world! </title>
</head>
<body>
<h1> Hello world! </h1>
```

SANDIP FOUNDATION

SANDIP FOUNDATION'S

SANDIP INSTITUTE OF TECHNOLOGY & RESEARCH CENTRE

Mahiravani, Trimbak Road, Tal. & Dist. Nashik-422213, Maharashtra, India.

Date :

```
<img src ="shark.png" alt="shark" > // download & store image
                                      in public directory
</body>
</html>
```

(Instead of building Hello world application, building applications like student's Registration form/main page of website is recommended)

Running your project:-

In your terminal, launch your Express project.

```
npm start
```

It will display

```
server listening on port : 3000
```

open your web browser and navigate to http://localhost:3000

Conclusion:-

Thus we have created a NodeJS Application which serves a static Website.