U. Thanmaya
18K81A0592
CSCB

WD

Assignment No. 4 Unit No. 4 Date :

1. Explain the process of Node.js file system for the following operations
(a) open a file (b) Read file (c) write a file

Ans Node.js includes fs module to access physical file system. The fs module is responsible for all the asynchronous or synchronous file I/O operations

(a) open a file:-

Alternatively, you can open a file for reading or writing using fs.open() method

fs.open(path, flags[mode], call back)

parameter Description:-

path:- Full path with name of the file as a string

flag:- The flag to perform operation

mode:- The mode for read, write (or) read write. Default to 0666 read write

callback:- A function with two parameter err and fd. This will get called when file open operation complete computer

(b) Read a file:-

Use fs.readfile() method to read the physical file asynchronously

fs.read file [File name [option] call back]

Ex:- Read a file

Copy

Var fs = require('fs');

fs.readfile("test file.txt" function(err, data){

if(err) throw err;

console.log(data);

C. writing a file:-

Use fs.writeFile() method to write data to a file. If file already exists then it over writes the existing content otherwise it creates a new file and writes data into it

fs.writeFile(filename, data[options], callback)

Ex: creating & writing file

copy

```
var fs = require('fs');
fs.writeFile('test.txt', 'Hello world!', function(err){
    if(err)
console.log(err);
    else
    console.log('write operation complete');
});
```

2 Discuss about Node.js Event Emitter with example.

Ans Node.js allows us to create and handle custom events easily by using events module. Event module raise and handle custom events

Ex:- Raise and Handle Node.js events

copy

```
var events = require('events');
var em = new events.EventEmitter();
em.on('First Event', function(data));
});
```

In the above example, we first import the 'events' module and then create an object of Event emitter class

we then specify event handler function using on() function. The on() method requires name of the event to handle and callback function which is called when an event is raised.

The emit() function raises the specified event first parameter is name of the event as a string and then arguments. An event can be emitted with zero or more arguments. you can specify any name for a custom event in the emit() function.

3. Explain the process of express.js web application?

Ans Express js provides an easy way to create web servers and render html pages for different HTTP requests by configuring routes to your application

web server:

app.js : Express.js webserver

```
var express = require('express');
var app = express();
var server = app.listen(8000, function() {
    console.log('Node server is running');
});
```

The app object includes methods for routing HTTP request, configuring middleware, reading HTML views and registering a template

Engine

The app.listen() function creates the Node.js module using require() function. The express module returns a function. The function returns an object which can be used to configure express application

Run the above example using node.app.js command and point your browser to http://localhost:8000. It will display cannot Get because we have not configured any routes yet.

Assignment No. Unit No. Date :

4) Explain the saving static resources in node.js?

In this section, you will learn how to save static resources like images, CSS, javascript or other static files using express and node static module.

Save static Resources using Express.js.

It is easy to save static files using built-in images in express js called express static

Server.js

copy

```
var express = require ('express');
var express1 = express ();
app.use (express.static (__dirname + 'public'));
var server1 = app.listen (5000);
```

Example : Server resources from different folders

Copy

```
var express = require ('express');
var app = express ();
app.use (express.static ('public'));
app.use ('images', express.static (__dirname+'/image
var server = app.listen (5000);
```

Example:- setting virtual path.

Copy

```
app.use('/resource', express.static(__dirname + '/image'));
```

Serve static Resources using node.static module:-

Example:- saving static resources using node-static

Copy

```
var http = require('http');
var nstatic = require('node-static');
var fileServer = function (req, res) {
    fileServer.serve(req, res);
}).listen(5000);
```