

Name :- Kubareya Jaydip JK.

Roll No :- 115007

Subject :- OSWD Computer Application
(Internal exam PUPET)

Class :- M.Sc.(I.C.T)

Semester :- 3rd

SPIID :- 2021018631

Q1 Explain node.js modules with examples: fs, http

→ modules in node.js is a simple or complex functionality organized in single or multiple JavaScript file which can be reused through the node.js application.

→ each module in node.js has own context so it's can not interfere with other modules or pollute global scope.

→ node.js module type

- 1) Core module
- 2) Local module
- 3) Third Party module.

→ 1) core module is

Node.js is a light weight framework. The core module include base minimum function of node.js. The core modules are completed into its binary distribution and load automatically when nodejs process start.

(1) FS :-

The node.js file system module allows you to work with the system on your computer.

To include the file system module, use the require methods.

```
var fs = require('fs');
```

```
var fs = require('fs');
```

(2) HTTP :-

Node.js has a built-in module called HTTP, which allows Node.js to receive data over the HTTP/1.1 protocol.

→ To include HTTP module to require methods.

```
var http = require('http');
```

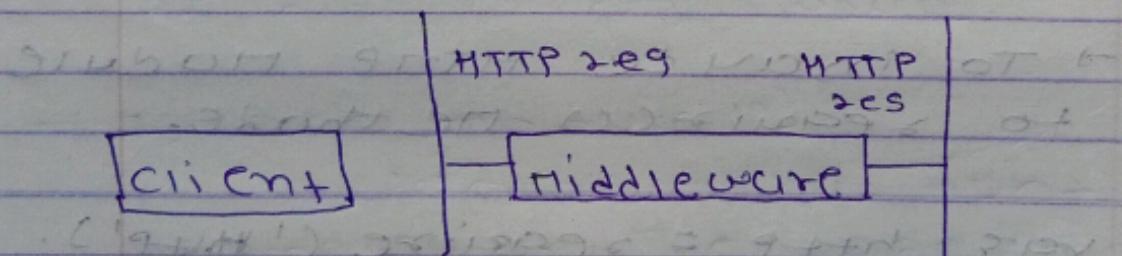
```
var http = require('http');
```

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World');
});
```

Q5-2 what is middleware in express. Explain with example.

→ express.js is a routing and middleware framework for handling the different routing of the web page. and it work the request and response cycle.

→ middleware has access to the request object and response object, and it can process the request before the server send a response. An express based application is a series of middleware function calls.



→ Advantages of using middleware.

- 1) middleware can process request object multiple time before the server work for the request.

- (2) Middleware can be used to add logging and authentication functionalities.
- (3) Middleware improve client side rendering performance.
- (4) Middleware using for setting some specific HTTP headers.
- (5) Middleware helps to optimized and better performs.

→ SIMPLE :-

```
app.get (path, (req, res, next)
         => { y, (req, res) => {} })
```

→ example :-

```
const express = require ('express')
const app = express ()
const port = process.env.PORT || 3000
```

```
app.get ('/ ', (req, res) => {
  res.send ('<div>
    <h1> welcome </h1>
    </div>
  y);
```

```
app.listen (port, () => {
  console.log ('Listening to
  port ${port}');
```

```
y);
```

Q-3 Explain Super Global Array
in PHP with example.

→ Type of super global array

- 1) \$GLOBALS
- 2) \$_SERVER
- 3) \$_REQUEST
- 4) \$_POST
- 5) \$_GET
- 6) \$_FILE
- 7) \$_ENV
- 8) \$_COOKIE
- 9) \$_SESSION

→ 1) \$_SERVER :-

$\$_{SERVER}$ is a PHP
super global array which
holds information about
headers, paths and script
location

ex:-

```
=
```

```
<?php
```

```
echo $SERVER['PHP-SELE'];
```

```
echo "BR";
```

```
echo $SERVER['SCRIPT_NAME'];
```

```
?>
```

Output :-

```
demo | demo-server.php
```

35.194.26.4,

2) PHP \$ - Request :-

\$ - Request is PHP super global array to use to collect data after submitting form.

ex :- <html>

```
<body>
<form method="post", action=
    "<?php echo $-secret[
        'php-self'];?>">
    NAME : <input type="text"
            name="fname">
    <input type="submit">
</form>
```

```
<?php
if ($-secret['request-method'] == "POST") {
    $name = $-Request['fname'];
    if (empty($name)) {
        echo "name is empty";
    } else {
        echo $name;
    }
}
?>
</body>
</html>
```

3) \$-POST

PHP \$-POST is PHP super global array which is used to collect form data after submitting an HTML form with method = "post".

\$-POST is also used to pass variable.

```
<html>
= <body>
<form method = "post" action =
    <?php echo $_SERVER['PHP_SELF'];
    ?>
NAME: <input type = "text"
        name = "fname">

<input type = "submit">
</form>
<?php
if($_SERVER['REQUEST_METHOD'] == "post"){
    $name = $_POST['fname'];
    if(empty($name)){
        echo "name is empty";
    } else {
        echo $name;
    }
}
```

Q 4 what is git? explain any 3 git commands?

- Git is a DevOps tool used for source code management. It is free and open source version control system use to handle small to very large project efficiently. Git is used to tracking change the source code, enabling multiple developer to work together.

→ git config command :-

This command config the user. The git config command is the first of and necessary command use to git command. This command to set other name and email address to be used your commit.

S 3 \$ git config --global user.name "Jagj" \$ git config --global user.email "jagjiani@gmail.com"

→ git clone command

This command is used to make a copy of repository from the existing URL. If I want local copy of my directory form the repository URL.

Syntax :-

\$ git clone URL

3) git add command,

This command is used to add one or more file to staging area.

Syntax :-

\$ git add filename

↳ To add more than one file

\$ git add *

Q:- 5 List and explain npm utilities.

→ 1) create

2) standard

3) benchmark

4) nodemon

5) iocatetunnel

6) now

7) davide

8) nvm

9) npm

→ 1) Create :-

Create node app is a project of facebook to spin up a application with node just command

Syntax :-

npm install create-node-app

-- global

2) Standard :-

Standard is a pre-pckaged linter the zero config choice to make the just the standard style.

Syntax :-

npm install standard --global

(3) branch-diff :-

branch diff is a CLI
diff two branches in a
git repository and generate
list of the difference
between two passed branch.

Syntax :-

npm install branch-diff --global

(4) nodemon :-

nodemon is a small
utility that will automate
restarting a node.js application
when it detects file change
in the directory run in.

Syntax :-

npm install nodemon --global

(5) localtunnel :-

localtunnel is helpful
model that will expose an
application running on
localhost with a single
command. These are some
examples of localtunnel.

Syntax :-

npm install localtunnel --global

(b) now :-

now is small service from Zeit that allows you to deploy node app, socket file and static site.

Syntax :-

npm install now --global

(c) david :-

Within david - is utility to let you know when a project dependency are out of date, and provided a way to update and out date module.

→ npm install david --global

Q6 write a program in express
to display GEGY in a
view template.

→ list.ejs

<html>

<head>

<meta charset = "UTF - 8" >

<meta name = "viewport" >

content = "width = device"

initial-scale = 1.0 >

<meta http-equiv = "X - UA -
compatible" content = "ie =
edge" >

<title> To do list </title>

<body>

<h1> <%= which day %> </h1>

<% for (var i = 0; i < news;
i++) { %>

 <% = newList[i].cm %>

<% } %>

<form class = "new item" >

action = "/" method = "post" >

```
<input type="text" class=""  
name="newItem" placeholder=  
"Add to list"/>
```

```
<button type="submit" name=  
"button">Add  
</button>
```

```
</form>  
</body>  
</html>
```

⇒ app.js

```
const bodyParser = require('body-  
parser');
```

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

```
var listofItem = ["something list"]
```

```
app.set('view engine', 'ejs');
```

```
app.use(express.static('public'));
```

```
app.use(bodyParser, () =>  
{});
```

```
app.get('/', function (req, res) {  
var today = new Date();
```

```
var options = {
```

weekday "'long',
day is 'nemonic',
months' long'

var day = today.toLocaleDateString('en-US', options);

res.send(list, { whichDay: day,
newList: item});

});

app.post('/', function (res, req) {

var item = req.body newItem;

list.push(item);

res.redirect('/');

});

Q8-7 design a user registration
form and displaying all details
on form submit in PHP

→ 2) registration.php

<body>

<?php

\$fullname = \$email = \$password = "";

if (\$HTTP_SERVER["REQUEST-METHOD"]
== "POST") {

\$fullname = test-input(\$POST
[name]);

\$email = test-input(\$POST
[email]);

\$password = test-input(\$POST
[password]);

?>

function test-input(\$datei) {

\$datei = trim(\$datei);

\$datei = stripslashes(\$datei);

\$datei = htmlspecialchars(\$datei);

\$datei = htmlspecialchars(\$datei);

return \$datei;

?>

→ using POST method.

```
<form method="post" action=
<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?
>>>

<div class="container">
<center><h1>Form </h1></center>

<hr>

<label><b>fullname</b><label>
<input type="text" name="name"
placeholder="full name"
size="50" required>

<label for="email"><b>email</b><label>
<input type="text" placeholder=
"enter email" name="email"
required>

<label for="password"><b>password</b><label>
<input type="password" name=
"password" placeholder=
"password" required>

<b><button type="submit"> Register </button>
</form>
```

<?php

```
echo "<h2>Your input:</h2>";  
echo $full_name;  
echo "<br>";
```

```
echo $email;  
echo "<br>";
```

```
echo $password;  
echo "<br>";
```

?>

```
<body>  
<html>
```