Experiment 6

Aim: Node.Js: Installation and Configuration, Callbacks, Event loops, Creating express app.

Theory:

Installation and Configuration:

Step-1: Downloading the Node.js '.msi' installer. The first step to install Node.js on windows is to download the installer.

Step-2: Running the Node.js installer. Now you need to install the node.js installer on your PC. You need to follow the following

steps for the Node.js to be installed:

- Double click on the .msi installer. The Node.js Setup wizard will open.
- Welcome To Node.js Setup Wizard. Select "Next"
- After clicking "Next", End-User License Agreement (EULA) will open. Check "I accept the terms in the License Agreement". Select "Next"
- Destination Folder. Set the Destination Folder where you want to install Node.js & Select "Next"
- Custom Setup. Select "Next"
- Ready to Install Node.js. Select "Install". Click "Finish".

Step 3: Verify that Node.js was properly installed or not. To check that node.js was completely installed on your system or not, you can run the following command in your command prompt or Windows Powershell and test it:

C:\Users\Admin> node -v

If node.js was completely installed on your system, the command prompt will print the version of the node.js installed.

Step 4: Updating the Local npm version. The final step in node.js installed is the updation of your local npm version(if required) – the package manager that comes bundled with Node.js. You can run the following command, to quickly update the npm

npm install npm –global // Updates the 'CLI' client

Code:

Callback Blocking Mode:

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

var filedata = fs.readFileSync("inputfile1.txt");

console.log(filedata.toString());
```

```
console.log("End of Program execution");
```

Output:

```
hello

End of Program execution
```

Non-Blocking Mode:

```
var fs = require("fs");
fs.readFile("inputfile1.txt", function (ferr, filedata) {
  if (ferr) return console.error(ferr);
  console.log(filedata.toString());
});
console.log("End of Program execution");
```

Output:

```
End of Program execution
hello
```

Event Loop:

```
var events = require(" events ");
// Create an eventEmitter object
var eventEmitter = new events.EventEmitter();
// Create an event handler as follows
var connectHandler = function connected() {
  console.log("connection succesful.");
  // Fire the data_received event
  eventEmitter.emit("data_received");
};
// Bind the connection event with the handler
eventEmitter.on("connection", connectHandler);
// Bind the data_received event with the anonymous function
eventEmitter.on("data_received", function () {
```

```
  console.log("data received succesfully.");
});
// Fire the connection event
eventEmitter.emit("connection");
console.log("Program Ended.");
```

Output:

```
connection succesful.
data received succesfully.
Program Ended.
```

Express:

```
var express = require("express");
var app = express();
app.get("/", function (req, res) {
  res.send("Hello World");
});
var server = app.listen(8081, function () {
  var host = server.address().address;
  var port = server.address().port;
  console.log("Example app listening at http://%s:%s", host, port);
});
```

Output:

Hello World

Conclusion:

In this experiment, we focused on Node.js, covering its installation, configuration, and key concepts like callbacks and event loops. We also created a basic Express application, showcasing Node.js' role in building server-side JavaScript applications.