Summary on day 24:

Started reading the node.js documents. To include modules we don’t need any further installation. We can use require() function.

After the module is created, the data is transferred over the http. To include http module use require() function as follows:

Var http=require(‘http’)

After including http module, the server should response back to the client, so use createserver() method as follows:

http.createserver(function(req,res))

if the response from the server is displayed in the html file, then use:

res.writeHead(200,{content-type’:’text.html’})

node.js file system:

I have an html file named bot.html. To include this file into the node.js use require method as follows:

var fs=require(‘fs’)

common file system(‘fs’) modules are:

- Read file
- Create file
- Update file
- Delete file
- Rename file

Read file: fs.readfile() is used to read the file on our computer

For eg:I want to include bot.html in node.js then,

Fs.readfile(‘bot.html’,function(err,data)

Create file:fs.appendfile() used to create a new file if the file does not exists.

Example: Fs.appendfile(‘mynewfile.html’,’hellocontent’)

Fs.writefile() used to replace the specified file.

Example: fs.writefile(‘mynewfile.html’,’hellocontent’)

Update file: the file system module has two methods:

Fs. appendfile()

Fs.writefile()

Delete file: To delete a file in the file system use fs.unlink() method.

Example: fs.unlink('mynewfile.html')

Rename file: To rename the file in the file system, use fs.rename() method.

Example: fs.rename('mynewfile.html','myrenamefile.html')

After reading the documents started applied in my project.

Implementing project:

1. Created webpage using html and css and saved as bot.html and bot.css
2. Installed node.js
3. Visual studio(for editing code)
4. Setting up node js environment.

   Npm init –y

   This will create a new package.json file in my project directory. The package.json file directory is created.

5.Installed the following packages for my chatbot:

- Express
- Socket.io
- Nodemon

Npm install express socket.io nodemon

6.Then created a new file named server.js

->this code setup the basic express server and socket.io server for real time communication.

7.created a new file named main.js

->this code setup he connection with the socket.io and sends message when the button is clicked.

Implementing chatbot logic: the code is updated in server.js. this code listens for the message from client and sends response with that chatbot message.

To run the chatbot, in the package.json file directory we need to code as,

“scripts”:

“start”:” nodemon server.js”

In this package.json file directory the error is occurred.

By implementing this project I got understood about socketio,require(),express,nodemon,express static,app.get(),io.on.

Socketio that enables communication between client and server.

Express is a framework used to routing API.

Nodemon which monitors the project directory.

Require() used to import modules.

Express static used to include static files such as images,css files and javascript files.

App.get() when the request is made, then the response will happen

Io.on which connects the server and client.

By 3’0 clock we had Koushik sir class. There he explained about the yesterday’s topic.

Call back function:

In the previous day problem he explained about task1,task2,task3,task4,task5.

In that task once the laundry function is executed then only the laundry dry function gets executed. If the laundry function stops then laundry dry function never gets executed.

Promises: it is similar to call back function but only the syntax gets changed.

.then

Example: clothes laundry.then{

Clothes dry

}

It is easier for the user to understand.

Futures: futures are used to reduce the code in promises.

Asynchronous awake

```
example: async int clotheslaundry
{
Print(0
}
Awake clothes dry();
```