**Node Js Zero to Goal (Week-1 Report)**

**Day-1**

1. **About Node Js**-Javascript runtime (type of rendering engine)  
   -Built-in V8 Javascript Engine
2. **About Rendering Engines**-Apple Safari -> Webkit (Rendering Engine)  
   -MIE ( Microsoft Internet Explorer ) -> Trident (Rendering Engine)  
   -Mozilla Firebox -> Gecko (Rendering Engine)  
   -Opera Browser -> Presto (Rendering Engine)
3. **File System Module**-node.js file system module -> to work file system on your poject file  
   declare variable to use nodej.s built-in file system module with **require  
    =>** const fs = require(‘fs’);  
   -to read file => fs.readFileSync(‘file path’, encoding – utf-8);  
   -to create or write new file => fs.writeFileSync(‘file path’, data);  
   -to edit or update file => fs.appendFileSync(‘file path’, data, ‘utf-8’);  
   -blocking synchronous nature

**Day-2**

1. **Linux Permissions CLI (Part-1)**1. **Pwd** => Print working directory  
   2. **ls**=> List directory contents  
   3. **ls -la** => List the content and its information  
   4. **mkdir** => Make or create a folder or directory  
   5. **touch** => to create file  
   6. **vim** => text editor  
   7. **cat** => read or print file content on the screen  
   8. **mv** => move file  
   9. **to rename file** => mv 'oldFileName' 'renameFileName'  
   10. **cp** => copy file to new directory  
   to copy file => cp 'fileName' ../Otherdirectory  
   11. **rm** => remove or delete file  
   12**. rm -rf** => remove directory or folder  
   13. **cd** => change directory
2. **How to create server**- to create server by using node.js built-in **HTTP** module with **require**  
    **=>** const http = require(‘http’);  
   -create server => const server = http.createServer((req,res)=> {}  
   -listen server with port => server.listen(8000:’port number’);
3. **HTTP request methods**1. **GET** - receive data from the sever  
   2. **POST** - send data to the server3. **PUT** - replace or update data4. **PATCH** - replace or update data  
   5. **DELETE** - remove data from the server  
   6. **HEAD** - return metadata about resource on the server  
   7. **OPTIONS** - return a listen of which HTTP methods are supported an allowed8. **TRACE** - use for debugging, diagnostic and troubleshooting9. **CONNECT** - use for connection with a sever-side resource (between client and server)

**Day-3**

1. **Linux File Permissions  
   - 0 = --- ( role permissions )  
   - 1 = --x ( execute permission )  
   - 2 = -w- ( write permission )  
   - 3 = -wx ( write and execute permission)   
   - 4 = r-- ( read permission )  
   - 5 = r-x ( read and execute permission )  
   - 6 = rw- ( read and write permission )  
   - 7 = rwx ( full permission )**
2. **To change file permissions  
   - sudo chomd permissions 'filename'**
3. **Linux Permission CLI (Part-2)  
   tail =>** View and paginate *file* **more =>** displaying one screen at a time in case the file is large  
   **less =>** Show last 10 lines of *file***alias =>** The alias command provides a string value that replaces a command name when it is encountered   
    **=>** alias name = ‘cd ./filePath’
4. Non-blocking Asynchronous nature
5. **Routing, simple api and fill HTML template**

**Day-4**

1. **Parsing Data to HTML template**
2. **Calling another html file => {%Name%}**
3. Passing data into html file with **replace** method **=>** htmlFile.replace(‘{%Name%}’, data);