3) Create a Node.js file that opens the requested file and returns the content to the client. If anything goes wrong, throw a 404 error.

Ans.

```
const http = require('http');
const fs = require('fs');
const path = require('path');

const server = http.createServer((req, res) => {
  // Extract the requested file path from the URL
  const filePath = path.join(__dirname, req.url);

  // Check if the file exists
  fs.access(filePath, fs.constants.F_OK, (err) => {
    if (err) {
      // If the file doesn't exist, send a 404 response
      res.writeHead(404, {'Content-Type': 'text/plain'});
      res.end('404 Not Found');
    } else {
      // If the file exists, read its content and send it in the response
      fs.readFile(filePath, (err, data) => {
        if (err) {
          // If there's an error reading the file, send a 500 response
          res.writeHead(500, {'Content-Type': 'text/plain'});
          res.end('500 Internal Server Error');
        } else {
          // Send the file content in the response
          res.writeHead(200, {'Content-Type': 'text/plain'});
          res.end(data);
        }
      });
    }
  });
});

const port = 3000;
server.listen(port, () => {
  console.log(`Server running at http://localhost:${port}/`);
});
```

In this code:
We create an HTTP server using the createServer method of the http module.

When a request is received, we extract the requested file path from the URL using the path.join method. We use the fs.access method to check if the requested file exists. If it does not exist, we send a 404 response.

If the file exists, we use the fs.readFile method to read its content. If there's an error reading the file, we send a 500 response. Otherwise, we send the file content in the response.

We listen for incoming connections on port 3000, and once the server is running, we log a message to the console.