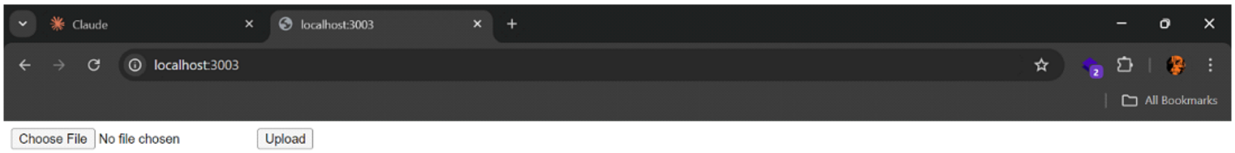
Task

1.Build a basic HTTP server that can handle different routes and HTTP methods (GET, POST, PUT, DELETE).

Program:

```
const express = require('express');
const multer = require('multer');
const path = require('path');
const fs = require('fs');

const app = express();
const port = 3003;


const storage = multer.diskStorage({
  destination: (req, file, cb) => {
    const uploadDir = 'uploads/';
    if (!fs.existsSync(uploadDir)) {
      fs.mkdirSync(uploadDir);
    }
    cb(null, uploadDir);
  },
  filename: (req, file, cb) => {
    cb(null, Date.now() + path.extname(file.originalname));
  }
});

const upload = multer({ storage: storage });


app.get('/', (req, res) => {
  res.send(`
    <form action="/upload" method="post" enctype="multipart/form-data">
      <input type="file" name="file">
      <input type="submit" value="Upload">
    </form>
  `);
});


app.post('/upload', upload.single('file'), (req, res) => {
  if (!req.file) {
    return res.status(400).send('No file uploaded.');
  }
  res.send('File uploaded successfully!');
});
app.listen(port, () => {
  console.log(`Server running at http://localhost:${port}`);
});
```

OUTPUT

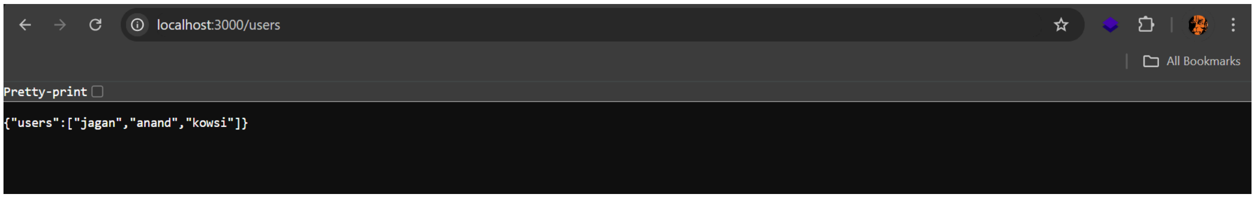


2.Create a server that allows users to upload files and save them to the server's filesystem.

Program:

```
const http = require('http');
const url = require('url');

const server = http.createServer((req, res) => {
  const parsedUrl = url.parse(req.url, true);
  const path = parsedUrl.pathname;
  const trimmedPath = path.replace(/^\/+|\/+$/g, '');
  const method = req.method.toLowerCase();


  const router = {
    'jagan': {
      'get': (req, res) => {
        res.writeHead(200, {'Content-Type': 'application/json'});
        res.end(JSON.stringify({message: 'jagan, World!'}));
      }
    },
    'users': {
      'get': (req, res) => {
        res.writeHead(200, {'Content-Type': 'application/json'});
        res.end(JSON.stringify({users: ['jagan', 'anand', 'kowsi']}));
      },
      'post': (req, res) => {
        let body = '';
        req.on('data', chunk => {
          body += chunk.toString();
        });
        req.on('end', () => {
          res.writeHead(201, {'Content-Type': 'application/json'});
          res.end(JSON.stringify({message: 'User created', data: JSON.parse(body)}));
        });
      },
      'put': (req, res) => {
        let body = '';
        req.on('data', chunk => {
          body += chunk.toString();
        });
        req.on('end', () => {
          res.writeHead(200, {'Content-Type': 'application/json'});
          res.end(JSON.stringify({message: 'User updated', data: JSON.parse(body)}));
        });
      },
      'delete': (req, res) => {
        res.writeHead(200, {'Content-Type': 'application/json'});
        res.end(JSON.stringify({message: 'User deleted'}));
      }
    }
  };


  if (router[trimmedPath] && router[trimmedPath][method]) {
    router[trimmedPath][method](req, res);
  } else {

    res.writeHead(404, {'Content-Type': 'application/json'});
    res.end(JSON.stringify({message: 'Not Found'}));
  }
});

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

OUTPUT:



```
{"users":["jagan","anand","kowsi"]}
```