SE3040 – Application Frameworks
BSc (Hons) in Information Technology
Software Engineering Specialization
3rd Year
Faculty of Computing
SLIIT

2023 - Practical

LAB 03 Introduction to NodeJS and GitHub flow

In this lab, you will learn the basics of Node.js, Git and GitHub flows. You will be introduced to some basic Node.js commands, create a Git repository and work with GitHub to commit, push, and pull changes.

1. Create a new git repository in GitHub, including the Readme file and the .gitignore for node.
2. Clone the repository to the local machine
3. Open the project folder in VS Code
4. Open git bash terminal.
5. Create a new branch named "feature/<your-name>/<feature-name>
6. Open a terminal and type the command "npm init" and add the necessary information and setup it
7. Add a file name "app.js"
8. Add a console log message to the file and commit the changes
9. Push the branch to the github.
10. Use the GitHub official documentation to create a Pull Request on Github for your branch.
11. Pull the changes to the local repository

Repeat the cycle for the following examples

1. Add new feature branch for the each section. Run the code. Push the branch to the remote and merge it and pull the changes

Node JS

1. Read file

```
const fs = require('fs');
fs.readFile('file.txt', 'utf8', function (err, data) {
  if (err) throw err;
  console.log(data);
});
```

2. Write to file

```
const fs = require('fs');
fs.writeFile('file.txt', 'Hello World!', function (err) {
  if (err) throw err;
  console.log('File saved!');
});
```

3. Creating a web server:

```
const http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write('Hello World!');
  res.end();
}).listen(8080);
```

4. Making an HTTP request:

```
const https = require('https');
https.get('https://jsonplaceholder.typicode.com/posts/1', (resp) => {
  let data = '';

  resp.on('data', (chunk) => {
    data += chunk;
  });

  resp.on('end', () => {
    console.log(JSON.parse(data));
  });
}).on('error', (err) => {
  console.log("Error: " + err.message);
});
```

5. Using a module

```
const myModule = require('./my-module.js');
console.log(myModule.myFunction());
```

6. Promises

```
const myPromise = new Promise((resolve, reject) => {
  if (condition) {
    resolve('Success!');
  } else {
    reject('Failure!');
  }
});

myPromise.then((result) => {
  console.log(result);
}).catch((error) => {
  console.log(error);
});
```

7. Async/Await

```
async function myFunction() {
  try {
    const result = await myPromise;
    console.log(result);
  } catch (error) {
    console.log(error);
  }
}

myFunction();
```

Git Collaboration

1. Now pair up with a colleague (preferably sitting next to you) and add them to your repository as a collaborator by referring to this documentation.
2. Now clone their repository to your computer.
3. Then as with step 1, create a branch and introduce a small change.
4. Push the branch to the remote repository.
5. Use the GitHub official documentation to create a Pull Request on Github for your branch. Then merge the pull request created by your friend to the main by referring to this official documentation.