FullStack App Guide

Front-End: React
Back-End: NodeJS (Express)
Database: MySQL
Version Control: Git
Style: Material UI (MUI)
Tools: XAMPP, Cmder, MySQLWorkbench, GitHub, GitHub Desktop, VSCode.
Cmder: https://cmder.app/
VSCode: https://code.visualstudio.com/

Cmder:

cd	Change directory
.	Represents the current directory
..	Represents the directory that contains the current directory
/	Separator
ls	Show directories in the working directory
pwd	Print working directory
mkdir	Create a new directory

React:

Create a react app: https://legacy.reactjs.org/
Need: NodeJS LTS: https://nodejs.org/es

VSCode:

File: package.json
File: src/index.js

React Components:

Function Component And Class Component

Function Component:

```
function App() {
  return ();
}
export default App;
```

Class Component:

```
import React from 'react';
class App extends React.Component {
  constructor(props){
    super(props);
    this.state = {}
  }
  render() {
    return ();
  }
}
export default App;
```

- ➔ Using React:
  - ◆ state:
    - this.state = {}
    - this.setState({})
  - ◆ functions and arrow functions
    - functionName(){}
      - ○ this.functionName.bind(this)
    - arrowFunctionName = () => {}
      - ○ this.arrowFunctionName
  - ◆ React.Fragment

Extend: LifeCycle Methods*
(https://legacy.reactjs.org/docs/react-component.html#componentdidmount)

NodeJS(Express):

Create a express app: https://expressjs.com/en/starter/installing.html

Nodemon :

npm install --save-dev nodemon
script : "start": "nodemon index.js"
npm start (to start with nodemon)

CORS:

const cors = require('cors')
app.use(cors())

React:

Connect with Back-End (Fetch):

fetch("http://localhost:3001")
.then(response => response.json())
.then(data => console.log(data));

Mysql:

Install XAMPP: https://www.apachefriends.org/es/index.html

Install MySQLWorkbench: https://www.mysql.com/products/workbench/

(or) Use PHPMyAdmin: http://localhost/phpmyadmin/

DataBase:

create database crud;

use crud;

create table todo_list (id int not null auto_increment, data longtext not null, primary key (id));

INSERT todo_list (data) VALUES ('{"title": "Title Example", "description": "Description Example", "image": "https://upload.wikimedia.org/wikipedia/commons/b/b6/Image_created_with_a_mobile_phone.png"}');

NodeJS(Express):

Install mysql : npm install mysql

Create Connection:

```
const mysql = require('mysql');
const connection = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: ""
});
connection.connect(function(err) {
  if (err) {throw err;}
  else{
      console.log("Connected!");
      app.listen(port,function(){
          console.log("servidor del api rest escuchando en HTTP://localhost:"+port)
      });
  }
});
```

Get data from myql:

```
function getAll(req, res) {
  // Create connection
  const conecction = mysql.createConnection({
      host: "localhost",
      user: "root",
      password: "",
      database: "crud"
  });
  conecction.connect(function(err) {
      // Connection Failed
      if (err) res.status(200).send({"conecction": "Database connection
error"});
      // Get todo_list data
      conecction.query("select * from todo_list", function (err,
result) {
          // String to JSON
          const json = JSON.stringify(result);
          // Send JSON
          res.send(json);
      });
      // End connection
      conecction.end();
  });
}
// Rute
app.get('/todo_list', getAll)
```

React:

Get and show data

Generate Post Method:

```
  handlePostData = () => {
    let dataToSave = { title: "Save Title", description: "Save
Description", image:
"https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSu3EMsr-8AjBp-m_
febOvDMpIQowfMaMwhDw&usqp=CAU" }
    fetch('http://localhost:3001/todo_list', {method: "POST", mode:
"cors", headers: { "Content-Type": "application/json"},  body:
JSON.stringify(dataToSave)})
        .then(response => response.json())
        .then(data => {console.log(data)});
  }
```

NodeJS(Express):

Install body-parser: npm install body-parser

```
const bodyParser = require("body-parser");
app.use(bodyParser.json());

// Function to post
function addData(req, res) {
  // Params
  const body = req.body;
  // See params TODO Temporal
  console.log(body);
  let dataToSave = JSON.stringify(body)
  // Create connection
  const conecction = mysql.createConnection({
      host: "localhost",
      user: "root",
      password: "",
      database: "crud"
  });
  conecction.connect(function(err) {
      // Connection Failed
      if (err) res.status(200).send({"conecction": "Database connection
error"});
      // Post user
      conecction.query("INSERT todo_list (data) VALUES
(?);",[dataToSave], function (err, result) {
          // if error
          if(err){res.status(200).send(err);}
          // if not
          else{res.status(200).send(result);}
      });
      // End connection
      conecction.end();
  });
}
// Rute
app.post('/todo_list', addData)
```

Git/GitHub:

Install GIT: https://git-scm.com/
Install GITHUB Desktop: https://desktop.github.com/
Commands: https://git-scm.com/docs

git init	Create an empty Git repository or reinitialize an existing one
git status	Show the working tree status
git add	Add file contents to the index
git add **.**	Add **all** files contents to the index
git commit	Record changes to the repository
git commit -m “Some message”	Record changes to the repository **with message**
git clone	Clone a repository into a new directory
git branch	List, create, or delete branches
git checkout	Switch branches or restore working tree files
git merge	Join two or more development histories together
git log	Show commit logs

Add all files to git and commit
Create a github repository: https://github.com/new or use Desktop App.
View: .gitIgnore
Use the same repository for both projects
Clone the repository and run the apps.

```
npm install
npm start
```

MaterialUI:

Install: https://mui.com/

React:

npm install @mui/material @emotion/react @emotion/styled
Add Material UI Style to Card component
Add Some Styles:

```
style={{display: 'flex', flexDirection: 'row', justifyContent:
'space-around', flexFlow: 'wrap'}}
```

GitHub Example Repository:
https://github.com/MACGalaviz/fullstack-app-guide/tree/master