**Instalation walkthrough & testing**

Working on NodeJs v20.15.1

Create **main** directory for Project directory

TERMINAL 1 (BACK-END)

Located in **main** directory

Create **flask-server** folder and open it

Create server.py file

TERMINAL 2 (FRONT-END)

Located in **main** directory

Type following command

>npx create-react-app client

Wait until finished

Install axios

>npm install axios

~~Install babel ?? TEST~~

~~>npm install babel~~

Install react-router-dom to be able to use Link

>npm install react-router-dom

Modify \scr\index.js file and comment React.StrictMode component, only use it during development to catch potential issues early

//   <React.StrictMode>

    <App />

//   </React.StrictMode>

TERMINAL 1 (BACKEND)

Creating virtual environment inside **flask-server** directory on **venv** directory

>python -m venv **venv**

Activating virtual environment

>.\venv\Scripts\activate

Installing Flask

>pip install Flask

Installing Flask-SQLAlchemy is an extension for Flask that adds support for SQLAlchemy to your application.

>pip install Flask-SQLAlchemy

Installing flask-marshmallow for beautiful APIs

>pip install flask-marshmallow

Instaling marshmallow-sqlalchemy

>pip install marshmallow-sqlalchemy

Installing flask-cors to avoid CORS errors when using APIs

>pip install flask-cors

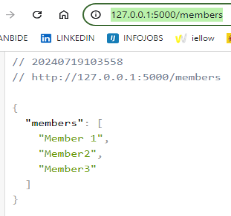
Starting Flask server 1 way

>python server.py

Starting Flask server better way, the server file has to be named as app.py

>flask run -- debug 🡨 debug option autoruns after app.py file changes

In browser type <http://127.0.0.1:5000/members> and the member array will be showed



DELETING unnecessary files in client/SRC

App.test.js

Index.css

Logo.svg

Inside index.js delete file

import ‘./index.css’

CONNECTING BACK-END TO FRONT-END

In **client** directory modify **package.json** file adding on beginning lines after **“private”: True,**

“proxy”: “http://localhost:5000”,

MODIFYING **client/src/App.js** file deleting all content filling file with following code

|  |
| --- |
| import React, {useState, useEffect} from 'react'  export default function App() {    const [data,setData] = useState([{}])      useEffect(()=> {      fetch("/members").then(        res => res.json()      ).then(        data => {          setData(data)          console.log(data)        }      )    }, [])    return (      <div>        ESTO ES UN COMIENZO !!!        <br/>CONECTADO BACK-END CON FRONT-END!!!        <br/>ya queda menos .... :\_(      </div>    )  } |

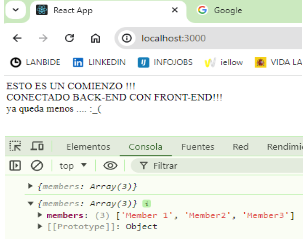
TERMINAL 2 (FRONT-END)

Running FRONT-END server

>npm start

Better for development and autoupdating after code changes >npm run dev

In browser type <http://localhost:30000> and the member array will be showed



-.sql file contains SQL commands to create posts table

Creating schema.sql

-create a python file to generate SQL lite .db database based on schema.sql

Creating initdb.py

Executing initdb.py with “python initdb.py” create the database.db database with 2 records

-creating app.py create a database connection and create Flask route to use that connection

|  |
| --- |
| **from** flask **import** Flask    app **=** Flask(\_\_name\_\_)    @app.route('/user/<username>')  **def** show\_user(username):      # Greet the user  **return** f'Hello {username} !'    # Pass the required route to the decorator.  @app.route("/hello")  **def** hello():  **return** "Hello, Welcome to GeeksForGeeks"    @app.route("/")  **def** index():  **return** "Homepage of GeeksForGeeks"    **if** \_\_name\_\_ **==** "\_\_main\_\_":      app.run(debug**=**True) |

|  |
| --- |
| **from** flask **import** Flask    app **=** Flask(\_\_name\_\_)    **def** show\_user(username):      # Greet the user  **return** f'Hello {username} !'    app.add\_url\_rule('/user/<username>', 'show\_user', show\_user)    **if** \_\_name\_\_ **==** "\_\_main\_\_":      app.run(debug**=**True) |
| [Flask - Variable Rule - GeeksforGeeks](https://www.geeksforgeeks.org/flask-variable-rule/?ref=next_article)   |  | | --- | | # First we import flask  **from** flask **import** Flask    # initialize flask  app **=** Flask(\_\_name\_\_)    # Display first simple welcome msg  @app.route('/')  **def** msg():  **return** "Welcome"    # We defined string  function  @app.route('/vstring/<name>')  **def** string(name):  **return** "My Name is %s" **%** name    # we run app debugging mode  app.run(debug**=**True) | |

DOWNLOAD sqlite database browser <https://sqlitebrowser.org/dl/>

TO DO

* Fill up schema.sql with all tables info to create them