Setup

Machine Requirements:

- Linux or Windows
- Visual Studio Code
- Internet access
- MySQL Database
- MySQL Workbench
- Postman

Application Setup:

Extract Archive:

- Extract the archive `SigmaCase.zip`

Database Setup:

- Create a new schema on MySQL Database:
    - Schema Name: `sigma_case`
    - Host: `localhost`
    - Port: `3306`
    - User: `root`
    - Password: `Sigma`

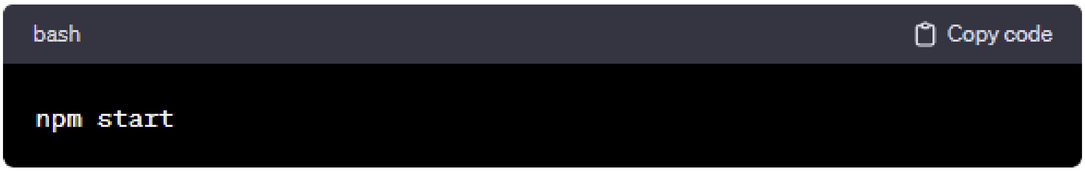
Front-End Application:

Start Front-End Application:

- Open the folder `Front-End` in Visual Studio Code
- Open the Visual Studio Code Terminal
- Execute the following command:
    - npm install –legacy-peer-deps

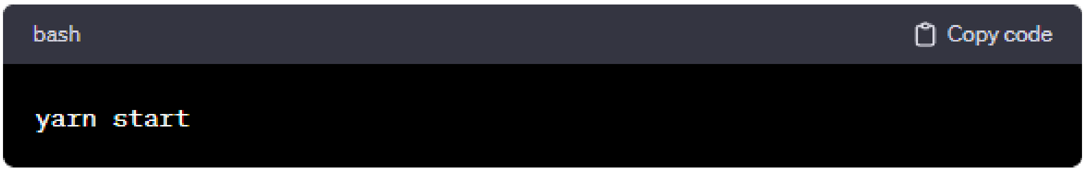
    or

    - yarn install
- Start the application with:



```bash
npm start
```

or

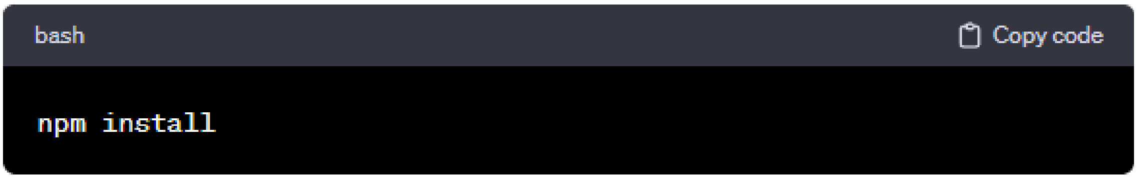


```bash
yarn start
```

Back-End Application:

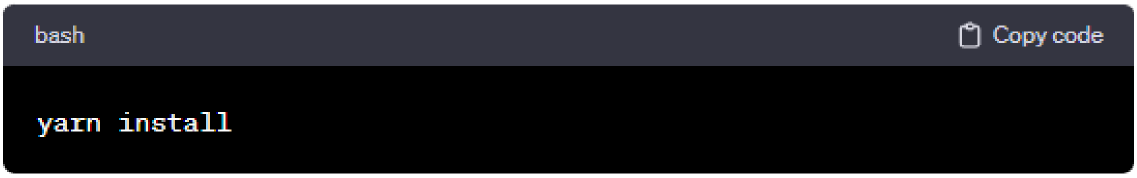
Start Back-End Application:

- Open the env file and change to your credentials if necessary
- Open the folder `Back-End` in Visual Studio Code
- Open the Visual Studio Code Terminal
- Execute the following command:



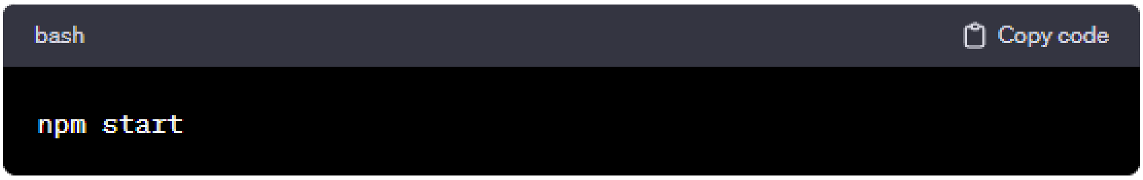
```bash
npm install
```

or



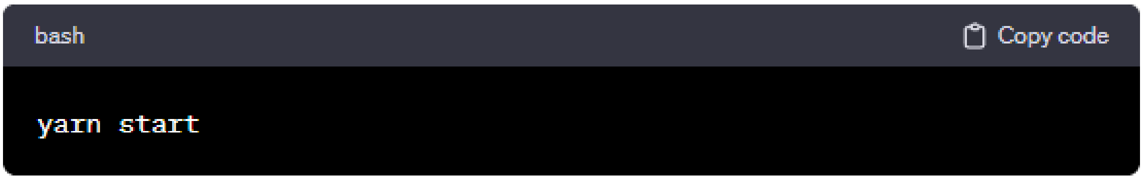
```bash
yarn install
```

- Start the application with:



```bash
npm start
```

or



```bash
yarn start
```

- Create Tables on MySQL Database:
  - Open Postman
  - Create a new HTTP request
  - Paste the following URL: `http://localhost:3001/tables`
  - Click "Send" to create tables
  - Verify if tables were created:

    Table Permissions:

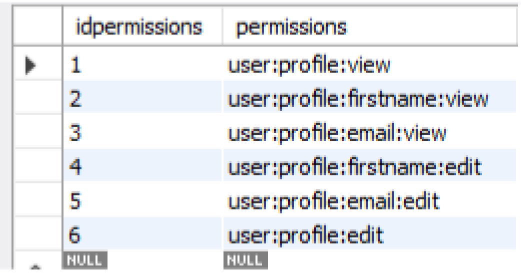
    - ○ Columns: `idpermissions`, `permissions`

    Table User:

    - ○ Columns: `iduser`, `firstname`, `email`

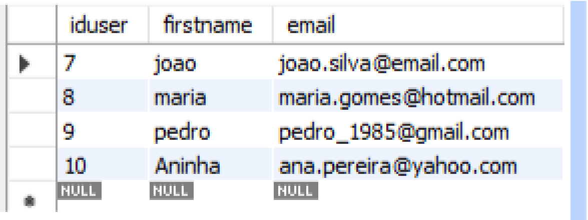
    Table User_Permissions_Rela:

    - ○ Columns: `iduser_permissions_rela`, `iduser`, `idpermissions`
- Create Data for Tests:
  - Table Permissions:
  - You can add automatic
  - Open the Postman and paste this urls and click send with the method Post
    - http://localhost:3001/permissions
    - http://localhost:3001/users
    - http://localhost:3001/permissionsUser
  - Or if you want you can add manually
  - Table Permissions:



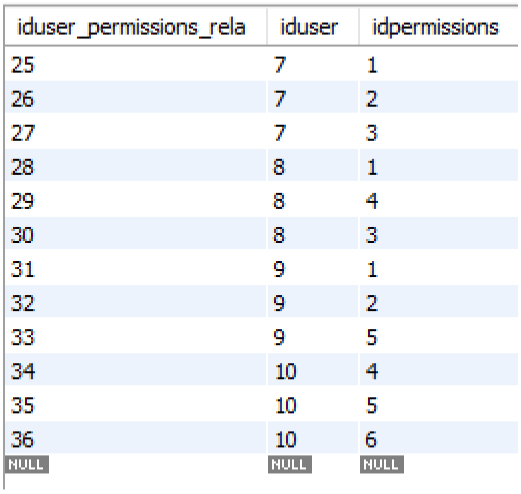
idpermissions	permissions
1	user:profile:view
2	user:profile:firstname:view
3	user:profile:email:view
4	user:profile:firstname:edit
5	user:profile:email:edit
6	user:profile:edit
NULL	NULL

- Table User:



iduser	firstname	email
7	joao	joao.silva@email.com
8	maria	maria.gomes@hotmail.com
9	pedro	pedro_1985@gmail.com
10	Aninha	ana.pereira@yahoo.com
NULL	NULL	NULL

- Table User_Permissions_Rela:



iduser_permissions_rela	iduser	idpermissions
25	7	1
26	7	2
27	7	3
28	8	1
29	8	4
30	8	3
31	9	1
32	9	2
33	9	5
34	10	4
35	10	5
36	10	6
NULL	NULL	NULL

Run the Application

Open Browser:

- Paste `http://localhost:3000` in the browser URL

Use Cases Tests:

Case 1:

- Fill the login input field with `joao.silva@email.com`
- User can just view the profile form in a readonly state.

Case 2:

- Fill the login input field with `maria.gomes@hotmail.com`

- User can edit the name and view the email in a readonly state.

Case 3:

- Fill the login input field with pedro_1985@gmail.com
- User can view the name in a readonly state and edit the email.

Case 4:

- Fill the login input field with ana.pereira@yahoo.com
- User can edit the name and the email.

Case 5:

- Fill the login input field with any email not registered
- User cant access the profile

Run Unit API Test

Open Browser:

- Paste `http://localhost:3000/apiTest` in the browser URL
- Check if all status are "Ok"