

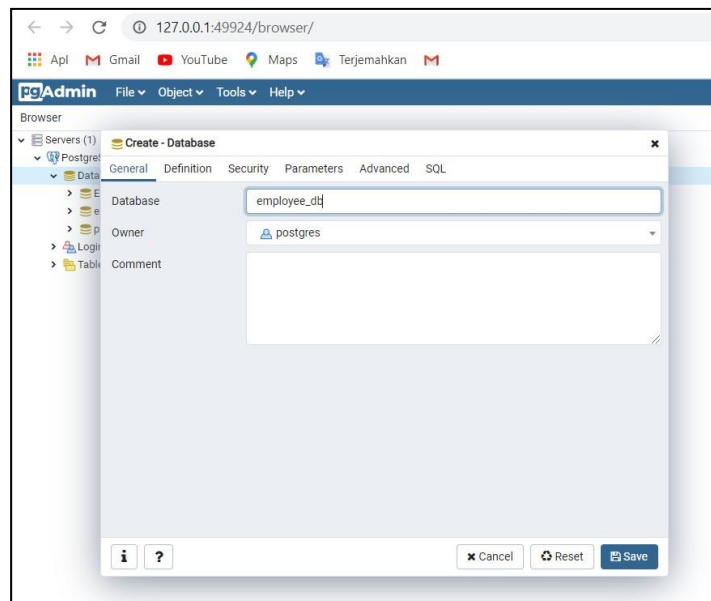
Ahsanul Qalbi – Employee Database Documentation

Github Repository : <https://github.com/AhsanulQalbi/CRUDEmployeeDatabase>

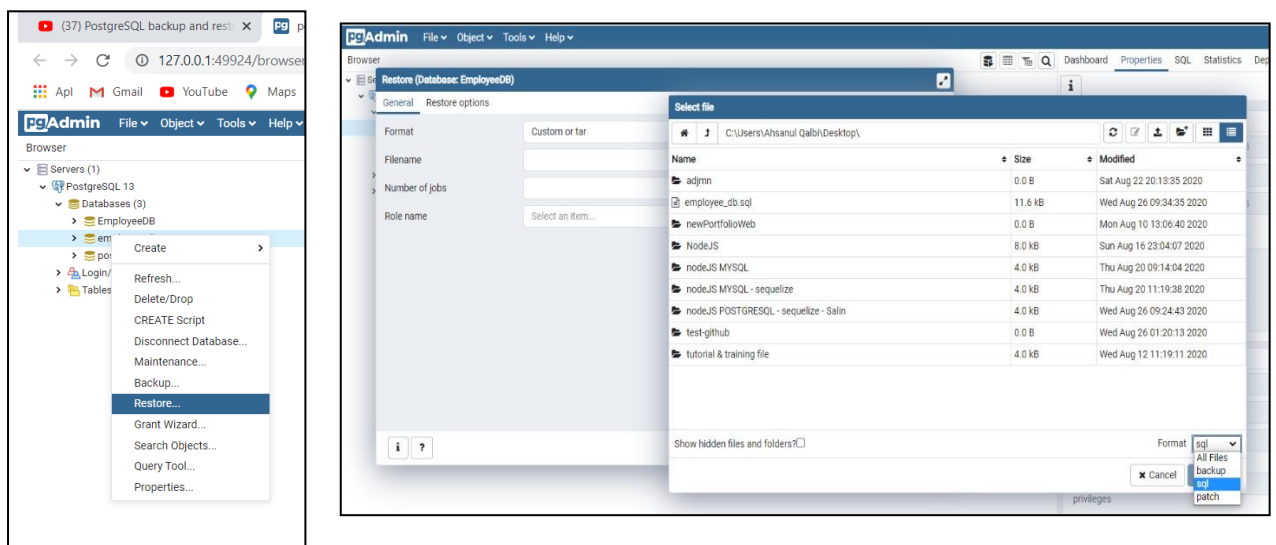
○ How to run the apps locally

Required programs : Node js, PostgreSQL

- Install Nodejs : <https://nodejs.org/en/download/>
- Install PostgreSQL : <https://www.postgresql.org/>
- Set PostgreSQL user : postgres, password : password
- After creating PostgreSQL server, create employee_db database on pgAdmin 4



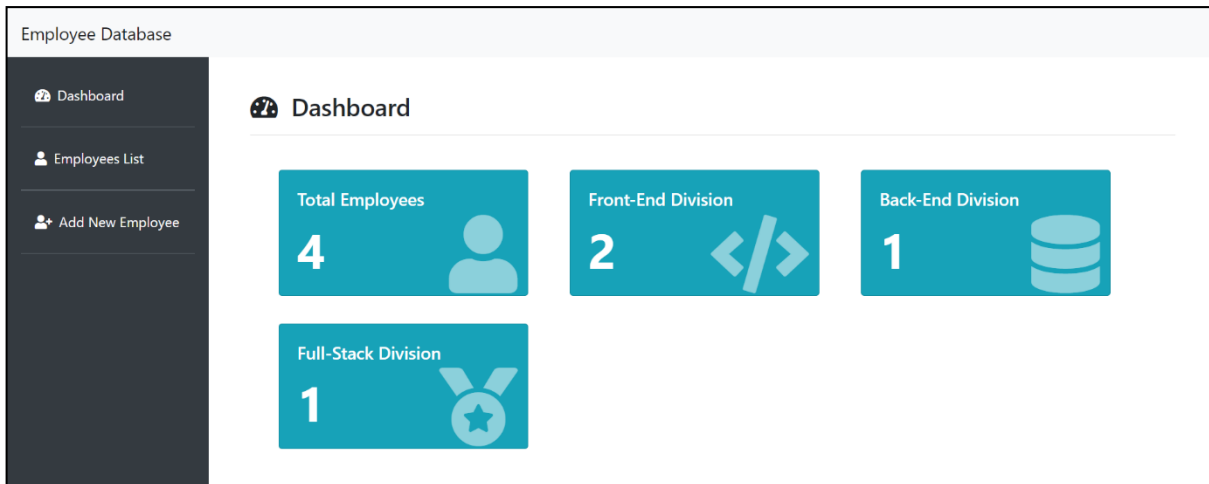
○ Restore “employee_db.sql” backup database file (optional)



- Install Nodemon (optional) : (in command prompt) npm install nodemon
- Run the apps : (in command prompt) nodemon server.js OR node server.js
- Go to browser : <http://127.0.0.1:3000/> to access webpage

- Website Pages

- Main page, go to browser : <http://127.0.0.1:3000/>



This page is the summary of the employee database, showing total employee from each divisions.

- Employee List Page, go to browser : http://127.0.0.1:3000/employee_list

The screenshot shows the 'Employee List' page. It has a sidebar with 'Dashboard', 'Employees List' (selected), and 'Add New Employee'. The main area has a title 'Employee List', a 'Show 10 entries' dropdown, and a search bar. Below is a table with 4 employees. Each row has buttons for 'Detail', 'Update', and 'delete'.

First Name	Last Name	Email	Address	Division	Action
Daniel	Ricciardo	daniel3@gmail.com	Angas Gardens, Sydney, Queensland, Australia,	Front-End	Detail Update delete
Kimi	Raikkonen	kr7@gmail.com	Furuhelm Street, Espoo, Southern Finland, Finland,	Full-Stack	Detail Update delete
Lewis	Hamilton	lewis44@gmail.com	Earlham Road, Oxford, London, United Kingdom,	Front-End	Detail Update delete
Mika	Hakkinen	mika1@gmail.com	Furuhelm Street, Akaa, Western Finland, Finland,	Back-End	Detail Update delete

Showing 1 to 4 of 4 entries

Previous 1 Next

This page is used to view, change, and delete employee data. “detail” button is used to display detailed employee data. “update” button is used to update employee data, and the “delete” button is used to delete an employee data.

- Add New Employee Page , go to browser : http://127.0.0.1:3000/add_employee

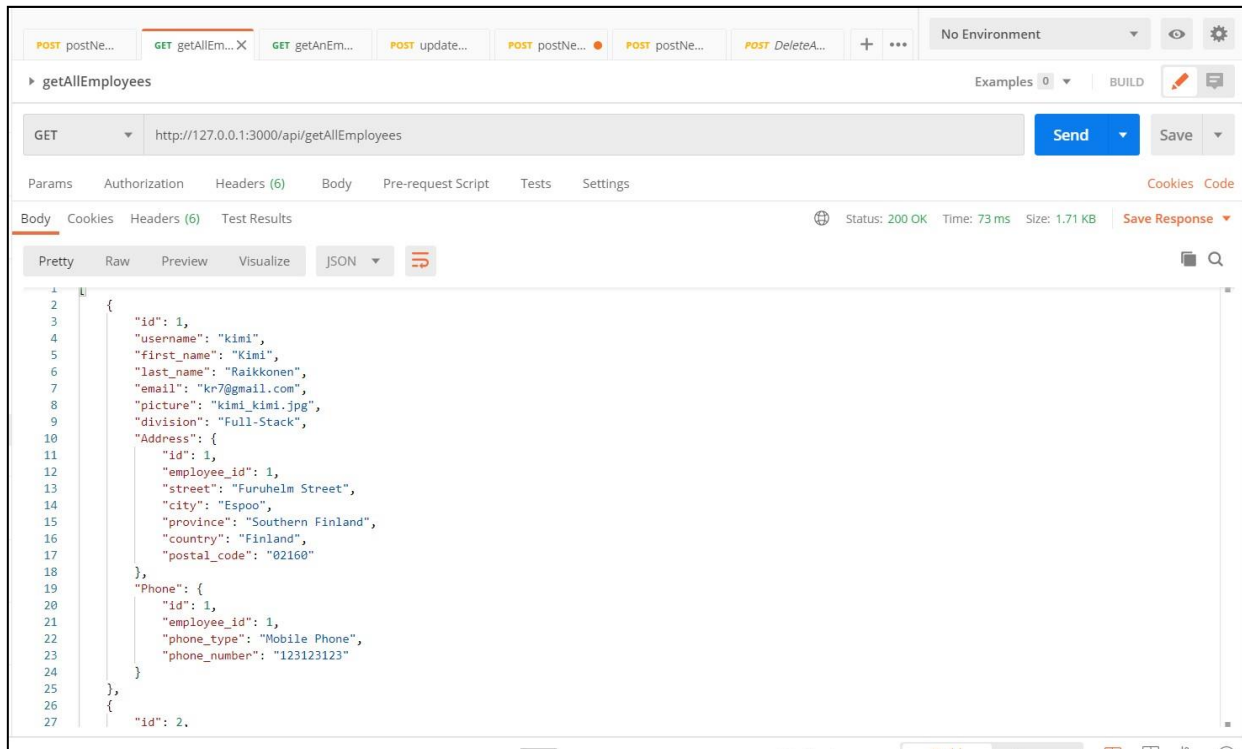
The screenshot shows the 'Add New Employee' page of an 'Employee Database' application. On the left is a dark sidebar with navigation links: 'Dashboard', 'Employees List', and 'Add New Employee'. The main content area has a title 'Add New Employee' with a person icon. Below the title are several form fields: 'Username' (with a placeholder 'Enter Username' and a note 'Please enter your unique username (required)'), 'First Name' (with a placeholder 'Enter Your Firstname' and a note 'Please enter your firstname (required)'), 'Last Name' (with a placeholder 'Enter Your Lastname' and a note 'Please enter your lastname (if existed)'), 'Email Address' (with a placeholder 'Enter Your email' and a note 'Please enter your email address (required)'), 'Division' (with radio buttons for 'Front-End' (selected), 'Back-End', and 'Full-Stack'), 'Photo' (with a 'Pilih File' button and a note 'Tidak ada file yang dipilih'), and 'Street Address'. Each field has a light blue border and a small error message below it.

This screenshot shows the bottom section of the 'Add New Employee' page. It includes the following form fields: 'Province Address' (placeholder: 'Enter Province Address', note: 'Please enter province address (required)'), 'Country Address' (placeholder: 'Enter Country Address', note: 'Please enter country address (required)'), 'Postal Code' (placeholder: 'Enter Postal Code', note: 'Please enter postal code (required)'), 'Phone Number' (placeholder: 'Enter Phone Number', note: 'Please enter phone number (required)'), and 'Phone Type' (radio buttons for 'Mobile Phone' (selected) and 'Home Phone'). At the bottom left of the form area is a blue 'Submit' button.

This page is used to add new employee data. When the submit button is clicked and the user has input all required data, the new employee data will be stored in database.

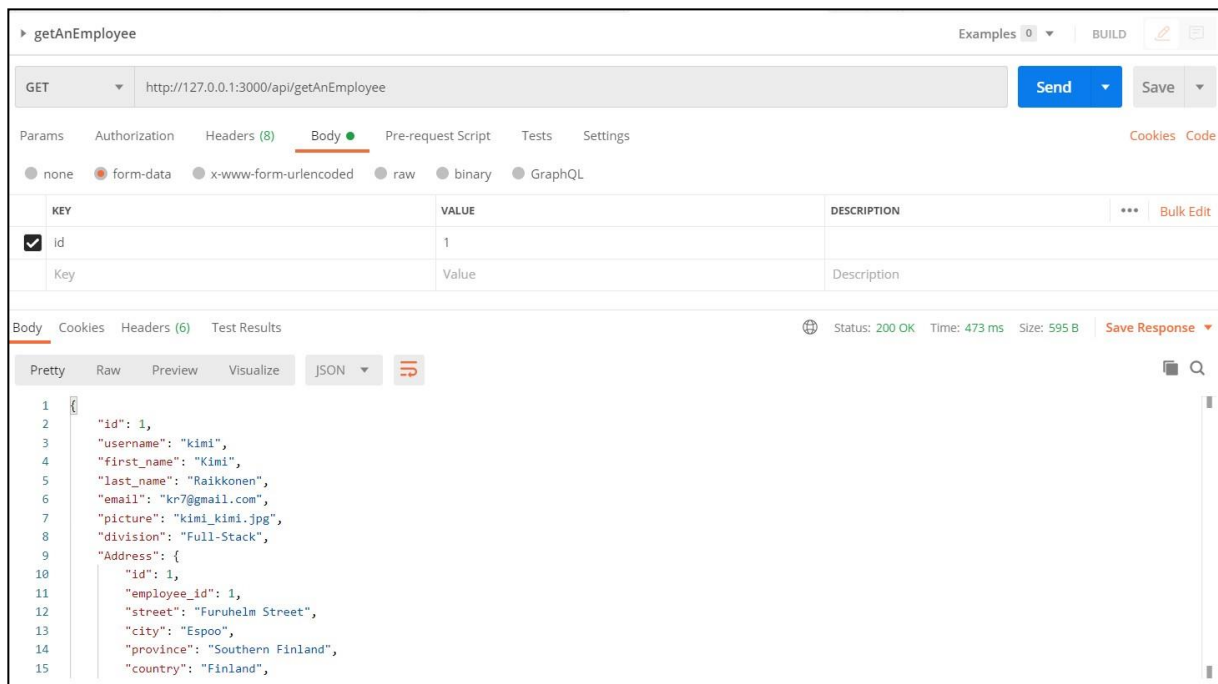
○ API Documentation

- Get All employee data : <http://127.0.0.1:3000/api/getAllEmployees>



This API is used to get all employee data including their address and phone data.

- Get an employee data : <http://127.0.0.1:3000/api/getAnEmployee>



This API is used to get a single employee data including their address and phone data.

- Post new employee data : <http://127.0.0.1:3000/api/employee>

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:3000/api/employee`. The request body is configured as form-data with the following parameters:

Key	Value	Description
<input checked="" type="checkbox"/> email	Lando4@gmail.com	
<input checked="" type="checkbox"/> username	lando	
<input checked="" type="checkbox"/> picture	lando.jpg X	
<input checked="" type="checkbox"/> street	Albion Street	
<input checked="" type="checkbox"/> city	Oxford	
<input checked="" type="checkbox"/> province	London	
<input checked="" type="checkbox"/> country	United Kingdom	
<input checked="" type="checkbox"/> postal_code	525252	
<input checked="" type="checkbox"/> phone_number	121212120	
<input checked="" type="checkbox"/> phone_type	Mobile Phone	
<input checked="" type="checkbox"/> division	Front-End	

This API is used to add new employee data.

Body params :

- username : STRING, unique
- email : STRING, unique
- picture : Image File
- street : STRING
- city : STRING
- province : STRING
- country : STRING
- postal_code : STRING
- phone_number : STRING
- phone_type : STRING (Mobile Phone / Home Phone)
- division : STRING (Back-End, Front-End, Full-Stack)

- Update an employee data : <http://127.0.0.1:3000/api/updateEmployee>

The screenshot shows a REST client interface with a tab for the `POST updateEmployee` request. The URL is `http://127.0.0.1:3000/api/updateEmployee`. The request body is configured as `form-data` and contains the following parameters:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id	5	
<input checked="" type="checkbox"/> email	LandoNorriss@gmail.com	
<input checked="" type="checkbox"/> username	lando	
<input checked="" type="checkbox"/> picture	lando.jpg	
<input checked="" type="checkbox"/> street	Albion Street	
<input checked="" type="checkbox"/> city	Oxford	
<input checked="" type="checkbox"/> province	London	
<input checked="" type="checkbox"/> country	United Kingdom	

This API is used to update employee data.

Body params :

- id : INTEGER, Primarykey
- username : STRING, unique
- email : STRING, unique
- picture : Image File
- street : STRING
- city : STRING
- province : STRING
- country : STRING
- postal_code : STRING
- phone_number : STRING
- phone_type : STRING (Mobile Phone / Home Phone)
- division : STRING (Back-End, Front-End, Full-Stack)

- Delete an employee data : <http://127.0.0.1:3000/api/deleteAnEmployee>

The screenshot shows a REST client interface with a tab for the endpoint `POST deleteAnEmployee`. The request is configured as follows:

- Method:** POST
- URL:** `http://127.0.0.1:3000/api/deleteAnEmployee`
- Body Type:** form-data
- Body Parameters:**

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id	5	
Key	Value	Description

This API is used to delete employee data

Body params :

- id : INTEGER, Primarykey