The Task: CRUD Application - Laravel and React

1. Complete application with Rest API alongside CRUD (add, update, delete, view) functionalities through React Js and Laravel.

2. Salary is calculated on list display via React Js by hourly wage, fixed each month, full pay or pay per hour.

3. Each employee is only in one department through one-to-one relationship in database.

4. The data type in the feed can be verified and appropriate error is also displayed if a field is not valid through network errors' access via Axios request.

Technologies

1. ***The data has been stored on the server using cache memory in Laravel in order to maximize the speed of fetching data from the database and the speed after implementing cache memory has been validated through Postman.***

2. Factories in Laravel are created in order to insert data for testing purposes. You can use tinker console to generate dummy data.

3. Events and Listeners are created on the models to despatch event while data is created.

4. Migrations are created for database scripts.

5. Rest API is created to perform CRUD operation for the application through JSON bridging Laravel and React Js.

6. There is still a room left for middlewares for authentication and seeders for insertion of data that could be employed in the future version of the app.

Application Setting: Laravel and React Js

There are two folders with the name of client and server, client contains React Js while the server contains Laravel. In order to set up the application, please follow below steps:

1. First go to the folder of server and in the .env file update your MYSQL database configuration.

2. Go to the server folder through terminal and run below command for migrations

    php artisan migrate

    this will create three tables on database.

3. After running the migration command, execute the below SQL script:

    departments table

    INSERT INTO `departments` (`id`, `name`, `created_at`, `updated_at`) VALUES

    (1, 'Psychology', NULL, NULL),

    (2, 'Mathematics', NULL, NULL),

    (3, 'Geology', NULL, NULL),
    (4, 'Pharmacy', NULL, NULL);
    And

salary_types table

INSERT INTO `salary_types` (`id`, `type`, `created_at`, `updated_at`) VALUES
(1, 'Hourly', NULL, NULL),
(2, 'Bonuses', NULL, NULL),
(3, 'Commission', NULL, NULL);

in order to insert the required values to the table **departments and salary_types.**

4. For adding dummy data into the **employee_details** table you have to use tinker: php artisan tinker and you will see **Psy shell** then run the below command:

factory(\App\EmployeeDetails::class,100)->create()

5. Run command : **php artisan config:cache**

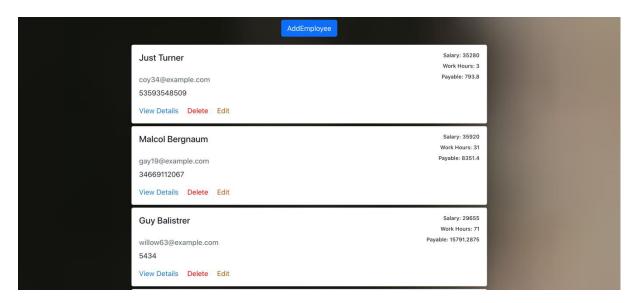6. On server folder run below command:

php artisan serve

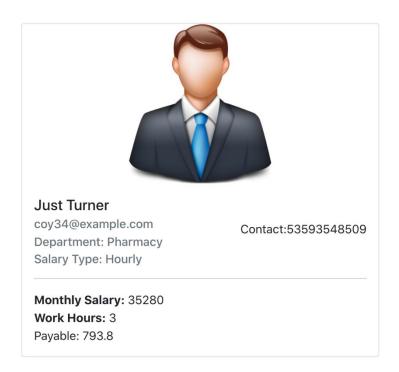7. Go to the client folder and run the below command :

npm start

the above command will run the client side and will give you access to the application services.

Working images of application:

List of employes:

View Details:

Just Turner

coy34@example.com
Department: Pharmacy
Salary Type: Hourly

Contact:53593548509

Monthly Salary: 35280
Work Hours: 3
Payable: 793.8

Add employee:

Edit Employee:

Edit Guy Balistrer   ✕

First Name
Guy

Last Name
Balistrer

Email Address
willow63@example.com

Phone
5434

Work Hours
71

Hourly   ⌄

Salary
29655

Psychology   ⌄

Save Changes