**Flycsv App**

**An application for reading and saving CSV file.**

GitHub Link: <https://github.com/tony831993/flycsv>

1. **The application has two major parts**
2. **Backend:** “flycsvBackend” this is backend for the flycsv application, developed in Symfony 5.8.

Backend has two main entities:

a. **User:** It contains the of users in the application.

b. **Employees:** It contains the list of employees added by a user, by uploading the csv and saving the data parsed from csv.

Backend has two controllers:

a. **UserController:** It have REST endpoints exposed for login, creating, updating and deleting a user.

* /api/login - [POST] - login with username and password
* /api/user - [POST] - Create new user
* /api/user/{id} - [PUT, POST] - Update existing user
* /api/user/{id} - [DELETE] - Delete / Remove a user

b. **EmployeeController:** It have REST endpoints exposed for saving and getting the data for employees added by the user

* /api/employee/savedata - [POST] - save data for employees
* /api/employee/getdata - [POST] - get employee data for a user

1. **Frontend:** “fly-csv-app“ this is frontend for the flycsv application, developed in Angular 16.0

There is single module app module (root module) in the application.

a. **Login Component:** Used by user to login in the application (username: admin, password: admin)

b. **Home Component:** \* After successful login the user will land on the home component. Where user can upload a csv file (Sample file: ./employees.csv) \* Once the user selects the csv file. The data will be rendered on the same page where user can sort / filter the data by columns Employee ID, First Name, Last Name & Salary. \* User can view the records and can do two actions Save and Reset. Save - Will save the records to employee data for the logged in user. Reset - Will reset the uploaded csv and its data.

c. **Records:** This components is used to list the employee data saved by a user. User can sort / filter the data by columns Employee ID, First Name, Last Name & Salary.

Application contains 3 routes:

* /login
* /home
* /records

**Directives:** Sortable header for sorting by columns of the table.

**Route Guards:** Auth Guard to check for logged in user and allow /home and /records route. (Unauthenticated user will be redirected to /login)

**Pipes:** Created custom pipe for filtering table of employee based on columns.

**\*\* MySQL Database:** DB Name - flycsv. DB have two tables "user" and "employee".

The .sql file for them are provided at root path

* ./flycsv\_employee.sql
* ./flycsv\_user.sql

1. **Run application via docker**

1. Navigate to root in the github repository. Repo link (https://github.com/tony831993/flycsv)

2. Clone the repository.

3. Repository has a docker-compose file

- https://github.com/tony831993/flycsv/blob/main/docker-compose.yml

4. Run `docker-compose up -d` this will generate the necessary images.

5. Application will be accessible on the below link

- http://localhost:4200/home

- Login Credentials -

username: admin

password: admin

6. After login :

1. User will redirect to Home screen.

2. Use sample csv to upload (https://github.com/tony831993/flycsv/blob/main/employees.csv).

3. On upload csv file user can see the csv data on same screen, with two action buttons `Save` & `Reset`.

4. Save - user can save the current data to backend database for the logged in user.

5. Reset - user can reset the csv data loaded.

7. View Records - is to show saved records for the user.

1. **Development server**

Database: MySQL run/import in your local mysql database the .sql files mentioned(“./flycsv\_employee.sql” and “./flycsv\_user.sql”).   
(This will create the necessary table, with **username: admin password: admin**)

Backend Application: Run in `http://localhost:8000/`

- Navigate to ./flycsvBackend folder

- Run `symfony server:start` for running symfony application.

- Runs in `http://localhost:8000/`

Frontend Application:

- Navigate to ./flycsv-app folder

- Run `ng serve` for a dev server.

- Navigate to `http://localhost:4200/`.

- Login the application using credential **username: admin password: admin**

1. **Assests**

Root:

\* employees.csv (Application is capable of reading with following structure of application)

\* ./flycsv\_employee.sql

\* ./flycsv\_user.sql

# **Screenshots for Reference**

**Login Screen**

# **A screenshot of a computer Description automatically generated**

# 

**Home Screen**

# 

**Records Screen**

# 