CLINIC MANAGEMENT SYSTEM

DEPLOYMENT

The website is deployed on AWS S3 Bucket and Server is hosting at Ubuntu 18.04

Instance: 165.22.249.19

For backend, of this application use:

- Tomcat 9 at port 8081 : http://165.22.249.19:8081
- PostgresSQL

For frontend, of this application use:

- ReactJS
- Webpack
- Amazon S3 Bucket Static Website Hosting: sadi-app

URL of this application:

http://sadi-app.s3-website.us-east-2.amazonaws.com/

Login with username**: doctor**

password: **1**

RUN AT LOCALHOST

To run the application at localhost, exact the file.

At Backend folder, to start:

- import this folder using IntelliJ as Maven project
- Go to Config, AppConfig.java to change database name, user and password

- Change “update" to “create”, run at terminal “mvn jetty:run” (use to create table in database since “update” can not do it)
- Then change back to “update” and run again with “mvn jetty:run”
- Start import value form resources (csv files at **backend/src/main/resources**) to the database table

Notes that the order of column in the table need to match with the column of the csv file and then open the query tools (pgAdmin) to query the content in the “insert.txt” file

Table name	Resources
drug	druglist.csv
medicalservice	services.csv
disease	icdenglish_new.csv
Query tools	insert.txt

At Frontend folder, run: npm install && npm run dev

Then open the localhost:5000 at web browser and login with username: **doctor** ,and password: **1**

TESTING BACKEND

To test the backend of application, using Postman and gain the access token before making request.

The backend API at local is : localhost:8080

If test it on Ubuntu server database, please replace http://localhost:8080 to **http://165.22.249.19:8081/sadi-1.0-SNAPSHOT**

1. Access Token

To get access token, using basic HTTP Authorization

Username**:** client-secret

Password**:** 1

So basic Authorization base64 after applied btoa() will be: Y2xpZW50LWlkOnNlY3JldA==

The request to get access token is:

Method - Get

Header:

Authorization: Y2xpZW50LWlkOnNlY3JldA==

Params:

grant_type: password

username:doctor

password: 1

URL: http://165.22.249.19:8081/sadi-1.0-SNAPSHOT/oauth/token?grant_type=password&username=doctor&password=1

The access token now is saved in local storage as access_token

2. Visit

a. Get Visits

To fetch the list of visit, the request is:

Method - Get

Params:

access_token: ${access_token}

URL: http://165.22.249.19:8081/sadi-1.0-SNAPSHOT/visits?access_token=${access_token}

To get a specific visit:

Method - Get

Params:

access_token: ${access_token}

URL: http://165.22.249.19:8081/sadi-1.0-SNAPSHOT/visits/${id}?access_token=${access_token}

b. Add a new visit

To add new visit:

Method - POST

Headers: Accept: application/json

Content-type: application/json

Params: access_token: ${access_token}

Body: raw - application/json

{ "patient": { "id": 11, "name": "6546465", "dateOfBirth": "2000-01-01", "gender": "MALE", "address": "42342343" }, "date": "2019-11-11", "time": "11:11:11"}

URL: http://165.22.249.19:8081/sadi-1.0-SNAPSHOT/visits/?access_token=${access_token}

Notes: The date, time and checkout will be auto correct so you can add new with only define patient

c. Update a visit

To update a specific visit:

Method - PUT

Headers: Accept: application/json

Content-type: application/json

Params: access_token: ${access_token}

Body: raw - application/json

{ "id":"3" "patient": { "id": 3, "name": "6546465", "dateOfBirth": "2000-01-01", "gender": "MALE", "address": "42342343" }, "date": "2019-11-11", "time": "11:11:11", "checkedOut": false }

URL : http://165.22.249.19:8081/sadi-1.0-SNAPSHOT/visits/?access_token=${access_token}

Notes: Need to define exsiting visit ID to update it

2. Patient

a. Get Patients

To fetch the list of patients, the request is:

Method - Get

Params: access_token: ${access_token}

API: http://165.22.249.19:8081/sadi-1.0-SNAPSHOT/patients?access_token=${access_token}

To get a specific patient:

Method - Get

Params: access_token: ${access_token}

API: http://165.22.249.19:8081/sadi-1.0-SNAPSHOT/patients/${id}?access_token=${access_token}

b. Add a new patient

To add new patient:

Method - POST

Headers: Accept: application/json

Content-type: application/json

Params: access_token: ${access_token}

Body:raw - application/json

{ "name": "Suzy Jasque", "dateOfBirth": "2001-11-11", "gender": "FEMALE", "address": "21/43 Pham Van Dong District 6" }

API: http://165.22.249.19:8081/sadi-1.0-SNAPSHOT/patients/?access_token=${access_token}

c. Update a patient

To update a patient:

Method - PUT

Headers: Accept: application/json

Content-type: application/json

Params: access_token: ${access_token}

Body: raw - application/json

{ "id": 8, "name": "Suzy Jasque", "dateOfBirth": "1990-12-21", "gender": "FEMALE", "address": "21/43 Pham Van Dong District 6" }

API: http://165.22.249.19:8081/sadi-1.0-SNAPSHOT/patients/?access_token=${access_token}

Notes: Need to define exsiting patient ID to update it

d. Delete a patient

To delete a patient:

Method - DELETE

Params: access_token: ${access_token}

API: http://165.22.249.19:8081/sadi-1.0-SNAPSHOT/patients/${id}?access_token=${access_token}

Notes: If patient is already exist in some visit, this patient CANNOT be deleted.

3. Drug

a. Get Drugs

To fetch the list of drugs, the request is:

Method - Get

Params: access_token: ${access_token}

API: http://165.22.249.19:8081/sadi-1.0-SNAPSHOT/drugs?access_token=${access_token}

To get a drug:

Method - Get

Params: access_token: ${access_token}

API: http://165.22.249.19:8081/sadi-1.0-SNAPSHOT/drug/${id}?access_token=${access_token}

b. Add a new Drug

To add new drug:

Method - POST

Headers: Accept: application/json

Content-type: application/json

Params: access_token: ${access_token}

Body:raw - application/json

{ "name": “Panacetamol 500mg"}

API: http://165.22.249.19:8081/sadi-1.0-SNAPSHOT/drug/add/?access_token=${access_token}

c. Update a drug

To update a drug:

Method - PUT

Headers: Accept: application/json

Content-type: application/json

Params: access_token: ${access_token}

Body: raw - application/json

{ "id": 8, "name": "Panacetamol 250mg"}

API: http://165.22.249.19:8081/sadi-1.0-SNAPSHOT/drug/update/?access_token=${access_token}

Notes: Need to define exsiting drug ID to update it

d. Delete a patient

To delete a specific patient:

Method - DELETE

Params: access_token: ${access_token}

API: http://165.22.249.19:8081/sadi-1.0-SNAPSHOT/drug/${id}?access_token=${access_token}

Notes: If drug is already exist in some prescription, this patient CANNOT be deleted.

For others backend API, please look at fetching function in jsx files at frontend folder to understand the structure of request and get the API to test on postman.