==========
Major Project
==========

Client : Rhode Island (State Govt)

Project Name : IES  - Integrated Eligibility System

It is used to provide fully integrated eligibility System for RI state citizens.

By using IES application, RI is providing below health and insurance plans for RI state citizens.

1) SNAP (Supplemental Nutrition Assistance Program)
2) CCAP (Child Care Assistance Program)
3) Medicaid
4) Medicare (Senior Citizens, above 60 years )
5) QHP (Qualified Health Plan)
6) RIW (Rhode Island Works)

==========

=> DHS : Department of Health Services

=> DHS Employees : Case Workers

=> IES application will be accessible only in DHS offices

=> Caseworker will use IES application to provide benefits for Citizens.

=> Citizens should visit nearest DHS office to apply for plans. Citizen should talk to case worker to apply for Plan.

==========

In IES project we have below modules

1) Admin

2) AR (Application Registration)

3) DC (Data Collection)

4) ED (Eligibility Determination)

5) CO (Correspondence Module)

6) BI (Benefit Issuance)

7) Reports

==========
Admin Module

==========

-> Admin is responsible to create accounts for case workers to access IES system.

-> Admin is responsible to create Plans

1) Case Worker Accounts Management (Create/Update/Delete/Retrieve)
2) Plans management (Create/Update/Delete/Retrieve)

3) Login
4) Dashboard
5) Edit Profile
6) Forgot Pwd

==========
AR Module
==========

=> AR stands for Application Registration

=> AR api will collect citizen basic information and it will verify Citizen belongs to RI state or not based on SSN (It will communicate with SSA-WEB API, which is federal gov project for verification).

=> If citizen belongs to RI state then application will be created for citizen to apply for the plan.

=> If citizen not belongs to RI state then application will not be created hence Citizen can't apply for the plan.

=========
DC Module
==========

=> DC stands for Data Collection

=> DC api is responsible to collect citizen data which is required to apply for the plan

1) Citizen details
2) Family details
3) Education Details
4) Employment Details
5) Income details
6) Kids Details etc....

==========
ED Module
==========

=> ED stands for Eligiblity Determination

=> ED is responsible to verify Citizen eligibility for the plan.

=> ED will execute Plan conditions with citizen data.

=> If citizen data matches with Plan conditions then citizen will be approved for the plan otherwise citizen will be denied for the plan.

Note: Approved citizens will get benefit amount from Govt.

==========
CO Module
==========

=> CO stands for Correspondence

=> This is responsible to send Notices to citizens regarding their plan eligibility

1) Approved Notice
2) Denied Notice
3) Termination Notice
4) Plan renewal Reminder Notices
5) Missing Documents Notice

==========
BI Module
==========

=> BI stands for Benefit Issuance

=> It is responsible to send benefit amount to approved citizens.

================
Reports module
================

-> It is used to generate reports in excel/pdf format with application data

1) Report Based on Plan Name

2) Report Based on Plan Status

3) Report Based on Duration (start date & end date)

4) Report based on Citizen Gender

1) What is IES
2) Why IES required
3) IES Client
4) How to access IES
5) IES Plans & Conditions
6) IES Modules
7) Purpose of Each Module in IES App

Mock Screens : http://iesapp.s3-website.ap-south-1.amazonaws.com/login.html

SSA Web API URL : http://65.2.166.5:8080/swagger-ui.html

==========
DB Design
==========

Table Name : IES_USERS (It is used to store IES app user account details)

USER_ID pk
FULL_NAME
USER_EMAIL
USER_PWD
USER_PHNO
USER_GENDER
USER_DOB
USER_SSN
ACTIVE_SW (Default : Y)
ACC_STATUS (Default : LOCKED)
ROLE_ID
CREATE_DATE
UPDATE_DATE
CREATED_BY (Foregin Key : user_id from IES_USER table)
UPDATED_BY (Foregin Key : user_id from IES_USER table)
ROLE_ID (1=ADMIN, 2=CW)

Table Name : IES_PLANS (It is used to store Plans of IES app)

PLAN_ID PK
PLAN_NAME
PLAN_START_DATE
PLAN_END_DATE
PLAN_CATEGORY
ACTIVE_SW (Defult : Y)
CREATED_BY (Foregin Key : user_id from IES_USER table)
UPDATED_BY (Foregin Key : user_id from IES_USER table)
CREATED_DATE
UPDATED_DATE

Table name : IES_APPS (It is used to store citizen application details)

CASE_NUM
NAME
EMAIL
PHNO
DOB
SSN
STATE_NAME
CITY_NAME
HOUSE_NUM
CREATED_BY (Foregin Key : user_id from IES_USER table)
CREATED_DATE

Table : DC_PLAN_SELECTION (it is used to store citizen selected plan info)

plan_selection_id PK
case_num FK
plan_id FK

Table : DC_INCOME (It is used to store citizen income info)

income_id PK
salary_income
rent_income
property_income
case_num FK

Table : DC_EDUCATION (it is used to store citizen education info)

education_id PK
highest_degree
graduation_year
university
case_num FK

Table : DC_CHILDREN (It is used to store citizen childrens info)

child_id PK
child_name
child_dob
child_ssn
case_num FK

Table : ED_ELIG_DTLS (It is used to store citizen eligibility details)

ed_trace_id PK
case_num FK
plan_name
plan_status
elig_start_date
elig_end_date
benefit_amt
denial_reason
created_date

Table : CO_NOTICES (This is used to store citizen notices)

notice_id PK
case_num FK
ed_trace_id FK
co_pdf_url
print_date
notice_status ( P or H ) (default : P)
created_date

Table : BI_INFO (It will maintain citizen benefits status)

benefit_id PK
case_num FK
benefit_month_year
benefit_amt
transaction_date
transaction_status

==========
Development Procedure
==========

1) DB Design (Schema Creation)

2) Create Entity classes & Repository Interfaces

3) Create Binding classes (request & response)

4) Create Constants and Properties classes

5) Create Service interface & impl classes

6) Create REST Controller to handle request & response

7) Implement Exception Handling

8) Implement Logging

9) Test api functionality using Postman / swagger

==========
Today's Task
==========

=> Create one Spring Boot application with Data JPA

=> Configure Data Source Properties in yml file

=> Create Entities & Repositories

=> Run the application and check tables creation in database

==========
Admin API Entity classes
==========

1) IES_USERS => UserEntity.java & UserRepo.java

2) IES_PLANS => PlanEntity.java & PlanRepo.java

==========
Admin API Binding Classes
==========

1) UserAccountForm.java
2) PlanForm.java
3) LoginForm.java
4) UnlockAccForm.java
5) DasboardCards.java

==========
Admin API Service Components
==========

1) AccountService.java ( User acc related CRUD Ops )

2) PlanService.java ( plan related CRUD ops )

3) UserService.java (login/forgot-pwd/profile/dashboard)

==========
Admin API Rest Controller classes
==========

1) AccountRestController.java

2) PlanRestController.java

3) UserRestController.java

==========
Admin API Utility Components
==========

1) AppConstants.java

2) AppProperties.java

3) AppExceptionHandler.java

4) EmailUtils.java

5) SwaggerConfig.java

==========
AR API Development
==========

==========
DC API Development
==========

Step-1 : Setup Entities & Repositories

1) IES_USERS

2) IES_PLANS

3) IES_APPS

4) DC_PLAN_SELECTION

5) DC_INCOME

6) DC_EDUCATION

7) DC_KIDS

Step-2 : Create Binding classes

Step-3 : Service Interface & Implementations

Step-4 : Create Rest Controllers

==========
Eligibility Determination
==========

-> After Data Collection completed, then Eligibility Determination module execution will start

-> ED module is responsible to determine citizen is eligble for the applied plan or not based on plan conditions and citizen data

-> If citizen data matching with plan rules then citizen will be approved for the plan

-> If citizen data not-matching with plan rules then citizen will be denied for the plan

==========
Plan wise Conditions
==========

SNAP Condition : If employment_income <=300$ then citizen is eligible for SNAP

CCAP Condition : If employment_income <=300$ and kids count > 0 and each kid age <=16 then eligible for CCAP

Medicaid : If employment_income <=300$ and Property Income + Rent Income is 0 then eligible for Medicaid

Medicare : If citizen age is >=65 then eligible for Medicare

RIW : If citizen is un-employed and graduated then eligible for RIW

==========
CO Module
==========

=> CO is responsible to generate and send Notices to citizens

1) Eligibility Approved Notice

2) Eligibility Denied Notice

3) Eligibility Renewal Reminder Notice

4) Eligiblity Termination Notice

5) Additional Docs Required Notice

6) Missing Docs Notice

Note: In Real IES application we have 185 Notice types are available

=> Notices generation will happen through nightly batch jobs

=> Batch Jobs will be scheduled for execution

1 sec => 1 notice

1 min => 60 notices

1 hour => 3600 notices

Note: Daily 1 hour slot given for CO batch execution in prodution.

1st Day : 2000 applications ==> 2000 notices sent (No Problem)

2nd Day : 3000 applications ==> 3000 notices sent (No Problem)

3rd Day : 5000 applications ==> 3600 notices sent (1400 pending)

4th Day : 10k applications ==> 3600 notices sent (6400 + 1400 = 7800 pending)

5th Day : 20k applications ==> 3600 notices sent (16400 + 7800 = 24k pending)

########## Create 10 threads in batch program to process 10 triggers at a time ###########

1 Thread ==> 1 trigger => 1 sec

10 Threads => 10 triggers => 1 sec

1 sec => 10 triggers

1 min ==> 600 triggers

1 hour ===> 36000 triggers

=> To implement multi-Threading we have used Executors framework available in Java.

```
ExecutorsService exService = Executors.newFixedThreadPool(10);

List<CoTriggers> cotriggers = service.getPendingTriggers();

for(CoTrigger trigger : cotriggers){

exService.submit(new Runnable(){
public void run ( ) {
processTrigger(trigger);
};
});

}
```

==========
Benefit Issuance Module
==========

=> It is used to send benefit amount for citizens who got approved for the plan eligibility

=> On Monthly basis benefits should be given for approved citizens.

=> BI Module will take approved citizens eligibility and it will prepare a file like below. In file every row represents one citizen benfits information.

```
34567,John,796968,3737497349,756.00 $
34568,Peter,76968,345345435435,356.00 $
34567,John,796968,3737497349,756.00 $
34568,Peter,76968,345345435435,356.00 $
34567,John,796968,3737497349,756.00 $
34568,Peter,76968,345345435435,356.00 $
34567,John,796968,3737497349,756.00 $
34568,Peter,76968,345345435435,356.00 $
```

=> Bi module will upload this file into FTP location then Bank App will load that file and will process payments.

==========
Reports Module
==========

=> It is used to generate reports in our application in excel and pdf format

1) Plan Name wise Report

2) Plan Status Wise Report

3) Gender wise Report

4) Date Range Report

==========
Angular Integration
==========