

Hi Team,

I have developed the scripts using PyCharm IDE,python,Robotframework and selenium.This is a very simple test suit that I prepared during limited timeframe and please go through this document if you face any difficulty in understanding. :)

Versions:

Python version : Python 3.11.1

Robot version :Robot Framework 6.0.2 (Python 3.11.1 on win32)

pip version: pip 22.3.1

Selenium:4.8.0

Note:I have highlighted my comments in blue color.

bookkeeper.py,clerks.py and gov.py are the python files contains all the functions.

USER STORIES

=====

(1) As the Clerk, I should be able to insert a single record of

working class hero into database via an API

AC1: Single record of a working class hero should consist of Natural Id

(natid), Name, Gender, Birthday, Salary and Tax paid

For this testcase I used robot framework and sent POST request to url:http://localhost:8080/swagger-ui.html#/calculator-controller/insertPersonUsingPOST_1.

Please refer TC01_AC01_insertOneRecord.robot file in the tst suit.

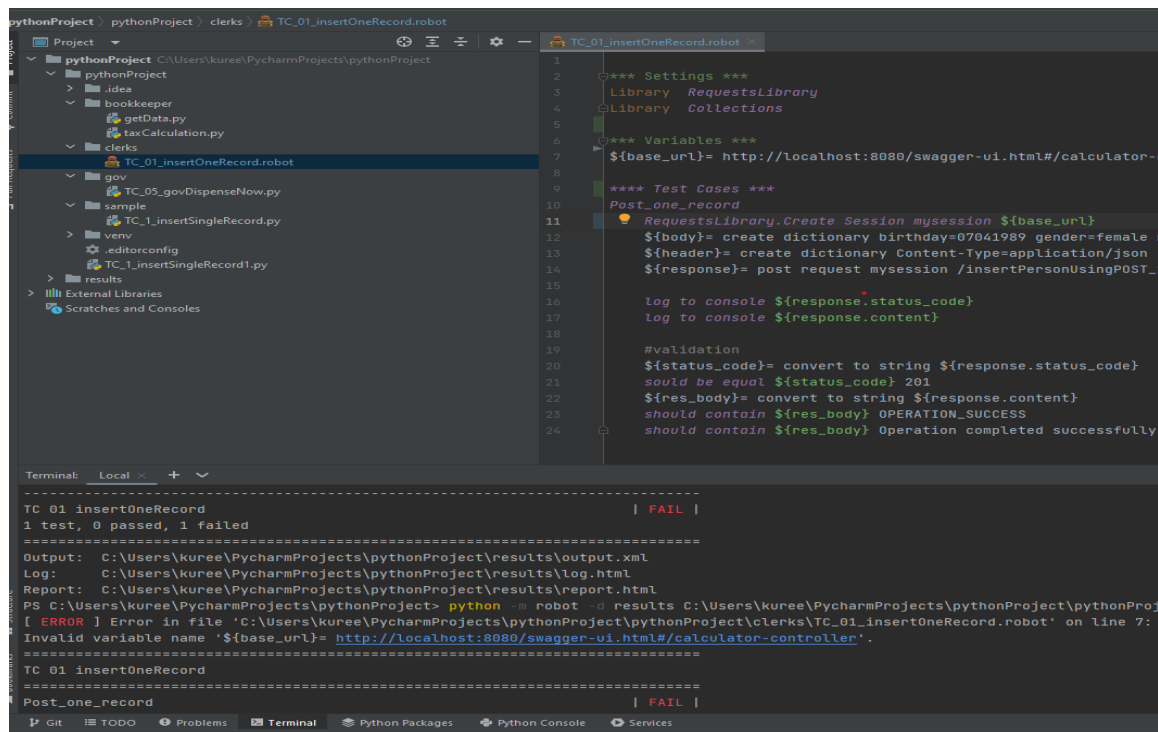
Installed robot class in my system and added pluggins in pyCharm.You can use below cmd to run the test.

sample:

python -m robot -d results

C:\Users\kuree\PycharmProjects\pythonProject\pythonProject\clerks\TC_01_insertOneRecord.robot

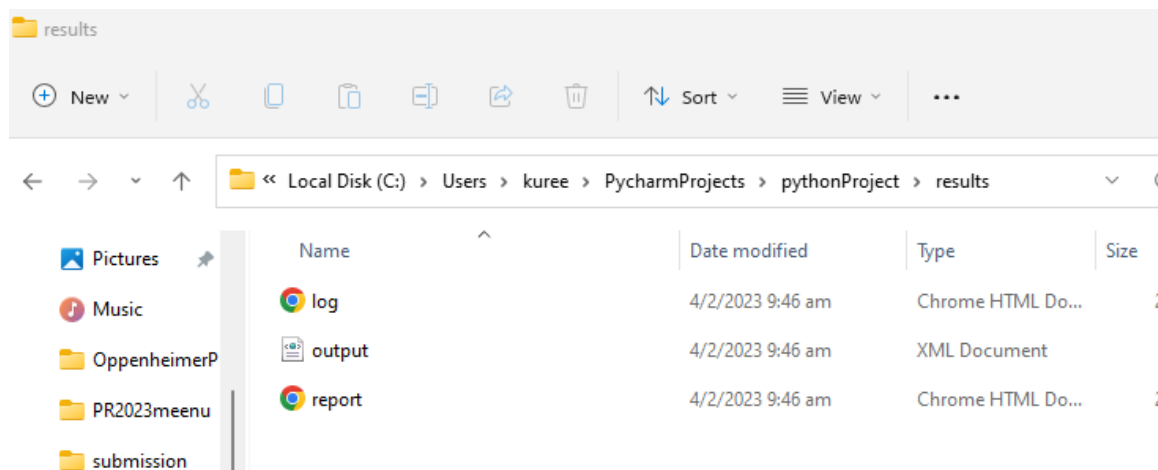
IDE snapshot:



How to see the Report:

Once the test is completed its exeution, go to your project folder and open 'results' folder.

Click on 'report.html' page and see the report.



HTML Report as shown below with details.

Below is the sample report.

Name	Documentation	Tags	Status	Message	Elapsed	Start / End
TC01insertOneRecord Post_one_record			FAIL	AssertionError: Response status is not 200 OK. Expected status: 200 OK. Actual status: 400 Bad Request. Request body: {"name": "John", "gender": "Male", "salary": 10000, "birthday": "1990-01-01", "tax": 1000}. Response body: {"error": "Invalid input"}	00:00:00.031	2023/02/04 09:46:14.208 2023/02/04 09:46:14.239

(2) As the Clerk, I should be able to insert more than one working class hero into database via an API

AC1: Enhancement of (1), with the ability to insert a list

For this testcase I used json file and python to send POST request to API.

Please see the testcase TC02_AC01_insertListOfRecords.robot.

(3) As the Clerk, I should be able to upload a csv file to a portal so that I can populate the database from a UI

Understanding that this is to upload a csv using a button in the UI.

AC1: First row of the csv file must be natid, name, gender, salary, birthday, tax

AC1 ,AC02 and AC3 clubbed together.

please refer TC02_AC03_uploadCSV.robot

AC2: Subsequent rows of csv are the relevant details of each working class hero

For Ac1 and AC2,I created a csv file as given above and saved in local folder.

please refer TC03_AC03_uploadCSV.robot

AC3: A simple button that allows me to upload a file on my pc to the portal

Functionality is not working in the application.So I completed the script in python for uploading csv file.

For this testcase I used robotframework+selenium+python to upload csv file to the application.

please refer TC03_AC03_uploadCSV.robot, This method is for uploading csv from UI.

Method 2 is for uploading using API. This also included in the same testcase.

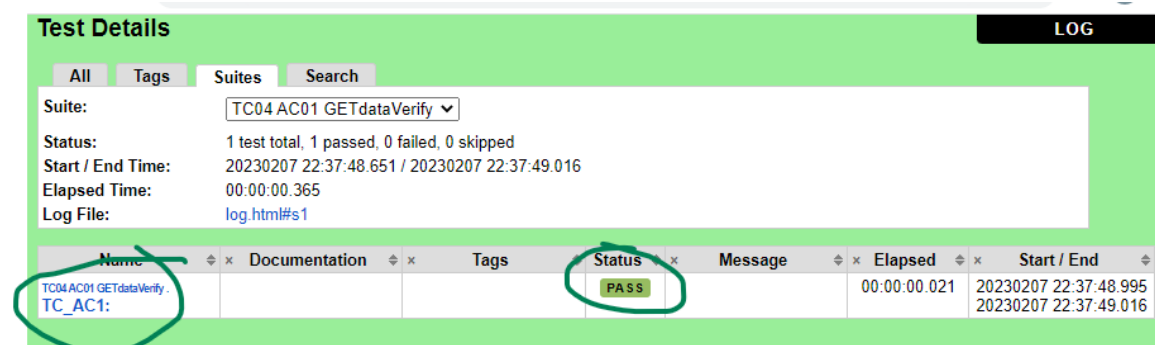
(4) As the Bookkeeper, I should be able to query the amount of tax relief for each person in the database so that I can report the figures to my Bookkeeping Manager

AC1: a GET endpoint which returns a list consist of natid, tax relief amount and name

To get these details I used robot framework+python to send GET request.

Used robot class and send GET request to "http://localhost:8080/calculator/taxRelief" saved the json file and verified the response data.

Please find the testcase TC04_AC01_GETdata&verify.robot file in test suit.



The screenshot shows a 'Test Details' window with a green header and a black 'LOG' button. Below the header, there are tabs for 'All', 'Tags', 'Suites', and 'Search'. The 'Suites' tab is selected, showing a dropdown menu with 'TC04 AC01 GETdataVerify'. Below this, the test status is '1 test total, 1 passed, 0 failed, 0 skipped'. The 'Start / End Time' is '20230207 22:37:48.651 / 20230207 22:37:49.016'. The 'Elapsed Time' is '00:00:00.365'. The 'Log File' is 'log.html#s1'. Below this, there is a table with columns: Name, Documentation, Tags, Status, Message, Elapsed, and Start / End. The table has one row with the following data: Name: 'TC04 AC01 GETdataVerify. TC_AC1:', Documentation: (empty), Tags: (empty), Status: 'PASS', Message: (empty), Elapsed: '00:00:00.021', and Start / End: '20230207 22:37:48.995 / 20230207 22:37:49.016'. The 'Name' and 'Status' columns are circled in green.

Name	Documentation	Tags	Status	Message	Elapsed	Start / End
TC04 AC01 GETdataVerify. TC_AC1:			PASS		00:00:00.021	20230207 22:37:48.995 / 20230207 22:37:49.016

AC2: natid field must be masked from the 5th character onwards with dollar sign '\$'

Please find the testcase TC04_AC02_verifyNatidMaskedwith\$.robot file in test suit.

Here I used selenium+python to verify if the column of 'natid' in the table contains '\$' symbol from 5th character.

File | C:/Users/kuree/PycharmProjects/pythonProject/results/report.html#suites?s1

Test Details

LOG

All Tags Suites Search

Suite: TC04 AC02 verifyNatidMaskedwith\$

Status: 1 test total, 1 passed, 0 failed, 0 skipped

Start / End Time: 20230207 22:44:17.193 / 20230207 22:44:17.504

Elapsed Time: 00:00:00.311

Log File: log.html#s1

Name	Documentation	Tags	Status	Message	Elapsed	Start / End
TC04 AC02 verifyNatidMaskedwith\$. Test			PASS		00:00:00.000	20230207 22:44:17.504 20230207 22:44:17.504

AC3: computation of the tax relief is using the formula as described:

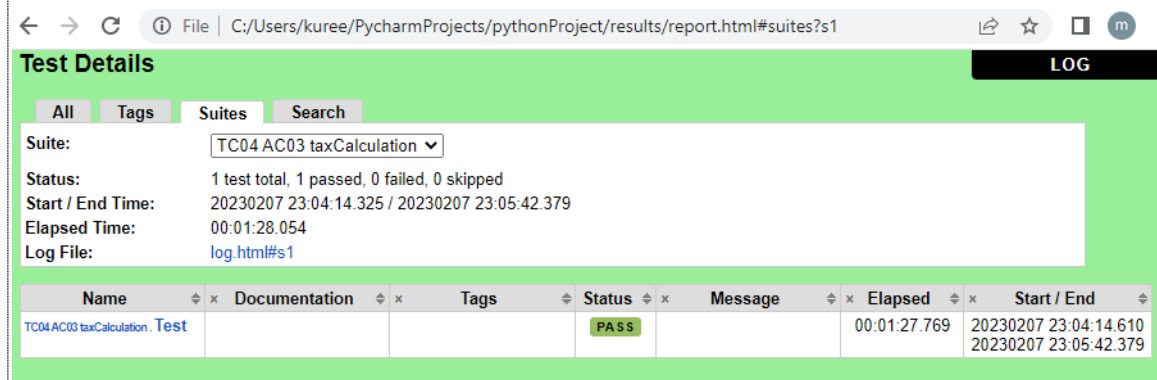
For this testcase I used python code to do the calculation and get data from csv.

Library used is panda.

Please refer testcase TC04_AC03_calculateTax.py

```
TC05_AC02_buttonTextDispenseTax.robot x TC04_AC05_taxCalculation
if gender == "M" and age <= 18:
    return calculate(1, 0, salary, taxPaid)
elif gender == "M" and age <= 35:
    return calculate(0.8, 0, salary, taxPaid)
elif gender == "M" and age <= 50:
    return calculate(0.5, 0, salary, taxPaid)
elif gender == "M" and age <= 75:
    return calculate(0.367, 0, salary, taxPaid)
elif gender == "M" and age <= 76:
    return calculate(0.05, 0, salary, taxPaid)
elif gender == "F" and age <= 18:
    return calculate(1, 500, salary, taxPaid)
elif gender == "F" and age <= 35:
    return calculate(0.8, 500, salary, taxPaid)
elif gender == "F" and age <= 50:
    return calculate(0.5, 500, salary, taxPaid)
elif gender == "F" and age <= 75:
    return calculate(0.367, 500, salary, taxPaid)
elif gender == "F" and age <= 76:
    return calculate(0.05, 500, salary, taxPaid)
```

we can use panda to get data from csv.here I used direct entry details from the user since upload feature was not working.



The screenshot shows a web browser window with the URL `C:/Users/kuree/PycharmProjects/pythonProject/results/report.html#suites?s1`. The page title is "Test Details" and there is a "LOG" button in the top right. Below the title, there are tabs for "All", "Tags", "Suites", and "Search". The "Suites" tab is selected, and a dropdown menu shows "TC04 AC03 taxCalculation". The status is "1 test total, 1 passed, 0 failed, 0 skipped". The start/end time is "20230207 23:04:14.325 / 20230207 23:05:42.379". The elapsed time is "00:01:28.054". The log file is "log.html#s1". Below this, there is a table with columns: Name, Documentation, Tags, Status, Message, Elapsed, and Start / End. The table contains one row for "TC04 AC03 taxCalculation . Test" with a status of "PASS", an elapsed time of "00:01:27.769", and start/end times of "20230207 23:04:14.610" and "20230207 23:05:42.379".

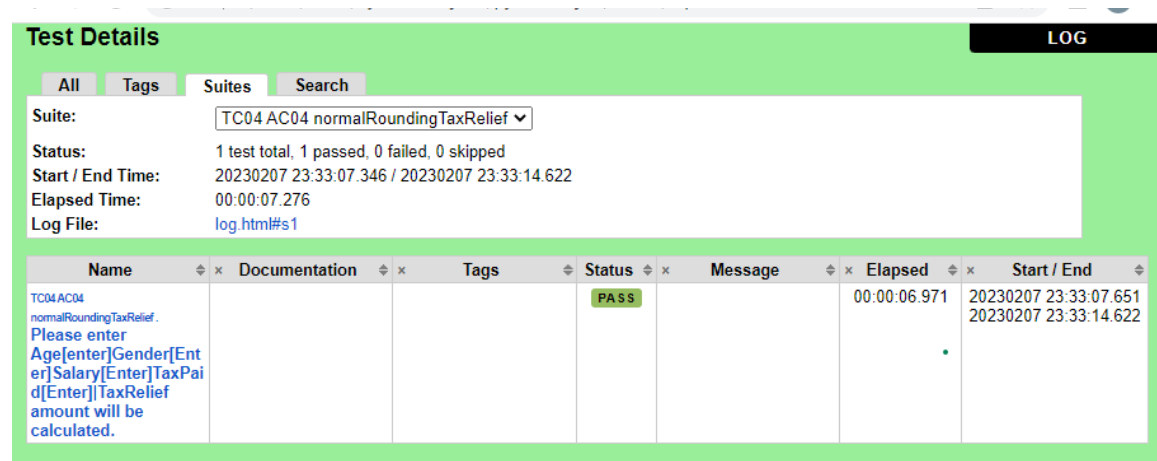
Name	Documentation	Tags	Status	Message	Elapsed	Start / End
TC04 AC03 taxCalculation . Test			PASS		00:01:27.769	20230207 23:04:14.610 20230207 23:05:42.379

*AC - Acceptance Criteria

AC4: After calculating the tax relief amount, it should be subjected to normal rounding rule to remove any decimal places

Please refer testcase `TC04_AC04_normalRoundingTaxRelief.robot`

Applied builtin function `'round(amount)'` in python.



The screenshot shows a web browser window with the URL `C:/Users/kuree/PycharmProjects/pythonProject/results/report.html#suites?s1`. The page title is "Test Details" and there is a "LOG" button in the top right. Below the title, there are tabs for "All", "Tags", "Suites", and "Search". The "Suites" tab is selected, and a dropdown menu shows "TC04 AC04 normalRoundingTaxRelief". The status is "1 test total, 1 passed, 0 failed, 0 skipped". The start/end time is "20230207 23:33:07.346 / 20230207 23:33:14.622". The elapsed time is "00:00:07.276". The log file is "log.html#s1". Below this, there is a table with columns: Name, Documentation, Tags, Status, Message, Elapsed, and Start / End. The table contains one row for "TC04 AC04 normalRoundingTaxRelief . Please enter Age[enter]Gender[Enter]Salary[Enter]TaxPaid[Enter]TaxRelief amount will be calculated." with a status of "PASS", an elapsed time of "00:00:06.971", and start/end times of "20230207 23:33:07.651" and "20230207 23:33:14.622".

Name	Documentation	Tags	Status	Message	Elapsed	Start / End
TC04 AC04 normalRoundingTaxRelief . Please enter Age[enter]Gender[Enter]Salary[Enter]TaxPaid[Enter]TaxRelief amount will be calculated.			PASS		00:00:06.971	20230207 23:33:07.651 20230207 23:33:14.622

AC5: If the calculated tax relief amount after subjecting to normal rounding rule is more than 0.00 but less than 50.00, the final tax relief amount should be 50.00

Please refer testcase `TC04_AC05_taxCalculation0-50.robot`

Applied if else condition.

Test Details

LOG

AllTagsSuitesSearch

Suite:

TC04 AC05 taxCalculation0-50

Status:

1 test total, 1 passed, 0 failed, 0 skipped

Start / End Time:

20230207 23:37:17.345 / 20230207 23:38:16.567

Elapsed Time:

00:00:59.222

Log File:

log.html#s1

Name	Documentation	Tags	Status	Message	Elapsed	Start / End
TC04 AC05 taxCalculation0-50. Test			PASS		00:00:58.917	20230207 23:37:17.650 20230207 23:38:16.567

AC6: If the calculated tax relief amount before applying the normal rounding rule gives a value with more than 2 decimal places, it should be truncated at the second decimal place and then subjected to normal rounding rule

For this testcase I used function 'format(taxRel,.2f)' function in python.

Please see the test case TC04_AC06_taxReliefRoundingafterRemoveDecimal.robot

Test Details

LOG

AllTagsSuitesSearch

Suite:

TC04 AC06 taxReliefRoundingafterRemoveDecimal

Status:

1 test total, 1 passed, 0 failed, 0 skipped

Start / End Time:

20230207 23:41:01.729 / 20230207 23:41:09.708

Elapsed Time:

00:00:07.979

Log File:

log.html#s1

Name	Documentation	Tags	Status	Message	Elapsed	Start / End
TC04 AC06 taxReliefRoundingafterRemoveDeci			PASS		00:00:07.651	20230207 23:41:02.042 20230207 23:41:09.693

(5) As the Governor, I should be able to see a button on the screen so that I can dispense tax relief for my working class heroes

AC1: The button on the screen must be red-colored

Please see the testcase TC05_AC01_buttonColorDispenseTax.robot

← → ↻ File | C:/Users/kuree/PycharmProjects/pythonProject/results/report.html

TC05 AC01 buttonColorDispenseTax Report LOG
Generated 20230207 22:10:07 UTC+08:00
26 seconds ago

Summary Information

Status: All tests passed
Start Time: 20230207 22:10:07.457
End Time: 20230207 22:10:07.674
Elapsed Time: 00:00:00.217
Log File: log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	1	1	0	0	00:00:00	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
No Tags						

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
TC05 AC01 buttonColorDispenseTax	1	1	0	0	00:00:00	

Test Details

All Tags Suites Search

Suite: TC05 AC01 buttonColorDispenseTax

Status: 1 test total, 1 passed, 0 failed, 0 skipped
Start / End Time: 20230207 22:10:07.457 / 20230207 22:10:07.674
Elapsed Time: 00:00:00.217
Log File: log.html#s1

Name	Documentation	Tags	Status	Message	Elapsed	Start / End
TC05 AC01 buttonColorDispenseTax Test			PASS		00:00:00.000	20230207 22:10:07.674 20230207 22:10:07.674

AC2: The text on the button must be exactly "Dispense Now"

Please see the testcase [TC05_AC02_buttonTextDispenseTax.robot](#)

Test Details

All Tags Suites Search

Status: 1 test total, 1 passed, 0 failed, 0 skipped
Total Time: 00:00:00.000

Name	Documentation	Tags	Status	Message	Elapsed	Start / End
TC05 AC02 buttonTextDispenseTax Test			PASS		00:00:00.000	20230207 22:16:33.558 20230207 22:16:33.558

AC3: After clicking on the button, it should direct me to a page with a

text that says "Cash dispensed"

For this testcase I used selenium webdriver with python to verify UI elements and its attributes.

Please see the testcase [TC05_AC03_clickButtonDispenseTax.robot](#)

Test Details							LOG
All	Tags	Suites	Search				
Status:	1 test total, 1 passed, 0 failed, 0 skipped						
Total Time:	00:00:00.000						
Name	Documentation	Tags	Status	Message	Elapsed	Start / End	
TC05 AC03 clickButtonDispenseTax. Test			PASS		00:00:00.000	20230207 22:18:52.402 20230207 22:18:52.402	

TASK

As the Quality Engineer for this project, you are to come up with the testing strategy must be in Python and recent version of Robot Framework. Do setup the necessary framework onto your machine so that you can demo them during the interview.

Yes.done.Thank you.

tegy on how to verify correctness and define quality for this system. Your code Please commit your code to a public repository. Do send us the url once you're done.

I have developed this small test suit in pyCharm and pushed the code to github.The url is as below.

<https://github.com/meenujoseph123/pythonProject>

● You may experience error(s) when testing the application, please note down the error(s) and your approach as QE to resolve the error(s) if any.

1.csv file upload functionality is not working in the application.

We can choose csv file from the explorer window but not selected or displayed on the screen.

2.'Refresh Tax Relief Table' button is not responding.The button is clickable but no functionality is incorporated.

My understanding is there should be a table in UI when clicking on 'Refresh Tax Relief' button, the latest details must be displayed on the table.

3.No ways to identify the login person or his role.

TIPS

● How do you test for functional requirements

First I tested manually and automated after that.

I read all the instructions and tried functionalities one by one by writing testcases manually first.

Verified each and every testable features and developed scripts.

● How do you test for non-functional requirements

Basic UI standards and userfrindliness we test manually.

Performance can test using jmeter,diffrent chrome extensions we can do.

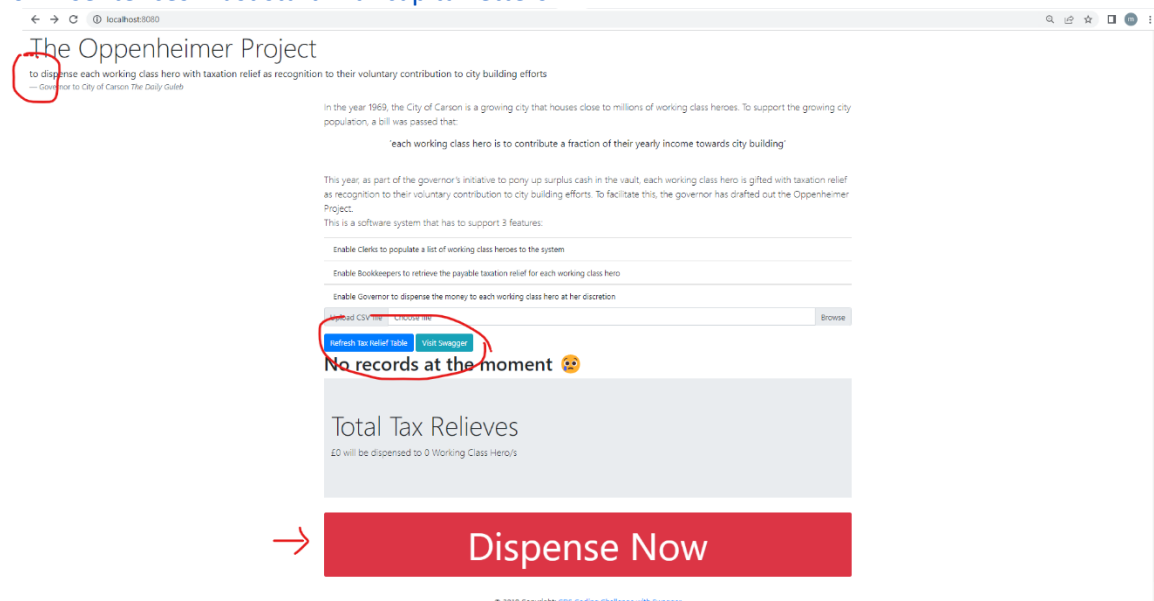
1.'Dispense Button size is large-(negative case)

2.Button alignments would be good if it comed right aligned.(negative case).can be more userfriendly.

3.Can add 'Cancel' buttons for not dispensing cases.

5.Project heading can be center aligned and in bold.

6.All sentences must start with capital letters.



7.It would be better if we can ass a cancel button to remove the selected csv file to remove from selection.

8.Verified messages.But not displayed any error message or success message except "Tax Dispensed".

● What are the test cases you've covered

All the mensioned scenarios has covered.

1.Upload the details in to the system-failed

2. Click on 'Visit Swagger' button and verify if the system navigates to the swagger page-pass
3. Click on 'Refresh Tax Relief Table' and verify the details in table is updated-failed
4. Verify if there is no data entered , application collapse or not-pass
5. Click on 'Dispense' button and verify if the user can move to a new page with 'Tax Dispensed' message-pass

● A short Readme on how to run these tools will definitely put a smile on our face =)

sure. Hope this document helps you.

● Feel free to add any other features which you feel necessary

1. Login feature is missing and can Implement role based access to the functionalities as explained in the scenarios given.
2. UI can be little more attractive and the button size is large and alignments are not as per the standards.
3. Tax relief table that missing can be added with filter functionality and a hyper link to open to the more details page corresponding to each heroes.
4. A tax calculator and a calendar can be add in UI(web services can use) Print functionality.
5. Cancel/Reject/confirm/Ok buttons(and alert popups) can add to make application more userfriendly.
6. Add a button to bring back the user from "Tax Dispersed" page. When click on "Go Back to Home Page" , the user must be able to come back to Oppenheim project home page where "Dispense Now" button available.
7. swagger page is not need to be available to the front end user since it is application programming interface to database.
8. Could have been better if the application display error messages and success messages when requires.
9. For upload functionality, we can add cancel button and display an error message if wrong file is selected.

