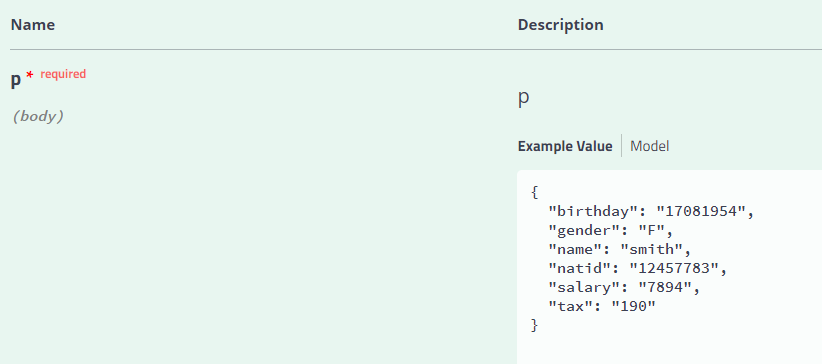
**Testing Documentaion forOppenheimer Project**

* How do you test for functional requirements.
* How do you test for non-functional requirements
* What are the test cases you’ve covered.
* A short Readme on how to run these tools will definitely put a smile on our face =).
* Feel free to add any other features which you feel necessary.

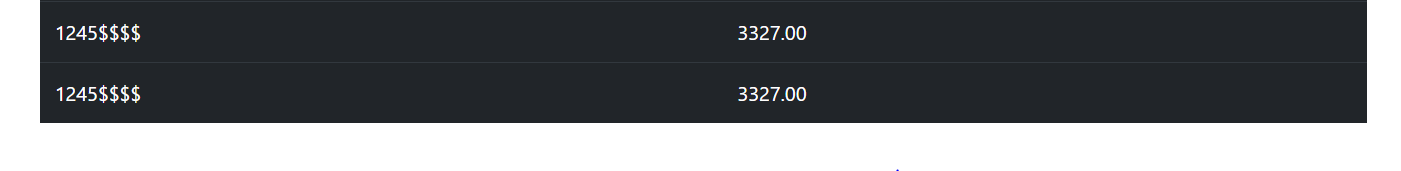
**How do you test for functional requirements?**

For functional requirements I have tested manually for all the acceptance criteria mentioned in the user stories or functional requirement.

1: As the Clerk, I should be able to insert a single record of working class hero into database via an API -- **My observation/test result: All acceptance criteria are not met.** There is bug.



Single record of working class is inserted in to database via an API

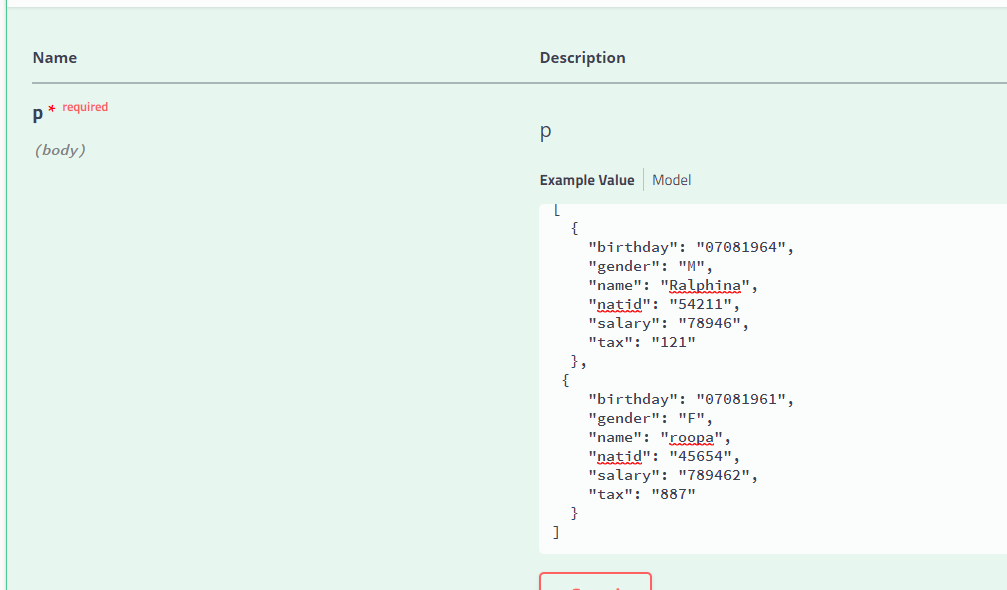


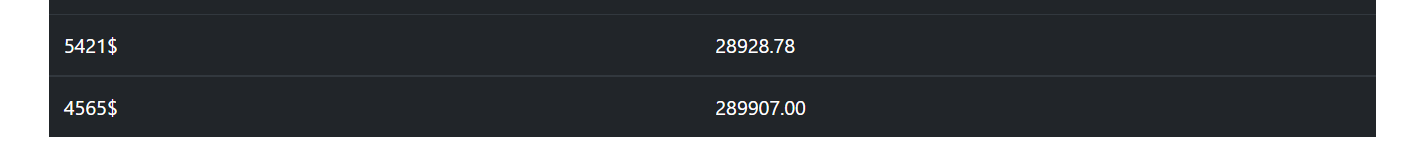
There is a bug, when we enter single record of working class via API and refresh tax relief table.

Two entries with same natid and tax relief are getting populated.

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

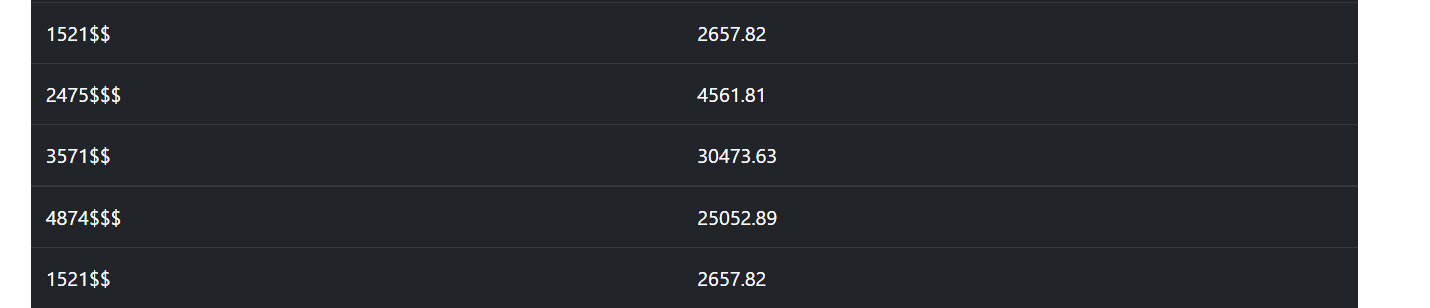
**My observation/test result: All acceptance criteria are met**

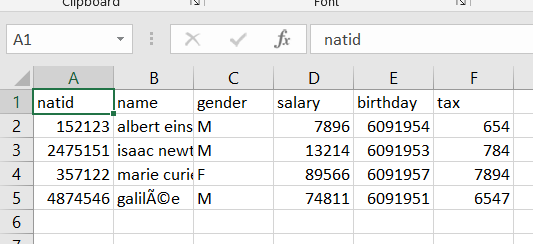




3: As the Clerk, I should be able to upload a csv file to a portal so that I can populate the database from a UI

**My observation/test result: All acceptance criteria are met**

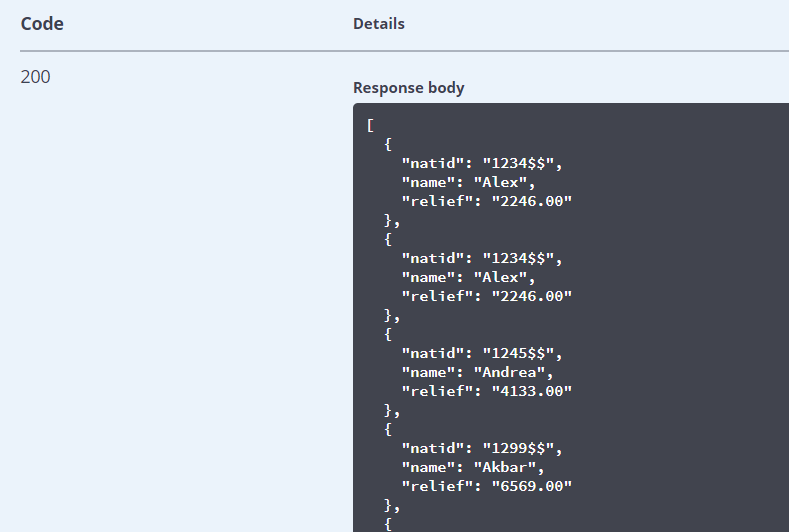




AC4: As the Bookkeeper, I should be able to query the amount of tax relief for each person in the database so that I can report the figures to my Bookkeeping Manager

**My observation/test result: All acceptance criteria are met**

Using the API endpoint[**/calculator/taxRelief**](http://localhost:8080/swagger-ui.html#/operations/calculator-controller/getTaxReliefUsingGET)**, all the three details are**



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

**My observation/test result: All acceptance criteria are met**

Dispense now button working fine.



**What are the test cases you’ve covered ?**

**Total testcase covered mentioned below:**

**Functional test cases – testing functional requirements**

1. Make the home page of the properly displayed without any error in GUI – **PASSED [Automated]**
2. Rack the data base with random entries with some count (say 10) – **PASSED [Automated]**
3. Refresh the tax relief table and verify 10 entries are getting populated – **PASSED [Automated]**
4. Verify the tax relief summary is showing proper count subjected to two decimal number – **PASSED [Automated]**
5. Insert details of one person (birthday, gender, salary, tax, name, natid) and verify the details are updated in the database by refreshing tax relief table in GUI – **PASSED [Automated]**
6. Insert details of multiple person (birthday, gender, salary, tax, name, natid) and verify the details are updated in the database by refreshing tax relief table in GUI - **PASSED**
7. Verify cash dispense button is shown in red colour and after clicking on “Dispense Now”, cash dispensed message is displayed – **PASSED [Automated]**
8. Upload a CSV file which has all the records (birthday, gender, salary, tax, name, natid) properly updated and upload it to the GUI. Verify data’s are updated in the database by refreshing tax relief table in GUI - **PASSED**

**How do you test for functional requirements?**

**Non-Functional test cases - testing functional requirements**

1. Verify the booting time of the application – It should be less than a minute - **PASSED**
2. Refresh the tax relief table and make sure getting message as ‘No records are available now” **- PASSED**
3. Even through there is not data in database, clicking on dispense now button is showing message as “Cash dispensed” **- FAILED**
4. Insert details of one person with wrong date (more than 31) (birthday, gender, salary, tax, name, natid) and verify the records are not populated & error code 500 is thrown – **PASSED [Automated]**
5. Insert details of one person with wrong date(31-Nov – which doesn’t exist) (birthday, gender, salary, tax, name, natid) and verify the records are not populated & error code 500 is thrown – **FAILED [Automated]**
6. Insert details of one person with wrong month(greater than 12) (birthday, gender, salary, tax, name, natid) and verify the details are not updated in the database by refreshing tax relief table in GUI **PASSED [Automated]**
7. Insert details of one person with wrong year(greater than 4 digits) (birthday, gender, salary, tax, name, natid) and verify the details are not updated in the database by refreshing tax relief table in GUI **PASSED [Automated]**
8. Insert details of one person with wrong gender(S instead of F/M) (birthday, gender, salary, tax, name, natid) and verify the details are updated in the database by refreshing tax relief table in GUI – **FAILED (wrong inputs are being accepted without validation). [Automated]**
9. Insert details of one person with wrong salary(include any character say s) (birthday, gender, salary, tax, name, natid) and verify the details are not updated in the database by refreshing tax relief table in GUI – **PASSED [Automated]**
10. Insert details of one person with wrong tax(include any character say s) (birthday, gender, salary, tax, name, natid) and verify the details are not updated in the database by refreshing tax relief table in GUI – **PASSED [Automated]**
11. Insert details of one person with wrong name(include any integer) (birthday, gender, salary, tax, name, natid) and verify the details are updated in the database by refreshing tax relief table in GUI – **FAILED [Automated]**
12. Upload a CSV file which has one records or one column missed say tax column (birthday, gender, salary, tax, name, Natid) properly updated and upload it to the GUI. Verify upload fails with error code 500 – **PASSED**
13. Upload a CSV file which has all the record but with some redundant (name with symbols) (birthday, gender, salary, tax, name, Natid) properly updated and upload it to the GUI. Verify upload fails with error code 500 or particular entry should not be updated – **FAILED**

Steps on how to run the tools:

1. Install git hub and clone the mentioned directory for the given link oppenheimer-project-dev.
2. Install JAVA latest version in the laptop.
3. Run the command “java -jar C:/Users/sarva/newproject/oppenheimer-project-dev/OppenheimerProjectDev.jar”
4. If running the above command gives you error, run jarfix after the JAVA installation.
5. Put the above command in bat file and trigger the bat file.
6. Tool can be accessed from GUI the link <http://localhost:8080/>
7. Tool can be accessed from API using the link <http://localhost:8080/swagger-ui.html>
8. Once the application is started successfully, we will get the below prints in the command prompt.

Dec 08, 2022 11:20:31 PM org.apache.coyote.AbstractProtocol start INFO: Starting ProtocolHandler ["http-nio-8080"] 2022-12-08 23:20:31.812 INFO 24912 --- [ main] o.s.b.w.e.t.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''2022-12-08 23:20:31.815 INFO 24912 --- [ main] c.g.s.SampleserviceApplication : Started SampleserviceApplication in 7.826 seconds (JVM running for 9.356)

**Robot Frame work details:**

C:\Users\sarva>pip show robotframework

Name: robotframework

Version: 6.0.1

Summary: Generic automation framework for acceptance testing and robotic process automation (RPA)

Home-page: https://robotframework.org