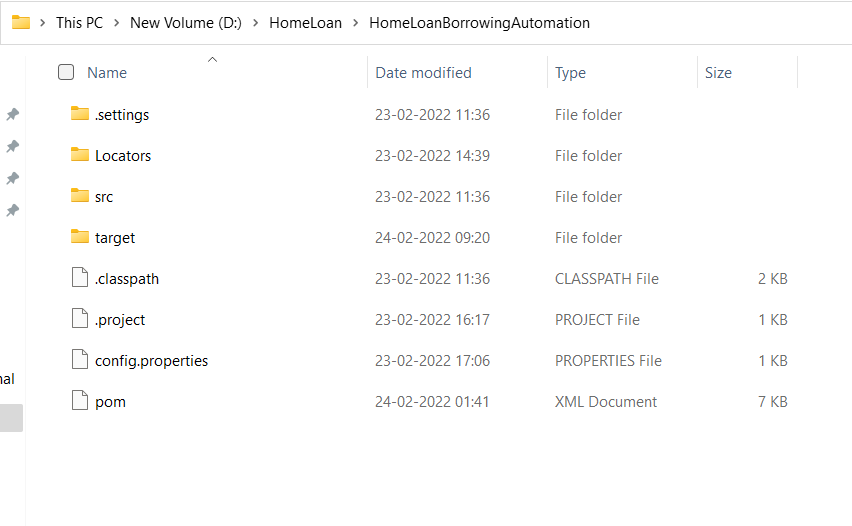
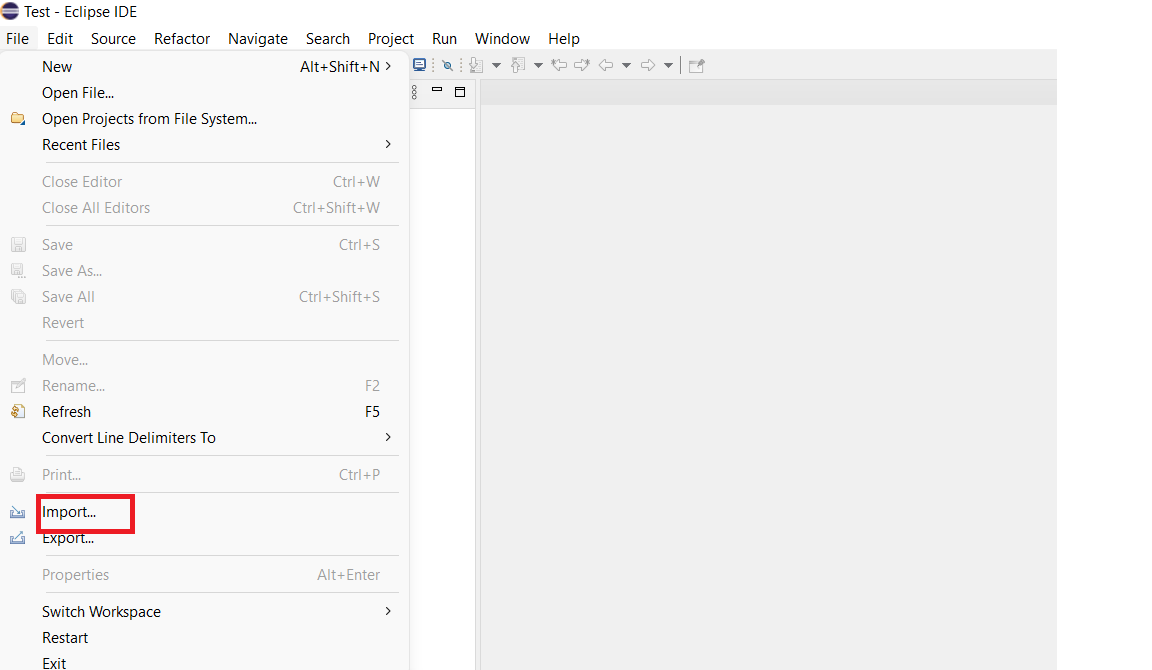
**How to Build the Project**

1. Open this URL <https://github.com/onkark-kulkarni/HomeLoanBorrowingAutomation>
2. Clone this project on your system
3. After cloning folder look like ,

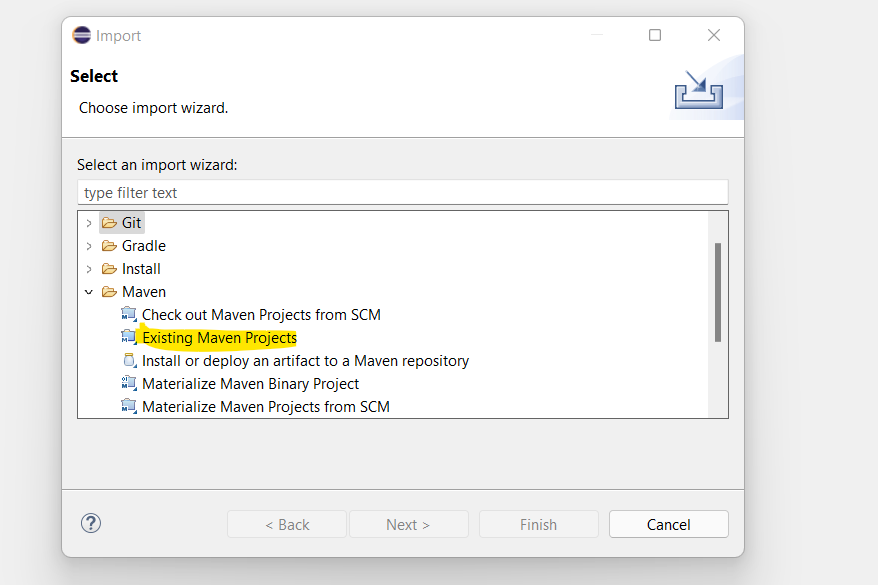


1. If you want to check the code then open any IDE like eclipse,

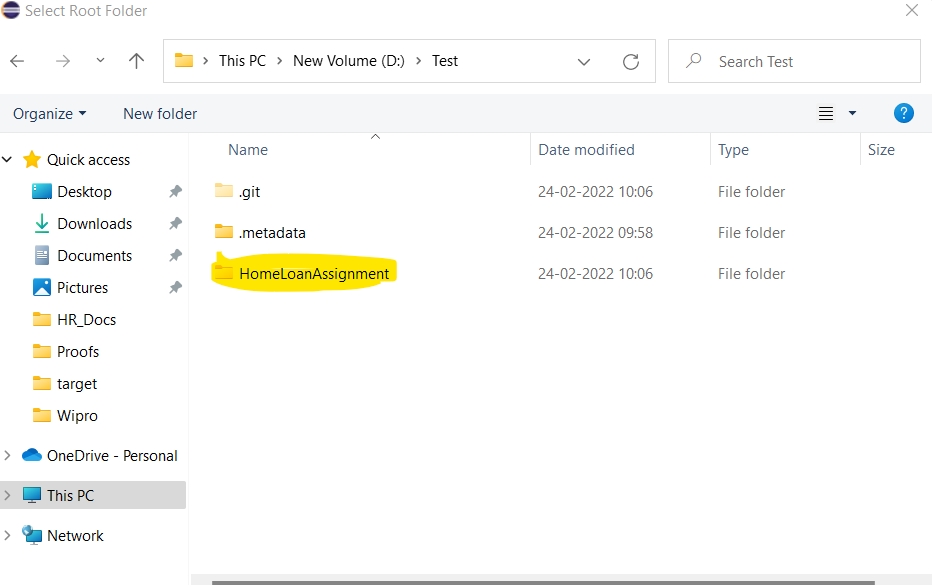
* Click on File and Import project



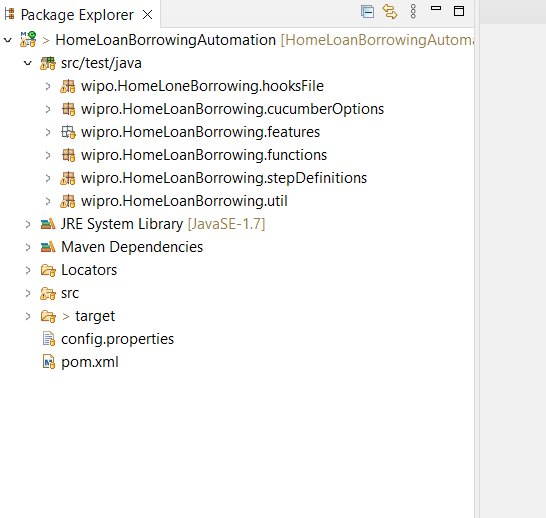
* Now select existing maven project option ad click on next button.



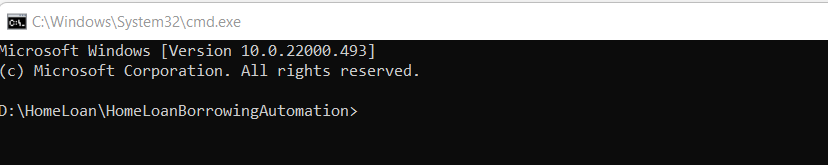
* Select the project name

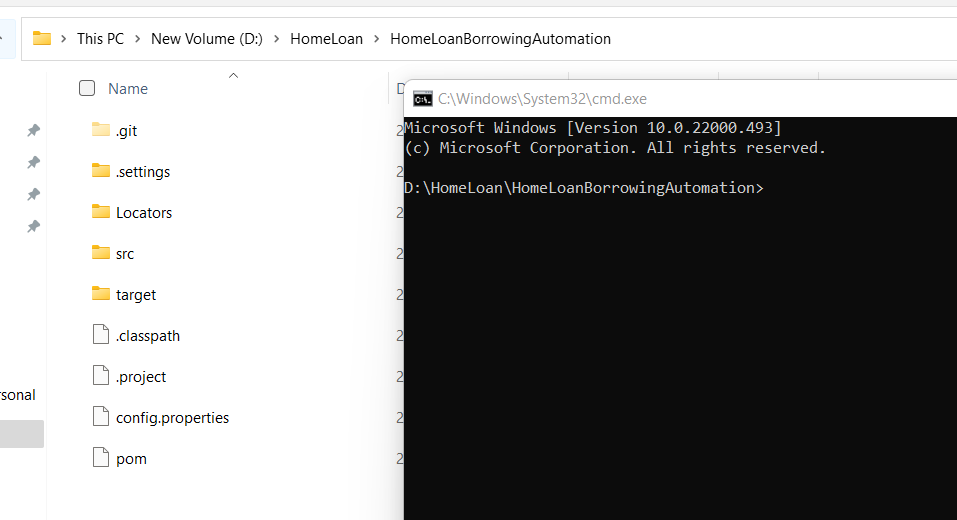


* Click on Finish button and after that project will looke like,

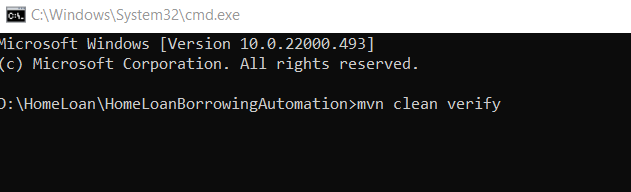


1. Open Command prompt from Project location





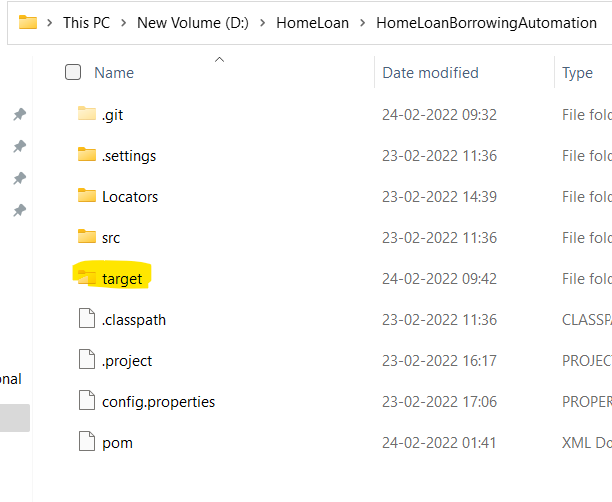
1. Type maven command as “mvn clean verify”



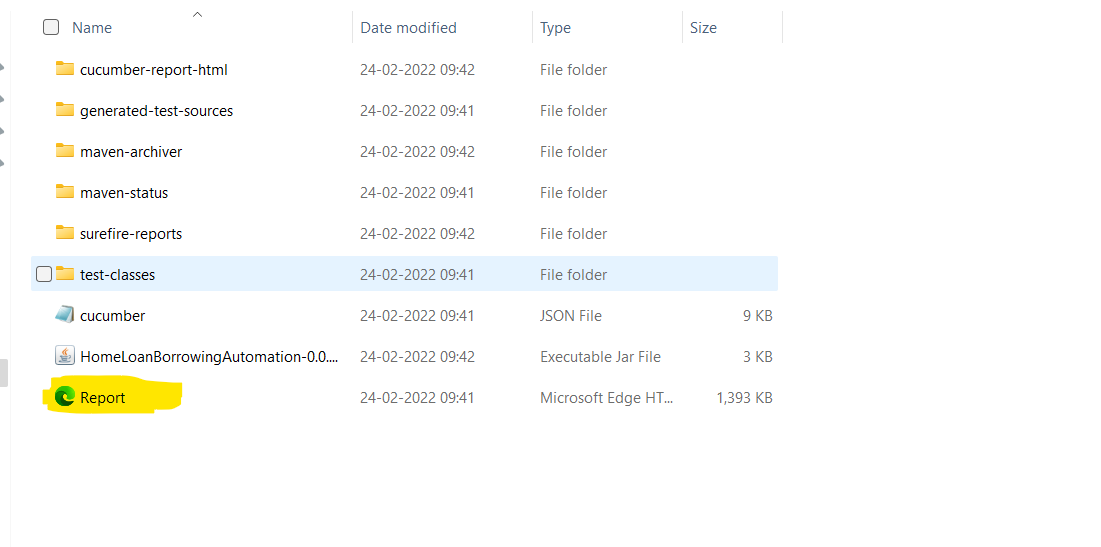
1. After the above step execution will be started, wait till execution gets finished.
2. **Hear build is getting failed as 2 Scenarios are failing as they don’t meet the expectations.**

* **Estimated borrowing amount is different from expected amount.**
* **On entering only living expenses one message is displayed, that message does not match with expected message (change in phone number).**

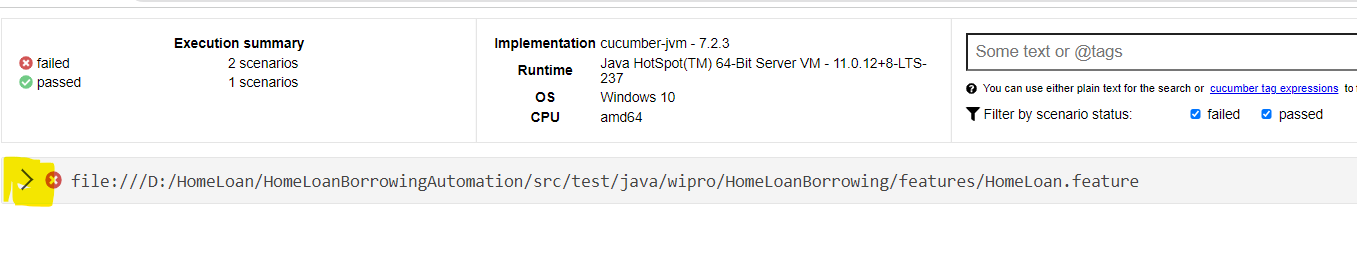
1. After the browser gets closed minimise the “cmd” and (after 5 to 10 sec wait for report generation) go to folder location and open **target** folder.



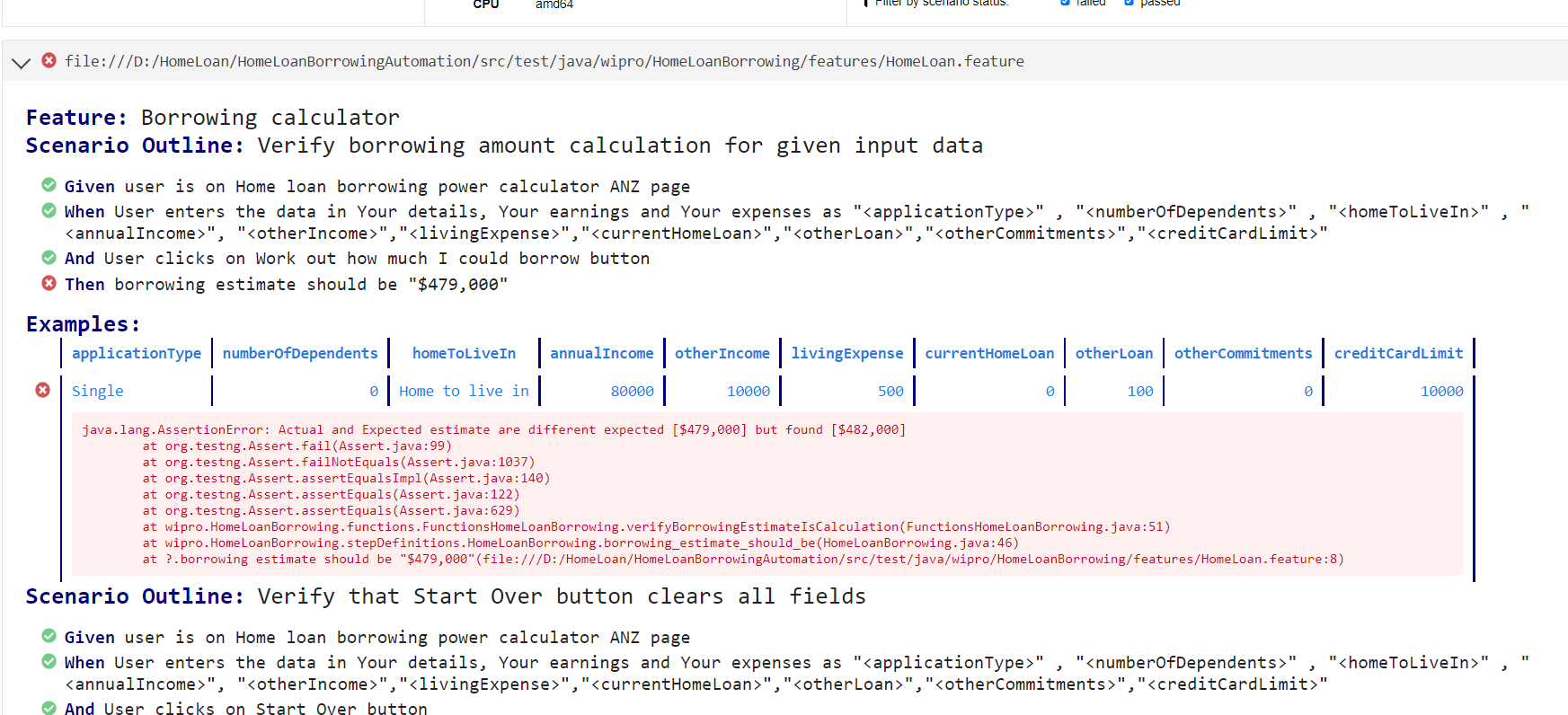
1. **From this folder you will get 2 types of reports one is simple and second one is advance report**,
2. Open Report.html, right click on it and open with chrome



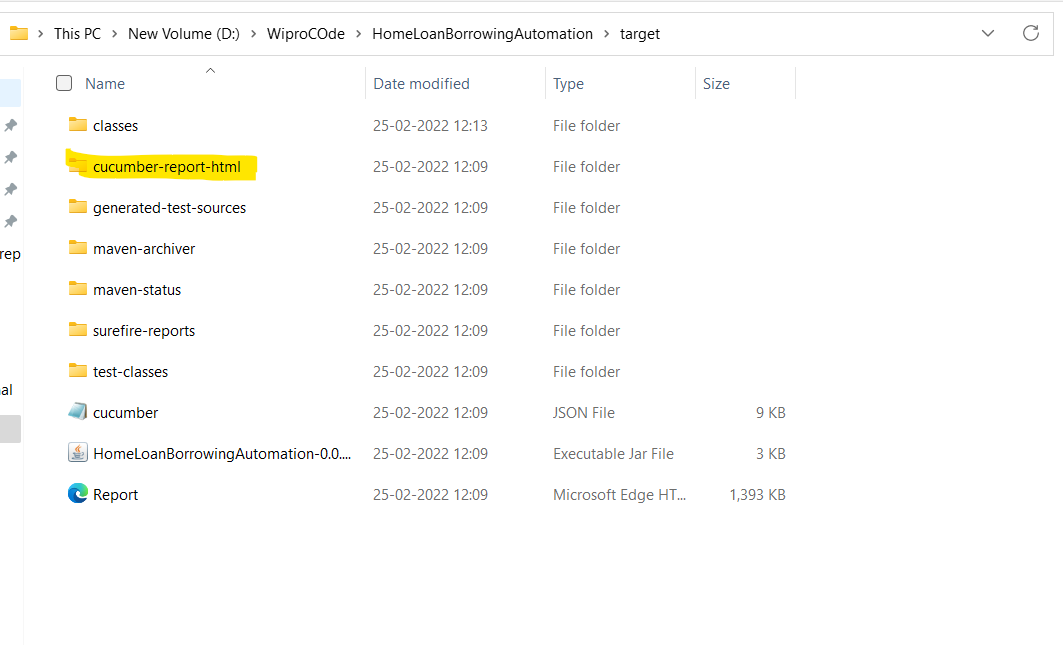
1. Click on the highlighted arrow

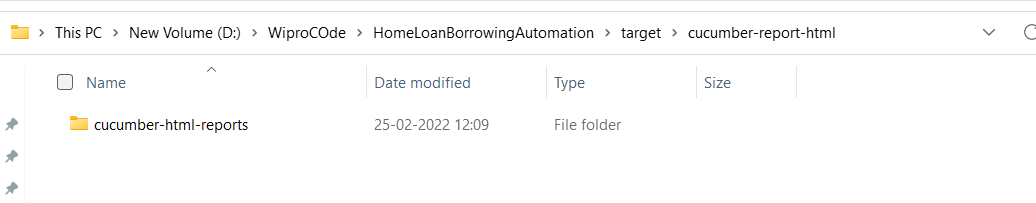


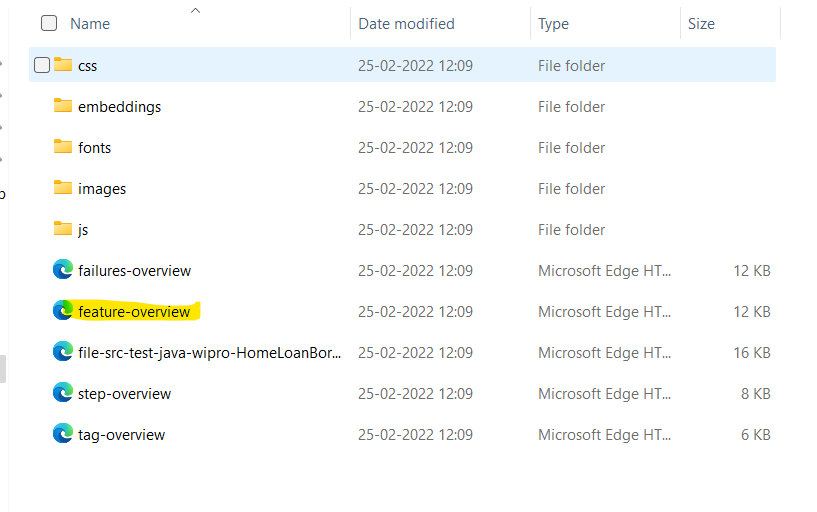
1. Here you will get all the detailed information on test activity



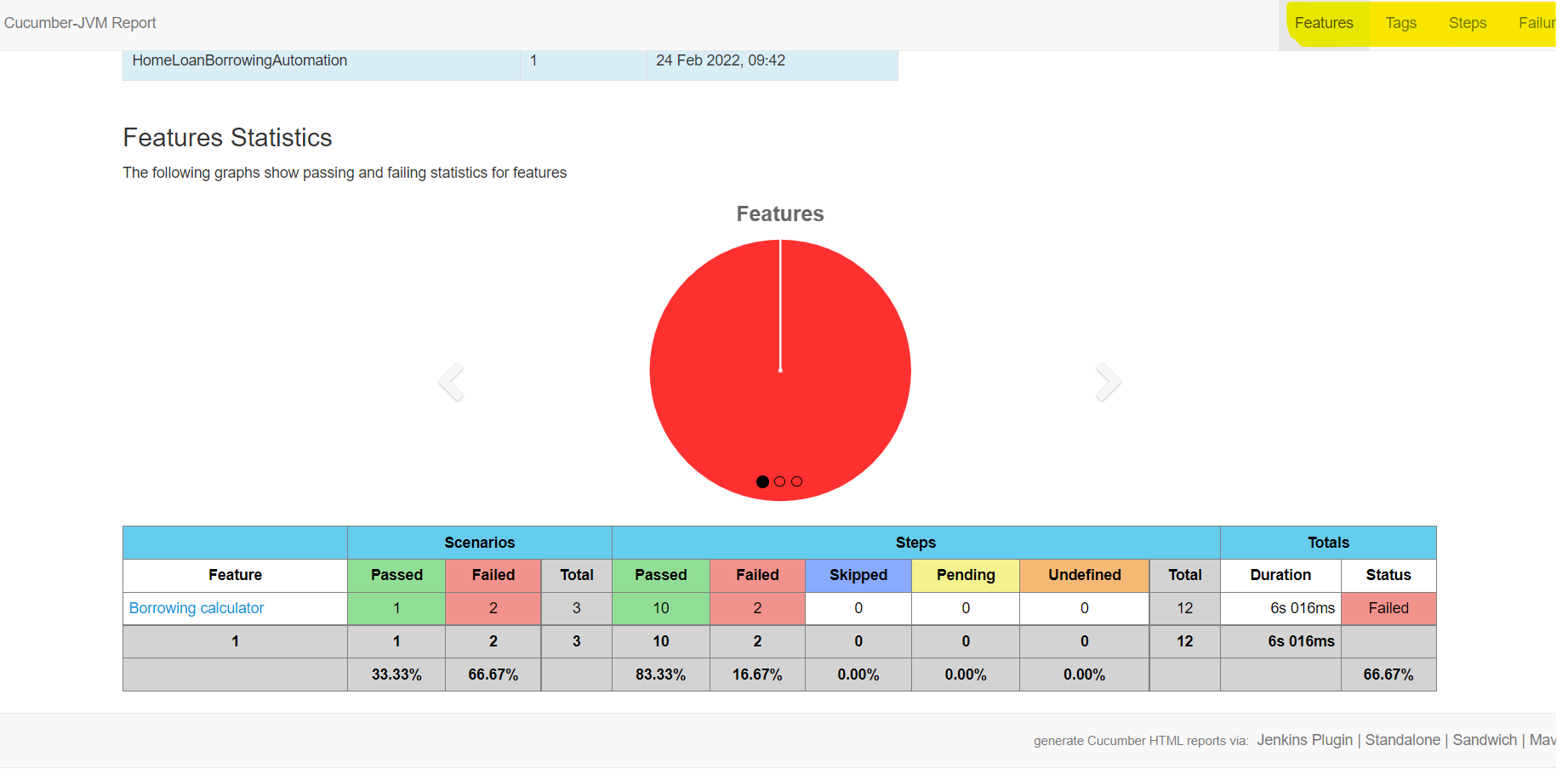
1. **There are advance reports also for that again go to “target” folders and open “cucumber-report-html” and open “cucumber-html-reports”.**



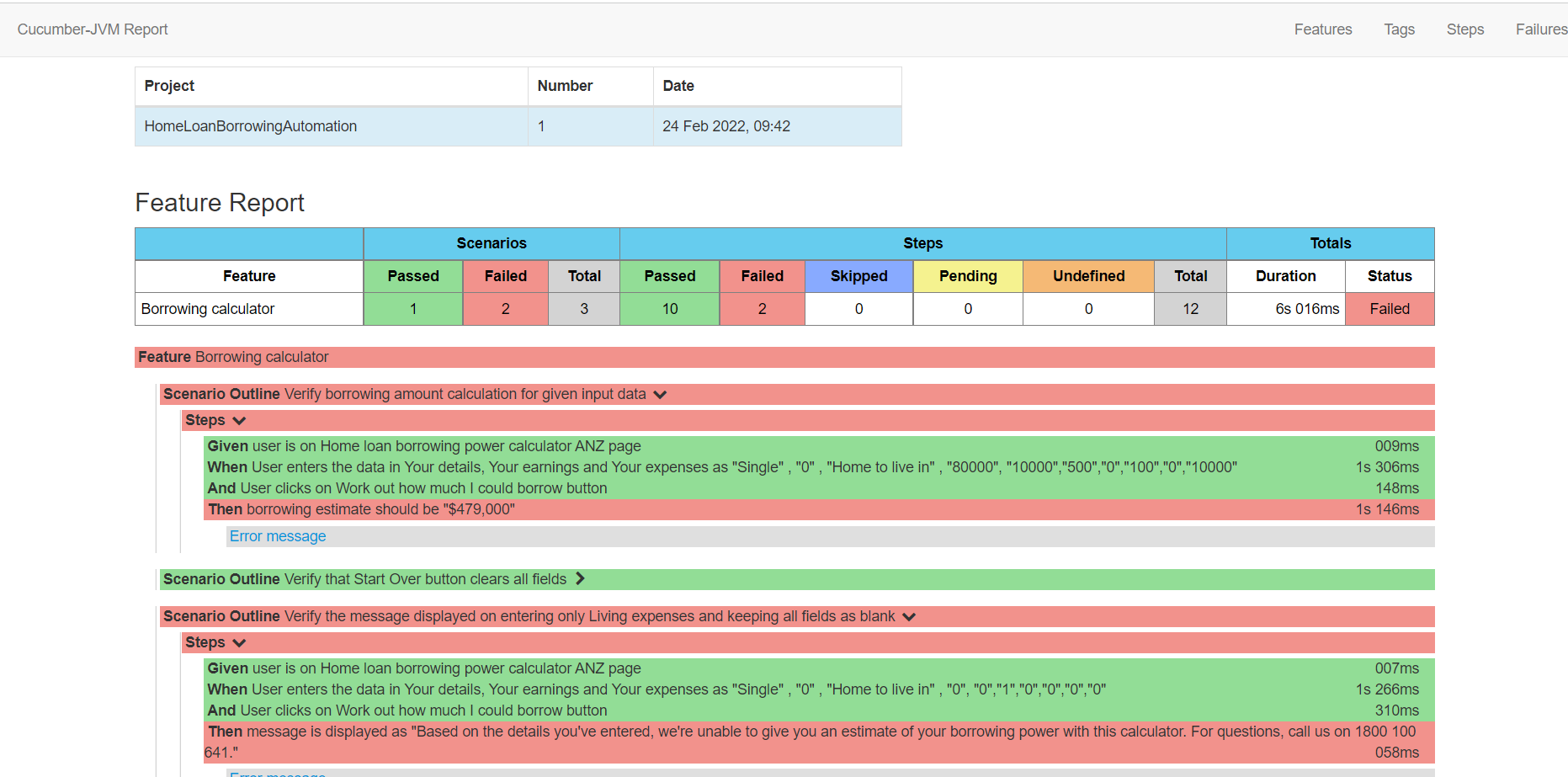




1. From here open “**feature-overview**” in chrome browser.



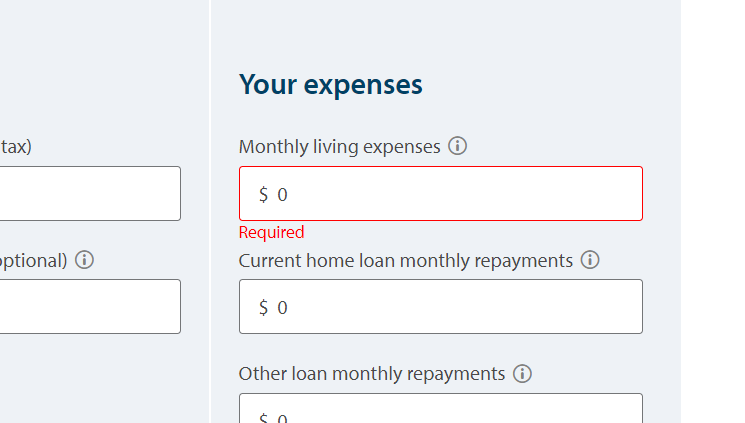
1. On above page click on different options from top right corner and you will get all detailed information.



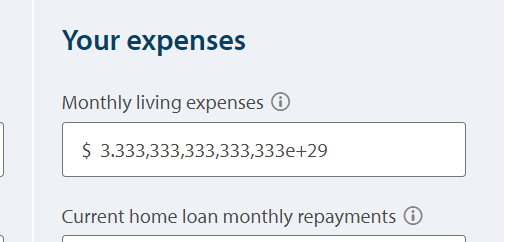
**Questions**

1. What other tests would you suggest could be written? You do not need to write them out in detail or code them.

* Entering high amount in annual income like 222222222 changes it’s value to “9,999,999”.
* Verify monthly living expense field is mandatory.
* Clicking on Start Over should remove the required error message



* For living expense field should have length limit



* Test cases to verify Application type and Property Buy toggles
* Verify Borrow amount with all available dependent options
* Verify editing functionality after clicking on work out button

2. If this test was part of a much larger test set, how would you speed it up?

* To speed up the we can cover all these scenarios with multiple test data and making them dependent on each other, so that if any of the scenario is failed, we don’t need to spend time on others.
* For common steps create common functions library and reuse the already performed steps rather than performing them again.

3. Sometimes UI tests can fail unpredictably. For example, the page may not have fully loaded before test automation attempts to click a button on a webpage. How would you improve reliability of these tests?

* We can add Implicit/Explicit/Fluent wait so that the automation will wait until the element will get displayed on UI.
* Add retry mechanism so that if any test case is failed, we can give one more execution to that test case.