This document helps in understanding my approach towards implementation of the SDET challenge.

1. I have used Selenium with Java as a programming language and written code in IntelliJ Idea.
2. I created maven project and executed the code in Firefox browser.
3. I created two java class files under \FetchCoding_AutomationProject\sdetchallenge\src\test\java
4. The file 'Inputdata_DynamicGeneration' is used to generate the user input bar selections to place in the weighing balance. I have used HashSet and ArrayList to generate the input data dynamically every time we run the testcase.
5. The file 'TestCase' has the actual test case steps which are used to perform actions on the webpage and find the fault bar.

Solution to the problem:

There are nine bars, and we need to find the fault bar with less times of using the weighing balance. I have divided the total bars into three groups and tried to weigh two parts and decide which group has fault in 1st iteration. In the 2nd iteration I used the same approach to find out the fault bar. In this way by using 2 times weighing, we can find the fault bar.

I have also recorded and added video of the execution and added it to the repository.