Kambi Code Test

Refactoring report

By: Farhad Mohseni | realfarhadmohseni@gmail.com

Essential modifications:

- The whole code is re-written with typescript in order to have strict type checking in the project which makes the software more robust and deployable

- File architecture is changed in order to separate development, build, and distribution related files

- I added the project into a git repository and uploaded that to GitHub for better project management

Code Refactorings:

- In order to have cleaner code, I moved User And Product entities to different classes and created separate enums for UserType and ProductType

- I removed the unnecessary switch statement and refactored the code from 20 lines to about 10 lines of code, then I extracted the rebate and product type price calculation codes to separate functions which made the calculate function about 3 lines of code.

- I removed the inline JS code from the index.html and move that to a separate JS file.

- I used webpack to bundle the whole typescript code into one minified bundle.js file for the browser, that's the only javascript file that loads at index.html.

- Since we are using typescript now, I also wrote the unit tests using chai and mocha + typescript, I moved the tests in a test directory, which can be executed by running `npm test`

- Because I chunked the code into separate functions, the code is more testable now, so instead of one test, a separate test is written for each function which leads to more code coverage and reliability

Potential Improvements :

- A nice UI can be implemented for parameters input
- The whole app can be dockerized and deployed to the cloud
- An API can be built for the code so it can be used as a microservice