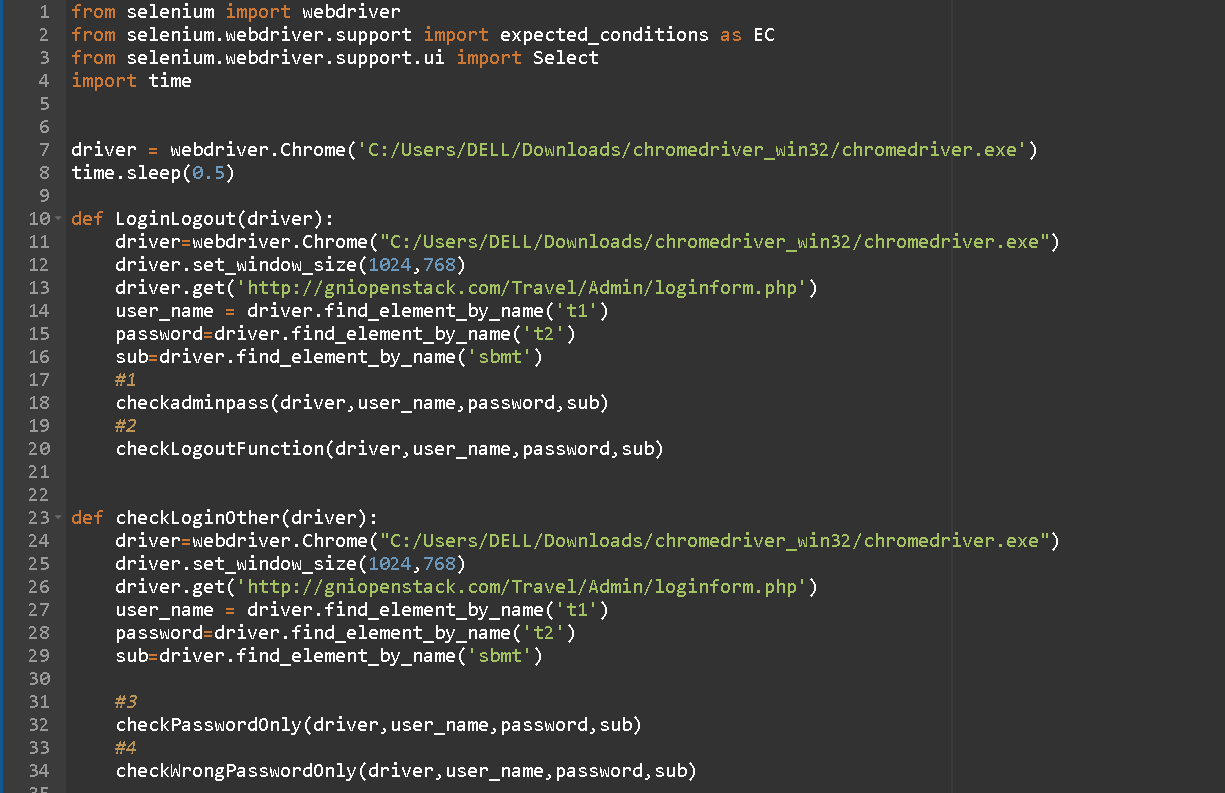
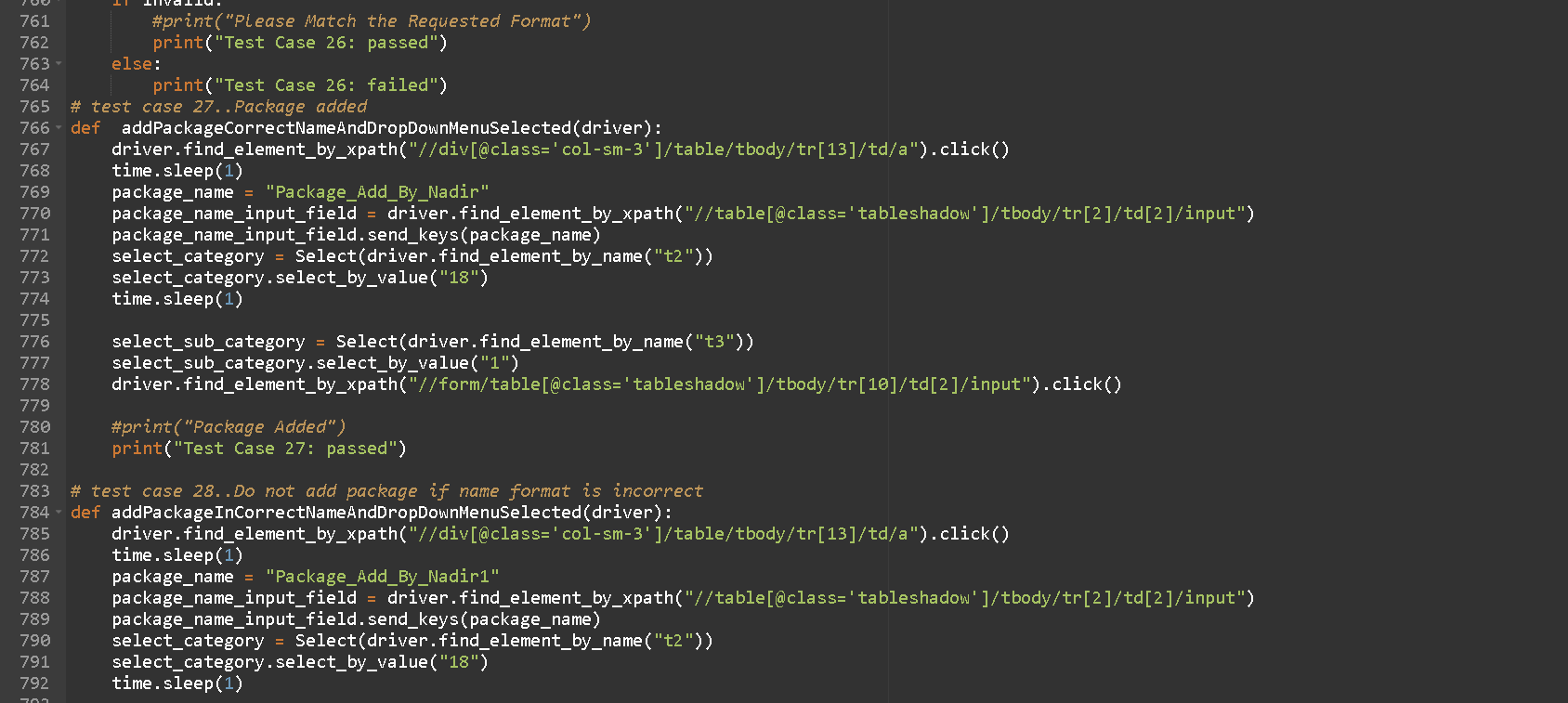
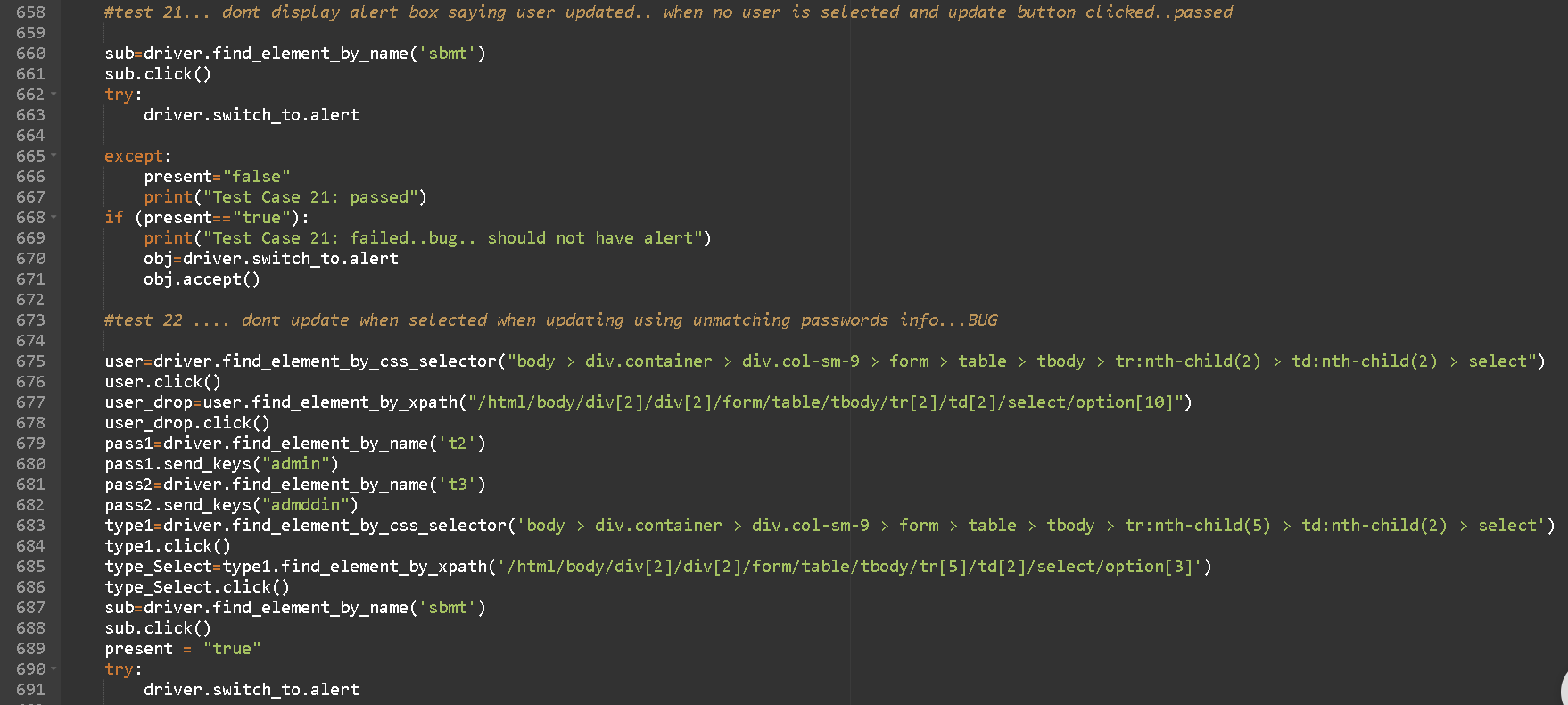
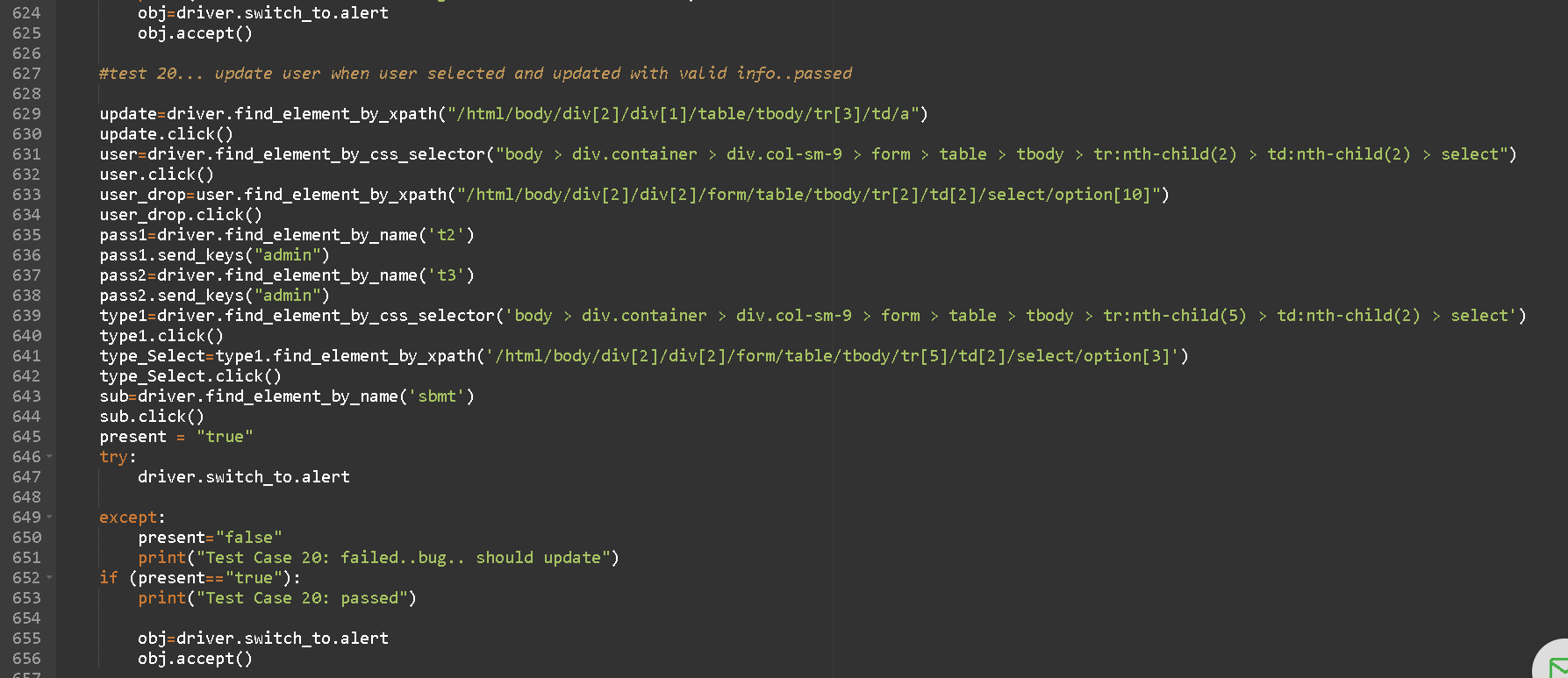
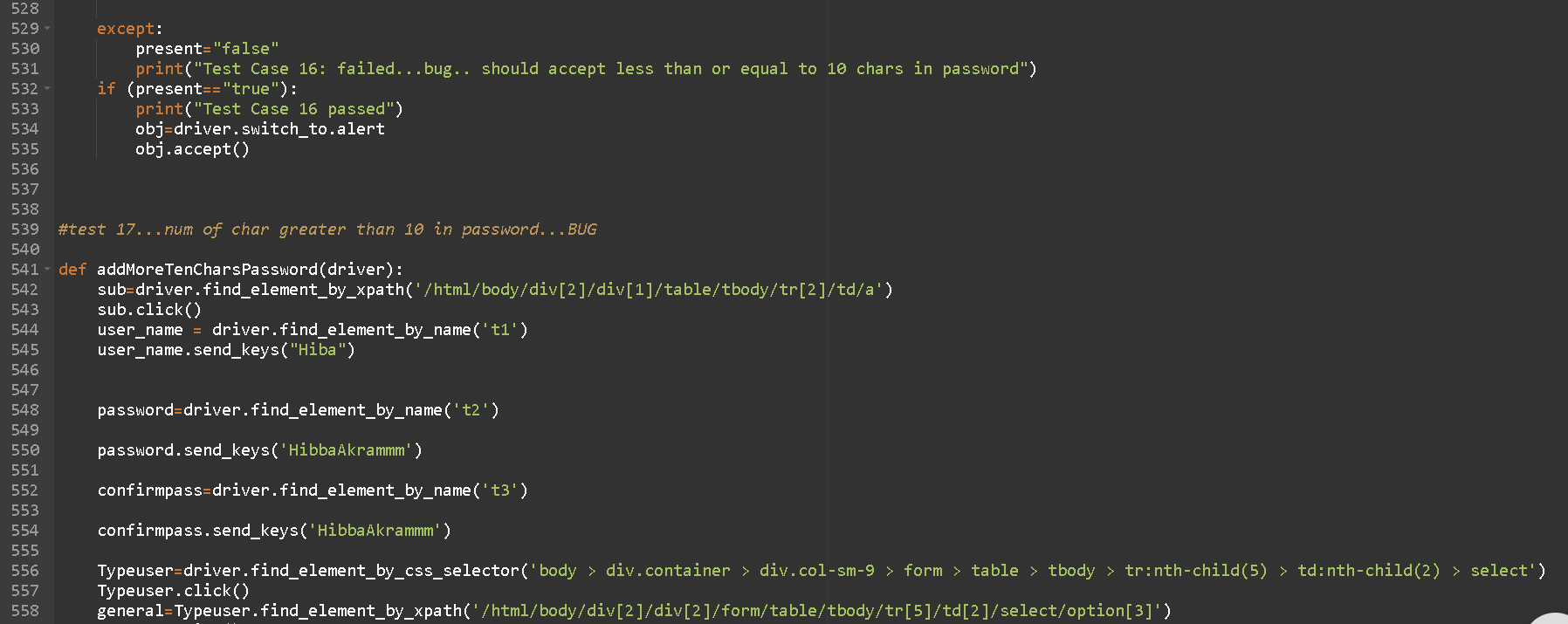
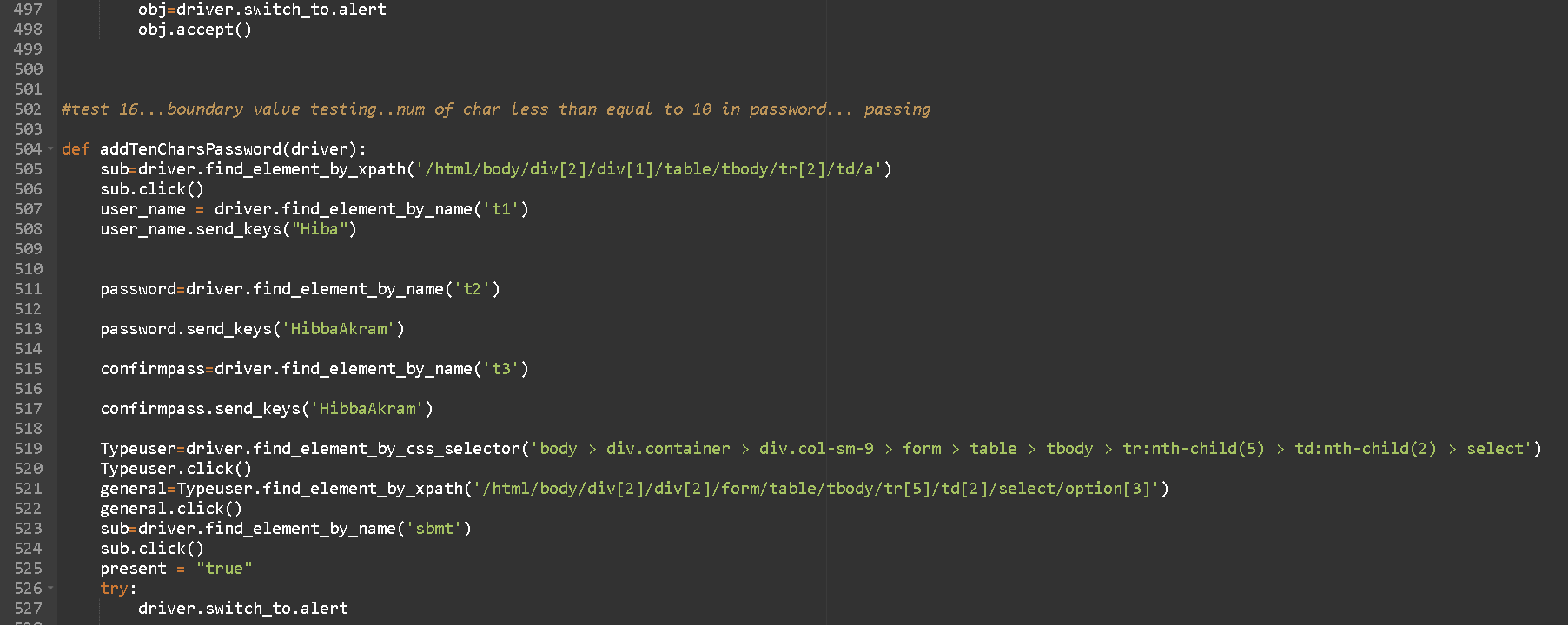
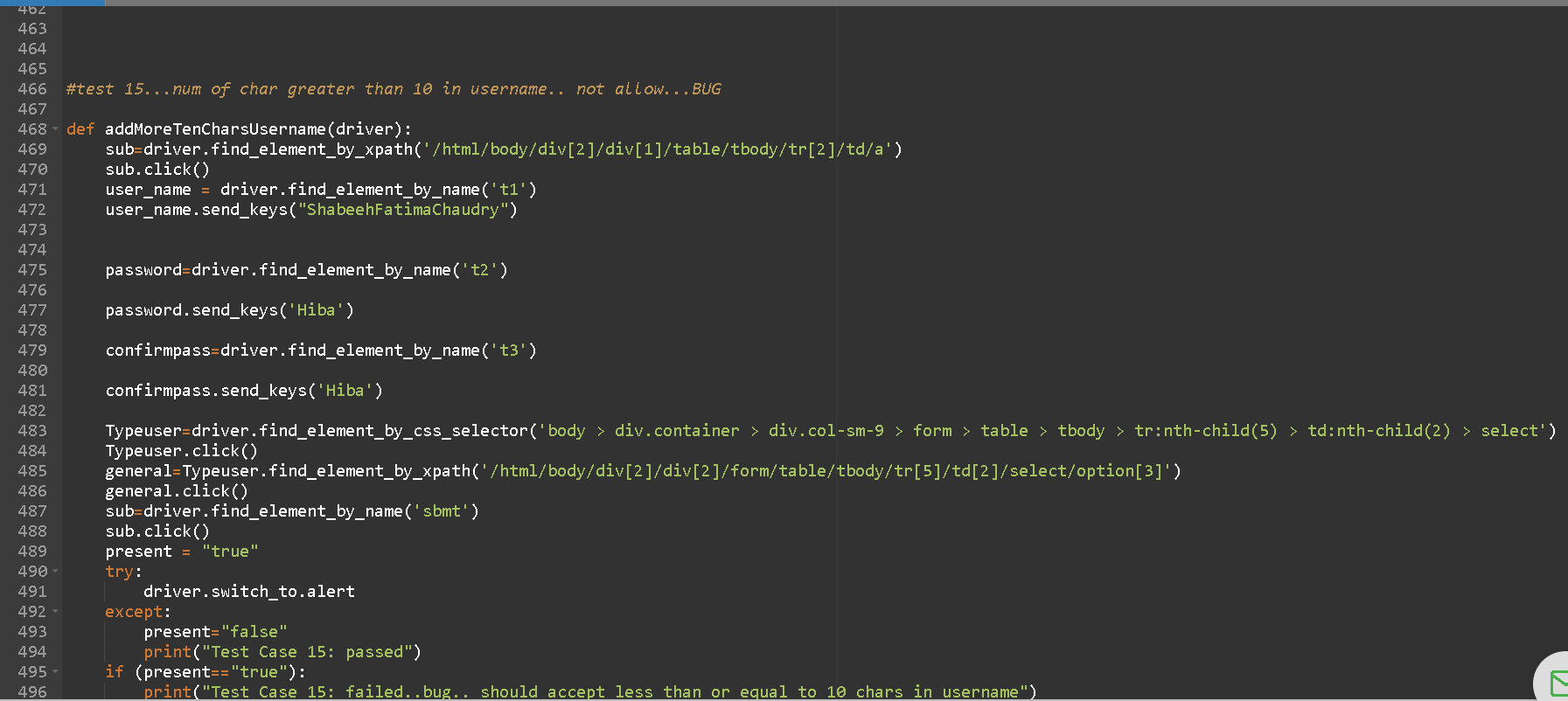
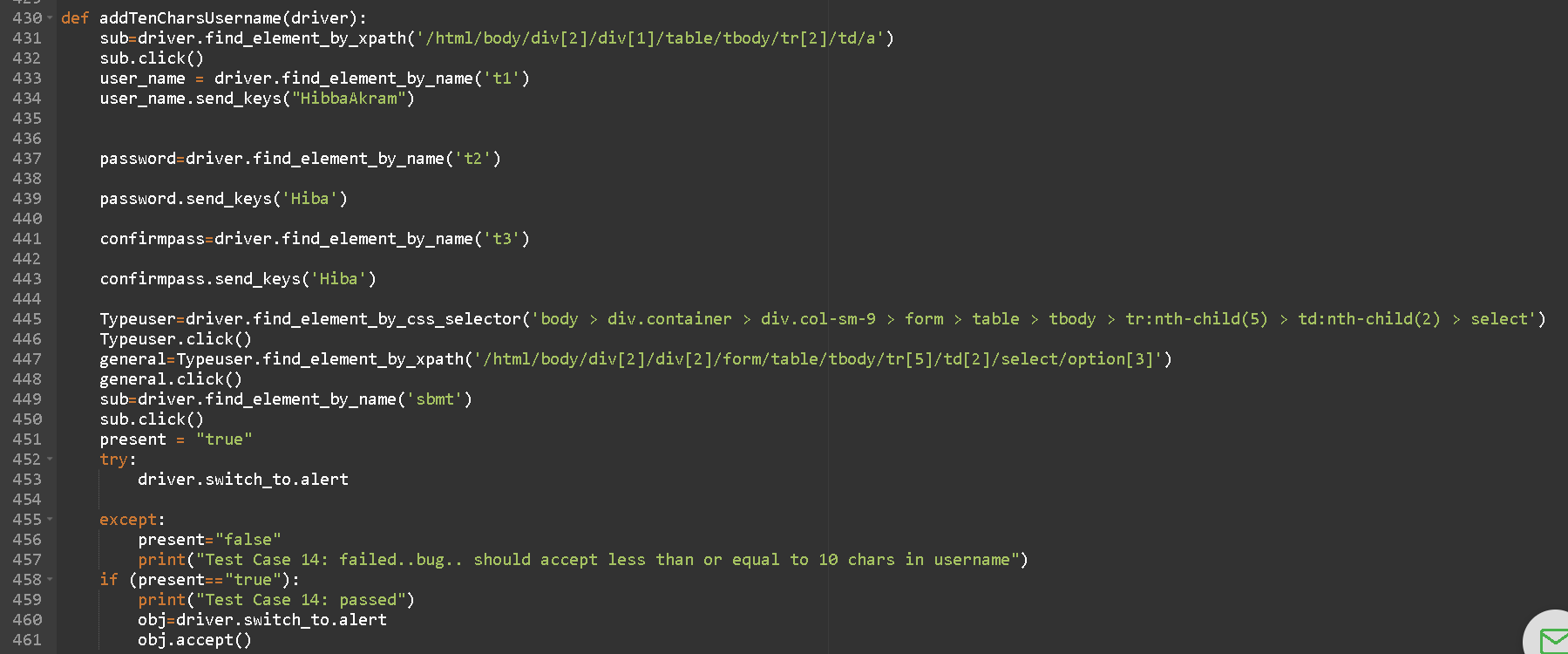
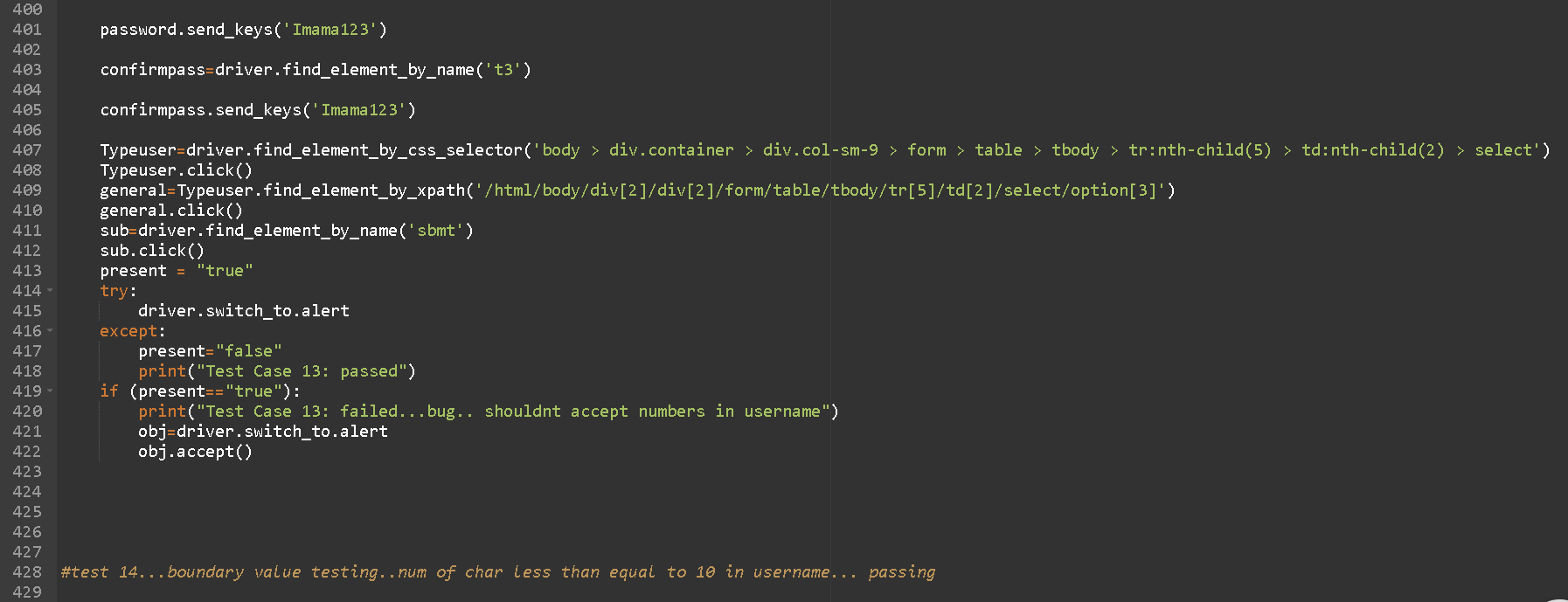
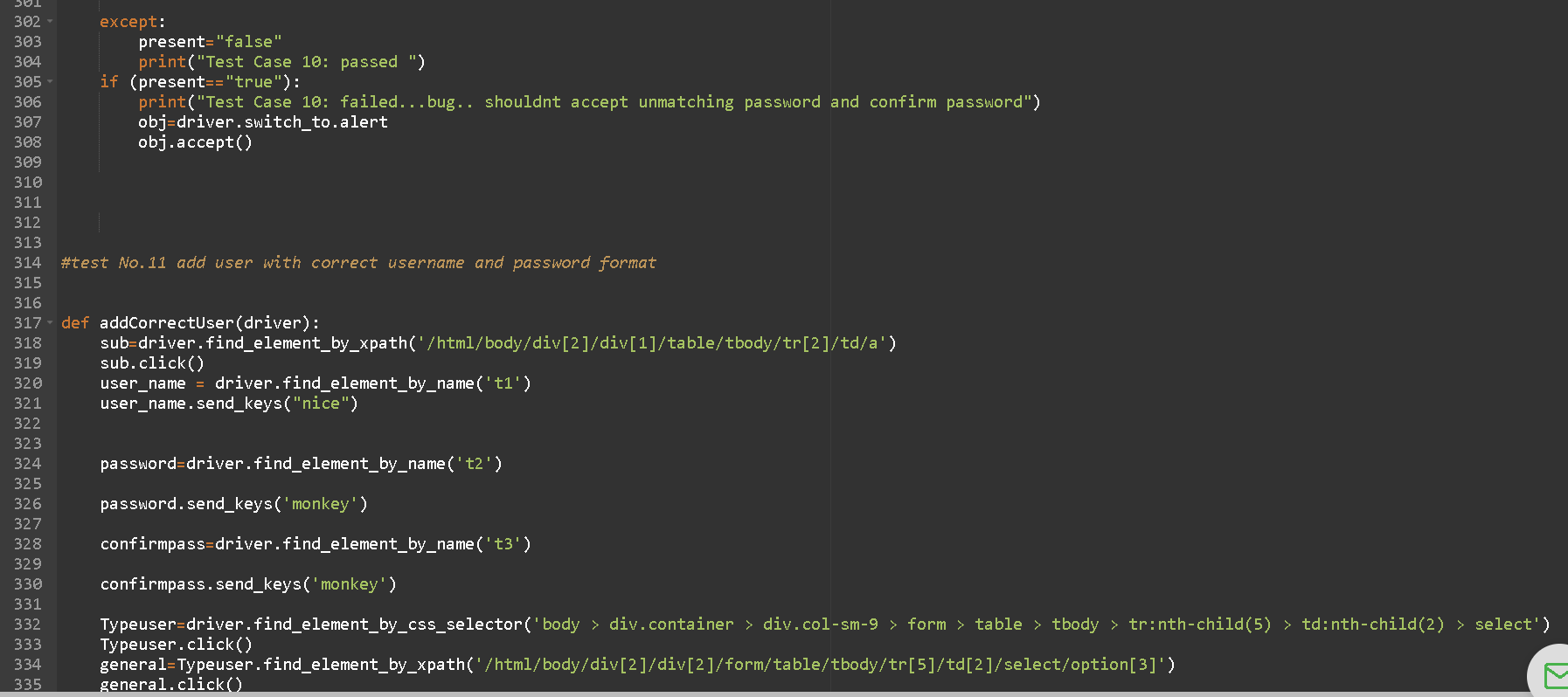
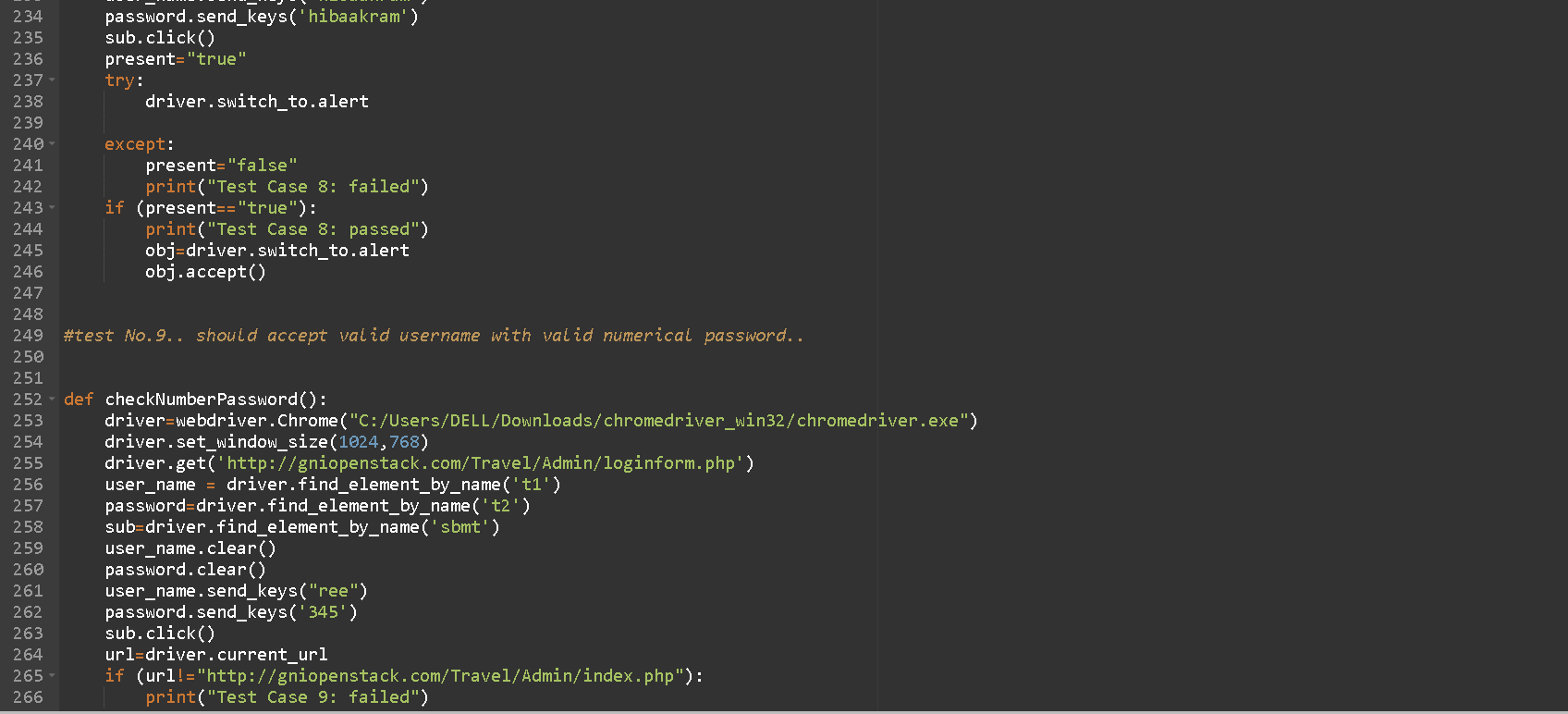
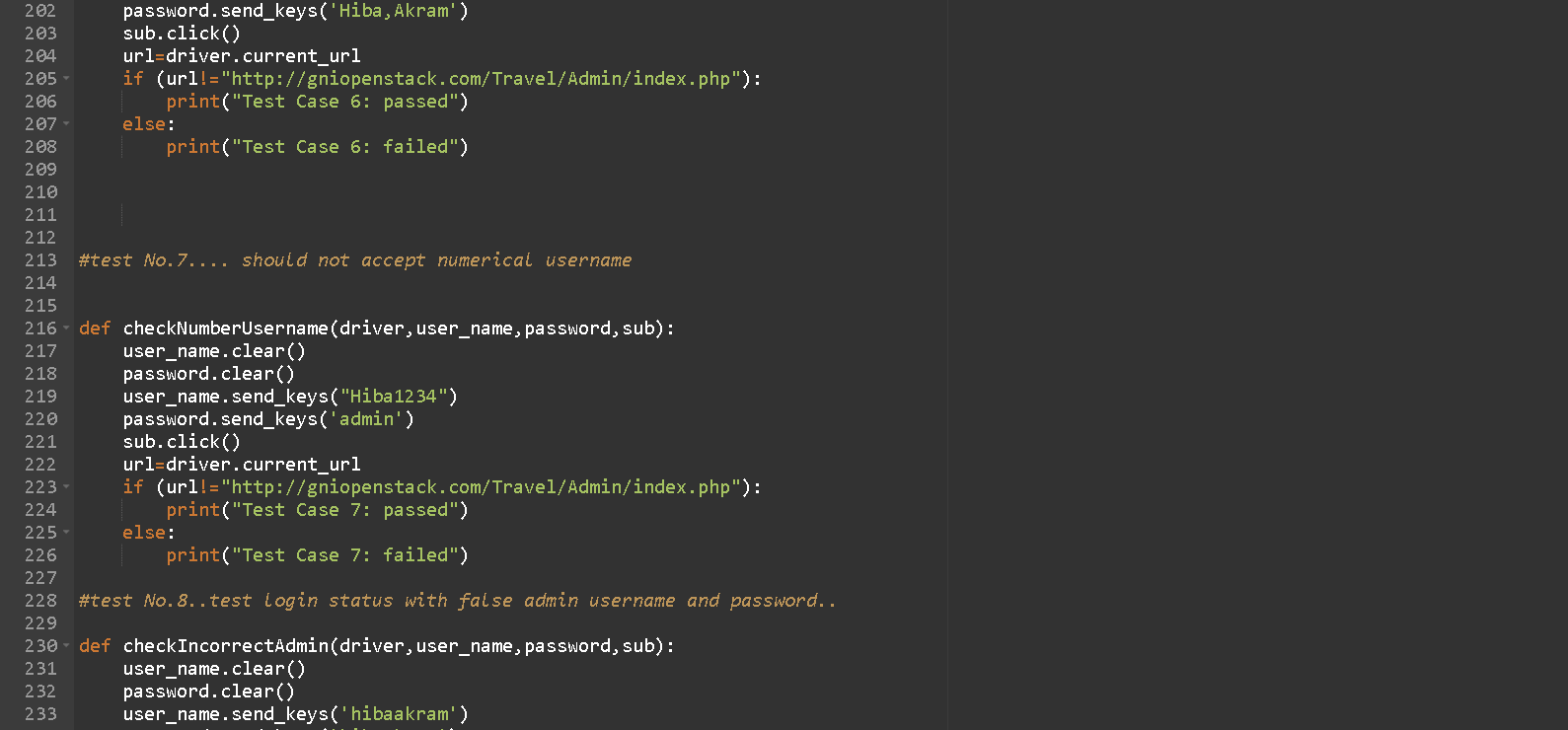
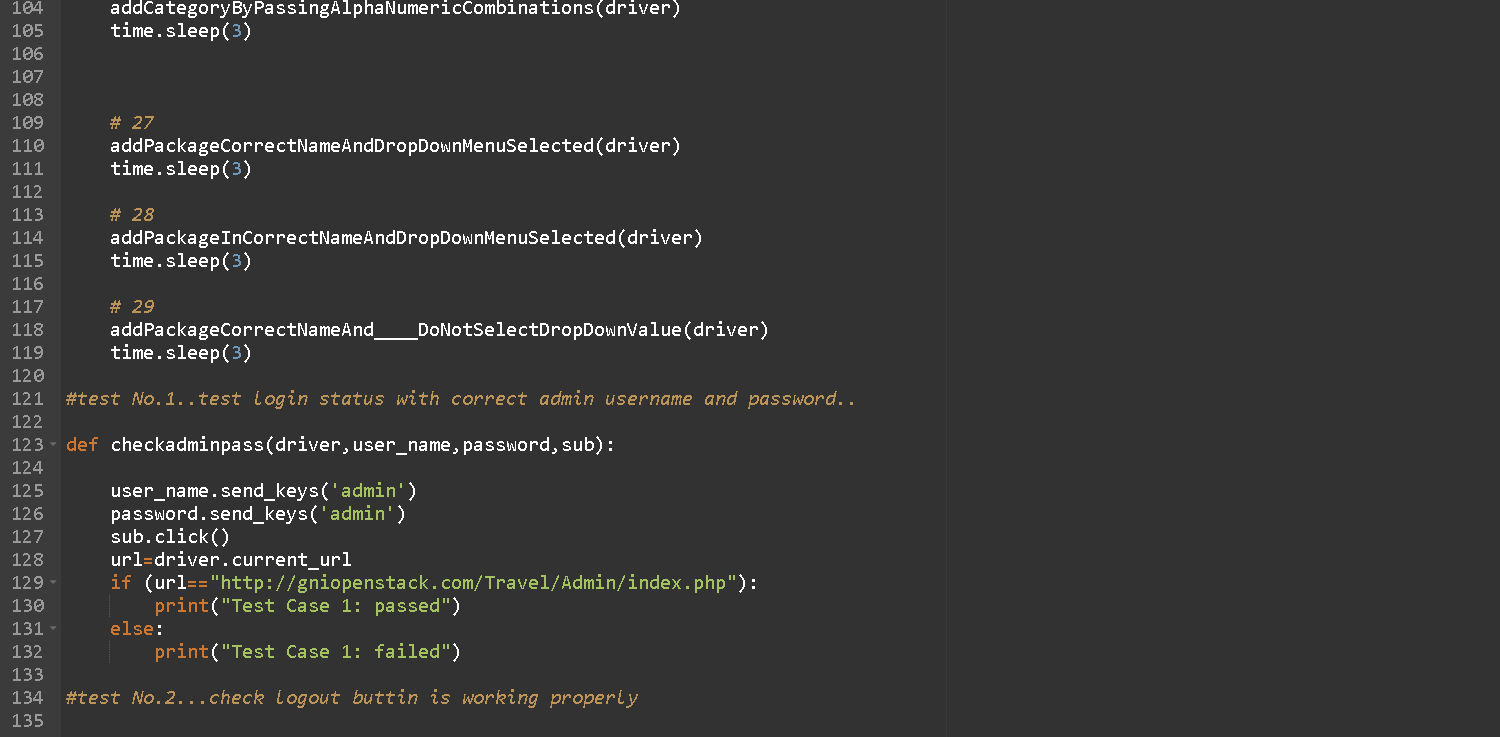
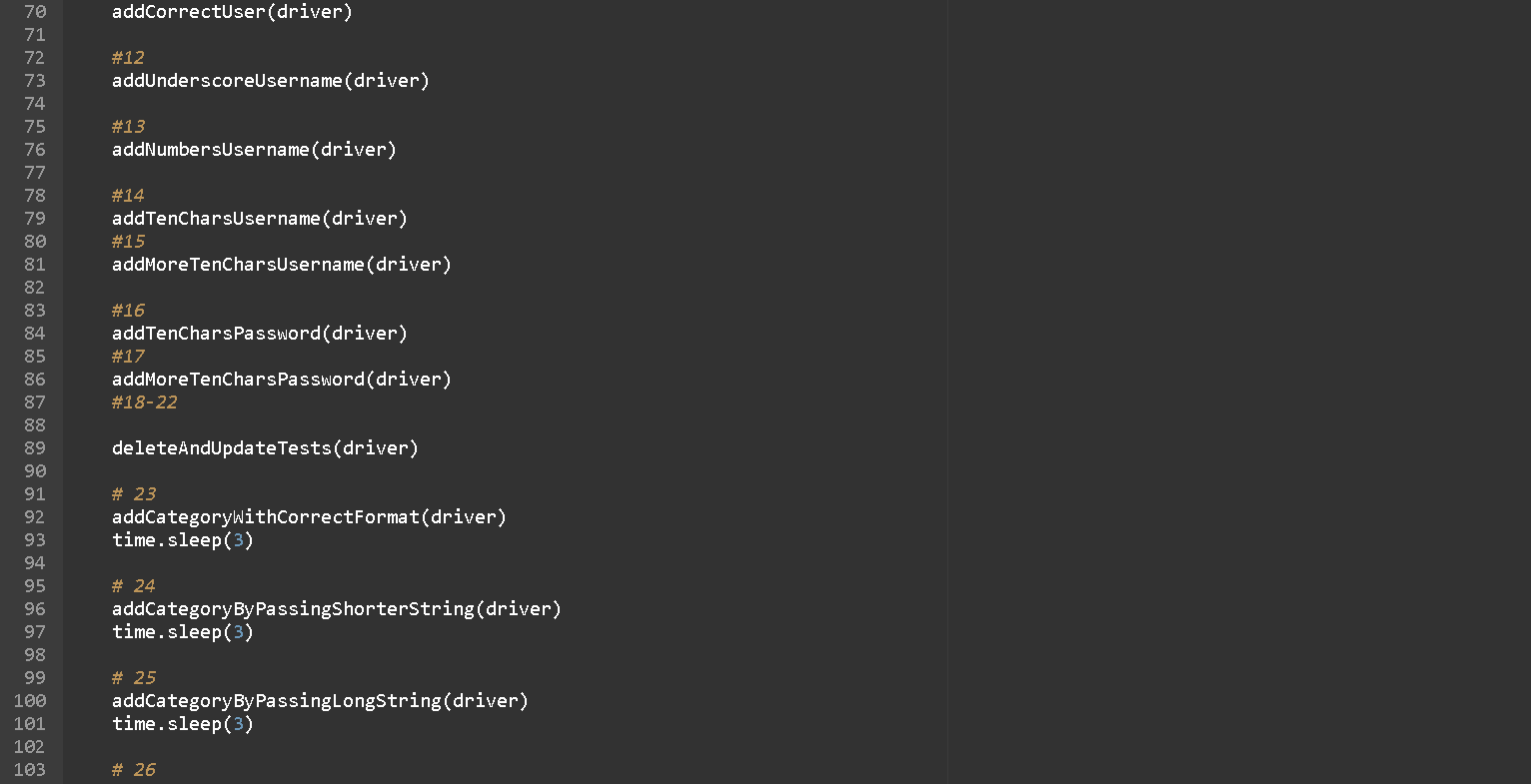
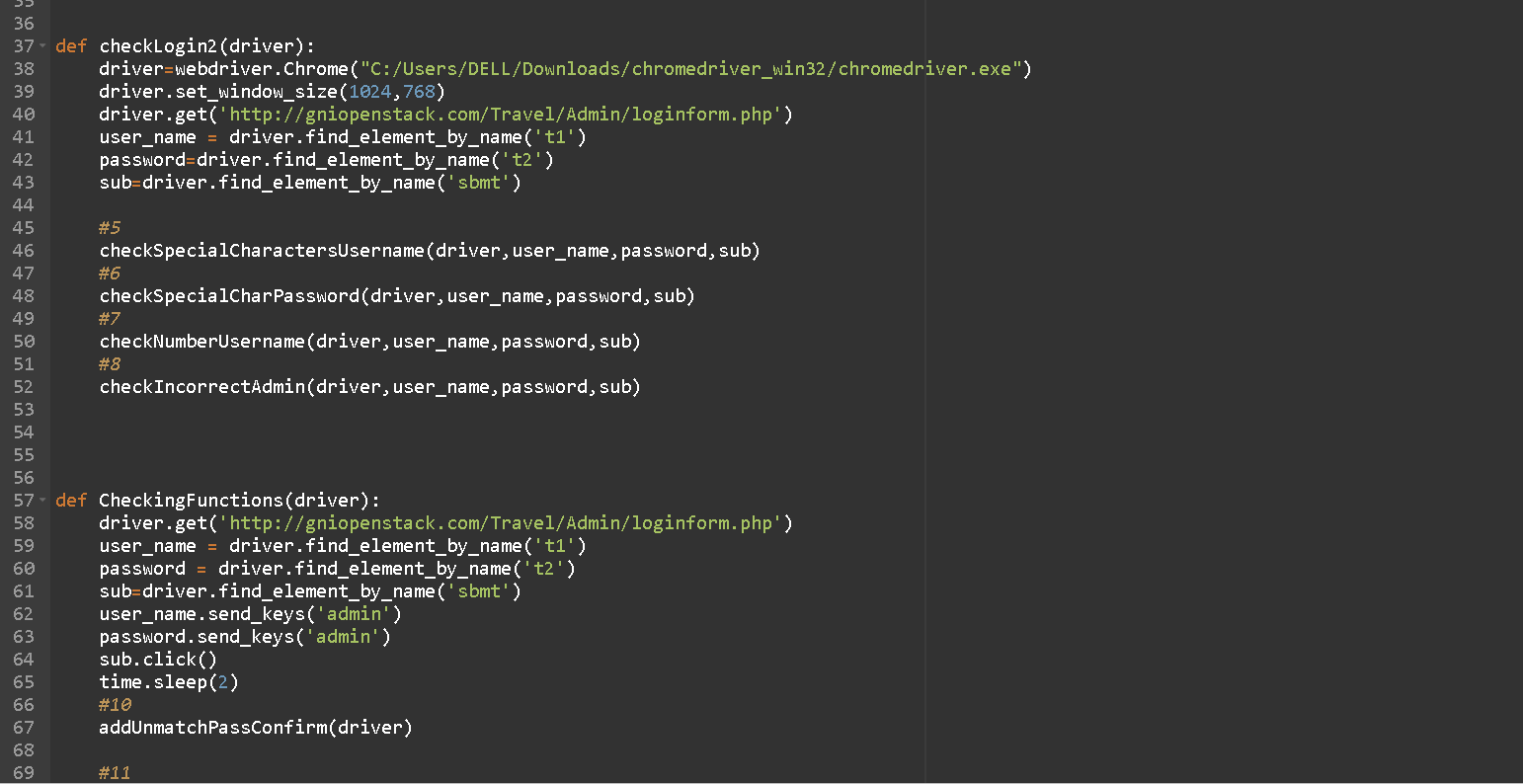
Selenium Testing:

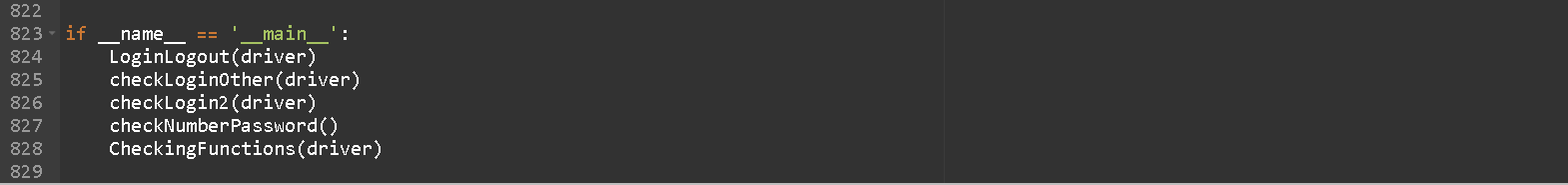
*Selenium Info*

Code:

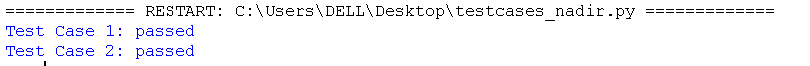
Inputs are generated through fuzzing



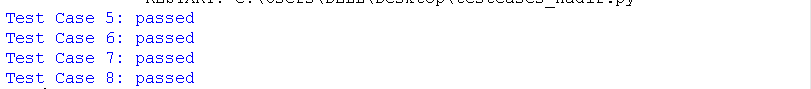


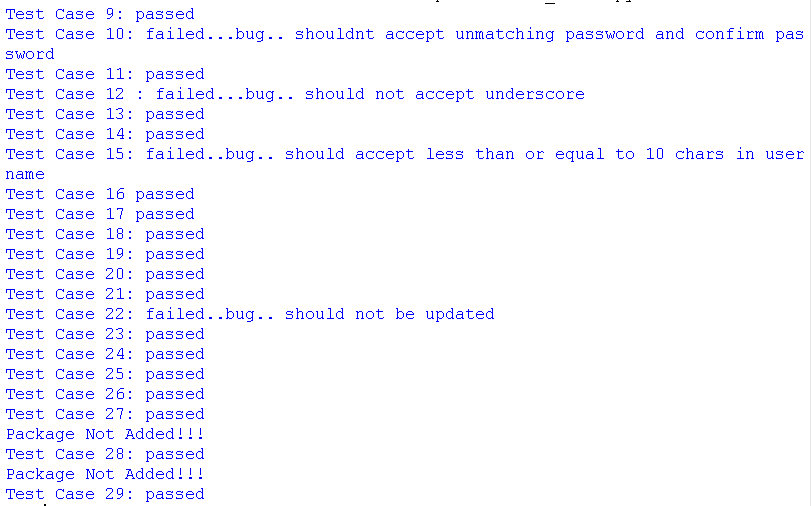


Outputs:









Test cases covered under Smoke Testing:

[Smoke testing](https://www.functionize.com/blog/smoke-testing-suite-what-it-is-why-you-need-it-and-how-to-automate/), also known as build acceptance testing (BAT), is a critical aspect of quality assurance that delivers quick and decisive answers on the viability of a particular build.

For web and GUI applications, smoke tests will typically cover essentials such as:

* Navigating through many of the foundational pages and clicking on key areas.
* Verifying the correct layout and accuracy of all visual elements.
* Exercising key functionality such as signup and login forms, additions/subtraction to shopping carts, checkout, return to shopping, and file exports.

Thus all our test cases in selenium would come under smoke testing as we have tested basic functionalities of the website: login forms, logout functions, as well as additions to the websites database

Test Cases covered under Fuzzing:

Bugs:

Test cases 10,12,15 and 22