## DESCRIPTION:

Purpose of this test is to check user requirements for main search function of google page and its additional functions. Search results will be also tested. Main test setup is to validate above functions and layout. Searching and providing list of result is key function and will be treated as main test object.
Particular test technics can be adjusted if tester finds its better to cover test scenarios.

## TEST ENVIRONMENT AND SETTINGS:

Google browser: 78.0.3904.108 64bit
Python Language/Pycharm
OS: Windows / Linux / Other

Frameworks and libraries:
Beautifulsoup – for pulling data out of HTML and XML files.
Selenium webdriver – to locate webside elements (optional for tests functions)
Pytest – to prepare tests

Other usefull tools:
Robot framework

## 3. DEFINITIONS FROM THE TEST SET

### 3.1 Definitions from the test plan:

For every test case there should be definitions that are different for each test case like:

- Number of repertitions
- Used setting and advance options

## 4. TESTING INSTRUCTIONS

### 4.1 Preconditions:

Home page is successfully loaded.

### 4.2 Test steps:

**1. Module and simple integration tests of homepage and search results page:**

- Structure tests to check if code doesn't have not necessary lines.
- Verification of layout of result (graphics, texts, labels).
- Assertion of objects: presence, editability, click ability, ability to carry all characters.
- Negative assertion of objects: not presence, not editable, not clickable.
- Assertion of faults and information shown.
- Test of login function.
- Test if functions are working and are connected to buttons (objects):
    - voice recognition,
    - pop-up keyboard,
    - search (verification if clicking search button is moving us to search result page),
    - link connections (email, links in footer, links to next page in search result page),
    - verification if filters of search results are working (movie, maps, graph etc.),
    - localisation.

**2. System tests of search tool:**

- Validation of system and its elements in comparison to provided use cases and documentation.
- Structure testing to check decision and instruction coverage path.
- Validation faults and information.
- Non functional test like:
  - Performance, stress, load, usability tests.
  - Maintainability and portability tests.

- **Test of equivalence partition for example:**
  - ">>>>" (query that not giving any results)
  - "aspirin" (test for present advertising)
  - "cat" (very popular)
  - "taylor swift" (popular)
  - "dezoksyrybonuklein" (very rare)

- **Functional tests during system tests:**
  - Searching by keyword that are mentioned in text.
  - Pagination correctness.
  - New or upgraded element should be possible to find and correct displayed (indexing).
  - It is not possible to find removed contents.
  - In search result there should be present contents linked to phrase.
  - The most important result are top.
  - Test of tips in search field:
    - Correctness and amount of tips.
    - Removal after providing next character.
    - Possibility to edit list of tips.
  - Searching of special characters / one character / long stings.
  - Layout of search result (main title, searching phrase in text fragment, edit date, page address, time of reaction, information about adequacy, other).
  - Advanced options and settings.

**3. User acceptance tests:**

Test done by key users of application at client side. Client testing application and providing defects to correct. Defects should be provided with proposed template to simplify communication.

**4. Preparing of regression tests:**

Selecting of key areas and processes that in case of generated faults there are the biggest loss. And selecting areas and processes that generate the highest amount of defects. Those areas should be checked in every update or new software release during regression tests.