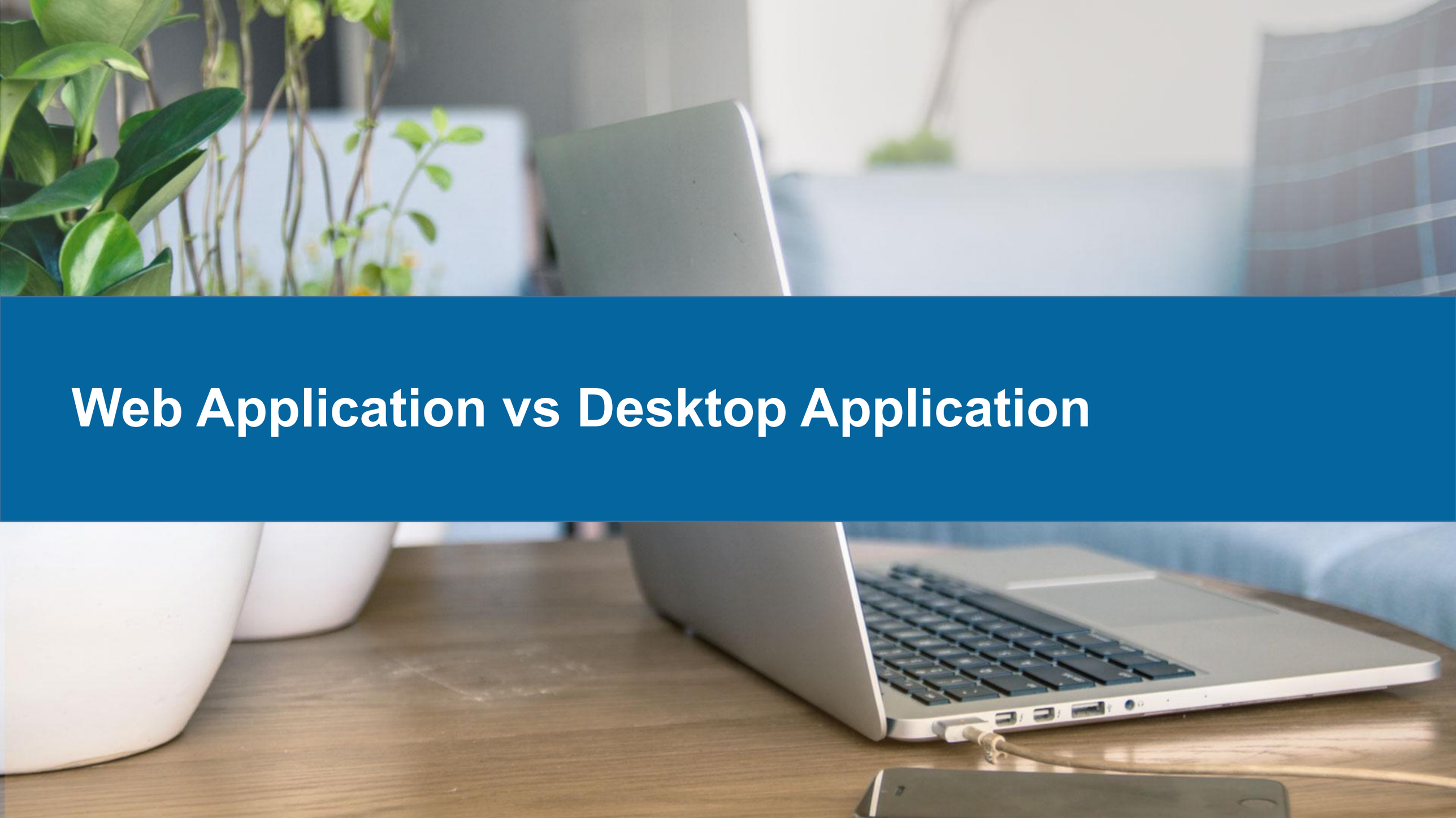
Web Application Testing

Coherent Solutions

Better Teams. Better Results.

Summary: General Overview

- Web and desktop applications comparison
- Web application benefits and challenges
- Focus of testing applications:
  - Functional tests
  - Non-functional tests
  - Points of tester’s Interest
- Helpful links and materials



Web Application vs Desktop Application

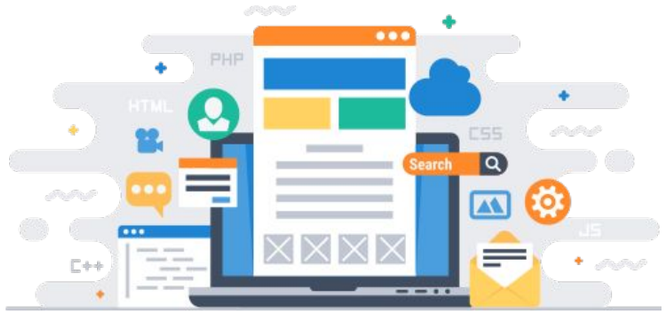
Web application vs Desktop application

WEB APPLICATION

is a client–server computer program which the client (including the user interface and client-side logic) runs in a web browser.

DESKTOP (Native) APPLICATION

is an application that runs standalone in a desktop or laptop computer.

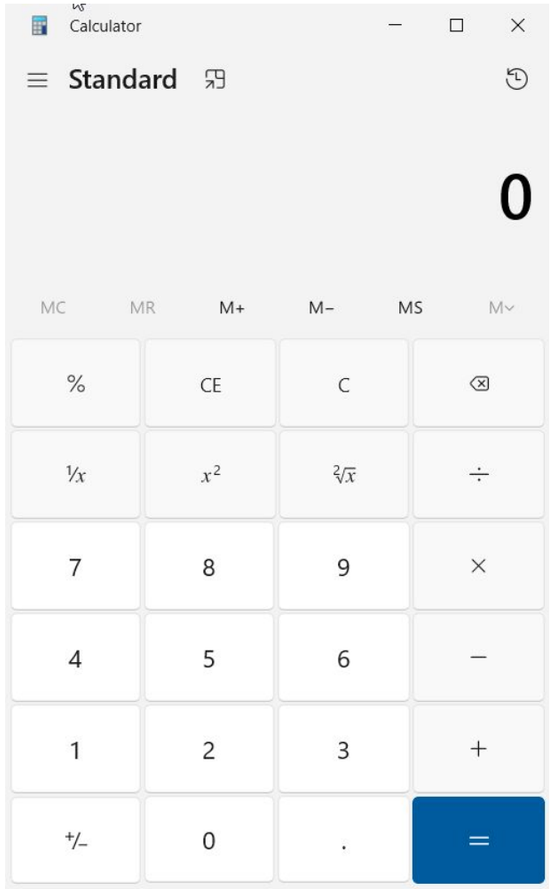


Desktop and Web Applications Comparison

	Desktop	Web-application
Internet access	Is not required	Is required
Installation/update	Must be installed. Update via special files or reinstallation	One time configuration for all the users. Instant update.
Interface	Standard	Diverse
Compatibility	Platform dependent	Browser dependent
Animation/graphics	Quick response	Relatively slow (connected with the data transmission)
Sharing	Only via inner net	The majority of the applications are configured to the shared access

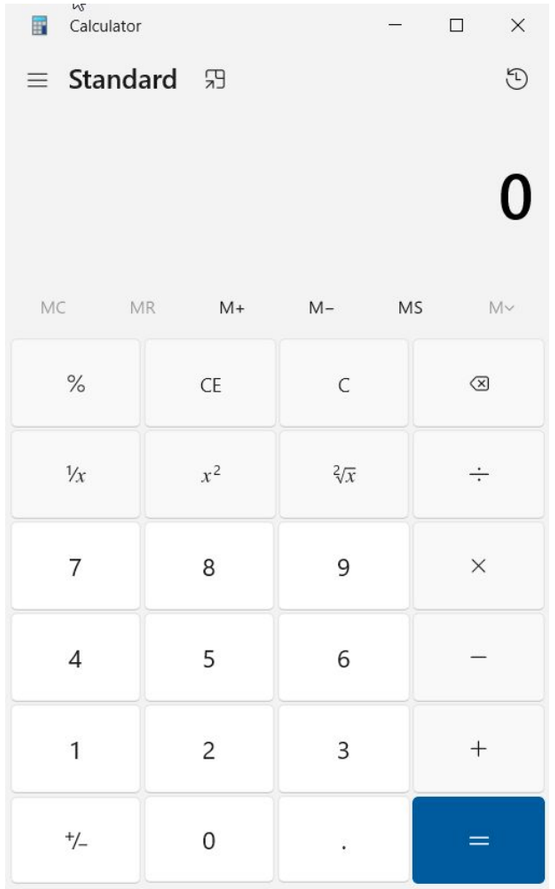
Examples

?

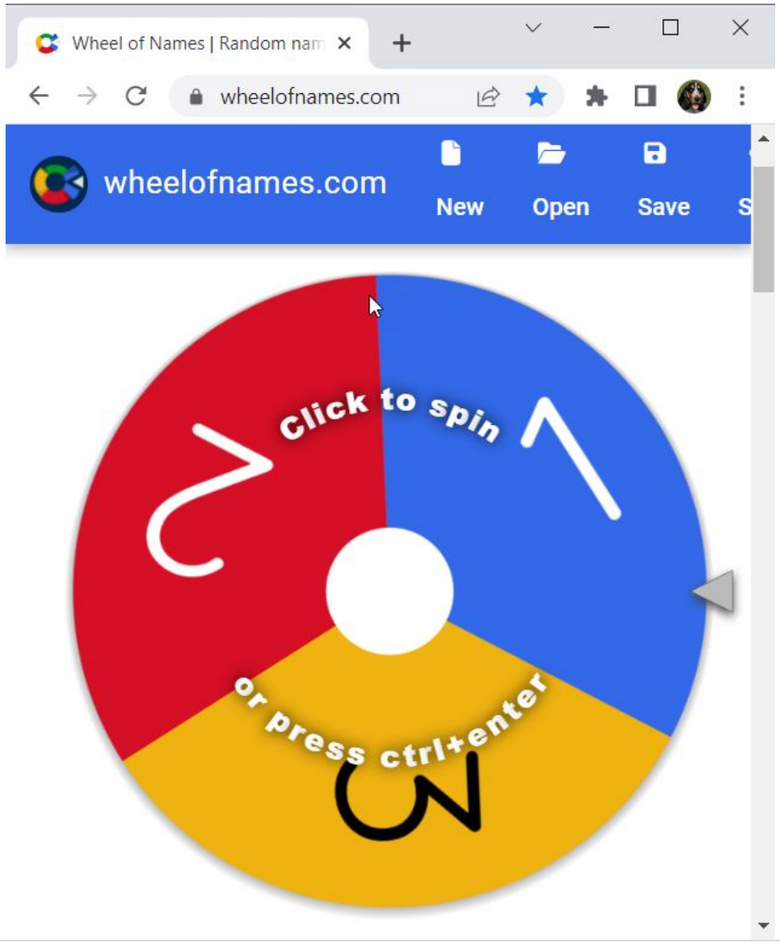


Examples

Native app

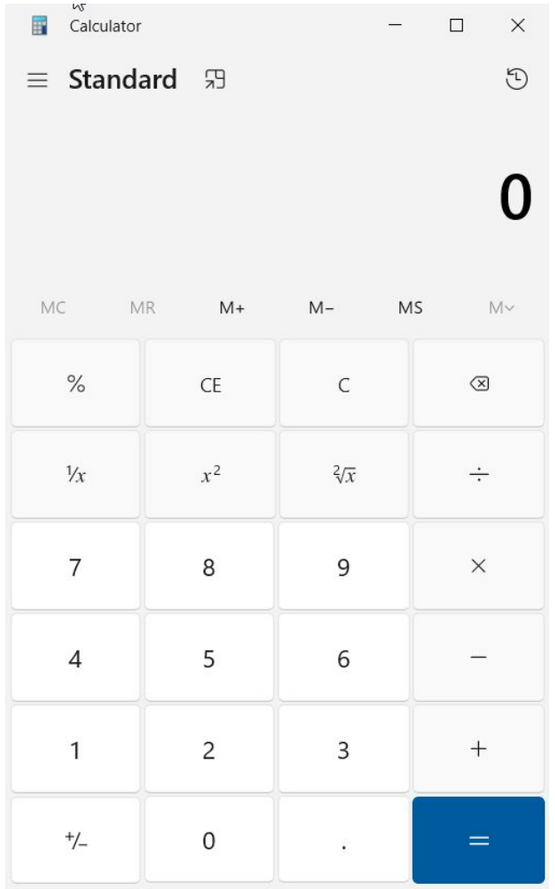


?

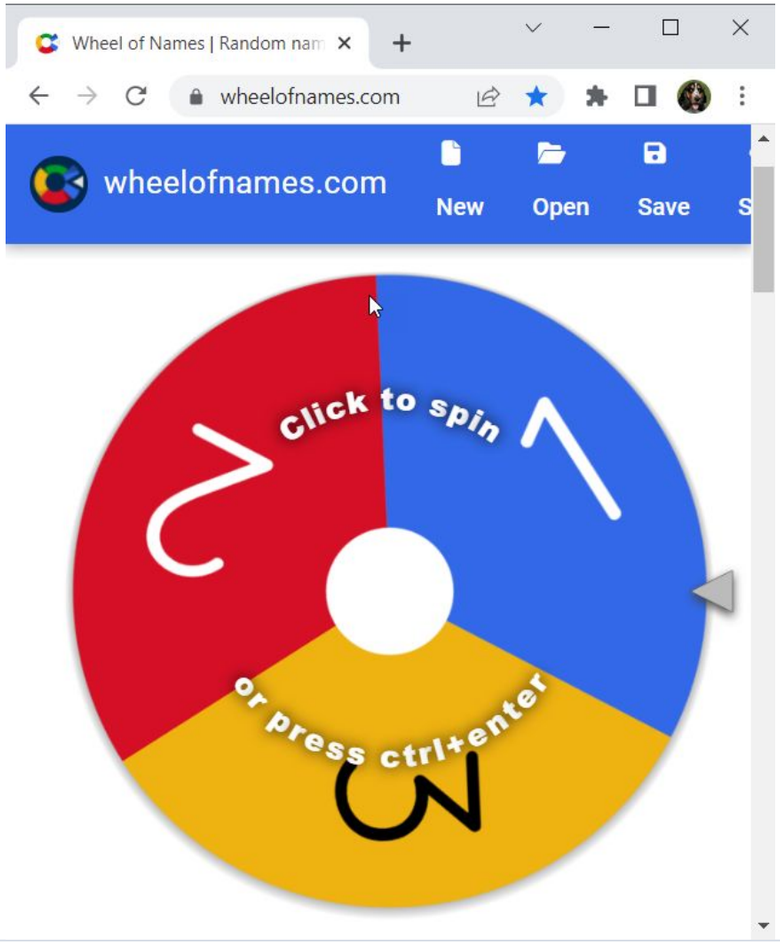


Examples

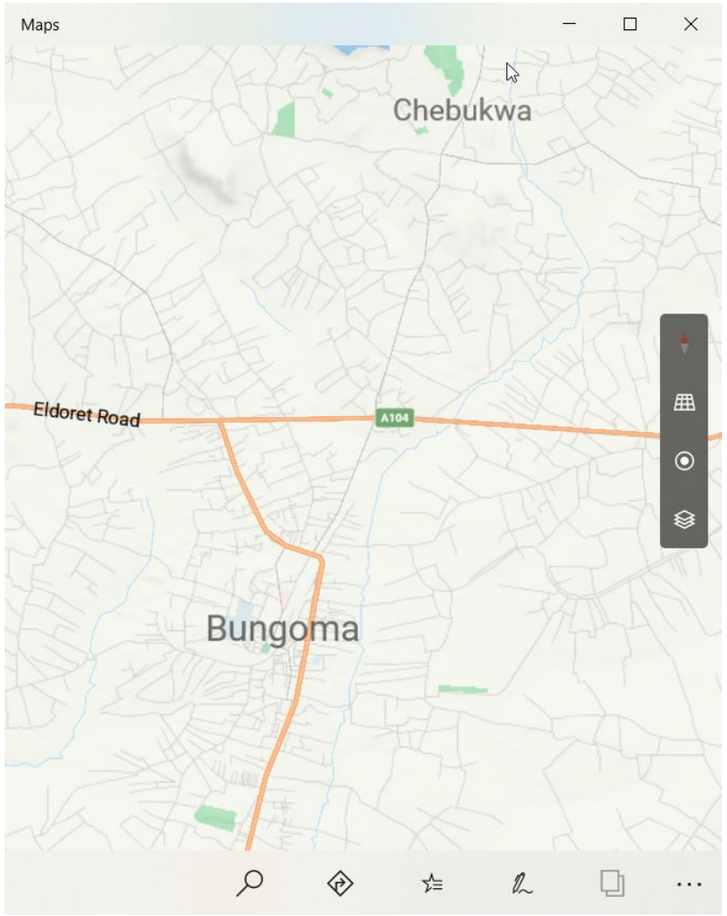
Native app



Web app



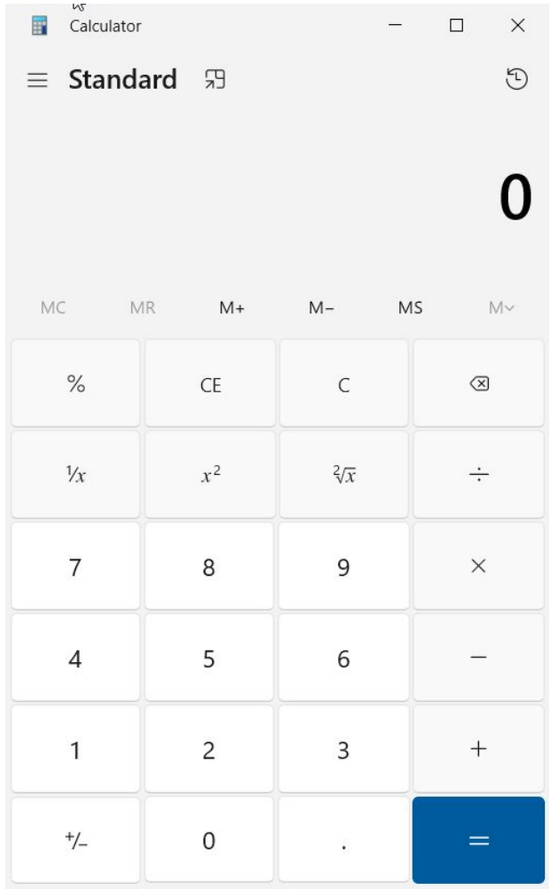
?



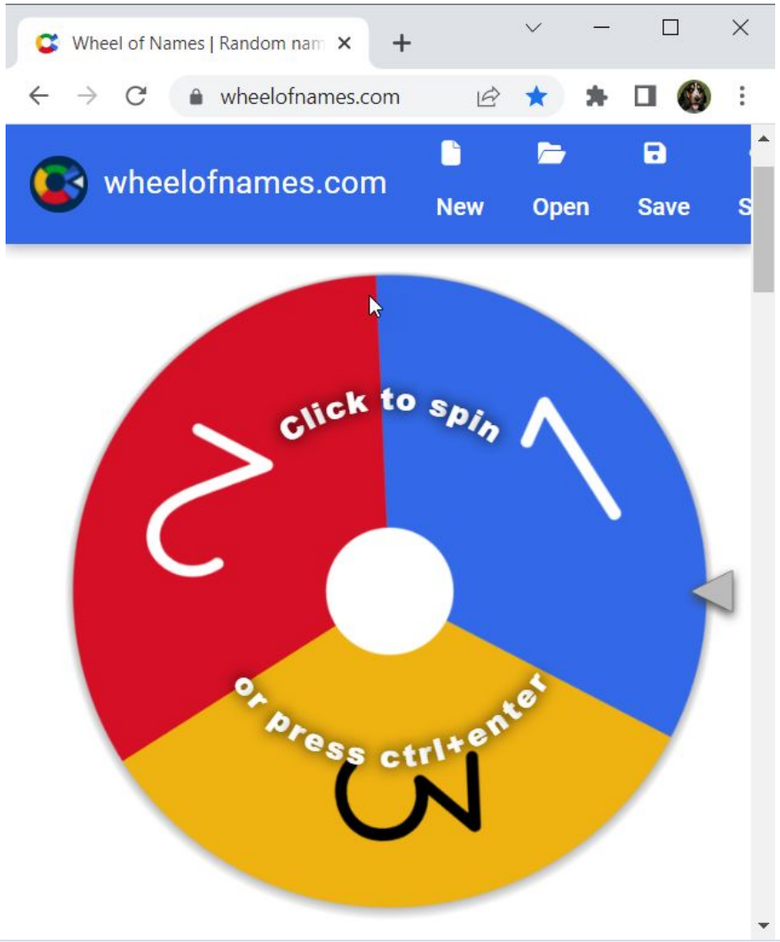
COHERENT SOLUTIONS

Examples

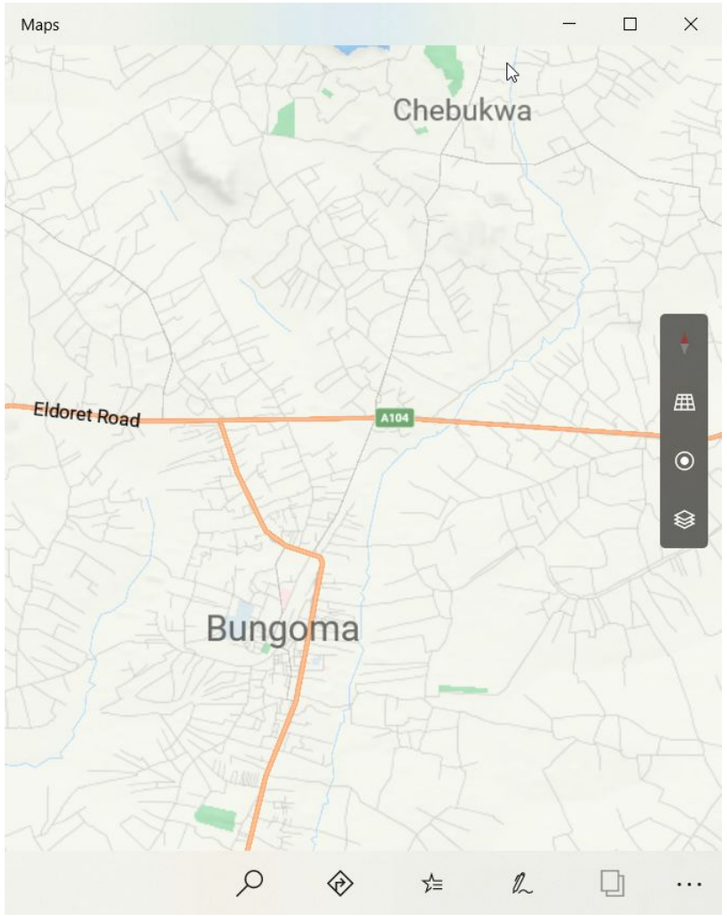
Native app



Web app



Hybrid app



Web vs Desktop Pros and Cons

Web Application Benefits & Challenges

Advantages

- Accessible anywhere (where internet is available)
- Less updating issues, always is up-to-date
- Platform happy
- Possibility of user tracking
- Easier issues' tracking
- Flexible core technologies
- Short releases cycles
- 24-hour availability of the website

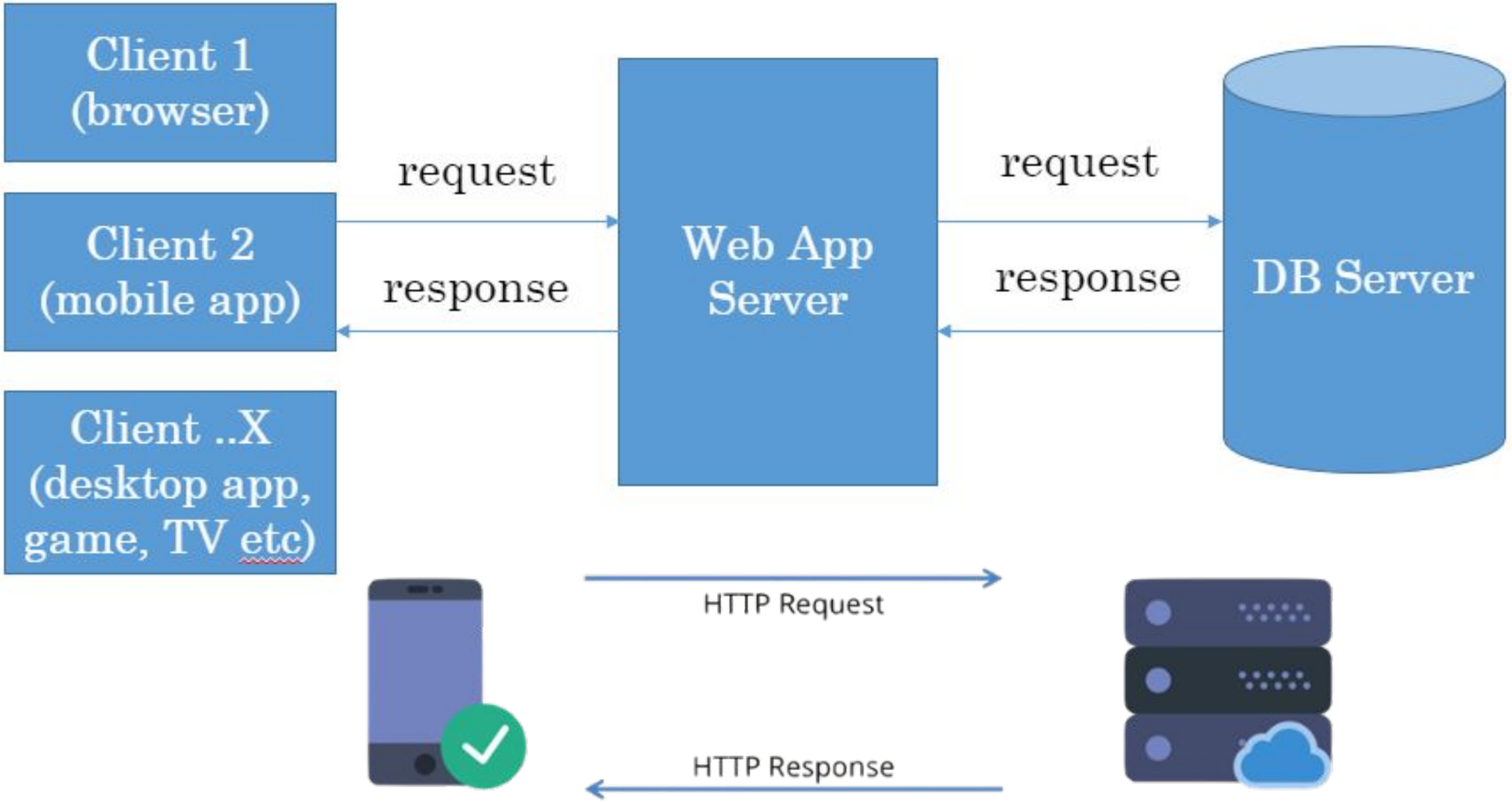
Disadvantages

- Internet Reliance – the app is accessible only if there is internet connection
- Less secure than desktop ones (risks of data breach, data leakage and so on)
- Speed of work is reduced
- Need of browser support
- Possible huge number of users during initial website launch
- Inability to control the user's running environment

Typical Web Application Flow

- **User** triggers a request to the **web server** over the **Internet**, either through a web browser or the application's user interface
- **Web server** forwards this request to the appropriate **web application server**
- **Web application server** performs the requested task- such as querying the **database** or processing the data – then generates the results of the requested data
- **Web application server** sends the results to the **web server** with the requested information or processed data
- **Web server** responds back to the client with the requested information that then appears on the user's display

Client-Server Model



Web applications testing focus

Web Applications Testing Focus

The initials tests for a website should focus on the site's intended behavior by assessing the following points:

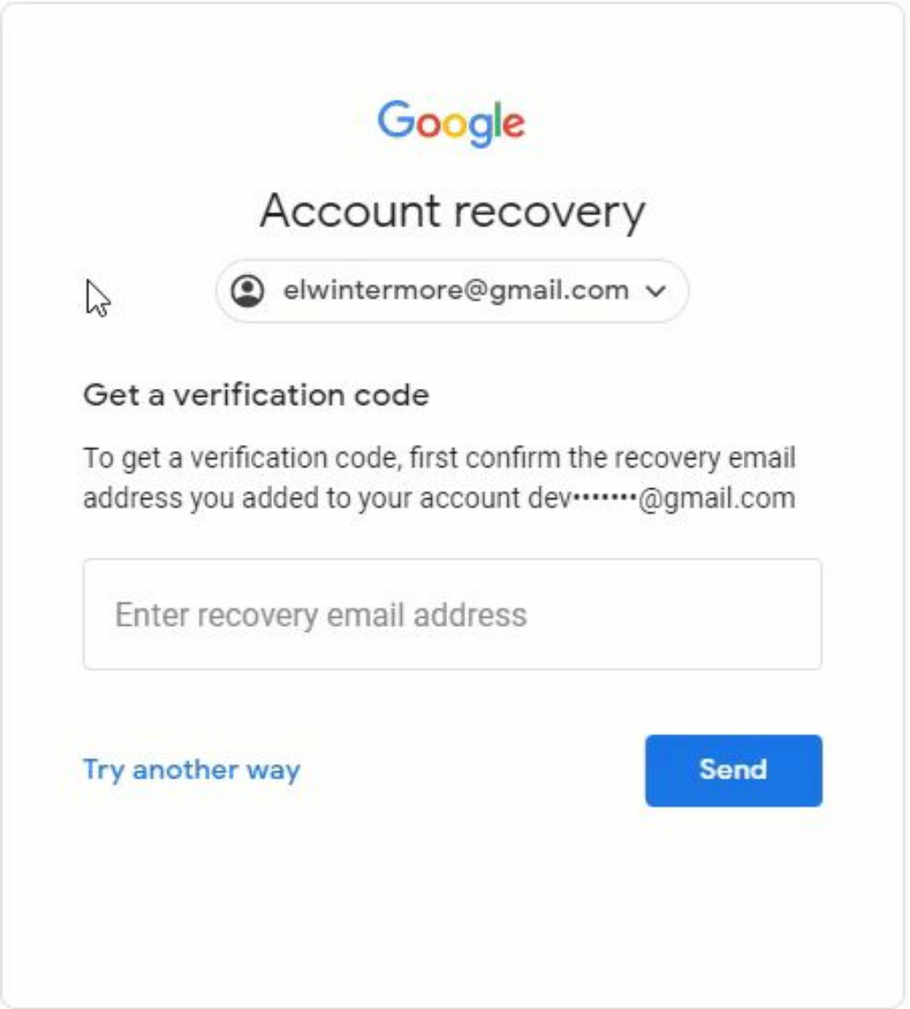
- Functional requirements
- Non-Functional requirements:
  - Usability
  - User Interface
  - Accessibility
  - Security
  - Compatibility
  - Performance and Availability

Functional Testing

Functional Testing evaluation of the functions the system performs, ‘what’ it should do from the standpoint of its behavior.

e.g. End-to-end reset password flow: user can reset the password and log to the system with a new one.

Note! End-to-End Transition Testing is the apex.



Functional Testing: Forms

Forms focus testing can be divided into the major aspects by action:

- **MODIFY**
  To New/Duplicate

- **CREATE**
  To New/Duplicate

- **DELETE**
  Delete selected
  Delete all
  Cascade deletion
  Deletion per permissions

Functional Testing: Forms

Forms focus testing can be divided into the major aspects by action:

- **MODIFY**
  To New/Duplicate

- **CREATE**
  To New/Duplicate

- **DELETE**
  Delete selected
  Delete all
  Cascade deletion
  Deletion per permissions

Functional Testing: Forms

- **VIEW**
  - After Creation
  - After Update
  - After Deletion

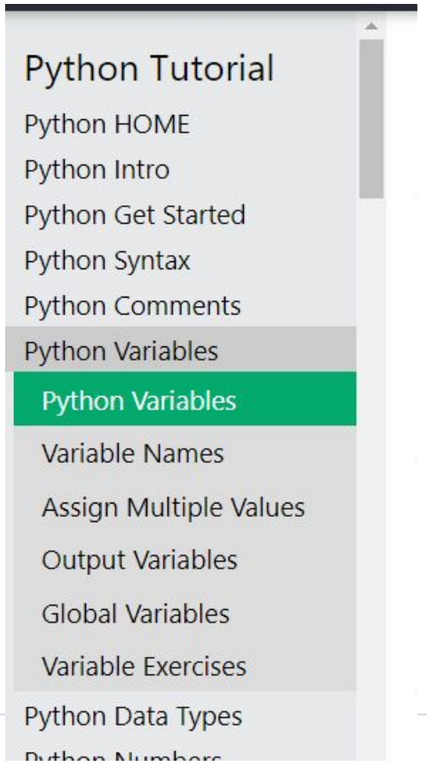
- **FILTER**
  - Search by each field
  - Search by all fields
- **DATE**
  - Date format test (>< current, incorrect format, empty fields)
  - From field/date picker

Functional Testing: Navigation

- Good navigation is an essential part of a website, especially those that are complex and provide a lot of information.

Most users expect the following:

1. Easy and quick access to the information they want
2. Logical hierarchy of pages
3. Confirmation of where they are at any point
4. Facility to return to previous states or the homepage



Navigation Menu

Functional Testing: Database

The key point is that not all data is shown on UI, this may happen when

Feature is concerned with DB specifically:

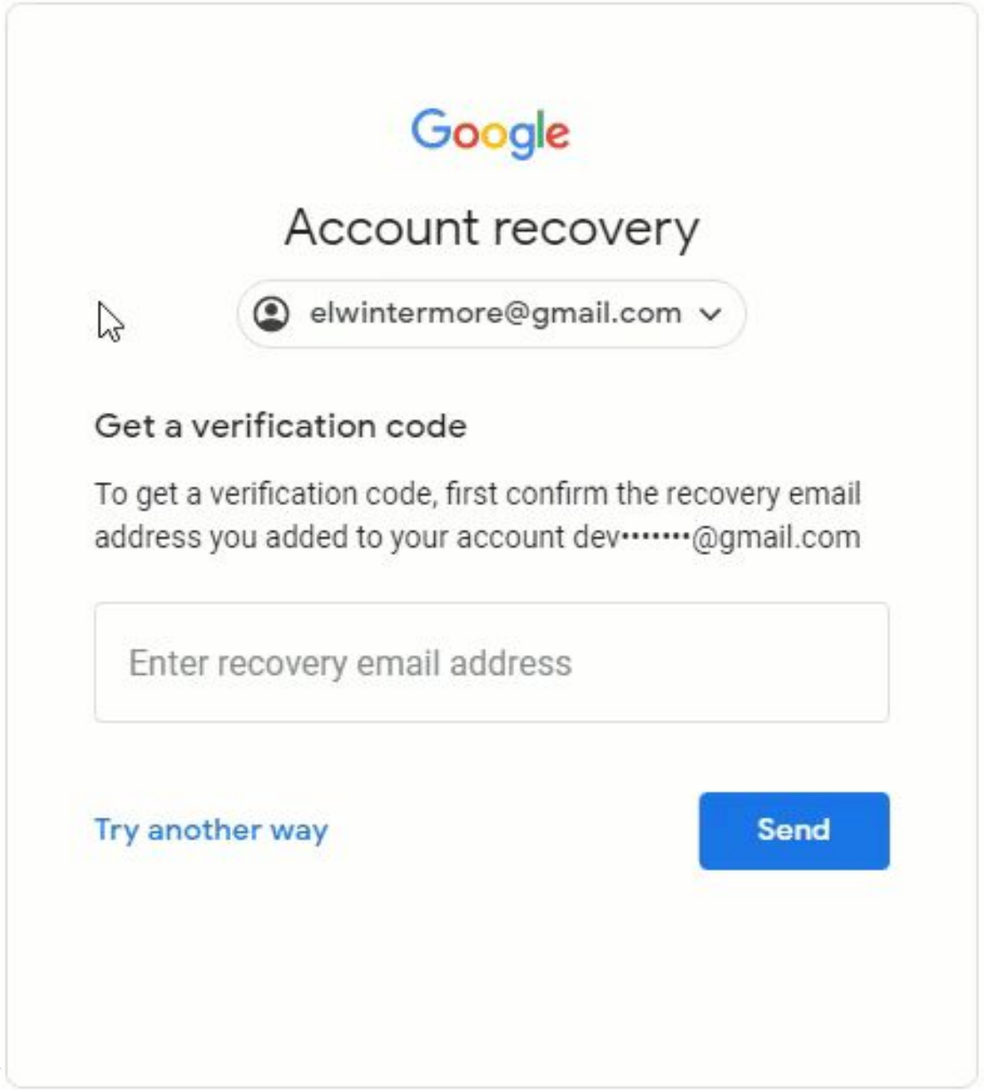
- StoredProcedures
- Columns and their values (e.g. data type, field length)
- NULL or empty?
- Duplicate records

Non-Functional Testing

Non-functional testing – evaluation of system characteristics as e.g. usability, performance, compatibility and security, ‘how well’ the system behaves.

e.g. **Accessibility** testing of how controls work in case only keyboard navigation is performed.

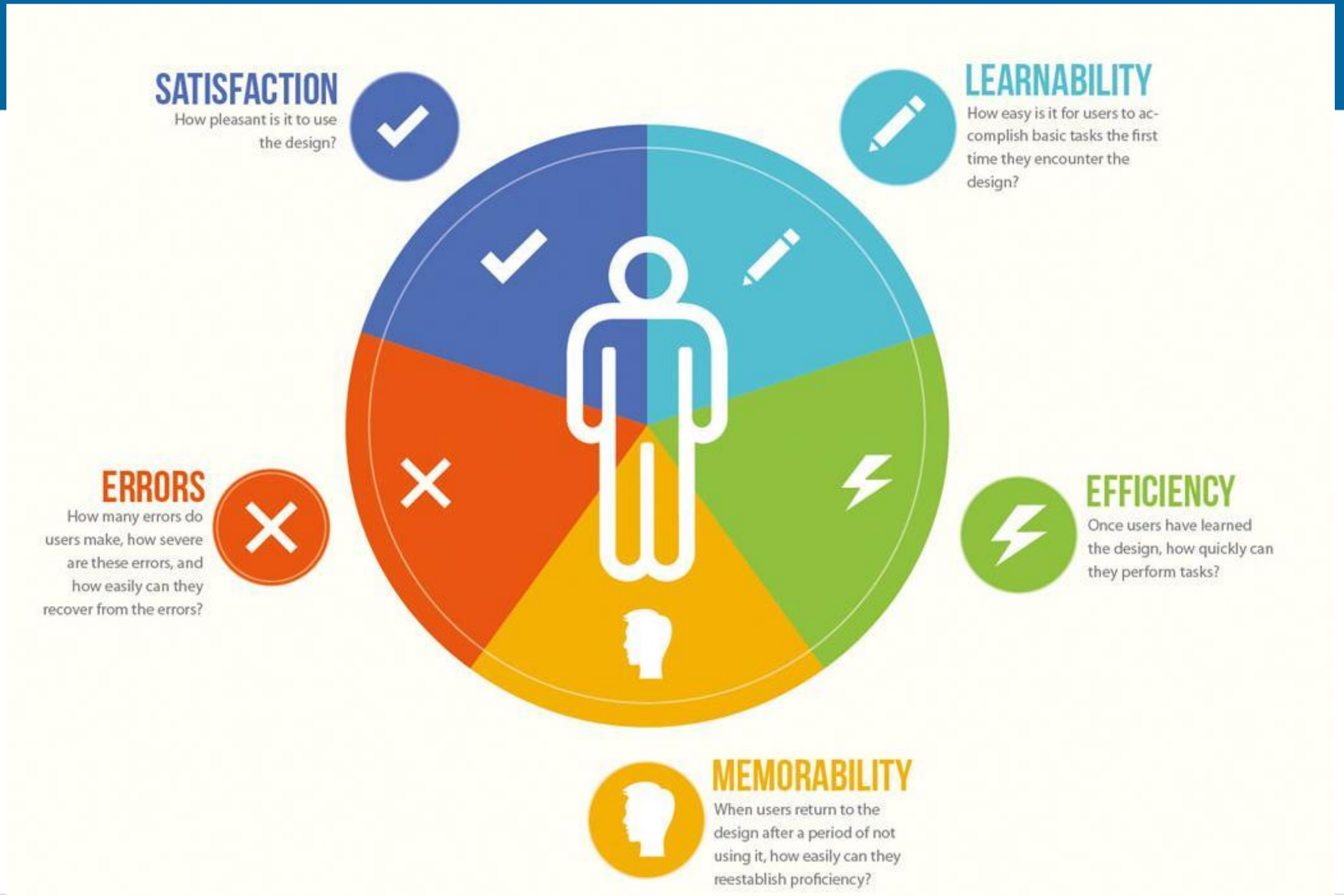
Security – reset password link is a one-time token.



Non-Functional Testing: Usability

Usability testing assesses the website's user friendliness and suitability by gathering information about how users interact with the site.

The key to usability testing is to study what a user **actually does.**


SATISFACTION
How pleasant is it to use the design?
LEARNABILITY
How easy is it for users to accomplish basic tasks the first time they encounter the design?
ERRORS
How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
EFFICIENCY
Once users have learned the design, how quickly can they perform tasks?
MEMORABILITY
When users return to the design after a period of not using it, how easily can they reestablish proficiency?

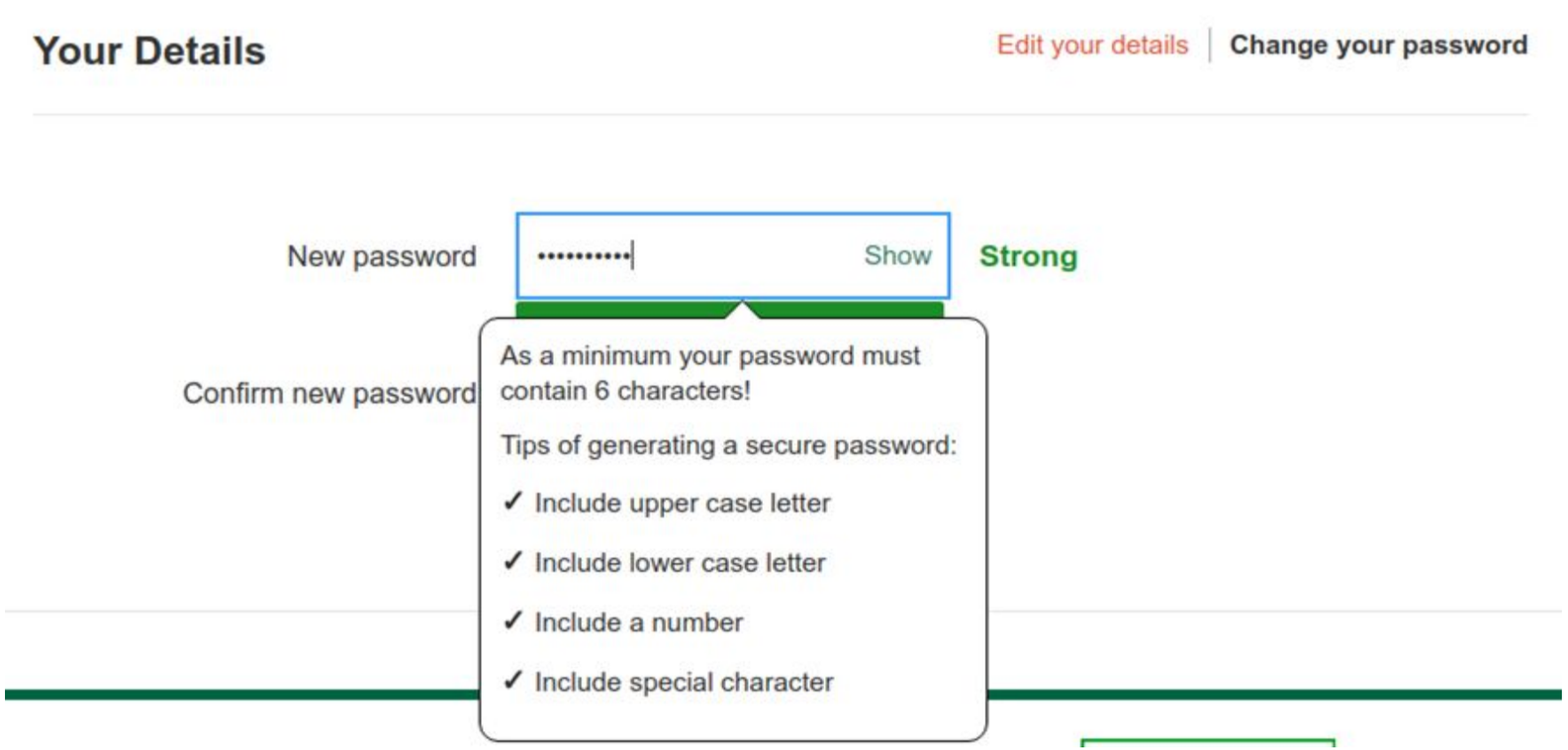
Points of Interest

General UI testing of both desktop and web applications can be based on Jakob Nielsen's 10 heuristics:

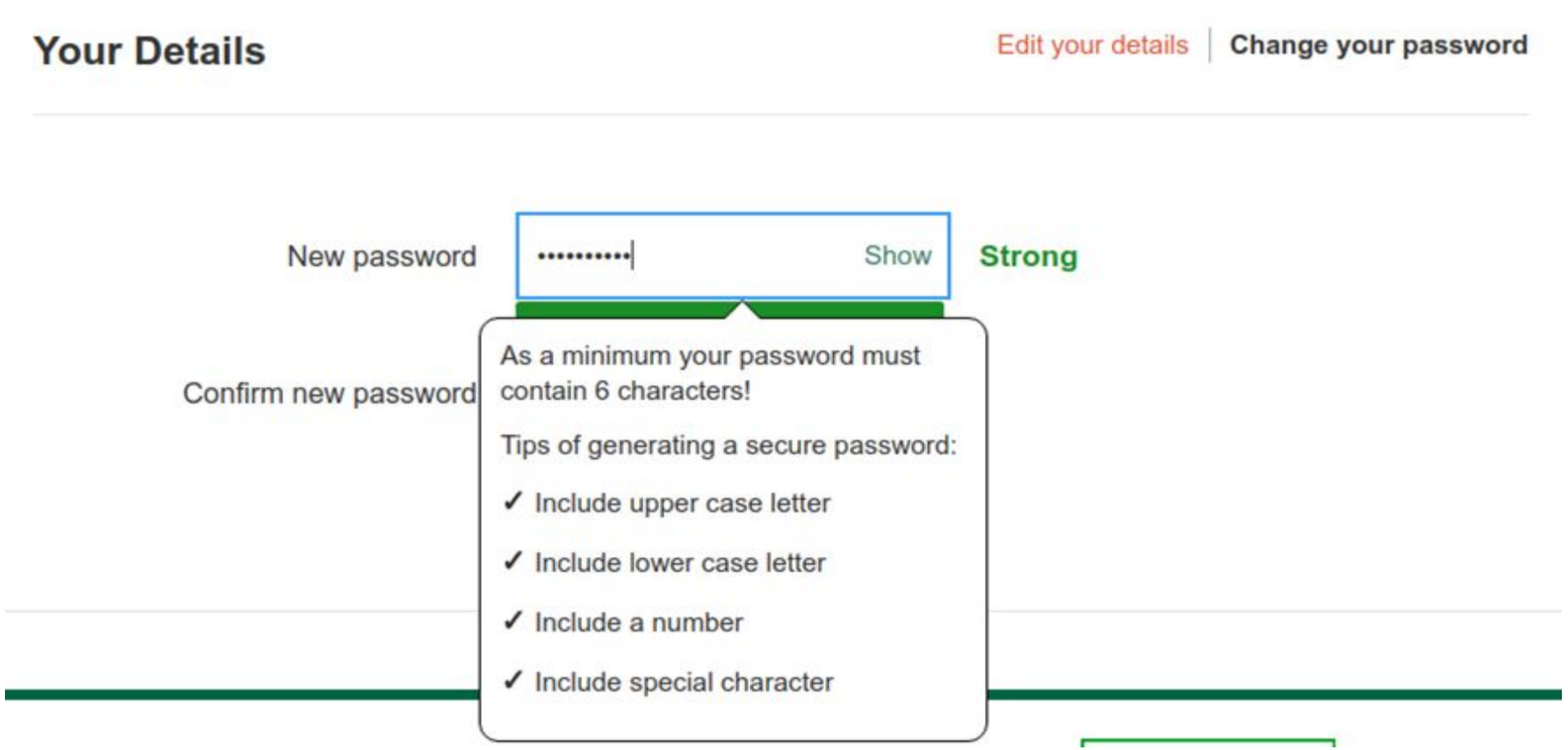
- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

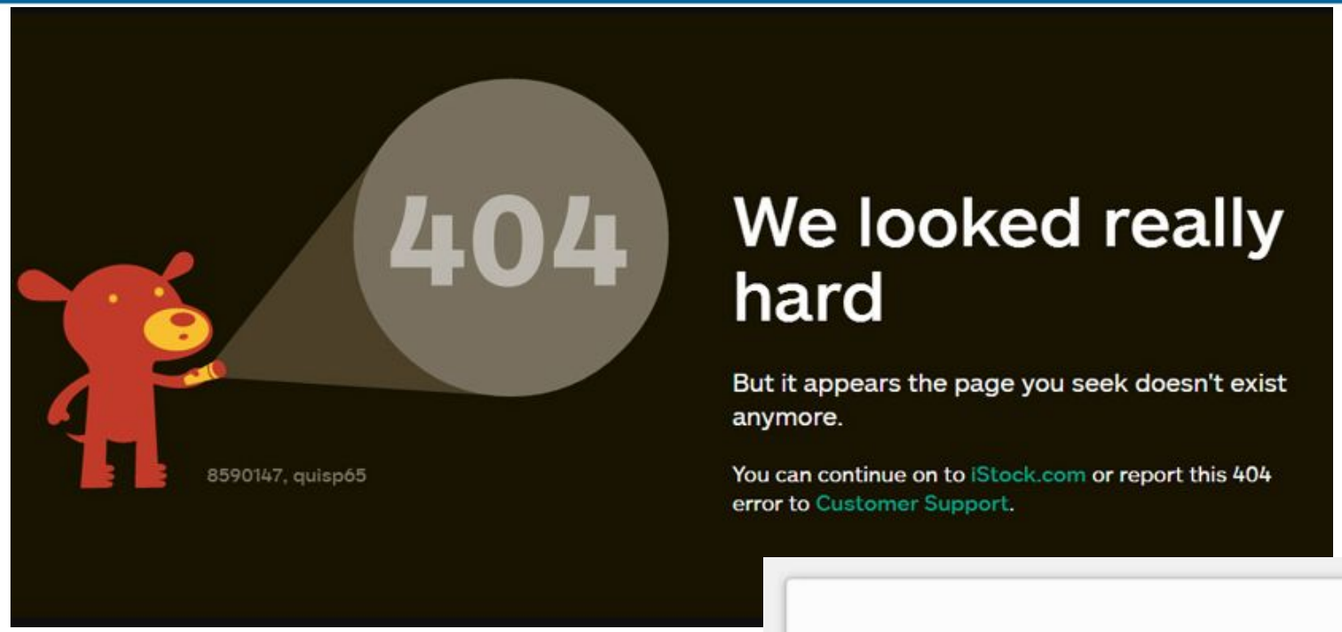
- Visibility of system status
- **Match between system and the real world**
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

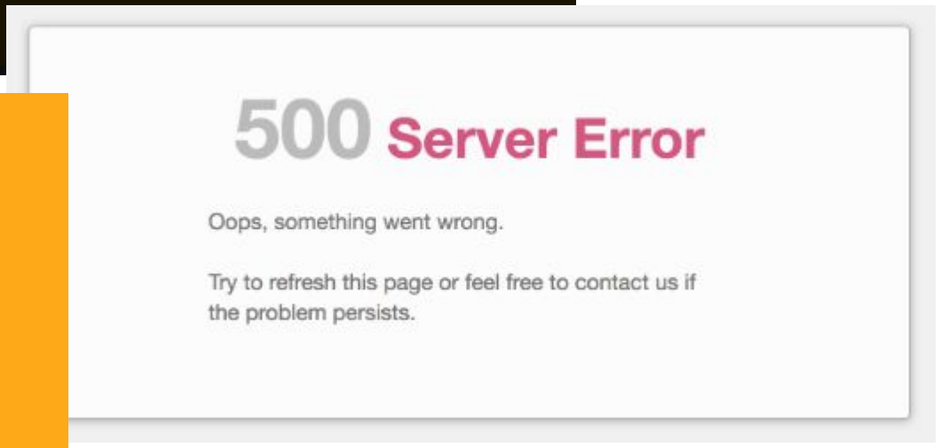


- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

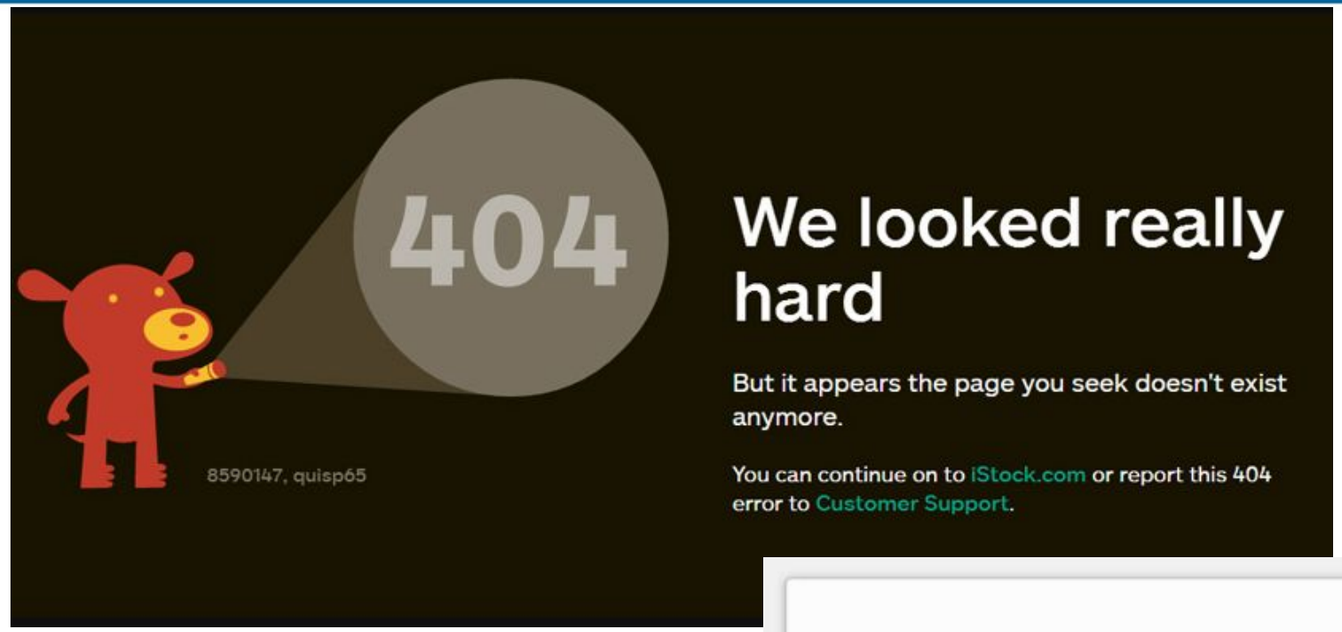


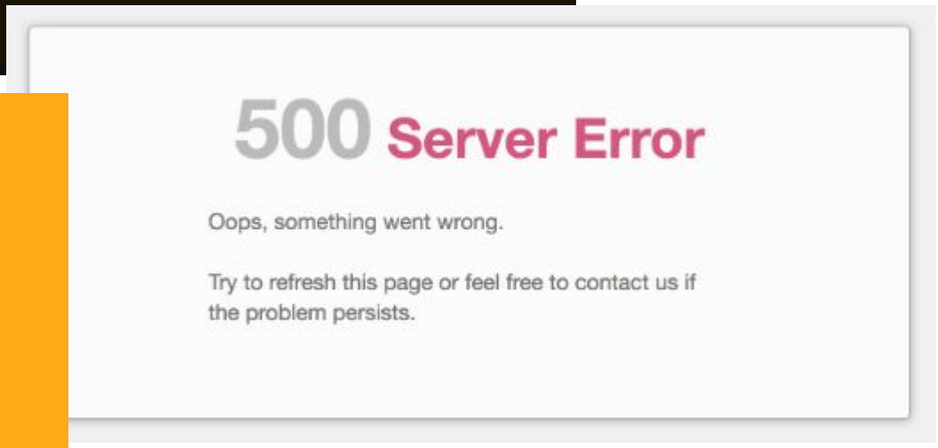
- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- **Error prevention**
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation



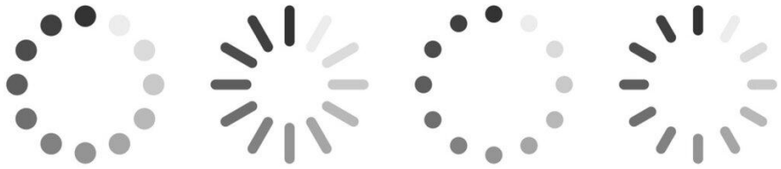


- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation



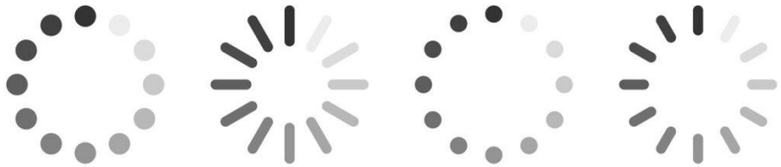


- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- **Help users recognize, diagnose, and recover from errors**
- Help and documentation



- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

Visibility of system status

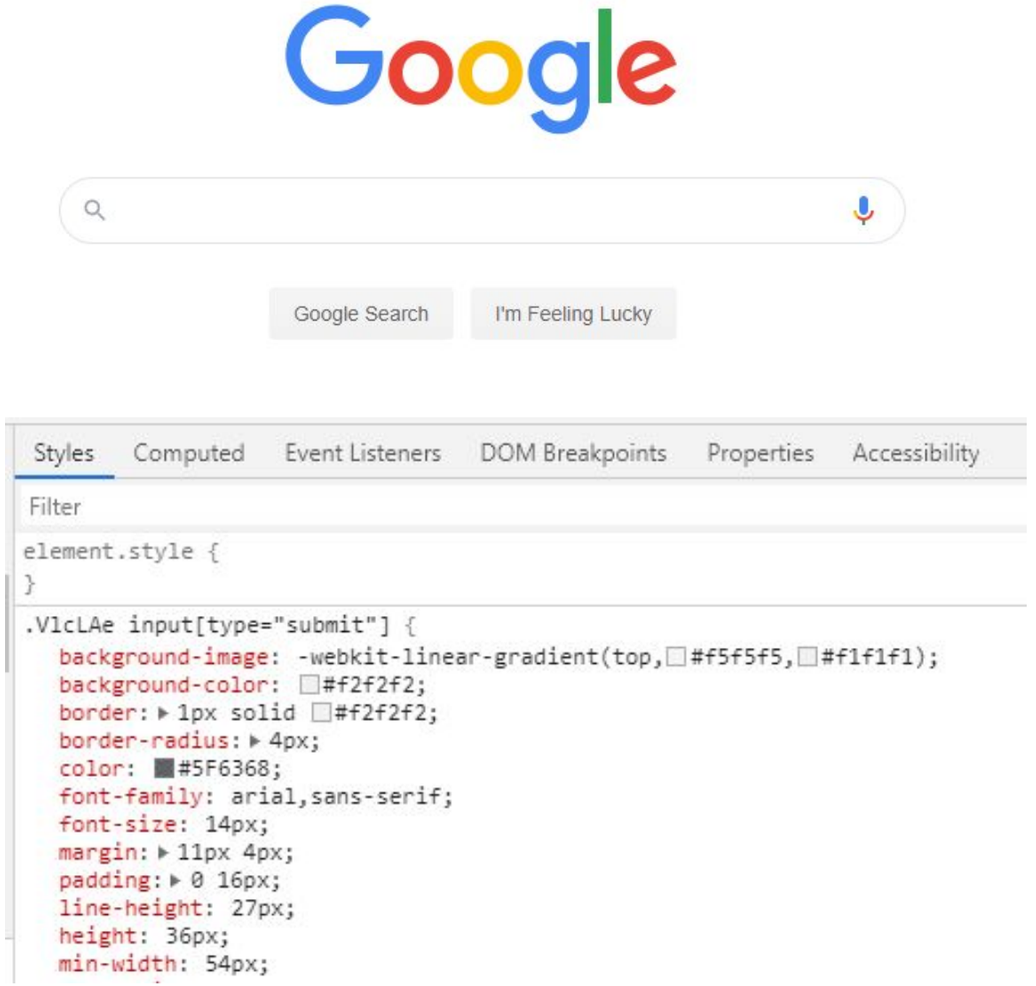


- **Visibility of system status**
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

Non-Functional Testing: User Interface Testing

Main parameters:

- Colors
- Fonts
- Sizes and locations
- Spelling
- Link validation
- Consistency



Non-Functional Testing: GUI - Content

Each web page must be tested for correct content from the user perspective.

These tests fall into two categories:

- Ensuring that each component functions correctly
- Ensuring that the content of each is correct, so check for:
  - Lorem ipsum fillers
  - Test Data
  - Spelling mistakes
  - Grammar mistakes
  - Punctuation

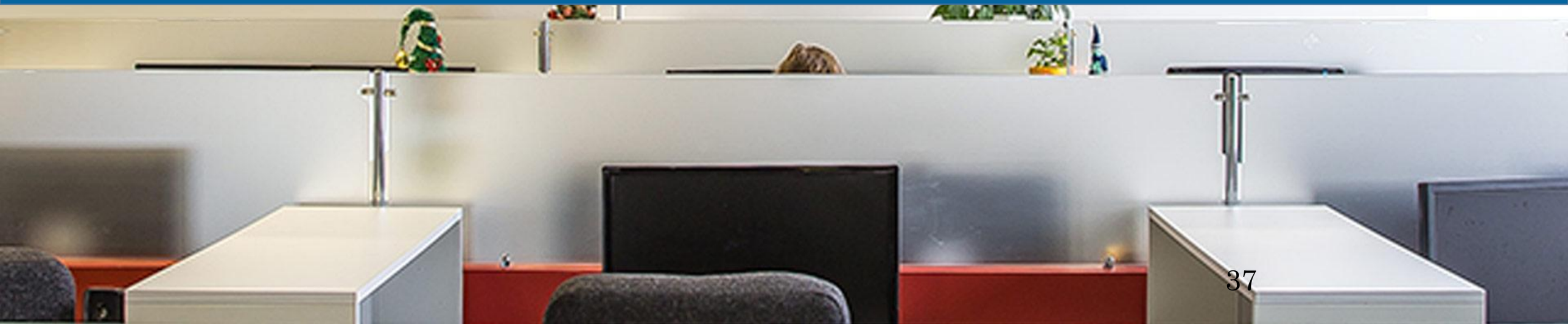
Non-Functional Testing: Compatibility Testing

- Browser Compatibility Testing (Native Browsers, Chrome, Firefox etc.)
- Browser Versioning Testing
- Testing Environments by OS (OS: Android, iOS; Windows, MacOS etc.)
- Testing Environments by Devices (Samsung, iPhone, Huawei etc.)

How to define?

- Requirements
- Customer needs
- Google Analytics
- Experience-based
- Budget/Resources

Testing Tools



Note!

- Ensure that the software you are going to install on your machines to perform the project-related activities is compliant with the **security standards/policy** used on your project.

Colors Identification

Desktop

Pixie – Free to use Color Picker

Web

Eyedropper (ColorZilla) – Free add-on for **Firefox**

PixelZoomer - Free add-on for **Firefox**

Font identification

Desktop

Font Matching Tool - A paid program. Free trial version is available, but with a limited set of functions.

Web

Font Finder - Free add-on for **Firefox**

WhatFont - Free add-on for **Chrome**

The Quick Brown Fox Jumps Over The Lazy Dog

Sizes and Locations

Desktop

Ruler Screen Calipers - A paid program. Free trial version is available, but with a limited set of functions.

Solid Capture - Paid

Web

Ruler MeasureIt - Free add-on for Firefox

Ruler PixelZoomer - Free add-on for Firefox

Spelling check

Desktop

Grammarly - desktop version

Web

Grammarly – **Chrome** add-on

IS SOFT

Links validation for Web Applications

Link Analyzer – Firefox add-on

Broken Link Checker – Firefox add-on

Check My Links – Chrome add-on

COHERENT
SOLUTIONS