

# Уеб тестване

Тестване на уеб приложения

## Уеб приложения

- ♦ Повечето модерни приложения са уеб базирани.
- ♦ Това са клиент-сървър приложения, при които клиента (браузъра) изпълнява част от кода (фронт-енда), а сървъра изпълнява кода на бек-енда (сървърна логика, конфигурации, бази данни и др.)



# Трудности при Уеб тестване

- ♦ Тестването на уеб приложение е по-трудно, от тестването на десктоп приложение (целия код се рънва на една машина):
  - ♦ Тайминг проблеми.
  - ♦ Различия между браузърите.
  - ♦ Мрежови проблеми.

# Трудности при Уеб тестване

- ♦ Новите технологии които обогатяват възможностите на уеб приложенията, създават и много трудности за QA:
  - ♦ Асинхронни процеси.
  - ♦ Клиентският код се изпълнява директно в браузъра.
  - ♦ Анимации и др.



# JavaScript

- ♦ Преди появата на JavaScript, уеб приложенията бяха изцяло сървър-базирани
  - ♦ Сървърът изпраща цяла статична страница, всеки път щом се натисне бутон по UI.
- ♦ JavaScript позволява да се променят елементи без да се рефрешва цялата страница.

# AJAX

- ♦ Asynchronous JavaScript And XML:
  - ♦ JavaScript може да прави заявки до сървъра и да ъпдейтва, само определени елементи от сайта.
  - ♦ AJAX е асинхронен – не ъпдейтва цялата страница, всеки път когато има промяна.
  - ♦ От гледна точка на QA – AJAX може да има известно забавяне, докато ъпдейтва данни от сървъра.



# Adobe Flash

- ♦ Adobe Flash – е браузър плъгин, който позволява рънването на специален код написан за флаш.
- ♦ Използва се за създаването на мултимедия – изображения, звук, видео, игри, мобилни и десктоп приложения.





# Adobe Flash

- ♦ Adobe Flash – е остаряла технология, но все още е широко използвана.
- ♦ Модерна алтернатива на флаш е HTML 5 (заедно с CSS и JavaScript).
- ♦ Недостатъци на Flash – не може да се маркира текст, не може да се скролва, няма опциите при десен клик, които имат HTML страниците (inspect, view source, save as, copy и др.)



## Инструменти за Уеб тестване

- ♦ Static Analysis – за валидиране на кода - <https://validator.w3.org>
- ♦ Link checker – проверява за неработещи линкове, редиректи и др. - <https://validator.w3.org/checklink>





zenmate

## Браузърни инструменти

- ♦ Firebug – плъг-ин за Firefox, позволяващ по-внимателни разглеждане на кода, конзола, мрежови данни и др.
- ♦ VPN клиенти – за промяна на IP (симулиране на потребители от други страни и мрежи):
  - ♦ Hola – разширение за Chrome.
  - ♦ Zenmate – разширение за Chrome и Firefox.



**hola!**

## Функционално уеб тестване

Всички линкове трябва да се проверят:

- ♦ Всички външни линкове, от всички страници на определен тестван домейн.
- ♦ Всички вътрешни линкове.
- ♦ Линкове за изпращане на и-мейли.
- ♦ Линкове за под страници.
- ♦ Проверете за счупени линкове, и линкове водещи до зацикляне (redirect loop).

## Функционално уеб тестване

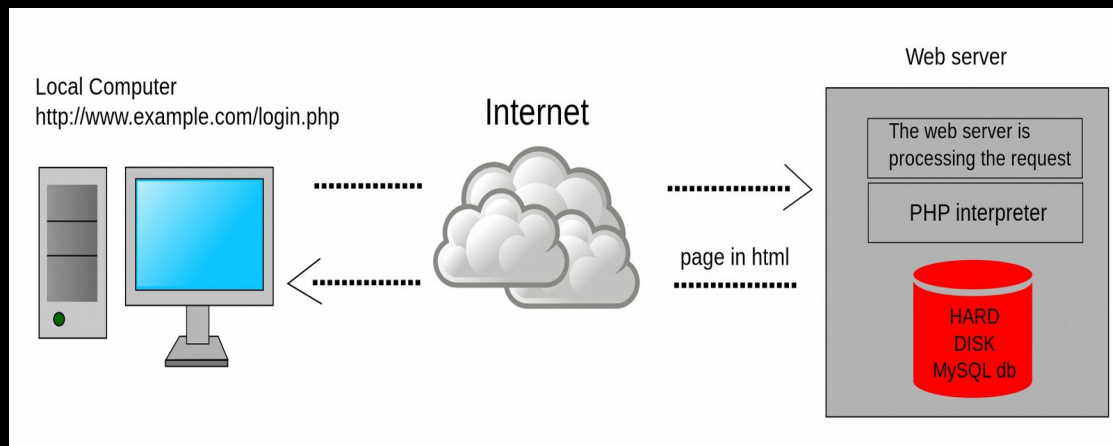
Всички форми (login, register, contact, comment и др) трябва да се проврят:

- ♦ Проверете валидациите на полетата.
- ♦ Проверете стойностите по подразбиране.
- ♦ Проверете опциите за добавяне, редактиране и изтриване на форми, и техните данни.

# Функционално уеб тестване

Тестване на базата данни:

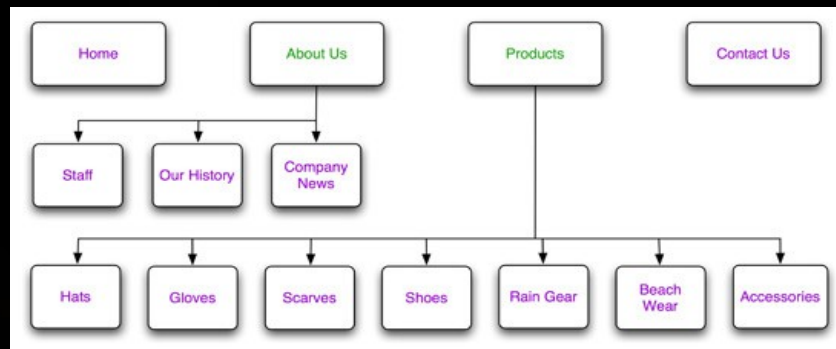
- ♦ Проверете дали правилните данни се записват в правилните таблици в базата данни.
- ♦ Проверете дали данните въведени в базата, се визуализират успешно в уеб приложението.



# Нефункционално уеб тестване

## Usability testing:

- ♦ Проверете навигацията на сайта:
- ♦ Проверете как потребителя може да “сърфира” страниците, различните контроли, като бутони, диалогови кутийки, линкове и др.
- ♦ Сайтът трябва да е интуитивен и лесен за използване.





# Нефункционално уеб тестване

## Usability testing:

- ♦ Проверете съдържанието на сайта:
  - ♦ Съдържанието трябва да е логично, лесно за четене и разбиране.
- ♦ Проверете за правописни грешки.
- ♦ Картинките и видеата трябва да са с подходящите размери за страницата.

# Тестване на съвместимостта

Compatibility testing:

- ♦ Верифициране на сайта под различни браузъри и операционни системи, в зависимост от клиентските изисквания.
- ♦ Верифициране на мобилни версии.



# Тестване на съвместимостта

Важно за тестване под различни браузъри:

- ♦ Валидации на полетата.
- ♦ AJAX елементите.
- ♦ Размер и стил на шрифта.
- ♦ Верифициране на Layout – при различни резолюции на екрана.
- ♦ Формат на датите, специални символи и др.

# Тестване на сигурността

## Security testing:

- ♦ Трябва да се провери че не опълномощен потребител, не може да достъпва лични или конфиденциални данни.
- ♦ Ограничените файлове, не трябва да бъдат публично достъпни.
- ♦ Проверете дали сесиите се прекратяват автоматично от сървъра (обикновено това е клиентско изискване).

# Тестване на производителността

Performance, Load & Stress testing:

- ♦ Трябва да се верифицира времето за отговор на сървъра.
- ♦ Трябва да се верифицира времето за отговори, при увеличено натоварване на сървъра.
- ♦ Трябва да се провери до колко може да се натовари сървъра.

# Тестване на производителността

Тестване на уеб приложението през различни нетуорк мрежи:

- ♦ За да проверим дали потребители от различни краища на света могат да достъпват приложението.
- ♦ Производителността на приложението е тясно свързано със скоростта на мрежата.
- ♦ Много често има мрежи от които достъпът до нашето приложение е забранен.



# Въпроси



## Задачи за домашна работа

1. Проверете QA test сайта за счупени линкове: <http://qa.soft-intellect.com/test>
2. Изтествайте съвместимостта на новия дизайн на QA test сайта, с Firefox, Opera, Safari и Internet Explorer 11. Може да използвате изгледа на Chrome за сравнение (там е изтествано и е ок). Логвайте бъговете в РМ системта.