Analisi

Il cliente chiede un’applicazione che disegna delle forme colorate

I colori (verde, rosso, blu) cambiano quando l’utente preme un pulsante

Le forme sono quadrate

Devono essere supportati tutti i principali tipi di dispositivo (desktop, tablet, mobile)

App per uso interno all’azienda

L’applicazione non richiede dati agli utenti

I pulsanti devono iportare il nome del colore che assegnano alle forme

Le periferiche aziendali sono continuamente aggiornate (max 3-5 anni di vita)

Il software deve essere mantenuto per tutta la durata del contratto

Specifiche funzionali

L’applicazione deve mostrare un’interfaccia costituita da un insieme di quadrati

L’applicazione deve essere utilizzata tramite browser (chrome 109 o successivi, firefox 109 o successivi)

L’interfaccia deve mostrare 3 pulsanti, ogni pulsante deve riportare scritto il nome di un colore (rosso, verde, blu)

Quando l’utente preme uno dei pulsanti, tutti i quadrati devono diventare del colore indicato sul pulsante

Quanti quadrati? – chiedere al cliente

Di quale dimensione devono essere i pulsantiti? – chiedere al cliente

Di quale dimensione devono essere i quadrati? – chiedere al cliente

Come sono disposti i quadrati? – chiedere al cliente

Come sono disposti i pulsanti? – chiedere al cliente

Di quale colore sono i quadrati all’avvio dell’applicazione? – chiedere al cliente

Specifiche funzionali rev 1

L'applicazione deve mostrare un'interfaccia costituita da 15 quadrati che abbiano 200px di lato, i quadrati devono essere disposte in righe e colonne

All'avvio dell'applicazione i quadrati devono essere bianchi, l’applicazione deve avere sfondo bianco

L'applicazione deve essere utilizzata tramite browser (Chrome 109 o successivi, FireFox 109 o successivi)

L'intefaccia deve mostrare tre pulsanti, ogni pulsante deve riportare scritto il nome di un colore (Rosso, Verde, Blu), la dimensione di ogni pulsante deve consentire di contenere la scritta su una sola riga

I pulsanti devono essere posizionati nella prima riga dell'interfaccia

Quando l'utente preme uno dei pulsanti, tutti i quadrati devono diventare del colore indicato sul pulsante

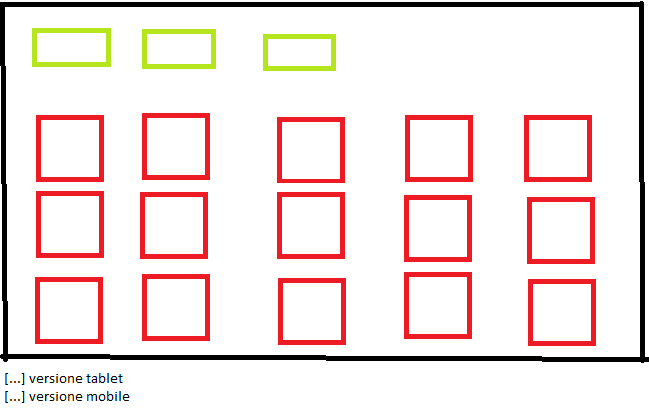
Mockup

Versione desktop

blu

verde

rosso



Progettazione

L’applicazione è web based e sarà sviluppata utilizzando php, html, css, javascript

* Html, definisce la struttura della pagina e i suoi contenuti
* Css, definisce l’aspetto estetico della pagina
* Php, genera il codice html
* Javascript, consente di interagire con l’utente, modificando la pagina in funzione del comportamento dell’utente (interazione lato client)

Php

La pagina index.php deve costruire un’interfaccia grafica come nel morkup allegato

La pagina html è costituita da 2 div

* Header, che contiene 3 pulsanti con testo “rosso”, “verde”, “blu” e id “btnrosso”,”btnverde”, “btnblu”
* Content, che contiene 15 div con classe “bgbianco”, “quadrato”

La pagina deve agganciare il file css basic.css

La pagina deve agganciare il file javascript automate.js

Css

Il File basic.css deve contenere le seguenti regole

Per gli elementi con classe “quadrato”: larghezza 200px, altezza 200px, margine destro 15px

Per gli elementi con classe “bgbianco”: colore di sfondo bianco

Per gli elementi con classe “bgverde”: colore di sfondo verde

Per gli elementi con classe “bgrosso”: colore di sfondo rosso

Per gli elementi con classe “bgblu”: colore di sfondo blu

Per l’elemento con tag body: colore di sfondo bianco

Javascript

Il file automate.js deve definire i seguenti comportamenti:

Quando l'utente clicca sull'elemento che ha ID "btnVerde", modificare tutti gli elementi con classe "quadrato" rimuovendo la classe "bgBianco" e aggiungendo la classe "bgVerde"

Quando l'utente clicca sull'elemento che ha ID "btnRosso", modificare tutti gli elementi con classe "quadrato" rimuovendo la classe "bgBianco" e aggiungendo la classe "bgRosso"

Quando l'utente clicca sull'elemento che ha ID "btnBlu", modificare tutti gli elementi con classe "quadrato" rimuovendo la classe "bgBianco" e aggiungendo la classe "bgBlu"

Javascript – definizione

Javascript è un linguaggio lato client che manipola HTML, aggiungendo, togliendo o modificando “pezzi” di HTML

Una tecnica molto semplice ed efficace è modificare le classi dei TAG HTML, in modo da attivare o disattivare le regole previste dal CSS

Javascript si basa sul concetto di DOM, partendo dalla struttura gerarchica del codice HTML, il linguaggio individua gli elementi che deve manipolare

Javascript può reagire agli eventi generati dall’utente (click, scroll, …), per questo motivo esistono i gestori di evento

Un gestore di evento è una funzione che viene chiamata dal browser quando si verifica l’evento richiesto

Javascript è un linguaggio ad oggetti, quindi si basa su proprietà e metodi

In javascript il DOM è contenuto nell’oggetto document

Per manipolare il documento HTML è sufficiente interagire con il document

Per raggiungere un elemento all’interno del DOM, il document fornisce diversi metodi

I metodi getElement permettono di raggiungere gli elementi del DOM in base al tag, alla classe o all’ID:

* getElementById(<ID>) resituisce l’elemento con l’ID specificato
* getElementByClass(<Class>) resituisce gli elementi con la classe specificata