## **Riepilogo risorse utilizzate nel progetto “Sound Sanctuary”**

Il progetto *Sound Sanctuary* fa uso di diverse librerie JavaScript, fogli di stile personalizzati, immagini, e funzionalità backend sviluppate in PHP. Riporto di seguito una panoramica.

### **1. Librerie CSS e JavaScript**

#### **1.1 WaveSurfer.js**

Libreria JavaScript utilizzata per visualizzare e modificare graficamente la forma d’onda dei file audio. Permette operazioni come la selezione e la riproduzione.

Utilizzata in: *editor\_audio.php*

#### **1.2 Font e Stili Web**

I font principali utilizzati sono Segoe UI, Tahoma, Verdana, sans-serif.

Alcuni file includono anche Google Fonts o font di sistema standard

**2. File di Stile e Immagini**

#### **2.1 CSS Personalizzati presenti all’interno dei file PHP**

Gli stili grafici sono definiti principalmente:

* inline nei file PHP tramite il tag <style>
* nel file esterno stile.css, incluso in:  
  + *upload.php*
  + *archivio\_musicista.php*
  + e altri file con layout simili.

#### 

#### 

#### 

#### 

#### **2.2 Immagini**

Immagini relative agli strumenti musicali vintage sono conservate nella cartella /images/.  
Esempi di immagini utilizzate:

* *../images/gibson.jpeg*
* *../images/preamp.jpg*
* *../images/ludwig.jpg*
* *../images/akai.jpg*
* *../images/eko.jpg*
* *../images/sa.webp*
* *../images/fe.jpg*

### **3. Componenti Backend (PHP/MySQL)**

#### **3.1 Autenticazione e Gestione Sessioni**

L'autenticazione è gestita tramite auth.php e sessioni PHP.  
Sono presenti controlli di accesso basati sul ruolo (musicista o produttore) per limitare e differenziare le funzionalità disponibili.

#### **3.2 Upload e Gestione File**

I file audio caricati dagli utenti vengono salvati nella cartella: *../uploads/registrazioni/*

Funzionalità gestite nei file:

* *upload.php*
* *archivio\_produttore.php*
* *modifica\_registrazione.php*
* *editor\_audio.php*

#### 

#### **3.3 Monitoraggio Strumenti**

Ogni strumento viene monitorato per le ore totali di utilizzo.  
Dopo ogni 20 ore viene richiesta una revisione, che può essere segnata come effettuata.

Funzionalità implementata in:

* *manutenzione\_strumenti.php*

Supporto lato database tramite i campi:

* *revisione\_richiesta* (booleano)
* *ore\_ultima\_revisione* (intero delle ore)

### **4. Organizzazione dei File**

Il progetto è strutturato logicamente secondo il ruolo dell’utente:

* **Musicista:**
  + *archivio\_musicista.php*
  + *prenotazioni\_musicista.php*
  + gestione notifiche e registrazioni
* **Produttore:**
  + *archivio\_produttore.php*
  + *manutenzione\_strumenti.php*
  + *gestione\_registrazioni.php*

Inoltre, è presente un uso modulare della cartella *includes/*, per separare componenti riutilizzabili:

* *db.php* per la connessione al database
* *auth.php* per l'autenticazione e la gestione delle sessioni