Architettura del progetto

- php
- html
- js (vanilla, no framework)
- sql
- css
- bootstrap

Intestazione file

Ogni file deve iniziare con la seguente intestazione:

```
/**
 * @project       : Domuscan App
 * @filename      : [NOME_FILE]
 * @author        : [NOME_E_COGNOME] | <[EMAIL_ADDRESS]>
 * @description   : [DESCRIZIONE_DEL_FILE]
 * @creation date : [GG-MM-AAAA]
 * @last update   : [GG-MM-AAAA]
 *
 * @version: [VERSIONE]
 */
```

Il commento deve essere formattato in modo tale da poter essere letto e interpretato correttamente da IDE moderni (come VSCode, PhpStorm, ecc.) e da strumenti di documentazione automatica (come PHPDoc). L'intestazione deve essere sempre presente all'inizio di ogni file e deve essere aggiornata ogni volta che il file viene modificato. Ogni riga deve essere allineata verticalmente e deve essere preceduta da un asterisco (*).

Naming

Utilizzare la seguente struttura per i nomi degli elementi, tutto rigorosamente in italiano:

- file: snake_case
- cartelle: snake_case
- classi: PascalCase
- funzioni: camelCase
- variabili: snake_case
- costanti: UPPER_SNAKE_CASE

Made with Markdown Inline Preview

Commenti

Ogni funzione deve avere un commento prima del suo blocco, che descriva in italiano:

- Cosa fa la funzione
- Quali parametri accetta
- Quali valori restituisce
- Quali eccezioni può sollevare
- Altre info che si ritiene possano essere utili al lettore

Esempio:

```
/**
 * @description: [DESCRIZIONE_DELLA_FUNZIONE]
 * @param [TIPO] [NOME_PARAMETRO] Descrizione del parametro
 * @return [TIPO] Descrizione del valore restituito
 * @throws [ECCEZIONE] Descrizione dell'eccezione
 */
```

Il commento deve essere formattato in modo tale da poter essere letto e interpretato correttamente da IDE moderni (come VSCode, PhpStorm, ecc.) e da strumenti di documentazione automatica (come PHPDoc).

Altre regole

- Evitare la creazione di classi, limitandosi ad utilizzarle se richieste da librerie esterne
- Privilegiare l'approccio procedurale (funzioni) rispetto a quello orientato agli oggetti
- Preferire PHP puro rispetto all'uso di framework o librerie esterne, a meno che non sia strettamente necessario, e anche in quel caso chiedere conferma all'utente
- Utilizzare SQL standard per la gestione dei database
- Privilegiare librerie open source o con licenze una-tantum, e non librerie con licenze che richiedono il pagamento di royalties o che limiterebbero l'uso del software

Made with Markdown Inline Preview