Progettazione WEB app che gestisce:

* Area riservata del responsabile per gestire operazioni di inserimento, modifica, visualizzazione, ricerca ed elimina di flora e fauna relative al proprio parco inserito nel Database

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Indice**

1. **Analisi dei Requisiti**
   1. Requisiti funzionali
   2. Requisiti non funzionali
2. **Progettazione**
   1. Tecnologia utilizzata
   2. Struttura Database
   3. Struttura file system su Server
3. **Sviluppo e test**
   1. Struttura template dinamico
   2. Oggetto connessione al DB
   3. Metodi di selezione ed esecuzione query
      * 1. **Analisi dei Requisiti**

La gestione dei requisiti (acquisizione, analisi, negoziazione, specifica, validazione) è il primo passo del processo di sviluppo e una delle fasi più critiche dello sviluppo software, perché influenza in modo diretto il successo o il fallimento dei progetti.

* 1. **Requisiti funzionali**

I requisiti funzionali descrivono le funzionalità ed i servizi forniti dal sistema (cosa deve essere fatto). Nel ciclo di sviluppo software i requisiti funzionali rappresentano nel dettaglio le funzionalità richieste.

* + Un responsabile del parco può entrare nella propria area personalizzata con un utente e una password (precaricato sul DATABASE)
  + Un responsabile del parco può INSERIRE o MODIFICARE un animale o una pianta
  + Un responsabile del parco può ELIMINARE una pianta o un animale presente nel database.
  + Un responsabile del parco può RICERCARE una pianta o un animale per vedere le informazioni.
  1. **Requisiti non funzionali**

I requisiti non funzionali non sono collegati direttamente con le funzioni implementate dal sistema, ma piuttosto alle modalità operative, di gestione. Definiscono vincoli sullo sviluppo del sistema.

* + **Responsive web design**: le pagine web dovranno adattarsi in modo automatico alla grandezza dello schermo per pc, tablet, smartphone.
  + **Durata Sessione**: la sessione viene distrutta effettuando il logout
  + **Convalida dei dati lato client:** i dati richiesti e la tipologia di determinate informazioni (campi obbligatori, email) dovranno essere validati lato client utilizzando i controlli HTML5 (o Javascript)
  + **Sicurezza**
    - La comunicazione di determinati dati da client a server dovrà avvenire con metodologia sicura (POST)
    - Le password su DATABASE dovranno essere criptate
    - Bisognerà strutturare e controllare la gestione delle query al DB tramite *PREPARED STATEMENT per evitare attacchi SQL INJECTION.*
    - Gestione dei privilegi dell’utente di connessione (GRANT) in modo tale da disabilitare funzionalità non richieste (esempio: l’utente utilizzato dall’applicativo WEB non avrà i diritti di DROP nel DB – possibile attacco in SQL INJECTION)

1. **Progettazione**

In questa fase si definisce l'architettura software su cui si baserà la realizzazione del progetto. Ho scelto una strutturazione tipica su tre livelli logico-funzionali (applicazioni Three-Tier ) ma che possono essere distribuiti anche su più li velli (applicazioni Multi-Tier ) .

L'applicazione Web Three-Tier si sviluppa su 3 livelli

- **Presentazione (Client):** costituisce l'interfaccia utente dell'applicazione e corrisponde al browser Web. Le tecnologie scelte da utilizzare sono HTML5 e CSS

- **Applicazione (Server Web):** è il livello logico della web-app, la componente elaborativa dal lato server (Server-side) e utilizza la tecnologia PHP

- **Dati (Server DBMS):** consente di modellare e gestire il contenuto informativo dell'applicazione. La tecnologia usata a questo livello è il Database relazionale MySQL.

**2.1 Tecnologia utilizzata**

Saper programmare per il web significa conoscere i diversi meccanismi e strumenti per conservare o passare i dati, detti parametri, tra le diverse pagine dell'applicazione web. In pratica una Web-application, è un programma che non necessita di essere installato nel computer in quanto esso si rende disponibile su un server in rete e può essere fatto funzionare attraverso un normale Web browser (es. Google Chrome, Mozilla Firefox, Opera, ecc.) in posizione di client. Il concetto fondamentale è che, a differenza dei siti web statici (HTML), l'applicazione web viene realizzata con una o più tecnologie (PHP, Ajax, Database ecc.) che permettono la creazione di un sito dinamico, cioè di un sito nel quale il contenuto delle pagine varia durante l'interazione.

* + **Web Server APACHE**: Apache (o meglio Apache HTTP Server ) è una piattaforma server web modulare in grado di operare nei sistemi operativi più diusi. Le pagine .php contengono dei codici destinati a produrre dei comportamenti e a generare dinamicamente contenuti HTML, perchè ciò sia possibile è necessaria la mediazione di un web server. Un web server è un programma che si occupa di ascoltare un canale di comunicazione per intercettare una richiesta da servire. Il client, utilizzando un browser, invia un messaggio di richiesta HTTP , contenente la URL, attraverso il collegamento di rete al web server; questo, catturata la richiesta, risponde, sempre attraverso il protocollo HTTP , con una pagina HTML con il contenuto informativo desiderato dal client. Il web server Apache presenta un'architettura modulare, quindi ad ogni richiesta del client, vengono svolte funzioni specifiche da ogni modulo di cui è composto.
  + **MYSQL:** Innanzitutto una base di dati (database) è un insieme di dati logicamente correlati fra loro. I Data Base Management System (DBMS) sono prodotti software in grado di gestire database che hanno grandi quantità di dati, che condividono i dati fra più utenti e applicazioni e che utilizzano dei sistemi di protezione e autorizzazione per l'accesso ai dati. Esistono diversi tipi di DBMS: gerarchico, reticolare, relazionale, ad oggetti; il modello che più si adatta alle mie esigenze è il modello relazionale che organizza i dati in tabelle, basandosi sulle relazioni fra essi. MySQL è un sistema di gestione di basi di dati relazionali multi-piattaforma distribuito come software libero. Più precisamente MySQL è un RDBMS ("Relational DataBase Management System"), ossia un sistema di gestione per database relazionali, che si basa sul linguaggio SQL (Structured Query Languageè il linguaggio standard di interrogazione dei database). MySQL si occupa della strutturazione e della gestione a basso livello dei dati stessi, in modo da velocizzarne l'accesso, la modifica e l'inserimento di nuovi elementi.
  + **PHP**: (acronimo ricorsivo di Hypertext Prepocessor) è un linguaggio di scripting interpretato (non compilato) server-side, con licenza open source, utilizzato principalmente per sviluppare applicazioni Web. Il codice PHP viene usato per generare dinamicamente i documenti HTML che il client deve ricevere e visualizzare nel browser.
  + **Altri Software utilizzati:**
    - Editor: VisualStudioCode
    - Per la pubblicazione On-line prot. FTP: FileZilla

**2.2 Struttura Database**

L’analisi dei requisiti ha spostato il focus della progettazione in una parte ben delineata ed inserita in un progetto più ampio di una gestione di un parco regionale.

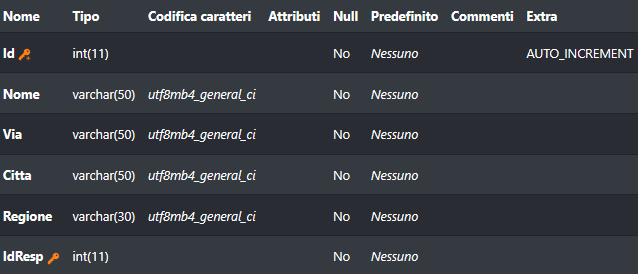
La progettazione porterà allo sviluppo delle seguenti tabelle

* **Parchi**
* **Responsabili**
* **Flora**

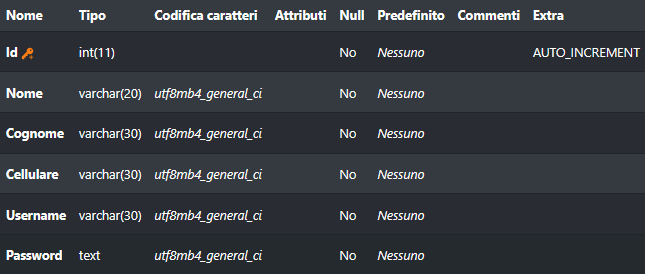
**FLORA**



**PARCHI**



**RESPONSABILI**



Le tabelle verranno create con i seguenti script SQL

**FLORA**

CREATE TABLE flora (

Id INT(11) NOT NULL AUTO\_INCREMENT,

Descrizione TEXT CHARACTER SET NOT NULL,

Categoria VARCHAR(50) CHARACTER NOT NULL,

Specie VARCHAR(50) CHARACTER NOT NULL,

Primavera TINYINT(1) NOT NULL,

Estate TINYINT(1) NOT NULL,

Autunno TINYINT(1) NOT NULL,

Inverno TINYINT(1) NOT NULL,

PRIMARY KEY (Id)

);

**PARCHI**

CREATE TABLE parchi (

Id INT(11) NOT NULL AUTO\_INCREMENT,

Nome VARCHAR(50) CHARACTER NOT NULL,

Via VARCHAR(50) CHARACTER NULL,

Citta VARCHAR(50) CHARACTER NOT NULL,

Regione VARCHAR(30) CHARACTER NOT NULL,

IdResp INT(11) NOT NULL,

PRIMARY KEY (Id),

INDEX (IdResp),

CONSTRAINT fk\_parchi\_responsabili FOREIGN KEY (IdResp) REFERENCES responsabili(Id)

ON DELETE RESTRICT

ON UPDATE CASCADE

);

**RESPONSABILI**

CREATE TABLE responsabili (

Id INT(11) NOT NULL AUTO\_INCREMENT,

Nome VARCHAR(20) CHARACTER NOT NULL,

Cognome VARCHAR(30) CHARACTER NOT NULL,

Cellulare VARCHAR(30) CHARACTER NOT NULL,

Username VARCHAR(30) CHARACTER NOT NULL,

Password TEXT CHARACTER NOT NULL,

PRIMARY KEY (Id)

);

**2.2 Struttura Filesystem su Server**

Strutturare in modo efficace ed ordinato le cartelle e i file che si andranno utilizzare è parte fondamentale di una buona progettazione. Il software dovrà essere progettato per garantire la caratteristica di buona manutenzione per un periodo futuro, l’organizzazione del codice legato alla parte grafica e di connessione al DB sono i punti principali:

|  |  |  |
| --- | --- | --- |
| progetto |  |  |
|  | * Conf | * Db\_config.php //classe per connessione e metodi selezione ed esecuzione query |
|  | * Templates | * Header.php //contiene header html e tasto logout * Header\_ris.php // contiene header per quando l’user è loggato * Navbar.php //gestisce in modo dinamico la visualizzazione delle opzioni per i diversi user * Section.php // contiene la parte principale del programma * Footer.php //chiusura del sito |
|  | * Php | * File .php con codice elaborazione dati |
|  | * index.php |  |
|  | * home.php |  |
|  | * cerca\_pianta.php * elimina\_pianta.php * visualizza\_piante.php |  |

Descrizione ed utilizzo dei file principali

**Connessione al DB**

La connessione al DB avverrà nella classe DBCONNECT contenuta nel file db\_config.php. All’interno ci saranno i metodi:

* \_\_construct: connessiona al DB
* db\_select\_single : passata una query di seleziona restituisce un array unico associativo
* db\_select\_multi: passata una query di selezione restituisce un array multiplo associativo
* db\_execute: procede ad eseguire la query passata che potrà essere di tipo INSERT, UPDATE, o DELETE.

Si è creata questa libreria per non duplicare il codice del PREPARED STATEMENT e garantire quindi per ogni operazione con il DB la sicurezza per il SQL INJECTION. Dato che si tratta di un database MySql si procederà ad utilizzare la connessione MYSQLI in modalità OOP per aver maggior performance utilizzando la connessione su driver dedicato. In alternativa si poteva utilizzare la connessione PDO con il vantaggio di poter gestire connessioni su DBMS diversi e non solo di tipologia MYSQL.

La procedura di esecuzione delle query utilizzerà il

*PREPARE*: metodo per la preparazione di un modello senza il passaggio di valori. Il DB analizza, compila ed esegue l’ottimizzazione della query sul modello inviato.

*EXECUTE*: in un secondo momento si inviano i valori dei parametri con un canale di comunicazione differente, si associano i valori con i parametri e si esegue l’istruzione.

Con questo metodo si hanno principalmente due vantaggi:

* riduzione dei tempi in quanto il DB analizza (controllo sintattico) solo una volta la query e le trasmissioni successive sono veloci perché di piccole dimensioni (solo i valori dei parametri)
* utili contro SQL INJECTION