



CERBOT GUI

—

Interfaccia Grafica Certbot

Reti di Calcolatori

a.a. 2024-2025

Corso di laurea in

Informatica e Comunicazione Digitale – Sede di Taranto

Team “Itopacci”

Cognome & Nome	Matricola	Email
Narracci Giovanni	776211	g.narracci6@studenti.uniba.it
Mongelli Antonio	777432	a.mongelli47@studenti.uniba.it
Pagliaro Francesco	777189	f.pagliaro@studenti.uniba.it

Docente: Lopez Ugo

Sommario

Documentazione Progetto: Certbot GUI	3
Contesto del Progetto	3
Obiettivi della GUI	3
Riferimenti Teorici	4
HTTP/HTTPS.....	4
SSL/TLS	4
Let's Encrypt.....	4
Guida all'installazione.....	5
Certbot vs Cerbot GUI	6
Crea un nuovo certificato:.....	7
Visualizza Certificati Attivi:.....	10
Gestisci Sites:	12
Crea Certificato Remoto	14
Domini di Test Utilizzati	14
Descrizione Dettagliata del Codice	17
Tecnologie Utilizzate	17
1. File PHP: Visualizzazione dei Certificati (index.php)	18
2. File CSS: Stile dell'Interfaccia (`index.css`)	18
3. File php: Creazione certificati (`create_certificate.php`)	22
4. File PHP: Stile Creazione certificati (create_certificate.css)	24
5. File JS: (create_certificates.php)	29
7. File CSS: (process.css)	31
8. File PHP: (show_sites.php)	33
9. File CSS: (show_sites.css)	37
10. File PHP: (view_certificates.php)	41
11. File CSS: (view_certificates.css)	45
12. File PHP: (delete.php)	47
13. File PHP: (renew.php)	48
14. File PHP: (download.php)	50
15. File PHP: (remote_certificate.php)	51
16. File PHP: (remote_process.php)	53
17. File PHP: (sender_enter.php)	56
18. File SH: (setup.sh)	57
Conclusioni	58
Possibili Miglioramenti Futuri:	58

Documentazione Progetto: **Certbot GUI**

Contesto del Progetto

Il progetto si concentra sulla creazione di un'interfaccia grafica utente (GUI) per semplificare l'utilizzo di **Certbot**, uno strumento open-source utilizzato per ottenere e rinnovare certificati SSL/TLS gratuiti tramite il servizio **Let's Encrypt**.

Certbot è uno strumento essenziale per garantire la sicurezza dei siti web tramite il protocollo **HTTPS**, ma il suo utilizzo tradizionale richiede l'uso della riga di comando, rendendo il processo poco accessibile a utenti con competenze tecniche limitate.

Obiettivi della GUI

L'interfaccia grafica proposta mira a:

1. **Semplificare l'interazione con Certbot**

- Fornire un'interfaccia visiva e intuitiva per generare certificati SSL/TLS senza necessità di conoscere la sintassi della riga di comando.
- Ridurre il rischio di errori di configurazione, offrendo un supporto grafico durante la procedura.

2. **Gestire i certificati esistenti**

- Permettere agli utenti di consultare facilmente l'elenco dei certificati installati.
- Mostrare informazioni chiave come **dominio associato, data di emissione, data di scadenza e stato di validità**.
- Rinnovo manuale dei certificati.
- Eliminazione dei certificati.

3. **Supportare diverse configurazioni**

- Consentire l'installazione di certificati utilizzando le modalità più comuni offerte da Certbot:
 - **Webroot:** metodo che utilizza una cartella specifica del server web per la validazione del certificato.
 - **Server Web Integrato:** per utenti che usano server web come **Apache e Nginx**.
- Offrire opzioni avanzate per la configurazione di rinnovi automatici.

Riferimenti Teorici

HTTP/HTTPS

HTTP (HyperText Transfer Protocol) è il protocollo utilizzato per la trasmissione di informazioni sul web.

HTTPS (HyperText Transfer Protocol Secure) è la versione sicura di **HTTP**, che utilizza protocolli di crittografia come **SSL/TLS** per proteggere la comunicazione tra client e server.

La principale differenza tra **HTTP** e **HTTPS** risiede nella sicurezza: mentre **HTTP** trasmette i dati in chiaro, rendendoli vulnerabili a intercettazioni, **HTTPS** cripta i dati, garantendo riservatezza e integrità delle informazioni scambiate. Inoltre, **HTTPS** utilizza la porta 443, mentre **HTTP** utilizza la porta 80.

L'adozione di **HTTPS** è fondamentale per proteggere i dati degli utenti e per migliorare il posizionamento nei motori di ricerca, poiché Google considera l'uso di **HTTPS** come un fattore di ranking.

SSL/TLS

SSL (Secure Sockets Layer) e **TLS** (Transport Layer Security) sono protocolli crittografici progettati per garantire la sicurezza delle comunicazioni su reti informatiche come Internet. Essi forniscono autenticazione, integrità dei dati e riservatezza, assicurando che le informazioni trasmesse tra client e server rimangano private e non siano alterate durante il trasferimento.

Inizialmente, **SSL** è stato sviluppato per proteggere le comunicazioni online, ma è stato successivamente sostituito da **TLS**, che offre miglioramenti in termini di sicurezza ed efficienza.

Un esempio comune dell'uso di **TLS** è **HTTPS**, la versione sicura del protocollo **HTTP**, che garantisce che i dati scambiati tra il browser dell'utente e il sito web siano crittografati e protetti da intercettazioni o manomissioni.

Let's Encrypt

Let's Encrypt è un'autorità di certificazione (CA) gratuita, automatizzata e aperta, gestita dal gruppo no-profit Internet Security Research Group (ISRG). Il suo obiettivo è rendere la rete web più sicura e rispettosa della privacy, offrendo certificati **SSL/TLS** senza costi. Il suo scopo è garantire che tutti i siti web siano sicuri e utilizzino il protocollo **HTTPS**.

Per facilitare l'implementazione dei certificati, **Let's Encrypt** fornisce strumenti come **Certbot**, che automatizzano il processo di ottenimento e rinnovo dei certificati **SSL/TLS**.

L'iniziativa ha contribuito in modo significativo all'aumento dell'adozione di **HTTPS**, rendendo più semplice e accessibile per i proprietari di siti web proteggere le comunicazioni dei loro utenti.

Let's Encrypt prevede la verifica del dominio attraverso metodi come il "**webroot**" o l'integrazione diretta con server web come **Apache** e **Nginx**.

Webroot

Metodo Webroot: Questo metodo utilizza una cartella specifica del server web per la validazione del certificato. Certbot, il client di **Let's Encrypt**, crea un file temporaneo in una directory designata (il "**webroot**") all'interno del server. Successivamente, **Let's Encrypt** tenta di accedere a questo file tramite **HTTP** per verificare che il richiedente abbia il controllo del dominio. Questo approccio è utile quando il server web è già in esecuzione e si desidera evitare interruzioni del servizio.

Server Web

Integrazione con Server Web: Per gli utenti che utilizzano server web come **Apache** o **Nginx**, **Certbot** offre plugin specifici che semplificano il processo di ottenimento e installazione dei certificati. Questi plugin possono configurare automaticamente il server per utilizzare i nuovi certificati e gestire le operazioni di rinnovo. Ad esempio, utilizzando il plugin **Apache**, **Certbot** può modificare i file di configurazione di Apache per abilitare **HTTPS** con il certificato ottenuto.

Entrambi i metodi garantiscono che solo chi ha effettivamente il controllo del dominio possa ottenere un certificato SSL/TLS valido, assicurando così la sicurezza delle comunicazioni tra il server e i client.

Apache

Apache HTTP Server: **Apache** è un server web open-source sviluppato dalla Apache Software Foundation. È noto per la sua flessibilità e per il supporto a numerosi moduli che estendono le sue funzionalità.

Nginx

Nginx: **Nginx** (pronunciato "Engine-X") è un server web open-source noto per le sue alte prestazioni e l'efficienza nella gestione di un gran numero di connessioni simultanee. Oltre a fungere da server web, **Nginx** può operare come reverse proxy, bilanciatore di carico e server di cache **HTTP**. È progettato per gestire in modo efficiente richieste statiche e dinamiche, ed è spesso utilizzato in ambienti ad alte prestazioni.

Entrambi i server possono essere utilizzati per ospitare siti web e applicazioni, ma la scelta tra Apache e **Nginx** dipende dalle specifiche esigenze del progetto, come la gestione del traffico, la facilità di configurazione e le prestazioni richieste.

Guida all'installazione

Guida all'installazione e configurazione di Certbot-GUI

1. Scaricare il codice da GitHub

Clonare il repository contenente il codice sorgente del programma.

2. Accedere alla cartella del programma

Dopo aver scaricato i file, aprire un terminale e spostarsi nella directory principale del programma.

3. Rendere eseguibile lo script di installazione

Per assicurarsi che lo script di setup possa essere eseguito correttamente, digitare il seguente comando:

```
sudo chmod +x setup.sh
```

4. Avviare il processo di installazione

Una volta reso eseguibile lo script, avviare la procedura di installazione con il comando:

```
sudo ./setup.sh
```

Durante l'installazione, il sistema chiederà se si desidera installare i servizi web server, come **Nginx** o **Apache**. Se si intende utilizzarli, confermare l'installazione seguendo le istruzioni a schermo.

5. Completamento dell'installazione

Al termine del processo, comparirà il messaggio: **"Installazione eseguita con successo."**

6. Configurazione dell'interfaccia web

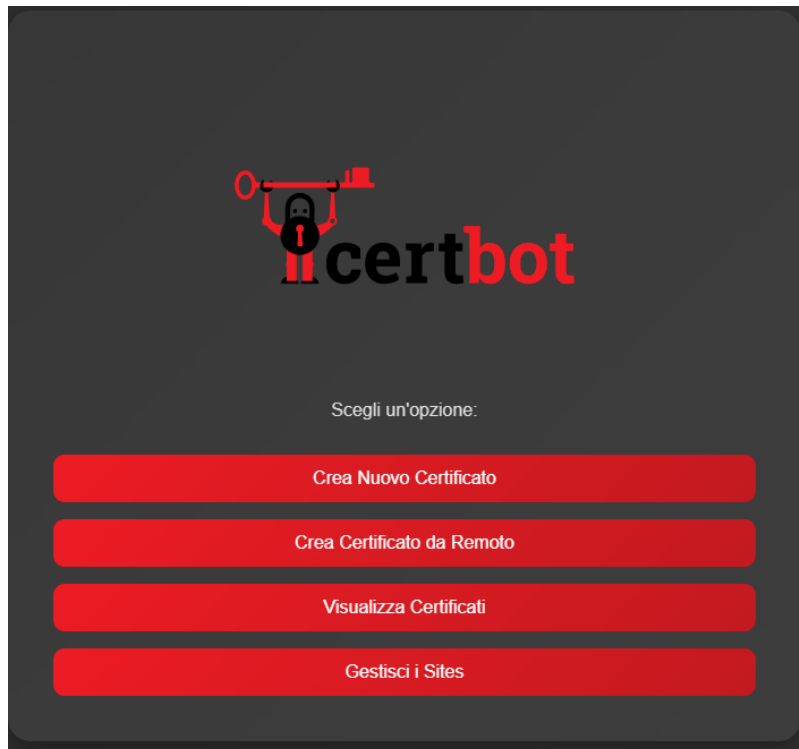
- Accedere alla directory della web root del proprio server.
- Copiare e incollare al suo interno i file contenuti nella cartella **Certbot-GUI**.

7. Accesso all'interfaccia utente

Dopo aver eseguito tutti i passaggi precedenti, aprire un browser e digitare nella barra degli indirizzi: <http://localhost>. Poiché il server web utilizza la porta di default **80**, non è necessario specificarla manualmente.

8. Inizio utilizzo

Una volta effettuato l'accesso all'interfaccia web, il programma sarà pronto all'uso.



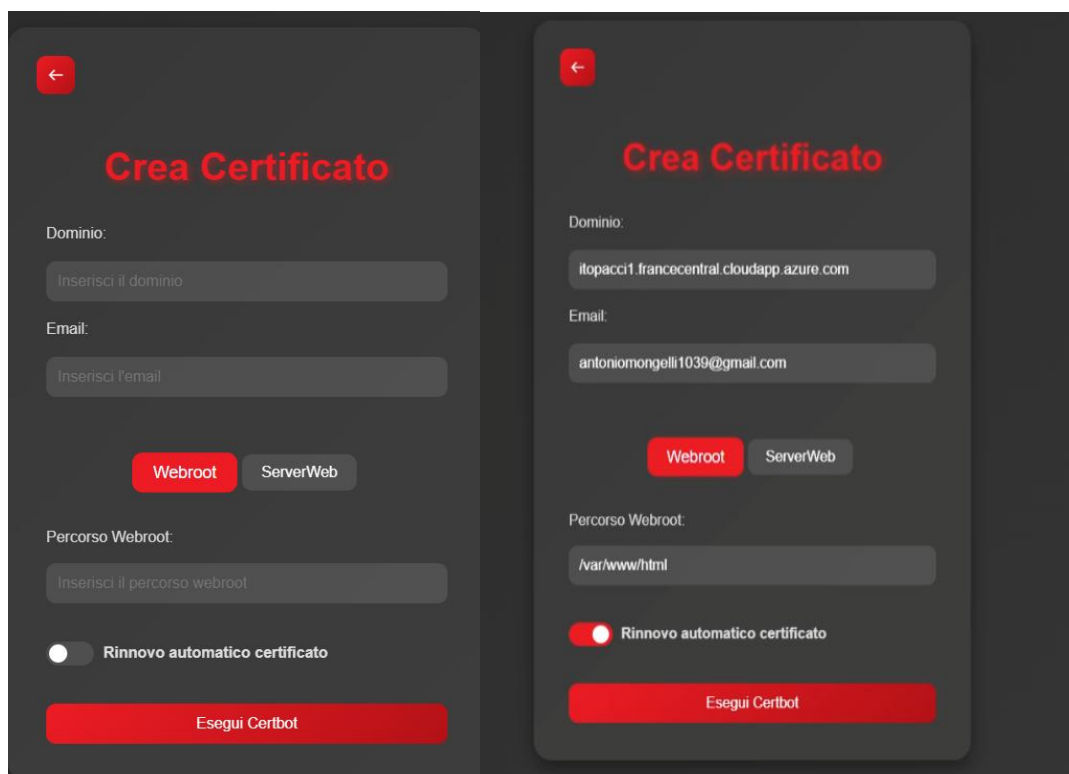
Nell'interfaccia iniziale di Certbot GUI troviamo quattro opzioni disponibili.

Certbot vs Cerbot GUI

Nell'interfaccia iniziale di certobot GUI troviamo tre opzioni disponibili.



Crea un nuovo certificato:



Che corrisponde al seguente comando:

Con Webroot:

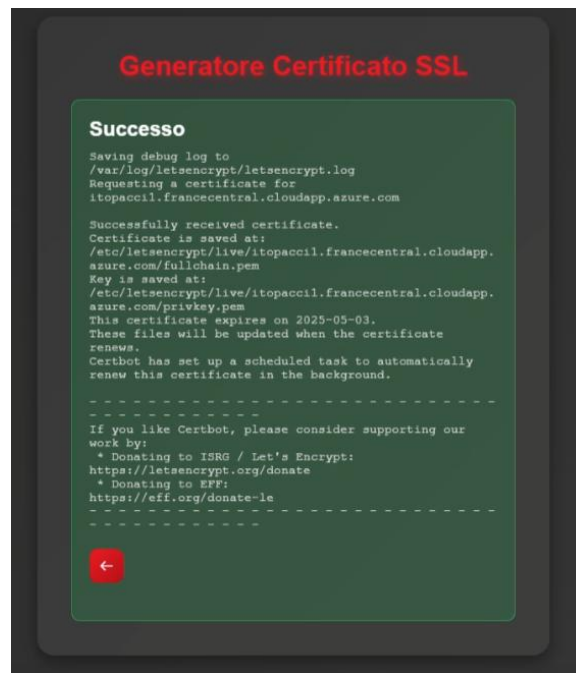
```
sudo certbot certonly --non-interactive --agree-tos --email prova@email.com -d example.it --webroot -w /var/www/html
```

Con ServerWeb:

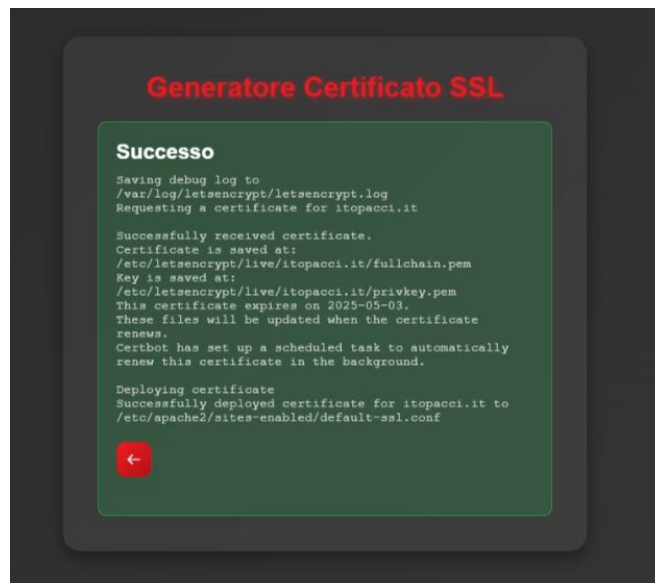
```
sudo certbot --nginx --non-interactive --agree-tos --email prova@email.com -d example.it
```

Esempio reale di creazione certificato con Certbot GUI

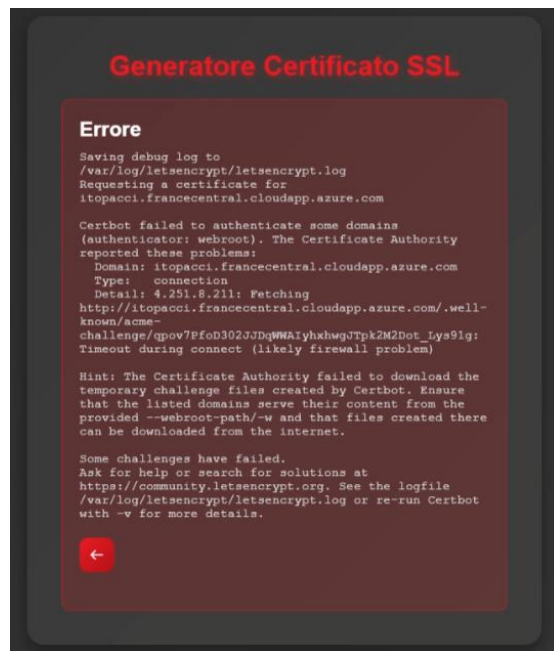
Con Successo(Webroot):



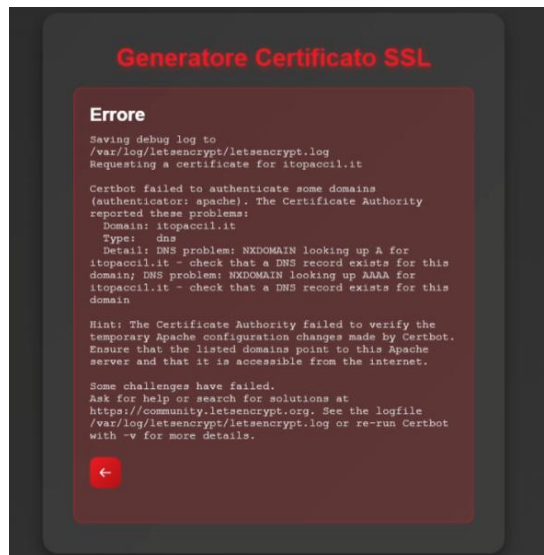
Con Successo (ServerWeb):



Con Errore(Webroot):

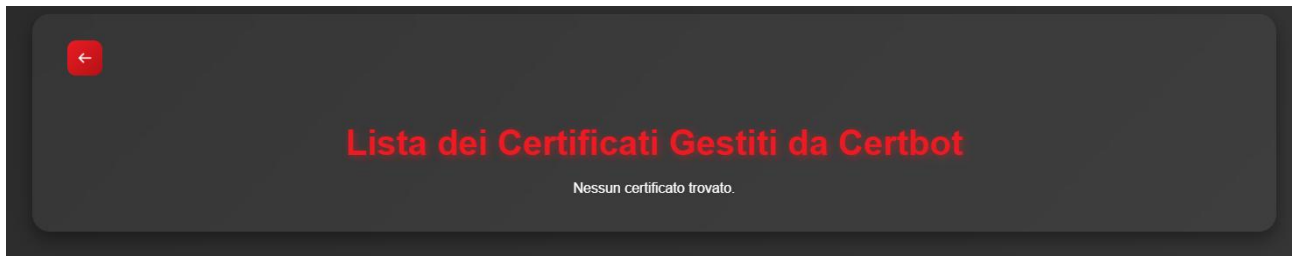


Con Errore(ServerWeb):



Visualizza Certificati Attivi:

Senza alcun Certificato:



Con Certificati:



Nel cmd corrisponde ad:

```
sudo certbot certificates
```

Eseguo Azione Rinnova:



Nel cmd corrisponde ad:

```
sudo certbot renew --cert-name example.it --force-renewal
```

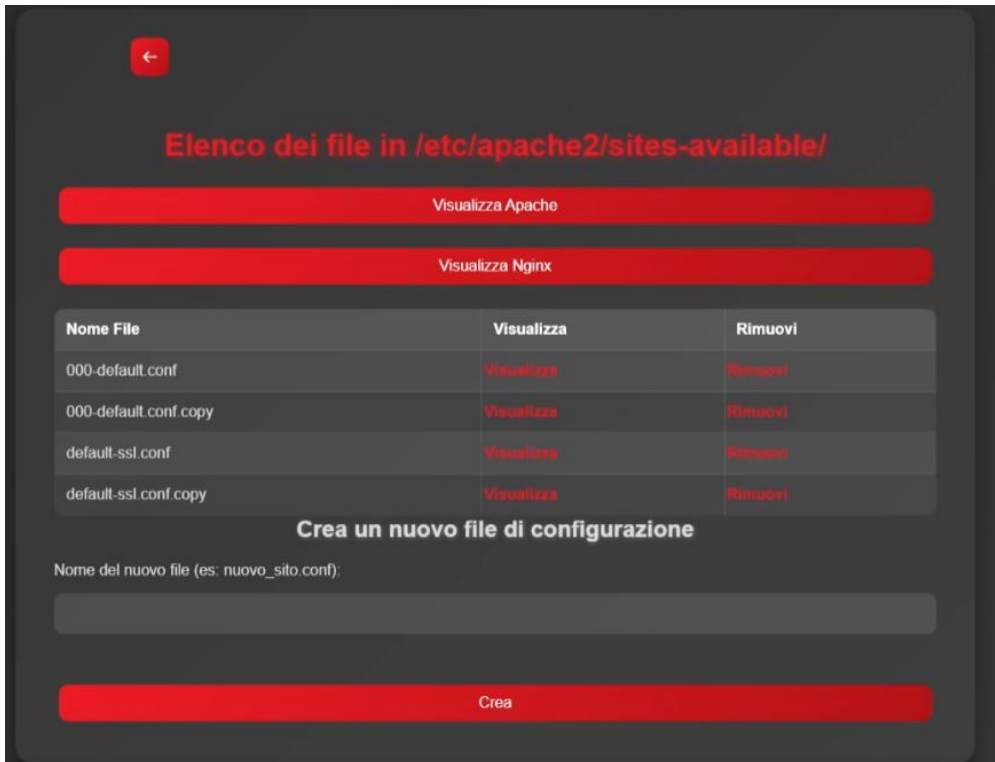
Eseguo Azione Elimina:



Nel cmd corrisponde ad:

```
sudo certbot delete --cert-name example.it
```

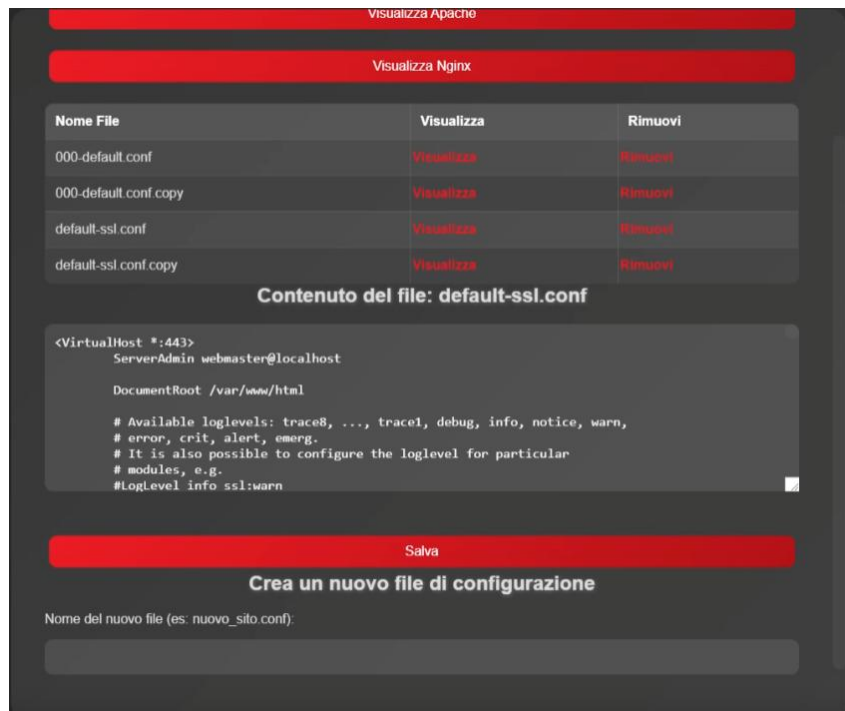
Gestisci Sites:



The screenshot shows the Certbot web interface for managing sites. At the top, there is a red back arrow button. Below it, the title "Elenco dei file in /etc/apache2/sites-available/" is displayed in red. There are two red buttons: "Visualizza Apache" and "Visualizza Nginx". Below these buttons is a table with three columns: "Nome File", "Visualizza", and "Rimuovi". The table lists four files: "000-default.conf", "000-default.conf.copy", "default-ssl.conf", and "default-ssl.conf.copy". Each file has a "Visualizza" link and a "Rimuovi" link. Below the table, there is a section titled "Crea un nuovo file di configurazione". It includes a text input field for the "Nome del nuovo file (es: nuovo_sito.conf):" and a red "Crea" button.

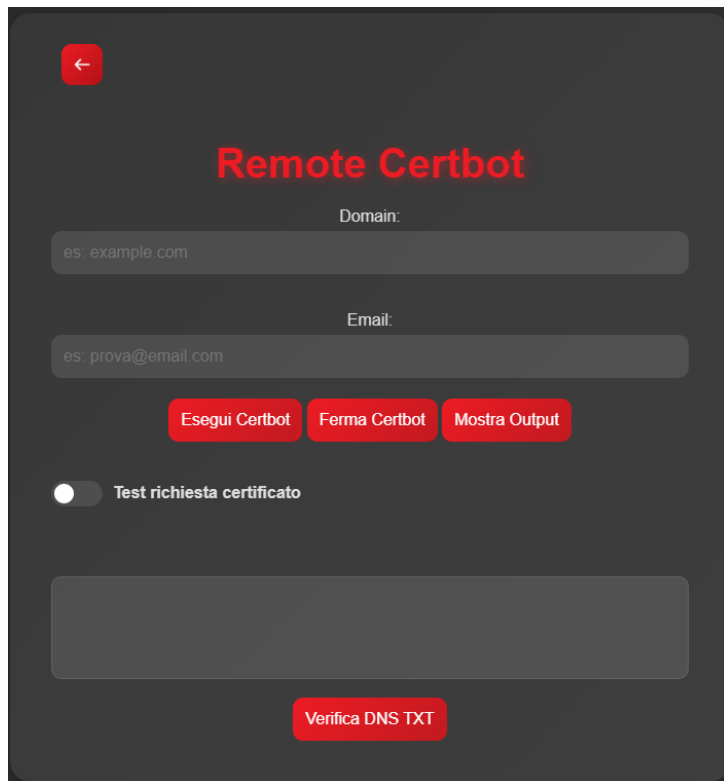
Nome File	Visualizza	Rimuovi
000-default.conf	Visualizza	Rimuovi
000-default.conf.copy	Visualizza	Rimuovi
default-ssl.conf	Visualizza	Rimuovi
default-ssl.conf.copy	Visualizza	Rimuovi

Premendo il pulsante visualizza Apache o Nginx verra visualizzato:



Per facilitare la creazione dei certificati Web Server Apache o Nginx si è implementato un editor di testo che fa accedere ai file dei sites permettendo direttamente dall'interfaccia di inserire il serverName al quale certBot farà riferimento per inserire il certificato. Inoltre, permette la creazione e la eliminazione dei file per i sites.

Crea Certificato Remoto



Esegui Certbot:

```
sudo certbot certonly --dry-run --email prova@gmail.com --agree-tos --manual --preferred-challenges dns -d example.com
```

Domini di Test Utilizzati

Per testare l'efficacia del sistema da noi progettato, sono stati utilizzati tre domini configurati su una **VPS di Azure**. uno di register.it, uno di Azure e uno di No-ip

1. <https://itopacci1.francecentral.cloudapp.azure.com/>
2. <https://itopacci.it/>
3. <https://itopacci.zapto.org/>
4. <https://itopacci.hopto.org/>
5. <https://itopacci1.it/>

1. Dominio "itopacci1.francecentral.cloudapp.azure.com":

Questo dominio è associato a un'applicazione ospitata su Microsoft Azure, una piattaforma cloud che offre servizi di hosting e gestione delle applicazioni.

2. Dominio "itopacci.it":

Questo dominio è stato registrato tramite Register.it, un provider italiano che offre servizi di registrazione domini, hosting e altri servizi web.

3. Dominio "itopacci.zapto.org":

1. Questo dominio è stato registrato tramite No-ip (<https://www.noip.com>), offre servizi di registrazione domini e altri servizi.

4. Dominio "itopacci.hopto.org":

1. Questo dominio è stato registrato tramite No-ip (<https://www.noip.com>), offre servizi di registrazione domini e altri servizi.

5. Dominio "itopacci.it":

1. Questo dominio è stato registrato tramite Register.it, un provider italiano che offre servizi di registrazione domini, hosting e altri servizi web.

Installazione e Configurazione dei Domini:

La registrazione di un dominio comporta la scelta di un nome univoco e la sua registrazione tramite un registrar accreditato. Una volta registrato, è possibile configurare i record DNS per indirizzare il dominio verso il server desiderato, che sia un server web come Apache o Nginx, o un'applicazione ospitata su una piattaforma cloud come Azure.

Per il **dominio "itopacci.it"**, la configurazione dei record DNS può essere gestita tramite il pannello di controllo di Register.it, dove è possibile impostare i record A, CNAME, MX e altri necessari per il corretto funzionamento del sito web e dei servizi associati.

Per il **dominio "itopacci1.francecentral.cloudapp.azure.com"**, essendo un dominio generato automaticamente da Azure, la configurazione avviene tramite il portale di Azure in automatico.

Per il **dominio "itopacci.zapto.org"**, la configurazione dei record DNS può essere gestita tramite il pannello di controllo di No-ip, dove è possibile impostare i record A, CNAME, MX e altri necessari per il corretto funzionamento del sito web e dei servizi associati.

Per il **dominio "itopacci.hopto.org"**

", la configurazione dei record DNS può essere gestita tramite il pannello di controllo di No-ip, dove è possibile impostare i record A, CNAME, MX e altri necessari per il corretto funzionamento del sito web e dei servizi associati.

Per il **dominio "itopacci1.it"**, la configurazione dei record DNS può essere gestita tramite il pannello di controllo di Register.it, dove è possibile impostare i record A, CNAME, MX e altri necessari per il corretto funzionamento del sito web e dei servizi associati.

Visualizzatore certificati: progetto-reti

Generali

Dettagli

Rilasciato a

Nome comune (CN)

progetto-reti

Organizzazione (O)

<Non parte del certificato>

Unità organizzativa (OU)

<Non parte del certificato>

Emesso da

Nome comune (CN)

progetto-reti

Organizzazione (O)

<Non parte del certificato>

Unità organizzativa (OU)

<Non parte del certificato>

Periodo di validità

Emesso in data

giovedì 30 gennaio 2025 alle ore 16:26:12

Scade in data

domenica 28 gennaio 2035 alle ore 16:26:12

Impronte SHA-256

Certificato

776f59feb542c39ee7495e8008f4cd825b276e28a1c6098410382d5e8480a348

Chiave pubblica

af8462b4ebb31ef545b879df6bcfb19d4340a63ed5437298a4fddb7905077e0

Non sicuro

https://itopacci1.francecentral.cloud

itopacci1.francecentral.cloudapp.azure.com

La tua connessione a questo sito non è sicura

Non dovresti inserire dati sensibili in questo sito (ad esempio password o carte di credito) perché potrebbero essere intercettati da utenti malintenzionati. [Scopri di più](#)

Dettagli certificato

Cookie e dati dei siti

>

Impostazioni sito

16

Itopacci1.francecentral.cloud con il certificato

The image shows two overlapping windows. The background window is a 'Visualizzatore certificati' (Certificate Viewer) for 'itopacci1.francecentral.cloudapp.azure.com'. It has tabs for 'Generali' and 'Dettagli'. The 'Generali' tab is active, showing details about the certificate's issuance and validity.

Rilasciato a	
Nome comune (CN)	itopacci1.francecentral.cloudapp.azure.com
Organizzazione (O)	<Non parte del certificato>
Unità organizzativa (OU)	<Non parte del certificato>

Emesso da	
Nome comune (CN)	E6
Organizzazione (O)	Let's Encrypt
Unità organizzativa (OU)	<Non parte del certificato>

Periodo di validità	
Emesso in data	domenica 2 febbraio 2025 alle ore 19:17:37
Scade in data	sabato 3 maggio 2025 alle ore 20:17:36

Impronte SHA-256	
Certificato	bc72c8614aae875386956e896b30207bddd4ff20190718c8bf8d7192934c7c9
Chiave pubblica	bb714793dc2babb3ca3e5e113c31c9fd8a3038485186eeb4bc8367c4a84014a2

The foreground window is a 'Sicurezza' (Security) overlay for the same site. It features a lock icon and the text 'La connessione è sicura' (The connection is secure), explaining that information like passwords remains private. It also includes a link 'Scopri di più' (Learn more) and a status 'Il certificato è valido' (The certificate is valid) with a checkmark icon.

Descrizione Dettagliata del Codice

Tecnologie Utilizzate

Il progetto è sviluppato utilizzando tecnologie web standard per garantire compatibilità e facilità di utilizzo:

- **HTML** → Definisce la struttura dell'interfaccia utente.
- **CSS** → Fornisce uno stile moderno e responsive per migliorare l'esperienza visiva.
- **JavaScript** → Implementa la logica interattiva lato client, migliorando la dinamicità dell'interfaccia.
- **PHP** → Si occupa delle operazioni lato server, inclusa:
 - L'esecuzione dei comandi **Certbot** in background.
 - La lettura e visualizzazione dei certificati esistenti.

1. File PHP: Visualizzazione dei Certificati (index.php)

```
<!DOCTYPE html>
<html lang="it">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Dashboard Certbot</title>
  <link rel="stylesheet" href="index.css"> <!-- Collegamento al file CSS -->
</head>
<body>
  <div class="container">
    
    <p>Scegli un'opzione:</p>
    <div class="button-container">
      <!-- Pulsante per creare un nuovo certificato -->
      <a href="createCertificate/create_certificate.php" class="btn">Crea Nuovo Certificato</a>
      <a href="remote_certificate.php" class="btn">Crea Certificato da Remoto</a>
      <!-- Pulsante per visualizzare i certificati attivi -->
      <a href="viewCertificates/view_certificates.php" class="btn">Visualizza Certificati</a>
      <a href="showSites/show_sites.php" class="btn">Gestisci i Sites</a>
    </div>
  </div>
</body>
</html>
```

Questo codice PHP realizza una dashboard dedicata a Certbot.

2. File CSS: Stile dell'Interfaccia (`index.css`)

```
/* Reset base */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

/* Stile generale del body */
body {
  font-family: 'Arial', sans-serif;
  background: linear-gradient(135deg, rgba(17, 17, 17, 0.9), rgba(34, 34, 34, 0.9));
  color: #ffffff;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  overflow: auto;
}

/* Stile del contenitore principale */
.container {
  background: rgba(255, 255, 255, 0.05);
```

```

    backdrop-filter: blur(10px);
    border-radius: 20px;
    box-shadow: 0 8px 20px rgba(0, 0, 0, 0.4);
    width: 100%;
    max-width: 700px;
    padding: 40px;
    text-align: center;
    animation: fadeIn 1s ease-in-out;
    overflow-y: auto;
}

/* Stile del pulsante "back" */
.back {
    align-self: flex-start; /* Lo sposta a sinistra */
    margin-bottom: 20px; /* Distanza dagli altri elementi */
    position: relative;
    left: -280px; /* Sposta l'elemento 20px più a sinistra */
    top: -30px;
    background: linear-gradient(135deg, #ed1c24, #b31217); /* Gradiente rosso */
    color: white; /* Testo bianco */
    padding: 10px; /* Spaziatura interna */
    border: none; /* Nessun bordo */
    border-radius: 10px; /* Bordi arrotondati */
    font-size: 5rem; /* Dimensione del testo */
    cursor: pointer; /* Cambia il cursore in una mano */
    transition: background 0.3s ease, transform 0.2s ease; /* Transizione fluida */
    margin-top: 20px; /* Spazio sopra il pulsante */
}

.back:hover {
    background: linear-gradient(135deg, #b31217, #ed1c24); /* Gradiente invertito */
    transform: scale(1.05); /* Effetto di ingrandimento */
}

/* Stile per un contenitore più grande */
.container-lg {
    background: rgba(255, 255, 255, 0.05);
    backdrop-filter: blur(10px);
    border-radius: 20px;
    box-shadow: 0 8px 20px rgba(0, 0, 0, 0.4);
    width: 100%;
    max-width: 600px;
    padding: 40px;
    text-align: left;
    animation: fadeIn 1s ease-in-out;
}

/* Stile del titolo */
h1 {
    font-size: 2.5rem;
    color: #ed1c24; /* Nuovo colore rosso */
    margin-bottom: 20px;
    text-shadow: 2px 2px 10px rgba(237, 28, 36, 0.5); /* Ombra del testo adattata al nuovo colore */
}

h2 {
    font-size: 1rem;
    color: #ddd;
}

/* Stile dell'area di output */
.output-name-value {
    background: rgba(255, 255, 255, 0.1);
    border-radius: 10px;
    padding: 3px;
    height: 100px;
    width: 100%;
    color: #ddd;
    font-family: 'Courier New', monospace;
    font-size: 0.5rem;
    white-space: pre-wrap;
}

```

```

    word-wrap: break-word;
    border: 1px solid rgba(255, 255, 255, 0.1);
}

/* Stile del paragrafo */
p {
    font-size: 1rem;
    color: #ddd;
    margin-bottom: 30px;
}

/* Stile del contenitore dei pulsanti */
.button-container {
    display: flex;
    flex-direction: column;
    gap: 15px;
}

/* Stile dei pulsanti */
.btn {
    background: linear-gradient(135deg, #ed1c24, #c21a20); /* Gradiente rosso */
    color: white;
    padding: 12px;
    border: none;
    border-radius: 10px;
    font-size: 1rem;
    cursor: pointer;
    transition: background 0.3s ease, transform 0.2s ease;
    text-decoration: none;
    text-align: center;
}

/* Effetto hover sui pulsanti */
.btn:hover {
    background: linear-gradient(135deg, #c21a20, #ed1c24); /* Gradiente invertito */
    transform: scale(1.05);
}

/* Effetto click sui pulsanti */
.btn:active {
    transform: scale(0.98);
}

/* Stile dell'area di output */
.output-box {
    background: rgba(255, 255, 255, 0.1);
    border-radius: 10px;
    padding: 20px;
    height: 300px;
    overflow-y: auto;
    color: #ddd;
    font-family: 'Courier New', monospace;
    font-size: 0.9rem;
    white-space: pre-wrap;
    word-wrap: break-word;
    border: 1px solid rgba(255, 255, 255, 0.1);
}

/* Add these styles */
#output {
    display: none;
}

.show {
    display: block !important;
}

/* Stile dei gruppi di input */
.input-group {
    margin-bottom: 20px;
    align-items: center;
}

```

```

}

.input-group label {
  font-size: 1rem;
  color: #ddd;
  margin-bottom: 5px;
  display: block;
}

.input-group input {
  width: 100%;
  padding: 12px;
  border: none;
  border-radius: 10px;
  background: rgba(255, 255, 255, 0.1);
  color: #ffffff;
  font-size: 1rem;
  transition: background 0.3s ease, transform 0.2s ease;
}

.input-group input:focus {
  background: rgba(255, 255, 255, 0.2);
  outline: none;
  transform: scale(1.02);
}

.test {
  display: flex; /* Allinea gli elementi in linea */
  align-items: center; /* Allinea verticalmente */
  gap: 10px; /* Spazio tra il checkbox e l'etichetta */
  margin-top: 20px; /* Spazio sopra il checkbox */
}

/* Nascondi il checkbox reale */
.test input[type="checkbox"] {
  display: none; /* Nasconde il checkbox predefinito */
}

/* Stile della label che funge da toggle */
.test label {
  position: relative;
  display: inline-block;
  width: 50px; /* Larghezza del toggle */
  height: 25px; /* Altezza del toggle */
  background: rgba(255, 255, 255, 0.1); /* Sfondo semi-trasparente */
  border-radius: 15px; /* Bordi arrotondati */
  cursor: pointer; /* Cambia il cursore in una mano */
  transition: background 0.3s ease; /* Transizione fluida */
}

/* Stile del cerchio interno (indicatore) */
.test label::before {
  content: '';
  position: absolute;
  top: 3px;
  left: 3px;
  width: 19px; /* Dimensione del cerchio */
  height: 19px; /* Dimensione del cerchio */
  background: white; /* Colore del cerchio */
  border-radius: 50%; /* Cerchio perfetto */
  transition: transform 0.3s ease; /* Transizione fluida */
}

/* Quando il checkbox è selezionato */
.test input[type="checkbox"]:checked + label {
  background: #ed1c24; /* Cambia il colore dello sfondo quando attivo */
}

/* Anima il cerchio interno quando il checkbox è selezionato */
.test input[type="checkbox"]:checked + label::before {

```

```

    transform: translateX(24px); /* Muove il cerchio a destra */
}

/* Stile del testo accanto al toggle */
.test span {
    font-size: 1rem; /* Dimensione del testo */
    color: #ddd; /* Colore grigio chiaro */
    font-weight: bold; /* Testo in grassetto */
}

/* Animazione di caricamento */
@keyframes fadeIn {
    from {
        opacity: 0;
        transform: translateY(-20px);
    }
    to {
        opacity: 1;
        transform: translateY(0);
    }
}

#status-messages {
    margin: 10px 0;
    padding: 10px;
    min-height: 30px;
}

.error-message {
    background-color: #ffebee;
    color: #c62828;
    padding: 10px;
    border-radius: 4px;
    border: 1px solid #ef9a9a;
}

.success-message {
    background-color: #e8f5e9;
    color: #2e7d32;
    padding: 10px;
    border-radius: 4px;
    border: 1px solid #a5d6a7;
}

.warning-message {
    background-color: #fff3e0;
    color: #ef6c00;
    padding: 10px;
    border-radius: 4px;
    border: 1px solid #ffcc80;
}

```

3. File php: Creazione certificati (`create_certificate.php`)

```

<!DOCTYPE html>
<html lang="en">
<head>
    <!-- Metadati della pagina -->
    <meta charset="UTF-8"> <!-- Codifica dei caratteri -->
    <meta name="viewport" content="width=device-width, initial-scale=1.0"> <!-- Ottimizzazione per
dispositivi mobili -->
    <title>Certbot </title> <!-- Titolo della pagina -->
    <link rel="stylesheet" href="create_certificate.css"> <!-- Collegamento al file CSS -->
</head>

```

```

<body>

  <!-- Contenitore principale -->
  <div class="container">
    <a href="../index.php">
      
    </a>

    <!-- Titolo della pagina -->
    <h1>Crea Certificato </h1><br>
    <!-- Form per l'input utente -->
    <form method="POST" action="processCertificate/process.php">
      <!-- Campo per il dominio -->
      <label for="domain">Dominio:</label>
      <input type="text" id="domain" name="domain" placeholder="Inserisci il dominio" required>

      <!-- Campo per l'email -->
      <label for="email">Email:</label>
      <input type="email" id="email" name="email" placeholder="Inserisci l'email" required>

      <!-- Pulsanti per alternare tra Webroot e ServerWeb -->
      <div class="toggle-container">
        <button type="button" class="toggle-btn active" data-target="webroot">Webroot</button>
        <button type="button" class="toggle-btn" data-target="server">ServerWeb</button>
      </div>

      <!-- Sezione Webroot (visibile di default) -->
      <div id="webroot">
        <label for="webroot-input">Percorso Webroot:</label><br>
        <input type="text" id="webroot-input" name="webroot" placeholder="Inserisci il percorso
webroot" required>
      </div>

      <!-- Sezione ServerWeb (nascosta di default) -->
      <div id="server" class="hidden">
        <label for="server-select">Seleziona il server web:</label><br>
        <label for="server-hint">(NB: Bisogna inserire il serverName nella configurazione Site
di interesse)</label><br>
        <select id="server-select" name="server">
          <option value="apache">Apache</option>
          <option value="nginx">Nginx</option>
          <!-- <option value="lighttpd">Lighttpd</option>
          <option value="caddy">Caddy</option> -->
        </select>
      </div>

      <!-- Checkbox del Rinnovo automatico -->
      <div class="renew">
        <input type="checkbox" id="renewCheckbox" name="renew">
        <label for="renewCheckbox"></label>
        <span>Rinnovo automatico certificato</span>
      </div>

      <!-- Checkbox del Test certificato -->
      <div class="test">
        <input type="checkbox" id="testCheckbox" name="test">
        <label for="testCheckbox"></label>
        <span>Test richiesta certificato</span>
      </div>

      <!-- Pulsante di invio del form -->
      <button type="submit">Esegui Certbot</button>
    </form>
  </div>

  <!-- Collegamento al file JavaScript -->
  <script src="create_certificate.js"></script>

```

```
</body>
</html>
```

Questo documento HTML definisce una pagina web per la creazione di certificati tramite Certbot e si compone di più sezioni funzionali:

Il form, che utilizza il metodo POST e punta al file PHP "processCertificate/process.php", raccoglie le informazioni necessarie per la generazione del certificato. Le componenti del form sono:

- **Input per il dominio:** Un campo di testo in cui l'utente deve inserire il dominio per il quale desidera il certificato.
- **Input per l'email:** Un campo email che richiede l'indirizzo email dell'utente, necessario probabilmente per le comunicazioni o il recupero del certificato.
- **Opzioni aggiuntive:**
 - Una checkbox per abilitare il rinnovo automatico del certificato, consentendo al sistema di gestire automaticamente gli aggiornamenti.
 - Una checkbox per effettuare una richiesta di test, probabilmente per verificare che il processo di creazione del certificato funzioni correttamente senza incidere sul sistema in produzione.

4. File PHP: Stile Creazione certificati (create_certificate.css)

```
/* Reset base */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

/* Stile generale del body */
body {
  font-family: 'Arial', sans-serif;
  background: linear-gradient(135deg, rgba(17, 17, 17, 0.9), rgba(34, 34, 34, 0.9));
  color: #ffffff;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  overflow: hidden;
}

/* Stile del contenitore principale */
.container {
  background: rgba(255, 255, 255, 0.05);
  backdrop-filter: blur(10px);
  border-radius: 20px;
  box-shadow: 0 8px 20px rgba(0, 0, 0, 0.4);
  width: 100%;
  max-width: 500px;
  padding: 40px;
```



```

    text-align: center;
    animation: fadeIn 1s ease-in-out; /* Animazione di caricamento */
}

/* Stile del pulsante "back" */
.back {
    align-self: flex-start;
    margin-bottom: 20px;
    position: relative;
    left: -200px;
    top: -30px;
    background: linear-gradient(135deg, #ed1c24, #b31217);
    color: white;
    padding: 10px;
    border: none;
    border-radius: 10px;
    font-size: 5rem;
    cursor: pointer;
    transition: background 0.3s ease, transform 0.2s ease;
    margin-top: 20px;
}

.back:hover {
    background: linear-gradient(135deg, #b31217, #ed1c24); /* Gradiente invertito */
    transform: scale(1.05); /* Effetto di ingrandimento */
}

/* Stile del titolo */
h1 {
    font-size: 2.5rem;
    color: #ed1c24;
    margin-bottom: 20px;
    text-shadow: 2px 2px 10px rgba(237, 28, 36, 0.5);
}

/* Stile del form */
form {
    display: flex;
    flex-direction: column;
    gap: 20px;
}

/* Stile delle etichette */
label {
    font-size: 1rem;
    color: #ddd;
    text-align: left;
    display: block;
}

/* Stile degli input e del select */
input {
    width: 100%;
    padding: 12px;
    border: none;
    border-radius: 10px;
    background: rgba(255, 255, 255, 0.1);
    color: #ffffff;
    font-size: 1rem;
    transition: background 0.3s ease, transform 0.2s ease;
}

/* Stile per il select */
select {
    width: 100%;
    padding: 12px;
    border: none;
    border-radius: 10px;
    background: rgba(255, 255, 255, 0.1);
    color: #ffffff;
    font-size: 1rem;

```

```

    transition: background 0.3s ease, transform 0.2s ease;
    appearance: none;
    -webkit-appearance: none;
    -moz-appearance: none;
    cursor: pointer;
}

/* Stile per il select quando è in focus */
select:focus {
    background: rgba(255, 255, 255, 0.2);
    outline: none;
    transform: scale(1.02);
}

/* Stile per la freccia del dropdown (personalizzata con un'icona) */
select {
    background-image: url("data:image/svg+xml,%3Csvg xmlns='http://www.w3.org/2000/svg' viewBox='0 0 24
24' fill='%23ffffff'%3E%3Cpath d='M7 10 5 5 15 5'/%3E%3C/svg%3E");
    background-repeat: no-repeat;
    background-position: right 10px center;
    background-size: 16px;
    padding-right: 40px; /* Spazio per la freccia */
}

/* Stile per il contenuto della tendina (limitato supporto) */
select option {
    background: #333;
    color: #ffffff;
    padding: 10px;
}

/* Stile degli input e del select quando sono in focus */
input:focus,
select:focus {
    background: rgba(255, 255, 255, 0.2);
    outline: none;
    transform: scale(1.02);
}

/* Stile dei pulsanti */
button {
    background: linear-gradient(135deg, #ed1c24, #b31217);
    color: white;
    padding: 12px;
    border: none;
    border-radius: 10px;
    font-size: 1rem;
    cursor: pointer;
    transition: background 0.3s ease, transform 0.2s ease;
    margin-top: 20px;
}

/* Stile dei pulsanti quando si passa sopra con il mouse */
button:hover {
    background: linear-gradient(135deg, #b31217, #ed1c24);
    transform: scale(1.05);
}

/* Stile dei pulsanti quando si clicca */
button:active {
    transform: scale(0.98); /* Effetto di compressione */
}

/* Classe per nascondere elementi */
.hidden {
    display: none !important; /* Nasconde completamente l'elemento */
}

/* Stile del contenitore dei pulsanti toggle */
.toggle-container {
    display: flex;

```

```

    align-items: center;
    justify-content: center;
    gap: 10px;
    margin: 20px 0;
}

/* Stile dei pulsanti toggle */
.toggle-btn {
    background: rgba(255, 255, 255, 0.1);
    color: #fff;
    padding: 10px 20px;
    border: none;
    border-radius: 10px;
    cursor: pointer;
    transition: background 0.3s ease, transform 0.2s ease;
    font-size: 1rem;
}

/* Stile del pulsante toggle attivo */
.toggle-btn.active {
    background: #ed1c24;
    transform: scale(1.1);
}

/* Stile del pulsante toggle non attivo quando si passa sopra con il mouse */
.toggle-btn:not(.active):hover {
    background: rgba(255, 255, 255, 0.2);
}

/* Stile per il contenitore del checkbox */
.renew {
    display: flex;
    align-items: center;
    gap: 10px;
    margin-top: 20px;
}

/* Nascondi il checkbox reale */
.renew input[type="checkbox"] {
    display: none;
}

/* Stile della label che funge da toggle */
.renew label {
    position: relative;
    display: inline-block;
    width: 50px;
    height: 25px;
    background: rgba(255, 255, 255, 0.1);
    border-radius: 15px;
    cursor: pointer;
    transition: background 0.3s ease;
}

/* Stile del cerchio interno (indicatore) */
.renew label::before {
    content: '';
    position: absolute;
    top: 3px;
    left: 3px;
    width: 19px;
    height: 19px;
    background: white;
    border-radius: 50%;
    transition: transform 0.3s ease;
}

/* Quando il checkbox è selezionato */
.renew input[type="checkbox"]:checked + label {
    background: #ed1c24;
}

```

```

/* Anima il cerchio interno quando il checkbox è selezionato */
.renew input[type="checkbox"]:checked + label::before {
  transform: translateX(24px); /* Muove il cerchio a destra */
}

/* Stile del testo accanto al toggle */
.renew span {
  font-size: 1rem;
  color: #ddd;
  font-weight: bold;
}

.test {
  display: flex;
  align-items: center;
  gap: 10px; /* Spazio tra il checkbox e l'etichetta */
  margin-top: 20px;
}

/* Nascondi il checkbox reale */
.test input[type="checkbox"] {
  display: none;
}

/* Stile della label che funge da toggle */
.test label {
  position: relative;
  display: inline-block;
  width: 50px;
  height: 25px;
  background: rgba(255, 255, 255, 0.1);
  border-radius: 15px;
  cursor: pointer;
  transition: background 0.3s ease;
}

/* Stile del cerchio interno (indicatore) */
.test label::before {
  content: '';
  position: absolute;
  top: 3px;
  left: 3px;
  width: 19px;
  height: 19px;
  background: white;
  border-radius: 50%;
  transition: transform 0.3s ease;
}

/* Quando il checkbox è selezionato */
.test input[type="checkbox"]:checked + label {
  background: #ed1c24;
}

/* Anima il cerchio interno quando il checkbox è selezionato */
.test input[type="checkbox"]:checked + label::before {
  transform: translateX(24px);
}

/* Stile del testo accanto al toggle */
.test span {
  font-size: 1rem;
  color: #ddd;
  font-weight: bold;
}

/* Animazione di caricamento */
@keyframes fadeIn {
  from {
    opacity: 0; /* Inizia invisibile */
  }
}

```

```

    transform: translateY(-20px);
  }
  to {
    opacity: 1; /* Diventa visibile */
    transform: translateY(0); /* Torna alla posizione originale */
  }
}

```

5. File JS: (create_certificates.php)

```

document.addEventListener("DOMContentLoaded", function () {
  // Seleziona gli elementi del DOM
  const toggleButtons = document.querySelectorAll('.toggle-btn');
  const webrootDiv = document.getElementById('webroot');
  const serverDiv = document.getElementById('server');
  const webrootInput = document.getElementById('webroot-input');
  const serverSelect = document.getElementById('server-select');
  document.getElementById('server-select').value = "";

  // Imposta lo stato iniziale
  webrootDiv.classList.remove('hidden');
  serverDiv.classList.add('hidden');
  webrootInput.setAttribute('required', 'true');
  serverSelect.removeAttribute('required');

  // Aggiunge event listeners ai pulsanti toggle
  toggleButtons.forEach(button => {
    button.addEventListener('click', () => {
      toggleButtons.forEach(btn => btn.classList.remove('active'));
      button.classList.add('active');

      // Mostra/nascondi le sezioni in base al pulsante selezionato
      const target = button.dataset.target;
      if (target === 'webroot') {
        document.getElementById('server-select').value = "";
        webrootDiv.classList.remove('hidden');
        serverDiv.classList.add('hidden');
        webrootInput.setAttribute('required', 'true');
        serverSelect.removeAttribute('required');
      } else if (target === 'server') {
        document.getElementById('webroot-input').value = "";
        document.getElementById('server-select').value = "apache";
        serverDiv.classList.remove('hidden');
        webrootDiv.classList.add('hidden');
        serverSelect.setAttribute('required', 'true');
        webrootInput.removeAttribute('required');
      }
    });
  });
});

```

Questo script PHP gestisce l'intero processo per la generazione di un certificato SSL tramite Certbot, integrando la validazione dei dati di input, la costruzione ed esecuzione del comando Certbot e, se richiesto, la configurazione automatica del rinnovo del certificato.

Inizializzazione e Acquisizione dei Dati

- **Avvio della sessione:** All'inizio viene avviata una sessione PHP con `session_start()`, utile per gestire eventuali variabili di sessione in seguito.
- **Preparazione delle variabili:** Le variabili `$output` ed `$error` vengono inizializzate per salvare rispettivamente il risultato del comando ed un flag che indica la presenza di errori.
- **Controllo del metodo POST:** Il codice verifica che la richiesta sia di tipo POST, in modo da elaborare solo i dati provenienti dal form.

Recupero e Validazione degli Input

Vengono estratti dalla richiesta i valori inseriti dall'utente per:

Dominio e Email: Essenziali per la configurazione del certificato.

Webroot e Server: A seconda della modalità scelta (Webroot o ServerWeb), viene usato uno dei due metodi per la validazione del dominio.

Opzioni aggiuntive:

Rinnovo automatico (renew): Verifica se l'utente ha abilitato il rinnovo automatico.

Test (test): Se abilitato, il comando verrà eseguito in modalità test (dry-run).

Validazione dei campi obbligatori: Se il dominio o l'email risultano vuoti, il codice imposta un messaggio di errore e blocca l'esecuzione, evidenziando che tutti i campi obbligatori devono essere compilati.

Costruzione del Comando Certbot

Scelta della modalità di validazione:

Metodo Webroot: Se il campo relativo al server è vuoto, viene costruito un comando che utilizza il parametro `--webroot` e specifica il percorso fornito dall'utente tramite l'input "webroot".

Metodo ServerWeb: Se il campo relativo al server è valorizzato, il comando viene configurato per utilizzare il server specificato (ad esempio Apache o Nginx) senza richiedere il percorso webroot.

Modalità di test: Se l'utente ha selezionato la modalità test, viene aggiunta l'opzione `--dry-run` al comando, permettendo di verificare il funzionamento del comando senza modificare l'ambiente di produzione.

Reindirizzamento degli errori: L'aggiunta di `2>&1` alla fine del comando garantisce che sia l'output standard che gli eventuali errori vengano catturati nella variabile `$output`.

Esecuzione del Comando e Gestione degli Errori

Esecuzione tramite `shell_exec()`: Il comando Certbot, completo di tutti i parametri e opzioni, viene eseguito con la funzione `shell_exec()`, che restituisce l'output (o l'eventuale messaggio d'errore).

Verifica dell'output: Il codice controlla la presenza di eventuali stringhe indicative di errore (come "failed" o "error") all'interno dell'output. Se viene rilevato un errore, la variabile \$error viene impostata su true.

Configurazione del Rinnovo Automatico tramite Systemd

Creazione di file di servizio e timer: Se l'opzione di rinnovo automatico è abilitata e non sono stati rilevati errori, lo script procede a:

Creare un file di servizio systemd che esegue il comando di rinnovo del certificato.

Creare un file di timer che programma l'esecuzione del comando ogni 45 giorni.

Scrittura dei file: I contenuti dei file vengono generati in modo dinamico (inclusendo il dominio) e salvati nelle directory di systemd (/etc/systemd/system/).

Abilitazione e avvio del timer: Dopo aver scritto i file, vengono eseguiti i comandi systemctl enable e systemctl start per rendere operativo il timer.

Gestione degli errori di sistema: Se durante la scrittura dei file o l'avvio del timer si verificano errori, il flag \$error viene aggiornato e il messaggio di output viene arricchito con ulteriori dettagli.

7. File CSS: (process.css)

```
/* Reset base */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

/* Stile generale del body */
body {
  font-family: 'Arial', sans-serif;
  background: linear-gradient(135deg, rgba(17, 17, 17, 0.9), rgba(34, 34, 34, 0.9));
  color: #ffffff;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  overflow: hidden;
  padding: 20px;
}

/* Stile del contenitore principale */
.container {
  background: rgba(255, 255, 255, 0.05);
  backdrop-filter: blur(10px);
  border-radius: 20px;
  box-shadow: 0 8px 20px rgba(0, 0, 0, 0.4);
  width: 100%;
  max-width: 600px;
  padding: 40px;
  text-align: center;
  animation: fadeIn 1s ease-in-out;
}

/* Stile del titolo */
h1 {
  font-size: 2rem;
  color: #ed1c24;
```

```

    margin-bottom: 20px;
    text-shadow: 2px 2px 10px rgba(237, 28, 36, 0.5);
}

/* Stile delle etichette */
label {
    font-size: 1rem;
    color: #ddd;
    text-align: left;
    display: block;
    margin-top: 15px;
}

/* Stile degli input */
input[type="text"], input[type="email"], select {
    width: 100%;
    padding: 12px;
    border: none;
    border-radius: 10px;
    background: rgba(255, 255, 255, 0.1);
    color: #ffffff;
    font-size: 1rem;
    transition: background 0.3s ease, transform 0.2s ease;
    margin-top: 5px;
}

input[type="text"]:focus, input[type="email"]:focus, select:focus {
    background: rgba(255, 255, 255, 0.2);
    outline: none;
    transform: scale(1.02);
}

/* Stile del pulsante */
button {
    background: linear-gradient(135deg, #ed1c24, #b31217);
    color: white;
    padding: 12px;
    border: none;
    border-radius: 10px;
    font-size: 1rem;
    cursor: pointer;
    transition: background 0.3s ease, transform 0.2s ease;
    margin-top: 20px;
}

button:hover {
    background: linear-gradient(135deg, #b31217, #ed1c24);
    transform: scale(1.05);
}

button:active {
    transform: scale(0.98);
}

/* Stile del risultato */
.result {
    margin-top: 20px;
    padding: 20px;
    border-radius: 10px;
    background: rgba(255, 255, 255, 0.1);
    text-align: left;
}

.result h2 {
    font-size: 1.5rem;
    margin-bottom: 10px;
}

.result.error {
    background: rgba(237, 28, 36, 0.2);
    border: 1px solid rgba(237, 28, 36, 0.5);
}

```



```

}

.result.success {
  background: rgba(34, 197, 94, 0.2);
  border: 1px solid rgba(34, 197, 94, 0.5);
}

.result pre {
  font-family: 'Courier New', monospace;
  font-size: 0.9rem;
  color: #ddd;
  white-space: pre-wrap;
  word-wrap: break-word;
}

.back {
  align-self: flex-start;
  margin-bottom: 20px;
  position: relative;
  background: linear-gradient(135deg, #ed1c24, #b31217);
  color: white;
  padding: 10px;
  border: none;
  border-radius: 10px;
  font-size: 5rem;
  cursor: pointer;
  transition: background 0.3s ease, transform 0.2s ease;
  margin-top: 20px;
}

.back:hover {
  background: linear-gradient(135deg, #b31217, #ed1c24);
  transform: scale(1.05);
}

/* Animazione di caricamento */
@keyframes fadeIn {
  from {
    opacity: 0;
    transform: translateY(-20px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
}

```

8. File PHP: (show_sites.php)

```

<?php
session_start();

// Gestisci la visualizzazione (Apache o Nginx)
if (isset($_GET['view'])) {
  $_SESSION['view'] = $_GET['view'];
}
$view = isset($_SESSION['view']) ? $_SESSION['view'] : 'apache';

// Percorso della cartella in base alla visualizzazione selezionata
$directory = ($view == 'nginx') ? '/etc/nginx/sites-available/' : '/etc/apache2/sites-available/';

// Funzione per verificare se il percorso è una cartella
if (is_dir($directory)) {

```

```

    $files = scandir($directory);
} else {
    die("La directory non esiste.");
}

// Funzione per visualizzare il contenuto del file
function viewFileContent($filePath) {
    if (file_exists($filePath)) {
        return htmlspecialchars(file_get_contents($filePath));
    } else {
        return "File non trovato.";
    }
}

// Funzione per salvare il contenuto del file
function saveFileContent($filePath, $content) {
    if (is_writable($filePath)) {
        file_put_contents($filePath, $content);
        return "File salvato con successo.";
    } else {
        return "Errore: Il file non è scrivibile.";
    }
}

// Funzione per creare un nuovo file
function createNewFile($filePath) {
    if (!file_exists($filePath)) {
        if (is_writable(dirname($filePath))) {
            file_put_contents($filePath, ""); // Crea un file vuoto
            return "File creato con successo.";
        } else {
            return "Errore: La directory non è scrivibile.";
        }
    } else {
        return "Il file esiste già.";
    }
}

// Funzione per rimuovere un file
function removeFile($filePath) {
    if (file_exists($filePath)) {
        if (is_writable($filePath)) {
            unlink($filePath);
            return "File rimosso con successo.";
        } else {
            return "Errore: Il file non è scrivibile.";
        }
    } else {
        return "Errore: Il file non esiste.";
    }
}

// Controlla se è stato selezionato un file
if (isset($_GET['file'])) {
    $selectedFile = $_GET['file'];
    $filePath = $directory . '/' . basename($selectedFile);
    $fileContent = viewFileContent($filePath);
}

// Controlla se è stata richiesta la rimozione di un file
if (isset($_GET['remove'])) {
    $fileToRemove = $_GET['remove'];
    $filePath = $directory . '/' . basename($fileToRemove);
    $message = removeFile($filePath);
    // Aggiorna l'elenco dei file dopo la rimozione
    $files = scandir($directory);
}

// Controlla se è stato inviato il modulo di modifica
if (isset($_POST['save'])) {
    $selectedFile = $_POST['file'];

```

```

    $filePath = $directory . '/' . basename($selectedFile);
    $newContent = $_POST['content'];
    $message = saveFileContent($filePath, $newContent);
    $fileContent = viewFileContent($filePath);
}

// Controlla se è stato inviato il modulo per creare un nuovo file
if (isset($_POST['create'])) {
    $newFileName = $_POST['new_file_name'];
    $newFilePath = $directory . '/' . basename($newFileName);
    $message = createNewFile($newFilePath);
}
?>
<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <title>Elenco File</title>
</head>
<link rel="stylesheet" href="show_sites.css"> <!-- Collegamento al file CSS -->

<body>

<div class="container">
    <a href="../index.php">
        
    </a>
    <h1>Elenco dei file in <?php echo htmlspecialchars($directory); ?></h1>

    <!-- Pulsante per cambiare visualizzazione -->
    <form method="get">
        <button type="submit" name="view" value="apache">Visualizza Apache</button>
        <button type="submit" name="view" value="nginx">Visualizza Nginx</button>
    </form>

    <?php if (count($files) > 2): // Controlla se ci sono file (escludendo '.' e '..') ?>
        <div class="table-wrapper">
            <table border="1" >
                <tr>
                    <th>Nome File</th>
                    <th>Visualizza</th>
                    <th>Rimuovi</th>
                </tr>
                <?php foreach ($files as $file): ?>
                    <?php if ($file != '.' && $file != '..'): ?>
                        <tr>
                            <td><?php echo htmlspecialchars($file); ?></td>
                            <td>
                                <a href="?file=<?php echo urlencode($file); ?>">Visualizza</a>
                            </td>
                            <td>
                                <a href="?remove=<?php echo urlencode($file); ?>" onclick="return
confirm('Sei sicuro di voler rimuovere questo file?');">Rimuovi</a>
                            </td>
                        </tr>
                    </tr>
                <?php endforeach; ?>
            </table>
        </div>
    <?php else: ?>
        <p>No Site attivi trovati.</p>
    <?php endif; ?>

    <?php if (isset($fileContent)): ?>
        <h2>Contenuto del file: <?php echo htmlspecialchars($selectedFile); ?></h2>
        <form method="post">
            <textarea class="table-wrapper" name="content" rows="20" cols="100"><?php echo
$fileContent; ?></textarea><br>
            <input type="hidden" name="file" value="<?php echo htmlspecialchars($selectedFile); ?>">

```

```

        <input type="submit" name="save" value="Salva">
    </form>
    <?php if (isset($message)): ?>
        <p><?php echo $message; ?></p>
    <?php endif; ?>
<?php endif; ?>

<h2>Crea un nuovo file di configurazione</h2>
<form method="post">
    <label for="new_file_name">Nome del nuovo file (es: nuovo_sito.conf):</label>
    <input type="text" name="new_file_name" id="new_file_name" required><br>
    <input type="submit" name="create" value="Crea">
</form>
<?php if (isset($message) && !isset($fileContent)): ?>
    <p><?php echo $message; ?></p>
<?php endif; ?>
<div>
</body>
</html>

```

Questo script PHP è progettato per gestire e visualizzare i file di configurazione dei siti web, sia per Apache che per Nginx, consentendo all'utente di visualizzare, modificare, creare e rimuovere file presenti nelle rispettive directory di configurazione.

Gestione della Visualizzazione della Directory

Il codice inizia con l'avvio di una sessione PHP e la gestione di un parametro GET denominato "view". Se l'utente passa questo parametro, il valore viene salvato nella sessione; in assenza di una scelta, la visualizzazione predefinita è impostata su "apache". In base al valore di "view", viene definita la variabile \$directory che punta alla cartella corretta:

Per Apache, la directory è /etc/apache2/sites-available/.

Per Nginx, invece, la directory è /etc/nginx/sites-available/.

Scansione e Visualizzazione dei File

Il codice verifica l'esistenza della directory con la funzione is_dir(). Se la directory esiste, viene eseguita la scansione utilizzando scandir(), che restituisce un array di file. In caso contrario, il programma termina con un messaggio di errore ("La directory non esiste."). I file trovati (escludendo i riferimenti speciali '.' e '..') vengono poi visualizzati in una tabella HTML, dove per ciascun file l'utente ha la possibilità di:

Visualizzarne il contenuto, tramite un link che passa il nome del file come parametro GET.

Rimuovere il file, con un ulteriore link che richiede conferma prima di eseguire l'azione.

Funzioni di Gestione dei File

Il codice definisce diverse funzioni per interagire con i file di configurazione:

viewFileContent(\$filePath): Controlla l'esistenza del file e, in caso affermativo, restituisce il contenuto del file convertito in formato sicuro (usando htmlspecialchars()), altrimenti segnala che il file non è stato trovato.

saveFileContent(\$filePath, \$content): Salva il contenuto passato nel file, previa verifica della scrivibilità del file. In caso di successo, restituisce un messaggio di conferma, altrimenti segnala l'errore.

createNewFile(\$filePath): Crea un nuovo file vuoto se questo non esiste già e se la directory di destinazione è scrivibile, restituendo un messaggio di successo o segnalando eventuali errori.

removeFile(\$filePath): Rimuove il file specificato se esiste e se è scrivibile, altrimenti restituisce un messaggio di errore.

Elaborazione delle Richieste

Il codice gestisce vari scenari in base ai parametri inviati dall'utente:

Visualizzazione del Contenuto di un File: Se viene passato un parametro GET "file", il file selezionato viene visualizzato usando la funzione viewFileContent(). Il contenuto viene poi mostrato all'interno di un modulo che permette all'utente di modificarlo e salvarlo.

Rimozione di un File: Se viene passato un parametro GET "remove", il file viene eliminato utilizzando la funzione removeFile(), e successivamente viene aggiornata la lista dei file con una nuova scansione della directory.

Modifica del Contenuto di un File: Se viene inviato un modulo POST con il campo "save", il nuovo contenuto viene salvato nel file selezionato tramite saveFileContent(), e il file viene riletto per aggiornare il contenuto visualizzato.

Creazione di un Nuovo File: Se viene inviato il modulo per creare un nuovo file (tramite il campo "create"), viene utilizzata la funzione createNewFile() per generare un file vuoto con il nome specificato dall'utente.

9. File CSS: (show_sites.css)

```
/* Reset base */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

/* Stile generale del body */
body {
  font-family: 'Arial', sans-serif;
  background: linear-gradient(135deg, rgba(17, 17, 17, 0.9), rgba(34, 34, 34, 0.9));
  color: #ffffff;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  overflow: hidden;
  padding: 20px;
}

/* Stile del contenitore principale */
.container {
  background: rgba(255, 255, 255, 0.05);
  backdrop-filter: blur(10px);
```

```

border-radius: 20px;
box-shadow: 0 8px 20px rgba(0, 0, 0, 0.4);
width: 100%;
max-width: 1000px;
max-height: 850px;
padding: 40px;
text-align: center;
animation: fadeIn 1s ease-in-out;
overflow: auto;
}

.container::-webkit-scrollbar {
  height: 8px; /* Altezza della scrollbar orizzontale */
}

.container::-webkit-scrollbar-thumb {
  background-color: rgba(255, 255, 255, 0.05); /* Colore del thumb (parte mobile) */
  border-radius: 10px; /* Arrotondamento degli angoli del thumb */
}

.container::-webkit-scrollbar-track {
  background: transparent; /* Colore del track (parte fissa) */
}

/* Stile del titolo */
h1 {
  font-size: 2rem;
  color: #ed1c24;
  margin-bottom: 20px;
  text-shadow: 2px 2px 10px rgba(237, 28, 36, 0.5);
}

/* Stile del sottotitolo */
h2 {
  font-size: 1.5rem;
  color: #ddd;
  margin-bottom: 15px;
  text-shadow: 1px 1px 5px rgba(255, 255, 255, 0.3);
}

/* Stile della tabella */
table {
  width: 100%;
  border-collapse: collapse;
  margin-top: 20px;
  background: rgba(255, 255, 255, 0.05);
  border-radius: 10px;
  overflow: hidden;
}

/* Stile delle celle della tabella */
th, td {
  padding: 12px;
  border: 1px solid rgba(255, 255, 255, 0.1);
  text-align: left;
  color: #ddd;
}

/* Stile dell'intestazione della tabella */
th {
  background: rgba(255, 255, 255, 0.1);
  color: #fff;
  font-weight: bold;
}

/* Stile delle righe pari */
tr:nth-child(even) {
  background: rgba(255, 255, 255, 0.05);
}

/* Stile dei link */

```

```

a {
  color: #ed1c24;
  text-decoration: none;
  font-weight: bold;
  transition: color 0.3s ease;
  position: relative; /* Aggiungi questa linea */
  left: -10px; /* Aggiungi questa linea */
}

a:hover {
  color: #b31217;
}

/* Stile dei pulsanti */
button, input[type="submit"] {
  background: linear-gradient(135deg, #ed1c24, #b31217);
  color: white;
  padding: 10px 20px;
  border: none;
  border-radius: 10px;
  font-size: 1rem;
  cursor: pointer;
  transition: background 0.3s ease, transform 0.2s ease;
  margin: 5px;
}

button:hover, input[type="submit"]:hover {
  background: linear-gradient(135deg, #b31217, #ed1c24);
  transform: scale(1.05);
}

button:active, input[type="submit"]:active {
  transform: scale(0.98);
}

/* Stile del form */
form {
  display: flex;
  flex-direction: column;
  gap: 15px;
  margin-top: 20px;
}

/* Stile delle etichette */
label {
  font-size: 1rem;
  color: #ddd;
  text-align: left;
  display: block;
}

textarea {
  height: 200px;
}

/* Stile degli input */
input[type="text"], textarea {
  width: 100%;
  padding: 12px;
  border: none;
  border-radius: 10px;
  background: rgba(255, 255, 255, 0.1);
  color: #ffffff;
  font-size: 1rem;
  transition: background 0.3s ease, transform 0.2s ease;
}

input[type="text"]:focus, textarea:focus {
  background: rgba(255, 255, 255, 0.2);
  outline: none;
  transform: scale(1.02);
}

```

```

}

/* Stile del messaggio di conferma/errore */
p {
  font-size: 1rem;
  color: #ddd;
  margin-top: 10px;
}

/* Contenitore della tabella con scrollbar orizzontale */
.table-wrapper {
  overflow-y: auto; /* Abilita la scrollbar orizzontale */
  max-width: 100%; /* Assicurati che il contenitore non ecceda il bordo del container */
}

/* Personalizzazione della scrollbar */
.table-wrapper::-webkit-scrollbar {
  height: 8px; /* Altezza della scrollbar orizzontale */
}

.table-wrapper::-webkit-scrollbar-thumb {
  background-color: rgba(255, 255, 255, 0.05); /* Colore del thumb (parte mobile) */
  border-radius: 10px; /* Arrotondamento degli angoli del thumb */
}

.table-wrapper::-webkit-scrollbar-track {
  background: transparent; /* Colore del track (parte fissa) */
}

/* Stile del pulsante "back" */
.back {
  align-self: flex-start; /* Lo sposta a sinistra */
  margin-bottom: 20px; /* Distanza dagli altri elementi */
  position: relative;
  left: -350px; /* Sposta l'elemento 20px più a sinistra */
  top: -30px;
  background: linear-gradient(135deg, #ed1c24, #b31217); /* Gradiente rosso */
  color: white; /* Testo bianco */
  padding: 10px; /* Spaziatura interna */
  border: none; /* Nessun bordo */
  border-radius: 10px; /* Bordi arrotondati */
  font-size: 5rem; /* Dimensione del testo */
  cursor: pointer; /* Cambia il cursore in una mano */
  transition: background 0.3s ease, transform 0.2s ease; /* Transizione fluida */
  margin-top: 20px; /* Spazio sopra il pulsante */
}

.back:hover {
  background: linear-gradient(135deg, #b31217, #ed1c24); /* Gradiente invertito */
  transform: scale(1.05); /* Effetto di ingrandimento */
}

/* Animazione di caricamento */
@keyframes fadeIn {
  from {
    opacity: 0;
    transform: translateY(-20px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
}

```


10. File PHP: (view_certificates.php)

```
<?php

/**
 * Questo script esegue "certbot certificates", ne acquisisce l'output
 * e mostra le informazioni principali in una semplice tabella HTML.
 */

$output = shell_exec('sudo certbot certificates 2>&1');

if (!$output) {

    // Gestione di un eventuale errore o output vuoto

    echo "Impossibile eseguire 'certbot certificates' o nessun output disponibile.";

    exit;

}

// Per debug, stampiamo l'output grezzo di certbot

// echo "<pre>$output</pre>";

// Suddividiamo l'output in righe

$lines = explode("\n", $output);

// Array per memorizzare i certificati trovati

$certificates = [];

$currentCert = [];

// Cerchiamo le righe significative

foreach ($lines as $line) {

    $line = trim($line);

    // Inizia un nuovo certificato

    if (strpos($line, 'Certificate Name:') === 0) {

        // Se esiste un certificato in corso di lettura, lo salviamo prima di iniziare il nuovo

        if (!empty($currentCert)) {

            $certificates[] = $currentCert;
```

```

    }

    // Avviamo un nuovo array di info per il certificato

    $currentCert = [

        'name' => substr($line, strlen('Certificate Name:'))

    ];

} elseif (strpos($line, 'Domains:') === 0) {

    $currentCert['domains'] = substr($line, strlen('Domains:'));

} elseif (strpos($line, 'Expiry Date:') === 0) {

    // L'output di certbot potrebbe contenere data, ora e stato di validità

    $currentCert['expiry_date'] = substr($line, strlen('Expiry Date:'));

} elseif (strpos($line, 'Certificate Path:') === 0) {

    $currentCert['cert_path'] = substr($line, strlen('Certificate Path:'));

} elseif (strpos($line, 'Private Key Path:') === 0) {

    $currentCert['key_path'] = substr($line, strlen('Private Key Path:'));

}

}

// Se l'ultimo certificato non è stato ancora aggiunto all'array
if (!empty($currentCert)) {

    $certificates[] = $currentCert;

}

?>

<!DOCTYPE html>

<html lang="it">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Certbot </title> <!-- Titolo della pagina -->

    <link rel="stylesheet" href="view_certificates.css"> <!-- Collegamento al file CSS -->

</head>

<body>

```

```

<div class="container">

    <a href="../../../index.php">

    </a>

    <h1>Lista dei Certificati Gestiti da Certbot</h1>

    <?php if (empty($certificates)): ?>

        <p>Nessun certificato trovato.</p>

    <?php else: ?>

        <div class="table-wrapper">

            <table>

                <thead>

                    <tr>

                        <th>Nome Certificato</th>

                        <th>Domini</th>

                        <th>Data Scadenza</th>

                        <th>Percorso Certificato</th>

                        <th>Percorso Chiave Privata</th>

                        <th>Azioni</th>

                    </tr>

                </thead>

                <tbody>

                    <?php foreach ($certificates as $cert): ?>

                        <tr>

                            <td><?php echo htmlentities($cert['name']) ?? ''; ?></td>

                            <td><?php echo htmlentities($cert['domains']) ?? ''; ?></td>

                            <td><?php echo htmlentities($cert['expiry_date']) ?? ''; ?></td>

                            <td><?php echo htmlentities($cert['cert_path']) ?? ''; ?></td>

                            <td><?php echo htmlentities($cert['key_path']) ?? ''; ?></td>

```

```

                <td>

                    <a href="operation/renew.php?domain=?php echo
urlencode($cert['domains']); ?>">Rinnova</a>

                    <a href="operation/delete.php?domain=?php echo
urlencode($cert['domains']); ?>">Elimina</a>

                    <a href="operation/download.php?cert_path=?php echo
urlencode($cert['cert_path']); ?>&key_path=?php echo urlencode($cert['key_path']); ?>">Scarica</a>

                </td>

            </tr>

        <?php endforeach; ?>

    </tbody>

</table>

</div>

<?php endif; ?>

</div>

</body>

</html>

```

Il comando `certbot certificates` viene eseguito tramite `shell_exec()`, acquisendo l'output.

Se il comando non produce output, viene mostrato un messaggio di errore e lo script si interrompe.

Se non ci sono certificati, viene mostrato il messaggio "Nessun certificato trovato".

Se ci sono certificati, vengono elencati in una tabella con le seguenti colonne:

- Nome del certificato
- Domini associati
- Data di scadenza
- Percorsi dei file del certificato e della chiave privata
- Azioni disponibili: **rinnovare o eliminare** un certificato.

Ogni certificato ha due link:

- **Rinnova** → Chiama `renew.php` passando il dominio come parametro GET.
- **Elimina** → Chiama `delete.php` con lo stesso parametro.
- I file `renew.php` e `delete.php` devono essere implementati separatamente per gestire il rinnovo o la rimozione dei certificati.

11.File CSS: (view_certificates.css)

```
/* Reset base */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

/* Stile generale del body */
body {
  font-family: 'Arial', sans-serif;
  background: linear-gradient(135deg, rgba(17, 17, 17, 0.9), rgba(34, 34, 34, 0.9));
  color: #ffffff;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  overflow: hidden;
  padding: 250px;
}

/* Stile del contenitore principale */
.container {
  background: rgba(255, 255, 255, 0.05);
  backdrop-filter: blur(10px);
  border-radius: 20px;
  box-shadow: 0 8px 20px rgba(0, 0, 0, 0.4);
  width: 100%;
  padding: 40px;
  text-align: center;
  animation: fadeIn 1s ease-in-out;
  overflow: hidden;
}

/* Stile del pulsante "back" */
.back {
  align-self: flex-start; /* Lo sposta a sinistra */
  margin-bottom: 20px; /* Distanza dagli altri elementi */
  position: relative;
  left: -650px; /* Sposta l'elemento 20px più a sinistra */
  top: -30px;
  background: linear-gradient(135deg, #ed1c24, #b31217); /* Gradiente rosso */
  color: white; /* Testo bianco */
  padding: 10px; /* Spaziatura interna */
  border: none; /* Nessun bordo */
  border-radius: 10px; /* Bordi arrotondati */
  font-size: 5rem; /* Dimensione del testo */
  cursor: pointer; /* Cambia il cursore in una mano */
  transition: background 0.3s ease, transform 0.2s ease; /* Transizione fluida */
  margin-top: 20px; /* Spazio sopra il pulsante */
}

.back:hover {
  background: linear-gradient(135deg, #b31217, #ed1c24); /* Gradiente invertito */
  transform: scale(1.05); /* Effetto di ingrandimento */
}

/* Contenitore della tabella con scrollbar orizzontale */
.table-wrapper {
  overflow-x: auto; /* Abilita la scrollbar orizzontale */
  max-width: 100%; /* Assicurati che il contenitore non ecceda il bordo del container */
}

/* Personalizzazione della scrollbar */
```

```

.table-wrapper::-webkit-scrollbar {
    height: 8px; /* Altezza della scrollbar orizzontale */
}

.table-wrapper::-webkit-scrollbar-thumb {
    background-color: rgba(255, 255, 255, 0.05); /* Colore del thumb (parte mobile) */
    border-radius: 10px; /* Arrotondamento degli angoli del thumb */
}

.table-wrapper::-webkit-scrollbar-track {
    background: transparent; /* Colore del track (parte fissa) */
}

/* Stile del titolo */
h1 {
    font-size: 2.5rem;
    color: #ed1c24; /* Nuovo colore rosso */
    margin-bottom: 20px;
    text-shadow: 2px 2px 10px rgba(237, 28, 36, 0.5); /* Ombra del testo adattata al nuovo colore */
}

/* Stile della tabella */
table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 20px;
    overflow-x: auto; /* Abilita la scrollbar orizzontale */
    max-width: 600px;
}

/* Stile delle celle della tabella */
th, td {
    padding: 12px;
    border: 1px solid rgba(255, 255, 255, 0.1);
    text-align: left;
    color: #ddd;
}

/* Stile dell'intestazione della tabella */
th {
    background: rgba(255, 255, 255, 0.1);
    color: #fff;
    font-weight: bold;
}

/* Stile delle righe pari */
tr:nth-child(even) {
    background: rgba(255, 255, 255, 0.05);
}

/* Stile dei link */
a {
    color: #ed1c24; /* Nuovo colore rosso */
    text-decoration: none;
    font-weight: bold;
    transition: color 0.3s ease;
}

a:hover {
    color: #b31217; /* Tonalità più scura di rosso */
}

.btn {
    background: linear-gradient(135deg, #ed1c24, #c21a20); /* Gradiente rosso */
    color: white;
    padding: 12px;
    border: none;
    border-radius: 10px;
    font-size: 1rem;
    cursor: pointer;
    transition: background 0.3s ease, transform 0.2s ease;
}

```

```

    text-decoration: none;
    text-align: center;
}

/* Effetto hover sui pulsanti */
.btn:hover {
    background: linear-gradient(135deg, #c21a20, #ed1c24); /* Gradiente invertito */
    transform: scale(1.05);
}

/* Effetto click sui pulsanti */
.btn:active {
    transform: scale(0.98);
}

/* Animazione di caricamento */
@keyframes fadeIn {
    from {
        opacity: 0;
        transform: translateY(-20px);
    }
    to {
        opacity: 1;
        transform: translateY(0);
    }
}

```

12.File PHP: (delete.php)

```

<html lang="it">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dashboard Certbot</title>
    <link rel="stylesheet" href="../../createCertificate/processCertificate/process.css"> <!--
Collegamento al file CSS -->
</head>
<body>
    <div class="container">
        <h1>Esito Certbot </h1><br><br><br>
        <p>
            <?php
            /**
             * delete.php
             * Elimina il certificato associato a un determinato dominio utilizzando Certbot.
             */

            // Recupera il dominio dalla query string
            if (!isset($_GET['domain'])) {
                echo "Dominio non specificato.";
                exit;
            }

            $domain = trim($_GET['domain']);

            // Esegui il comando per eliminare il certificato
            // '--cert-name' specifica il certificato da eliminare
            $command = "sudo certbot delete --cert-name {$domain} --non-interactive 2>&1";
            $output = shell_exec($command);

            // Verifica l'output e fornisci un feedback
            // L'uscita tipica in caso di successo non è particolarmente chiara; si può cercare
            // stringhe specifiche per verificare la riuscita, o semplicemente mostrare l'output
            if (empty($output)) {
                // Se l'output è vuoto, cerchiamo di dare un messaggio di conferma

```

```

        echo "Certificato per il dominio {$_GET['domain']} eliminato (o non presente).";
    } else {
        echo "Risultato dell'eliminazione del certificato per il dominio
{$_GET['domain']}:<br>";
        echo nl2br(htmlentities($output));
    }
}
?>
</p><br><br><br>
<a href="../../index.php">
    
</img>
</a>
</div>
</body>
</html>

```

Lo script delete.php è un file PHP che elimina un certificato SSL associato a un dominio utilizzando Certbot.

1. **Recupero del dominio:** Controlla se è stato passato un dominio tramite la query string (\$_GET['domain']). Se il parametro non è presente, viene mostrato un messaggio di errore e lo script si interrompe.
2. **Esecuzione del comando Certbot:** Se il dominio è presente, lo script costruisce ed esegue il comando `sudo certbot delete --cert-name {dominio} --non-interactive`, che elimina il certificato associato al dominio specificato. L'output del comando viene catturato con `shell_exec()`.
3. **Gestione della risposta:** Se l'output è vuoto, significa che il certificato è stato eliminato o non esisteva. Altrimenti, viene mostrato il risultato dell'operazione.
4. **Interfaccia grafica:** Lo script genera una pagina HTML con il risultato dell'operazione e un'icona per tornare alla pagina principale (index.php).

13.File PHP: (renew.php)

```

<html lang="it">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dashboard Certbot</title>
    <link rel="stylesheet" href="../../createCertificate/processCertificate/process.css"> <!--
Collegamento al file CSS -->
  </head>
  <body>
    <div class="container">
      <h1>Esito Certbot </h1><br><br><br>
      <p>
        <?php
        /**
         * renew.php
         * Rinnova il certificato associato a un determinato dominio utilizzando Certbot.
         */

        // Recupera il dominio dalla query string e rimuove gli spazi non necessari

```



```

        if (!isset($_GET['domain']) || trim($_GET['domain']) === '') {
            echo "Dominio non specificato.";
            exit;
        }

        $domain = trim($_GET['domain']);

        // Esegui il comando per rinnovare il certificato
        // Con l'opzione --force-renewal forza il rinnovo anche se non necessario.
        // Utilizza '--cert-name' per indicare il nome del certificato,
        // che di solito corrisponde a uno dei domini configurati.
        $command = "sudo certbot renew --cert-name {$domain} --force-renewal 2>&1";

        // Esegui il comando
        $output = shell_exec($command);

        // Verifica l'output e fornisci un feedback
        if (strpos($output, 'Congratulations') !== false) {
            echo "Certificato per il dominio {$_GET['domain']} rinnovato con successo.";
        } else {
            echo "Errore durante il rinnovo del certificato per il dominio
{$_GET['domain']}:<br>";
            echo nl2br(htmlentities($output));
        }
    }
}
?>
</p><br><br><br>
<a href="../../../index.php">
    
    </img>
</a>
</div>
</body>
</html>

```

Il file `renew.php` è uno script PHP che gestisce il rinnovo di un certificato SSL per un dominio specificato tramite Certbot. Ecco un riassunto delle sue funzionalità:

1. **Recupero del dominio:** Lo script inizia controllando se è stato fornito un dominio tramite la query string (`$_GET['domain']`). Se non è presente o se è vuoto, viene visualizzato un messaggio di errore e l'esecuzione termina.
2. **Esecuzione del comando di rinnovo:** Se il dominio è valido, lo script costruisce ed esegue il comando `sudo certbot renew --cert-name {dominio} --force-renewal`. Questa istruzione forza il rinnovo del certificato anche se non è scaduto, utilizzando il nome del certificato associato al dominio.
3. **Gestione della risposta:** L'output del comando viene catturato e analizzato. Se il comando restituisce una risposta contenente la parola "Congratulations", significa che il rinnovo è avvenuto con successo, e viene mostrato un messaggio di conferma. In caso contrario, viene visualizzato un messaggio di errore con i dettagli dell'output.
4. **Interfaccia utente:** Infine, lo script genera una pagina HTML con il risultato del rinnovo e include un'icona che permette di tornare alla pagina principale (`index.php`).

14.File PHP: (download.php)

```
<?php
if (isset($_GET['cert_path']) && isset($_GET['key_path'])) {
    $cert_path = escapeshellarg(trim($_GET['cert_path']));
    $key_path = escapeshellarg(trim($_GET['key_path']));
    $tempDir = sys_get_temp_dir();
    $cert_tempFile = tempnam($tempDir, 'cert_');
    $key_tempFile = tempnam($tempDir, 'key_');

    // Usa sudo per copiare i file
    $cert_command = "sudo cp $cert_path $cert_tempFile";
    $key_command = "sudo cp $key_path $key_tempFile";
    $cert_output = [];
    $key_output = [];
    $cert_return_var = 0;
    $key_return_var = 0;
    exec($cert_command, $cert_output, $cert_return_var);
    exec($key_command, $key_output, $key_return_var);

    // Debug: stampa di debug nella console del browser
    echo "<script>console.log('Cert command: " . addslashes($cert_command) . "');</script>";
    echo "<script>console.log('Cert output: " . addslashes(implode('\n', $cert_output)) .
    "');</script>";
    echo "<script>console.log('Cert return var: $cert_return_var');</script>";

    echo "<script>console.log('Key command: " . addslashes($key_command) . "');</script>";
    echo "<script>console.log('Key output: " . addslashes(implode('\n', $key_output)) . "');</script>";
    echo "<script>console.log('Key return var: $key_return_var');</script>";

    if ($cert_return_var === 0 && $key_return_var === 0 && file_exists($cert_tempFile) &&
    file_exists($key_tempFile)) {
        // Estrai il nome del dominio dal percorso del certificato
        $domain = basename(dirname(trim($_GET['cert_path'])));

        // Imposta gli header per il download dei file
        header('Content-Type: application/zip');
        header("Content-Disposition: attachment; filename=\"$domain.zip\"");
        header('Pragma: no-cache');
        header('Expires: 0');

        // Crea un file zip temporaneo
        $zip = new ZipArchive();
        $zipFile = tempnam($tempDir, 'zip_');
        if ($zip->open($zipFile, ZipArchive::CREATE) === TRUE) {
            $zip->addFile($cert_tempFile, basename($_GET['cert_path']));
            $zip->addFile($key_tempFile, basename($_GET['key_path']));
            $zip->close();

            // Leggi il contenuto del file zip e invialo come download
            readfile($zipFile);

            // Elimina i file temporanei
            unlink($cert_tempFile);
            unlink($key_tempFile);
            unlink($zipFile);
        } else {
            echo "Impossibile creare il file zip.";
        }
        exit;
    } else {
        echo "Copia dei file fallita o file temporaneo non trovato.";
    }
} else {
    echo "Percorsi dei file non specificati.";
}
?>
```

Questo script PHP permette di scaricare un file ZIP contenente un certificato e una chiave privata, specificati tramite i parametri cert_path e key_path nell'URL.

15.File PHP: (remote_certificate.php)

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Certbot Command Runner</title>
  <link rel="stylesheet" href="index.css"> <!-- Collegamento al file CSS -->
</head>

<body>
  <div class="container">
    <!-- Pulsante per tornare alla pagina principale -->
    <a href="index.php">
      
    </a>
    <h1>Remote Certbot</h1>

    <!-- Input per il dominio e l'email -->
    <div class="input-group">
      <label for="domain">Domain:</label>
      <input type="text" id="domain" name="domain" placeholder="es: example.com"
required><br><br><br>
      <label for="email">Email:</label>
      <input type="text" id="email" name="email" placeholder="es: prova@email.com" required>
    </div>

    <!-- Pulsanti per eseguire e fermare Certbot -->
    <div class="input-group">
      <button id="runButton" class="btn">Run Certbot</button>
      <button id="stopButton" class="btn">Stop Certbot</button>
      <button id="toggleOutput" class="btn">Show Output</button><br><br>
    </div>

    <!-- Checkbox per il test della richiesta certificato -->
    <div class="test">
      <input type="checkbox" id="testCheckbox" name="test">
      <label for="testCheckbox"></label>
      <span>Test richiesta certificato</span>
    </div>

    <!-- Area per messaggi di stato -->
    <div id="status-messages"></div>

    <!-- Output dei log di Certbot -->
    <div id="output" class="input-group"></div><br>

    <!-- Visualizzazione delle informazioni DNS TXT -->
    <div class="output-name-value">
      <h2 for="domain" id="txtName"></h2>
      <h2 for="domain" id="txtValue"></h2>
    </div><br>

    <!-- Pulsante per verificare DNS TXT -->
    <button id="pressEnter" class="action-button btn">Verifica DNS TXT</button>
```

```

</div>

<script>
  let eventSource;
  let waitingForEnter = false;

  // Avvia Certbot con i dati inseriti
  document.getElementById('runButton').addEventListener('click', function () {
    const domain = document.getElementById('domain').value;
    const email = document.getElementById('email').value;

    if (!domain || !email) {
      alert('Please enter both domain and email');
      return;
    }

    const outputDiv = document.getElementById('output');
    outputDiv.innerHTML = 'Running...\n';

    // Creazione della richiesta a remote_process.php con i parametri
    const testCheckbox = document.getElementById('testCheckbox');
    eventSource = new EventSource('remote_process.php?domain=' + encodeURIComponent(domain) +
      '&email=' + encodeURIComponent(email) +
      '&test=' + testCheckbox.checked);

    // Gestione delle risposte dal server
    eventSource.onmessage = function (event) {
      outputDiv.innerHTML += event.data + '\n';
      outputDiv.scrollTop = outputDiv.scrollHeight;

      // Estrazione di informazioni DNS TXT
      if (event.data.startsWith('Name:')) {
        document.getElementById('txtName').textContent = event.data;
      }
      if (event.data.startsWith('Value:')) {
        document.getElementById('txtValue').textContent = event.data;
      }

      // Messaggi di stato in base all'output
      let messageState = '';
      if (event.data.includes('Certbot failed to authenticate') ||
event.data.includes('[ERROR] Some challenges have failed')) {
        messageState = '<div class="error-message">La verifica del DNS non è andata a buon
fine</div>';
      } else if (event.data.includes('Successfully')) {
        messageState = '<div class="success-message">Certificato generato con
successo</div>';
      } else if (event.data.includes('existing certificate')) {
        messageState = '<div class="warning-message">Certificato con questo nome o Dominio
già esistente</div>';
      }

      if (messageState) {
        document.getElementById('status-messages').innerHTML = messageState;
      }

      // Mostra il pulsante per confermare il record DNS
      if (event.data.includes('[DNS TXT Record Details]') ||
event.data.includes('Please configure this DNS TXT record')) {
        document.getElementById('pressEnter').style.display = 'inline-block';
        waitingForEnter = true;
      }
    };

    eventSource.onerror = function () {
      outputDiv.innerHTML += '\n[ERROR] An error occurred while running the command.\n';
      eventSource.close();
    };
  });

  // Interrompe l'esecuzione di Certbot

```

```

        document.getElementById('stopButton').addEventListener('click', function () {
            if (eventSource) {
                eventSource.close();
                document.getElementById('output').innerHTML += '\n[INFO] Command execution stopped by
user.\n';
            }
        });

        // Invia comando di conferma per il DNS TXT
        document.getElementById('pressEnter').addEventListener('click', function () {
            if (waitingForEnter) {
                fetch('sender_enter.php')
                    .then(response => response.json())
                    .then(data => {
                        if (data.status === 'success') {
                            document.getElementById('output').innerHTML += '[INFO] Continuing with
certificate verification...\n';
                            document.getElementById('pressEnter').style.display = 'none';
                            waitingForEnter = false;
                        } else {
                            document.getElementById('output').innerHTML += '[ERROR] ' + data.message +
'\n';
                        }
                    })
                    .catch(error => {
                        document.getElementById('output').innerHTML += '[ERROR] Failed to send Enter
command: ' + error + '\n';
                    });
            }
        });

        // Mostra/nasconde l'output
        document.getElementById('toggleOutput').addEventListener('click', function() {
            const outputDiv = document.getElementById('output');
            if (outputDiv.classList.contains('show')) {
                outputDiv.classList.remove('show');
                this.textContent = 'Show Output';
            } else {
                outputDiv.classList.add('show');
                this.textContent = 'Hide Output';
            }
        });
    </script>
</body>
</html>

```

Il codice HTML e JavaScript crea un'interfaccia web per eseguire il comando **Certbot** su un server remoto.

16.File PHP: (remote_process.php)

```

<?php

// Imposta gli header per il protocollo Server-Sent Events (SSE)
header('Content-Type: text/event-stream');
header('Cache-Control: no-cache');
header('Connection: keep-alive');

// Funzione per inviare un messaggio al client
function send_message($message)
{
    echo "data: {$message}\n\n";
}

```

```

    ob_flush();
    flush();
}

// Funzione per validare un dominio
function validateDomain($domain) {
    return filter_var($domain, FILTER_VALIDATE_DOMAIN, FILTER_FLAG_HOSTNAME);
}

// Funzione per eseguire il comando Certbot
function runCertbotCommand($domain, $email, $test)
{
    // Termina eventuali processi Certbot già in esecuzione
    exec("pkill -f certbot");

    // Verifica la validità del dominio
    if (!validateDomain($domain)) {
        send_message("[ERROR] Invalid domain format");
        return false;
    }

    // Specifica i descrittori per la gestione di input/output ed errori del processo
    $descriptorspec = [
        0 => ["pipe", "r"], // stdin
        1 => ["pipe", "w"], // stdout
        2 => ["pipe", "w"]  // stderr
    ];

    // Imposta un timeout ragionevole
    set_time_limit(300); // 5 minuti di timeout

    // Avvia il processo Certbot con i parametri specificati
    $process = proc_open("sudo certbot certonly $test --email $email --agree-tos --manual --preferred-
challenges dns -d " . escapeshellarg($domain), $descriptorspec, $pipes);

    if (!is_resource($process)) {
        send_message("[ERROR] Failed to start the process");
        return false;
    }

    // Imposta i flussi di output e errori come non bloccanti
    stream_set_blocking($pipes[1], false);
    stream_set_blocking($pipes[2], false);

    // Assegna i flussi a variabili
    $stdin = $pipes[0];
    $stdout = $pipes[1];
    $stderr = $pipes[2];

    // Inizializza buffer e altre variabili di supporto
    $buffer = '';
    $error_buffer = '';
    $complete_output = '';
    $start_time = time();
    $timeout = 300; // Timeout di 5 minuti

    // Legge l'output del comando Certbot
    while ((!feof($stdout) || !feof($stderr)) && (time() - $start_time < $timeout)) {
        $read = [$stdout, $stderr];
        $write = null;
        $except = null;

        if (stream_select($read, $write, $except, 1) === false) {
            break;
        }

        foreach ($read as $stream) {
            if ($stream === $stdout) {
                $content = fgets($stdout);
                if ($content !== false) {
                    $complete_output .= $content;
                }
            }
        }
    }
}

```

```

        $buffer .= $content;

        if (strpos($buffer, "\n") !== false) {
            send_message(trim($buffer));
            $buffer = '';
        }
    }
} elseif ($stream === $stderr) {
    $content = fgets($stderr);
    if ($content !== false) {
        $error_buffer .= $content;
        if (trim($error_buffer) !== '') {
            send_message("[ERROR] " . trim($error_buffer));
            $error_buffer = '';
        }
    }
}
}

// Gestione dell'output per la configurazione del record DNS TXT
if (strpos($complete_output, '_acme-challenge') !== false && strpos($complete_output, 'with the
following value:') !== false) {
    $lines = explode("\n", $complete_output);
    $record_name = '';
    $record_value = '';

    foreach ($lines as $line) {
        $line = trim($line);
        if (strpos($line, '_acme-challenge') !== false && empty($record_name)) {
            $record_name = $line;
        } else if (!empty($record_name) && !empty($line) && strpos($line, '_acme-challenge')
=== false && strpos($line, 'with the following value:') === false) {
            $record_value = $line;
            break;
        }
    }

    if (!empty($record_name) && !empty($record_value)) {
        send_message("[DNS TXT Record Details]");
        send_message("Name: " . $record_name);
        send_message("Value: " . $record_value);
        send_message("[INFO] Please configure this DNS TXT record and press Enter when
ready...");

        // Attende l'input dell'utente
        while (!file_exists('/tmp/certbot_input') ||
trim(file_get_contents('/tmp/certbot_input')) === '') {
            usleep(100000); // Sleep per 0.1 secondi
            if (time() - $start_time >= $timeout) {
                send_message("[ERROR] Timeout waiting for user input");
                break;
            }
        }

        if (file_exists('/tmp/certbot_input')) {
            send_message("[INFO] Processing DNS verification...");
            fwrite($stdin, "\n");
            file_put_contents('/tmp/certbot_input', '');
            $complete_output = '';
        }
    }
}

// Chiude i flussi e termina il processo
foreach ($pipes as $pipe) {
    if (is_resource($pipe)) {
        fclose($pipe);
    }
}
}

```

```

$return_value = proc_close($process);
send_message("Command returned: $return_value");
return $return_value === 0;
}

// Gestisce i parametri GET per avviare Certbot
if (isset($_GET['domain']) && isset($_GET['email'])) {
    $domain = trim($_GET['domain']);
    $email = trim($_GET['email']);
    $test = isset($_GET['test']) && $_GET['test'] === 'true' ? '--dry-run' : '';

    send_message("Domain received: $domain");
    runCertbotCommand($domain, $email, $test);
} else {
    send_message("No domain provided.");
}

?>

```

Il codice PHP che hai condiviso implementa un server SSE (*Server-Sent Events*) per eseguire Certbot in modalità manuale con sfida DNS. Ecco un'analisi dettagliata del codice:

17.File PHP: (sender_enter.php)

```

<?php
// Imposta l'header della risposta come JSON
header('Content-Type: application/json');

try {
    // Scrive il valore '1' nel file temporaneo '/tmp/certbot_input'
    $result = file_put_contents('/tmp/certbot_input', '1');

    // Se il risultato è false, significa che la scrittura del file è fallita
    if ($result === false) {
        throw new Exception("Failed to write to input file");
    }

    // Se la scrittura è avvenuta con successo, restituisce una risposta JSON di successo
    echo json_encode([
        'status' => 'success',
        'message' => 'Enter key signal sent'
    ]);
} catch (Exception $e) {
    // In caso di errore, imposta il codice di risposta HTTP a 500 (errore interno del server)
    http_response_code(500);

    // Restituisce una risposta JSON con il messaggio di errore
    echo json_encode([
        'status' => 'error',
        'message' => $e->getMessage()
    ]);
}

```

Questo script PHP scrive il valore '1' nel file /tmp/certbot_input e restituisce una risposta JSON.

18.File SH: (setup.sh)

```
#!/bin/bash

# ESEGUIRE QUESTO PRIMA DI AVVIARE LO SCRIPT: chmod +x setup.sh

# Funzione per richiedere l'input dell'utente con convalida
ask_user() {
    local prompt="$1"
    local input

    while true; do
        read -p "$prompt" input
        if [[ "$input" == "s" || "$input" == "n" ]]; then
            echo "$input"
            break
        else
            echo "Risposta non valida. Inserisci 's' per sì o 'n' per no."
        fi
    done
}

# Chiedi all'utente se desidera installare Apache
install_apache=$(ask_user "Vuoi installare Apache? (s/n): ")

# Se l'utente desidera installare Apache
if [[ "$install_apache" == "s" ]]; then

    # Installa Apache e PHP
    sudo apt update
    sudo apt install -y apache2
    sudo apt install -y php libapache2-mod-php
    sudo apt install -y php-mysql php-xml php-mbstring php-cli php-cgi php-fpm

    # Cambia il proprietario delle directory di configurazione di Apache
    sudo chown -R www-data:www-data /etc/apache2/sites-available/
    sudo chmod -R 755 /etc/apache2/sites-available/
fi

# Chiedi all'utente se desidera installare Nginx
install_nginx=$(ask_user "Vuoi installare Nginx? (s/n): ")

# Se l'utente desidera installare Nginx
if [[ "$install_nginx" == "s" ]]; then

    # Installa Nginx e PHP-FPM
    sudo apt update
    sudo apt install -y nginx
    sudo apt install -y php-fpm

    # Cambia il proprietario delle directory di configurazione di Nginx
    sudo chown -R www-data:www-data /etc/nginx/sites-available/
    sudo chmod -R 755 /etc/nginx/sites-available/
fi

# Aggiungi regole a sudoers per www-data
echo "www-data ALL=(ALL) NOPASSWD: /usr/bin/certbot" | sudo tee -a /etc/sudoers
echo "www-data ALL=(ALL) NOPASSWD: /bin/cp /etc/letsencrypt/live/*" | sudo tee -a /etc/sudoers

# Installa Python 3, Certbot
sudo apt install -y python3 python3-venv libaugeas0

# Crea un ambiente virtuale per Certbot
sudo python3 -m venv /opt/certbot/

# Aggiorna pip e installa Certbot
sudo /opt/certbot/bin/pip install --upgrade pip
sudo /opt/certbot/bin/pip install certbot certbot-apache certbot-nginx
```

```
# Crea un link simbolico per Certbot
sudo ln -s /opt/certbot/bin/certbot /usr/bin/certbot

# Permette all'utente www-data di eseguire routine di rinnovo
sudo chown www-data:www-data /etc/systemd/system/
sudo chmod 775 /etc/systemd/system/

echo "Installazione eseguita con successo."
```

Conclusioni

Questa GUI semplifica la gestione dei certificati SSL/TLS tramite Certbot, offrendo un'interfaccia intuitiva che rende il processo accessibile anche agli utenti meno esperti. Grazie all'integrazione di HTML, CSS, JavaScript e PHP, l'applicazione fornisce un'esperienza fluida e funzionale.

Possibili Miglioramenti Futuri:

- Implementare un sistema di notifiche automatiche per avvisare gli utenti della prossima scadenza dei certificati.
- Ampliare il supporto per configurazioni avanzate e ulteriori server web, migliorando la compatibilità con diverse infrastrutture.